

**PIXI
LAB**

Blocks[®]

ВЕРСИЯ 6

СОДЕРЖАНИЕ

1. Знакомство с PIXILAB Blocks _____	4	Widget (виджет)_____	56
2. Начало работы_____	6	Reference (ссылка)_____	56
Mac _____	6	Live Video(живое видео)_____	57
Windows PC _____	7	Subtitle (субтитры)_____	58
Доступ к редактору _____	8	Schedule (расписание)_____	59
Linux _____	10	6. Поведение (behaviors) _____	61
3. Основные понятия _____	12	Appears (появление)_____	61
Споты _____	12	Disappears (исчезание)_____	61
Блоки _____	12	Move (перемещение)_____	61
Управляемые подсистемы _____	13	Mute (без звука)_____	62
Задачи _____	14	Opacity (непрозрачность)_____	62
4. Споты _____	15	Play _____	62
Настройки спота _____	16	Playing (воспроизведение)_____	63
Дисплей-спот _____	16	Press (нажатие)_____	63
Гостевой спот (visitor)_____	22	Reveal (раскрывать)_____	63
Location (местоположение)_____	24	Rotate (поворот)_____	63
WATCHOUTспот _____	25	Scale (масштаб)_____	64
Группа спотов _____	25	Speed (скорость)_____	64
5. Блоки _____	27	Volume (громкость)_____	64
Защита блока и шаблоны _____	30	Classify (применение классов)_____	64
Slideshow _____	30	7. Управление и панели _____	65
Composition _____	33	Button (кнопка)_____	66
Book (книга)_____	35	Indicator _____	70
Дочерняя репликация _____	36	Slider (ползунок)_____	70
Stack (штабель)_____	37	Bar (полоса)_____	70
Scroller (прокрутка)_____	38	Joystick _____	70
Grid (сетка)_____	39	Text _____	70
Attractor _____	39	Text Input (ввод текста)_____	71
Tag Selector _____	41	QR Code _____	71
Media File _____	42	QR Scanner _____	72
Media URL _____	43	8. Параметры спота _____	73
Camera _____	43	Интерактивность и поведение _____	75
Panorama _____	44	Пользовательские свойства спота _____	75
3D _____	45	Использование путей к свойству _____	75
Text _____	47	9. PIXILAB Player _____	77
Locator _____	50	10. Коллекция гостевых данных _____	78
Synchronizer _____	53		
Web _____	55		

СОДЕРЖАНИЕ

11. Управление _____	79
Пользователи _____	79
Лицензии _____	79
Mirroring Service _____	80
Server Status _____	80
Модули Modbus I/O _____	81
Сетевые устройства _____	83
Свет: Art-Net/DMX-512 _____	84
Кластеры WATCHOUT _____	86
12. Задача _____	87
Realm (область) _____	87
Group (группа) _____	88
Task (задача) _____	89
Операторы _____	92
13. Свойства и функции _____	101
Дисплей-спот _____	101
Гостевой спот _____	103
Местоположение _____	103
Группа спотов _____	103
Задача _____	103
Канал Modbus _____	104
Art-Net/DMX _____	104
WATCHOUT _____	104
Расписание _____	107
Network (сеть) _____	107
14. Авторизация и роли _____	108
Настраиваемые ограничения доступа _____	109
A. Файловая структура сервера _____	113
B. Сценарии и драйверы устройств _____	114

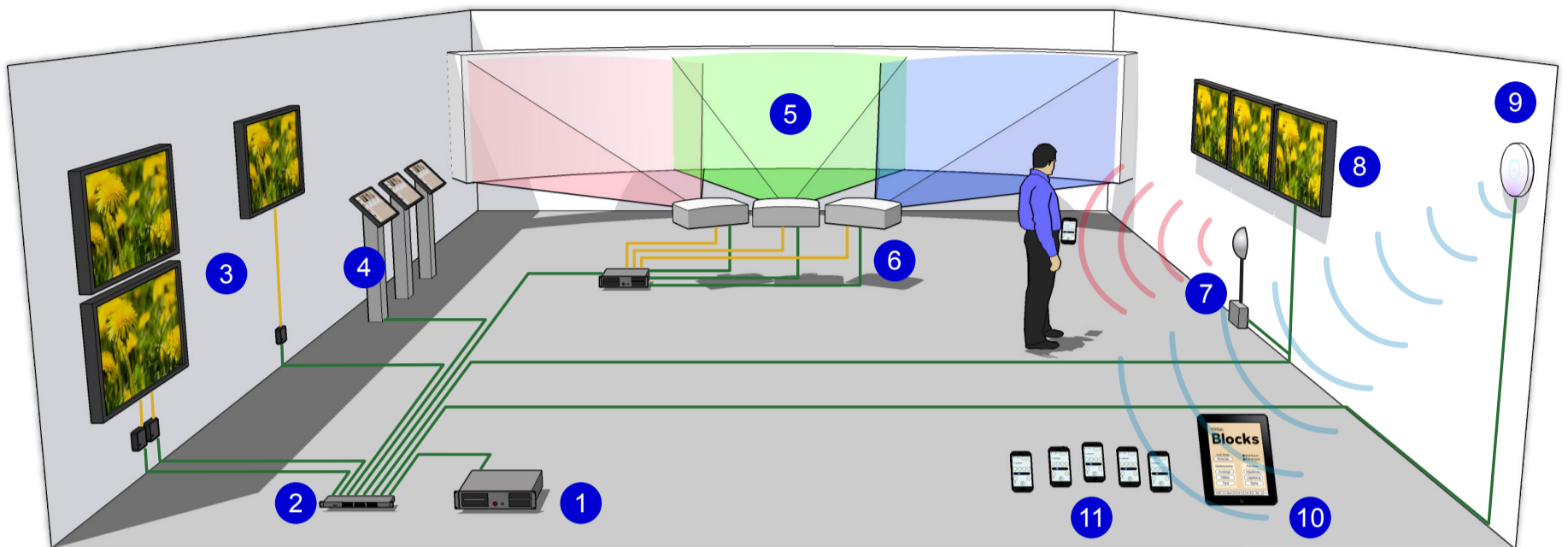
© Copyright, PIXILAB Technologies AB, все права защищены.

Торговые марки: PIXILAB Blocks, PIXILAB Moments (PIXILAB Technologies AB), Google Chrome (Google, Inc), Windows (Microsoft, Inc), macOS (Apple, Inc), Dataton WATCHOUT (Dataton AB).

Иллюстрации: Матрёшка; [James Jordan](#). Интерактивные блоки; [Filmbyn](#). Коллекция фото автомобилей, [Museum of Applied Arts & Sciences](#), Сидней.

1. Знакомство с Blocks

PIXILAB Blocks® – программное обеспечение для создания мультимедийных сред музеев, выставок, общественных пространств и корпоративных мероприятий. ПО BLOCKS – это сочетание управления контентом и дисплеями, функций системы управления, интерактивности и возможностей мобильного гида.



Система на базе Blocks и схема подключений. Сеть (зеленый), видео (желтый), Wi-Fi (синий) ИК / метки (красный).

На приведенной выше иллюстрации показана типичная система Blocks, содержащая следующие важные элементы (они помечены цифрами в синих кружочках).

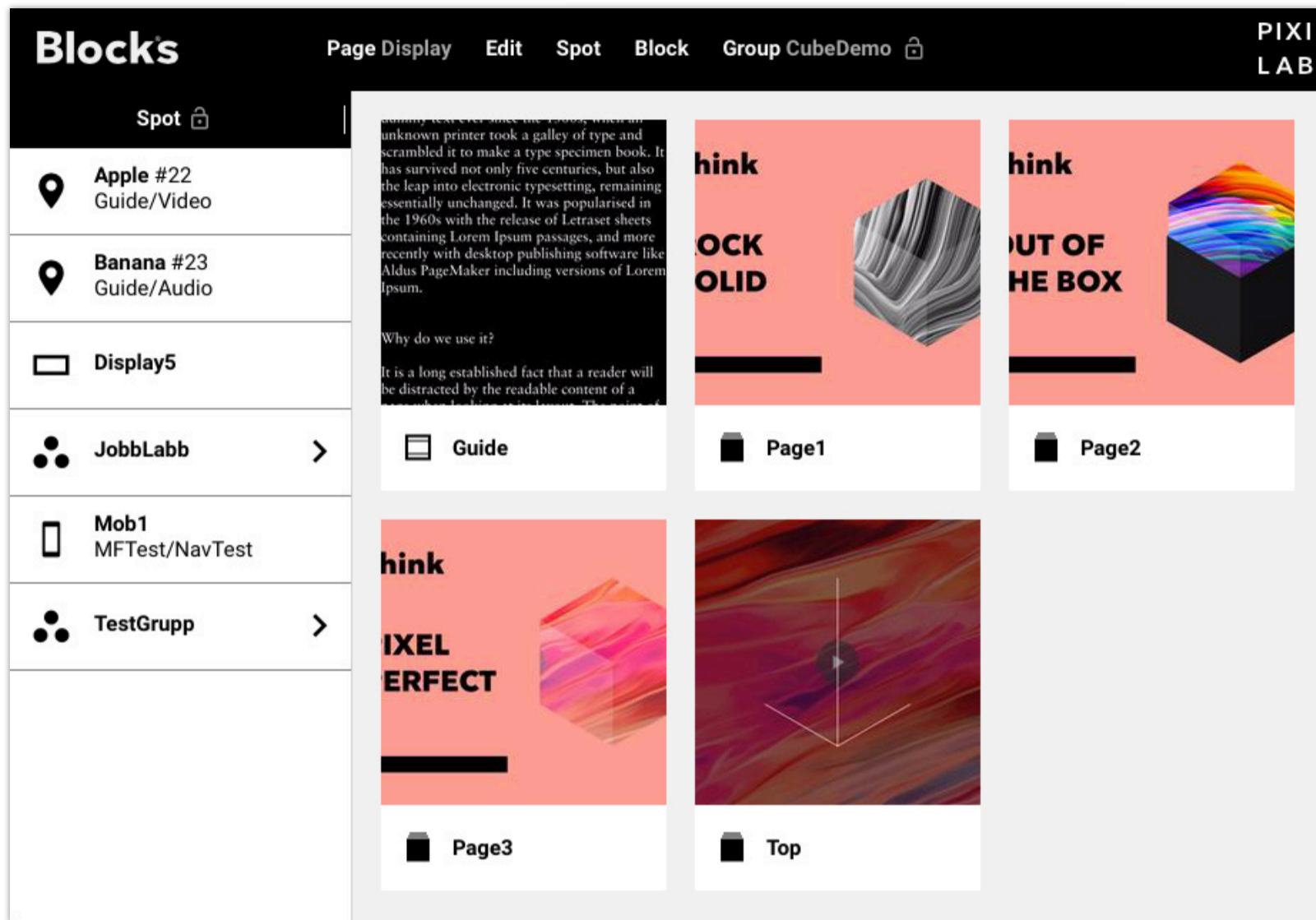
1. Сервер PIXILAB Blocks.
2. Проводная сеть.
3. Дисплеи с отдельными плеерами.
4. Интерактивные сенсорные киоски.
5. [Dataton WATCHOUT®](#) – в качестве подсистемы для проекции на вогнутый экран.
6. Проекторы и другие устройства, контролируемые через сеть.
7. Реле, переключатели и датчики (здесь - ИК-датчик движения), подключаются через стандартные интерфейсы.
8. Интеллектуальные дисплеи со встроенными плеерами подключают непосредственно к сети.
9. Сеть Wi-Fi для внутреннего управления и/или общего доступа.
10. Панели управления для персонала.
11. Смартфоны посетителей или арендуемые смартфоны для использования в качестве мобильных гидов.

Сервер

В основе BLOCKS лежит серверное программное обеспечение. Эта программа может работать на вашем персональном компьютере или на мощном специализированном серверном оборудовании. Персональный Mac или Windows PC удобно использовать на стадии продакшна и для проведения быстрых тестов, а также для запуска небольших проектов. Для более масштабных и сложных инсталляций необходимо специальное серверное оборудование, работающее под ОС Linux.

Редактор

Вы создаёте и редактируете презентации с помощью обычного веб-браузера, который подключает к серверу Blocks по сети. Здесь контент (например, изображения, видео и звук) объединяется в блоки. Затем контент назначается дисплеям или мобильным устройствам, которые мы будем называть *спотами* (*spots*).



На странице Display редактора видны дисплеи и другие споты, а также блоки контента.

Споты (Spots)

Точки интереса, такие как дисплеи, кинотеатры, локации на местности или внутри зданий, обычно называются спотами. Споты указаны в левой части страницы Дисплей (см. выше) и могут быть собраны в группы и подгруппы для упрощения навигации и программирования.

Спотами будем считать и мобильные устройства, которые могут быть постоянной частью системы (например, iPad в качестве панелей управления) или временной (например, смартфоны посетителей).

Блоки (Blocks)

Подобно кирпичикам LEGO, блоки можно комбинировать практически как угодно. Это позволяет готовить контент для любых случаев - от простого слайд-шоу в стиле цифровых вывесок до интерактивного киоска или системы мобильного гида, где видео и звук воспроизводятся синхронно на нескольких устройствах.

Например, используйте блок композиция (Composition), компануя изображения и кнопки для создания пользовательского интерфейса. Если ваш проект сложный, используйте блок Книга (Book) для размещения нескольких блоков композиций (Composition) на нескольких страницах. Чтобы рассказать историю, используйте блок слайд-шоу (Slide Show), собрав ряд изображений или коротких видеоклипов. Для привлечения посетителей, например, к интерактивному киоску можно использовать блок Attractor.

2. НАЧАЛО РАБОТЫ

В этой главе описывается, как создавать и запускать Blocks для создания, тестирования, а также для запуска готовой инсталляции. Вы можете использовать ноутбук Mac или Windows на стадии продакшн и базовых тестов всех функций Blocks, но для запуска готовой инсталляции мы рекомендуем использовать выделенный сервер на базе Linux.

Mac

Использование ноутбука или настольного компьютера удобно для работы со сторонним программным обеспечением для создания контента (изображений, видео и звука). Хотя вы можете получить доступ к серверу Blocks с любого компьютера через локальную сеть, наличие приложения Blocks на вашем ноутбуке означает, что вы можете работать и над созданием контента, и тестировать споты (spots).

Инсталляция

- Чтобы установить приложение Blocks на вашем компьютере Mac, выполните следующие действия:
- Загрузите Мастер-установщик PIXILAB_Blocks.dmg из <https://pixilab.se/docs/blocks/macos>.
- Откройте файл DMG, прочтите и подтвердите согласие с лицензионным соглашением.
- После открытия образа диска вы увидите окно, подобное показанному на рисунке ниже.
- Если вы впервые устанавливаете Blocks на данном компьютере, запустите установщик CmInstall.pkg installer. В результате будет установлен драйвер лицензионного ключа. После установки драйвера необходимо перезагрузить Mac. Не нужно повторять этот шаг при обновлении Blocks до более свежей версии.
- Перетащите значок приложения PIXILAB Blocks в папку «Приложения» (Applications) или в другое подходящее место.
- Подключите лицензионный ключ PIXILAB к порту USB на вашем компьютере.

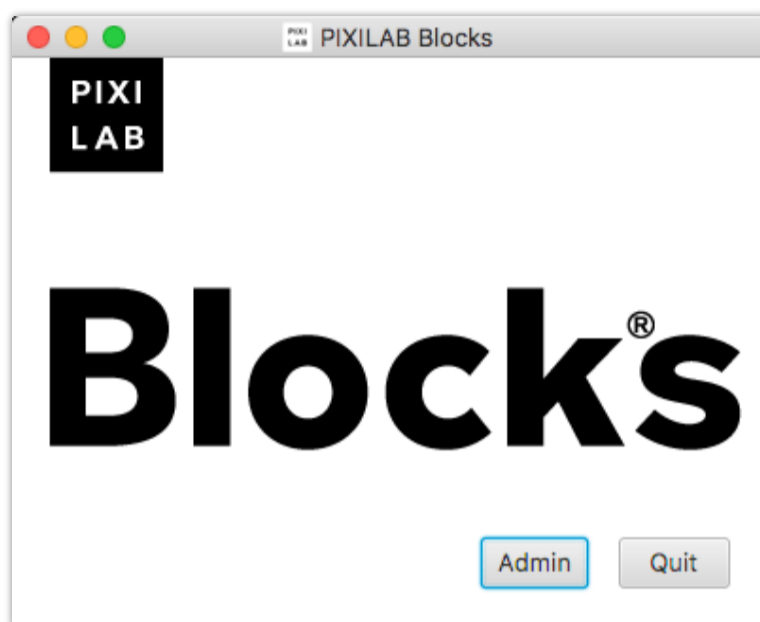


Окно Мастера установки для Mac после открытия файла PIXILAB_Blocks.dmg.

Запуск сервера Blocks

Найдите приложение «PIXILAB Blocks» в папке «Приложения» (Applications) или там, где вы его оставили. Чтобы упростить использование в будущем, можно перетащить значок приложения в Dock. Затем запустите приложение Blocks. Откроется окно, показанное на рисунке ниже. Поскольку всё администрирование сервера выполняется по сети, в этом окне мало информации. Откройте редактор Blocks (Blocks Editor), нажав кнопку «Администратор» (Admin)..

- ◆ **ПРИМЕЧАНИЕ.** Если вы получаете системное сообщение, а не окна, представленные ниже, следуйте инструкциям страницы загрузки (см. выше).



Окно сервера PIXILAB Blocks.

Откройте редактор блоков с помощью кнопки «Администратор» в окне сервера, затем действуйте, как сказано выше в разделе [«Доступ к редактору»](#) далее в этой главе.

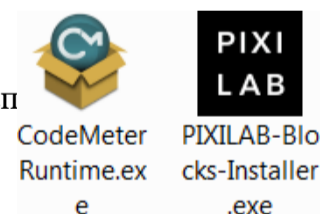
Windows PC

Использование ноутбука или настольного компьютера упрощает работу с другим программным обеспечением для создания контента для изображений, видео и звука. Хотя вы можете получить доступ к серверу с любого компьютера через локальную сеть, наличие ПО Blocks, установленного на вашем ноутбуке, означает, что вы можете работать локально над производством контента, а также с тестовыми спотами.

Установка

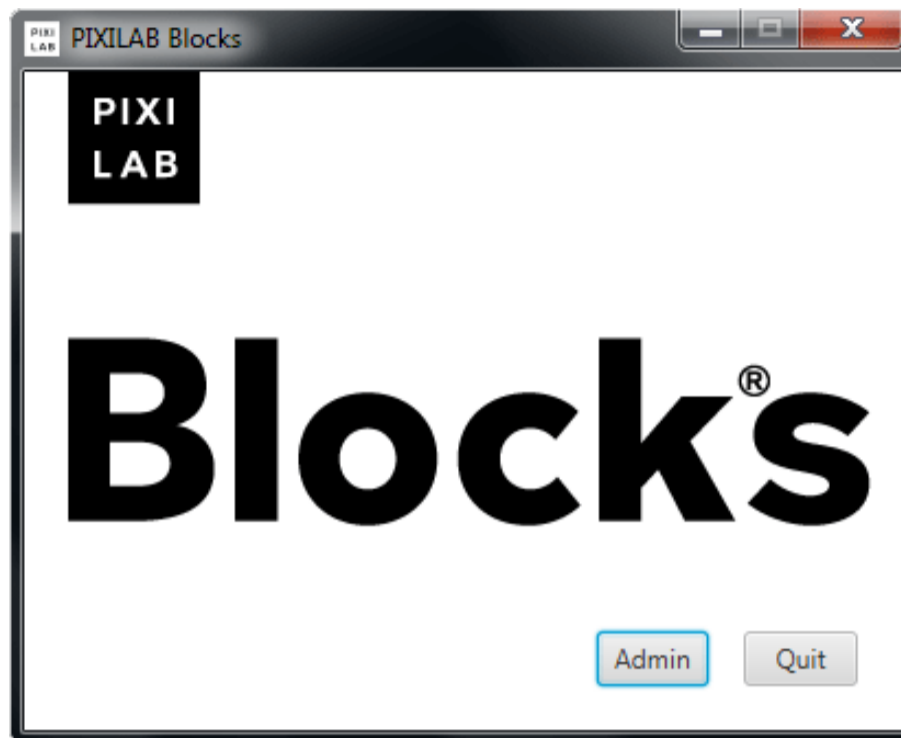
Чтобы установить Blocks на ПК с Windows, сделайте следующее:

- Загрузите установщик PIXILAB_Blocks.zip с <https://pixilab.se/docs/blocks>.
- Щелкните правой кнопкой мыши файл PIXILAB_Blocks.zip и выберите «Извлечь все», чтобы распаковать архив.
- Откроется окно с двумя значками, показанными справа.
- Если вы впервые устанавливаете Blocks на этот компьютер, запустите программу установки «CodeMeter Runtime.exe» для драйвера лицензионного ключа. После этого может потребоваться перезагрузить компьютер. Не нужно повторять этот шаг при обновлении Blocks до более новой версии.
- Дважды щелкните файл «PIXILAB-Blocks-Installer.exe» и следуйте инструкциям по установке Blocks.
- Подключите любой лицензионный ключ к USB-порту вашего компьютера.



Запуск сервера Blocks

Нажмите «PIXILAB Blocks Server» в меню «Пуск». Откроется окно, показанное ниже. Поскольку все администрирование сервера осуществляется по сети, в этом окне мало что можно увидеть. Вы можете открыть редактор Blocks, нажав кнопку «Администратор».



Окно сервера PIXILAB Blocks.

Доступ к редактору

Откройте Редактор Blocks, нажав кнопку «Администратор» (Admin) в окне сервера или перейдя по адресу сервера, за которым следует `/edit`. Например, если ваш сервер Blocks имеет IP адрес 10.2.0.10, вы должны иметь доступ к редактору, введя следующее в веб-браузере на компьютере, подключенном к той же сети, что и ваш сервер Blocks:

```
http://10.2.0.10:8080/edit
```

Появится запрос на имя пользователя и пароль. При первом запуске сервера есть только один пользователь с именем «admin» и с паролем «pixi». Вы должны как можно скорее изменить этот пароль, если сервер общедоступен (это делается на странице «Управление» ([Manage](#)), как описано в разделе «Пользователи» "[Users](#)").

После входа в систему, вы увидите окно, похожее на окно, показанное выше в разделе "[Editor](#)".

- ◆ Номер IP-порта, используемый сервером (8080 в приведенном выше примере), может различаться в зависимости от конфигурации вашего сервера. Например, образ сервера Linux настроен на использование порта по умолчанию 80, и в этом случае номер порта указывать не нужно.

Добавление дисплея

Для проверки контента, который вы создаёте в приложении Blocks, вам нужно отображать его на дисплее. Это может быть просто компьютер с современным веб-браузером, например Google Chrome или Apple Safari. Желательно, чтобы браузер работал в полноэкранном режиме, а адресная строка и другие элементы управления были скрыты. Пусть ваш Мас-сервер имеет IP-адрес 10.0.2.10, как в приведенном выше примере, назначьте ваш дисплей-компьютер на:

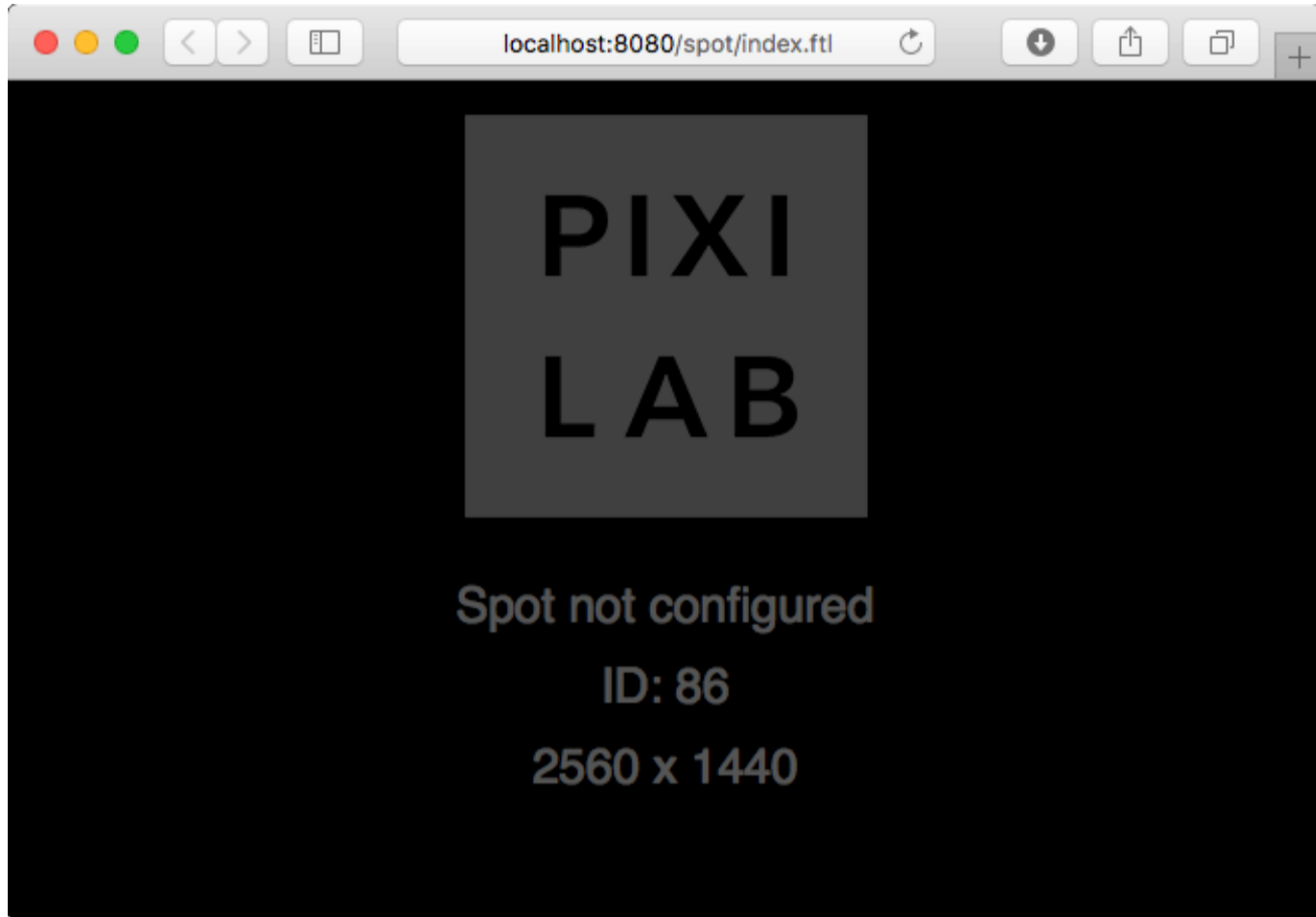
```
http://10.2.0.10:8080/spot
```

Если у вас нет отдельного компьютера, вы можете просто открыть ещё одно окно браузера на своем ноутбуке, указав его по следующему адресу:

`http://localhost:8080/spot`

Имя *localhost* всегда ссылается на IP-адрес локального компьютера, поэтому его можно использовать, как сокращенное имя вместо IP-адреса компьютера.

Вы увидите чёрное окно, подобное показанному ниже, с указанием ID спота и его полноэкранных размеров.



Окно Display перед добавлением на сервер.

Оставив окно браузера на экране, вернитесь к редактору Blocks. Выберите "Add Display" в меню Spot. Выберите номер ID отображенный на дисплее как "Spot ID" и введите имя в поле Name.

A screenshot of a configuration dialog box titled 'Display'. It has a blue header bar with the title. Below the header, there are several input fields: a dropdown menu for 'Spot ID Displayed On Screen' with the value '77', a text field for 'Name' with the value 'My first spot', a text field for 'Fixed Tags', and a text field for 'Location ID#'. At the bottom, there are two columns for 'Width' (1920) and 'Height' (1080). At the very bottom right, there are 'CANCEL' and 'OK' buttons.

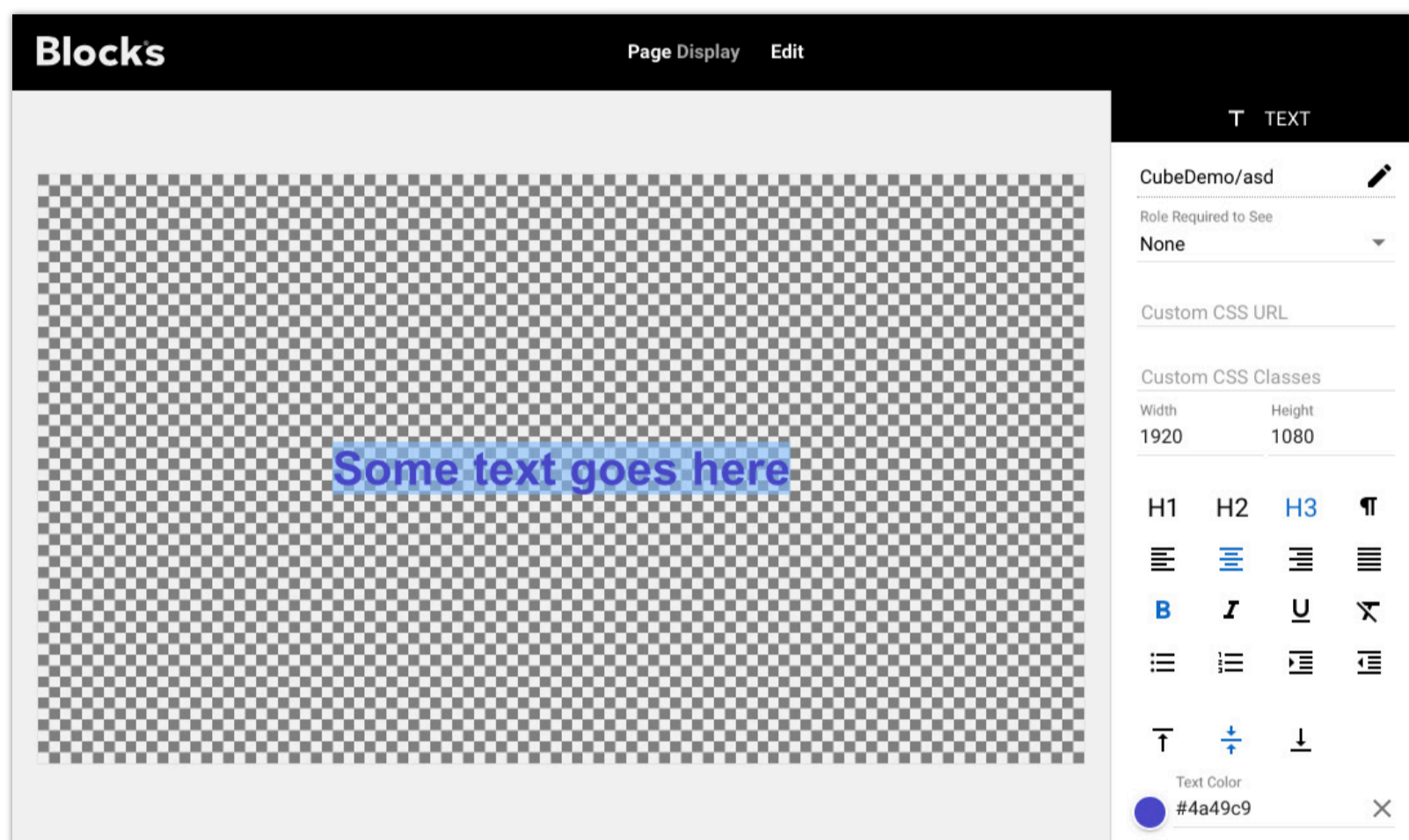
Добавление Display Spot через назначение Spot ID.

При нажатии ОК в диалоговом окне появится имя спота вместо “Spot not configured” (spot не сконфигурирован), это значит, что дисплей успешно принят в систему.

Отображения блока

Чтобы что-то появилось на дисплее, достаточно просто перетащить блок в спот на странице [Display page](#). Если в вашей системе ещё нет блоков, вы можете создать простой текстовый блок следующим образом:

- Выберите “Add Text” (добавить текст) в меню Block, укажите имя блока и нажмите «ОК».
- Дважды щёлкните на текст, чтобы отредактировать его, центрируйте его по горизонтали и вертикали, используя соответствующие кнопки, расположенные справа.
- Измените цвет текста.
- Вернитесь на главную страницу (Display page), щёлкнув на логотип “Blocks” в верхнем левом углу.
- Перетащите созданный текстовый блок на спот, чтобы он отобразился на дисплее.



Создание текстового блока.

Linux

Для постоянных инсталляций рекомендуется использовать сервер на Linux. В отличие от версий для Mac и Windows, которые могут быть установлены на вашем личном компьютере, версия приложения Blocks для Linux предназначена для специального использования. Это гарантирует наилучшую долговременную стабильность, особенно в сочетании с проверенным оборудованием. Версия Blocks для Linux предоставляется в следующих двух видах:

- Предварительно установленная на серверное оборудование, предоставляемое PIXILAB или одним из наших авторизованных посредников.
- В качестве предварительно настроенного “образа Linux” (Linux image) для установки на подходящее оборудование.

Первый вариант предоставляет вам укомплектованную под ключ комбинацию оборудования/программного обеспечения, готовую к использованию оптимизированную систему. Она поставляется в 19-дюймовом корпусе для монтажа в стойку. Базовая конфигурация имеет два сетевых порта 1Gb, но также может быть снабжена двойной картой 10Gb. Свяжитесь с PIXILAB для получения подробной информации о технических характеристиках и доступных опциях.

Опция образ Linux позволяет загрузить и установить предварительно настроенный сервер на собственное оборудование. Он протестирован и гарантированно работает на компьютерах Intel NUC малого форм-фактора с любым диском M.2.

- ◆ Хотя “образ Linux”, скорее всего, будет работать и на другом оборудовании, его пригодность, надежность и производительность на таком оборудовании отнюдь не гарантирована.

На [PIXILAB Wiki](#) можно найти инструкции, как получить и установить образ Linux.

В дополнение к настроенному ядру сервера Linux, аппаратное обеспечение/образ сервера Linux имеет следующие предустановленные и настроенные функции:

- Программное обеспечение PIXILAB Blocks Server и связанные с ним дополнительные компоненты и конфигурацию.
- Поддержка резервирования серверов на базе службы зеркал [Blocks mirroring service](#) (нужна дополнительная лицензия).
- Программа SSH для безопасного удаленного доступа.
- Поддержка сетевой загрузки [network booting](#) для [PIXILAB Player](#) (PXE).
- Серверы DNS и DHCP.
- Программное обеспечение для удаленного управления.
- Средства разработки и библиотеки драйверов и расширенных сценариев
- Программное обеспечение для устранения неполадок и диагностики сети
- Программное обеспечение для тестирования Artnet/ DMX-512.
- Сетевое взаимодействие Windows/Mac (SMB).
- Совместимость с USB-накопителем Windows / Mac (с использованием файловой системы ExFAT).

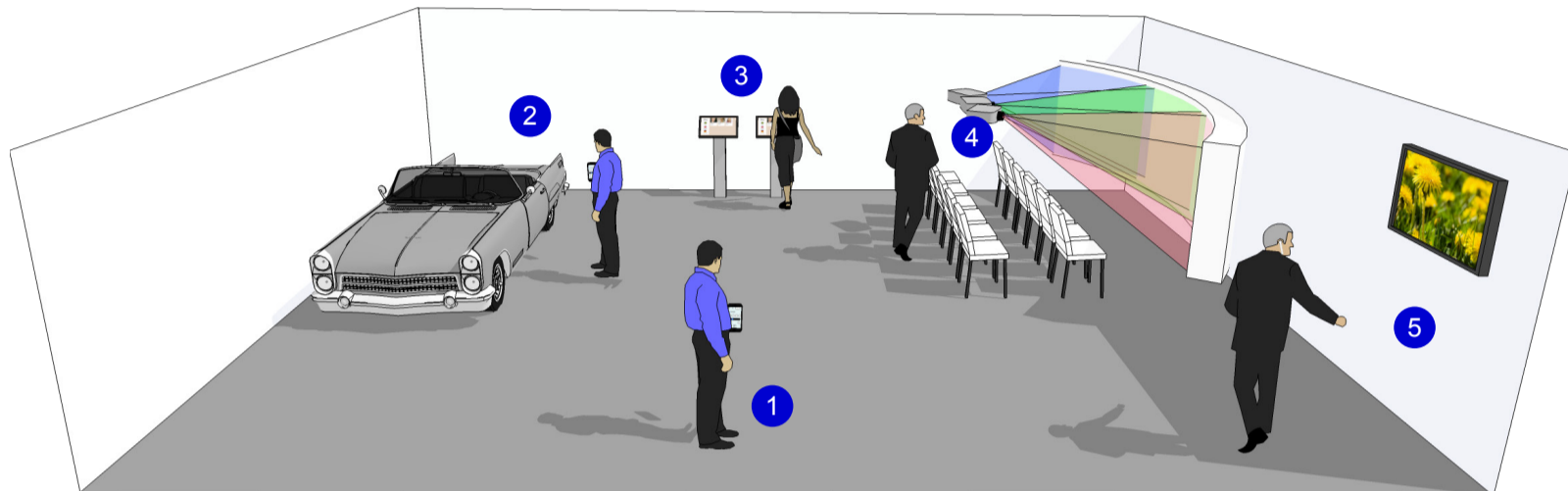
Все перечисленные выше функции предварительно установлены и настроены, но большинство вспомогательных функций отключены при установке по умолчанию из соображений безопасности. Инструкции по их включению, а также информацию о связанных с этим последствиях для безопасности можно найти в разделе «Сервер на базе Linux» по адресу <https://pixilab.se/docs/blocks>.

3. ОСНОВНЫЕ ПОНЯТИЯ

В этой главе дается описание понятий, используемых в Blocks, таких как споты, блоки и задачи (tasks). Эти термины подробно рассматриваются в следующих главах.

Спот (Spot)

В самом общем смысле spot - это место. Это может быть комната, объект на местности или внутри здания. В контексте Blocks спот, как правило, связан с некоторыми медиа, такими как звук или видео, привязанными к этому споту. Хотя спот может быть просто некоторой точкой в пространстве, он также часто привязан к какому-то оборудованию: дисплею, сенсорному киоску или проекционной системе с широкоэкранным шоу WATCHOUT.



Споты внутри помещения.

Ссылаясь на приведенный выше рисунок, перечислим типы спотов, принятых в Blocks:

1. Гостевой спот [Visitor spot](#): персональные мобильные устройства посетителей.
2. Локация [Location](#): объект или местоположение, в данном случае – это автомобиль.
3. Сенсорный киоск на базе Дисплей-спота [Display Spot](#) с интерактивным блоком (контентом).
4. Кинотеатр с широким экраном, где [WATCHOUT Spot](#) обеспечивает синхронизированный звук через мобильные устройства посетителей.
5. Дисплей-спот [Display Spot](#), воспроизведение видео, изображений и другого контента, созданного в приложении Blocks. Кроме того, синхронизированный звук на нескольких языках может быть предоставлен через мобильные устройства посетителей.

См. главу Споты "[Spots](#)" для получения подробной информации о типах спотов.

Блоки

В приложении Blocks материалы, которые вы используете для создания контента (и это вносит путаницу), называются блоками. Подобно кирпичикам LEGO, эти блоки можно комбинировать как угодно для получения разнообразных презентаций. Споты соотносятся с физическими местоположениями или дисплеям, блоки же более абстрактны, и чтобы освоить работу с ними, потребуется немного больше времени.

Хотя некоторые блоки могут использоваться сами по себе (например, можно использовать только текстовый блок для отображения текстового сообщения), блоки обычно объединяют разными способами. Это получается путём размещения блоков внутри других блоков, что напоминает устройство русской матрёшки.

Вот некоторые из типов блоков:

- *Slideshow* – слайд-шоу для показа последовательности изображений, видео или другого контента.



- *Книга (Book)* для создания интерактивных презентаций с использованием нескольких страниц.
- *Композиция (Composition)* для размещения других блоков или интерактивных элементов на странице.
- *Скроллер (Scroller)* для интерактивной прокрутки контента по горизонтали или вертикали.
- *Аттрактор (Attractor)* для отображения цикла привлечения посетителя вплоть до момента его взаимодействия со спотом и переключением на активный контент этого спота.
- *Синхронизатор (Synchronizer)*, синхронизирует воспроизведение на разных спотах видео и звука. Например, звук идёт через смартфон синхронно с видео, отображаемым на дисплее.

Кнопки, движки и другие элементы интерактивного управления используются наряду с другими блоками. Но такие элементы управления можно использовать только внутри блока *Composition*.

В главе "[Blocks](#)" подробно описывается каждый тип блоков.

Управляемые подсистемы

Споты и блоки обеспечивают создание контента, его отображение и интерактивность, но приложение PIXILAB Blocks может управлять и многими другими вещами, которые можно добавить и настроить на странице [Manage](#) (управление).

Module	IP Address	Enabled	Notes
Jobblabb Moxa4	10.0.1.113	YES	

Channel	Direction	Type	Alias
1	Out	Digital	03-B-FixaFelet Nasan
10001	In	Digital	03-B-FixaFelet Knapp1
10002	In	Digital	03-B-FixaFelet Knapp2
10003	In	Digital	03-B-FixaFelet Knapp3
10004	In	Digital	03-B-FixaFelet Knapp4

На приведенном выше рисунке показано, что можно добавить модули ввода/вывода, поддерживающие протокол Modbus для управления и взаимодействия с контактными разъемами и датчиками. Modbus – это стандартный протокол, поддерживаемый большинством поставщиков. Таким же образом вы можете управлять освещением (через Art-net / DMX512), сетевыми устройствами, такими как проекторы (включение / выключение питания, переключение входов), внешние источники данных (например, считыватель штрих-кода) и т. д.. Для взаимодействия с такими сетевыми устройствами доступно специализированное программное обеспечение ([device drivers](#)). Можно контролировать кластеры WATCHOUT для интеграции в систему Blocks крупномасштабных систем отображения.

Задачи (Tasks)

Многие основные функции управления можно реализовать прямым соединением кнопок с объектами управления. Такое однозначное соединение «один-к-одному» выполняется в редакторе блока Композиция (Composition) при добавлении и размещении кнопок. В качестве бонуса обеспечивается обратная связь: на кнопке отображается статус устройства. Всё это происходит без какого-либо программирования с вашей стороны.

Однако некоторые функции не так просты, как однозначное взаимодействие кнопки и функции устройства. Более того, прямая связь – это просто прямая связь. Нет никаких «если» (if) и «но» (but) или других вариантов. Иногда вы хотите, чтобы всё происходило только при определенных условиях; например, не в выходные дни, или когда в комнате не более 10 человек. Вы можете захотеть, чтобы некие события происходили в зависимости от какого-то состояния.

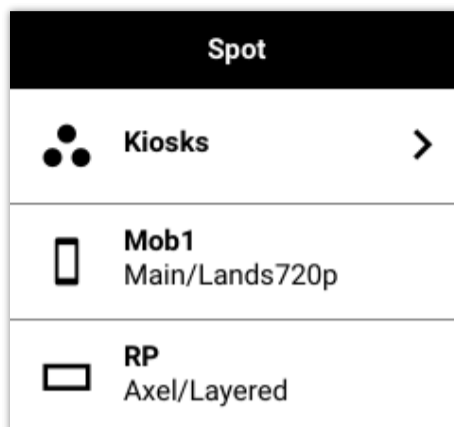
Все это и многое другое можно предусмотреть на странице [Task](#) (Задачи). Эта страница в приложении Blocks предоставляет базовую, но удивительно мощную среду программирования. Так же, как вы совмещаете блоки контента на странице Display, на странице «Задача» (Task) можно подключить команды программирования простым перетаскиванием. Выберите в раскрывающемся меню свойства, которыми нужно управлять, так же, как вы это делали при назначении кнопок. Затем используйте простой язык выражения, чтобы детально определить, что и как нужно сделать. Язык выражений основан на *JavaScript* и *lingua franca* в Интернете.

The screenshot shows the 'Blocks' application interface for configuring a 'TASK'. At the top, there are navigation options: 'TASK', 'Edit', and 'Realm Main'. Below this is a 'Trigger Condition' field containing the code: `trigger.getDay() > 0 && trigger.getDay() < 6`. The main workspace is divided into three sections:

- Left Sidebar:** Contains a 'Group' section with a '+' icon, a 'Task' section with a '+' icon, and a 'Realm Variable' section with a '+' icon. The 'Task' section is currently selected, showing a 'StartShow' task.
- Central Workspace:** A list of tasks numbered 1 through 7. Each task has a 'do' field and a 'set to' field. The tasks are:
 - 1. do Network.Barco1.power, set to ProjectorPower.value
 - 2. do Network.Barco2.power, set to ProjectorPower.value
 - 3. do Network.Barco3.power, set to ProjectorPower.value
 - 4. do WATCHOUT.WO2.wakeUp, timeout Literal value or expression
 - 5. do WATCHOUT.WO2.load, showName "Thirid Generation Intro"
 - 6. do WATCHOUT.WO2.play, timeline Literal value or expression
 - 7. do IO.DALI_5.value, set to !ProjectorPower.value
- Right Sidebar:** Contains the 'TASK' settings. It includes a 'StartShow' field with an edit icon, an 'Enabled' checkbox (checked), a 'Trigger' dropdown set to 'Time of Day', a 'Time' field set to '9:00', 'Starting at Date' and 'Ending after Date' checkboxes (unchecked), a 'Repeat' dropdown set to 'At Interval', an 'Interval' field set to '1:00', and an 'Ending at Time' checkbox (checked) with an 'End Time' field set to '18:00'.

4. СПОТЫ

В этой главе представлены подробные описания разных типов спотов. Вводная информация – в разделе “[Spots](#)” в главе «Основные понятия». Вы найдете все свои споты на странице [Display](#) в списке слева. Споты могут быть внутри группы спотов [Spot Group](#), и чтобы войти в группу, нужно щёлкнуть на значок стрелки.



В крайнем левом столбце списка спотов (Spot list) содержится перечень свойств спотов. Перетащите правый край столбца спотов (Spot), чтобы открыть более подробную информацию. При этом станут видны ещё две колонки:

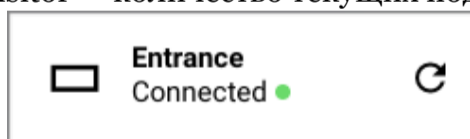
1. **Назначенный блок (Assigned Block)** указывает тип и имя блока, который в данный момент назначен этому споту.
2. **Воспроизводимый блок (Playing Block)** указывает текущий воспроизводимый блок (выделен серым цветом, если спот находится в автономном режиме). Это часто совпадает с назначенным блоком, но может и отличаться, если назначенный блок является Расписанием (Schedule) или если назначенный блок был переопределён блоком приоритета, что можно сделать с помощью кнопки или задачи.

Если в символе назначенного или воспроизводимого блока отображается вопросительный знак, это означает, что блок не найден, и поэтому не может быть воспроизведён. Это происходит, если вы установили несуществующий блок с помощью кнопки или задачи или если вы удалили/переименовали блок после того, как назначили его споту.

Чтобы заставить спот (Spot) воспроизводить определённый блок, просто перетащите блок на спот на [странице Дисплей](#) (см. “[Displaying a Block](#)”). Если вы перетащите блок в группу [Spot Group](#), блок будет воспроизводиться на всех спотах этой группы, если только он не будет переопределен для какого-то из них, если не переопределено для вложенного спота.

Статус спота

Статус подключения спотов указан в списке. Если спот находится в сети, отображается символ перезагрузки. Кроме того, при расширении столбца «Спот» для просмотра дополнительных сведений отображается индикатор «Подключено» или — для гостевого спота Visitor — количество текущих подключений.



После внесения изменений в блок щелкните символ перезагрузки, чтобы отобразить измененный блок (или включите «Автообновление» в настройках спота). Щелчок символа перезагрузки при нажатой клавише Shift также перезагружает программное обеспечение дисплей-спота.

При использовании [PIXILAB Player](#) с включенной функцией кэширования блоков символ перезагрузки будет вращаться, а любой индикатор подключения будет оранжевым до тех пор, пока по умолчанию кэшируется блок. Не отключайте плеер во время кэширования, если вы хотите использовать плеер в автономном режиме.

Настройки спота

Дважды щелкните Spot, чтобы изменить его настройки. Настройки зависят от типа спота, и как правило содержат имя, теги, ID местоположения и назначенную роль.

Имя спота

Используется для распознавания спота в пользовательском интерфейсе, а также для управления им при помощи кнопок и задач.

- ◆ Избегайте изменения имени спота, если вы запрограммировали какие-либо действия (кнопками или задачами), связанные с этим спотом, так как это нарушает соединение.

Тэги (Tags)

Вы можете назначить любой набор тэгов споту. Тэг – это произвольное слово или комбинация слов. Тэги разделены запятой. Тэги могут быть использованы вместе с блоком выбора тэгов [Tag Selector](#) для выбора контента для этого спота. Дополнительные тэги могут быть применены к споту через кнопку [Button](#) (используется в основном для гостевых спотов Visitor).

ID местоположения (Location ID)

Споту присваивается короткий числовой идентификатор (ID). Поиск спота происходит с помощью блока [Locator](#) введением ID спота на клавиатуре. Это можно использовать для создания мобильного гида, когда посетители воспроизводят релевантный контент, вводя номер экспоната, показанный на этикетке. Вам нужно только назначить ID спотам, которые вы хотите использовать таким образом. Как правило, достаточно двух- или трехзначного числа.

Назначение роли

Вы можете ограничить доступ к споту или управлению им, назначив роли [role](#):

- **Роль, необходимая для использования спота.** Обозначает круг лиц, допущенных к использованию спота и к работе со спотом. Это особенно полезно для панелей оператора и т. д., предоставляя функции управления системой, которые не должны быть доступны для гостей. Если установлено любое значение, кроме «Нет», перед предоставлением доступа к споту потребуется войти в систему.
- **Роль, необходимая для управления.** Ограничивает круг лиц, допущенных к управлению свойствами спота, такими как назначенный ему блок или состояние включения/выключения питания.

Display (дисплей-спот)

Представляет один дисплей, обычно в фиксированном месте. В качестве Дисплей-спота можно использовать:

- Обычный веб-браузер, работающий на ноутбуке.
- Небольшой ПК, такой как Intel NUC, с нашим бесплатным программным обеспечением [PIXILAB Player](#).
- «Умный дисплей» со встроенным проигрывателем от таких производителей, как [Samsung](#), LG, [Philips](#), Panasonic или EIZO.
- Другие плееры для цифровых вывесок с полной поддержкой HTML5, например, некоторые плееры Brightsign.

Такие дисплеи/проигрыватели обычно подключаются к серверу Blocks с помощью проводной сети. Помимо имени, тегов и ID местоположения (см. выше), он включает следующие настройки. Сервер Blocks лицензируется на максимальное количество дисплей-спотов и гостевых спотов. Если вы попытаетесь добавить сверхнормативные дисплей-споты, появится сообщение об этом, и тогда следует связаться с PIXILAB для расширения вашей лицензии.

- ◆ **ВАЖНО:** производительность и возможности сильно различаются у разных плееров. Для наилучшей производительности и функциональности используйте наше ПО [PIXILAB Player](#), работающее на достаточно мощном процессоре Intel NUC или аналогичном. При использовании смарт-дисплеев или плееров обязательно проверьте все функциональные возможности с использованием репрезентативного контента, прежде чем выбирать такие устройства для использования с Blocks. Некоторые функции Blocks, требующие современных возможностей браузера, могут не работать на некоторых смарт-дисплеях.

General (основные)

Основные настройки, общие с другими типами спотов, описанными [выше](#).

Display MainDisplay

GENERAL

AUDIO/VIDEO

GEOLOCATION

ADVANCED

INFORMATION

PARAMETERS

Name
MainDisplay

Fixed Tags

Location ID#
85

Discard Display ID

Re-assign to Spot ID

CANCEL OK

Discard Display ID (отменить ID дисплея)

Этот флажок появляется только для спота, который уже назначен для дисплея. Это полезно, если вы хотите отсоединить плеер, чтобы переназначить его на другой дисплей-спот.

Re-assign to Spot ID (переназначить на Spot ID)

Эта опция появляется только при подключении новых неназначенных дисплеев. Это позволяет переназначить этот дисплей-спот на один из неназначенных. Это полезно, когда вы хотите заменить физический плеер другим плеером, сохранив при этом все настройки. В зависимости от типа и модели нового плеера некоторые настройки могут быть неактуальны. Например, некоторые настройки в разделе «Видео/Аудио» могут не применяться, если новый плеер не имеет той же аппаратной конфигурации, что и старый.

Video/Audio

Поля под этим заголовком различаются в зависимости от типа используемого плеера. Основные настройки – это ширина и высота дисплея, заданные при добавлении в Blocks. При использовании [PIXILAB Player](#) версии 3 или более поздней доступны следующие параметры.

Name	Resolution	Refresh rate	Orientation	Top	Left	Touch
DP1	1920x1080	60	Normal	1080	0	0 - None
HDMI1	1920x1080	60	Normal	1080	1920	15 - eGalax Inc. eGalaxTo...

Audio
Interface: HDA Intel PCH
Profile: hdmi-stereo

Name, Resolution, Refresh Rate, Display Orientation (имя, разрешение, частота обновления, ориентация экрана)

Управляет разрешением, частотой обновления и ориентацией каждого дисплея, подключенного к этому дисплей-споту.

Обратите внимание, что поле «Имя» может не соответствовать используемому физическому соединению. Например, на приведенном выше рисунке выходы названы DP1 и HDMI1, хотя они оба являются разъемами HDMI.

- ◆ К одному проигрывателю можно подключить несколько дисплеев, при этом проигрыватель по-прежнему представляет собой один дисплей-спот. Любой блок, назначенный этому дисплей-споту, будет охватывать объединенное пространство пикселей всех подключённых дисплеев.

Top, Left (сверху, слева)

Эти столбцы отображаются только для проигрывателей с несколькими выходами, и в этом случае относительное положение каждого дисплея и другие параметры можно указать независимо. Это позволяет расположить дисплеи горизонтально, вертикально или в виде сетки. Тщательно позиционируя дисплеи, вы можете компенсировать ширину рамок дисплея или указать перекрывающиеся области изображения для проекторов со встроенной функцией краевого смешивания.

Touch (касание)

Если на дисплее есть сенсорная накладка, она обычно подключается к плееру по USB. Обязательно выберите правильное сенсорное устройство для каждого такого дисплея. В меню «Сенсорный» могут быть перечислены другие устройства ввода, не относящиеся к сенсорным. Меню может отображать один и тот же сенсорный ввод несколько раз с немного разными именами, поэтому потребуются некоторые эксперименты, чтобы сделать выбор правильно.

- ◆ **ПРИМЕЧАНИЕ.** Настройка по умолчанию «0-None» может обеспечивать или не обеспечивать надежный сенсорный ввод в зависимости от поведения операционной системы по умолчанию. Избегайте этого параметра при использовании сенсорного наложения, даже если на первый взгляд может показаться, что оно работает.

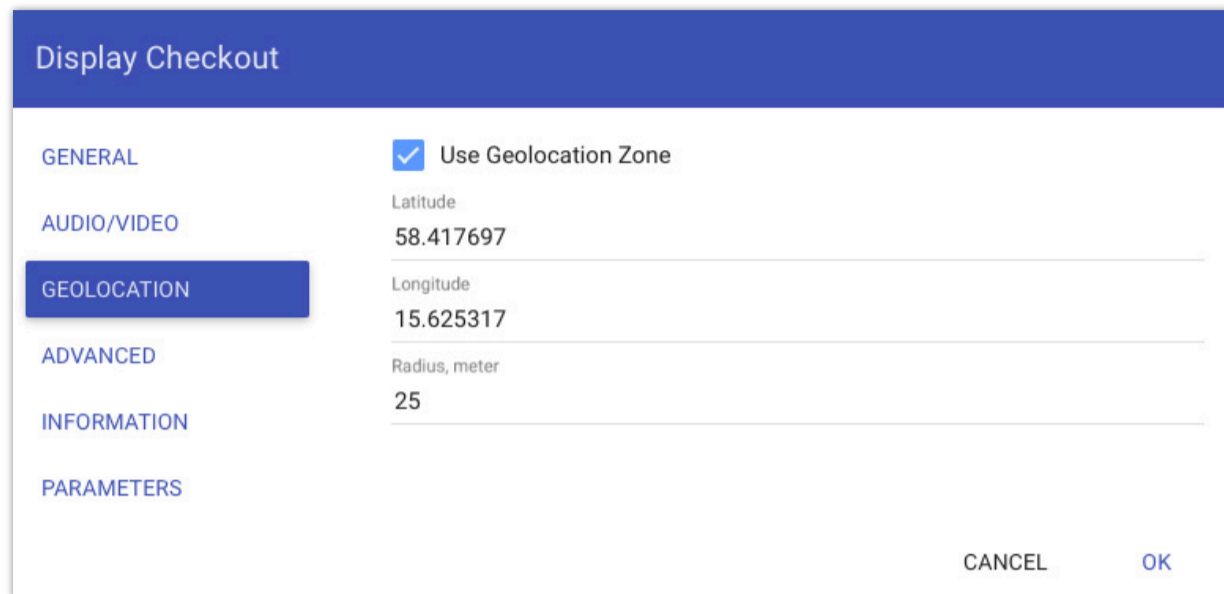
Настройки звука

Выбирает устройство вывода звука и конфигурацию. Это полезно, если доступно несколько вариантов вывода звука, например, аналоговый и HDMI, чтобы обеспечить использование нужного вывода. Убедитесь, что здесь выбран правильный аудиовыход. Он также может отображать одни и те же выходные данные несколько раз с немного разными именами, поэтому могут потребоваться некоторые эксперименты.

- ◆ **ПРИМЕЧАНИЕ.** Настройка по умолчанию «Авто» может обеспечивать надежное воспроизведение звука в зависимости от поведения операционной системы по умолчанию. Избегайте использования этого параметра, даже если кажется, что он работает нормально.

Geolocation (геолокация)

Присваивает споту географическое положение, которое впоследствии можно использовать в сочетании с функцией GPS [Локатора \(Locator\)](#). Это похоже на упомянутый выше ID местоположения в том, что он позволяет отслеживать местоположение, но использует GPS-позицию, а не ввод числа для указания местоположения.



Category	Value
GENERAL	<input checked="" type="checkbox"/> Use Geolocation Zone
AUDIO/VIDEO	Latitude: 58.417697
GEOLOCATION	Longitude: 15.625317
ADVANCED	Radius, meter: 25
INFORMATION	
PARAMETERS	

Указание зоны геолокации [geolocation zone](#) по широте, долготе и радиусу.

Advanced (расширенные настройки)

Некоторые редко используемые параметры, включая требования авторизации для дисплей-спота.

Small Blocks

Как и дисплеи, большинство типов блоков имеют определённые размеры. Если блок меньше дисплея, то параметр Small Blocks будет определять, что произойдет:

- **Scale to Fit (Уместить на дисплее)** – масштабирует блок (сохраняя соотношение сторон) для заполнения дисплея.
- **Center (по центру)** – блок в середине дисплея, без масштабирования.
- **Top Left (в верхнем левом углу)** – помещает блок в угол дисплея. Эта функция удобна при использовании с процессором светодиодной видеостены или аналогичным, и в этом случае предпочтительно всегда помещать блок в известное положение, например, в верхний левый угол.

Update on Block Change (обновление при изменении блока)

Если выбрано, назначенный этому споту блок будет автоматически перезагружен при изменении в редакторе. Включите эту опцию, чтобы получать живую обратную связь при редактировании блоков. Не включайте это для общедоступных роликов, так как это приведет к частой перезагрузке ролика при редактировании его блока. Вместо этого используйте [кнопку ручной перезагрузки](#) и после редактирования блока опубликуйте изменённый блок.

Block Caching (кэширование блока)

Кэширует блок спота по умолчанию (тот, который перетаскивается на спот на странице Display) и все связанные с ним медиафайлы в плеере. Это минимизирует трафик данных при воспроизведении этого блока. Кроме того, это позволяет воспроизводить блок в автономном режиме, когда плеер не подключен к серверу. Такое автономное использование не поддерживает никаких функций, зависящих от сервера.

- ◆ **ВАЖНО:** Для этой функции требуется наше программное обеспечение PIXILAB Player. Плеер должен быть настроен на кэширование блоков и снабжён носителем (например, SD-картой) для хранения файлов. Для каждого плеера, для которого включено кэширование блоков, требуется отдельная лицензия "[Block-Caching Players](#)". Эта лицензия остается действующей, даже если этот плеер больше не подключен к системе, или вы отключите для него кэширование блоков. Чётко отмечайте такие плееры. Плееры, для которых включено кэширование блоков, отслеживаются в файле с именем «Licenses-XXX» в каталоге model корневого каталога вашего Blocks, где XXX — номер лицензии. Не переименовывайте, не удаляйте и не делитесь этим файлом между серверами или заблокируйте корневые каталоги.

Preferred Synchronization Master (предпочтительный мастер синхронизации)

Этот параметр существует только для дисплей-спотов внутри группы спотов. При выборе этого параметра дисплей-spot назначается мастером при воспроизведении блока синхронизатора на спотах этой группы (например, путем перетаскивания блока в группу спотов [Spot Group](#)). Если ничего не выбрано, Blocks автоматически выберет мастер, поэтому в большинстве случаев вам не нужно указывать это. Вы можете захотеть явно указать мастер, если в какой-то момент понадобится заимствовать дисплеи из этой группы для других целей. Затем назначьте spot, который не будет заимствован, в качестве основного, чтобы все еще синхронизированные споты работали правильно.

Power Down Automatically (автоматическое отключение питания)

Если этот параметр выбран, а дисплей-spot имеет средства для управления питанием, оно будет отключено, если для спота не назначен ни один блок. Это можно использовать вместе с расписанием "[Schedule](#)" которое будет автоматически управлять электропитанием, отключая питание всякий раз, когда блок не отображается.

Block Transition (блок Переход)

Выбирает переход, который будет применён при смене блока, воспроизводимого на споте. Это полезно, когда проигрываемый блок должен меняться из-за взаимодействия с пользователем или по расписанию. Подробную информацию о доступных переходах см. в разделе "[Transition](#)" блока «Слайд-шоу».

Numeric Keys as Inputs (цифровые клавиши для ввода)

Когда эта функция включена, цифровые клавиши в основной области клавиатуры могут использоваться как входы GPIO (Интерфейс ввода/вывода общего назначения). Эти данные затем выставляются как доступные только для чтения свойства `keyPress` спота. Это особенно полезно в сочетании с различными устройствами, имитирующими клавиатуру [keyboard-emulating devices](#).

- ◆ Если этот параметр выбран, он будет мешать обычному использованию клавиатуры для ввода текста и цифр. Эта опция является взаимоисключающей с опцией «Ввод со сканера RFID/QR», описанной далее.

RFID/QR Scanner Input (ввод со сканера RFID/QR)

Когда включено, ввод с клавиатуры будет отображаться как свойство спота `scannerInput` (ввод сканера). Это полезно в сочетании с:

- Портативным плеером со встроенным сканером QR/штрих-кода, например, это устройство [Zebra](#).
- Считывателем [RFID/NFC reader](#) подключенным к USB-порту спота (поддерживает только считыватели с эмуляцией клавиатуры).
- В блоке Локатор с установленным "[Use RFID/QR Scanner](#)" чтобы установить ID местоположения спота (в качестве альтернативы использованию цифровой клавиатуры).
 - ◆ Если этот параметр выбран, он будет мешать обычному использованию клавиатуры для ввода текста и цифр. Этот параметр является взаимоисключающим с описанным выше параметром «Numeric Keys as Inputs».

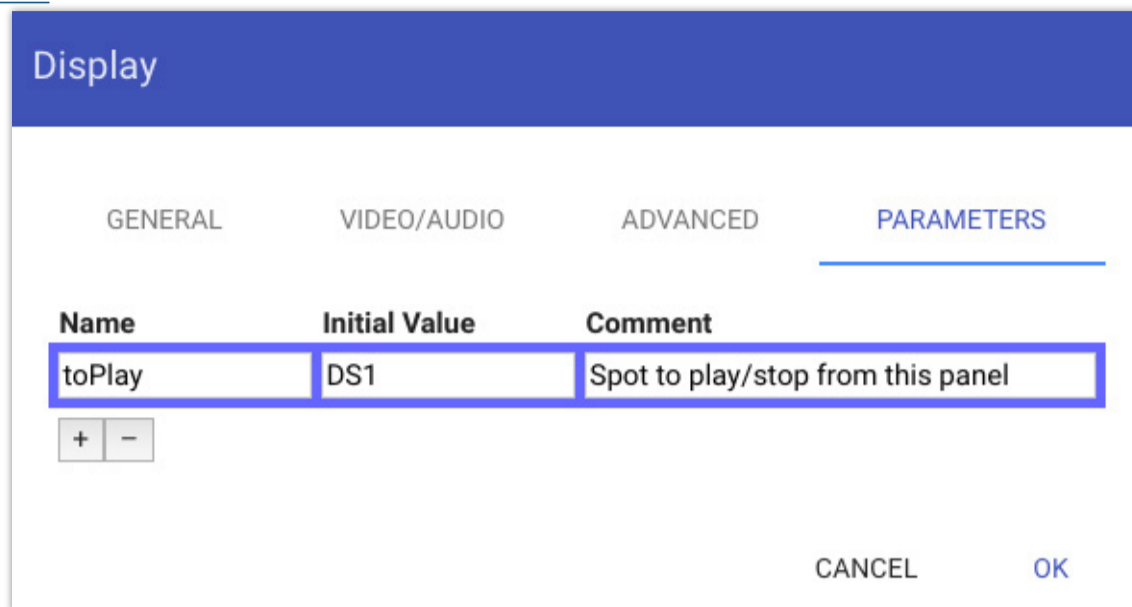
RFID/QR Input Lingers (задержка ввода RFID/QR)

Доступно только при выборе «Ввод RFID/QR-сканера». Определяет, как долго (в секундах) отсканированный код остается установленным на свойстве. Этот параметр полезен для сканеров, которые повторяют код до тех пор, пока поблизости находится метка RFID/NFC. Установите значение времени немного больше, чем интервал повторения считывателя. Это имеет два преимущества:

- Это позволяет избежать многократной установки (и очистки) свойства, что потенциально может привести к повторному запуску любого действия, вызванного тегом.
- Это также позволяет определить, когда тег *удален* по тому факту, что свойство `scanInput` становится ложным (пустая строка).

Parameters (параметры)

Присваивает начальные значения параметрам спота, специфичным для этого спота. Это позволяет переопределять начальное значение указанных параметров для каждого спота. Информацию об использовании этой функции см. в разделе "[Spot Parameters](#)".



Spot URL Query Parameters (параметры запроса URL спота)

Проигрыватель дисплей-спота или веб-браузер использует URL-адрес, как описано ранее в этом руководстве (см. "[Добавление дисплея](#)"). После обязательной части URL-адреса вы можете указать ряд необязательных параметров запроса URL-адреса. Каждый такой параметр состоит из имени и значения, разделенных знаком равенства. Список параметров URL начинается со знака вопроса. Последующие параметры разделяются амперсандом. Поддерживаются следующие параметры:

- `display=N` позволяет запускать несколько дисплей-спотов в отдельных окнах браузера, где каждое окно имеет собственный номер N.
- `mobile=SPOT` указывает, что это гостевой спот, где SPOT — это имя соответствующего гостевого спота, используемое в Blocks.
- `mac=MAC` использует указанный MAC-адрес в качестве уникального ID дисплей-спота вместо динамически выделяемого ID. Полезно для плееров, у которых нет локального хранилища для хранения уникального ID.
- `spot-id=123XYZ` применяет 123XYZ пользовательский ID на споте вместо того, чтобы полагаться на автоматическое назначение ID. Иногда полезно, если вы хотите привязать плеер к определенному ID, а автоматическое присвоение ID по какой-то причине не может быть применено.
- `class=CLASSNAMES` добавляет CLASSNAMES (через запятую, если их несколько) к корневому элементу отображения. Позволяет отображать одни и те же блоки с разными стилями CSS на разных спотах.
- `tags=tag1,tag2,tag3` применяет указанный набор тегов к споту. Это похоже на настройку фиксированного набора тегов в общих настройках дисплея, но позволяет указать их как часть URL-адреса.
- `param-XXX=value` устанавливает начальное значение [параметра спота](#) с именем XXX к значению. Это похоже на назначение параметров в диалоговом окне настроек, как описано выше, но позволяет указать их как часть URL-адреса.
- `spot=group.name or Location ID` позволяет любому активному [Локатору](#) найти указанный спот, используя его имя (с именами группы и спота, разделенными точкой) или [Location ID](#) спота. Это может быть полезно в сочетании с режимом чтения QR-кода Локатора, при этом один и тот же QR-код можно использовать как для указания вашего местоположения, так и для открытия нужного URL-адреса спота с помощью мобильного устройства посетителя.

Например, чтобы использовать дополнительный дисплей-спот, управляемый отдельным окном браузера, а также собственное имя класса, вы должны использовать такой URL-адрес:

```
http://pixi.guide/spot?display=2&class=myspecialclass
```

Visitor ☐ (гость, посетитель)

Представляет мобильные устройства, такие как смартфон, планшет iPad или Android. Гостевые споты могут быть такими:

- личными смартфонами посетителей (что предполагает наличие доступной для посетителей локальной сети Wi-Fi для связи с сервером Blocks);
- прокатными устройствами для посетителей;
- управляющими планшетами операторов (для персонала).

В отличие от обычных дисплей-спотов, когда вы знаете точные размеры дисплея при добавлении спота, гостевой спот (Visitor spot) может быть использован на разных устройствах с разными размерами экрана, разрешениями и ориентацией. Это особенно актуально, если вы разрешаете посетителям использовать их собственные мобильные устройства. Следовательно, при добавлении спота Visitor, как правило, приходится рассматривать наименьший общий знаменатель с точки зрения размера экрана.

Использование более крупного или более продвинутого дисплея обычно не является проблемой (за исключением того, что кнопки и другие элементы управления могут выглядеть слишком большими). Однако ориентация при проектировании на более продвинутый дисплей, чем у некоторых ваших пользователей, может привести к тому, что какие-то элементы исчезнут с экранов пользователей. Поэтому лучше быть немного консервативными. Например, можно принять, что дисплей iPhone SE является подходящим общим знаменателем. Разрешение экрана составляет 1136 пикселей (высота) на 640 пикселей, но некоторые пиксели теряются в адресной строке сверху и кнопках навигации внизу, в результате получается около 920 на 640 пикселей эффективной области отображения.

Обратите внимание, что точные значения пикселей, которые вы указываете, не так важны, вы, по сути, указываете соотношение сторон и координатное пространство для размещения. Хорошо, если размеры контента и отображения совпадают, но контент в конечном итоге будет масштабироваться в соответствии с шириной любого используемого устройства.

Если вы хотите сделать высоту более динамичной, используйте блок Штабель [Stack Block](#) и установите для основного контента «Заполнить остаток» "Fill Remainder".

Доступ к гостевому споту

В отличие от дисплей-спотов (Display Spot), где есть взаимно однозначное соответствие между физическим дисплеем и элементом в списке спотов, любое количество физических устройств может быть связано с одним гостевым спотом из этого списка. Все мобильные устройства, подключённые к URL-адресу гостевого спота, будут отображать один и тот же контент. После добавления гостевого спота в список спотов можно получить доступ к нему с помощью URL-адреса, например:

```
http://10.2.0.10:8080/spot/?mobile=Mob1
```

Это очень похоже на URL-адрес, используемый для дисплей-спота (Display Spot), см. раздел ["Adding a Display"](#) (Добавление дисплея). Замените 10.0.2.10 на IP-адрес вашего сервера. Число «:8080» после IP-адреса – это номер порта, используемого сервером. Этот номер может быть другим в зависимости от конфигурации вашего сервера. Если вы используете стандартный http-порт «:80», то не нужно указывать эту часть в URL-адресе.

- ◆ Причина, по которой сервер Blocks не использует стандартный порт 80, заключается в том, что некоторые операционные системы не позволяют приложениям использовать порты ниже 1024 без авторизации. Windows не имеет этого ограничения, что позволяет легко переключиться на порт 80, если это необходимо. Но для согласованности порт по умолчанию также в Windows — 8080. При использовании нашего Linux-сервера можно использовать стандартный порт 80, так как имеется внутреннее перенаправление.

Последняя часть, начинающаяся со знака вопроса, указывает, что это Visitor spot (гостевой спот). Имя, введенное после знака равенства, является названием гостевого спота (здесь «Mob1»). Если гостевой спот находится внутри [группы спотов](#), вам необходимо добавить к этому имени префикс с названием окружающих его групп. Например, если он находится внутри группы под названием «Mobiles», вы должны использовать такой URL-адрес:

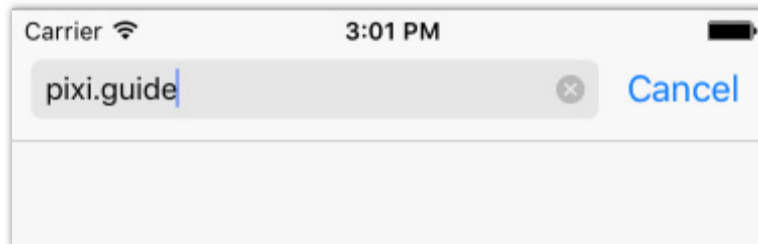
```
http://192.168.0.33:8080/spot/?mobile=Mobiles.Mob1
```



Хотя такая конфигурация подходит для локально управляемых устройств и устройств, предоставленных во временное пользование, она не очень практична для посетителей. Чтобы упростить регистрацию посетителей, вам нужно изменить некоторые параметры сервера:

- Заставьте сервер использовать порт 80.
- Установите перенаправление сервера по умолчанию на `"/spot/?mobile=Mob1"` (или любой другой путь к нужному гостевому споту).

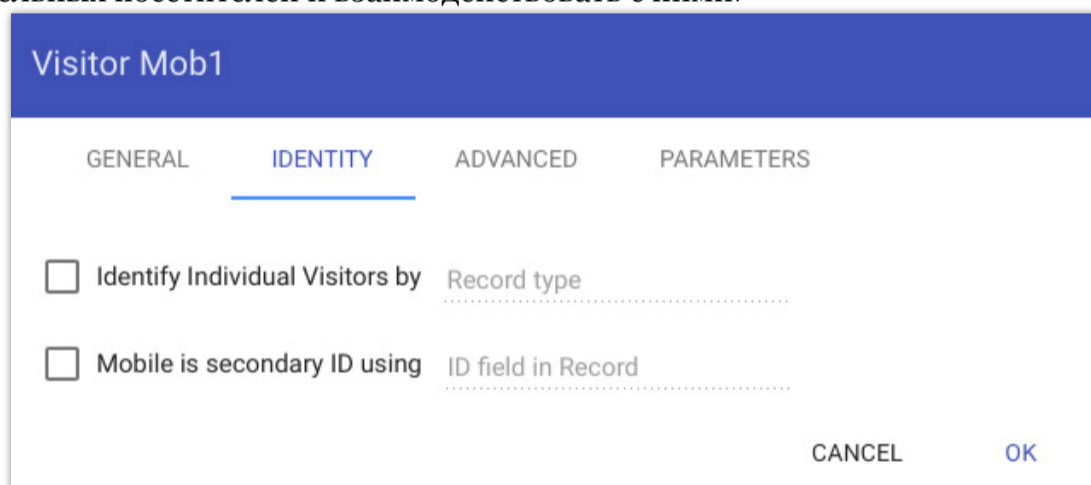
Оба эти параметра описаны в разделе "[Server Configuration Options](#)" в Blocks wiki. Как только это будет сделано, достаточно ввести только IP-адрес сервера в браузере. Чтобы еще больше упростить процесс, вы можете использовать DNS-сервер, который связывает легко запоминающееся имя с IP-адресом сервера. Этот DNS используется доступной для посетителей сетью. Предполагая, что вы указали имя «`pixi.guide`» на IP-адрес сервера (10.2.0.10 в приведенном выше примере), вы можете просто ввести `pixi.guide` в поле адреса браузера вашего мобильного телефона, чтобы получить доступ к функциям гида, как показано ниже.



- ◆ Наш сервер [Linux-based](#) включает в себя всё необходимое для настройки, как описано выше, включая функции DNS-сервера.

Visitor Identity (идентификация посетителя)

Вкладка «Идентификация» в настройках области для посетителей позволяет вам идентифицировать отдельных посетителей и взаимодействовать с ними.



- ◆ ВАЖНО: Эти настройки вступают в силу, только если в вашей лицензии активирован сбор данных о посетителях [Visitor Data Collection](#).

Identify Individual Visitors by (идентифицировать отдельных посетителей по)

Выберите этот параметр и введите имя допустимого типа записи (Record Type), чтобы включить отслеживание и взаимодействие с отдельными посетителями. Эта опция предоставляет возможность контролировать и взаимодействовать с каждым человеком через его мобильное устройство. Должен быть предоставлен пользовательский сценарий ([User Script](#)) определяющий тип записи, указывающий, какие данные нужно собирать. Этот пользовательский сценарий также контролирует, как взаимодействовать с посетителем или что происходит во время посещения.

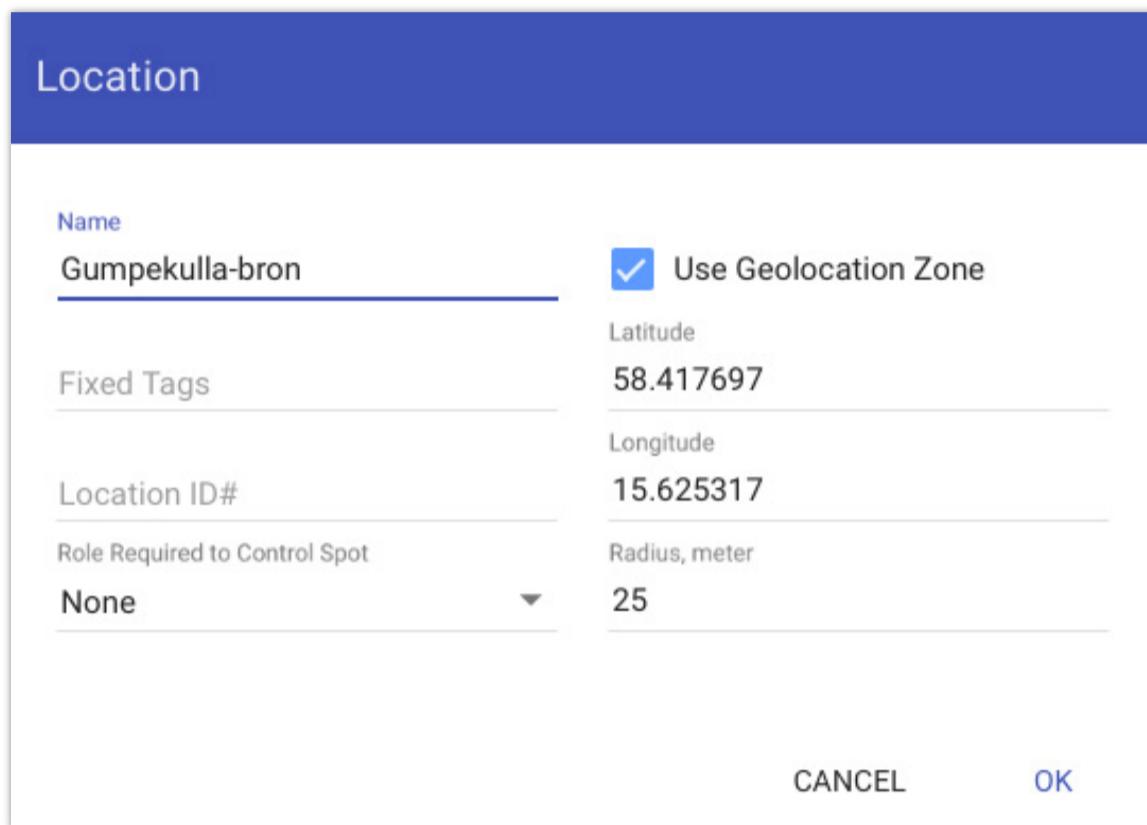
Mobile is Secondary ID (мобильный является вторичным ID)

Выберите этот параметр, если мобильное устройство, получающее доступ к гостевому споту (Visitor), не является основным средством идентификации посетителя. Например, вы можете предпочесть использовать метку RFID или другой ID токен в качестве основного средства идентификации, чтобы не требовать, чтобы у каждого посетителя было мобильное устройство. Однако, если у посетителя есть мобильное устройство, вы все равно можете использовать его для дальнейшего улучшения и персонализации посещения. Если это так, выберите этот параметр, а затем введите имя поля ID в Record Type (тип записи), нужное для хранения ID мобильного устройства. Ваш пользовательский сценарий отвечает за назначение идентификатора мобильного телефона (называемого внутри сценария его *PUID* - постоянным уникальным идентификатором) в этом поле, чтобы затем Blocks могли связать ID токен и мобильное устройство с одной и той же записью данных.

Location 📍 (местоположение)

Представляет физическое местоположение или объект интереса (например, автомобиль на иллюстрации в разделе ["Spots"](#) в главе «Основные понятия»). Это полезно, когда вы хотите связать контент с местоположением для доступа с мобильных устройств посетителей. Это может быть аудиоклип для аудиогuida, видео, текстовое описание или что-то еще, что вы хотите представить для этого места.

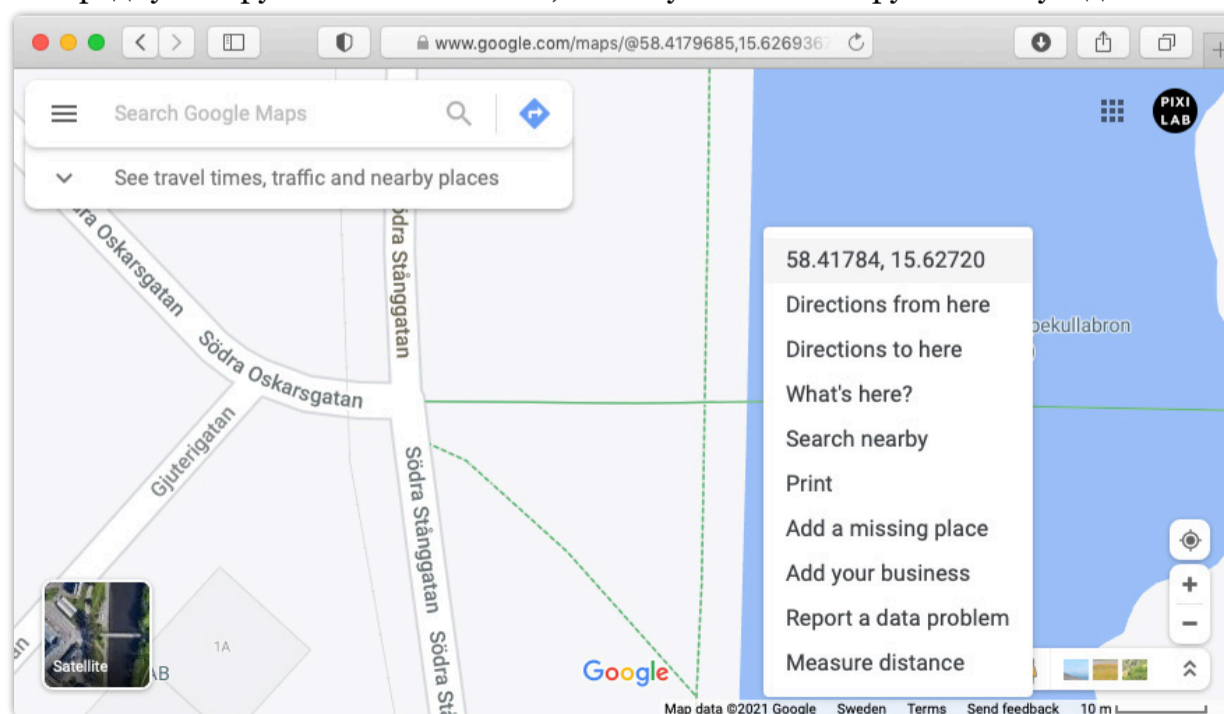
Добавьте местоположение в список спотов (сначала вы можете создать [Группу спотов](#) для этой цели). Дайте ему имя и назначьте ID местоположения. Этот ID должен быть сообщен посетителю, часто с использованием какого-либо четко обозначенного знака в этом месте. Свяжите контент, который вы хотите отображать на мобильном устройстве посетителя, с этим местом. Затем с помощью Локатора ([Locator](#)) посетитель вводит ID местоположения спота, чтобы получить доступ к контенту (например, 44, показанный ниже для спота WATCHOUT)..



Location	
Name	<input type="checkbox"/> Use Geolocation Zone
Gumpekulla-bron	Latitude
	58.417697
Fixed Tags	Longitude
	15.625317
Location ID#	Radius, meter
Role Required to Control Spot	25
None	
CANCEL OK	

Use Geolocation Zone (использовать геолокацию)

Выберите эту опцию и введите местоположение и радиус, чтобы использовать функцию GPS локатора для определения местоположения этого спота. Это обеспечивает альтернативный метод позиционирования, основанный на вашей позиции GPS, а не на вводе ID номера местоположения на цифровой клавиатуре. Вы можете получить геолокацию с помощью таких приложений, как [Google maps](#), и подобные приложения для мобильных телефонов. Укажите значительный радиус вокруг местоположения, чтобы учесть некоторую ошибку в данных GPS.



Щелкните правой кнопкой мыши на картах Google, чтобы выбрать положение GPS.

- ◆ **ПРИМЕЧАНИЕ.** Геолокация на основе GPS надежно работает только на открытом воздухе. Здания и другие крупные сооружения могут значительно снизить точность.

Перекрывающиеся зоны

Зоны геолокации могут перекрываться. Когда это случается, Blocks выбирает наименьшую совпадающую зону. Может случиться, что вы захотите создать очень большую зону, охватывающую все ваши фактические точки интереса. Это позволит отображать контент за пределами этой локации, например, карту, показывающую, где находятся точки интереса, или какое-либо другое сообщение, объясняющее, как их найти.

WATCHOUT спот

Спот WATCHOUT позволяет воспроизводить звук, синхронизированный с презентацией Dataton WATCHOUT, что иногда используется в кинотеатрах или других специальных местах, таких как проекция на изогнутый экран или какой-либо физический объект (мэппинг). WATCHOUT может воспроизводить звуковую дорожку, связанную с шоу, можно также предоставить описание на дополнительных языках через мобильные устройства посетителей.

- ◆ В отличие от спотов Display и Visitor, блок, связанный с спотом WATCHOUT, не будет отображаться на экране WATCHOUT. Блок будет отображаться только на спотах посетителей при переходе к местоположению спота WATCHOUT.

Кроме добавления WATCHOUT spot следует добавить и настроить кластер WATCHOUT на странице Manage (управление). Вы также должны получить звуковую дорожку, в точности соответствующую временной шкале шоу WATCHOUT. Звуковая дорожка должна совпадать с временной шкалой с самого её начала, даже если контент на временной шкале не начинается с начала. Например, если контент на временной шкале WATCHOUT начинается через пять секунд, вы должны добавить эти пять секунд молчания в начале звукового клипа для точного соответствия шкал.

- ◆ Для воспроизведения на мобильных устройствах звуковой файл должен быть закодирован в формате AAC, который обычно хранится в файле .m4a.

WATCHOUT Cluster (Кластер WATCHOUT)

Введите сюда имя кластера, как указано на странице [Manage](#) (управление).

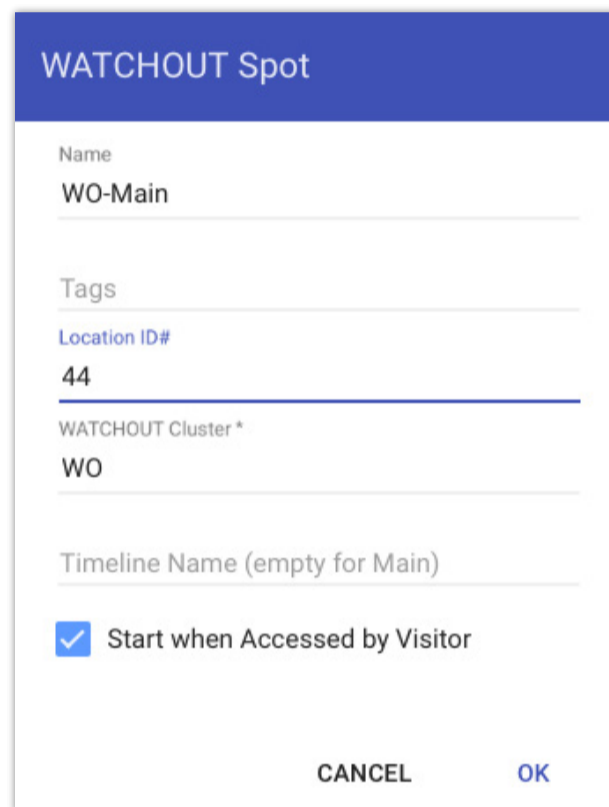
Timeline Name (имя временной шкалы)

Введите имя временной шкалы WATCHOUT для синхронизации. Оставьте это поле пустым для синхронизации с основной временной шкалой.

Start when Accessed by Visitor (запуск при появлении посетителя)

Если этот флажок установлен, приложение Blocks сообщает программе WATCHOUT, что нужно начать воспроизведение, когда посетитель обращается к споту, введя идентификатор местоположения (Location ID) этого спота в мобильное устройство. Используйте этот метод, чтобы начать просмотр WATCHOUT «по требованию» первым посетителем. Последующие посетители смогут присоединиться к просмотру в текущий момент времени и не будут перезапускать шоу.

Если шоу WATCHOUT запускается другим способом (например, по запланированной задаче [Task](#)), оставьте это поле без «галочки».



Группа спотов

Группа спотов (Spot Group) действует как «папка», позволяя группировать споты способами, которые имеют смысл для вашей инсталляции. Вы можете использовать группу только для организационных целей, представляя этаж или комнату в здании. Группы спотов могут быть вложены друг в друга, что позволяет иметь внешние группы для этажей с вложенными в них группами для комнат на каждом этаже.

Можно использовать группу спотов (Spot Group) для соединения родственных дисплеев. Например, у вас может быть набор из трёх дисплеев, работающих вместе. Поместите их в группу спотов (Spot Group), а затем назначьте контент всем трём дисплеям за один раз, перетаскив блок в группу спотов (Spot Group), не тратя времени на каждый отдельный спот. Используя тэги для спотов внутри группы в сочетании с блоком выбора тэгов (Tag Selector), вы можете настроить распределение контента между дисплеями, например, пометить дисплеи слева, в центре и справа, и соответственно настроить селектор тэгов ([Tag Selector](#)).

Наконец, вы можете использовать группу спотов для синхронизации воспроизведения видео в нескольких дисплей-спотах :

- Поместите эти дисплей-споты в группу спотов.
- Пометьте каждый дисплей соответствующим образом (например, левый, центральный и правый).
- Используйте синхронизатор для воспроизведения видео с соответствующими тегами, назначенными каждому из видео, чтобы заставить их воспроизводиться на нужном дисплее.
- Назначьте этот синхронизатор группе спотов (что делает его применимым ко всем спотам в группе, если он не будет переопределен).
- При необходимости назначьте один из дисплеев в группе в качестве [Preferred Synchronizer Master in Group](#) (предпочтительного главного синхронизатора в группе).

См. также "[Synchronizing Video Across Multiple Displays](#)" (Синхронизация видео на нескольких дисплеях).

5. БЛОКИ

Управление контентом осуществляется с использованием различных типов блоков. Хотя многие типы блоков могут показаться довольно простыми, их истинные широкие возможности становятся очевидными при использовании их в комбинации. Вводная информация – в разделе [Blocks](#) главы «Основные понятия».

Блок может быть либо на верхнем уровне страницы [Display](#), либо вложен в другой блок. Блоки верхнего уровня можно вручную назначить спотам (Spots), перетаскив блок на спот (Spot) на странице Display. Они также могут быть динамически назначены при помощи кнопок и задач (buttons, tasks). Для автоматического изменения назначения блока в зависимости от времени, даты или дня недели используйте расписание [Schedule](#). Когда назначение блока изменяется, соответствующие дисплеи обновляются.

Создание блоков

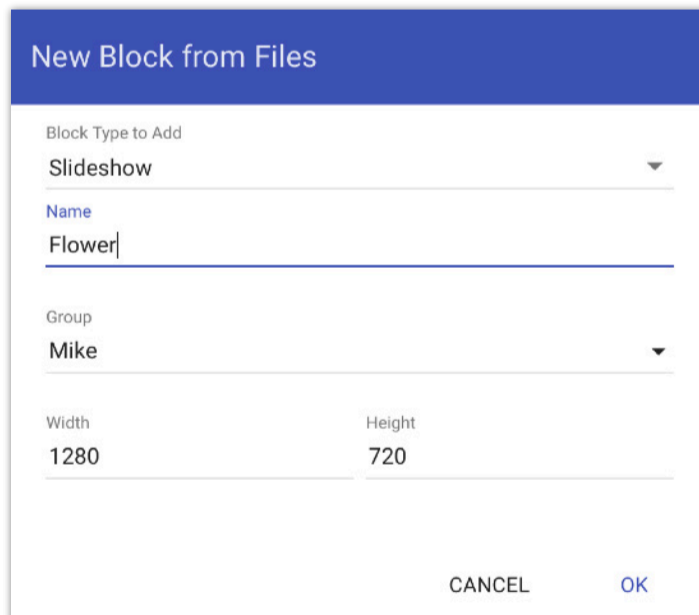
Создайте блок верхнего уровня, используя меню Block на странице Display, как показано справа. Многие типы блоков могут содержать внутри себя другие блоки. Например, слайд-шоу часто содержит изображения и видео, но также может содержать более сложные типы блоков, такие как Composition или Web.

Вот некоторые из наиболее часто используемых типов блоков:

- Composition (композиция) - для объединения нескольких блоков, кнопок или других элементов управления на одной странице.
- Book (книга) - для размещения контента на нескольких страницах.
- Slideshow (слайд-шоу) - для показа последовательности изображений или других блоков с переходами.

Создание блока из медиафайлов

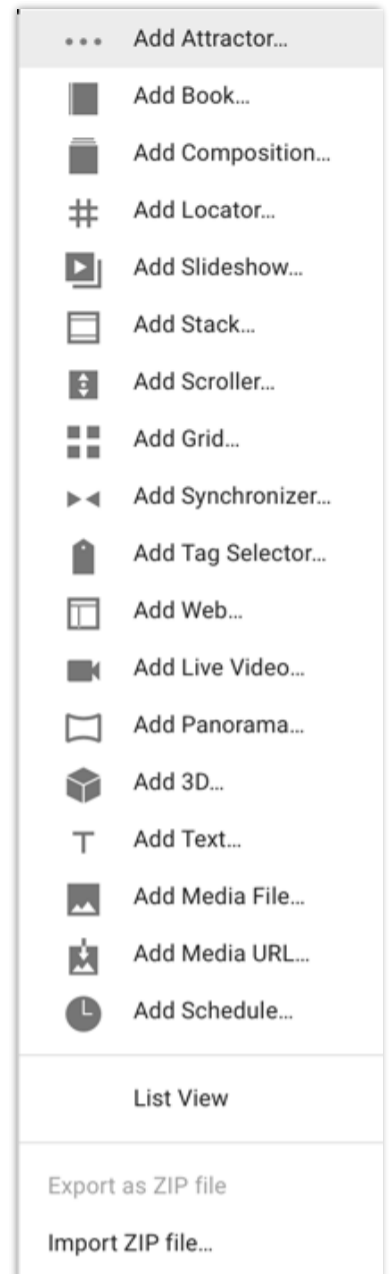
В качестве альтернативы ручному методу, описанному выше, вы можете создать некоторые типы блоков из набора медиафайлов. Например; чтобы создать слайд-шоу из ряда изображений, просто поместите эти изображения на фон страницы «Дисплей», выберите «Слайд-шоу» в появившемся диалоговом окне, укажите имя, группу и другие параметры и нажмите «ОК».



Block Group

При создании блока верхнего уровня вы присваиваете ему имя и связываете его с группой блоков. Изначально существует одна группа под названием Main (основная). Чтобы создать новую группу, введите желаемое имя в поле Group при добавлении блока. Организация блоков по группам облегчает их поиск по мере увеличения количества блоков. Планируя заранее, заранее принимая решение, как назвать и сгруппировать блоки, вы сэкономите время при использовании этих блоков.

Когда у вас много блоков, иногда сложно найти блоки, с которыми вы работаете. Если блоки организованы в группы, вы можете отфильтровать набор блоков, отображаемых на странице Display, с помощью меню Group (Группа). Здесь вы можете выбрать просмотр только одной группы или блоков из всех групп.



List View (просмотр списка)

Выберите "List View" в меню Block чтобы просмотреть блоки верхнего уровня в более компактном виде. Отсортируйте список по любому столбцу, щелкнув его заголовок.

Скрыть блоки в зависимости от роли

Связав группы и отдельные блоки с ролями, вы можете скрыть блоки для пользователей, у которых нет назначенной роли (или выше). Скрытие блоков может упростить использование системы и сделать её менее запутанной. Подробнее см. в разделе "[Авторизация и Роли](#)".

Чтобы ограничить видимость блоков в группе на основе роли:

1. Выберите нужную группу в меню «Группа» на главной странице «Дисплей».
2. Щелкните значок замка рядом с названием группы.
3. Выберите нужную роль в диалоговом окне.

Чтобы настроить видимость отдельных блоков:

1. Дважды щелкните блок на странице Display, чтобы просмотреть его редактор.
2. Выберите роль в раскрывающемся меню "Role Required to See" (роль, необходимая для просмотра).

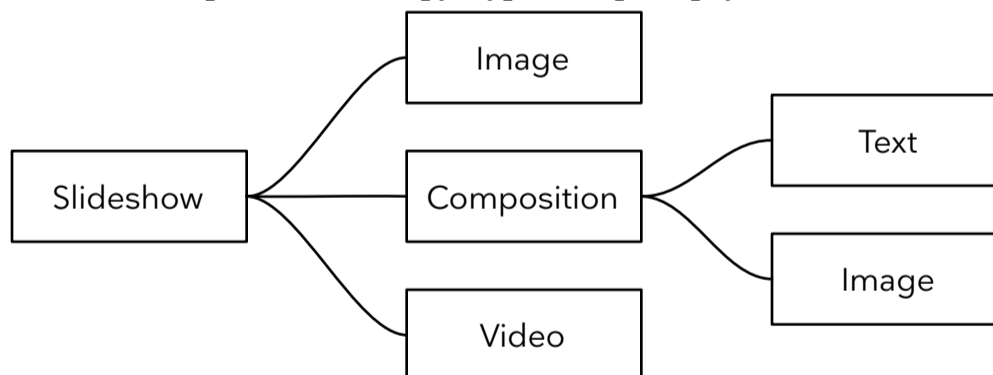
Если вы устанавливаете роль на уровне группы, эта роль наследуется всеми теми блоками в группе, которые явно не переопределяют этот параметр.

◆ Чтобы увидеть эффект, вы должны войти в систему как пользователь с ролью ниже той, которая указана для видимости блока.

Редактирование блоков

Отредактируйте блок, дважды щелкнув его на странице Display. Щелкните логотип Blocks в левом верхнем углу, чтобы вернуться на страницу Display, или выберите Display в меню Page (страница).

Блоки могут содержать другие блоки в древовидной структуре. Например, у вас может быть иерархия блоков, которая выглядит так:



Просмотрите иерархию блоков, используя кнопку иерархии в верхней части списка дочерних блоков. Представление иерархии позволяет перейти непосредственно к любому блоку, дважды щелкнув его.



Блок, который содержит вложенные блоки, такие как композиция на иллюстрации выше, открывается двойным щелчком по нему в списке дочерних блоков ([Child Blocks](#)). Находясь внутри такого вложенного блока, вдоль нижней части окна показывается строка навигации, путь к этому блоку. Щелкнув на имя внешнего блока, вы вернетесь на уровень этого блока.



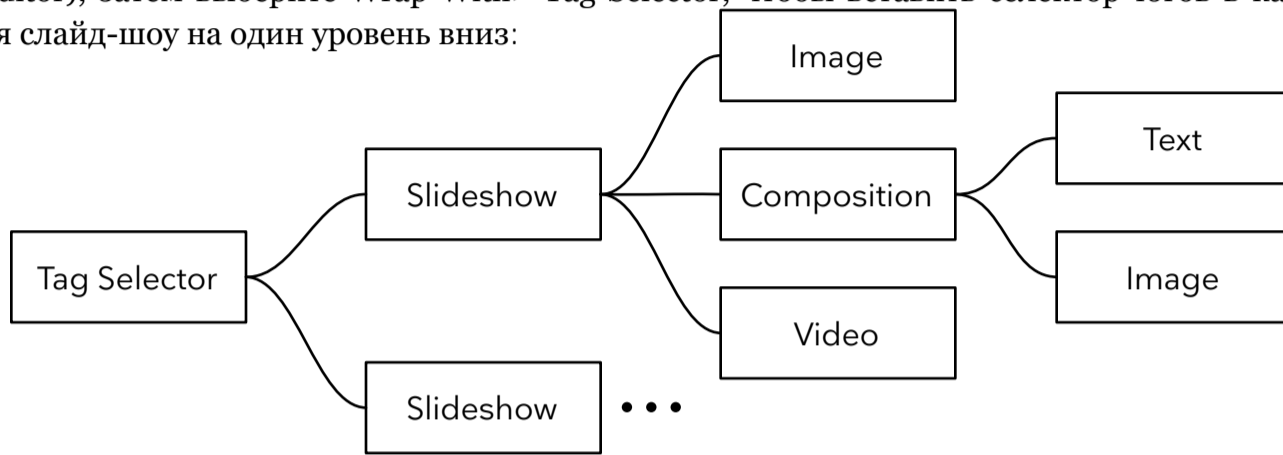
Строка навигации показывает типы и имена вмещающих блоков, что позволяет вам вернуться к любому внешнему уровню.

Добавление дочерних блоков (Child Blocks)

Добавьте дочерний блок, перетащив мультимедийные файлы в список дочерних блоков [Child Blocks list](#), или через меню “Add Child” (добавить дочерний блок). Различные типы блоков описаны далее в этой главе. Про изображение, видео и звук – см. раздел “[Media File](#)” в этой главе.

Сборка корневого блока (Wrapping a Root Block)

Меню «Wrap With» отображается только на самом верхнем уровне блока и позволяет вставить новый блок на корневом уровне, перемещая предыдущий уровень, бывший корневым, на уровень ниже, внутрь добавляемого блока. Например, начав с иерархии блоков со слайд-шоу в качестве внешнего блока (см. рис. выше), вы можете через некоторое время понять, что вам нужен селектор тэгов [Tag Selector](#) в качестве внешнего блока, чтобы представить два разных слайд-шоу в зависимости от некоторого тэга (например, для поддержки нескольких языков). Для этого откройте редактор слайд-шоу (Slideshow editor), затем выберите Wrap With > Tag Selector, чтобы вставить селектор тэгов в качестве корневого блока, перемещая слайд-шоу на один уровень вниз:



Сборка корневого блока слайд-шоу при помощи селектора тэгов. Это позволяет добавлять другие блоки, например второе альтернативное слайд-шоу.

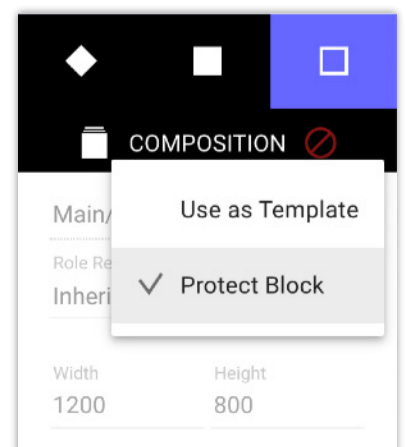
Предварительный просмотр блока во время редактирования

При редактировании сложного блока со многими уровнями вложенности иногда бывает полезно увидеть результат в контексте редактирования. Вы можете сделать это одним из следующих способов:

- Прежде чем приступить к редактированию, перетащите блок на дисплей-спот, который вы используете в качестве монитора для предварительного просмотра. Выберите “[Update on Block Change](#)” (обновить при изменении блока) на вкладке Advanced в настройках дисплей-спота. Это позволит использовать реальный спот (например, проигрыватель PIXILAB) для предпросмотра. При изменении настроек в редактируемом блоке дисплей-спот будет автоматически выбран для отображения редактируемого блока.
- Выберите “Show Editor Preview” (показать предпросмотр редактора) в меню Edit во время редактирования блока. Это откроет отдельное окно на вашем компьютере, показывающее результат в контексте. Обратите внимание, что это окно предварительного просмотра не является полноценным спотом, поэтому такой метод может не сработать. Если ваш блок зависит от параметров спотов [Spot Parameters](#) или других свойств спотов, используйте предыдущий метод для предварительного просмотра блока при редактировании.

Экспорт и импорт блоков

Используйте команды в нижней части меню Block для экспорта и импорта блоков либо для архивирования, либо для передачи в другую систему Blocks. Чтобы экспортировать блок, сначала выберите его, а затем выберите «Export as ZIP file». Это загрузит ZIP-файл с именем блока на ваш компьютер. Такой ZIP-файл можно позже импортировать с помощью команды «Import ZIP file». Выберите ZIP-файл, содержащий ранее экспортированный блок, затем укажите желаемую группу и имя для импортированного блока.



Защита блока и шаблоны (Templates)

Вы можете защитить блок или его контент от случайного изменения или пометить блок как шаблон, на основе которого создаются новые блоки. Эти функции выбираются в меню Protection, доступ к которому осуществляется с помощью символа, показанного в правом верхнем углу панели настроек рядом с названием типа блока. Цвет и форма символа указывают на текущую защиту. Щелкните символ, чтобы открыть меню, как показано справа. В зависимости от контекста это меню содержит комбинацию следующих элементов:

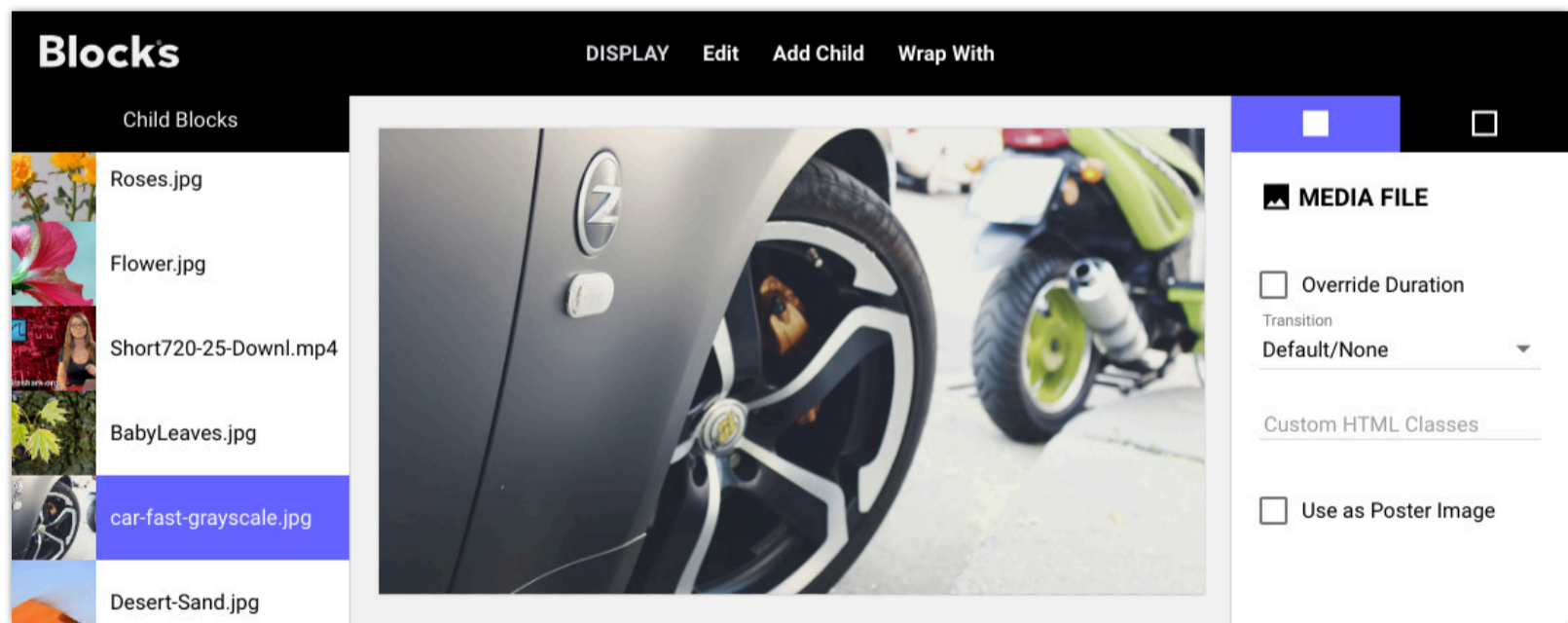
- Use as Template (использовать как шаблон) доступно только на уровне корневого блока и помечает блок как шаблон (см. ниже).
- Protect Block (защита блока) доступна для всех блоков на всех уровнях. Выберите эту опцию для блоков, которые не следует удалять, перемещать или изменять. Если вы не выберете также “Protect Block Content” (защитить контент блока), вы все равно можете изменить любые дочерние блоки внутри.
- Protect Block Content (защита контента блока) доступна для текстовых блоков и блоков, которые могут содержать дочерние блоки, например композиции. Если этот флажок установлен, вы не сможете редактировать текст или спуститься в блок, дважды щелкнув его.
 - ◆ **ПРИМЕЧАНИЕ:** Выбор “Use as Template” или “Protect Block” для корневого блока предотвращает удаление этого блока со страницы «Библиотека». Это служит дополнением к защите, обеспечиваемой текущей ролью пользователя.

Использование шаблонов

Если вы часто создаете похожие блоки, вы можете сэкономить время, сначала создав блок с основным контентом. Затем отметьте этот блок как “Use as Template” (использовать как шаблон). Когда вы позже попытаетесь отредактировать этот блок, вам будет предложено сначала создать дубликат, используя другое имя блока. Затем эта копия будет открыта для редактирования, а исходный шаблон останется без изменений. Пользователи с [ролью Creator](#) или выше могут редактировать шаблон или делать копию шаблона. Пользователи ниже роли Creator не могут редактировать шаблон, но могут создавать и редактировать копию шаблона.

Слайд-шоу

Слайд-шоу представляет свои дочерние блоки последовательно, с переходом между ними.



Дочерние блоки (Child Blocks)

Дочерние блоки, из которых составляют слайд-шоу, отображены списком в левой части окна редактора. Изменить порядок появления блоков можно простым перетаскиванием. Дважды щелкните на составной блок, чтобы открыть его редактор (не относится к простым блокам, таким как изображения).

Щелкните дочерний блок в списке или сплошной квадрат в верхнем правом углу, чтобы просмотреть настройки выбранного дочернего блока. Эти настройки различны для блоков разных типов и зависят также от типа внешнего блока.

Override Duration (переопределить длительность)

По умолчанию слайд отображается в течение времени, указанного в настройках слайд-шоу (см. далее). Видео отображается в течение всей его продолжительности. Щёлкните на **Override Duration**, чтобы изменить продолжительность конкретного дочернего блока, затем установите нужную продолжительность, перетаскив ползунок или введя числовое значение в это поле. Ползунком можно задавать продолжительность только до 10 секунд, в поле вы можете ввести любую продолжительность.

Transition (переход)

Если установлено значение «По умолчанию/Нет» (Default/None), то будет использован переход, указанный для слайд-шоу в настройках его внешнего блока (см. ниже). Выберите другой тип, и этот переход будет действовать только для этого блока.

Пользовательские классы CSS

Применяет класс CSS (или несколько классов, разделенных пробелами) к блоку, чтобы придать ему особый стиль при использовании в сочетании с пользовательским CSS. Подробности см. [Custom Styling](#) (пользовательские стили).

Use as Poster Image (использовать как заставку)

Выбирает дочерний блок для представления блока на странице Display. По умолчанию это первый видимый дочерний блок. Вы можете выбрать более представительный дочерний блок, используя этот флажок. Этот параметр не влияет на воспроизведение блока.

Настройки внешнего блока

Щёлкните на пустой квадратик в правом верхнем углу, чтобы отредактировать настройки слайд-шоу в целом. Многие из этих настроек также доступны для блоков других типов. Щёлкните на символ пера рядом с названием блока, чтобы переименовать этот блок или переместить его в другую группу блоков [Block Group](#).

- ◆ Переименование или перемещение блока приведет к разрыву ссылок на этот блок (например, **Spot assignments** и [Reference Blocks](#)), и вам придется заново устанавливать их вручную.

Role Required to See (роль, необходимая для просмотра)

Эта опция появляется только на верхнем уровне и позволяет ограничить определенной ролью круг лиц, которые могут видеть этот блок на странице Display.

Width/Height (ширина/высота)

Размеры блока. Вы устанавливаете эти размеры при первоначальном создании блока. Здесь можно изменить его настройки. Как правило, вы не изменяете эти настройки после создания блока. Если такое случится, следует пересмотреть места, где используется этот блок, чтобы убедиться, что он выглядит так, как ожидалось. Как правило, размеры блока должны соответствовать разрешению спота, на котором он будет отображаться, или, по крайней мере, иметь такое же соотношение сторон.

Custom CSS URL and Custom CSS Classes (пользовательский URL-адрес CSS и пользовательские классы CSS)

Эти поля позволяют настраивать стили кнопок и других элементов.

- ◆ **ПРИМЕЧАНИЕ:** Стиль CSS — это расширенная функция. Дополнительные сведения см. в разделе [Custom Styling](#) (пользовательские стили).

Поле «Custom CSS URL» применяется только к блокам верхнего уровня и может быть одним из следующих:

- Полный URL-адрес, начинающийся с «http://» или «https://», указывающий на любой доступный веб-сервер.
- Абсолютный URL-адрес сервера Blocks, начинающийся с «/public/», для загрузки файла CSS, хранящегося в каталоге public на вашем сервере Blocks.
- Локальный URL-адрес блока, начинающийся с «~/» (например, ~/style.css), для загрузки файла CSS, хранящегося в папке блоков на сервере. Это полезно для CSS, который применяется только к одному блоку. Любой такой файл CSS, хранящийся в папке блока, будет включен при [экспорте/импорте](#) блока.
 - ◆ Создайте или отредактируйте локальный файл CSS, щелкнув значок пера в поле «Custom CSS URL» (пользовательский URL-адрес CSS). Это создает локальный CSS файл блока, если он еще не существует, и открывает окно редактора для этого файла. Доступ к этой функции также можно получить с помощью команды «Edit Block CSS» (редактировать CSS блока) в меню Edit, которая также доступна при редактировании дочернего блока.

Наконец, можно автоматически загружать пользовательский CSS глобально для всех спотов в системе, используя параметр defaultSpotCSS в файле конфигурации сервера ([server configuration file](#)).

Play Automatically and Loop (автовоспроизведение и цикл)

В большинстве случаев требуется, чтобы слайд-шоу воспроизводилось сразу. Если слайд-шоу является блоком верхнего уровня, вы, как правило, хотите его зациклить. Если слайд-шоу - внутренний блок (например, во внешнем слайд-шоу), вы можете захотеть, чтобы внутреннее слайд-шоу воспроизводилось только один раз, а затем внешний блок продолжил свою работу.

Pause Video when Slideshow Paused (видео на паузе, когда слайд-шоу приостановлено)

Если выбрано, текущий слайд будет приостановлен, когда слайд-шоу будет приостановлено. Иногда это полезно для слайдов, содержащих видео. Если этот параметр не выбран, слайд-шоу будет поставлено на паузу, но любое видео продолжит воспроизведение до конца и остановится, ожидая возобновления воспроизведения слайд-шоу.

Default Slide Duration (продолжительность слайда по умолчанию)

Определяет, как долго показывать дочерние блоки, которые не имеют заданной продолжительности, например неподвижные изображения. Видео или другой дочерний блок, который имеет заданную продолжительность, будет показан полностью (если вы не переопределите продолжительность показа этого дочернего блока).

Default Transition (переход по умолчанию)

Переход, используемый между дочерними блоками в этом слайд-шоу, если не переопределено для дочернего блока.

Доступны следующие типы переходов:

- **Cut (прямая склейка)**. Никаких эффектов при переходе. Новое изображение (или другой дочерний блок) появляется мгновенно.
- **Dissolve (растворение)**. Новое изображение постепенно заменяет предыдущее.
- **Crossfade**. Новое изображение появляется постепенно поверх, в то время как предыдущее исчезает. Полезно в качестве альтернативы Dissolve при переходе между изображениями с разным соотношением сторон или когда одно из изображений содержит прозрачные области.
- **Fade through Color (исчезание через цвет)**. Черное или цветное сплошное изображение закрывает текущее изображение, а затем исчезает, открывая следующее изображение.
- **Slide In (слайд)**. Новое изображение скользит поверх старого. Можно указать направление соскальзывания.
- **Push (вытеснение)**. Новое изображение наползает по мере того, как старое изображение «отползает».
- **Pop (всплывание)**. Новое изображение всплывает поверх старого из-за него.
- **Zoom (масштаб)**. Старое изображение исчезает и масштабируется вниз, когда появляется новое, с дополнительным эффектом вращения (spin).

Для большинства переходов возможно указать продолжительность (то есть его скорость). Некоторые переходы имеют дополнительные настройки. Чтобы увидеть действие этих настроек, измените настройки, щёлкните на предыдущий слайд в списке, а потом вернитесь к текущему слайду.

Переходы для звука

Чтобы выполнить переход для звука, выберите один из параметров в раскрывающемся списке *Audio Transition*:

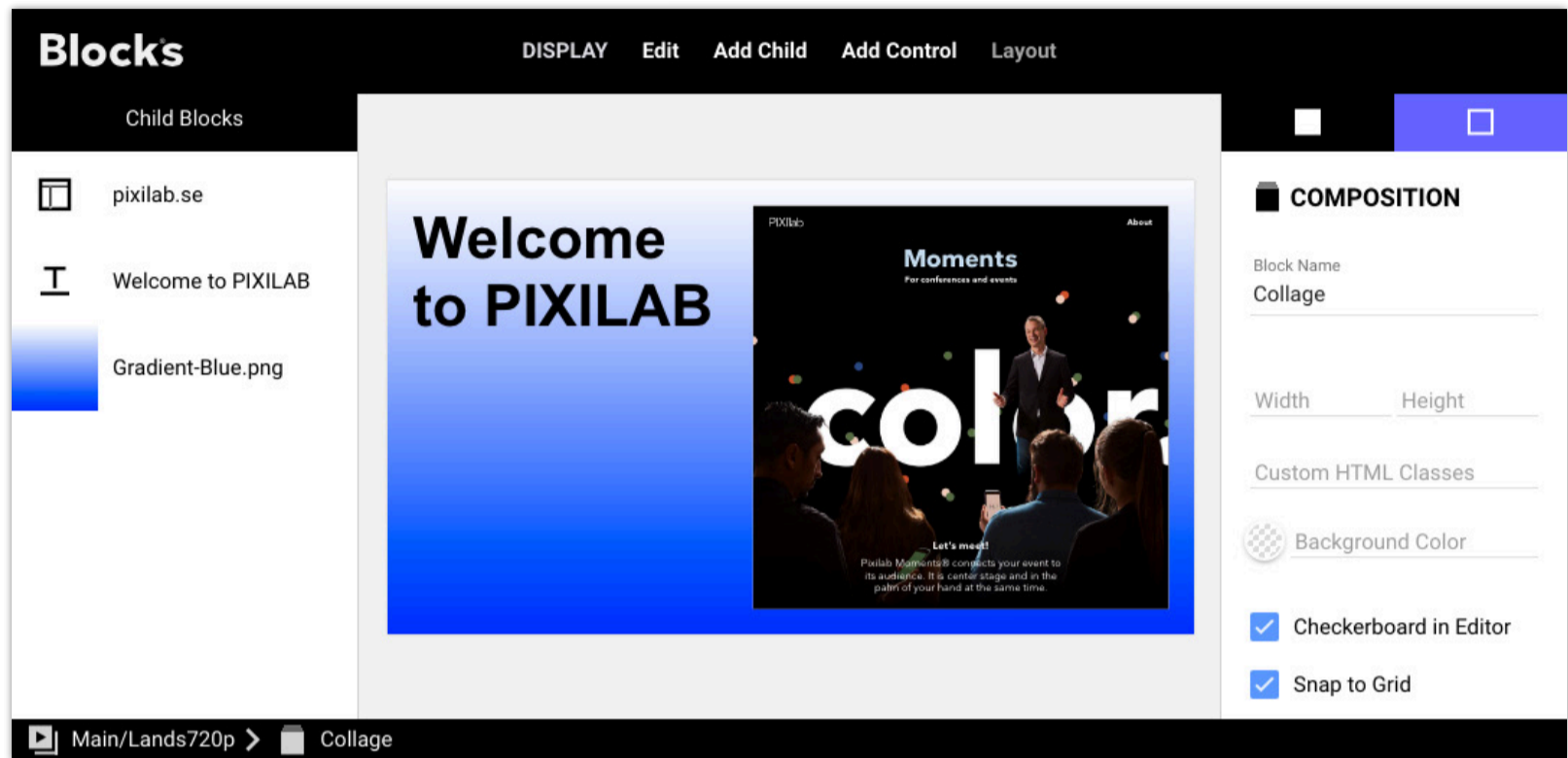
- **None**. Любой входящий звук начинается сразу. Исходящий звук обрывается в конце перехода.
- **Fade In**. Любой входящий звук начинается сразу. Исходящий звук обрывается в конце перехода.
- **Fade Out**. Любой входящий звук начинается сразу. Исходящий звук затухает.
- **In and Out**. Громкость любого входящего звука постепенно увеличивается. Исходящий звук затухает.

Child Replication (дочерняя репликация)

Позволяет автоматически генерировать слайды на основе данных из внешней базы данных или аналогичной. Это делается путем репликации одного прототипа дочернего блока. Дополнительные сведения см. в разделе [Child Replication](#) в блоке «Книга» (Book) ниже.

Композиция (composition)

Блок Composition отображает свои дочерние блоки вместе, рядом или друг над другом, или даже в обратном порядке. На рисунке, показанном ниже, синяя градиентная заливка является самым нижним слоем, а блоки Текст (Text) и Веб-сайт (Website) расположены сверху. Перетащите дочерние блоки в списке, чтобы изменить порядок их укладки. Строка навигации, показанная внизу, показывает, что это вложенный блок. Он находится в слайд-шоу Land-s720p, который является основным блоком группы.



Многие настройки внешнего блока аналогичны описанным выше для блока слайд-шоу.

Block Name (имя блока)

Дайте блоку осмысленное имя, чтобы его было легче найти внутри его внешнего блока. Это имя также можно использовать при переходе к дочернему блоку внутри книги, слайд-шоу и т. п. Таким образом, избегайте изменения имени блока, если вы используете его для поиска блока с помощью кнопки, задачи или чего-то подобного.

Width/Height (ширина/высота)

Хотя иногда бывает полезно переопределить ширину и/или высоту внутреннего блока, в большинстве случаев вы оставляете эти поля пустыми, чтобы наследовать размеры внешнего блока. Если композиция – блок верхнего уровня, эти поля являются обязательными и определяют общий размер блока

Background Color (цвет фона)

По умолчанию фон композиции прозрачен. Это позволяет использовать композиции во внешних композициях в качестве средства группировки внутренних блоков. Если вы хотите, чтобы в композиции был непрозрачный фон, выберите его цвет здесь. В приведенном выше примере это не имеет значения, так как синий градиент заполняет всю композицию..

Checkerboard in Editor (шахматная доска в редакторе)

Задаёт фон, используемый при редактировании композиции. Не применяется, если выбран сплошной цвет в качестве фона. Выберите стиль шахматной доски, чтобы ваш контент был хорошо виден для редактирования.

Editor Scale (масштаб Редактора)

Большая композиция, как правило, автоматически масштабируется, чтобы поместиться в редакторе. В некоторых случаях, например, при редактировании текста, это может быть нежелательно, так как текст получится слишком мелким. Выберите нужный коэффициент масштабирования и прокрутите страницу, чтобы просмотреть нужную часть для редактирования.

Snap to Grid and Grid Size (привязать к сетке и размер сетки)

Если выбрать, элементы, перемещаемые мышью, будут привязаны к пиксельной сетке. Это часто облегчает выравнивание объектов. Можно указать степень детализации движения, выбрав размер сетки. Отмените выбор этого параметра, чтобы свободно перемещать элементы. Вы всегда можете расположить элементы с помощью чисел или с помощью команд в меню Layout, даже если выбран этот параметр.

Keep Children Inside (держат дочерние блоки внутри)

Когда выбрано, дочерние блоки нельзя перетаскивать за пределы окружающей композиции. Отмените выбор этого параметра, если вы хотите использовать композицию для обрезки ее внутренних блоков, показывая только ту часть, которая пересекает композицию.

Hide After Delay (скрыть после задержки)

Если выбрать, композиция (и все ее дочерние элементы) будут скрыты через несколько секунд бездействия. Коснитесь дисплея (или переместите мышь), чтобы они снова появились. Это полезно, когда интерактивные элементы управления, такие как кнопки, размещаются поверх другого контента, тем самым скрывая контент за собой. Если установить этот флажок, элементы управления будут скрыты автоматически. Просто коснитесь дисплея, чтобы они снова появились.

- ◆ Этот параметр доступен только для вложенных композиций, но не для композиций верхнего уровня.

Prototypical Binding (прототипная привязка)

Позволяет указать ожидаемый тип привязки для использования в качестве целевого блока при переходе от реплицируемого блока. Это может быть использовано, например, для отображения дополнительных сведений об источнике данных, связанном с реплицируемым блоком.

- ◆ **ПРИМЕЧАНИЕ.** Это расширенный параметр, используется вместе с репликацией дочерних блоков [child block replication](#).

При переходе от вложенного реплицированного блока к блоку, находящемуся за пределами иерархии это необходимо, поскольку в этом случае целевой блок не будет иметь контекста вложенного блока. Используйте этот параметр, чтобы указать, какой тип контекста привязки следует ожидать дочерним элементам этого композиционного блока, указав тот же тип объекта, от которого вы собираетесь переходить. Затем дочерние элементы (такие как [Text](#) и [Media URL](#)) могут использовать корень свойства Relative в своих привязках.

Настройки дочернего блока (Child Block Settings)

Большинство настроек дочерних блоков в композиции зависят от типа и описаны для каждого типа блока в этой главе. Следующие настройки являются общими для всех дочерних блоков Composition.

- Left, Top, Width, Height (**слева, сверху, ширина, высота**). Положение и размер дочернего блока в композиции. Либо перетащите дочерний элемент или один из его краев или углов в основную область. Чтобы пропорционально изменить размер изображения или видео, нажмите Shift при перетаскивании.
- Custom CSS Classes. Измените внешний вид дочерних блоков, например, примените пользовательские шрифты или стили кнопок и других элементов управления. Дополнительные сведения см. в разделе [Custom Styling](#).
- Blend Mode (**Режим смешивания**). Управляет тем, как пиксели этого дочернего блока [объединяются с пикселями](#) нижележащих блоков этой композиции. Это похоже на соответствующую функцию в таких приложениях, как Adobe Photoshop, и бывает полезно для эффектов, сочетающих изображения, видео или текст — возможно, вместе с поведением, таким как непрозрачность.

Элементы управления и текстовые блоки могут быть привязаны к локальным или системным свойствам для обеспечения интерактивности. Дополнительные сведения см. в разделе [Элементы управления и панели](#).

Layout (макет)

В этом меню предоставлены команды для выравнивания выбранных блоков. Выберите нужные элементы либо в списке дочерних блоков (слева), либо с помощью мыши. Затем выберите команду для выравнивания выбранных блоков.

Align Horizontally (выровнять по горизонтали)

Перемещает выбранные блоки по горизонтали так, чтобы их центр, левая или правая стороны выстроились вдоль одной вертикали.

Align Vertically (выровнять по вертикали)

Перемещает выбранные блоки по вертикали так, чтобы их центр, левая или правая стороны выстроились вдоль одной горизонтали.

- ◆ Если вы выберете один объект, он будет выровнен с окружающей композицией.

Set Width Ratio (задать соотношение по ширине)

Задайте ширину выбранного блока (-ов) в указанном соотношении размеров дочернего и вмещающего блоков. Это упрощает подбор размеров дочернего блока, чтобы заполнить половину (треть, четверть...) ширины вмещающего блока.

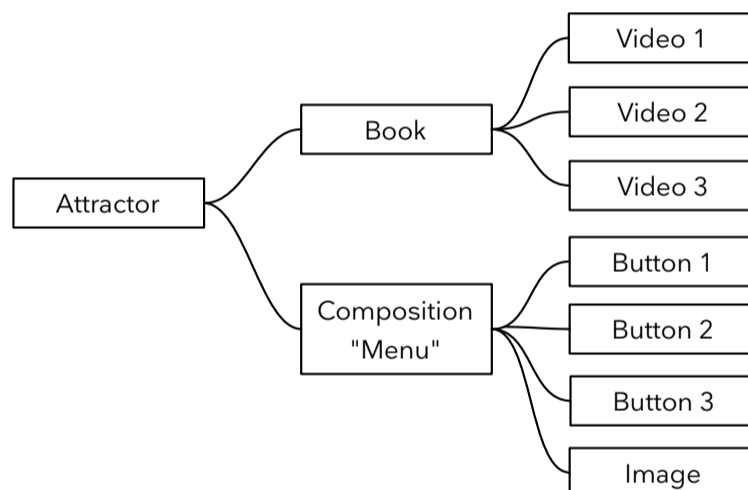
Add Control (добавить элемент управления)

Элементы управления – это такие интерактивные элементы, как Кнопка (Button) или Ползунок - Slider (см. главу “Панели управления”). Эти элементы считаются блоками, но они имеют смысл только в композиции, поскольку никогда не используются в полный размер экрана и должны некоторым образом располагаться друг по отношению к другу. Меню Добавить дочерний блок (Add Child) доступно во всех блоках, которые могут иметь дочерние блоки, а меню Добавить элемент управления (Add Control) доступно только при редактировании композиции.

- ◆ Если вы хотите использовать блоки управления внутри блока другого типа, сначала добавьте композицию (Composition), а затем уже внутрь композиции добавьте блок управления.

Book ■

Книга отображает свои дочерние элементы как отдельные страницы с именами. Вы можете перейти на страницу с помощью задачи [Task](#) или кнопки (см. “[Go to Local Block](#)”). В приведенном ниже примере показана книга с тремя страницами, каждая из которых демонстрирует видео. Кнопки в композиции “Menu” управляют книгой. Книгу и композицию «Меню» помещают в блок [Attractor](#), что заставляет меню каждый раз появляться, когда видео, показанное на странице в книге, заканчивается.



- Page Transition (**переход страницы**). Применяет переход страницы при смене страницы в книге. Доступные переходы см. выше в разделе [Slideshow](#).
- Auto-reverse Transition (**переход автореверс**). Изменение направления некоторых переходов, таких как Push и Slide In, при переходе на предыдущую страницу книги. Часто используется с “Swipe to Turn Pages” (перелистыванием).
- Swipe to Turn Pages (**проведите, чтобы перевернуть страницу**). Поддержка жестов смахивания для навигации пользователя по книге. Если этот параметр не выбран, навигация должна быть приспособлена для использования кнопок или задач.
- Wrap Around at End (**возврат по кругу**). Если выбрано, навигация от последней/первой страницы переходит к другому концу книги. Применяется только в том случае, если выбрано “Swipe to Turn Pages” .

Другие настройки внешнего блока описаны выше в разделах про Слайд-шоу и Композицию.

Special treatment of first/last page (особая обработка первой/последней страницы)

Иногда требуется, чтобы первая и/или последняя страница в книге отличались от других страниц. Например, вы захотите скрыть или выделить серым цветом некие элементы на этих страницах. Для этого Blocks добавляет класс CSS first-child или last-child к элементу страницы, когда они достигают этих пределов. Затем можно использовать [пользовательский CSS](#), чтобы обрабатывать эти страницы по-разному.

Child Replication (дочерняя репликация)

Это расширенная функция, позволяющая несколько раз реплицировать один дочерний блок, при этом каждый репликант привязывается к другому объекту. Эти объекты могут происходить из индексированных подсвойств устройства (например, каналов на аудиомикшере) или элементов источника данных, такого как база [база музейной коллекции](#) или другого внешнего API. Внутри книги такая репликация приводит к созданию страницы для каждого репликанта. Другие типы блоков, поддерживающие дочернюю репликацию (такие как Slideshow, Grid и Scroller), обеспечивают другие механизмы.

Если у вас есть драйвер устройства с индексированными подсвойствами или скрипт для доступа к внешним данным, вы можете использовать дочернюю репликацию для создания соответствующего количества дочерних блоков во время воспроизведения:

- Установите для режима "Child Replication" значение "Indexed Property" (индексированное свойство).
- Укажите частичную привязку к индексированному свойству устройства или фрагменту контента. Любой фрагмент контента находится в разделе *Script, feed, <script-name>, <instance-name>*, где *<script-name>* — это имя скрипта фида и *<instance-name>* — это имя предоставленного фрагмента контента по этому сценарию.
- Укажите, сколько элементов следует пропустить с начала списка (по умолчанию — ноль). Это позволяет вам, например, показать первое изображение в ленте на видном месте, а все дополнительные изображения перечислить ниже, и в этом случае список дополнительных изображений должен пропустить первое, чтобы оно больше не отображалось.
- Ограничьте количество создаваемых дочерних блоков (по умолчанию отображаются все элементы, доступные из драйвера или канала).
- Для доступа к специфической информации канала или элемента в дочернем элементе (например, номер текущей страницы или общее количество страниц), введите произвольное имя для Replicator Score. Затем это можно использовать внутри реплицированных дочерних элементов для доступа к такой информации в *Local, score, <score-name>, <property-name>*, где *<score-name>* — указанное имя.
- Внутри реплицированного дочернего блока в разделе *Relative* появятся соответствующие свойства для привязки текста, изображений, элементов управления и т. д. Подчеркнутые относительные свойства (*Relative properties*) доступны непосредственно для использования в реплицированных дочерних блоках. Неподчеркнутые относительные свойства должны быть доступны только в обзорах деталей, например, при отображении одного объекта - не сетки и не списка объектов.
 - ◆ **ПРИМЕЧАНИЕ.** В редакторе вы увидите только один прототип дочернего блока без отображения каких-либо данных, специфичных для экземпляра, что позволяет привязывать текст, изображения, видео и т. д. к нужным свойствам. Фактическая репликация происходит только на месте, предоставляя реальные данные каждому экземпляру, как это определено источником данных.

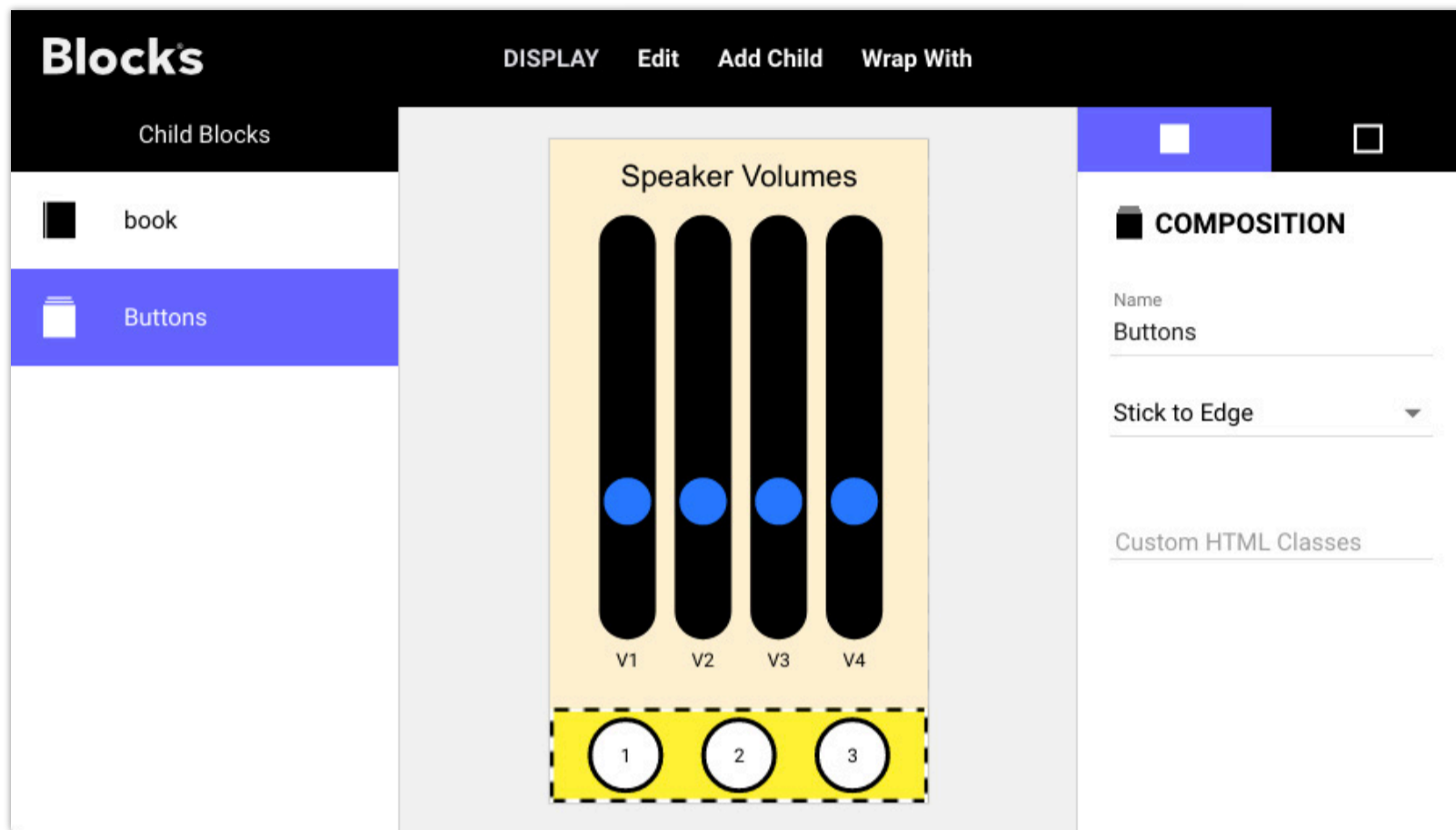
Навигация от реплицированного дочернего элемента

В то время как механизм репликации делает все соответствующие свойства доступными для дочерних блоков, иногда нужно перенести привязки таких свойств к блоку, который не является дочерним блоком репликатора. Например, у вас может быть Scroller, использующий дочернюю репликацию для перечисления ряда объектов, возможно, с эскизом изображения и заголовком для каждого. Щелкнув по этому блоку, вы хотите отобразить подробную информацию об объекте. Для этого добавьте кнопку «Перейти к локальному блоку», чтобы перейти к другому блоку, предназначенному для отображения подробной информации. Однако этот другой блок сам по себе не может быть дочерним по отношению к репликатору, поскольку он не входит в список.

Чтобы облегчить такое использование, Blocks будут передавать контекст от выбранного элемента списка к целевому блоку (обычно это Композиция), позволяя ему отображать данные из исходного элемента списка. Однако, для того чтобы спроектировать эту композицию (которая сама по себе не находится внутри репликатора), вам необходимо предоставить контекст, который в конечном итоге будет использоваться, чтобы вы могли связывать изображения, текст и другие элементы управления во время разработки. Это делается путем указания прототипа привязки [Prototypical Binding](#) внутри целевой композиции.

Stack (штабель)

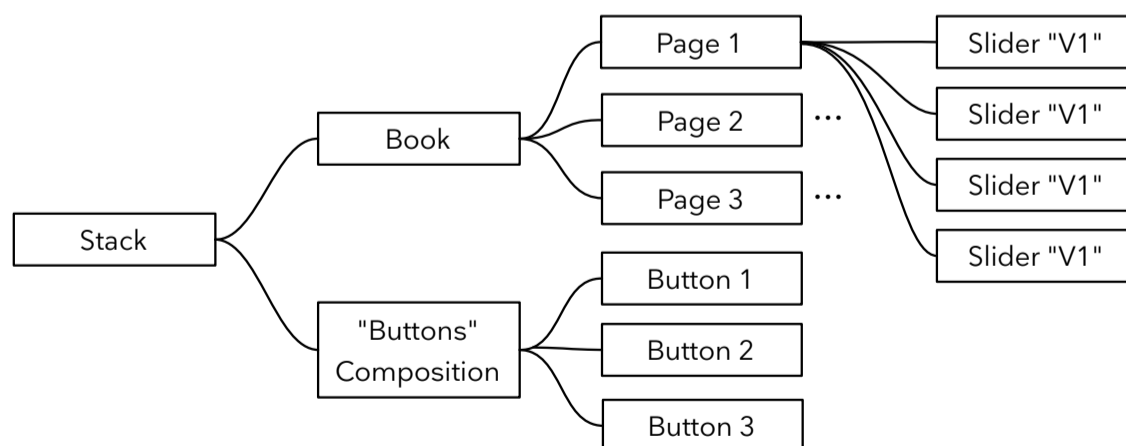
Стек отображает свои дочерние элементы в виде двух или более вертикально расположенных секций. Обычно он используется для добавления навигационных панелей в верхней и/или нижней части мобильного устройства (или аналогичного), при этом основная область занимает центр экрана.



Блок Stack с двумя дочерними блоками. Дочерний блок Book (четыре движка) занимает большую часть пространства, назначено "Fill Remainder" (заполнить остаток). Композиция "Buttons" прикреплена к нижнему краю.

- ◆ Чтобы блок стека заполнил всю высоту устройства, используемого для его отображения, оставьте поле Height (высота) блока Stack пустым. Тогда высота будет адаптирована к используемому устройству.

Дочерний блок может быть **Stick to Edge** (прикреплён к краю), как в случае с кнопками (Buttons) в примере, показанном выше. Такой дочерний блок «прилипает» к верхнему краю, если он помещен в верхней части списка, или к нижнему краю, если он расположен внизу. Блоку "Book" (это четыре ползунка в приведенном выше примере) затем назначают **Fill Remainder** (заполнить остаток).



- ◆ Можно создать аналогичную компоновку и используя композицию (Composition), однако только блок стек Stack гарантирует, что верхний/нижний блок с установкой «Прикрепить к краю» (Stick to Edge) всегда будет в верхней или нижней части экрана, независимо от размеров этого экрана, при этом основная область будет занимать оставшееся пространство. Это делает стек более адаптивным к разным размерам экрана мобильных устройств, в то время как дочерние блоки композиции занимают фиксированные положения.

Scroller (прокрутка)

Блок прокрутки размещает своих дочерние элементы горизонтально или вертикально, позволяя зрителю вручную прокручивать полученный макет в нужном направлении. Его также можно использовать с одним широким или высоким блоком, позволяя прокручивать его. Дочерние блоки могут появляться друг за другом или с указанным промежутком.

- ◆ Поскольку этот блок явно создан для ручного взаимодействия, размещенные на нем видео не будут воспроизводиться автоматически. А если вы показываете несколько видео, запуск одного автоматически останавливает любое другое уже проигрываемое видео.
- `Scroll Direction` (направление прокрутки) можно установить как Горизонтальное или Вертикальное, при этом вертикальная прокрутка упорядочивает дочерние блоки в виде списка строк
- `Gap` (зазор) добавляет промежуток с указанным количеством пикселей между дочерними элементами.
- `Position Snap` пытается расположить дочерний блок в указанной позиции (по центру или сверху/слева), когда прокрутка заканчивается. По умолчанию Выкл., чтобы не было привязки прокрутки.

Размер и масштабирование прокрутки и дочерних блоков

Вы должны указать размер, перпендикулярный направлению прокрутки. То есть вы должны задать видимую высоту горизонтального скроллера. Дочерние блоки `Scroller` обычно должны иметь размер, соответствующий указанной высоте или ширине. Сам скроллер будет масштабирован, чтобы соответствовать окружающему его блоку. В некоторых случаях, например, при использовании в качестве основной области блок `Stack` (штабель), этот размер становится динамическим, т.е. подстраивается под любое доступное пространство, которое зависит от разрешения и ориентации устройства отображения.

Вам не нужно указывать тот размер дочерних блоков, который соответствует направлению прокрутки. То есть, если направление прокрутки вертикальное, вы можете не указывать высоту. В этом случае высота будет адаптирована к контенту дочернего блока. Это полезно, если контент имеет переменный размер, например, динамический текст или URL-изображение. Укажите значение для принудительного применения определенного размера.

Child Replication (дочерняя репликация)

При использовании [child replication](#) в `Scroller` репликаны располагаются в виде прокручиваемого списка. Используйте одну композицию в качестве прототипа дочернего элемента с нужными элементами, включая элементы, привязанные к свойствам источника данных.

Grid сетка

Подобно Scroller, дочерние элементы располагаются в виде горизонтально или вертикально прокручиваемого списка. Однако Grid упорядочивает свои дочерние блоки сначала в виде столбца (для горизонтальной сетки) или строки (для вертикальной сетки), помещая как можно больше дочерних элементов, прежде чем продолжить движение в направлении прокрутки. Для отображения однородной сетки все дочерние блоки должны иметь одинаковый размер.

Child Replication (дочерняя репликация)

Блок Grid особенно полезен в сочетании с дочерней репликацией, когда у вас нет всех дочерних блоков впереди, а количество дочерних элементов определяется источником данных. В этом случае вы добавляете один дочерний элемент Composition с фиксированным размером. Внутри этого дочернего элемента добавьте [динамический текст](#), изображения (используя [Media URL](#)) и другие элементы управления, привязанные к реплицируемому источнику данных.

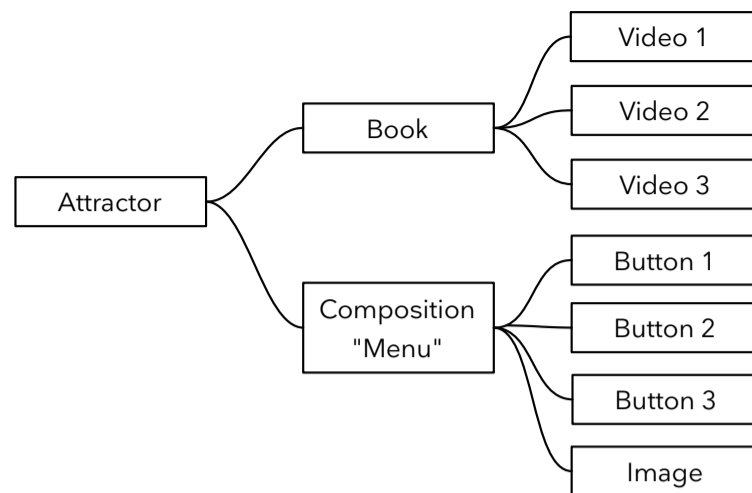


Блок Grid, показывающий изображения из базы данных коллекции.

Attractor ...

Блок Attractor имеет один активный дочерний блок, который всегда является его первым дочерним блоком, за которым следует один или несколько пассивных дочерних блоков. Это применяют для интерактивности, когда нужно показать что-то на экране, чтобы привлечь посетителей. Когда посетитель начинает взаимодействовать, дисплей переключается в активное состояние. Как только активное состояние заканчивается (например, когда видео перестает воспроизводиться), блок Attractor возвращается в пассивное состояние.

- ◆ Активное состояние самого внешнего аттрактора также отражается через активное свойство Дисплей-спота, используемого для представления блока. Эта привязка является двунаправленной, так что вы можете перевести аттрактор в активное или пассивное состояние, установив активное свойство спота. Обратите внимание, что это относится только к самому внешнему аттрактору, если вы используете несколько вложенных блоков аттрактора.



Блок *Attractor* с книгой *Book* в качестве активного дочернего блока и композицией *Меню* в качестве пассивного дочернего блока.

В приведенном выше примере показан блок *Attractor* в сочетании с другими блоками для создания интерактивного киоска. Здесь в пассивном состоянии отображается меню. Когда посетитель начинает взаимодействовать, нажав кнопку меню, чтобы начать воспроизводить видео, блок *Attractor* автоматически переключается в активное состояние, показывая одну из страниц книги (видео). Книга - первый, и следовательно, активный дочерний блок аттрактора. Как только видео заканчивается, блок *Attractor* автоматически возвращается в пассивное состояние, показывая композицию "Menu".

Вы можете указать переход ([transition](#)), который будет использован для переключения между активным и пассивным состояниями.

Спот становится активным в следующих случаях:

- Кто-то прикоснулся к нему. Применяется только при выборе "Activate by Touch/Click".
- При использовании [Локатора](#) для доступа к споту. Это позволяет посетителю инициировать активное состояние с помощью мобильного устройства.
- Если программно установить для свойства активного спота значение *true*.
- Если задать поведение *Reveal* (раскрыть), показывающее блок под активным дочерним элементом.
- При переходе к такому дочернему блоку любым другим способом, например, с помощью кнопки или внешней команды.

Спот и его Аттрактор возвращаются в пассивное состояние одним из следующих способов:

- При достижении конца воспроизведения (видео, слайд-шоу и т. д.). Применяется только при выборе "Deactivate on Play End" (деактивировать в конце воспроизведения). Здесь вы также можете указать *Linger after Play* (время задержки после воспроизведения), чтобы предоставить паузу для поиска или воспроизведения другого контента, прежде чем вернуться в пассивное состояние.
- Если программно установить для свойства активного спота значение *false*, поведение *Reveal* (раскрыть), открывающее блок в пассивном дочернем блоке, или путем перехода к такому дочернему блоку любым другим способом, например, с помощью кнопки или внешней команды.
- Через определенное время. Применяется только в том случае, если выбрано "Deactivate on Timeout", после того как спот бездействовал в течение указанного времени. Любое взаимодействие, например сенсорные жесты, может сбросить тайм-аут, сохраняя аттрактор в активном состоянии.

◆ **ПРИМЕЧАНИЕ.** Взаимодействие внутри веб-блока не может быть обнаружено и не будет автоматически распознаваться как действие. Однако, если вы контролируете страницу, отображаемую веб-блоком, туда можно добавить код, чтобы намеренно уведомлять Vlocks о любых взаимодействиях, тем самым сохраняя аттрактор в активном состоянии.

Использование нескольких пассивных дочерних блоков

Как правило, есть только один пассивный дочерний блок, как в приведенном выше примере, но таких блоков может быть сколько угодно. Если есть несколько пассивных дочерних блоков, *Attractor* будет чередовать их, и получатся некоторые вариации в цикле привлечения.

Использование вложенных аттракторов (Nested Attractors)

Иногда приходится использовать несколько аттракторов одновременно. Внутренний аттрактор может, например, использоваться для управления меню, где кнопки нужны для воспроизведения нескольких видео. Как только такое видео останавливается, Аттрактор используется для возврата на страницу меню. Этот вложенный аттрактор/меню затем становится частью более крупной структуры, возможно, вмещающей несколько таких меню. Помимо всего этого, есть еще один Аттрактор, который управляет активным/пассивным режимом всего спота, возвращая его в какое-то нейтральное состояние после тайм-аута. В этом сценарии только самый внешний аттрактор будет управляться и контролировать активное свойство спота.

Tag Selector 📌 (селектор тэгов)

Блок Tag Selector работает в сочетании с тэгами, назначенными спотам. Тэги могут быть назначены статически (как описано в разделе ["Tags"](#) главе «Споты»), или их можно назначать динамически и интерактивно при помощи кнопок и задач. Типичное использование тэгов:

- Указание на то, что спот - это мобильное устройство, статически присваивая ему тэг «мобильный» (mobile).
- Указание на относительное положение спота в группе спотов (Spot Group); например «слева» (left), «в центре» (center) и «справа» (right), что обеспечивает подачу корректного контента на соответствующий дисплей в группе.
- Указание возрастной группы или языковых предпочтений посетителей. На мобильном устройстве может отображаться страница меню, такая как показана справа, что позволяет посетителям выбирать предпочтительный язык из набора поддерживаемых языков.



Svenska

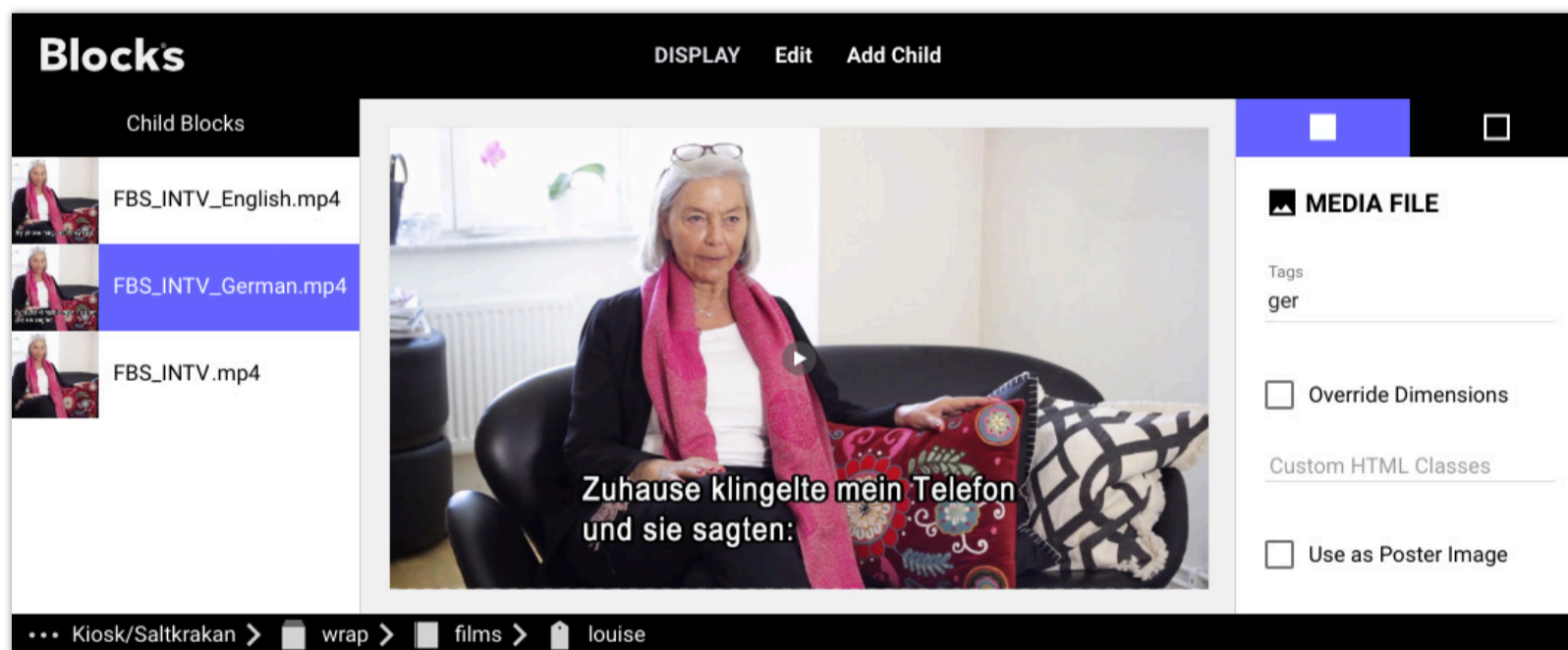


English



Deutsch

Независимо от того, как тэг применяется к споту, он может быть использован блоком выбора тэгов (Tag Selector) при выборе подходящего дочернего блока из набора блоков для отображения на этом споте.



Блок Tag Selector отображает подходящий дочерний элемент. Здесь тэгом «ger» назначено воспроизведение немецкой версии видео..

Использование нескольких тэгов

Вы можете применить любое количество тэгов к споту. Вы можете также указать любое количество тэгов для дочерних блоков селектора тэгов (Tag Selector). При отображении селектора тэгов он покажет дочерний блок с наиболее подходящими тэгами. Если тэгов нет, он покажет первый дочерний блок.

Переопределение размеров дочерних блоков

Дочерние блоки селектора тегов (а также синхронизатора) имеют параметр "Override Dimensions" (переопределить размеры). Это полезно, когда у вас есть один блок, который должен воспроизводиться на устройствах с экранами разного размера, где тег используется для выбора правильного типа экрана. Это можно использовать, чтобы различать альбомную или портретную ориентацию, а также дисплеи и мобильные устройства. В этих случаях выберите

“Override Dimensions” для дочернего блока, который будет использован с другими размерами, отличными от размеров корневого блока, затем введите нужные размеры.

Media File (медиафайл)

Блок этого типа представляет собой один мультимедийный файл. Добавьте его, либо выбрав «Media File ...» в меню Add Child, либо просто перетащив медиафайл в список дочерних блоков.

Можно использовать следующие типы файлов:

- Изображения форматов JPEG, PNG, SVG и GIF.
- Видео, кодированное кодеком H.264, как правило, в файл MP4 (см. ниже о кодировании видео и форматах).
- Звук кодируется кодеком AAC, как правило, в файл m4a или подобный.
- 3D-модель в формате GLB. Это разрешено только внутри [3D block](#).

В общем, изображения и видео должны иметь те же размеры, что и блок, к которому вы их добавляете. Например, если размеры блока составляют 1920x1080, то изображения и видео должны иметь такие же размеры. Исключением является блок композиция [Composition](#) где изображения и другие дочерние блоки, как правило, меньше самой композиции, чтобы их можно было разместить рядом или наложить друг на друга. Композиция позволяет масштабировать дочерние блоки, но все-таки лучше создавать контент с корректными размерами, это обеспечит лучшее качество и снизит нагрузку на воспроизводящее устройство, что особенно важно для видео и больших изображений. Максимальное разрешение и битрейт для видео зависят от того, какое устройство вы планируете использовать в качестве дисплей-спота (Display Spot). Подробную информацию и рекомендации смотрите в документации производителя устройства воспроизведения.

◆ Подробнее о кодировании видео и использовании других форматов файлов здесь [здесь](#).

Замена медиафайлов

Вы можете заменить файл, используемый блоком медиафайла, выбрав блок и выбрав “Replace Media” (заменить медиафайл) в меню Edit. При замене только медиафайла сохраняются все остальные настройки блока.

Поведение при воспроизведении

Звук и видео имеют следующие дополнительные настройки:

- Play Automatically запускает воспроизведение при появлении медиаблока. В противном случае воспроизведение медиа необходимо запустить вручную, нажав кнопку воспроизведения, которая появляется в центре блока.
 - ◆ В некоторых веб-браузерах автоматическое воспроизведение мультимедиа может быть ограничено. Это не относится к проигрывателю PIXILAB. [Эта статья](#) рассказывает, как убедиться, что контент может воспроизводиться без какого-либо взаимодействия с пользователем. Некоторые мобильные устройства (например, iOS) не позволяют воспроизводить медиафайлы со звуком без вмешательства пользователя.
- Loop инициирует многократное воспроизведение медиа. В противном случае медиа воспроизводится только один раз.
- Exclude from Sync (исключить из синхронизации). Иногда требуется исключить из синхронизации видео или звук, используемые в контексте синхронизатора. Это может быть, например, зацикленный фоновый звуковой клип или анимированный логотип.
- Playback Controls определяет, будет ли виден индикатор выполнения во время воспроизведения. Выберите **None**, чтобы скрыть индикатор, **Progress**, чтобы отобразить стандартный индикатор выполнения Blocks (одинаковый для всех браузеров/проигрывателей и легко настраиваемый с помощью CSS), или **Native**, чтобы отобразить индикатор выполнения браузера.

Local Sync Slave (ведомое устройство локальной синхронизации)

Этот параметр доступен только в том случае, если блок Media (или Media URL) находится внутри синхронизатора. Если выбрано, медиа будут синхронизированы только в пределах текущего спота. Это полезно, например, для синхронизации двух видео, производимых на одном споте. Если это так, поместите их в синхронизатор, а затем выберите этот параметр для одного из них. Если в этом случае не требуется синхронизировать внешний спот,

то выберите опцию "Local Synchronization".

Настройки воспроизведения видео на iPhone

Видеофайлы имеют настройку, управляющую их поведением на айфонах.

- iPhone Video Fullscreen автоматически увеличивает видео до полноэкранного размера при воспроизведении. Это стандартный (и ранее единственный) способ воспроизведения видео на iPhone.
- iPhone Video Inline заставляет видео оставаться в текущей позиции на экране и воспроизводиться там. Это позволяет другим соседним элементам оставаться видимыми, но приводит к тому, что видео остается маленьким на экране, если только пользователь явно не увеличил его до полного размера экрана (нажав кнопку масштабирования, расположенную в углу видео).
 - ◆ Этот параметр применим только к iPhone и не влияет на другие устройства отображения

Media URL

Этот блок похож на описанный выше блок Media File, но он получает контент из URL-адреса, а не файла. Поэтому отображаемые медиафайлы могут меняться в соответствии со значением привязанного свойства. Это имеет два основных варианта использования:

- Для управления тем, что отображается во время выполнения. Полезно, когда медиафайл для показа недоступен во время создания блока. Это делается привязкой URL-адреса к любому системному свойству (включая [параметр спота](#)) которое может предоставить действительный URL адрес.
- Для отображения медиа в реплицированном блоке, на которое ссылается URL-адрес, где URL-адрес происходит из [базы данных управления контентом](#).

Хотя большинство настроек идентичны описанным выше для блока Media File, этот блок имеет несколько дополнительных настроек:

- Media Type указывает, какой тип медиа (неподвижное изображение, видео, аудио) вы ожидаете получить.
- Aspect Ratio Adaptation определяет, что делать, если соотношение сторон отображаемого медиафайла не соответствует размеру блока. *Stretch* масштабирует носитель неравномерно. *Fit Inside* и *Crop* сохраняют соотношение сторон, либо за счет [letterboxing](#), либо за счет обрезки (crop) медиа.
- Image URL Binding указывает URL-адрес неподвижного изображения или, в случае видео, кадра заставки, который нужно показать до начала проигрывания видео.
- A/V URL Binding (привязка) указывает URL-адрес любого аудио- или видеофайла.
 - ◆ **ПРИМЕЧАНИЕ.** Если вы заранее не знаете, будет ли доступно изображение или видео, установите для параметра «Тип носителя» значение «Видео». Затем укажите URL-адрес аудио/видео, если вы действительно получаете видео (возможно, вместе с изображением постера в качестве URL-адреса изображения). Если вы получаете только неподвижное изображение, просто укажите URL-адрес изображения, оставив URL-адрес аудио/видео пустым, таким образом показывая только изображение без каких-либо элементов управления видео.

Camera

Блок Camera позволяет вам или посетителю сделать снимок. При этом используется либо веб-камера, подключенная к Display Spot, либо мобильная камера Visitor Spot.

- ◆ **ВАЖНО:** это премиальный тип блока, требующий дополнительных лицензионных опций. Для дисплей-спотов, не использующих наше программное обеспечение [PIXILAB Player](#) для доступа к камере требуется безопасное подключение к серверу (https). Это требование не распространяется на гостевые споты [Visitor Spots](#), где используется встроенная камера устройства. Для использования на гостевых спотах вы также должны иметь вариант лицензии на сбор данных о посетителях ([Visitor Data Collection](#)).

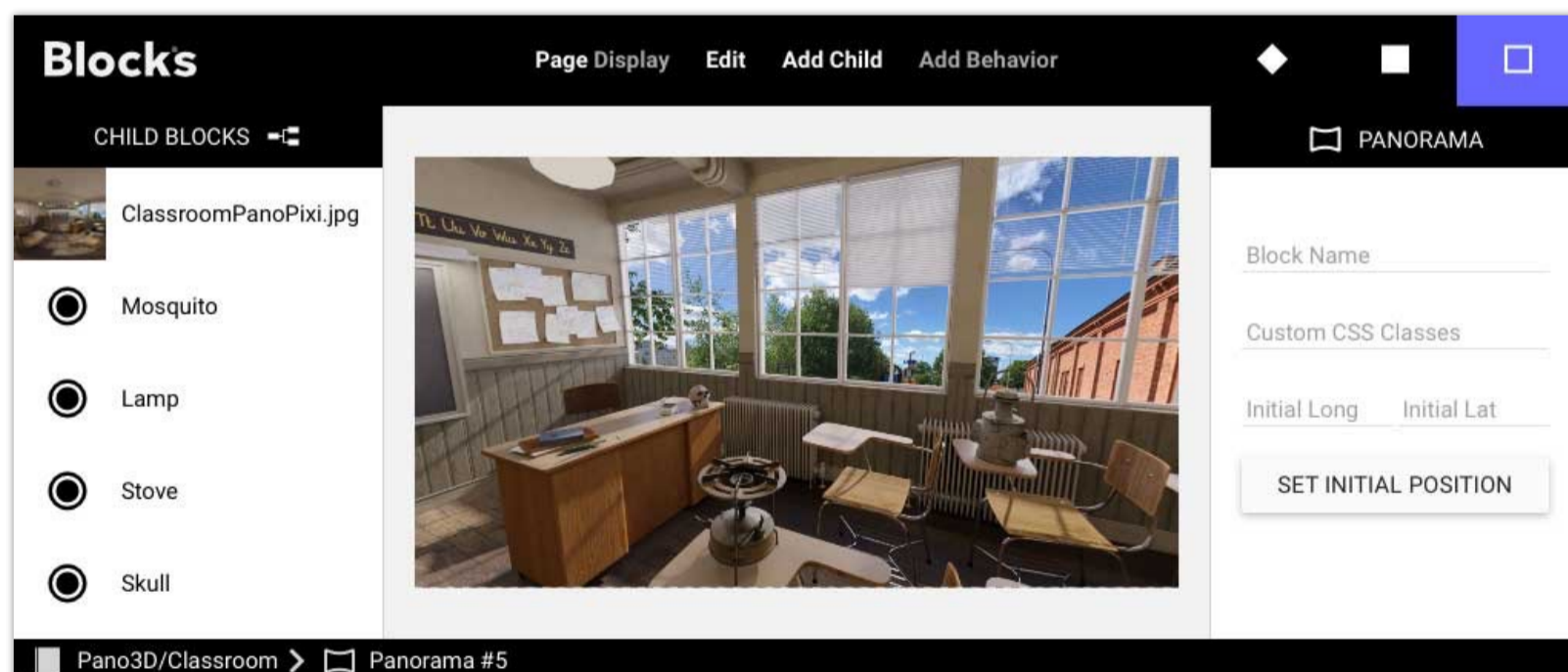
Полученное изображение затем загружается на сервер Blocks, где требуется пользовательский сценарий ([User Script](#)) для получения изображения путем подписки на событие «изображение». Затем пользовательский сценарий должен будет решить, что делать с изображением. Например, он может быть связан с отдельным посетителем в рамках сбора данных о посетителях ([Visitor Data Collection](#)).

- Start Automatically (автозапуск) включает камеру в режим видеоискателя немедленно, без вмешательства пользователя. Этот вариант применяется только для дисплей-спота, на котором запущен PIXILAB Player, и не может использоваться на гостевом споте (т. е. на мобильных устройствах). Если автозапуск не выбран, вы должны один раз нажать на символ камеры, чтобы активировать режим видеоискателя, прежде чем изображение можно будет захватить вторым нажатием.
- Mirror Camera View - зеркальный вид камеры переворачивает видеоискатель камеры по горизонтали, что часто желательно для камеры, обращенной лицом к посетителю. Эта настройка не применяется к гостевым спотам, где стандартная камера обычно имеет собственную соответствующую настройку.
- Photo Source on Mobile. Источник фото на мобильном споте задаёт, должен ли посетитель сделать новую фотографию (только камера) или предоставляет выбор: сделать новую фотографию или выбрать из существующих на устройстве (Камера или Библиотека).
- Camera Rotation. Поворот камеры позволяет установить веб-камеру боком. Многие веб-камеры предназначены для захвата широкоэкранный изображения. Если вы предпочитаете вертикальное изображение, установите камеру с поворотом на 90° и соответственно измените этот параметр, чтобы повернуть картинку предварительного просмотра и результирующее изображение.
- Camera Roll Name Имя фото пленки предоставляет дополнительное имя пользовательскому сценарию получения изображения. Это имеет смысл, если вам нужно показать блок Камера в более чем одном контексте, как средство указания контекста к сценарию.
 - ◆ **СОВЕТ:** Изображения будут захватываться программно, если объединить “Start Automatically” (автозапуск) с поведением “нажатие” ([Press behavior](#)), тогда не нужно будет нажимать символ камеры на экране.

Panorama

Блок [Panorama](#) представляет собой 360-градусную среду, например, комнату или сцену на открытом воздухе — это полезно, если вы хотите привести свою аудиторию в место, которое они не могут посетить лично. Он основан на 360-градусном сферическом изображении, которое иногда называют равноугольным изображением ([equirectangular image](#)). Такое изображение обеспечивает панорамный вид с точки расположения камеры. Чтобы посмотреть панораму, проводите пальцем, чтобы увеличить / уменьшить масштаб. Кнопки могут быть добавлены внутрь этой сферы в качестве интерактивных горячих точек.

- ◆ **ПРИМЕЧАНИЕ.** Это премиальный блок, для отображения которого требуются дополнительные варианты лицензии.



Блок Panorama с равноугольным фоновым изображением и несколькими кнопками.

Чтобы указать начальную ориентацию панорамы при первом отображении на споте, поверните ее до нужного вида, затем нажмите “Set Initial Position” (установить исходное положение). Выберите “Auto-spin while Inactive” (автовращение во время бездействия), чтобы восстановить исходную ориентацию панорамы после настраиваемого периода бездействия. Если выбрано, вы также можете включить автоматическое вращение панорамы, когда она неактивна, указав скорость вращения.

Обратите внимание, что вы можете добавить только одно равноугольное фоновое изображение, которое должно быть первым блоком в списке дочерних блоков. Это изображение должно иметь соотношение сторон 2 к 1. Равноугольное изображение можно сделать любым из следующих способов:

- Используйте [специализированную камеру](#) с панорамным объективом, склеивайте в Photoshop.
- Используйте приложение [Google Street View](#) для Android/iOS, чтобы создать его, а затем перенесите его на свой компьютер.
- Рендеринг из [3D программы](#) (если вся ваша сцена представляет собой 3D-модель).
- Создавайте/редактируйте в [Adobe Photoshop](#).

Добавление кнопок к панораме

Добавьте кнопки поверх панорамы с помощью команды в меню «Добавить дочерний элемент». Расположите кнопку, перетащив ее в нужное место, здесь указанное как широта и долгота внутри сферы. Размеры кнопки должны быть установлены в числовом виде. Используйте [пользовательский CSS](#) для изменения внешнего вида кнопок, если это необходимо. [Запрограммируйте кнопку](#), заставив её перейти к другому представлению с помощью Книги, управляя какой-либо внешней функцией или чем-то еще, что кнопка может делать в Blocks.

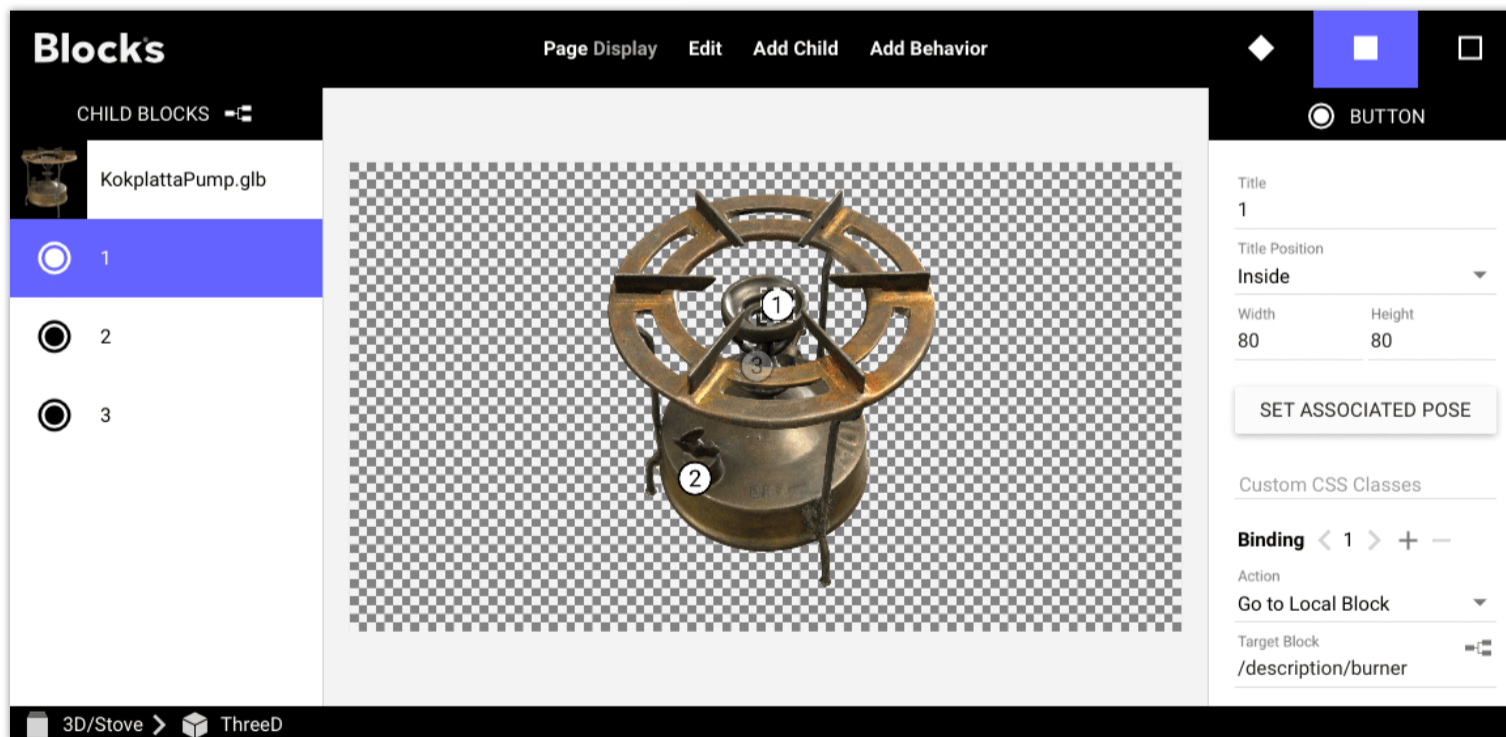
Distort with Perspective (искажение в соответствии с перспективой)

Выберите этот параметр, чтобы форма кнопки соответствовала перспективе панорамы. Это особенно полезно при использовании большой, прямоугольной и невидимой кнопки, чтобы сделать область изображения доступной для щелчка, поскольку форма области искажается, когда она находится близко к краям области просмотра.

3D

Этот тип блока [отображает 3D модель](#), которую можно вращать, перемещать и масштабировать в интерактивном режиме. Используйте его, чтобы показать отсканированный 3D-артефакт или модель, созданную в приложении 3D или САПР. Он основан на 3D-модели формата [GLB](#), этот стандартизированный формат объединяет геометрию, текстуры, анимацию и другие связанные данные в один компактный файл. Кнопки могут быть добавлены на поверхность 3D-объекта в виде интерактивных горячих точек.

◆ **ПРИМЕЧАНИЕ.** Это премиальный блок, для отображения которого требуются дополнительные варианты лицензии.



3D блок со своей моделью и несколькими кнопками.

Посетители могут перемещаться по 3D-блоку с помощью сенсорного экрана (рекомендуется) или мыши.

Навигация	Касание	Мышь
Вращение	Перемещение одним пальцем	Перемещение левой кнопкой мыши
Zoom in/out	Зажмите двумя пальцами	Колесом прокрутки
Перемещение в сторону или вверх/вниз	Перемещение двумя пальцами	Нажмите колесо прокрутки и перетащите

Настройки внешнего блока

- Illumination Factor (коэффициент освещения) управляет интенсивностью стандартного [трехточечного освещения](#) автоматически применяемого Blocks. Отрегулируйте этот параметр, если ваша модель выглядит переэкспонированной или недоэкспонированной. Установите этот коэффициент равным нулю или малым, если ваша модель имеет собственное освещение внутри 3D-файла.
- Show Shadows (показать тени) позволяет ключевому источнику света отбрасывать тени на модель. Это может замедлить визуализацию сложных моделей.
- Tilt Min/Max - эти элементы ограничивают возможность поворота модели вокруг ее горизонтальной оси. Это особенно полезно для моделей, у которых сканируется только одна сторона (обычно верхняя). Затем установите для параметра Tilt Min небольшое отрицательное значение, например -10 градусов.
- Set as Initial Pose (установить в качестве начальной позы). Укажите ориентацию и размер модели при появлении на споте, расположив её в редакторе, затем нажмите "Set as Initial Pose".
- Auto-spin while Inactive (автоматическое вращение в неактивном состоянии) автоматически возвращает модель в исходное положение после настраиваемого периода бездействия. Если выбрано, можно задать автоматическое вращение модели, когда она неактивна, указав скорость вращения.

Получение 3D моделей

Обратите внимание, что вы можете добавить только одну 3D-модель, которая должна быть первым блоком в списке дочерних блоков. Эта модель должна быть файлом GLB. Модель можно получить несколькими способами, часто - их комбинацией.

- Для физических объектов, где требуется высокая степень точности, используйте профессиональный 3D-сканер, доступный от ряда производителей (например, [этот](#)). Они часто сочетают в себе несколько методов, таких как фотография и лидар.
- [Фотограмметрия](#), получение 3D-модели из большого количества фотографий.
- Для быстрой проверки можно использовать [приложение для сканирования на телефоне](#).
- Создайте или отредактируйте модель в 3D-приложении, а затем [экспортируйте ее в файл GLB](#).
- Загрузите один из множества сайтов в Интернете, предоставляющих 3D-файлы, таких как [Sketchfab](#) или [Smithsonian](#).

Sketchfab — отличный сайт для поиска 3D-моделей. К сожалению, GLB не является одним из поддерживаемых им форматов. Однако он предоставляет близкий формат GLTF. Его можно преобразовать в GLB с помощью ряда онлайн-сервисов, а также многих 3D-программ, таких как [Blender](#). В Смитсоновском музее есть множество 3D-моделей в формате GLB. Обратите внимание, что некоторые модели очень большие и нуждаются в дополнительной обработке для лучшей производительности. Точно так же приходится обрабатывать выходные данные 3D-сканирования, чтобы уменьшить размер текстуры и сложность модели, что делает ее подходящей для интерактивного использования в Blocks. Используйте графический редактор, например Photoshop, чтобы уменьшить разрешение текстур до разумного значения, например 2048x2048. Многие 3D приложения имеют функции для уничтожения 3D-сеток. Для этого также существуют специализированные программы, такие как [Meshlab](#).

Добавление кнопок в 3D-блок

Добавьте кнопки в 3D-модель с помощью команды в меню «Добавить дочерний элемент». Расположите кнопку, перетащив ее в нужное место. Размеры кнопки должны быть заданы в числовом виде. Используйте [пользовательский CSS](#) для изменения внешнего вида кнопок, добавления значков или других символов и т. д.. [Запрограммируйте кнопку](#) для перехода в другое представление с помощью Книги, для управления какой-либо внешней функцией или чем-то еще, что кнопка может делать в Blocks.

- ◆ **СОВЕТ:** Кнопки в 3D-блоке можно размещать только на поверхности модели. Чтобы разместить кнопки вне модели, оберните 3D-блок композицией и добавьте туда кнопки.

Кнопку также можно использовать для позиционирования 3D-объекта, к которому она прикреплена:

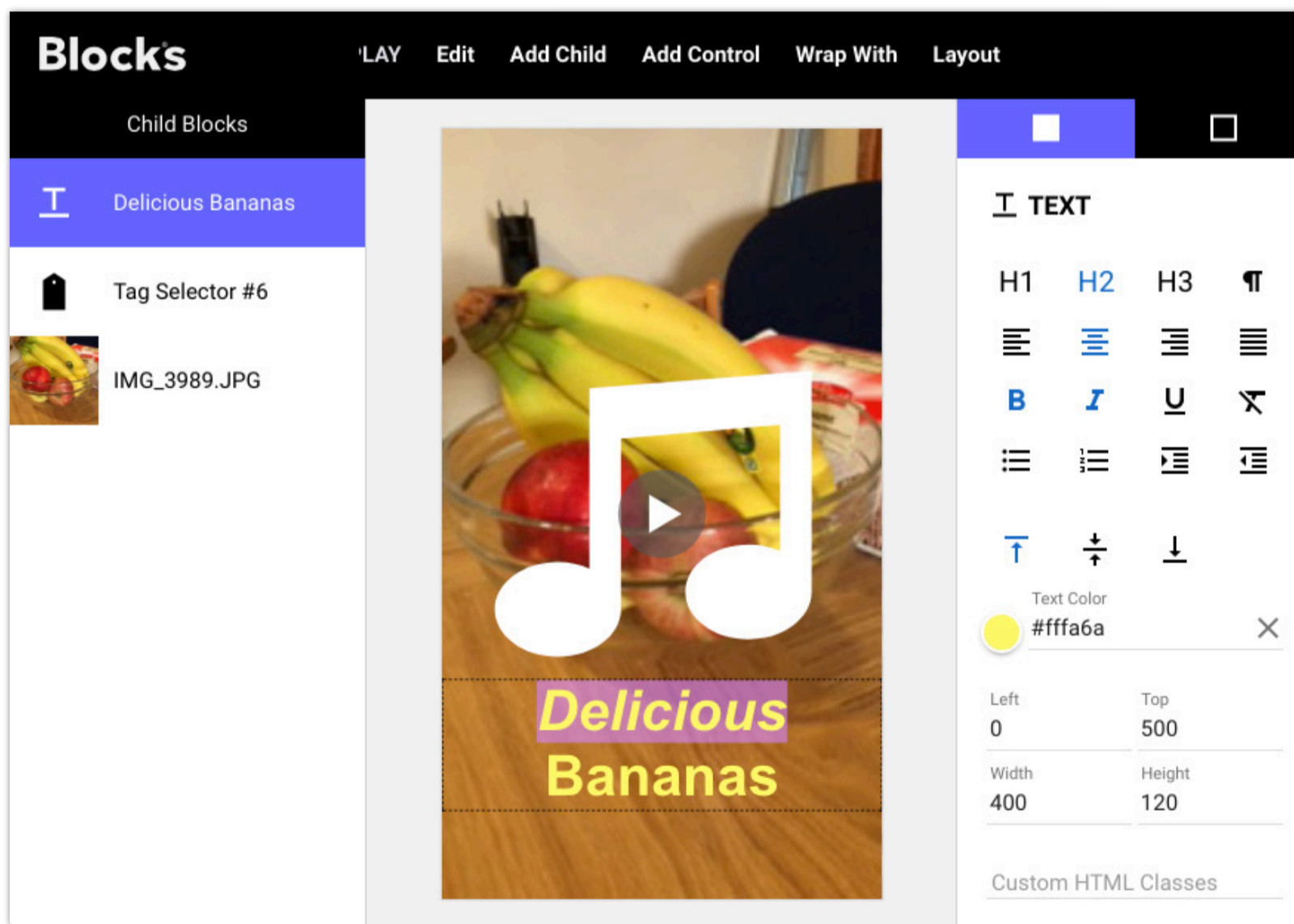
- Выберите кнопку в 3D-блоке.
- Расположите 3D-модель как задумано
- Нажмите "Set Associated Pose" (установить связанную позу).

Теперь найдите эту кнопку, используя, например, другую кнопку, нацеленную на кнопку, имеющую ассоциированную позу, для которой установлено значение "Go to Local Block" (перейти к локальному блоку), или задачу, указывающую споту, демонстрирующему 3D-модель, путь к кнопке.

- ◆ **СОВЕТ:** Если вы хотите использовать кнопку только для позиционирования, вы можете скрыть ее или изменить ее внешний вид с помощью CSS.

Text I (текст)

Используйте текстовый блок для добавления текста в презентацию.



Текстовый блок с выделенным текстом «Вкусно».

Расположите и измените размер текстового блока внутри композиции с помощью мыши или численно. Дважды щелкните текстовый блок, чтобы отредактировать текст. Используйте кнопки форматирования справа, чтобы изменить размер и положение, а также другие атрибуты текста. Кнопки H1, H2, H3 изменяют размер текста на три предопределенных размера заголовка. Кнопка «Абзац» (paragraph) возвращает размер к обычному размеру текста. Второй ряд кнопок управляет горизонтальным выравниванием выделенного текста. В третьей строке применяются варианты, такие как полужирный или курсив (при этом последняя кнопка удаляет любые такие переопределения формата).

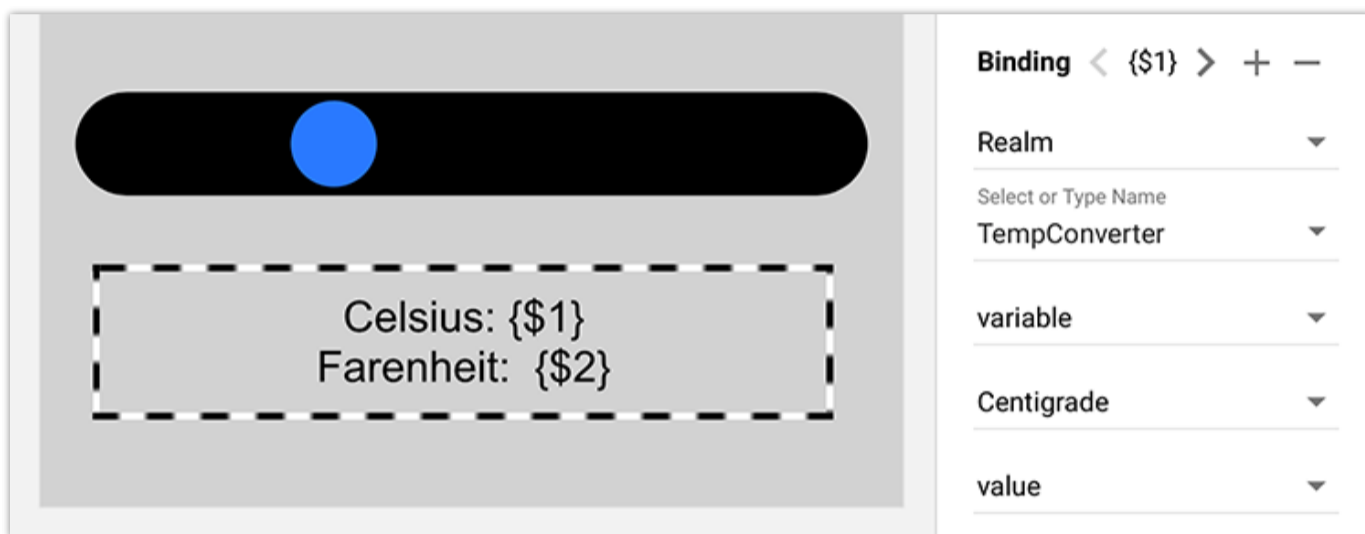
- ◆ Другие шрифты, размеры или варианты могут быть добавлены с помощью [пользовательского CSS](#).

Четвертый ряд кнопок можно использовать для добавления маркеров или номеров к абзацам, что часто полезно при отображении списков, а также для применения отступов. Последняя строка управляет вертикальным выравниванием текста внутри текстового блока.

- ◆ Текстовый блок не может быть добавлен как прямой дочерний элемент. Text не отображается в меню Add Child, и чтобы использовать текстовый блок, сначала добавьте композицию, а затем добавьте текстовый блок внутрь этой композиции.

Динамический текст

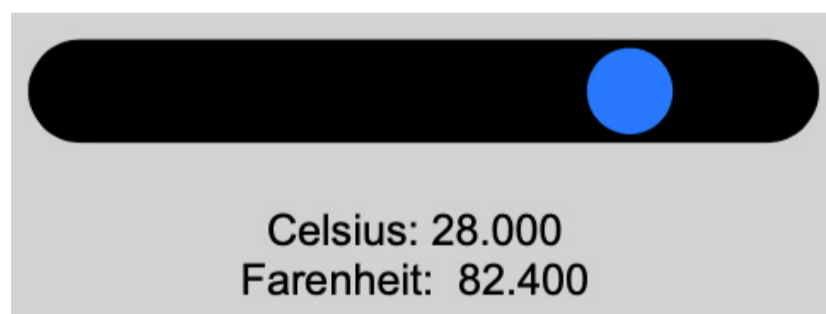
Кроме описанного выше статического текста, блок Text может отображать динамический текст, происходящий из некоторых свойств. Это похоже на работу индикаторов (Indicators and Bars), но показывает значение в виде текста, а не графически. После добавления свойства в текстовый блок, на свойство можно ссылаться внутри текста, используя специальный синтаксис, состоящий из пары фигурных скобок, содержащих знак доллара и однозначного номера индекса свойства (1-based property index number). Текст, показанный ниже, включает в себя два таких динамических свойства.



Первое свойство связано со значением переменной области с именем Centigrade, вставленной в текст как {\$ 1}. Аналогично, вторая привязана к другой переменной области с именем Fahrenheit и вставляется как {\$ 2}. Ползунок над текстом привязан к той же переменной области Цельсия. Задача запускается при изменении переменной Цельсия, устанавливая переменную Фаренгейта на основе переменной Цельсия, например:

```
1 set realm variable Fahrenheit .value to (Centigrade.value * 9/5) + 32
```

Отображая на устройстве с сенсорным экраном блок, содержащий вышеупомянутое, вы можете изменять значения с помощью ползунка, показывая температуру в градусах по Цельсию и Фаренгейту.



Форматирование динамического текста

В дополнение к порядковому номеру свойства некоторые типы данных также допускают дополнительные параметры форматирования. Эти параметры отделены двоеточием от индекса свойства (а также друг от друга, если используется более одного). Каждый форматор состоит из имени, знака равенства и значения. Применимые средства форматирования зависят от типа представляемого значения.

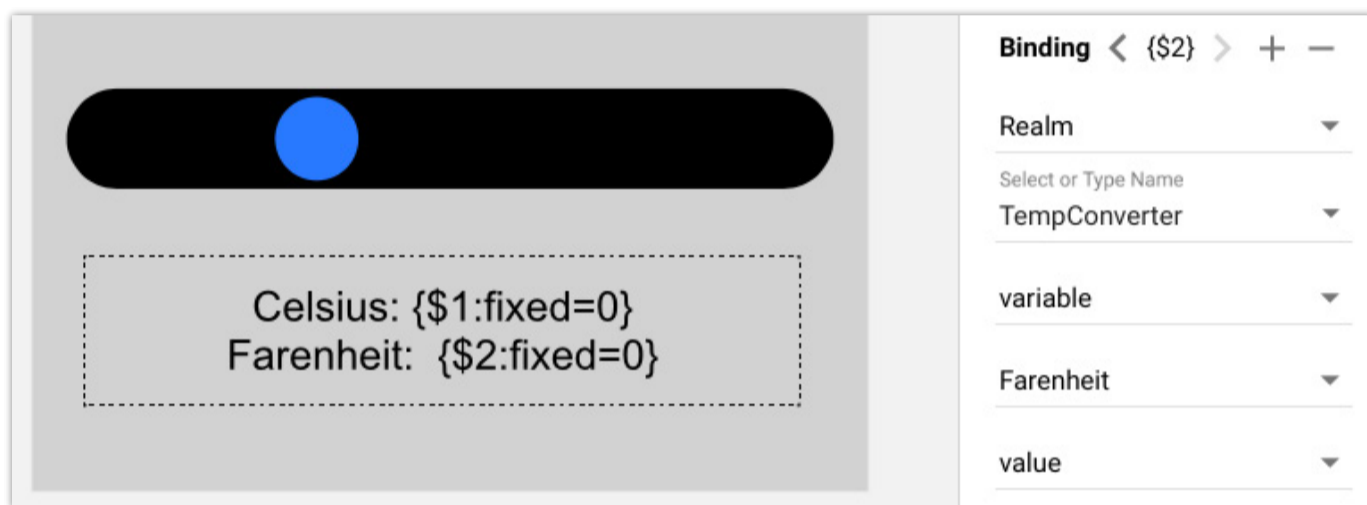
Name	Applies to	Value
true	boolean	Текст для отображения истинного значения

false	boolean	Текст, отображаемый для ложного значения.
scale	number	Коэффициент для умножения на
fixed	number	Количество десятичных цифр для отображения.
time	time	Комбинация символов hmsf, указывающая, какие компоненты времени следует включать.
locale	number	Применяет форматирование, специфичное для конкретной страны, на основе параметра, например, en-GB для английского стиля.

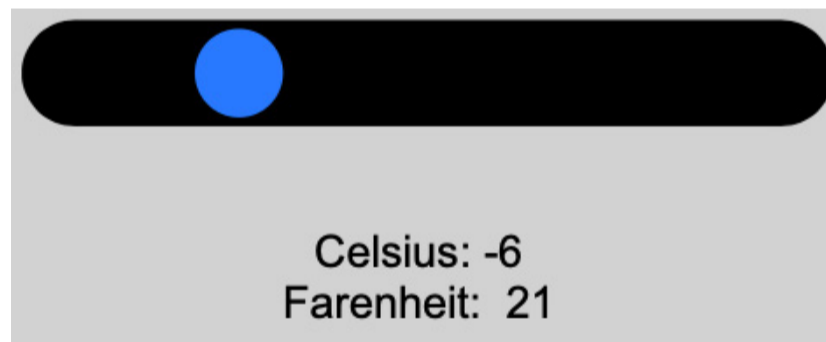
Логическое значение по умолчанию отображается как *true* или *false*. Если вместо этого вы хотите, чтобы оно отображалось как YES или NO, укажите это свойство так:

```
{ $1 : true=YES : false=NO }
```

Число может быть отформатировано с использованием масштабного коэффициента (например, чтобы показать значение 0... 1 в виде процентов 0... 100) или с ограничением числа десятичных знаков. В примере, показанном выше, можно убрать десятичные знаки в значениях температур таким образом:



Здесь число дробных цифр установлено равным нулю, и мы видим округлённые до целых значения температуры:



Значения времени, такие как временная позиция для видео или временная шкала WATCHOUT, могут быть отформатированы с использованием форматтера *show*, за которым следует нужная комбинация hmsf, представляющая часы, минуты, секунды, доли секунд. Например, следующая спецификация форматирования покажет минуты и секунды:

```
{ $1 : show=ms }
```

Locator # (локатор)

Блок Локатор отличается от других типов блоков. Он не отображает собственного контента и не имеет дочерних блоков. При его появлении всё, что вы увидите, это (дополнительная) цифровая клавиатура. Цель этого блока — определить, где находится посетитель, чтобы представить контент, связанный с этим спотом.

ПРИМЕЧАНИЕ. Это премиальный тип блока, для отображения которого требуются дополнительные варианты лицензии.

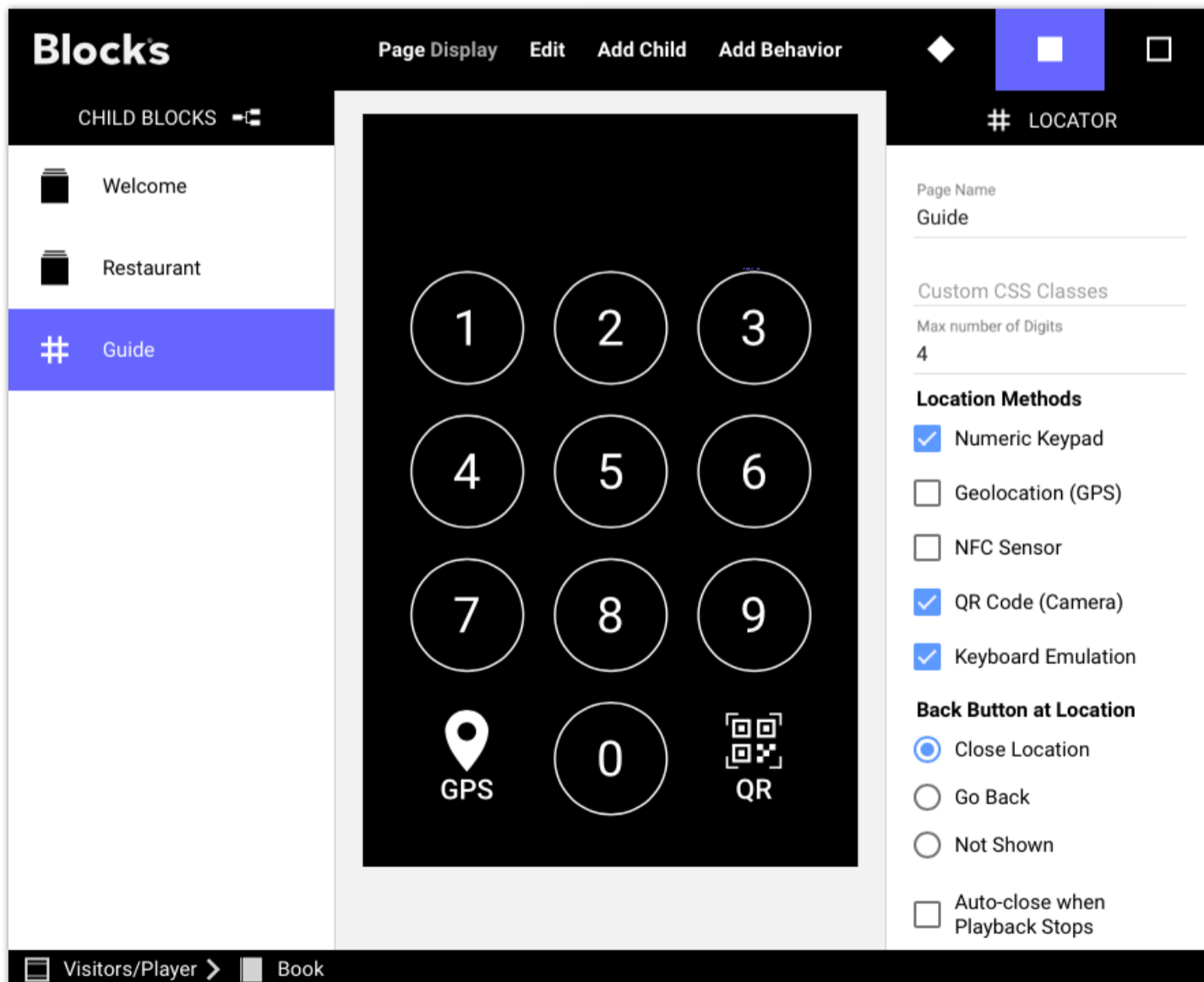
◆ Местонахождение посетителя может быть получено одним из следующих способов:

- Путем ввода номера на цифровой клавиатуре. Номер соответствует [Location ID](#) (ID местоположения спота).
- По GPS. Это положение сопоставляется с [зонами геолокации](#), указанными для спотов, и может быть использовано только на открытом воздухе устройствами со встроенным приемником GPS, такими как большинство мобильных телефонов.
- Сканированием QR-кода, найденного на споте. Это можно сделать с помощью камеры мобильного устройства. Можно также использовать специализированное оборудование для сканирования RFID/QR, эмулирующее клавиатуру (например, линейку сканеров [Zebra](#) на базе Android).
- Вручную нажатием кнопки на экране, чтобы указать свое положение. Кнопка может отображать изображение объекта, что позволяет выбрать его из меню объектов.

ПРИМЕЧАНИЕ: Параметры GPS и QR-кода на основе камеры можно использовать только при доступе к Blocks через https-соединение. Это расширенный вариант конфигурации сервера, для которого требуется

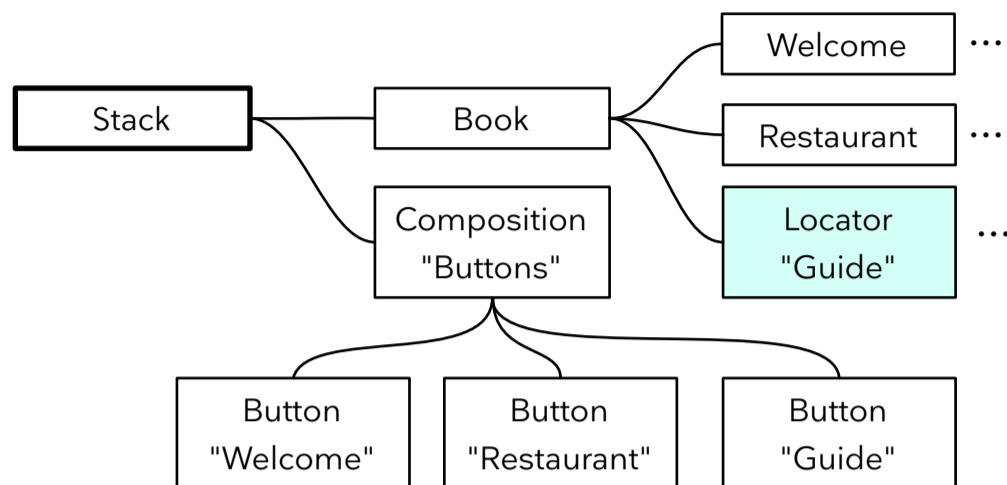
- ◆ действующий сертификат https.

В зависимости местонахождения спота конкретного типа контент может быть разным: текстом или изображением, интерактивной многостраничной презентацией, видео или звуком. Часто вы будете использовать только звуковой клип, возможно, сопровождаемый изображением, подтверждающим, что вы находитесь на правильном пути. Это можно использовать для создания аудиогuida, воспроизведения звука, локализованного в том месте, где вы находитесь, или для описания определенного объекта.



Блок Локатор с цифровой клавиатурой и включенными режимами GPS и QR.

Блок Locator можно использовать внутри других блоков для интерактивности. На рисунке (см. ниже) блок локатор находится внутри книги (Book), которая, в свою очередь, находится в блоке штабель (Stack). Кнопки, показанные внизу блока Stack, обеспечивают общую навигацию (как в примере, показанном в 40 40 разделе Stack этой главы). Страницы книги предоставляют разнообразный контент, в том числе композицию с кнопками для выбора языка и веб-блок со спецпредложениями в местном ресторане.



Локатор использован как часть интерактивного многостраничного мобильного гида.

На странице гида (Guide page) в приведенном выше примере отображается клавиатура Локатора. При вводе номера или использовании GPS/QR для определения местоположения вашего посетителя определяется местонахождение соответствующего спота и его блок появится на мобильном устройстве. Если это спот [Location](#) (местоположение), вы получите информацию (включая изображения, звук и видео), связанную с объектом в этом месте. Если это спот [Display](#) или WATCHOUT-спот, вы услышите синхронизированный с видео звук с повествованием на выбранном вами языке.

Когда контент, назначенный споту, заканчивается, цифровая клавиатура появляется снова. Кнопка накладывается на контент и во время воспроизведения, что позволяет остановить воспроизведение в любой момент.

Настройки Локатора

Локатор имеет ряд опций, позволяющих адаптировать его к особым требованиям.

Методы определения местоположения

Контролирует, какие методы можно использовать для определения местоположения вашего посетителя.

- Numeric Keypad - цифровая клавиатура показывает клавиатуру, позволяющую посетителю указать свое местоположение путем ввода номера, соответствующего каждому месту (увидев номер на экране или распечатанным).
- Geolocation (GPS) показывает кнопку GPS. При нажатии для определения местоположения вашего посетителя используется GPS телефона. Положение GPS сопоставляется с информацией о [геолокации](#) спотов.
- NFC Sensor (датчик NFC) показывает кнопку NFC. Нажмите эту кнопку, чтобы активировать считыватель NFC, который затем можно использовать для чтения URL-адреса, содержащего путь спота или ID местоположения как параметр запроса к споту. На Android большинство браузеров изначально поддерживают чтение тегов после нажатия кнопки NFC и предоставления разрешения. На iOS операционная система перехватывает NFC и перенаправляет в Safari.
- QR Code (Camera) показывает кнопку QR-кода. При нажатии камера телефона может быть использована для сканирования QR-кода, отображаемого на дисплее (с использованием блока [QR Code](#)) или на печатной табличке в каждом соответствующем месте.
- Keyboard Emulation. Эмуляция клавиатуры позволяет ввести ID местоположения с помощью сканера RFID/QR-кода, имитирующего клавиатуру, путем ввода кода и нажатия клавиши Enter. Для этого также необходимо, чтобы в настройках спота был выбран вход [вход сканера RFID/QR](#).

- ◆ **ВАЖНО:** При использовании на гостевом споте методы геолокации, NFC и QR-кода требуют, чтобы доступ к серверу Blocks осуществлялся с использованием безопасного URL-адреса https. Кроме того, эти действия требуют взаимодействия с пользователем и подтверждения. Вместе с цифровой клавиатурой можно использовать не более двух методов: геолокацию, NFC и QR-код.

Если цифровая клавиатура не включена, она отображаться не будет. Это позволяет видеть все, что находится за Локатором в композиции, и Локатор служит лишь средством отображения блока, назначенного определенному месту. Это полезно, когда местоположение устанавливается другими способами:

- С помощью сканирования RFID/NFC/QR.
- С помощью кнопки навигации, где кнопка затем устанавливается на "[Locate Spot](#)" (найти спот).
- Программно, когда используется задача или сценарий.

Кнопка Back (назад) в локации

Управляет поведением кнопки, используемой для возврата из отображения блока в найденном споте:

- Close Location. Немедленно закрывает локацию, возвращаясь к клавиатуре или меню.
- Go Back. Возврат к предыдущему виду, что в большинстве случаев эквивалентно закрытию локации. Полезно, если блок местоположения обеспечивает дальнейшую поднавигацию (например, с помощью кнопок), и в этом случае этот параметр работает как кнопка «Назад» в браузере, переходя назад на одну страницу за раз, а не сразу закрывая местоположение.
- Not Shown (**не показано**). Полностью скрывает кнопку back (назад). Полезно, когда окончание игры контролируется другими средствами, такими как задача или сценарий, или когда для закрытия локации вы хотите использовать обычную кнопку (с установленным для нее действием «Назад»).

Auto-close when Playback Stops (автоматическое закрытие при остановке воспроизведения)

Этот параметр определяет, будет ли локатор возвращаться к цифровой клавиатуре по окончании воспроизведения (например, звука или видео). Как правило, это то, что вы хотите. Однако, если обнаруженные цели содержат несколько отдельных воспроизводимых элементов, вы можете предпочесть остаться на цели, чтобы разрешить воспроизведение другого контента. В этом случае вам придется вручную вернуться к клавиатуре с помощью кнопки «Назад», показанной на целевой странице.

Geolocation Performance (производительность геолокации)

Контролирует точность GPS-позиционирования в зависимости от скорости. Применяется только при включенной геолокации. Выбирайте «Самая точная» только тогда, когда вам нужно, чтобы зоны были как можно меньше или располагались в непосредственной близости. Другие варианты обеспечивают более быструю работу, при этом «Сбалансированный» обычно обеспечивает разумный компромисс между точностью и требуемым временем.

- ◆ **ПРИМЕЧАНИЕ.** Точность определения местоположения GPS зависит от марки и модели телефона. Обязательно протестируйте свою систему на нескольких репрезентативных моделях, убедившись в достаточной точности и скорости.

Optional Binding (дополнительная привязка)

Если указано, свойство будет установлено при обнаружении спота. Аналогичным образом, если свойство изменено другими способами, локатор выполнит соответствующее размещение. Значение, которое устанавливается или на которое воздействуют, зависит от типа значения связанного свойства:

- Boolean. Логическое значение будет установлено как true при обнаружении чего-либо и как false, когда локатор находится в состоянии ожидания обнаружения чего-либо. Установка свойства другими способами не имеет никакого эффекта.
- Number (номер) будет установлен на [ID местоположения](#) обнаруженного спота (если используется ID местоположения) или на -1, если не используется ID местоположения (например, при поиске места по имени). При установке другими способами Локатор попытается найти место с этим ID местоположения.
- String. В качестве строки будет указано имя обнаруженного спота (или путь к нему, разделенный точками, если он находится внутри группы). При установке другим способом локатор попытается найти спот с указанным свойством по имени.

Использование блока и для Дисплей-спота, и для гостевого спота

Контент, предназначенный как для дисплей-спотов, так и для гостевых спотов (с помощью локатора), должен быть разработан с учетом обоих этих размеров экрана. Селектор тегов (или синхронизатор) часто используется в качестве самого верхнего блока. Первый дочерний блок должен появляться на дисплей-споте. Последующие дочерние блоки помечаются тегом «мобильный», благодаря чему они появляются в гостевых спотах (тег «мобильный» добавляется ко всем гостевым спотам по умолчанию). Подробнее см. в разделе [Synchronizer](#) (синхронизатор).

Synchronizer ▶◀ (синхронизатор)

Блок Synchronizer синхронизирует контент: видео, звук или субтитры, внутри или между спотами. Варианты использования такие:

- Гостевой спот (Visitor) воспроизводит звук синхронно с спотом Display или WATCHOUT, например, используя Локатор для временного соединения спотов в целях синхронизации.
- Два или более дисплей-спота, воспроизводящие синхронизированное видео или другой контент (например, субтитры). Эти дисплей-споты должны находиться в одной и той же группе спотов и должны воспроизводить один и тот же блок (подробнее см. ниже в разделе «Синхронизация видео на нескольких дисплеях»).
- Воспроизведение субтитров, синхронизированных с видео или звуком, на том же или другом споте (например, звуковое сопровождение, отображаемое в виде текста на телефонах посетителей для слабослышащих).

Синхронизатор имеет следующие настройки:

- Loop Content - заставляет медиафайлы внутри синхронизатора воспроизводиться циклом. Используйте этот флажок, а не флажки «Цикл» в каждом медиа-блоке для правильного зацикливания в синхронизаторе.
- Local Synchronization. Локальную синхронизацию следует выбирать, если вы собираетесь синхронизировать контент только в пределах одного спота, а не между отдельными спотами. Это часто используется с блоком субтитров, но также может быть использовано для синхронизации нескольких видео- или аудиофайлов на одном споте.

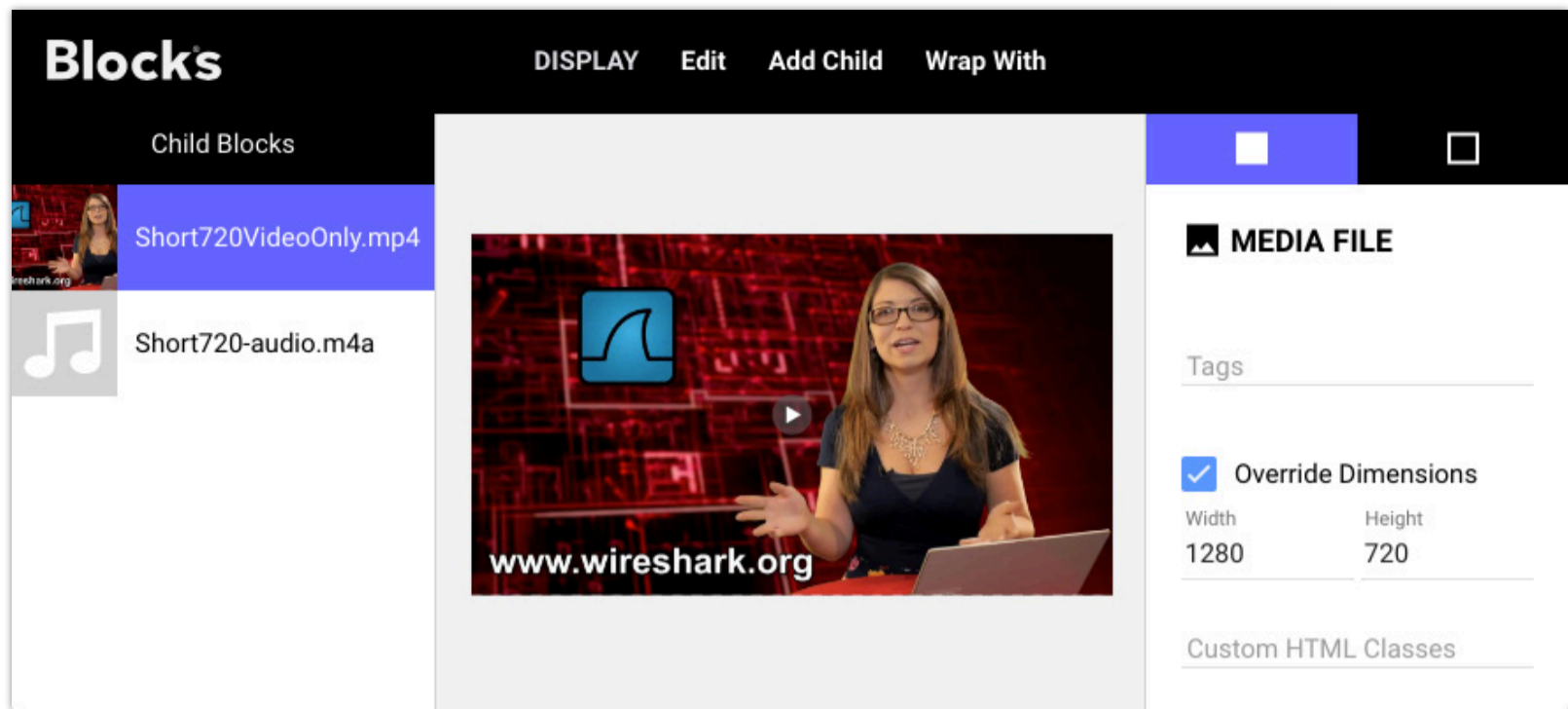
Этот блок имеет такие же возможности сопоставления тегов, что и селектор тегов, часто используемый для выбора звуковой дорожки на основе языковых предпочтений пользователя.

◆ **ПРИМЕЧАНИЕ.** Синхронизатор — это блок премиум-класса, для отображения которого требуются дополнительные варианты лицензии.

Чтобы воспроизвести звук на споте Visitor синхронно с дисплеем или спотом WATCHOUT, сделайте следующее:

1. Создайте блок Synchronizer, задав ему размеры, соответствующие мобильному устройству, как показано в примере справа.
2. Перетащите блок Synchronizer на спот Display или WATCHOUT и присвойте споту [Location ID](#).
3. Добавьте видео, которое будет отображаться на дисплей-споте в качестве первого дочернего блока синхронизатора, как показано на рисунке ниже. Переопределите размеры этого первого дочернего элемента, чтобы он соответствовал дисплей-споту (этот шаг не применяется в случае спота WATCHOUT).
4. Добавьте один или несколько аудиоблоков в качестве дочерних блоков синхронизатора, назначив соответствующие теги (как показано на второй иллюстрации, см. ниже).
5. Свяжите локатор с гостевым спотом.
6. Используйте цифровую клавиатуру на мобильном устройстве, чтобы ввести ID дисплей-спота.

Звук будет воспроизводиться через мобильное устройство синхронно с видео. Чтобы визуальный контент появлялся на мобильном устройстве вместе со звуком, сначала добавьте композицию, а затем добавьте в нее звук и другой контент.



Добавьте видео в качестве первого дочернего элемента Синхронизатора при использовании с дисплей-спотом. Переопределите размеры этого дочернего блока, чтобы он соответствовал целевому дисплей-споту



Добавьте один или несколько аудиоблоков в синхронизатор, назначив теги - мобильный тег, а также любые языковые теги, если это необходимо.

Синхронизация видео на нескольких дисплеях

Вы также можете использовать синхронизатор для синхронизации воспроизведения видео на нескольких дисплеях. Эти ролики должны иметь одинаковую продолжительность и могут быть небольшими фрагментами видео высокого разрешения, обрезанными в соответствии с расположением дисплеев. Синхронизируемые дисплей-споты должны находиться в одной и той же группе спотов. Вы можете назначить один из них "[Preferred Synchronization Master](#)" (предпочтительным мастером синхронизации), установив флажок на вкладке "Advanced" настроек дисплей-спота.

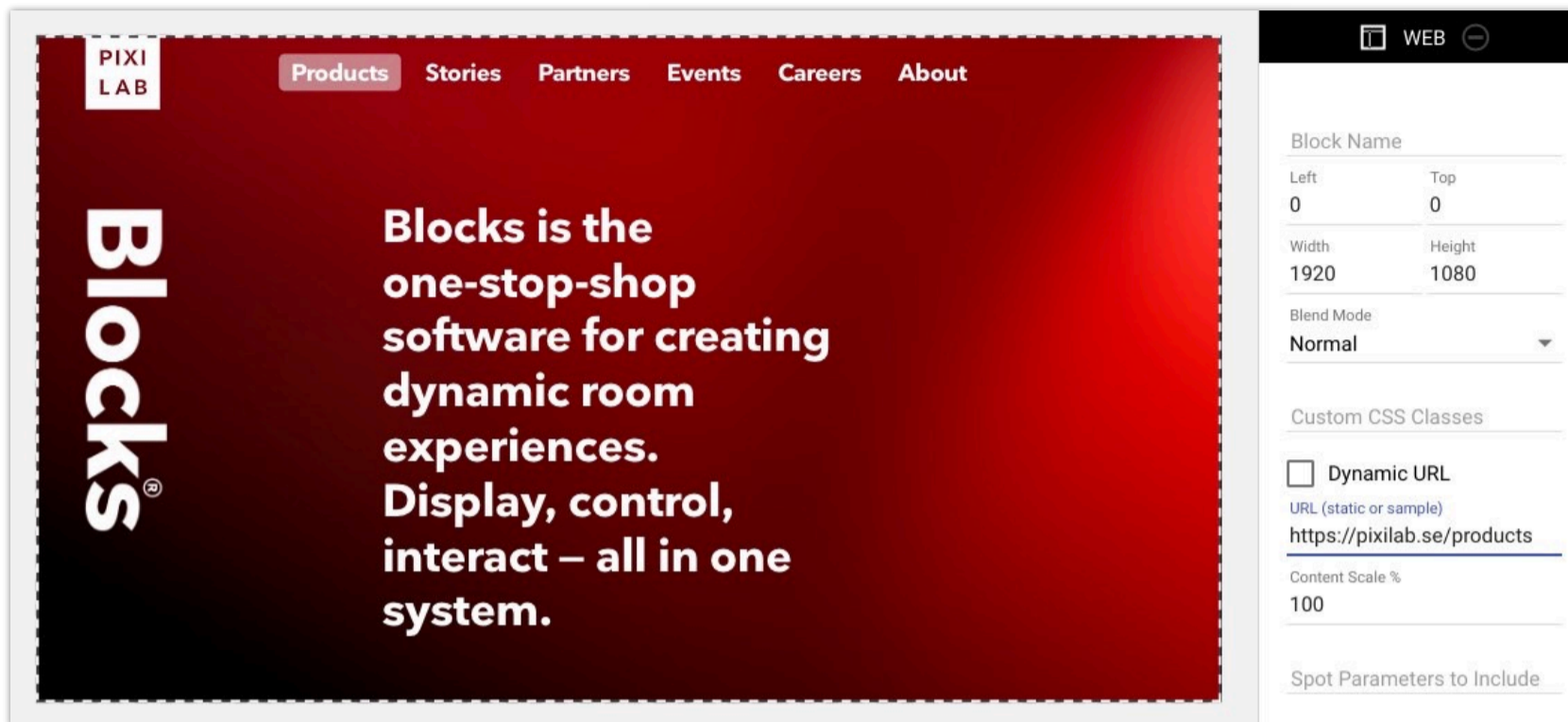
Используйте теги, чтобы связать видео с дисплеем, назначив один и тот же тег дисплей-споту и соответствующему дочернему блоку синхронизатора. Для трех дисплей-спотов, установленных рядом, вы можете использовать теги left, center, right. (лева, по центру, справа).

Обратите внимание, что не нужно добавлять блок селектора тегов, чтобы это работало, так как синхронизатор также имеет ту же возможность выбора тегов.

- ◆ Достижимая точность зависит от типа используемых устройств воспроизведения, а также от продолжительности видео. Наилучшая точность достигается при использовании PIXILAB Player для стационарных дисплеев и устройств Android для мобильных устройств.

Web блок

Добавьте веб-блок для отображения веб-сайта. Это может быть блок верхнего уровня или дочерний блок внутри композиции, книги, слайд-шоу или другого контейнерного блока.



- ◆ Имейте в виду, что загрузка веб-сайтов может занять некоторое время. При использовании в слайд-шоу веб-блок начнет загружаться при отображении предыдущего блока. Убедитесь, что предыдущий блок остается на экране достаточно долго для загрузки следующего веб-блока, прежде чем показывать его.

Настройки

- **Dynamic URL** Динамический URL-адрес позволяет указать URL-адрес, привязав его к системному свойству или [параметру спота](#), что позволяет управлять им динамически, а не показывать всегда один и тот же веб-сайт.
- **URL** указывает адрес веб-страницы для отображения. Как правило, вы можете скопировать и вставить это из поля URL-адреса в браузере, чтобы та же страница отображалась в веб-блоке (см. «Ограничения» для некоторых случаев, когда это может не работать). При использовании динамического URL-адреса этот параметр управляет URL-адресом, отображаемым в редакторе, в то время как URL-адрес времени показа будет браться из связанного свойства.
- **Content Scale** (масштаб контента) масштабирует веб-страницу на указанный процент, который иногда требуется для правильного отображения веб-страницы или с желаемым размером шрифта. Отрегулируйте по мере необходимости.
- **Spot Parameters to Include** (параметры спота для включения) необязательно указывают количество разделенных запятыми [параметров спота](#), которые должны быть добавлены к URL-адресу в качестве [параметров запроса](#). Иногда это полезно для веб-блоков, созданных специально для использования в Blocks, что позволяет передавать параметры на целевую веб-страницу.
 - ◆ **ПРИМЕЧАНИЕ.** URL-адрес относительно блока, начинающийся с "~/", (например, ~/custom.html), может использоваться, если веб-блок хранится в папке блока на сервере. Это удобный настраиваемый веб-контент, разработанный специально для использования в конкретном блоке, поскольку он станет частью этого блока при [экспорте/импорте](#). Это похоже на то, как можно использовать файл [block-relative CSS](#).

Сайты с ограниченным доступом

Некоторые веб-сайты несовместимы с веб-блоком. Например, если веб-сайт требует авторизации для работы, он не будет автоматически отображаться в веб-блоке. В этом случае уточните у владельца веб-сайта, поддерживает ли он другие средства авторизации, например токен, передаваемый как часть URL-адреса. Веб-сайт может указать, что его не следует встраивать, передав "X-Frame-Option". Вы можете обойти это с помощью расширений браузера, таких как [xframe-assassin](#) или [iframe-allow](#) для Chrome.

Наконец, если вы по какой-то причине используете защищенное (https) соединение, веб-сайты на простом HTTP-протоколе не могут быть показаны — отобразятся только сайты, которые также используют HTTPS. Запуск Blocks через безопасное соединение требует наличия сертификата и конкретной конфигурации сервера, которая выходит за рамки этого документа. Свяжитесь с PIXILAB для получения дополнительной информации, если вам это необходимо.

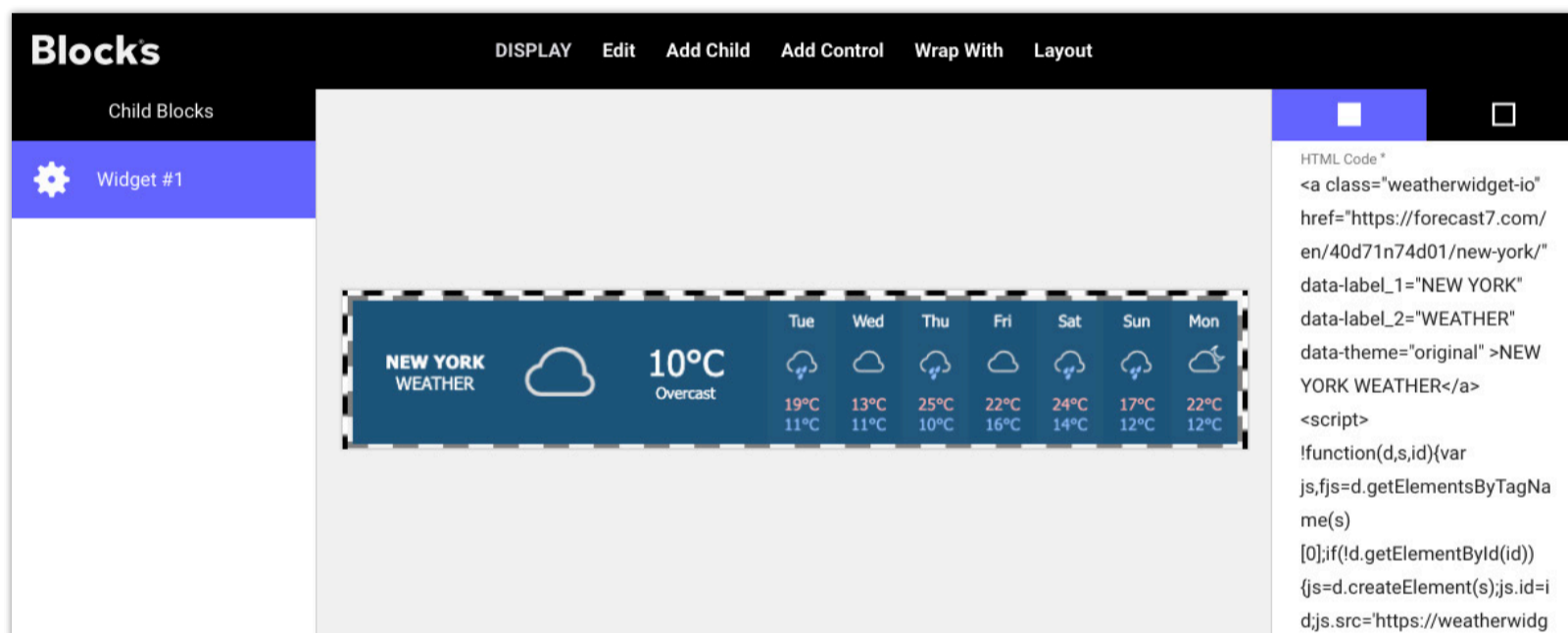
Отображение PDF-файлов

Большинство веб-браузеров имеют некоторые возможности рендеринга PDF. Следовательно, вы можете использовать веб-блок для просмотра PDF-файла, введя URL-адрес этого PDF-файла. В зависимости от того, на какую среду воспроизведения вы ориентируетесь, вы можете добавить [дополнительные параметры](#) к URL. Например, чтобы скрыть панели инструментов и увеличить коэффициент масштабирования (масштаб), введите такой URL-адрес (при условии, что PIXILAB Player или другая среда на основе Chrome:

```
http://pixi.guide/public/pdf/broschure.pdf#toolbar=0&zoom=200
```

Widget (виджет)

Этот тип блока похож на веб-блок, описанный выше. Однако вместо URL-адреса, ссылающегося на веб-контент, блок виджета размещает контент внутри себя. Этот метод обычно используется для отображения динамического контента, такого как карты погоды или часы.



```
HTML Code *
<a class="weatherwidget-io"
href="https://forecast7.com/
en/40d71n74d01/new-york/"
data-label_1="NEW YORK"
data-label_2="WEATHER"
data-theme="original" >NEW
YORK WEATHER</a>
<script>
!function(d,s,id){var
js,fjs=d.getElementsByTagName(s)
[0];if(!d.getElementById(id))
{js=d.createElement(s);js.id=i
djs.src="https://weatherwidg
```

Показанный выше виджет был получен со следующего веб-сайта:

<https://weatherwidget.io>

Код виджета, полученный с этого сайта, затем был вставлен в поле «HTML-код», как показано выше. Об использовании виджетов читайте на Blocks wiki на странице https://pixilab.se/docs/blocks/signage#using_widgets.

Reference (ссылка)

Используйте блок Reference, чтобы ввести весь блок верхнего уровня в качестве дочернего элемента другого блока. Это особенно полезно для контента, который появится в нескольких местах. Вместо того чтобы копировать один и тот же контент или блок во все эти места, сделайте следующее:

1. Добавьте общий контент на страницу Display в виде отдельного блока верхнего уровня (например, в виде [Media File](#) или [Composition](#)).
2. Включите этот блок верхнего уровня там, где необходимо, используя Reference блок, указав желаемый блок как "Referenced group/block".

Такой общий доступ к образцу часто используемого контента упрощает обновление этого контента в тех местах, где он используется.

Select Target Block (выбрать целевой блок)

Имя вводимого блока может быть указано явно или получено динамически через системное свойство.

- По group/block Name (по имени группы/блока). Выбирает нужный блок с помощью "Target Block" (блока Цель)
- По Property Value (значению свойства). Имя задается соответствующим системным свойством.

Последний из этих двух вариантов можно использовать, например, для того, чтобы один спот отображал тот же блок, что и другой спот, возможно, в виде небольшого окна предварительного просмотра. Это делается путем привязки к свойству playBlock другого спота.

Live Video 📺 (живое видео)

Используйте блок Live Video для отображения потокового видео, поступающего с камеры, потокового устройства, веб-службы или другого источника. Как и в описанном выше веб-блоке, в большинстве случаев вам потребуется URL-адрес видеопотока.

◆ **ПРИМЕЧАНИЕ.** Это тип премиального блока, для отображения которого требуются дополнительные варианты лицензии.



Форматы и настройки живого видео

Выберите ожидаемый тип видеопотока в разделе Format:

- RTSP stream. Включает звук. Низкая задержка (обычно 0,5 секунды или меньше). Поддерживается большинством сетевых камер и потоковых боксов. Совместимость с большинством браузеров/плееров.
- RTMP stream. Включает звук. Низкая задержка (обычно 0,5 секунды или меньше). Поддерживается большинством потоковых боксов. Совместимость с большинством браузеров/плееров. Обратите внимание, что некоторые потоковые решения требуют, чтобы сервер (здесь Blocks) прослушивал поток для подключения, а это означает, что поток должен отображаться, чтобы соединение работало. Другие решения для потоковой передачи (такие как популярный [ATEM Mini Pro](#)) более гибкие, они автоматически повторяют попытку подключения до тех пор, пока оно не станет успешным.
- HLS stream. Включает звук. Задержка значительна (от 3 до 10 секунд и более). Поддерживается некоторыми потоковыми приставками и большинством веб-сервисов. Совместимость с большинством браузеров/плееров.
- MJPEG stream. Низкая задержка. Поддерживается многими видеокамерами и потоковыми приставками. Звук не включён. Совместимость с большинством браузеров/плееров.
- JPEG polling. Достаточно низкая задержка (менее 1 секунды). Как правило, не полная частота кадров. Звук не включён. Поддерживается некоторыми видеокамерами. Совместим со всеми браузерами/плеерами.
- Local Video In. Видео предоставляется через локальный адаптер захвата или веб-камеру. Очень низкая задержка (несколько кадров). Полезно, например, для передачи живой презентации PowerPoint в Blocks путем подключения ноутбука, на котором запущена программа PowerPoint, к адаптеру HDMI-USB, подключенному к порту USB

проигрывателя PIXILAB. Адаптер должен поддерживать стандарт "[USB video device class](#)" иногда называемый «UVC», «без драйверов» или «совместимый класс». Если имеется более одного источника видеовхода, выберите, какой из них использовать, введя его номер в поле Индекс устройства, начиная с номера 1. Номера устройств могут измениться, если такие устройства будут добавлены, удалены или подключены к другим портам на плеере.

- ◆ Local Video In может потребовать определенных настроек для автоматического предоставления доступа к видеоустройству. PIXILAB Player предварительно сконфигурирован со всеми соответствующими настройками. Если вы используете обычный веб-браузер, проверьте дополнительные настройки браузера.

Stream Includes Audio (поток включает звук)

Выберите это, если требуется звук, или если источник предоставляет звук.

- ◆ Для форматов RTSP и RTMP этот параметр должен соответствовать исходному потоку с точки зрения доступности звука. Несоответствие здесь может привести к сбою потока.

Use TCP Transport (использовать TCP-транспорт)

Применяется только к формату RTSP. Обычно RTSP использует UDP для передачи потока с камеры/кодировщика в блоки. В некоторых сетях TCP может потребоваться для надежного соединения. Попробуйте этот вариант, если обычный метод UDP не работает надежно.

Optimize Stream for (оптимизировать поток для)

Применяется для RTSP и RTMP. Эти форматы потоков могут обеспечить потоковую передачу с низкой задержкой. Однако попытка минимизировать задержку может привести к пропаданию видео, если исходные данные не поступят вовремя.

- Lowest Latency (минимальная задержка) сводит к минимуму задержку с риском зависаний или временного пропадания видео в случае задержки данных.
- Best Reliability (лучшая надежность) поддерживает больший буфер данных, тем самым снижая риск потери видео за счет более длительной задержки.

Subtitle (субтитры)

Используйте этот блок для рендеринга субтитров или текста субтитров, возможно, синхронизированного с видео или аудио. Blocks принимает файлы субтитров на основе стандартных форматов файлов [SRT](#) или [VTT](#). Чтобы синхронизировать субтитры с видео, поместите видео вместе с субтитрами в синхронизатор. Используйте [Tag Selector](#), чтобы выбрать различные субтитры на основе выбора языка или аналогичного.

Перетащите файл субтитров в поле «Перетащите сюда файл субтитров», которое отображается справа при выборе блока субтитров. В качестве альтернативы загрузите данные субтитров из URL-адреса, указанного связанным свойством, для дополнительной гибкости, аналогично типу блока [Media URL](#).

Если видео и субтитры воспроизводятся в одном месте, выберите "Locally Synchronized" для субтитров. Снимите этот флажок, чтобы синхронизировать субтитры между роликами.

Узнайте больше о том, как использовать субтитры в блоках здесь: <https://pixilab.se/docs/blocks/app-note/subtitles>

Schedule 🕒 (расписание)

Этот тип блока позволяет отображать другие блоки в заранее определенное время или дату. Расписание назначается споту, как и любому другому блоку. Однако Расписание нельзя использовать внутри других блоков; только как блок верхнего уровня.

The screenshot displays the 'Blocks' scheduler interface. On the left, under 'Schedule Layers', there are two layers: 'Christmas' (STARTS: 2018-01-20, ENDS: 2018-01-27) and 'Base Content' (STARTS: 2017-11-01). The main grid shows a weekly schedule from Monday to Sunday. The time slots range from 8:00 to 18:00. Content blocks are scheduled as follows:

Day	Time Slot	Content Block
Monday	11:00-12:00	My Pix (Main)
Monday	12:00-13:00	S_Film1 (FSTest)
Monday	13:00-15:00	Saltkrakan (Kiosk)
Monday	15:00-18:00	Collage3 (epi)
Tuesday	8:00-11:00	Collage1 (epi)
Saturday	8:00-17:00	Lands1080p (Main)

On the right, the 'SCHEDULER' configuration panel is visible, showing settings for the selected block: 'Main/Lobby', 'Layer Name: Base Content', 'Enabled' (checked), 'Starts on: 2017-11-01', and 'Ends after: 2018-08-01'.

Разнообразный контент запланирован в течение дня, а в выходные воспроизводится другой контент.

Чтобы добавить контент в расписание, нажмите кнопку «плюс» в правом верхнем углу. Найдите нужный блок и перетащите его на график. Блоки можно фильтровать по имени включающей их группы, что упрощает их поиск, если в вашей системе много блоков.

После добавления блока в расписание переместите его или измените его размер, чтобы определить, когда он появится, или введите время его начала/окончания.

Schedule Layers (слои расписания)

Содержимое, добавляемое в расписание, организовано в виде слоев. Изначально есть только один слой. Чтобы добавить слой, выберите “Layer” (слой) в меню Add (добавить). Слой можно отключить, сняв флажок «Включено». Отключенные слои не будут воспроизводиться. Точно так же вы можете ограничить даты воспроизведения слоя, используя “Starts on” и “Ends after”. Снимите флажок “Ends after” (заканчивается после), чтобы не указывать дату окончания.

Используйте дополнительные слои для планирования альтернативного контента, сохраняя при этом обычный контент. Система будет воспроизводить контент самого верхнего включенного слоя, который находится в пределах диапазона дат и имеет содержимое для текущего времени и дня недели. Если в самом верхнем слое нет контента, соответствующего этим критериям, будет рассматриваться нижний слой и так далее.

The screenshot displays the 'Blocks Scheduler' interface. On the left, a 'Schedule Layers' list shows 'Christmas' (STARTS: 2018-01-20, ENDS: 2018-01-27) and 'Base Content' (STARTS: 2017-11-01). The main area is a grid with days of the week (Mon-Sun) and times (10:00-15:00). A blue block is visible on the grid, labeled 'P1 (Panels) 11:00-14:00', with a small image of a woman. On the right, a settings panel for the 'Main/Lobby' layer shows 'Christmas' as the layer name, 'Enabled' checked, 'Starts on' 2017-12-20, and 'Ends after' 2017-12-27.

Одиночный блок на слое Christmas будет воспроизводиться с 11:00 до 14:00 между указанными датами вместо блоков на базовом слое. Содержимое базового слоя будет продолжать воспроизводиться в другое время.

6. BEHAVIORS (ПОВЕДЕНИЕ)

Поведения добавляют интерактивные возможности дочерним блокам. Вы можете добавить одно или несколько поведений к блоку с помощью меню Add Behavior (добавить поведение). Поведения привязаны к локальным или системным свойствам. В зависимости от выбранного поведения свойство может либо влиять на дочерний блок, либо наоборот.

Чтобы просмотреть поведение, прикрепленное к дочернему блоку, сначала выберите блок в списке слева в редакторе, затем щелкните символ в форме ромба в верхнем правом углу, как показано на рисунке справа.

Вы можете добавить несколько вариантов поведения в блок. Пример, показанный справа, имеет одно выбранное в данный момент поведение Move. Дополнительные варианты поведения отображаются в списке вверху. Поведению можно присвоить имя, что может быть полезно при применении нескольких поведений одного типа. Щелкните поведение в этом списке, чтобы просмотреть его настройки ниже.

Вы можете копировать и вставлять поведение между блоками:

1. Выберите желаемое поведение в списке.
2. Выберите Copy в меню Edit.
3. Перейдите к целевому дочернему блоку.
4. Выберите Past (вставить).

Блоки с поведением отображаются курсивом в списке дочерних блоков, а также в представлениях иерархии.

- ◆ Поведения применяются при отображении блока на месте. Они не применяются в редакторе. Используйте тестовую область, для которой установлено значение “Update on Block Change” (обновлять при изменении блока), чтобы просмотреть результирующее поведение при редактировании.



Appears (появление)

Привязанному свойству будет присвоено указанное значение (или true, если значение не указано) при появлении блока. Это можно использовать для запуска действия, когда блок появляется на экране, например, при прокрутке в поле просмотра в Scroller. Пороговое значение указывает, какая часть блока должна быть видимой, чтобы его можно было рассмотреть. Установите небольшое значение (например, 5%), чтобы поведение вступало в силу, когда блок виден лишь частично, или большое значение, чтобы оно вступало в силу, когда весь блок виден полностью (100%).

Disappears (исчезание)

Привязанному свойству будет присвоено указанное значение (или true, если значение не указано), когда блок исчезнет. Это может, например, использоваться для запуска действия при выходе со страницы в Книге или когда видео прокручивается вне поля зрения.

Move (движение)

Заставляет блок изменить свое положение на экране в соответствии со свойством привязки (bound property).

Direction (направление)

Указывает направление движения.

Factor and Offset (коэффициент и смещение)

Блок будет перемещен на расстояние, полученное из такого уравнения:

$bound\ property\ value * factor + offset$

Это полезно, например, при работе с нормализованным (0...1) значением свойства. Чтобы переместить блок на 200 пикселей в пределах диапазона свойства, установите коэффициент равным 200. Коэффициент и/или смещение могут быть отрицательными.

Moved when at (перемещено, когда)

При привязке к строковому или логическому свойству укажите состояние свойства, когда следует применять движение.

Moved when not at value

Инвертирует условие движения, указанное параметром “Moved when at” (перемещено, когда). Иногда это полезно в сочетании со строковым свойством, позволяющим применять движение, не достигая указанного значения.

Minimum Fade Rate (минимальная скорость исчезновения)

Set to a time, in seconds, to move gradually over the specified time, rather than jumping to the destination position.

Mute (беззвучно)

Отключает звук в зависимости от значения привязки свойства. Применяется к следующим типам блоков: Media, Media URL и Live Video.

- ◆ **ПРИМЕЧАНИЕ.** Если нет привязки свойства, безоговорочно отключить звук. Иногда это полезно для отключения нежелательного звука в видео. Это также может помочь при нацеливании на браузеры, которые не будут автоматически воспроизводить видео, содержащее звуковую дорожку (даже без звука).

Mute while at Value (отключить звук при значении)

Если указано, звук этого блока будет отключен, пока свойство имеет указанное значение. Если не указано, значение свойства напрямую управляет функцией отключения звука, при значении 1, true или непустой строке звук отключается.

Opacity (непрозрачность)

Показывает и скрывает блок на основе значения привязки свойства. Безоговорочно скрывает, если нет привязки свойства.

Visible while at Value (отображается при значении)

Если указано, блок будет виден при указанном значении, иначе он будет скрыт. Иногда это полезно в сочетании со строковым свойством, значение которого определяет видимость. Если не указано, значение свойства напрямую управляет непрозрачностью блока, при этом значение 0 или false означает, что блок скрыт, а значение 1 или true означает, что блок полностью виден.

Minimum Fade Rate (минимальная скорость затухания)

Установите время в секундах для постепенного исчезновения в течение указанного времени, а не для мгновенного появления или исчезновения.

Play (воспроизводить)

Заставляет блок (например, видео, звук или слайд-шоу или любой блок, содержащий такие блоки) воспроизводиться или приостанавливаться, основываясь на значении свойства привязки. Применяется только к блокам, которые могут воспроизводиться: видео, звуку и слайд-шоу.

Play while at Value (воспроизводить при значении)

Если указано, дочерний блок будет воспроизводиться при указанном значении, в противном случае он будет приостановлен. Иногда это полезно в сочетании со строковым свойством, значение которого затем управляет воспроизведением. Если не указано, значение свойства напрямую управляет состоянием блока, при этом значение 0 или false приостанавливает его, а 1 или true воспроизводит.

Playing (воспроизведение)

Свойство bound (привязки) будет установлено в true, пока блок (или любой из его дочерних элементов) воспроизводится, и будет установлено в false, когда воспроизведение остановится. Это можно, например, использовать для запуска действий, когда начинает воспроизводиться определенное видео.

Press (нажатие)

Имитирует нажатие кнопки, видео, камеры и т. д. на основе значения свойства привязки. Это позволяет программно управлять функциями кнопок и других блоков, которые можно «нажимать».

Press while at Value (нажать, пока свойство имеет значение)

Если указано, то блок будет нажат, пока свойство находится в указанном значении, в противном случае он будет отпущен. Если не указано, значение свойства напрямую управляет поведением, при этом ненулевое значение или булево true приводит к тому, что блок будет нажат.

Reveal (раскрыть)

Вызывает отображение блока, когда свойство установлено. Это особенно полезно для показа глубоко вложенных блоков, например, блока внутри книги, слайд-шоу или подобного.

◆ **ПРИМЕЧАНИЕ.** Это поведение не работает внутри реплицированных дочерних блоков ([replicated child blocks](#)).

Reveal when at Value (раскрыть при значении)

Если указано, блок будет раскрыт при указанном значении. Иногда это полезно в сочетании со строковым свойством, значение которого регулирует появление. Если не указано, то значение свойства непосредственно вызывает появление блока, если установлено истинностное значение, т.е. значение, которое не является нулем, ложью, неопределенным или пустой строкой.

Only if Parent Visible (только если родитель виден)

Когда выбрано, блок не будет показан, если его охватывающий блок уже не виден. Если не выбрано, блок будет раскрыт независимо от того, даже если он глубоко вложен в окружающие блоки, которые могут быть или не быть уже видимыми.

Rotate (поворот)

Заставляет блок вращаться на экране в соответствии с привязанным свойством, используя значение, указанное в градусах, при положительных значениях - вращение по часовой стрелке. Если нет привязки свойства, поверните на 180 градусов или на величину, указанную в качестве смещения (Offset).

Factor and Offset (коэффициент и смещение)

Блок будет повернут на число градусов, полученное из этого уравнения:

$$\text{bound property value} * \text{factor} + \text{offset}$$

Это полезно, например, при работе с нормализованным (0...1) значением свойства. Чтобы повернуть блок на 360 градусов в диапазоне свойства, установите коэффициент равным 360. Коэффициент и/или смещение могут быть отрицательными.

Applied while at (применяется в то время как)

При привязке к строковому свойству укажите состояние свойства, к которому следует применить поворот.

Minimum Fade Rate (минимальная скорость затухания)

Установите время в секундах для постепенного вращения в течение указанного времени, а не для перехода к целевому положению (углу).

Scale (масштаб)

Похоже на поведение Rotate (поворот), описанное выше, но вместо этого влияет на размер блока. Значение масштабирования больше 1 делает блок больше, а меньше 1 делает его меньше.

Speed (скорость)

Влияет на скорость воспроизведения видео и звука. Значение привязки его свойства управляет скоростью. Значение 1 воспроизводится с нормальной скоростью. Значение больше 1 воспроизводится быстрее. Выберите “Preserve Audio Pitch” (сохранить высоту звука), чтобы сохранить постоянную высоту звука при изменении скорости воспроизведения. Снимите этот флажок, чтобы высота тона соответствовала скорости воспроизведения

- ◆ **ПРИМЕЧАНИЕ.** Это поведение несовместимо с некоторыми проигрывателями, например с iOS, где скорость нельзя изменять постепенно. Очень быстрое воспроизведение не будет работать должным образом на большинстве устройств из-за ограничений производительности декодирования аудио/видео. Очень медленное воспроизведение приведет к прерывистому воспроизведению видео.

Volume (громкость)

Управляет громкостью воспроизведения своего медиаблока. Настройки аналогичны поведению [Opacity](#) описанному выше.

Classify

Условно применяет к блоку пользовательский класс CSS на основе значения привязки свойства.

CSS Class

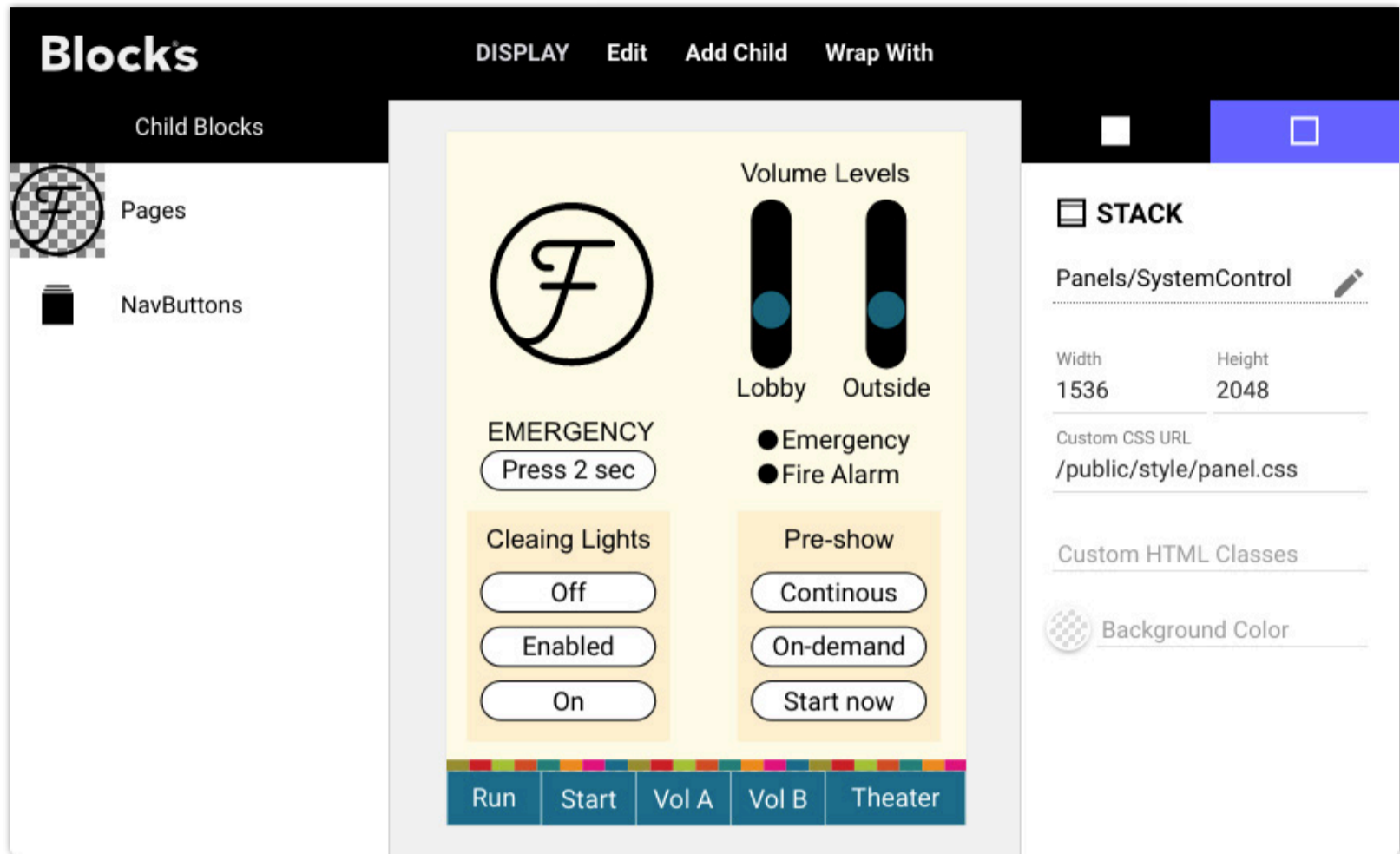
Применяемый пользовательский класс. Это похоже на пользовательские классы CSS, которые можно применять статически к любому блоку, но здесь класс применяется только при указанном условии.

Apply at Value (применить при значении)

Если указано, то класс будет применяться при указанном значении. Иногда это полезно в сочетании со строковым свойством. Если значение свойства не указано, то класс будет применяться при истинном значении (т.е. значении, которое не равно нулю, false, undefined или пустой строке).

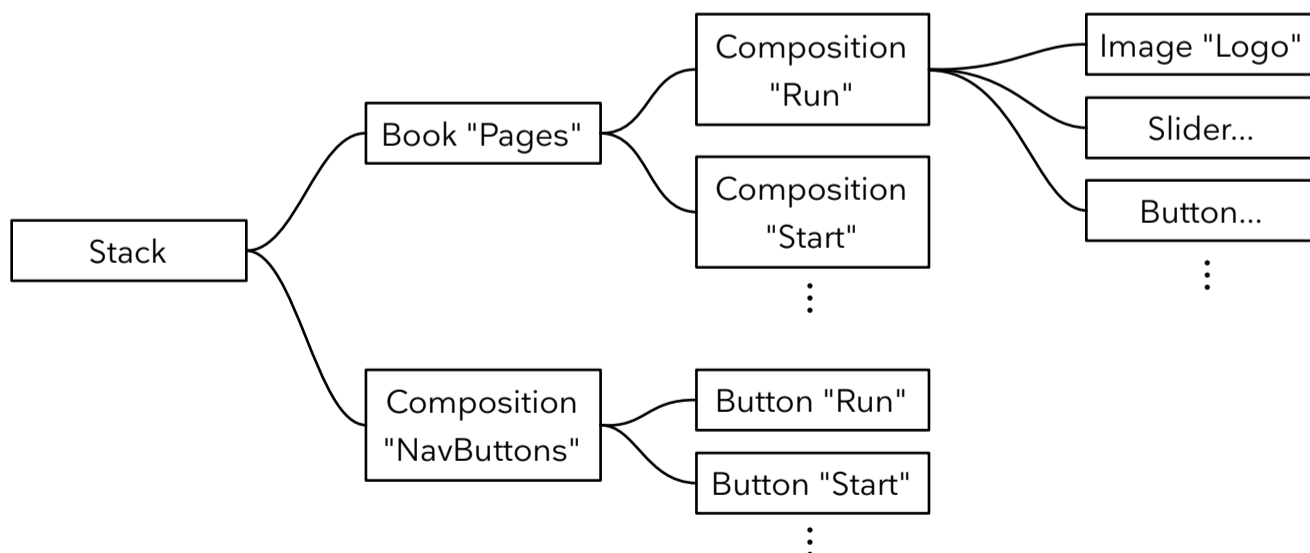
7. УПРАВЛЕНИЕ И ПАНЕЛИ

Используйте блок [Composition](#) для создания панели с кнопками и другими элементами управления, а также других типов блоков, которые вы хотите включить. Это можно использовать для создания интерактивных киосков для посетителей, а также сложных панелей управления для внутреннего использования персоналом (например, для управления воспроизведением мультимедиа на дисплеях, питанием, подсветкой, уровнями громкости и т. д.).



Панель с кнопками и ползунками. Синие кнопки на панели навигации внизу имеют индивидуальный стиль.

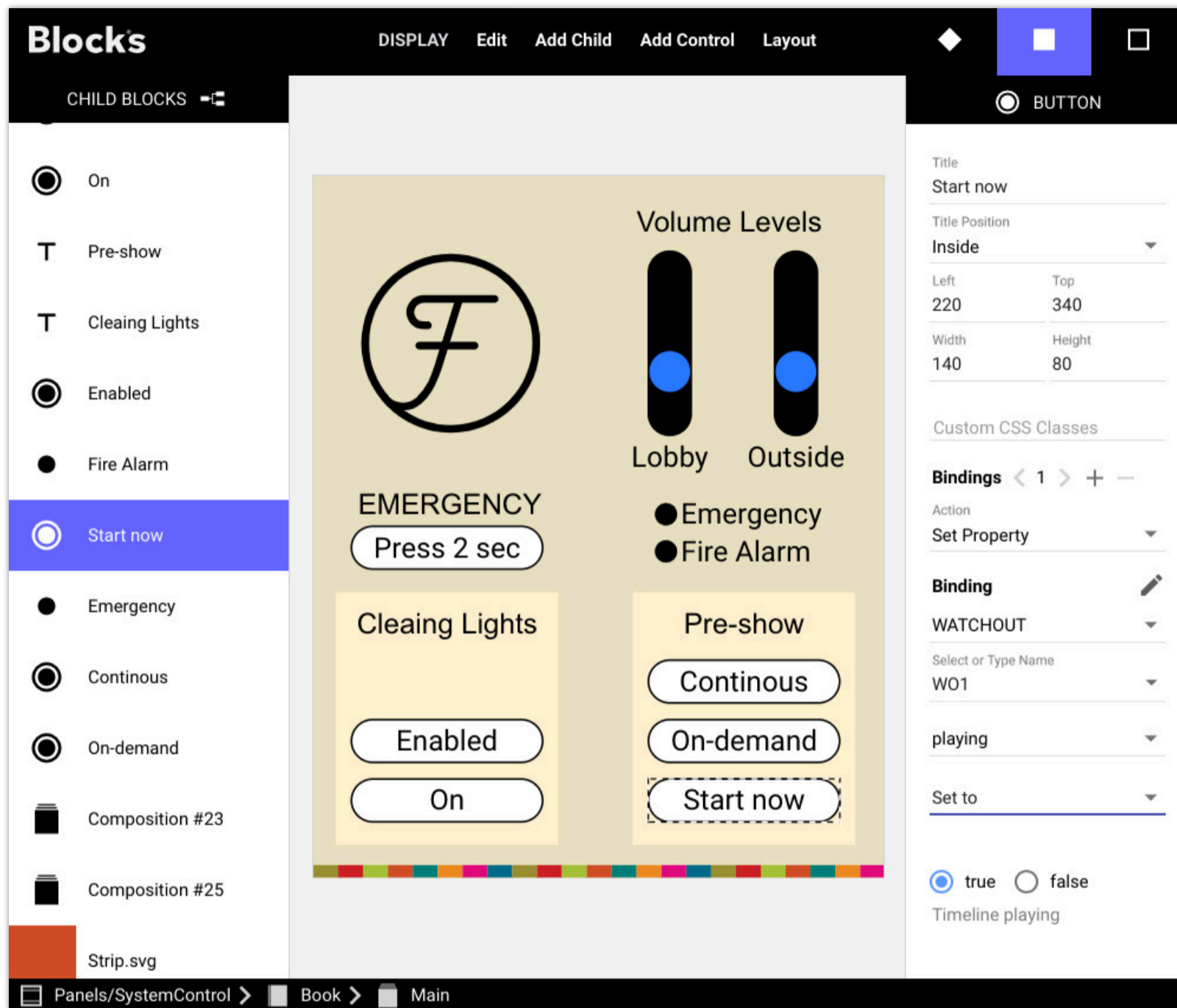
Кнопки и другие элементы управления добавляются в композицию с помощью меню ["Add Control"](#). Показанный выше пример имеет блочную структуру, приведённую на схеме (см. ниже).



Панель управления, основанная на блоке Stack, с набором кнопок навигации, установленным на «Stick to Edge» внизу.

Button (кнопка)

Используйте кнопку для управления состоянием вашей системы, для навигации по интерактивному контенту или для интерактивного назначения тегов споту.



Кнопка запуска шоу WATCHOUT.

Title and Title Position (титул и его расположение)

Определяет имя кнопки, а также то, где оно должно появляться на кнопке или за ее пределами (или не отображаться вообще).

Position (расположение)

Позволяет численно указать положение и размер кнопки. Либо перетащите кнопку или один из ее краев мышью. Включите "[Snap to Grid](#)" (привязать к сетке) в настройках внешнего блока, чтобы упростить выравнивание. Вы также можете использовать команды в меню *Layout*, чтобы расположить несколько кнопок или других элементов управления по горизонтали или по вертикали.

Bindings (привязки)

Изначально кнопка имеет одну привязку. Чтобы добавить больше привязок, щелкните символ +. Используйте стрелки для перехода между привязками, на что указывает цифра между стрелками. Подсветка кнопки определяется ее первой привязкой.

Action (действие)

Выбирает желаемый тип действия. Это может быть установка свойства (например, “play a video on a spot” или “включить свет”) или локальное действие, влияющее только на локальный спот, например, переход на другую страницу или установка тега.

Set Property (задать свойство)

После выбора “Set Property” (задать свойство) в качестве действия кнопки, вы привязываете кнопку к целевому свойству с помощью полей в разделе Binding (привязка). В примере, показанном выше, выбранная кнопка привязана к состоянию “playing” (воспроизведение) кластера WATCHOUT с именем “WO1”.

- ◆ **СОВЕТ:** В качестве альтернативы использованию меню Binding нажмите на символ пера рядом с заголовком Binding и введите, скопируйте или вставьте привязку как текст. Это особенно удобно при настройке большого количества одинаковых привязок

Параметры, отображаемые под целью привязки, зависят от объекта и типа выбранного свойства. В приведенном выше примере свойство “playing” имеет тип true/false (“булево”), поэтому значения указаны внизу. Последний селектор содержит следующие опции:

- Set to Constant устанавливает состояние свойства на указанное буквальное значение, когда кнопка нажата.
- Set while pressed устанавливает состояние в одно значение при нажатии и в другое значение при отпускании. Это приводит к кратковременному действию, которое сбрасывается на альтернативное значение, как только вы отпускаете кнопку.
- Toggle переключает состояние между двумя возможными значениями при каждом нажатии кнопки. Это может быть использовано для того, чтобы заставить одну кнопку включать или выключать что-то в зависимости от текущего состояния.
- Append добавляет указанную текстовую строку к уже заданной. Доступно только для текстовых свойств.
- Increase by и Decrease by добавляет (или вычитает) число к свойству. Используется для увеличения/уменьшения числового свойства, например, уровня освещенности, на фиксированное значение при каждом нажатии кнопки.
- Set to Property Value устанавливает состояние связанного свойства на текущее значение другого свойства (“Property Value Source”). Позволяет параметризовать значение [spot parameter](#) или другим подходящим свойством.


Типы свойств

Значения, которые можно установить с помощью кнопки, зависят от типа выбранного свойства. Доступны следующие типы:

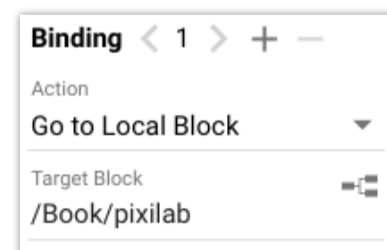
- Boolean (**булево**). Доступны только *true* и *false*. Это относится к свойству “воспроизведение”, показанному на рисунке выше.
- Number (**число**). Числовое значение, которое в некоторых случаях может быть отрицательным и содержать дробную часть. Примером может служить уровень громкости дисплей-спота или значение “Input” для подсистемы WATCHOUT.
- String (**строка**). Текстовое значение, например, имя блока, назначенного споту.
- Time (**время**). Временная позиция, выраженная в формате ЧЧ:ММ.СС.fff, единицей измерения по умолчанию являются секунды. Примером может служить временная позиция временной шкалы WATCHOUT.

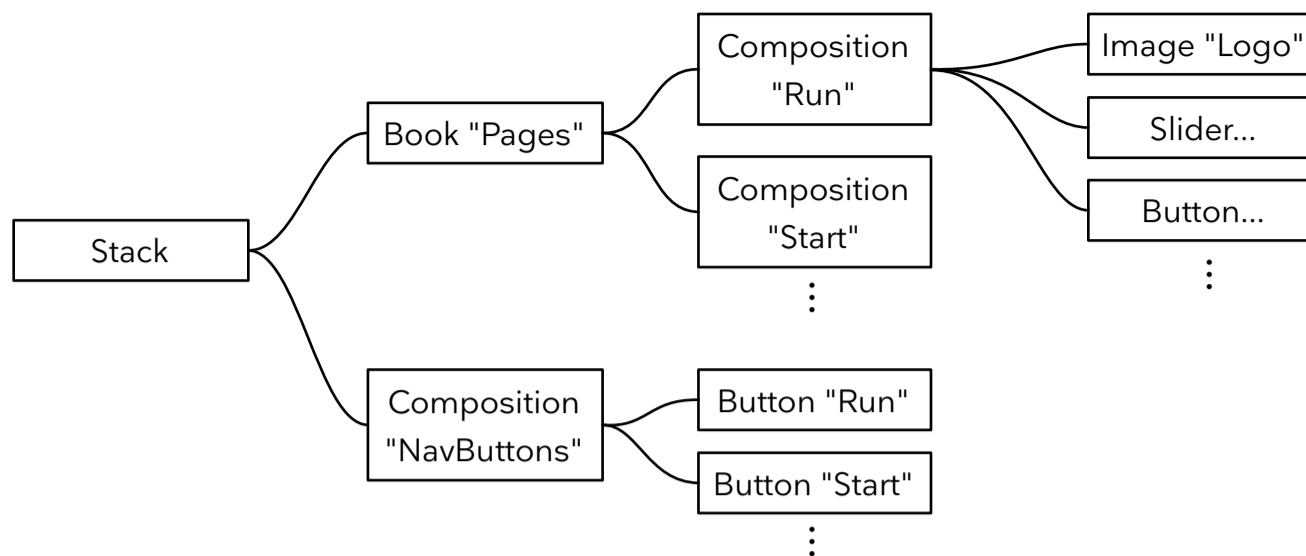
Смотрите [“Properties and Functions”](#), там есть детальное описание свойств кнопок.

Go to Local Block (перейти к локальному блоку)

Переход к другому дочернему блоку в текущем корневом блоке. Выберите целевой блок с помощью кнопки выбора иерархии  затем дважды щелкните нужный целевой блок в представлении иерархии.

В качестве альтернативы укажите целевой блок с помощью текстовой строки. В ней указывается путь к целевой странице, либо начиная со страницы, содержащей кнопку (относительный путь), либо с самого верхнего блока (абсолютный путь). Используйте кнопку Copy Path (копировать путь) в адресной строке при редактировании вложенного блока, чтобы получить его полный (абсолютный) путь внутри корневого блока.





Например, если вы хотите, чтобы кнопка “Start”, показанная на рисунке выше, открывала страницу “Start” в Книге, вы можете указать путь одним из следующих способов:

- As “/Pages/Start”. Это задает абсолютный путь, начинающийся с верхнего уровня блока Stack, который содержит Книгу с именем “Pages”, которая, в свою очередь, имеет композицию с именем “Start”.
- As “../Pages/Start”. Это относительный путь, начинающийся со страницы, содержащей кнопку. Кнопка находится на странице “NavButtons”. композиции в блоке верхнего уровня Stack. Две точки (...) означают “подняться на один уровень”, что позволяет перейти на верхний уровень блока Stack из которого вы перейдете в “Pages” и в “Start”.

В этом случае предпочтительнее абсолютная форма, поскольку она и короче, и проще для понимания. Однако в более сложных и глубоких иерархии блоков часто проще использовать относительные пути.

- ◆ Абсолютный путь начинается с прямой косой черты, а относительный - нет. Синтаксис пути похож на то, как указываются пути файловой системы в операционных системах на базе Unix, таких как macOS.

Определение местоположения цели численно или инкрементально (пошагово)

В качестве альтернативы именованным целям можно использовать числовое положение цели. Вместо имени можно использовать просто число, где первый дочерний блок имеет номер 0 (ноль). В приведенном выше примере вы можете перейти ко второй странице книги “Pages”, указав путь “/Pages/1” (где 1 - это вторая страница, так как первая страница имеет номер 0).

- ◆ Число в пути всегда указывает целевой индекс. Таким образом, не называйте страницу только цифрами, если вы хотите перейти к этой странице по имени. Если вы хотите использовать здесь число, добавьте к нему текст, например, “page2”, “page3” и т.д.

Аналогичный синтаксис можно использовать для постепенного перемещения вперед или назад на заданное количество страниц. Чтобы переместиться на две страницы вперед в книге, укажите “/Pages/+2”, или на одну страницу назад - “/Pages/-1”. Этот синтаксис можно использовать для перехода вперед или назад, но он не обернется, когда будет достигнута первая или последняя страница/слайд.

Наконец, использование только + или - (без какой-либо последующей цифры) приведет к перемещению на один шаг, оборачиваясь в слайд-шоу или книге от последней страницы к первой, или наоборот. Перемещение вперед по слайд-шоу таким образом применит любые переходы.

Нахождение целей внутри блока Reference (ссылка)

При использовании блока [Reference](#) для приведения другого блока верхнего уровня, вы можете перемещаться внутри блока со ссылкой, используя любой из методов, описанных выше. Однако если вам нужно найти цель в блоке за пределами ссылочного блока, необходимо использовать немного другой синтаксис. Это связано с тем, что синтаксис абсолютной цели (начинающийся с одной прямой косой черты, как обсуждалось выше) работает на основе корня ссылающегося блока - точно так же, как если бы ссылающийся блок использовался сам по себе.

Чтобы выйти за пределы ссылающегося блока, вплоть до конечного корня крайнего блока в цепочке ссылок, используйте абсолютный синтаксис который начинается с двух прямых косых черт //. Затем вы можете перемещаться вниз, в обычные дочерние блоки или ссылающиеся на них блоки (которые, в свою очередь, могут ссылаться на другие блоки на нескольких уровнях).

- ◆ Этот синтаксис двойной косой черты не требуется при определении местоположения целей с помощью постановки задачи, поскольку такие ссылки всегда начинаются на конечном корневом уровне.

Exclude from "Go Back" (исключить из "Вернуться назад")

Обычно навигация по блокам, инициируемая кнопками, записывается в историю навигации, что позволяет использовать кнопку "[Go Back](#)" чтобы вернуться назад по этой последовательности блоков. Однако в некоторых случаях - например, при использовании кнопок для перелистывания страниц в Книге - это может быть нежелательно. В таком случае выберите этот параметр, чтобы любая кнопка "Go Back" пропускала промежуточные шаги навигации.

Locate Spot

Это действие используется вместе с [Locator](#), в качестве альтернативы другим способам определения местоположения. Оно не имеет никакого эффекта, если нет активного Локатора. Такие кнопки можно использовать как способ указания местоположения на основе меню, а не вводить ID или сканировать QR-код. Если вы не используете ни один из других способов определения местоположения, клавиатура Локатора не будет отображаться. Но она должна оставаться активной, чтобы действие "Определить местоположение" сработало. Оставьте поле "Имя целевого спота" пустым, чтобы закрыть любой открытый в данный момент Локатор.

- ◆ При использовании этого метода в поле "Spot Name" введите название спота, а не его ID местоположения. ID местоположения не для работы этого метода. Если целевой спот находится в группе спотов, введите полный путь, начиная с самого дальнего спота группы с каждого уровня, разделенного точкой.
- ◆ **СОВЕТ:** Вы можете оставить поле целевого спота пустым, чтобы закрыть любой текущий Локатор, а не переходить к новому споту.

Go Back (вернуться)

Это действие эквивалентно нажатию кнопки "Назад" в браузере. Строка URL и кнопки навигации браузера часто скрываются при использовании в Blocks. Это действие можно использовать для перехода к предыдущему блоку. Оно часто используется в сочетании с действием "Go to Local Block" (описано выше), как способ обратной навигации..

Локальные теги Select/Reset/Toggle (выбор/сброс/переключение)

Опции, связанные с тегами, используются для добавления/удаления тегов. Тег (метка) - это одно слово или несколько слов, используемых для влияния на поведение спота. Укажите несколько тегов, разделяя их запятой. Доступны следующие опции:

- Toggle Tags - переключить теги. Нажатие кнопки добавит тег(и), если он еще не установлен, или удалит установленный тег(и).
- Reset Tags - сбросить теги. При нажатии кнопки указанные теги будут удалены.
- Select Tags - выбрать теги. Устанавливает указанный тег(и) из набора тегов. То есть, будут установлены теги, указанные в разделе "Теги", а все дополнительные теги, указанные в разделе "Of Tag Set" (из набора тегов), будут очищены. Например, чтобы выбрать одноязычный тег swe из набора swe, eng, ger, введите первый в разделе "Теги" и укажите swe, eng, ger в разделе "Of Tag Set".

Теги могут быть присвоены месту с самого начала с помощью поля "[Tags](#)" в диалоговом окне спота. Дополнительные метки могут быть применены с помощью кнопок, как описано выше, или с помощью задач и скриптов. Набор тегов является локальным для данного спота и не влияет на другие споты.

Текущий набор тегов используется вместе с блоком [Tag Selector](#) или [Synchronizer](#) для выбора дочернего блока для отображения. Это полезно для выбора языка и других подобных локальных настроек.

Setting a Spot Parameter (установка параметра спота)

Выберите "Set Spot Parameter", затем выберите или введите имя параметра и его желаемое значение. Дополнительные сведения см. в разделе "Параметризация путей свойств" ([Parametrization of Property Paths](#)).

Button Feedback (кнопка обратной связи)

Кнопка загорается, чтобы показать состояние первого (или единственного) свойства или локального поведения, которым она управляет. В частности, она загорается при следующих условиях:

- При управлении свойством System кнопка загорается, когда свойство имеет значение "Set to", или первое из двух значений, если "Set while pressed" или "Toggle". если выбрано значение "Set while pressed" (установить при нажатии) или "Toggle" (переключить).

- Если установлено значение Local; Go to Page (Локальный; Перейти на страницу), она загорается, когда указанная страница видна. Это особенно полезно для кнопок "навигационной панели", как в примере, показанном на рисунке выше, где одновременно видны и кнопка, и целевая страница. одновременно.
- При установке значения Local; Select/Toggle Tags загорается, когда все указанные теги установлены.
- При установке параметра Local; установке [Spot Parameter](#), загорается, когда параметр имеет указанное значение.
 - ◆ Эта индикация состояния не отображается в редакторе блоков; только при взаимодействии с реальным спотом Display/Visitor.

Indicator ● (индикатор)

Индикатор похож на кнопку (см. выше), но обеспечивает только функционал обратной связи. Он может быть привязан только к одному свойству.

Slider ⇄ (ползунок)

Слайдер управляет значением числового свойства постепенно. Он подходит для таких свойств, как громкость или яркость. Вы можете указать максимальное и минимальное значения, которые можно установить с помощью ползунка (они могут быть меньше максимального диапазона, поддерживаемого свойством). Его можно использовать и с булевыми свойствами (вкл/выкл), в этом случае ползунок выступает в роли переключателя.

Подключите ползунок к целевому свойству так же, как описано выше в разделе Кнопка, [System Behaviors](#). Ползунок также указывает значение свойства, которым он управляет.

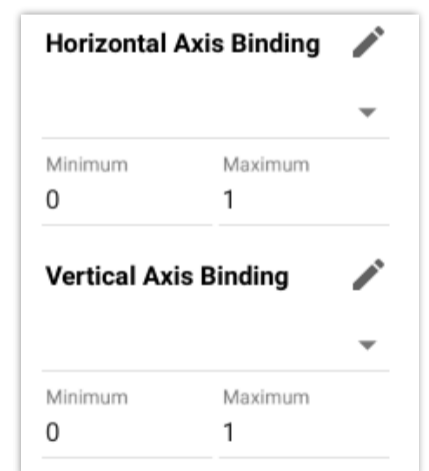
Bar ◻ (полоса)

Bar ведет себя как ползунок и может быть использован только для указания его значения, но не для управления свойством.

Joystick ◻ (джойстик)

Двумерный ползунок с ручкой, которую можно перетаскивать как по горизонтали, так и по вертикали. Особенно полезен для управления панорамированием/наклоном световой головы или камеры типа PTZ. Он должен быть связан с двумя свойствами числовых значений - по одному для каждой оси. Для каждого из этих свойств можно указать допустимый диапазон.

- ◆ **СОВЕТ:** Вы можете переместить связанное свойство между любыми двумя крайними значениями. Например, если установить максимальное значение меньше минимального, то при перемещении джойстика вправо/вниз значение будет двигаться в обратном направлении.



Text T (текст)

Как часть панели управления, блок Text может использоваться для добавления надписей и другой статической текстовой информации. Он также может быть использован для отображения значений системных свойств, таких как числовые значения, временные позиции, строковые переменные или булевы данные (вкл/выкл). См. раздел "[Text](#)" в главе "Блоки" для получения более подробной информации.

Custom Styling (пользовательские стили)

Хотя текст (а также кнопки и другие элементы управления) по умолчанию использует простой шрифт и стиль, вы можете изменить их с помощью каскадных таблиц стилей (CSS), как описано здесь: https://pixilab.se/docs/blocks/custom_styling.

Text Input (ввод текста)

Используйте элемент управления Text Input для ввода числовых значений или текста. Привяжите элемент управления Text Input к нужному свойству, например, к переменной realm. При привязке к числовому свойству можно вводить только числа.

Установите флажок "Multiple Lines", чтобы можно было вводить несколько строк текста. При выборе этого параметра нажатие клавиш Enter/Return при вводе значения создает новую строку, а не принимает введенное значение. Для просмотра нескольких строк текста измените размер элемента управления по вертикали.

QR Code

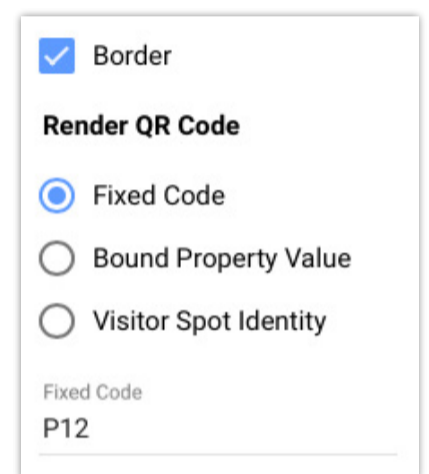
Отображает на экране [QR Code](#) с фиксированными или программируемыми данными.

- ◆ **ПРИМЕЧАНИЕ:** Это премиальный тип блока, для отображения нужны дополнительные лицензионные опции.

Включите "Border" (граница) чтобы автоматически наложить белую рамку вокруг QR-кода для улучшения читаемости на цветном фоне.

Данные, представленные QR-кодом, могут быть числом или коротким текстом, но чаще всего это URL. Он может быть получен из одного из следующих источников:

- Фиксированный код. Просто введите то, что вы хотите отобразить в виде QR-кода.
- Значение связанного свойства. Это может быть локальный параметр Spot, переменная Realm или любое другое свойство вашей системы.
- Идентификатор мобильного устройства посетителя. Эта опция позволяет идентифицировать посетителя путем предъявления QR-кода, отображаемого на дисплее его мобильного устройства. Затем он может быть считан блоком [QR Scanner](#) в посещаемом киоске, тем самым позволяя системе Blocks знать, кто где находится. Эта опция особенно полезна при использовании функции Blocks [Visitor Data Collection](#) (сбор данных о посетителях), чтобы следить за посетителями для лучшей персонализации.



Если вы собираетесь использовать QR-код вместе [Locator](#), QR-код должен содержать одно из следующих значений:

- Обычный номер [Location ID](#). В результате получается самый короткий QR-код, что повышает его читабельность. Это также полезно в случаях, когда вы хотите представить посетителям одновременно ID местоположения и QR-код.
- Имя спота или путь (с сегментами пути, разделенными точкой). Это работает независимо от любого ID местоположения, и не такой ID должен быть присвоен споту.
- URL-адрес, указывающий, какой спот необходимо найти, с помощью параметра запроса spot, добавляемого к URL-адресу, со значением, установленным на ID местоположения или путь спота, как описано в предыдущих двух пунктах.

Последний вариант позволяет использовать QR-код с мобильного устройства для выполнения следующих действий:

1. Открыть веб-страницу Visitor Spot (с помощью части URL).
2. Найти нужный спот в Blocks (используя параметр запроса spot).

Большинство мобильных телефонов имеют встроенный сканер QR-кодов - например, приложение Камера на iOS, Google Lens на Android или отдельное приложение для сканирования QR-кодов. Вот пример URL-адреса такого QR-кода (здесь используется параметр "путь к споту", а не ID):



`https://pixi.guide?spot=Entrance.Welcome`

QR Scanner

Использует камеру, например, веб-камеру, подключенную к дисплей-споту, на котором установлено программное обеспечение PIXILAB Player, для сканирования QR-кода, представляя результат в виде свойства. Хотя это средство можно использовать для сканирования любого QR-кода, оно особенно полезно в сочетании с блоком [QR Code](#), представляющим личность посетителя, как описано в предыдущем разделе.

- ◆ NOTE: ПРИМЕЧАНИЕ: Это премиальный тип блока, для отображения нужны дополнительные лицензионные опции. При отображении на других типах спотов, кроме тех, на которых установлен PIXILAB Player, ваш сервер Blocks должен поддерживать https, а спот должен подключаться по https.
- Выберите Mirror Viewfinder (зеркальный видоискатель) для зеркального отображения предварительного просмотра. Это иногда делает позиционирование изображения более естественным. Это не влияет на сканируемый код.
- При сканировании кода свойство, отображающее результат, будет установлено на отсканированный код на время, заданное параметром QR Code Lingers, а затем сброшено в пустую строку. Это время должно быть достаточно большим, чтобы код был распознан сценарием или задачей, инициированной при сканировании кода, и в то же время достаточно коротким, чтобы не мешать последующему сканированию того же кода. Как правило, здесь достаточно одной секунды, но может потребоваться и меньшее время, если вам необходимо сканировать один и тот же код многократно с высокой частотой.
- Выберите, как передать (expose) отсканированный код: свойству *scannerInput* спота или произвольному свойству (например, переменной [Realm variable](#) или свойству Script).

Mirror Viewfinder
QR Code Lingers (sec)
1.000

Expose Code
 As Spot's scannerInput
 Using Property Binding

8. ПАРАМЕТРЫ СПОТА

Вы можете добавить произвольные пользовательские свойства к дисплей-спотам. Они называются "параметры спота". Они могут использоваться для различных целей, например:

- Интерактивность на локальном или системном уровне, часто в сочетании с поведением.
- Раскрытие пользовательских свойств спота, доступных для других частей Blocks.
- В качестве параметров в привязках свойств, что делает их более гибкими.

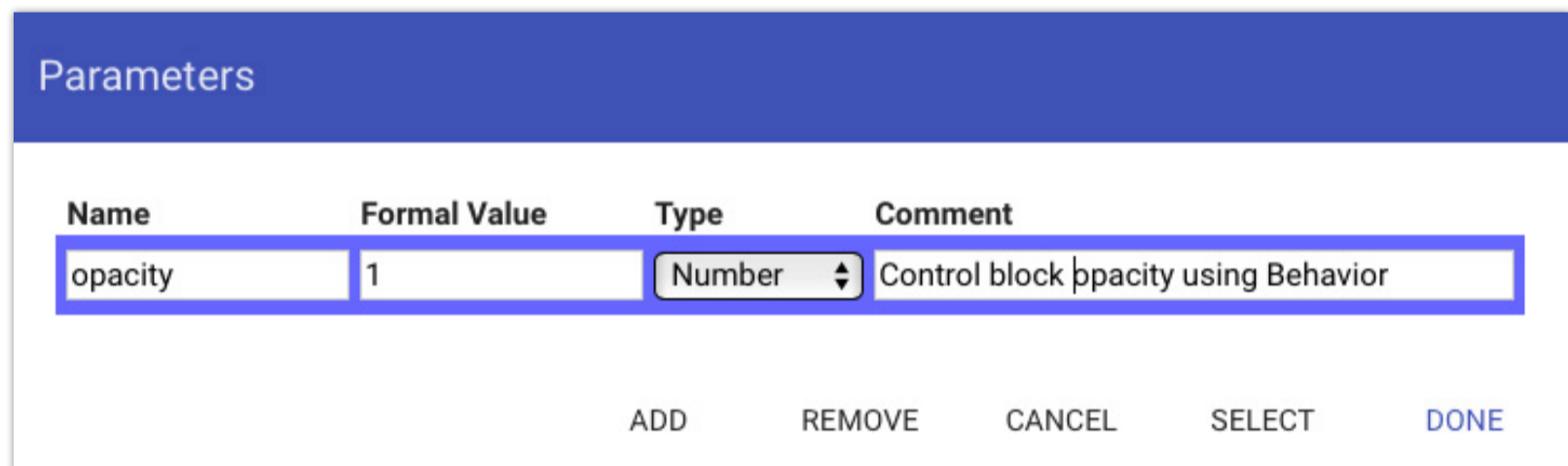
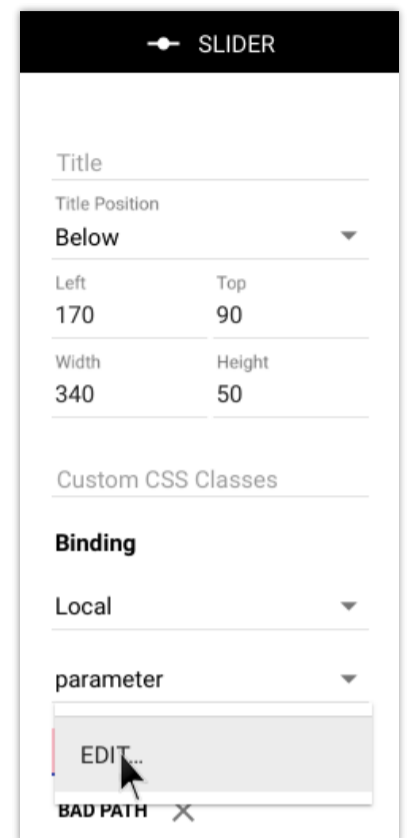
Параметры спота могут быть определены для каждого блока верхнего уровня (как формальные параметры, используемые в данном блоке). Они также могут быть переопределены для каждого спота, задавая таким образом начальное значение для конкретного спота. Более того, значения параметров могут быть изменены во время выполнения, поскольку они открыты как системные свойства. Параметры спота, как и другие системные свойства, имеют определенный тип, например, число, булево (true/false) или текстовая строка.

Определение параметров в блоке

Поскольку параметры обычно используются блоком, они должны быть известны этому блоку. Изначально параметры не определены. Воспользуйтесь командой "Edit Parameters" в меню Edit. Можно также установить и использовать параметр как часть привязки свойства таким образом:

1. Выберите Local в разделе binding, как показано на рисунке справа.
2. Выберите параметр в следующем выпадающем списке.
3. Выберите EDIT... в третьем выпадающем списке.

Уже существующие параметры появляются в третьем выпадающем списке, позволяя привязываться к этому параметру, как к любому другому свойству. Если параметры еще не определены, в этом выпадающем списке есть только команда EDIT.



1. Нажмите ADD, чтобы добавить новый параметр.
2. Введите его имя, которое должно начинаться с алфавитного символа и содержать только алфавитные символы, цифры или знак подчеркивания.
3. Введите его формальное значение. Это не требуется, если параметр имеет тип String.
4. Выберите тип параметра.
5. Введите комментарий, описывающий, для чего вы собираетесь использовать этот параметр.
6. Нажмите кнопку SELECT, чтобы привязать новый параметр.

Привязанный к нему элемент управления здесь является ползунком, который будет управлять значением параметра в виде числа 0...1.

После определения параметра его можно использовать в любом месте этого блока. Поскольку это числовое значение, первоначально установленное на 1, его можно использовать для управления непрозрачностью другого блока с помощью [Behavior](#), связанного с этим параметром.

Formal Value (формальное значение)

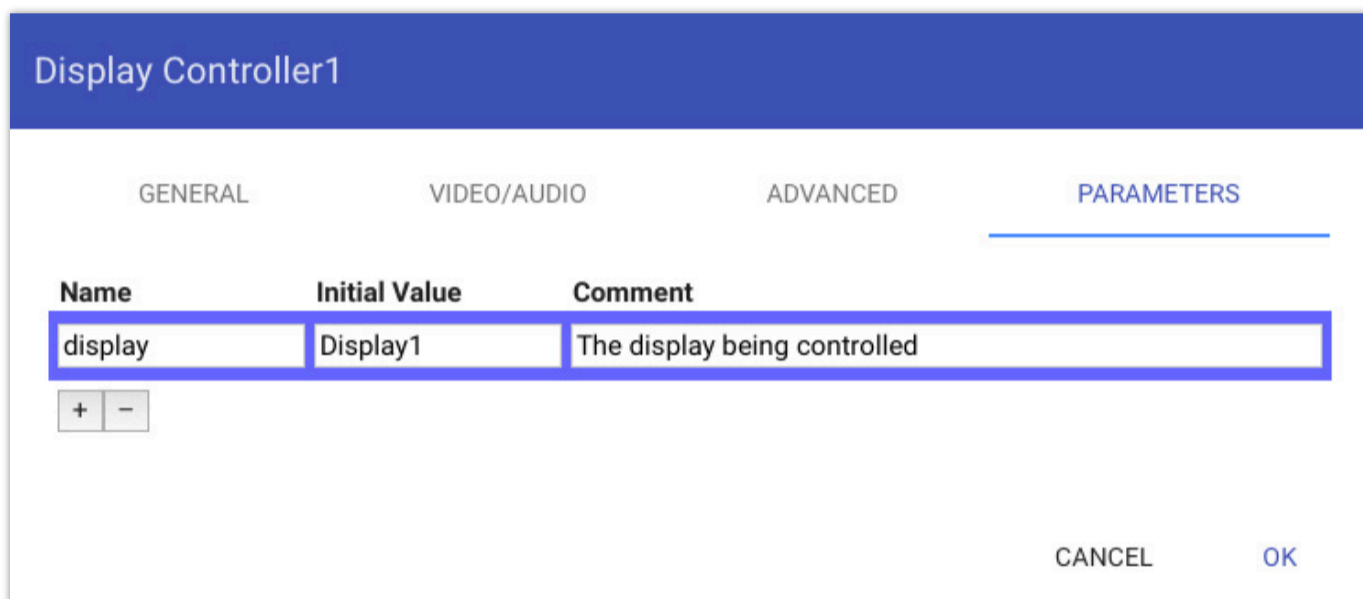
Формальное значение параметра служит для следующих целей:

- Обеспечивает значение параметра при редактировании блока, в котором он определен.
- Предоставляет значение по умолчанию, если оно не переопределено спотом или программно.

Первый пункт особенно важен при использовании параметров в путях свойств, поскольку формальное значение затем используется для разрешения пути свойств, находя репрезентативное системное свойство для привязки. Следовательно, замена заключенного в скобки параметра формальным значением должна привести к существующему пути свойства. Если это не так, измените формальное значение параметра на такое, которое сделает путь действительным.

Определение или переопределение параметров спота

Один из способов использования параметров спота - переопределить их для конкретного спота. Введя то же имя параметра в список Parameters в настройках спота, вы можете переопределить его начальное значение только для этого спота.



Это позволяет использовать один и тот же блок в нескольких спотах с настраиваемой функциональностью (либо в поведении, либо в привязке свойств) на основе локального значения параметра в споте. Подробнее об этом случае использования см. ниже в разделе [Параметризация пути свойства](#).

Доступ к параметрам во время выполнения

Вы можете изменить значение параметра во время выполнения одним из следующих двух способов:

- С помощью элемента управления - например, кнопки, ползунка или текстового ввода - связанного с параметром с помощью пути `Local.parameter.paramname`, как показано на рисунке в начале этой главы. Перемещение ползунка изменяет значение параметра только в локальном споте.
- Установите его через спот, к которому приписан блок. Это можно сделать из любого места в системе, включая элементы управления и задачи. При этом используется путь свойства, например `Spot.spotname.parameter.paramname`.
- При использовании гостевого спота с включенным сбором данных посетителя ([Visitor Data Collection](#)). Поля записи данных посетителя, оформленные с помощью `@spotParameter`, будут отражать значение параметра спота, и наоборот. Это расширенная функция, доступная только из пользовательских скриптов ([User Scripts](#)).

Второй метод доступен только в том случае, если спот с именем `spotname` находится в режиме онлайн и имеет параметр с именем `paramname`.

- ◆ **ВАЖНО:** Чтобы сделать параметр спота доступным, убедитесь, что вы установили параметр в блоке, который назначен для этого спота и что спот подключен к серверу. Вы должны установить его в назначенном блоке, даже если вы собираетесь использовать его только внутри [referenced block](#) или для блока, показанного с помощью [Locator](#).

Определение значения параметра

Текущее значение точечного параметра может быть получено из нескольких различных источников. Эти источники значений используются в следующем порядке, с наивысшим приоритетом - номер 1.

1. Значение, установленное во время выполнения с помощью пути свойства `Local.parameter.paramName` или `Spot.spotname.parameter.paramName` property path.
2. Значение параметра, установленное на споте, на которое нацелен [Locator](#) временно отменяет любое начальное значение при нахождении на этом споте. Это особенно полезно, когда нужно переопределить параметр гостевого спота, находясь в определенных местах.
3. *Initial value* (начальное значение), установленное в настройках спота ([spot's settings](#)). Это также относится к гостевым спотам.
4. Формальное значение, [заданное блоком](#), который в данный момент представлен спотом. Применяются только значения из корневого блока - не значения исходящие из любых блоков, выделяющихся при использовании опорного блока.

Интерактивность и поведение

Параметры спота можно использовать для обеспечения интерактивности в сочетании с [поведением](#). В качестве примера можно привести страницу, на которой показывает три разных видео, расположенных рядом друг с другом. Кнопка под каждым видео может включить воспроизведение своего видео и приостановить любое другое из этих трех видео.

1. Добавьте три видео в композицию.
2. Добавьте три кнопки, по одной под каждым видео.
3. Свяжите каждую из этих кнопок с локальным параметром `videoToPlay` типа String без формального значения, при этом кнопка устанавливает параметр на имя соответствующего видео (или просто имена А, В и С для простоты).
4. Примените поведение Play к каждому из видео, связав его с одним и тем же локальным параметром с именем `videoToPlay` и сопоставив его с теми же тремя именами.
5. Назначьте этот блок споту.

Теперь нажатие любой из этих трех кнопок на этом месте будет вызывать соответствующее поведение Play, заставляя видео воспроизводиться, приостанавливая при этом воспроизведение любого другого видео.

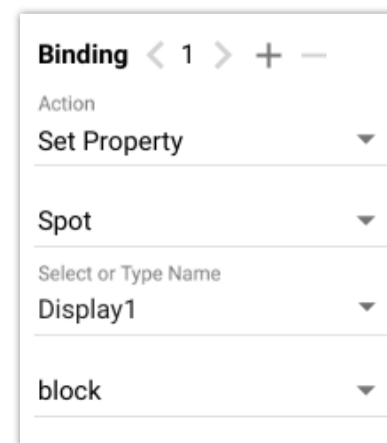
Пользовательские свойства спота

Параметры, установленные блоком, также отображаются как свойства спотов, которым назначен этот блок. Таким образом, вы можете вызывать соответствующие ~~действия извне, как описано выше~~ в разделе "[Доступ к параметрам во время выполнения](#)". Это позволяет вам определять собственные свойства с помощью параметров, которые, в свою очередь, могут управлять или влиять на поведение и привязки свойств.

Использование путей к свойству

Путь к целевому свойству выбирается с помощью набора меню, как показано на рисунке справа. В результате получается фиксированный путь к свойству, то есть кнопка всегда привязывается к одному и тому же свойству (а именно "Spot.Display1.block" в данном примере) независимо от контекста.

Однако в некоторых случаях необходимо, чтобы целевое свойство менялось в зависимости от контекста. Например, рассмотрим набор дисплеев вдоль стены, каждый из которых управляется собственным iPad на подставке. Эти дисплеи отображают один и тот же



контент (блок). На всех iPad отображается одна и та же панель управления дисплеем. Однако вы хотите, чтобы каждый iPad управлял своим дисплеем. Это невозможно решить с помощью кнопок с фиксированными путями свойств, поскольку часть пути свойства (в данном примере имя Spot) должна быть разной для каждой пары iPad/дисплей.

Одним из решений может быть дублирование блока, отображаемого на iPad, с созданием одной копии, соответствующей каждому целевому дисплею. Пути можно изменить, чтобы они соответствовали названию нужного целевого дисплея. Однако, хотя этот способ и работает, он приводит к решению, которое трудно поддерживать и обновлять с появлением новых функций для iPad, поскольку теперь необходимо поддерживать три разных блока.

Лучшим решением является использование параметра вместо жестко закодированного имени дисплея. Когда это возможно, появляется дополнительный элемент в нижней части выпадающего меню под названием [PARAMETER]. При выборе этого пункта появляется диалоговое окно для определения параметров вместе с их формальными значениями (*formal Value*).

Name	Formal Value	Type	Comment
display	Display1	String	Target display to control

ADD REMOVE CANCEL SELECT DONE

Добавьте параметр с именем "display", как показано выше. Задайте его формальное значение, которое можно использовать для тестирования функциональности. Это формальное значение используется редактором для получения набора допустимых дочерних свойств, поэтому оно должно быть именем репрезентативного дисплея в данном примере. Кроме того, это значение будет использовано в качестве значения по умолчанию для параметра, если не будет найден специфичный для спота параметр с таким же именем (см. выше раздел "[Определение параметров спота](#)").

Чтобы использовать этот новый добавленный параметр вместо жестко заданного названия дисплей-спота, выберите его, как показано выше, затем нажмите кнопку SELECT. В результате в выпадающем меню появится [display] вместо фиксированного имени дисплея. Поскольку этот параметр преобразуется в то же имя (здесь "Display1"), все работает одинаково.

Однако, когда вы отображаете блок, содержащий эту кнопку, на iPad-споте, значение этого параметра можно переопределить для каждого iPad с помощью аналогичного списка параметров, который находится в настройках спота (см. раздел "[Определение параметров спота](#)" выше). Каждый iPad будет управлять своим спотом, даже если все они отображают одну и ту же панель управления, когда вы переопределите параметр display каждого iPad с нужным именем цели.

Binding < 1 > + -

Action

Set Property

Spot

Select or Type Name

[display]

block

Этот же метод можно использовать для параметризации любого типа привязки свойств, включая сетевые устройства, кластеры WATCHOUT, реле и т. д. Следует помнить, что значение, присвоенное параметру на споте, имеет приоритет перед формальным значением, указанным для параметра с тем же именем в блоке. Вы должны убедиться, что полученное значение параметра имеет смысл для вашей системы, и обновить его соответствующим образом, если вы переименовываете объекты. Если подходящего объекта не найдено, кнопка не будет иметь никакого эффекта, и будет записано сообщение об ошибке, указывающее на эту проблему.

9. PIXILAB PLAYER

[PIXILAB Player](#) – это бесплатное программное обеспечение, которое вы можете установить на Intel NUC или аналогичный компьютер для использования в качестве дисплей-спота. В дополнение к базовой функциональности браузеров, оно обладает расширенными функциями, такими как:

- Простая настройка звука и [multiple displays](#), включая разрешение дисплея, частоту обновления и поворот.
- Возможность удаленного выключения и включения дисплей-спота, а также любого подключенного дисплея.
- Загрузка с USB-накопителя или непосредственно с сервера Blocks по сети - SSD/HD не требуется.
- Поддержка [Block Caching](#), что позволяет использовать плеер в автономном режиме и сократить объем трафика данных.

При включенной функции *Block Caching* блок и все его медиафайлы по умолчанию хранятся локально на плеере. Эта функция требует дополнительного носителя, например SD-карты, и лицензируется отдельно ([licensed separately](#)).

10. СБОР ДАННЫХ О ПОСЕТИТЕЛЯХ

Blocks версии 6 имеют возможность [сбора данных о посетителях](#) на основе информации о местоположении, предоставляемой блоком [Locator](#) block, а также других форм сбора данных.. Эта функция, которая лицензируется отдельно, добавляет следующие возможности:

- Идентификация отдельных посетителей либо по их мобильному устройству, либо по какому-либо маркеру (например, RFID-браслету), либо по комбинации того и другого.
- Использование мобильных устройств и гостевого спота ([Visitor Spot](#)) позволяет взаимодействовать с каждым посетителем, предоставляя индивидуальный опыт с детальным контролем над тем, что появляется на мобильном устройстве гостя.
- Посетитель может ввести информацию или предпочтения – например, имя, любимый цвет, возраст и т.д. - через свое мобильное устройство, через регистрационные киоски или другие точки взаимодействия.
- При посещении определенного места эта информация может быть собрана либо на основе сканирования посетителем "жетона" (token), либо с помощью блока [Locator](#) на мобильном устройстве.
- Блок [Camera](#) может собирать фотографии, сделанные посетителем (или фото посетителя), с последующим показом таких фотографий во время его экскурсии.
- Сбор индивидуальных баллов на игровых станциях или аналогичных устройствах.
- Представление информации о посетителях, оценках, фотографиях и т.п. при посещении соответствующих станций.
- Агрегировать информацию от нескольких посетителей, представляя результат в виде "high scores" (высоких баллов) или других форм коллективных данных.

Эти возможности основаны на следующих ключевых особенностях:

- Усовершенствованный Visitor Spot, позволяющий программно взаимодействовать с каждым посетителем. Это обеспечивает возможности сценариев, которые ранее были доступны только для дисплей-спотов, например, возможность представления персонализированного контента каждому посетителю.
- Встроенная высокопроизводительная база данных, управляемая Blocks, собирающая и регистрирующая все взаимодействия с посетителями в виде отдельных записей.
- Пользовательские сценарии позволяют создавать, использовать и управлять жизненным циклом записей данных о посетителях, а также взаимодействовать с отдельными гостевыми спотами.

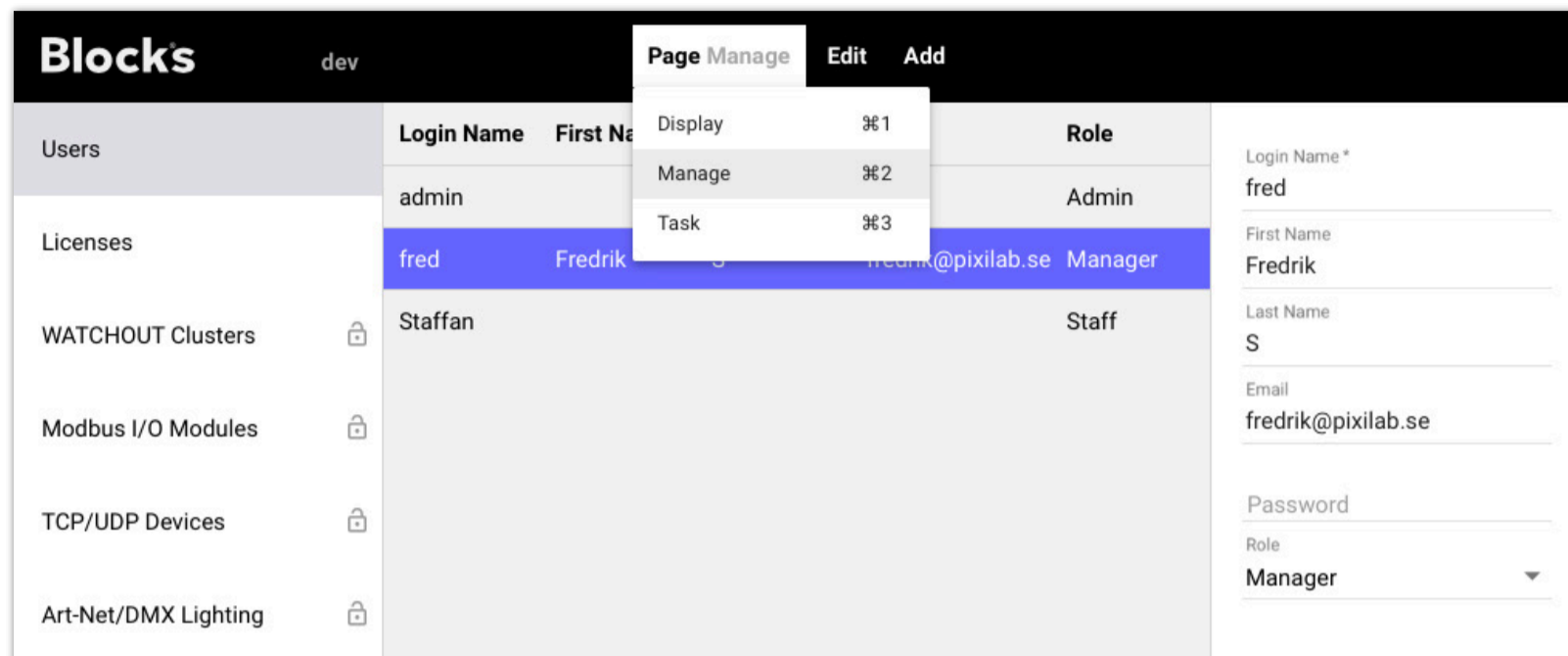
Пользовательский сценарий определяет схему базы данных, используемую для сбора информации о посетителях, и отвечает за создание записи, обновление и использование ее содержимого во время посещения и, наконец, за удаление записи. Этот механизм специально разработан для управления данными в течение одного посещения, при этом данные удаляются (или архивируются) после ухода посетителя или в конце дня. Он не предназначен для долгосрочного хранения данных. (Существует отдельный Blocks API для работы с внешними базами данных ([SQL databases](#)), если вам это необходимо).

О конфиденциальности

Поскольку данный механизм потенциально собирает [персонально идентифицируемую информацию](#), необходимо принять во внимание правила, действующие в вашей юрисдикции (например, европейский GDPR). Если такие данные собираются, вам, возможно, придется проинформировать об этом своих посетителей и возможно, получить их разрешение на это. Тот факт, что данные могут быть удалены, когда посетители уходят (или в конце дня), смягчает некоторые потенциальные трудности (например, "право на забвение" – [right to be forgotten](#)).

11. УПРАВЛЕНИЕ

Страница Manage (Управление) используется для добавления и настройки различных объектов, таких как пользователи, кластеры WATCHOUT, устройства ввода/вывода и сетевые устройства, и многое другое. Вы получаете доступ к этой странице с помощью команды Manage в строке меню. Затем выберите область для управления из списка, представленного слева.



Users (пользователи)

Система содержит по крайней мере одного пользователя, всегда называемого "admin". Этот пользователь не может быть удален или переименован. Вы можете (и должны) изменить пароль пользователя "admin". Вы также можете задать его имя/фамилию и адрес электронной почты.

Пользователи могут быть добавлены через меню "Add". Вы должны указать имя (Login Name) для входа в систему, а также пароль. Вы можете указать дополнительную информацию для дальнейшей идентификации пользователя. Роль определяет функции пользователя, при этом роль "Admin" имеет полный доступ ко всем областям Blocks (включая добавление пользователей и установку их ролей). Подробнее о ролях смотрите в разделе "[Авторизация и роли](#)".

- ◆ Если ваша система Blocks настроена на использование внешнего провайдера единой регистрации (SSO), например, Active Directory, любые пользователи, определенные в разделе Manage, игнорируются. В этом случае обратитесь к своему системному администратору за разъяснениями.

Лицензии

Содержание вашей лицензии Blocks показано в разделе "Лицензии". Управление лицензией осуществляется одним из следующих способов:

- С помощью аппаратного лицензионного ключа, подключенного к USB-порту компьютера, выполняющего роль сервера Blocks. В качестве альтернативы, USB может быть подключен к другому компьютеру, доступному по сети с вашего сервера Blocks.
- Использование облачной лицензии. Это особенно подходит для запуска Blocks в среде, где физический USB-порт недоступен, например, на виртуальном частном сервере (VPS) или в другой виртуализированной среде. Этот метод требует, чтобы ваш сервер Blocks имел доступ в интернет.

Лицензируемые опции

Слоты лицензируются отдельно, как и некоторые премиальные типы блоков. Для каждой лицензируемой функции вы можете увидеть:

- Количество слотов или других учитываемых функций, которые вы лицензировали.
- Количество, которое в настоящее время используется или настроено в системе.

- Количество спотов, активно используемых в данный момент.
- Для премиум-блоков - является ли тип блока лицензионным. Если да, то вы можете использовать неограниченное количество таких блоков.

Существуют две недорогие лицензии для специальных целей, называемые *Producer* и *Evaluation*, с дополнительными ограничениями:

- Лицензия *Producer*. *Blocks* автоматически отключаются через четыре часа, после чего систему необходимо перезапустить вручную.
- Лицензия *Evaluation*. Те же ограничения, что и в лицензии *Producer*, плюс ограничение срока (обычно 30 дней с момента первого использования).

Block-Caching Players (плееры с кешированием)

Лицензия *Block-Caching Players* контролирует количество плееров, для которых можно включить [Block Caching](#). Эта лицензия дополняет лицензию дисплей-спота и применяется только к функции *Block Caching*. Плееры, для которых вы включили эту функцию, отслеживаются в файле *Licenses-XXXX*, расположенном в каталоге *model* корня вашего *Blocks*, поскольку такие права не могут быть отозваны. Не переименовывайте, не удаляйте и не делитесь этим файлом между серверами или корнями *Blocks*.

Сбор данных о посетителях (Visitor Data Collection)

Эта дополнительная лицензия позволяет отслеживать [пути отдельных посетителей](#), собирать данные во время посещения и т. д..

- Для гостевых спотов она активирует настройки, расположенные на вкладке *Identity*, позволяя вам связать тип записи данных с посетителями.
- Пользовательские сценарии получают возможность создавать и использовать записи данных (*data records*), например, связанные с гостевыми спотами или другими формами идентификации, такими как RFID-метки.

Обновление лицензии

Ваша лицензия *Blocks* может быть обновлена путем заказа дополнительных лицензируемых элементов. Следуйте инструкциям, приведенным справа на странице Лицензии.

Поддержка серверов-зеркал (Mirroring Service)

Эта страница доступна, если ваша лицензия включает опцию [Mirroring Service](#). На ней отображается статус зеркал, например, подключен ли сервер-зеркало, когда была выполнена последняя синхронизация и т.д.

Состояние сервера (Server Status)

Показывает основные параметры сервера *Blocks*, например, объем свободной памяти и дискового пространства. Значения, отображаемые на этой странице, обновляются автоматически примерно два раза в минуту.

- *Free disk* показывает объем свободного дискового пространства на томе сервера, содержащем данные *Blocks*. Смотрите за этими значениями, особенно перед загрузкой новых медиафайлов, чтобы убедиться, что свободного места достаточно. Если это значение опускается ниже 10 ГБ или около того, вам нужно больше места на диске.
- *Free memory* (свободная память) указывает на количество неиспользуемой памяти RAM в серверной программе. Это нормальное явление, когда это значение сильно колеблется, так как память управляется динамически во время работы сервера. Чтобы увидеть максимальное количество памяти, доступное в данный момент, нажмите кнопку "Free Unused Memory". Если это значение опустится ниже 10 мегабайт или около того, возможно, вашему серверу не хватает памяти. Если на вашем сервере достаточно оперативной памяти, выделенной для *Blocks*, можно увеличить, изменив настройки сервера. За подробностями обращайтесь на support@pixilab.se.
- *Max backlog* (максимальное отставание) указывает на загруженность сервера. Если это значение превышает 200 или около того, вам может понадобиться более быстрый сервер. Вы можете уменьшить нагрузку, оптимизировав какой-либо скрипт, задачу или настройки сервера.
- *Started* показывает дату и время последнего запуска сервера, давая представление о том, как долго он работает.

Модули ввода/вывода Modbus

[Modbus](#) – это промышленный стандартный протокол для связи с такими устройствами, как датчики и реле. Blocks поддерживает версию протокола "Modbus TCP", позволяя вам подключать такие устройства через сеть Ethernet.

Чтобы получить доступ к модулю I/O Modbus из Blocks, вы должны присвоить ему известный IP-адрес. Пожалуйста, обратитесь к документации производителя.

- ◆ Протокол Modbus TCP не имеет встроенной защиты. Не используйте модули I/O Modbus в открытой сети. Если вы хотите иметь публичную сеть Wi-Fi для функций мобильного гида, вы должны отделить эту сеть от сети управления таким образом, чтобы гости могли получить доступ только к серверу Blocks. Это можно сделать, например, с помощью правильно настроенного маршрутизатора и VLAN.

Module	IP Address	Enabled	Notes
Moxa4	10.0.1.113	NO	

Channel	Direction	Type	Alias
1	Out	Digital	03-B-FixaFelet Nasan
10001	In	Digital	03-B-FixaFelet Knapp1
10002	In	Digital	03-B-FixaFelet Knapp2
10003	In	Digital	03-B-FixaFelet Knapp3
30018	In	Analog	Counter1

Configuration panel for the selected module (Moxa4):

- Module Name: Moxa4
- IP Address: 10.0.1.113
- Enable Module
- Modbus Addressing: Classic (Modicon style)
- Modbus Channel Number: 30018
- Alias (under "IO"): Counter1
- Signal Type: Analog 16-bit word
- Output Input
- Poll Rate: Slow

Настройки модуля Modbus

Когда устройство Modbus правильно настроено и имеет известный IP-адрес, добавьте его в список с помощью команды "Modbus Module" в меню Add. Дайте ему имя и укажите его IP-адрес. Заданное имя может быть использовано для доступа и программирования каналов ввода/вывода.

Вы можете временно отключить модуль Modbus, сняв флажок "Enable Module". В этом режиме никакие команды не будут отправляться ни на один из выходных каналов этого модуля.

Красный индикатор слева от названия модуля в списке станет зеленым, когда сервер Blocks подключится к модулю.

- ◆ Протокол Modbus TCP предусматривает регулярное отключение от сервера, после чего индикатор временно становится красным. Это нормально и не должно мешать работе устройства, так как Blocks переподключается по мере необходимости.

Каналы Modbus

Большинство модулей ввода/вывода Modbus оснащены винтовыми клеммами для подключения датчиков или устройств, которыми необходимо управлять. Эти клеммы имеют номер канала. Иногда номер канала указывается рядом с винтовой клеммой. Если это не так, то в документации модуля должно быть указано соответствие между номерами каналов и винтовыми клеммами.

- ◆ Некоторые модули Modbus имеют настраиваемые номера каналов, в этом случае вам может потребоваться задать их при конфигурировании модуля.

Blocks поддерживают две формы адресов Modbus:

- Классическая (Modicon style) адресация, при которой выходы начинаются с 1, а входы с 10000, как показано в примере выше.
- Современная (полная 16-битная), где любой числовой адрес может быть свободно назначен любой функции.

Чтобы добавить канал к модулю, выберите модуль в списке, затем выберите "I/O Channel" в меню Add. Настройте параметры канала соответствующим образом.

Modbus Channel Number (номер канала Modbus)

Введите здесь номер канала, указанный в документации модуля (или настроенный вами, если номера каналов являются конфигурируемыми).

Alias (псевдоним)

Доступ к каналу можно получить под именем модуля Modbus (например, "chN", где N - номер канала), но можно также обращаться к функции по описательному имени. На рисунке выше каналу 1 присвоен псевдоним "cleaningLights". Использование этого метода упрощает задачу, если вам нужно переместить функцию на другую винтовую клемму или даже на другой модуль Modbus, так как в этом случае вы можете просто переместить псевдоним, и все ваши программы продолжают работать.

Если вы присвоите псевдоним каналу, вы сможете получить доступ к этому каналу для целей программирования: "IO.YYY", где YYY - псевдоним, присвоенный каналу.

Signal Type (тип сигнала)

Определяет, является ли сигнал "цифровым" (с состояниями вкл./выкл., например, реле или кнопка) или "аналоговым" (например, датчик температуры). Кроме того, аналоговые сигналы могут быть 16- или 32-битными. Обратитесь к документации на используемый модуль. Если это не указано, можно считать, что точность составляет 16 бит.

Signal Direction (направление сигнала)

Определяет, является ли канал выходом (для управления предметами, например, реле) или входом (например, замыкание контактов или датчик температуры).

Poll Rate (скорость опроса)

Modbus требует опроса для получения входных данных. Для входов, которые изменяются редко или медленно, оставьте для параметра Poll Rate (скорость опроса) значение Slow (медленно). Если входное значение может меняться очень быстро, и вам нужно получить эти изменения быстрее, выберите Fast. Не выбирайте Fast, если это не требуется, так как это увеличивает нагрузку на систему

- ◆ Некоторые модули Modbus фильтруют входные данные. Если это так, убедитесь, что значение фильтра достаточно низкое для самой высокой ожидаемой частоты входных данных. Например, многие модули MOXA Modbus имеют время фильтрации по умолчанию 100 мс, поэтому входные данные, изменяющиеся быстрее 5 Гц, могут быть подавлены. В этом случае простая установка скорости опроса на *Fast* не поможет, поскольку сигнал уже отфильтрован модулем. Пожалуйста, ознакомьтесь с документацией модуля для получения подробной информации..

Сетевые устройства

Многими устройствами можно управлять по локальной сети с помощью стандартных IP-протоколов, таких как TCP или UDP. В некоторых случаях достаточно послать устройству строку символов. Такое устройство можно контролировать с помощью Blocks, добавив его в раздел TCP/UDP Devices на странице Manage (Управление) и указав его IP-адрес и номер порта. IP-адрес должен быть установлен на устройстве; обычно это делается через некоторое меню конфигурации. Номер порта часто фиксирован и должен быть указан в документации устройства.

The screenshot shows the 'Blocks' management interface. At the top, there are buttons for 'Page Manage', 'Edit', and 'Add'. Below is a table with columns: Device, IP Address, Port, Driver, and Enabled. The 'Samsung' device is selected and highlighted in blue. To the right of the table is a configuration form for the selected device, including fields for Device Name, IP Address, Driver, Port Number, and a Note section. At the bottom of the form, there are radio buttons for 'TCP' (selected) and 'UDP'.

	Device	IP Address	Port	Driver	Enabled
Users	• NEC	10.0.1.228	4352	PJLinkPlus	No
Licenses	RcvBytes	10.2.0.12	4445	UDP_Bytes_Input	Yes
WATCHOUT Clusters	• Samsung	192.168.0.102	1515	SamsungMDC	Yes
Modbus I/O Modules					
TCP/UDP Devices					
Art-Net/DMX Lighting					

Device Name *
Samsung

Enabled

IP Address *
192.168.0.102

Driver
SamsungMDC

Port Number *
1515

Note

Communications Mode

TCP UDP

Чтобы добавить устройство, выберите “Network Device” в меню Add. Затем укажите его имя, IP-адрес и номер порта. Имя будет использоваться для программирования устройства, поэтому используйте короткое, но описательное имя. Если в документации не указано, следует ли использовать режим TCP или UDP, выберите TCP, так как он является наиболее распространенным.

При использовании режима связи TCP Blocks автоматически попытается подключиться к устройству. Успешное подключение меняет цвет индикатора состояния в списке на зеленый. UDP не поддерживает соединения и поэтому не имеет индикатора.

Драйверы сетевых устройств

Некоторые устройства имеют более сложные протоколы или требуют специализированного подтверждения или других действий для установления надежной связи. Такие устройства трудно надежно контролировать, просто посылая строки команд. Кроме того, синтаксис, используемый некоторыми протоколами, нелегко понять и правильно использовать.

Для преодоления таких проблем могут быть предусмотрены специализированные драйверы. Драйвер - это часть программного обеспечения, которая обладает подробными знаниями о протоколе связи и знает, как надежно взаимодействовать с устройством. Драйверы устройств доступны для стандартных протоколов, таких как протоколы для видеопрокторов (например, Barco rw392 и PJ-LINK, используемые многими производителями прокторов).

Дополнительным преимуществом драйвера является то, что он может раскрывать функционал более высокого уровня, чем основной протокол связи. Например, вам может понадобиться свойство "power" для включения/выключения проектора, или, возможно, свойство "input" для переключения видеовходов. Драйвер обычно раскрывает такие свойства, а затем связывает их с кнопками и задачами для прямого управления.

После установки драйвера устройства на сервер его можно выбрать в меню “Driver” на панели настроек. Новые драйверы разрабатываются компанией PIXILAB и ее партнерами по мере возникновения потребностей. Свяжитесь с PIXILAB для получения последней информации о доступных драйверах или проверьте на странице [github page](#) компании PIXILAB, где мы храним большинство драйверов.

Пользовательские опции

Произвольные данные, доступные выбранному драйверу устройства. Могут использоваться для специфических для драйвера пользовательских настроек конфигурации, часто с использованием синтаксиса JSON. Подробности см. в документации к драйверу.

Свет: Art-Net/DMX-512

Театральные осветительные приборы обычно управляются с помощью протокола стандарта [DMX512](#). Некоторые другие системы освещения имеют "мосты", которые позволяют управлять ими по протоколу DMX512. DMX512 по своей сути является простым протоколом, позволяющим установить или изменить значения до 512 каналов. Их можно использовать для прямого управления интенсивностью большого количества световых приборов. Каналы могут быть объединены различными хитроумными способами для управления сложными световыми приборами, такими как "движущиеся головы" с функцией панорамирования/наклона, а также множеством других эффектов и функций.

DMX512 использует аппаратную проводку световых приборов, но управляющие данные DMX-512 часто передаются по Ethernet, где они могут быть использованы совместно с другими функциями. Этот метод основан на протоколе [Art-Net](#) для "туннелирования" DMX-данных через обычную сеть. Сеть Ethernet подключается к диммерам и другим устройствам, напрямую совместимым со стандартом Art-Net. Можно также использовать мостовое соединение по адаптеру Ethernet-to-DMX.

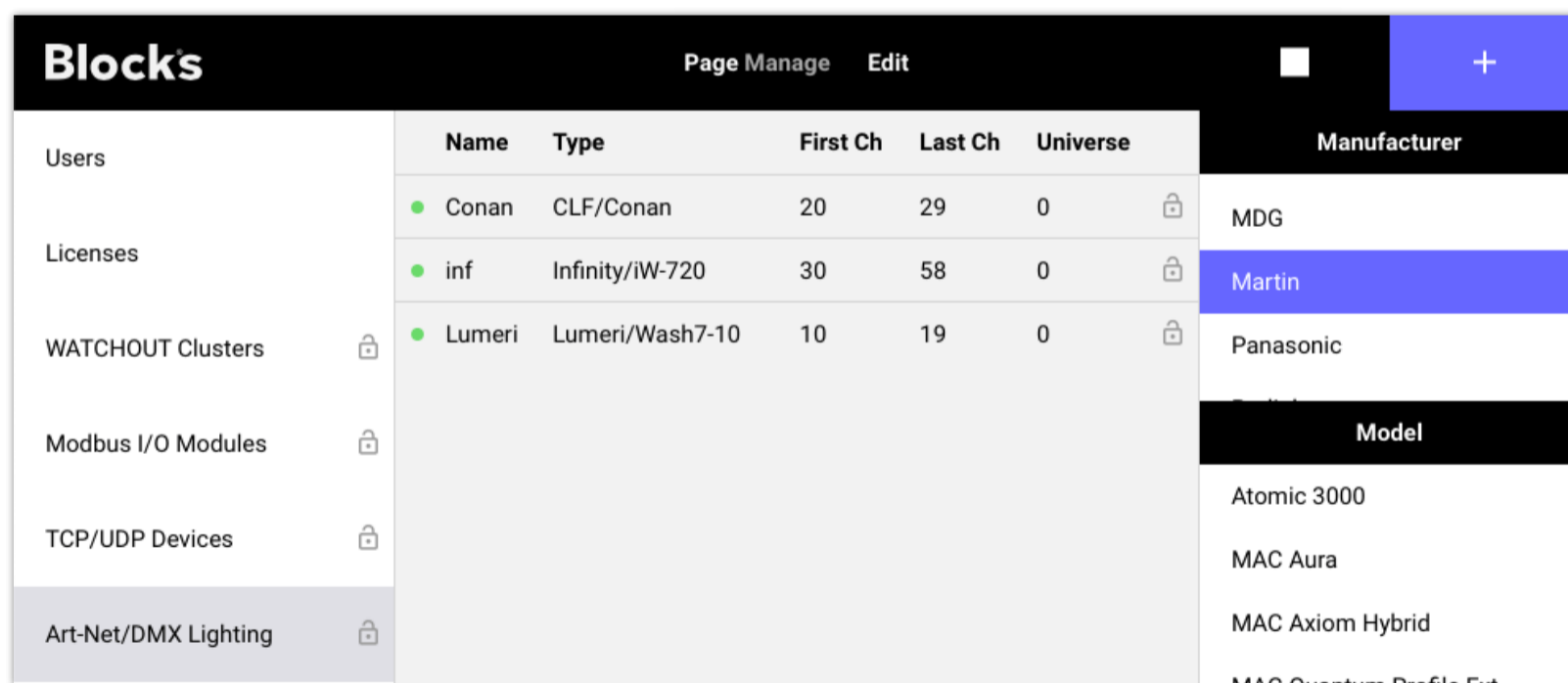
Blocks могут управлять светильниками и диммерами с помощью протокола Art-Net: либо интерактивно – кнопками и слайдерами, либо программно с помощью Задач. Для светильников с разъемами DMX512 XLR необходим адаптер Ethernet-to-DMX, как упоминалось выше.

Добавление светового прибора

Чтобы добавить прибор для управления, сделайте следующее:

1. Выберите "Art-Net/DMX lighting" на странице "Manage", как показано ниже.
2. Нажмите кнопку "+" в правом верхнем углу.
3. Перетащите модель прибора из списка Model в основную область окна.

Blocks поставляется с набором описаний типовых приборов (от компании "Generic"). Дополнительные описания приборов можно найти на странице [PIXILAB github](#) вместе с инструкциями по их установке, а также по созданию собственных, если это необходимо.

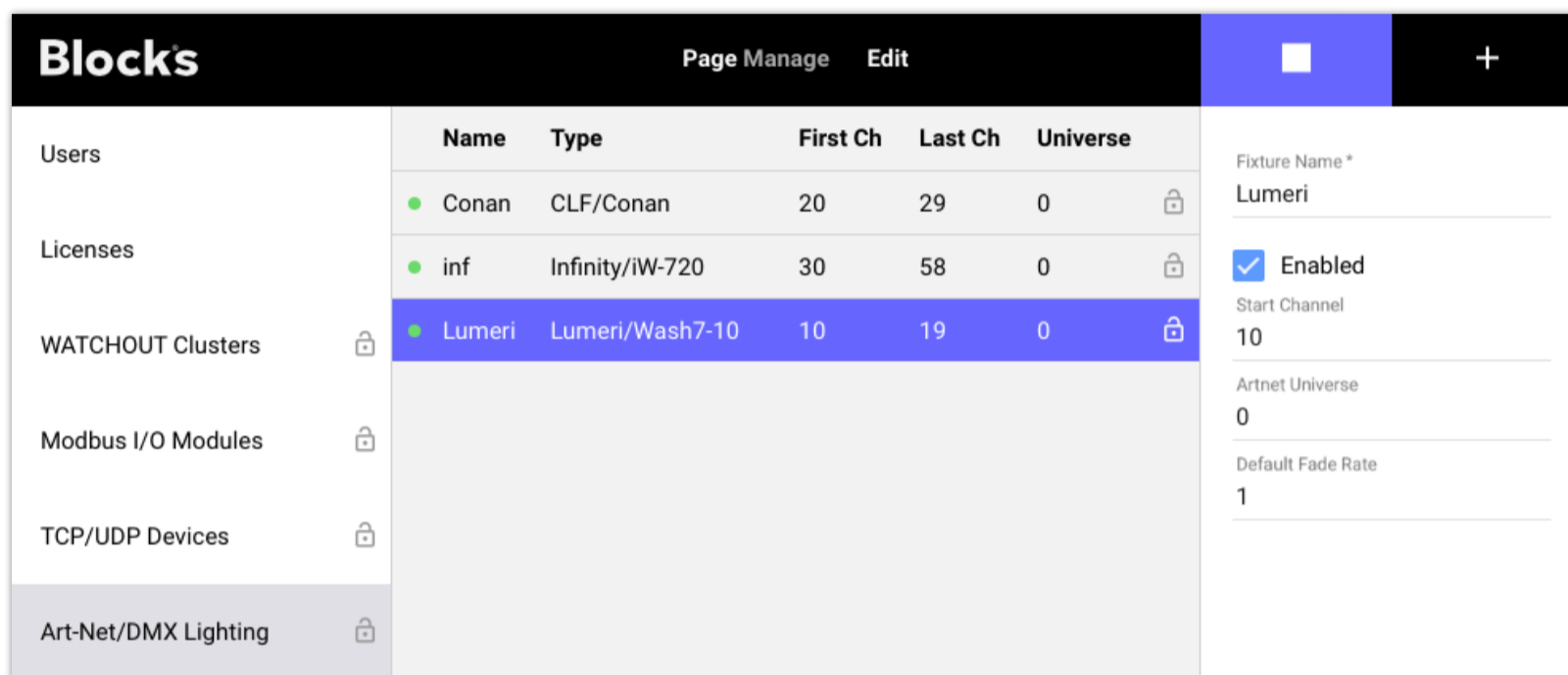


Для базового, одноканального управления выберите "Desk Channel" в разделе Model после выбора "Generic" в разделе Manufacturer. Перетащите "Desk Channel" в основную область окна, чтобы добавить этот канал. Несколько других базовых описаний можно найти в разделе "Generic", например, "RGBD Fader" для управления цветовой подсветкой с помощью каналов Red, Green Blue и Dimmer, или "Pan Tilt 16-bit" для управления световой головой с двумя каналами высокого разрешения для панорамирования и наклона.

Настройки светового прибора

После добавления описания прибора нажмите на квадрат в правом верхнем углу, чтобы настроить его параметры, как показано ниже. Введите название описывающее то, что светильник освещает, или другое, имеющее для вас смысл, поскольку это имя будет использоваться для программирования прибора. Укажите стартовый канал (номер канала DMX512), соответствующий настройке базового канала прибора. Если используется более одного канала, дополнительные каналы отображаются последовательно за первым каналом. Вы можете увидеть диапазон каналов, используемых устройством, в списке, в столбцах "First Ch" и "Last Ch". Количество каналов, используемых прибором, определено в описании прибора.

Простое устройство "Generic Desk Channel" использует только один 8-битный канал, о чем свидетельствуют одинаковые цифры в столбцах "First Ch" и "Last Ch". Выбранное ниже приспособление "Lumeri" использует десять каналов с номерами от 10 до 19, причем 10 является базовым каналом.



	Name	Type	First Ch	Last Ch	Universe	
Users	Conan	CLF/Conan	20	29	0	🔒
Licenses	inf	Infinity/iW-720	30	58	0	🔒
WATCHOUT Clusters	Lumeri	Lumeri/Wash7-10	10	19	0	🔒
Modbus I/O Modules						🔒
TCP/UDP Devices						🔒
Art-Net/DMX Lighting						🔒

Fixture Name *
Lumeri

Enabled

Start Channel
10

Artnet Universe
0

Default Fade Rate
1

Artnet Universe (вселенная Artnet)

Благодаря высокой пропускной способности Art-Net/Ethernet может туннелировать несколько шин DMX512 по одному проводу Ethernet в отличие от более медленной шины DMX512. В протоколе Art-Net они называются "universes" (вселенные). Вы указываете, какую вселенную Art-Net использовать для прибора, используя настройку "Artnet Universe", которая должна совпадать с соответствующей настройкой на принимающей стороне.

- ◆ В то время как номер канала отсчитывается от 1, юниверс отсчитывается от 0. Некоторые устройства могут иметь другое соглашение, что приводит к проблемам «отдельности». Поэтому если ваше устройство не отвечает, вы можете попробовать одну вселенную выше / ниже числа, установленного в Blocks, чтобы увидеть, заставит ли это работать.

Скорость затухания по умолчанию (Default Fade Rate)

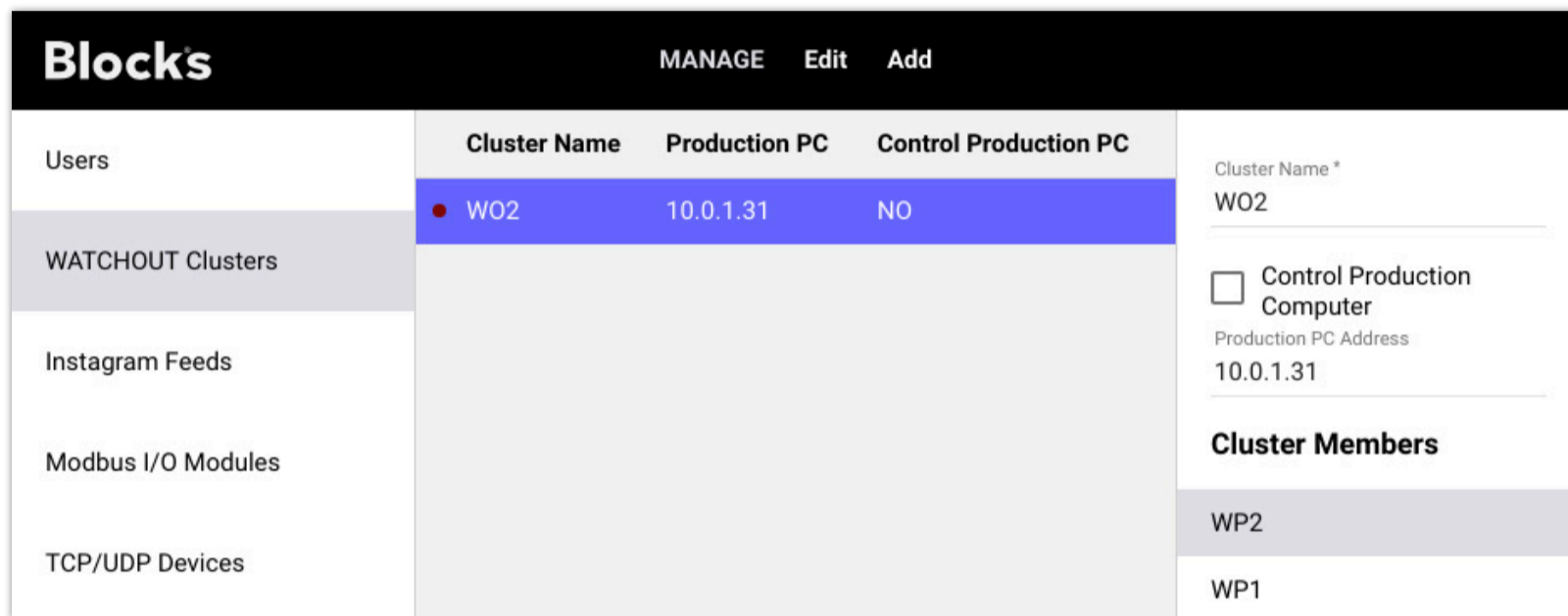
Этот параметр определяет, насколько быстро канал достигнет целевого значения при управлении с помощью кнопки. При управлении каналом из задачи, вы можете явно указать время затухания в каждом операторе задачи, не полагаясь на этот параметр.

Установите это значение на 0, чтобы достигнуть целевого значения сразу при нажатии кнопки, а не использовать постепенное затухание.

Кластеры WATCHOUT

Добавьте и настройте любое количество кластеров WATCHOUT. WATCHOUT иногда используется вместе с Blocks для мультискранный проекции или системы дисплеев. Кластер WATCHOUT состоит из одного или нескольких дисплей-компьютеров и (необязательно) продакшн-компьютера. Продакшн-компьютер обычно используется только во время подготовки шоу и затем может быть удалён. Однако в некоторых случаях он остается частью системы, чтобы быстро вносить необходимые изменения.

Добавьте кластер дисплеев с помощью меню “Add” (добавить) после выбора “WATCHOUT Clusters” в списке слева. Если вы планируете сохранить продакшн-компьютер как часть системы, введите его IP-адрес в поле “Production PC Address”. В противном случае оставьте это поле пустым. Переключение между управлением продакшн-компьютером и кластером дисплей-компьютеров происходит через установку флажка “Control Production Computer” (управление продакшн-компьютером).

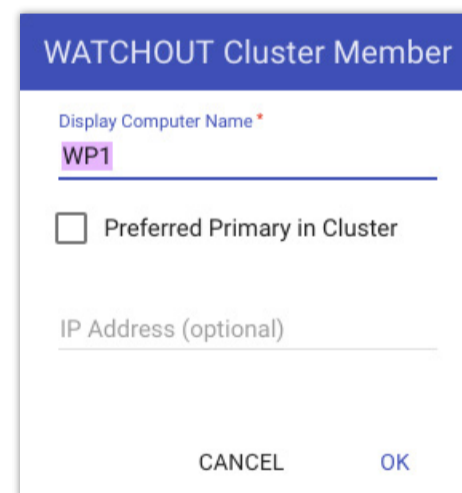


Имя, которое вы присваиваете кластеру, должно соответствовать имени кластера (“Cluster Name”), указанному в диалоговом окне «Настройки» (Preferences) в программе WATCHOUT. Использование одинаковых имен кластера и имен дисплей компьютеров (см. далее) позволяет Blocks найти кластер в сети без указания IP-адресов дисплей компьютеров.

Красная точка рядом с именем кластера в списке становится зеленой при успешном соединении с кластером. Сервер Blocks автоматически повторяет попытки соединения, пока оно не будет успешно установлено.

Добавьте дисплей-компьютеры в кластер командой “Add Member” (добавить участника) в меню “Add”. Если вы управляете продакшн-компьютером, такая команда не является необходимой, так как всё управление в любом случае будет проходить через продакшн-компьютер. Но если вы позже захотите удалить продакшн-компьютер и напрямую управлять дисплей-компьютерами, вы должны добавить и их, чтобы поддержать такой режим работы. При добавлении дисплей-компьютеров вы должны дать им имена. Эти имена должны соответствовать именам, заданным для дисплей-компьютеров в WATCHOUT.

Можно указать IP-адрес для каждого дисплей-компьютера. Примените этот метод, если вы используете фиксированные IP-адреса для дисплей-компьютеров в шоу WATCHOUT. Можно назначить один дисплей-компьютер первичным (“primary”) в кластере, он действует как мастер внутри своего кластера. Обычно неважно, какой компьютер становится мастером, и в этом случае можно не устанавливать флажок на эту опцию.



Управление WATCHOUT

После добавления в Blocks вы можете управлять WATCHOUT посредством кнопок, движков и задач. Вы также можете использовать события WATCHOUT, например, достижение конца временной шкалы, для переключения задач.

12. ЗАДАЧА (TASK)

Базовые функции управления могут быть выполнены привязкой кнопок и других элементов управления к контролируемым свойствам. Этот метод можно использовать для включения питания проектора при помощи кнопки или для задания уровня громкости при помощи движка. Однако в других случаях требуется одновременное управление несколькими функциями - например, включение нескольких проекторов и задание уровня освещённости в помещении - или выполнение команд в определённой последовательности. Такие более сложные сценарии управления могут быть выполнены с использованием задач (tasks).

The screenshot displays the 'Blocks' application interface. At the top, there's a header with 'BLOCKS', 'Edit', and 'Realm Main'. Below the header, a 'Trigger Condition' is set to 'now.getHours() < 17'. The main area is divided into three sections: 'Group', 'Task', and 'Realm Variable'. The 'Task' section is active, showing a sequence of steps: 1. 'do Network.Barco.power' with 'set to TheaterShow.value'; 2. 'wait 1 seconds'; 3. 'do Network.Barco2.power' with 'set to TheaterShow.value'; 4. 'if TheaterShow.value'; 5. 'do WATCHOUT.WO2.wakeUp' with 'timeout? Literal value or expression'; 6. 'do WATCHOUT.WO2.load' with 'showName "New Wine"'; 7. 'do WATCHOUT.WO2.play' with 'timeline Literal value or expression'; 8. 'end if'; 9. 'do IO.DALI_5.value' with 'set to !TheaterShow.value'. The 'Realm Variable' section shows 'TheaterShow' set to 'false'. On the right, a configuration panel for the 'StartShow' task is visible, showing it is 'Enabled', triggered by 'Property Change', and watching 'Realm.Main.variable.Theater Show.value'.

Задача, которая включает и выключает проекторы и проигрывает презентацию.

Сервер Blocks может обрабатывать несколько задач одновременно. Задачи организованы в области задач (Realms), а затем - в группы (Groups). Задачи могут связываться со спотами (Spots) и другими устройствами, добавленными на странице Manage (такими как модули ввода/вывода и сетевые устройства). Задачи могут также общаться друг с другом, вызывая друг друга или публикуя заметки, известные как переменные области *Realm Variables*, в «почтовых ящиках» (mailboxes).

- ◆ Задачи справляются с большинством сценариев управления, но Blocks поддерживает и расширенные сценарии, основанные на более формальном языке программирования. Подробнее об этом см ["Расширенные сценарии и драйверы устройств"](#).

Область (Realm)

На верхнем уровне Задачи организованы в области (realms). Как правило, вам нужна только одна область. Используйте несколько областей в более крупных системах, где нужно управлять несколькими более или менее независимыми областями, например, разными комнатами в здании. В каждой комнате может быть много вещей, которые можно

контролировать – часто во взаимосвязанном режиме, а отдельные комнаты могут иметь мало общего или вообще не иметь ничего общего друг с другом. В этом случае имеет смысл управлять каждой комнатой как отдельной областью.

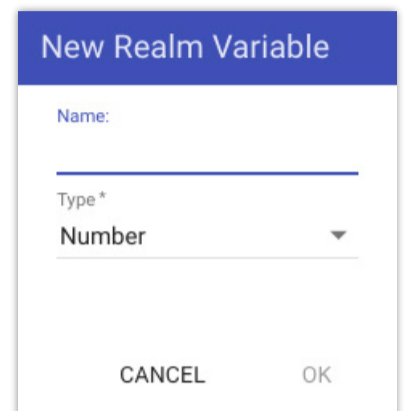
Выберите текущую область, используя меню Область (Realm) в строке меню. Это меню также используется для добавления и удаления областей. Когда вы выбираете область в этом меню, списки Группа (Group), Задача (Task) и Переменная (Variable) изменяются для отображения этих объектов в данной области.

Переменные области (Realm Variables)

В дополнение к группам и содержащимся в них задачам область может также содержать общие переменные. Эти переменные напрямую доступны для всех задач этой области. Они могут быть использованы для передачи состояния системы между задачами. Они также могут быть связаны с кнопками и ползунками, обеспечивая интерактивное управление системой.

Чтобы добавить переменную области, нажмите на знак плюс в заголовке списка. Дайте переменной имя и укажите её тип. Имя переменной должно начинаться с буквы и может содержать только буквы, цифры и символ подчеркивания. Имя не может содержать пробелов.

Выберите имя со смыслом, указав, для чего предназначена эта переменная. Это имя будет использовано для доступа к переменной из задач, а также для подключения к кнопкам и другим элементам панели управления.



Типы переменных

Переменная имеет фиксированный тип, определяющий, какие данные могут быть сохранены в ней, а также как она взаимодействует с кнопками и другими элементами управления.

- Number (число). Содержит числовое значение, которое может иметь дробную часть и может быть отрицательным.
- String (строка). Содержит произвольные символы. Строки буквенных символов (литеральные строки) должны быть заключены в одинарные или двойные кавычки.
- Boolean (логические). Значение является либо истинным (true), либо ложным (false).

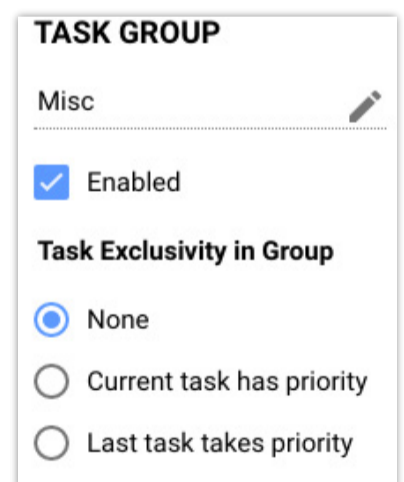
После создания переменную можно дополнительно настроить в панели параметров. Вы можете указать значение по умолчанию, которое переменная будет принимать при запуске системы. Для числовой переменной можно указать ее минимальное и максимальное допустимое значение, а также может ли она содержать дроби или должна быть целым числом. Эти ограничения будут соблюдаться при установке переменной через элемент управления панели или задачу. Если оставить минимальное и максимальное значение равным нулю, ограничения диапазона не будут применяться.

Group

Задачи организуют в группы. Добавьте группу, щёлкнув на знак «плюс» в заголовке группы. Создание групп задач имеет две цели:

- *Организационную*, позволяющий объединить связанные задачи.
- *Функциональную*, для взаимоисключения задач внутри группы.

Выберите группу, чтобы изменить её настройки, (см. справа). Чтобы отключить группу, снимите флажок “Enabled” (включено). Отключенные группы будут выделены курсивом. Их задачи не будут выполняться.



Эксклюзивная задача

Несколько задач могут выполняться одновременно, но это не всегда то, что вам нужно. Бывают группы взаимоисключающих задач, из которых только одна может выполняться в любой момент времени. Например, у вас могут быть задачи, которые подготавливают помещение к различным режимам презентации. В одном режиме проекционный экран должен быть развёрнут, а в другом он должен быть скрыт. Может потребоваться некоторое время для переключения режимов, чтобы экраны успели развернуться, а проекторы “проснулись”. Можно учитывать эти задержки с помощью операторов ожидания “wait” или предупредить задержки, вызванные некоторыми длительными действиями, отдавая команды в правильном порядке и в соответствующее время.

Пока потенциально продолжительная задача выполняется, вы можете изменить свое решение и нажмёте другую кнопку, чтобы выбрать другой режим, в этом случае всё может выйти из-под контроля, так как теперь у вас есть две задачи, которые будут конкурировать друг с другом, пытаясь установить свой режим в помещении.

Эксклюзивность задач может помочь избежать таких хаотичных ситуаций, гарантируя, что в любой момент времени может выполняться только одна задача в рамках группы задач. Это делается путем выбора одного из эксклюзивных режимов:

- **Current task has priority** (текущая задача приоритетна). Новая задача в этой группе не может быть запущена, пока выполняется другая задача. Текущая задача должна быть выполнена первой до конца.
- **Last task takes priority** (последняя задача приоритетна). При успешном запуске задачи в этой группе все уже выполняющиеся задачи в группе будут прерваны.
 - ◆ Режим “Last takes priority” иногда имеет неожиданные последствия, так как любая предыдущая задача может быть прервана на полпути. Вы должны убедиться, что задачи в такой группе задают все соответствующие состояния системы недвусмысленно, ясно, без ссылок на другие задачи.

Задача (Task)

Задача – это список операторов с переключателем (триггером). Переключатель управляет выполнением задачи, а операторы внутри задачи определяют, что делать. Как вы видите на рисунке в начале этой главы, операторы занимают основную область, а переключатель отображается справа, когда задача выбрана в списке задач.

Узнайте больше о программировании задач на Blocks wiki: <https://pixilab.se/docs/blocks/tasks>

Создание и наименование задач

Добавьте задачу, щёлкнув на символ плюса в заголовке списка задач. Присвойте задаче описательное имя. Можно использовать любые символы в имени задачи, но иногда программирование задач становится проще, если вы начинаете имя с буквенного символа, а затем используете только символы, цифры и подчеркивания.

Enabled (включено)

Для задачи можно устанавливать флажок Enabled. Это позволяет и отключать задачи. Обратите внимание, что это дополнение к подобному флажку охватывающей группы. Для выполнения задачи должны быть установлены оба флажка. Отключенные задачи отображаются курсивом в списке задач.

Abort on Error (прервать при ошибке)

Если этот флажок установлен, любая ошибка, возникающая при выполнении задачи, приведет к немедленному завершению задачи, никакие следующие операторы не будут выполнены. Будет зарегистрировано сообщение об ошибке с указанием, что пошло не так. Это поведение по умолчанию. Иногда нужно, чтобы задача продолжала выполнять следующие операторы, даже если происходит ошибка. Если такой вариант вас устраивает, снимите этот флажок. Это приведет к регистрации ошибки, но не прервет выполнение задачи.

Triggers (переключатели)

Существует несколько способов запуска задачи:

- Вручную, при нажатии на маленький треугольник рядом с названием задачи. Нажатие Shift позволяет запустить задачу независимо от любого [Trigger Condition](#) (условия переключателя).
- При изменении состояния свойства системы. Это может быть начало воспроизведения на споте, обнаружение движения датчиком или изменение уровня громкости.
- По времени суток, что позволяет запланировать выполнение задач в нужный момент или многократно с некоторым интервалом.
- При запуске сервера: инициируется один раз при запуске сервера.

Запуск задачи в явном виде

В дополнение к этим способам можно запустить задачу, напрямую связав кнопку на панели с «запущенным» (running) состоянием задачи или из другой задачи, используя оператор do (который будет описан далее в этой главе). Можно использовать этот явный метод независимо от режима переключения (Trigger) задачи. Установите для режима переключения значение “None”, если вы хотите, чтобы задача запускалась именно таким образом.

Переключатель изменения свойств

В дополнение к этим способам можно запустить задачу, напрямую связав кнопку на панели с «запущенным» (running) состоянием задачи или из другой задачи, используя оператор do (который будет описан далее в этой главе). Можно использовать этот явный метод независимо от режима переключения (Trigger) задачи. Установите для режима переключения значение “None” (нет), если вы хотите, чтобы задача запускалась именно таким образом.

- ◆ Для присвоения имени "motion1" была использована функция псевдонима ([Alias](#)) канала Modbus. Это делает программирование проще, чем при использовании имени, заданного по умолчанию по номеру канала Modbus (например, “ch10004”).

После выбора свойства, используемого в качестве переключателя, вы можете квалифицировать переключатель, используя опцию “Start when property” (начать, когда свойство...):

- Becomes true (становится истинным), переключает задачу, когда датчик движения регистрирует выходной сигнал датчика перекрыт).
- Becomes false (становится ложным), переключает задачу, когда датчик больше не регистрирует выходной сигнал датчика свободен).
- Changes, (изменения), переключает задачу при изменении значения независимо от направления. Это особенно полезно с числовыми или строковыми значениями и может быть дополнительно квалифицировано с использованием условия переключения (Trigger Condition), что будет описано далее [Trigger Condition](#).

Если для числового свойства выбрано “Becomes true” (становится истинным) или “Becomes false” (становится ложным), значение 0 считается ложным, а любое другое значение считается истинным. При подключении к строковому свойству пустая строка считается ложной, а непустая строка считается истинной (это применимо, даже если она содержит один пробел).

Переключатель по времени

Чтобы запустить задачу в определённое время, сначала выберите “Time of Day” в меню Trigger, затем укажите время начала и другие ограничения, используя поля, расположенные ниже.

В первом поле Time задают время начала первого запуска задачи. Время следует вводить в 24-часовом формате (часы и минуты).

Затем вы можете указать дату начала; рекомендуем заранее создать задачу, чтобы можно было назначить дату её начала последующим числом. Аналогичным образом можно указать дату окончания..

- ◆ Дата окончания (Ending date) указывает последний день, когда будет запущена задача. Чтобы запустить задачу на один день, задайте эту дату для начала и окончания выполнения задачи.

Задача, запускаемая на сутки, может выполняться один или несколько раз в течение определённого времени. Вы задаёте это, используя меню “Repeat” (повторить):

- None запускает задачу только один раз. Это полезно для режимов работы, которые выполняются только утром или вечером, например, выключение питания.
- At Interval, задача перезапускается в заданном интервале относительно времени начала. Например, если время начала составляет 9:00, а интервал составляет 0:30, он также будет запущен в 9:30, 10:00, 10:30 и т.д. до достижения времени окончания (End Time).
- Continuously. Continuously (непрерывно) Задача будет выполняться многократно, в течение дня, до заданного времени окончания (End Time). Используйте оператор ожидания (**wait**) в задаче, чтобы уменьшить частоту повторения.

Когда вы задаёте повторяющуюся задачу, вы можете указать время её окончания, после которого в этот день задача перестанет выполняться.

Trigger
Property Change

Property to Watch
Custom Path (advanced)
IO.motion1.value

Property
IO

motion1

value

Input
Start when property

Becomes true

Trigger
Time of Day

Time
9:00

Starting at Date
2017-10-26

Ending after Date
2017-10-26

Repeat
At Interval

Interval
0:30

Ending at Time
End Time
18:00

Trigger Condition (условие переключения)

Вы можете применить дополнительные ограничения в дополнение к настройкам триггера (Trigger). Например, при использовании режима “Changes” триггера Property Change вы можете принимать во внимание текущее значение, чтобы запускать задачу только при превышении определённого уровня. Или при использовании триггера Time of Day вы можете ограничить задачу в определённые дни недели.

Поле “Trigger Condition” над списком операторов может быть использовано для применения таких ограничений. Синтаксис, используемый в этом поле, основан на JavaScript, поэтому ознакомление с этим языком программирования поможет вам максимально эффективно использовать эту функцию. Выражение, которое вы вводите сюда, имеет неявный доступ к переменной с именем `trigger`. В зависимости от режима запуска задачи эта переменная содержит одно из следующих значений:

- Если режим запуска – “Time of Day”, то это объект JavaScript Date. Этот объект предоставляет методы для определения текущего времени, даты, дня недели и т.д.
- Если режим запуска – “Property Change”; то это – число, логическое значение, строка, или объект, указывающие на новое состояние свойства.
- Для режима срабатывания “Server Startup” (запуск сервера) значение переменной `trigger` не определено. Вы можете использовать предопределённую переменную `now`, если вам нужно знать, который час.

Например, если режим запуска – “Property Change”, а наблюдаемое значение – уровень громкости, выраженный как дробное число от 0 до 1, то следующее условие запускает задачу только в том случае, если текущее значение выше 0,6:

```
trigger > 0.6
```

Если режим запуска – “Time of Day”, вы можете записать условие, подобное этому, чтобы запустить задачу во все дни, кроме воскресенья:

```
trigger.getDay() != 0
```

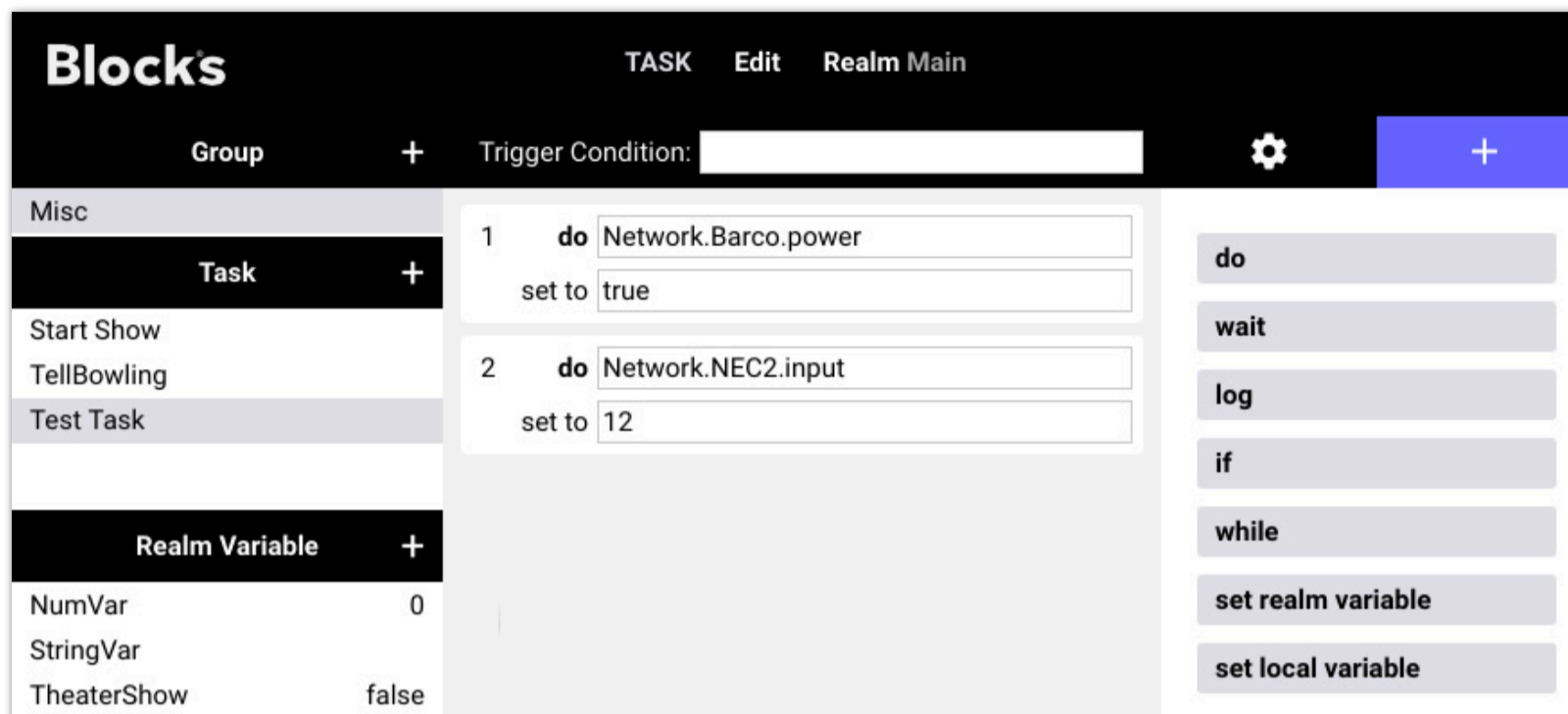
При этом запускается процедура `getDay()` для объекта Date, чтобы ввести день недели числом от 0 ... 6 (с воскресенья по субботу). После этого используется оператор «не равно» (not equal to) для сравнения введенного значения с числом 0 (которое соответствует воскресенью). Задача запускается только в том случае, если условие оценивается как истинное (true), которое будет выполняться в течение всех дней недели, кроме воскресенья.

Статус, ошибки, журнал

Запущенная задача отображается номером справа от имени в списке задач (Task list). Этот номер указывает текущий номер строки задачи и может быть использован для отслеживания хода выполнения задач. Этот номер отображается только во время выполнения задачи. Для коротких задач этот статус иногда трудно увидеть, так как задача может выполняться в течение секунды. Вы можете замедлить выполнение задачи, вставив операторы ожидания (**wait**) в нужные позиции. Кроме того, вы можете использовать операторы журналов (**log**) для явного отображения текста и значений из задач. Такие сообщения отображаются в панели журнала, автоматически появляясь под операторами, когда это необходимо. Здесь также будут перечислены ошибки, возникающие при выполнении задачи.

Операторы

Операторы определяют, что и как будет делать задача. Операторы отображаются в центре страницы. На рисунке ниже показана страница Task при выборе задачи из списка Task . Чтобы добавить оператор, щёлкните на знак «плюс» в верхнем правом углу и перетащите оператор из правого столбца в список.



Перетащите новые операторы из правой колонки в список операторов.

Изменить порядок следования операторов можно простым перетаскиванием. Для перемещения операторов между задачами сначала выберите операторы, а затем примените команды Cut/Copy/Paste в меню Edit. Выберите несколько операторов нажатием клавиш Shift-клик или Command/Control-клик.

Настройки операторов

Настройки оператора можно редактировать непосредственно в теле оператора. В качестве альтернативы после выбора оператора можно делать некоторые настройки в меню панели настроек справа.

Параметры оператора

Некоторые операторы очень просты, например, оператор ожидания, который принимает только одно значение. Другие более сложны, имеют несколько значений, называемых параметрами. Они используют выражения, похожие на те, что вы можете использовать в поле Trigger Condition, также основанные на синтаксисе JavaScript. Знакомство с языком JavaScript поможет вам при программировании задач.

Параметры иногда имеют обязательный тип. Например, единственный параметр для оператора ожидания (wait) должен быть числом в секундах. Чтобы указать значение для параметра, используйте либо точное значение, либо выражение.



Оператор ожидания **wait** использует точные числовые значения.

В некоторых случаях значение неизвестно заранее и должно быть рассчитано во время выполнения задачи. Это делается с использованием выражения (*expression*), которое похоже на формулы в электронных таблицах. Например, вы можете указать значение в приведенном выше примере следующим образом:

1+1

Для вычисления значения используется оператор сложения (*addition operator*). Но поскольку это выражение использует только литеральные константы, у него нет никакого преимущества перед простым вводом числа 2. Чтобы создать что-то более интересное, нужно вставить в выражение некоторые переменные. Например, предположим,

что вы хотите, чтобы задача подошла к этой точке в течение 4 секунд до полудня и через 2 секунды после полудня. Это может быть реализовано с помощью следующего выражения:

```
now.getHours() >= 12 ? 2 : 4
```

Это выражение сначала получает часовую составляющую текущего времени, посредством “now.getHours()”. Слово “now” (сейчас) - это глобальная переменная, которая всегда представляет текущее время, как объект Date. Метод getHours () извлекает компоненту часов как число 0 ... 23. Затем это сравнивается со значением 12. Если оно больше или равно 12 (оператор «> =»), оно даёт значение 2, в противном случае оно даёт число 4. На этом последнем этапе используется оператор [JavaScript](#), который выбирает одно из двух этих значений в зависимости от результата логического (true/false) выражения. Подробнее об операторах JavaScript читайте здесь:

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Expressions_and_Operators

Что можно использовать в выражениях

Как вы видели в примере выше, слово now можно использовать для получения текущего времени в виде объекта [Date](#). [Math](#) и [Number](#) предоставляют некоторые [полезные константы и вспомогательные функции](#) связанные с числами и математическими операциями. Литеральные (постоянные) значения указываются одним из следующих способов в зависимости от их типа:

- 23 или -345.22 для чисел.
- true или false для логических операторов.
- "Will" или 'Smith' для строковых (текстовых) литералов. Вы можете использовать одиночные ('single') или двойные ("double") кавычки для заключения строки. Это должны быть 'прямые' кавычки, а не «кудрявые».
- ◆ Если вы новичок в программировании, легко забыть о кавычках вокруг строковых литералов. Если вы забудете и напечатаете Will вместо "Will", все будет работать не так, как ожидалось. Если вы сомневаетесь, используйте оператор log, чтобы узнать, какой результат дает выражение.

Вы можете сослаться на локальную переменную, просто указав её, см. ["set local variable"](#) (задать локальную переменную) далее в этой главе. Можно также сослаться на значение переменной области (*Realm Variable*):

```
NumVar.value
```

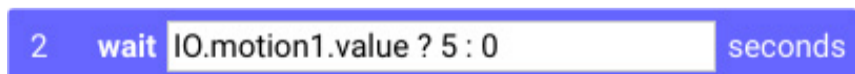
где NumVar это имя переменной (см. ["Properties and Functions"](#)).

- ◆ Вы должны добавить суффикс “.value” при доступе к значению реальной переменной (*real variable*), как показано выше. Это не относится к локальным переменным.

Наконец, у вас есть прямой доступ к статусу всей системы, это представлено свойствами объекта (object properties). Здесь используется тот же синтаксис, что и в поле “Custom Path” (пользовательский путь) триггера задач “Property Change”. В примере, показанном ранее (см. ["Property Change Trigger"](#)) это:

```
IO.motion1.value
```

Таким образом, ссылаясь на это свойство объекта, вы можете заставить оператор ожидания (wait) задерживать дальнейшее выполнение на 5 секунд, только если активирован датчик с именем "motion1":



```
2 wait IO.motion1.value ? 5 : 0 seconds
```

Те же правила и возможности применимы в поле [Trigger Condition](#) (условие переключения), где можно использовать те же постоянные, переменные, свойства объекта и операторы, а также доступна триггерная переменная, которая работает при изменении свойств или времени суток (она содержит новое значение свойства или текущее время как объект Date).

Подробнее о выражениях на Blocks wiki: <https://pixilab.se/docs/blocks/tasks>

do (выполнить)

Это наиболее часто используемый оператор. Это оператор выполнения действия. Он имеет две формы:

- *Setting a property* (настройка свойства) на системном объекте. В этом случае используется тот же синтаксис, что и при назначении кнопки для свойства, но при необходимости вы можете указать значение при помощи сложного выражения.
- *Calling a function* (вызов функции) на системном объекте. Такие функции предоставляют дополнительные возможности, недоступные, если просто задавать свойство.

Вот пример для включения проектора “Barco”, который управляется как сетевое устройство.

```
1 do Network.Barco.power
  set to true
```

Используйте настройки Target на панели настроек, чтобы выбрать свойство, которое вы хотите задать, так же, как и при подключении кнопки. Если нужно, чтобы проектор всегда включался при запуске этого оператора, просто введите true (истина) в поле “set to”. Поскольку это буквенное значение, не заключайте в кавычки true (см. [“Что вы можете использовать в выражениях”](#)).

Вызов функции

Не все функции управления имеют тип простого управления электропитанием или громкостью, где значение свойства непосредственно представляет эту функцию. Например, можно включить компьютеры в кластере WATCHOUT, используя функцию wakeUp (просыпайся). Вывод кластера из режима ожидания – это то, что должно произойти перед загрузкой и воспроизведением шоу, например, если его электропитание было отключено ночью.

```
5 do WATCHOUT.WO2.wakeUp
  timeout? Literal value or expression
```

Выполнение этого действия займет некоторое время. Вы не можете перейти к загрузке шоу до тех пор, пока система не будет запущена. Функция *wakeUp* делает это автоматически, заставляя задачу приостанавливать выполнение оператора до тех пор, пока включение не будет выполнено. Предоставляя контроль над временем ожидания, функция принимает параметр *timeout* и позволяет узнать, сколько времени придется ждать.

Этот параметр (*timeout*) является необязательным, как указано в его описании (и отмечено вопросительным знаком, следующим за его именем, см. выше). Если вы не укажете значение, по умолчанию будет применено приемлемое значение.

Другой пример использования функции – это когда нужно принять во внимание несколько значений. Например, при управлении входом WATCHOUT (функция WATCHOUT часто используется для управления поведением WATCHOUT извне) вы можете указать как значение, так и скорость перехода для этого значения. Можно привязать ползунок непосредственно к входному значению, но ползунок использует только одно значение – то, на которое установлен вход. Если вы хотите указать ещё и скорость перехода, следует использовать функцию *setInput* (задать вход). Эта функция может иметь до трёх параметров:

1. Имя входа (строка).
2. Значение для входа (число).
3. Дополнительно: скорость перехода, в миллисекундах (число).

```
1 do WATCHOUT.WO2.setInput
  name "Arne"
  value 0.7
  rate? 8
```

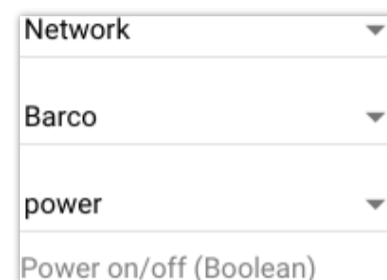
Если вы не указали скорость перехода, она будет иметь такое же значение, как и при прямом вводе при помощи кнопки. Но если вы установите последний параметр на 1000 миллисекунд, то значение на входе (Input) будет изменяться постепенно, более 1 секунды, от текущего значения до достижения новой величины.

Использование выражений в операторах

Иногда нужно, чтобы оператор выполнял разные действия в зависимости от контекста. Например, задача, которая может включать или выключать электропитание в зависимости от состояния переменной или какого-либо другого состояния. Для этого используйте оператор `do`, но замените буквенное значение `true` (истина) выражением, как показано на рисунке в начале главы “Task”. В этом примере мощность установлена на значение переменной области `TheaterShow`. Вы можете использовать любое допустимое выражение в поле “set to”, чтобы определить, включать или выключать электропитание.

Типы выражений и результаты

Свойства могут быть разных типов, это видно в нижней части панели настроек (см. рисунок справа). Это те типы, которые описаны в разделе “[Variable Types](#)”. Значение выражения, используемого для задания свойства, должно, как правило, иметь тот же тип, что и заданное свойство. Например, свойство «power» проектора – логического типа (Boolean), требует значений `true` (истина) или `false` (ложь). Уровень громкости требует цифрового выражения (обычно от 0 до 1). При использовании выражения результат иногда не совсем такой, какой вы ожидаете. Если операторы ведут себя плохо, можно вставить оператор `log` с тем же выражением, чтобы узнать, что происходит.



wait – оператор ожидания

Задерживает выполнение последующих операторов задачи на величину значения выражения.



Вы можете использовать произвольное выражение, дающее положительное числовое значение (включая дробные части, например 0.5, чтобы подождать полсекунды).

log – оператор регистрации

Выводит результат выражения в окно журнала регистрации. Это окно появляется под списком операторов при выполнении оператора `log`, или когда что-то пойдет не так при выполнении задачи. Оператор журнала регистрации часто используется для определения значения выражения или локальной переменной и является весьма полезным средством для устранения неполадок, когда задачи не работают должным образом.

note – примечание

Этот оператор не имеет никакого действия, он нужен для добавления заметок или комментариев к задаче. Например, с его помощью можно добавить описание того, что делает задача, или указать меры предосторожности, которые следует учитывать при изменении задачи.

set realm variable (задать переменную области)

Устанавливает значение переменных области ([Realm Variable](#)). Переменные области обычно используются для отслеживания состояния системы, естественно не представленного каким-либо уже существующим системным свойством. Они также полезны как средство для передачи информации между задачами. Кроме того, переменные области доступны как системное свойство для привязки к кнопке или ползунку, для переключателей задач (Task triggers), условий и выражений.

Для доступа извне к переменной области начинайте с уровня Realm, например:

```
Realm.Main.variable.TheaterShow.value
```

Переменная области в этом примере называется “TheaterShow” и существует в основной области (Main). Когда вы ссылаетесь на одну и ту же переменную Realm из задачи в основной области, вы можете просто ввести:

```
TheaterShow.value
```

set local variable (задать локальную переменную)

Локальная переменная существует только внутри задачи, где она определена. Используйте этот оператор для определения локальной переменной, а также для изменения её значения. Он не может быть доступен вне пределов задачи и не может быть использован в переключателях задач или условиях. Применение локальной переменной – принять к сведению некоторое значение в какой-то момент задачи, а затем прочитать или изменить это значение в более поздней точке. Она также используется с операциями `if` (если) и `while` (в то время, как), чтобы контролировать их поведение.

Локальная переменная не имеет заданного типа и других ограничений, как переменные области (Realm Variables). Локальная переменная принимает любое значение, которое вы в неё заложили, и может даже изменить тип, используя оператор задать локальную переменную (`set local variable`). Кроме того, в отличие от переменных области, вы не можете так просто увидеть, какое значение имеет локальная переменная. Используйте оператор регистрации **log**, чтобы отобразить её текущее значение.

if (если)

Оператор `if` позволяет назначать набор операторов, которые должны выполняться, только когда `if` истинно. В примере, показанном ниже, утверждения 5, 6 и 7 будут выполняться только в том случае, если значение `TheaterShow` истинно. См. полный пример в начале этой главы, где эта переменная области (Realm Variable) управляет включением или выключением проекторов и запуском переключателя задачи. Переменной `TheaterShow` управляют напрямую кнопкой запуска и остановки театра. Только при истинном значении (`true`), кластер `WATCHOUT` должен выходить из режима ожидания, загружать и воспроизводить шоу, используя оператор `if`.

```
4  if TheaterShow.value
5  do WATCHOUT.WO2.wakeUp
   timeout? Literal value or expression
6  do WATCHOUT.WO2.load
   showName "New Wine"
7  do WATCHOUT.WO2.play
   timeline Literal value or expression
8  end if
```

Операторы `if` и `end if` выполняются только тогда, когда выражение в операторе `if` истинно.

Если результат выражения уже является логическим, как в приведенном выше примере, нет нужды записывать

```
TheaterShow.value === true
```

чтобы убедиться, что это истина. Вместо этого просто используйте логическое значение как есть:

```
TheaterShow.value
```

Аналогично, если вы хотите выполнять операторы внутри только его значения ложь (`false`), не нужно писать

```
TheaterShow.value === false
```

Вместо этого припишите ссылку на значение с оператором `not`, например:

```
!TheaterShow.value
```

Подробнее о выражениях и операторах JavaScript читайте здесь:

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Expressions_and_Operator

Оператор while

Оператор *while* очень похож на оператор *if*, но он выполняет внутренние инструкции повторно до тех пор, пока условие остаётся истинным. Примеры ситуаций, когда вы можете захотеть повторения некоторых операторов:

- Использовать один и тот же набор команд для нескольких целей.
- Повторить некоторую команду с предварительно заданным интервалом.

Можно, конечно, применять один и тот же набор команд к нескольким объектам, просто дублируя эти утверждения и изменяя целевой объект в каждой копии, но это быстро становится громоздким по мере увеличения количества целей. Если имена целей содержат последовательный номер, как показано в примере ниже, проще использовать цикл *while*, чтобы пройти через все цели. Пример, показанный ниже, относится к включению нескольких силовых реле в стойке. Имеется четыре реле, каждое из которых управляет электропитанием в отдельной части здания. Может потребоваться включать их последовательно, (а не сразу все) с небольшой задержкой между срабатыванием каждого реле.

```
1 set local variable newState to !IO.power1.value
2 set local variable count to 1
3 while count <= 4
4   do IO["power" + count].value
   set to newState
5   set local variable count to count + 1
6   wait 0.5 seconds
7 end while
```

Силовым реле были даны последовательно пронумерованные псевдонимы: *power1*, *power2*, *power3* и *power4*. Число в конце имени можно привязать к имени с помощью выражения конкатенации (сцепления) строки, как видно по выражению в квадратных скобках в операторе *do*. Значение локальной переменной *count* (отсчёт) сначала устанавливается равным 1 (в операторе 2). Конкатенация (сцепление) “*power*” со значением счётчика даёт «*power1*», это имя первого силового реле. Затем значение счётчика увеличивается на единицу оператором 5, а цикл *while* повторяет операторы, если значение меньше или равно 4 (номер последнего силового реле).

Красота этой конструкции по сравнению с простым дублированием операторов внутри цикла *while* заключается в том, что её можно легко изменить, если система расширяется, чтобы использовать шесть силовых реле вместо четырёх. Просто измените критерий в операторе *while* на

```
count <= 6
```

а всё остальное оставьте без изменений. Аналогичным образом, если вы хотите увеличить задержку между включением каждого реле питания, достаточно изменить один оператор ожидания (*wait*), вместо того чтобы обновлять несколько операторов ожидания.

Комбинирование операторов while и if

Структурные операторы **if** и **while** могут быть использованы внутри друг друга. В приведенном выше примере задержка между переключением каждого реле электропитания действительно необходима только при их включении. Следовательно, вы можете исключить оператор *wait* при выключении, добавив оператор **if**, например так:

```

1 set local variable newState to !IO.power1.value
2 set local variable count to 1
3 while count <= 4
4   do IO["power" + count].value
   set to newState
5   set local variable count to count + 1
6   if newState
7     wait 0.5 seconds
8   end if
9 end while

```

Как и в предыдущем примере, локальная переменная *newState* сначала устанавливается в требуемое новое состояние (набором инверсии старого состояния первого силового реле, с помощью оператора *not*, отмеченного восклицательным знаком). Затем переменная *newState* используется для установки состояния силовых реле в операторе 4. Затем она также используется для управления оператором *if*, так что оператор *wait* внутри него будет применяться только тогда, когда значение оператора *newState* истинно (т.е. электропитание включено).

- ◆ Даже если не использовать операторы **wait**, выполнение всех операторов происходит за доли секунды. Следовательно, даже после вышеупомянутого изменения, реле не будут отключены абсолютно одновременно.

for each (для каждого)

Используйте **for each**, чтобы применить один и тот же набор операторов к ряду объектов Blocks, таких как прожекторы, сетевые устройства или световые приборы. Это удобная альтернатива повторению операторов вручную.

```

1 for each Spot
  recursive  Apply to any nested collections as well
  properties? power
  pattern? Optional name-matching pattern string
2   do Relative.power
   set to false
3 end for each

```

1. Выберите *Target Collection* с помощью выпадающих меню справа. Это могут быть, например, все споты в системе или определенная группа спотов, все устройства сети или Artnet.
2. При выборе целевой коллекции, которая может содержать вложенные коллекции – например, *Spot* (как показано на рисунке) или *Spot* группы с вложенными группами - выберите рекурсивный, чтобы применить вложенные операторы также к объектам внутри подгрупп (т.е. внутри любой вложенной группы спотов). Если рекурсивный *recursive* не выбран, вложенные утверждения применяются только к прямым дочерним объектам целевой коллекции (*Target Collection*).
3. Чтобы выбрать только объекты с определенным свойством или набором свойств/функций, введите их имена через запятую. Будут выбраны только объекты, обладающие всеми указанными свойствами/функциями. На рисунке выше выбраны только объекты, имеющие свойство *power*. При этом будут пропущены все гостевые споты или группы спотов, поскольку у них нет свойства *power*.

4. Чтобы выбрать объекты в целевой коллекции по имени, введите шаблон, соответствующий имени. Этот фильтр имен может быть объединен с фильтром свойств, описанным выше, в этом случае будут выбраны только объекты, имеющие эти свойства и совпадающие имена.

- ◆ **ВАЖНО:** Для доступа к свойствам или функциям целевых объектов с помощью относительного (*Relative*) пути, всегда указывайте имя таких свойств или функций в явном виде, как показано на примере свойства *power* на рисунке выше.

Необязательное сопоставление имен, описанное в шаге 4 выше, задается с помощью регулярного выражения ([regular expression](#)). Например, чтобы выбрать объекты с именем, начинающимся с *Proj*, за которым следует хотя бы одна цифра, введите следующий шаблон:

```
^Proj\d+
```

Синтаксис регулярных выражений является мощным и гибким, но может быть немного сложным для понимания. Возможно, вам понадобится онлайн-проверка регулярных выражений [regex checker](#) для проверки правильности синтаксиса.

Для утверждений внутри *for each* сначала выберите *Relative* на верхнем уровне, затем выберите нужное свойство или функцию. Это похоже на то, как вы получаете доступ к свойствам объектов внутри реплицированных дочерних блоков [replicated child blocks](#).

atomic

Обычно на выполнение каждого оператора уходит 0,05 секунды (т.е. 20 операторов в секунду). Хотя это, как правило, достаточно быстро для большинства управляющих приложений, вы можете иногда держать операторы вместе как единую временную единицу. Например, вы можете захотеть запрограммировать одновременное включение нескольких световых приборов, а не с интервалом в 0,05 секунды, т.к. это может привести к эффекту пульсации. Для этого оберните операторы, которые нужно держать вместе, внутри *atomic/end atomic*, и все они будут выполняться вместе, а не будут разнесены во времени.

Обратите внимание, что внутри атомарной секции вы не можете использовать операторы *while* или *wait*. Выполнение также не будет приостанавливаться для ожидания завершения каких-либо функций (например, функция *wakeUp*, ожидающая подключения Spot к сети). При желании вы можете обернуть секцию *atomic* внутри секции *await*, где дальнейшее выполнение будет приостановлено в конце оператора *await*, ожидая также любых таких функций внутри обернутой секции *atomic*.

await

Оператор **await** позволяет объединить несколько потенциально медленных операторов, а затем ждать их всех, а не по одному. Некоторые операторы требуют времени для завершения. Самый очевидный пример – оператор *wait*. Но в некоторых случаях и оператор *do* заставляет себя ждать. Например:

- Включение Spot Spot с помощью функции *wakeUp* при использовании [PIXILAB Player](#).
- Загрузка шоу в кластер WATCHOUT.

Последующие операторы не будут выполняться, пока не сработает медленный оператор. Хотя выполнение задачи и ожидание завершения каждого оператора, как правило, является желательным поведением, это может помешать работе. Например, вы хотите включить пять дисплеев перед началом презентаций на этих дисплеях. Если дисплеи будут включаться по очереди, это вряд ли вас устроит. Лучше сделать так, чтобы дисплеи включались одновременно.

Это может быть выполнено с помощью оператора *await*. Проще говоря, медленные операторы выполняются параллельно между *await* и *end await*. Все они будут запущены сразу, а оператор *end await* отложит выполнение последующих операторов, пока медленные операторы не будут завершены.

Вы можете узнать, какие функции могут потребовать времени для выполнения, посмотрев описания системных объектов на странице [github PIXILAB](#). Любая функция, которая возвращает Promise, потенциально может занять много времени. Например, в объекте WATCHOUT вы можете видеть, что функции *connect* и *wakeUp* делают это. Таким образом, если оператор вызывается из задачи, задача может ждать, прежде чем двигаться дальше. Аналогично,

драйвер устройства может возвращать *Promise* в некоторых случаях, например, функция *powerUp* драйвера *ChristiePerformance* (он есть на github).

13. СВОЙСТВА И ФУНКЦИИ

В приложении Blocks свойства есть у многих объектов. Свойством мы будем называть полезное системное состояние, такое как включение или выключение электропитания или уровень громкости. Свойства могут быть напрямую выведены на элементы панели управления, например, кнопки, движки, индикаторы и текст. Они также могут быть назначены или использованы в более сложных сценариях управления, таких как описанные в предыдущей главе «Задачи» (Tasks). Набор доступных свойств зависят от типа управляемого объекта. Например, дисплей-споты могут быть включены или выключены через свойство электропитания (power), но это не относится к гостевому споту (Visitor Spot) по понятным причинам.

- ◆ Некоторые свойства предоставляются через драйверы устройств ([device drivers](#)) или сценарии или могут быть добавлены уже после публикации данного руководства. Поэтому обращайтесь к информации, отображаемой внутри Blocks, так как она всегда будет наиболее актуальной.

Двунаправленность значений свойств

Свойство представлено единственным значением (*single value*), например, электропитание: его значение может быть либо истинным, либо ложным, или уровнем громкости, для которого задаётся определенный уровень, скажем, 0.5. Это позволяет легко управлять свойством – просто задать его значение. Этот механизм работает в двух направлениях; ползунок может использоваться для управления уровнем громкости, а также для указания его текущего значения. Если громкость изменяется, ползунок автоматически отображает новый уровень. Например, если Задача (Task) содержит оператор *do* для изменения уровня громкости, любой ползунок, прикрепленный к этому свойству, будет отображать новое значение.

Это позволяет значениям свойств действовать, как переключатели задач (Task triggers) в условиях переключения (Trigger Conditions) и в выражениях оператора. Любое свойство, представляющее полезное системное состояние (например, состояние электропитания – включено или отключено) может использоваться во всех этих случаях.

Свойства, доступные только для чтения

Некоторые свойства нельзя задать напрямую, они просто отображают состояние устройства. Например, свойство спота *connected* указывает, подключен ли этот спот к серверу. Если нет, вы не сможете управлять этим спотом и, естественно, хотите знать об этом. Причиной может быть отсоединение сетевого кабеля, отключение электропитания и т.п. Доступные только для чтения свойства показаны ниже курсивом и в раскрывающихся меню выбора свойств. Они не могут быть использованы в качестве целей для операторов *do*, а также не могут управляться кнопками/движками. Но они могут использоваться как переключатели задач (*Task triggers*) или в выражениях..

Функции

Как было указано выше, свойство представляет собой единственное значение (*single value*). Оно может быть привязано непосредственно к элементу управления или индикатору. Однако в некоторых случаях требуется несколько значений. Например, чтобы задать цвет осветительного прибора, вы должны предоставить красный, зелёный и синий цветовые компоненты, а также скорость затухания (при необходимости). Такое поведение с несколькими значениями не может управляться с помощью свойства, у которого только одно значение. Вместо свойства должна быть использована функция. Функция выделена **жирным шрифтом** в раскрывающихся меню. Она не может быть напрямую связана с кнопкой. Чтобы запускать функцию «с кнопки»; вызовите функцию из задачи, затем подключите кнопку к свойству выполнения задачи (*running*) задачи, чтобы запустить её..

Display spot (дисплей-спот)

[Display Spot](#) управляет фиксированными дисплеями, как правило подключенными по проводу Ethernet. Чтобы получить доступ к свойствам дисплей-спота, выберите «Spot» на верхнем уровне, затем имя спота или охватывающей группы спотов (Spot Group), и наконец, имя самого спота. Споты могут быть вложены внутрь группы спотов на любую глубину, поэтому вам нужно будет ориентироваться при указании свойства.

Свойство	Тип	Описание
active	boolean	Спот активно просматривается. Например, блоком Locator.
block	string	Полное имя блока по умолчанию в виде "group/name".
connected	boolean	Истина (true), если плеер подключен к серверу.
pictureSource	string	Имя альтернативного источника входного сигнала, например 'HDMI', если таковой поддерживается.
playing	boolean	Текущий блок воспроизводится (или временно остановлен). Указывает, что <i>любой</i> блок воспроизводится.

<i>playingBlock</i>	string	Имя текущего воспроизводимого корневого блока. Может исходить из назначенного блока, <i>priorityBlock</i> (см. ниже) или блока расписания, примененного к споту или окружающей группе.
<i>power</i>	boolean	Установите для вкл./выкл. питания. Указывает на желаемое состояние питания. Поддерживается не всеми спотами. См. также функцию <i>wakeUp</i> ниже.
<i>priorityBlock</i>	string	Полное имя блока приоритета, в виде "group/name". Переопределяет блок, заданный по умолчанию.
<i>time</i>	time	Позиция во времени при воспроизведении видео/звука. Необходим Синхронизатор (Synchronizer). Смотрите также функцию <i>seek</i> ниже.
<i>volume</i>	number	Уровень громкости воспроизведения звука 0...1. Поддерживается не всеми спотами.
<i>keyPress</i>	boolean	Одно логическое подсвойство на числовую клавишу для использования в качестве входных данных общего назначения. Доступно, только если включена функция " Numeric Keys as Inputs ".
<i>scannerInput</i>	string	Последняя отсканированная QR/NFC/RFID-метка. Доступно, если включен " RFID/QR Scanner Input " или из блока QR-сканера, выставленного на "scannerInput".

Помимо перечисленных выше свойств, дисплей-спот может иметь любое количество параметров. Они доступны под

`Spot.parameter.paramName`

где *paramName* - это имя параметра. Такие свойства доступны только тогда, когда спот подключен и либо имеет блок с параметрами, либо параметры были добавлены в настройки спота.

Дисплей-спот также предоставляет следующие функции, которые могут быть вызваны из задач и сценариев.

Параметры в круглых скобках являются необязательными и могут быть не указаны.

Функция	Параметр	Описание
<i>forceTags</i>	<i>tags</i>	Принудительно указать теги в списке тегов, разделенных запятыми (плюс любые теги в настройках спота).
	(<i>ofSet</i>)	Набор тегов для воздействия (другие оставлены в покое).
<i>gotoBlock</i>	<i>path</i>	Показать указанный путь к блоку внутри текущего корневого блока. Здесь используется тот же синтаксис, что и в действиях кнопки " Go to Local Block ", включая числовые и инкрементные сегменты. Относительный путь использовать нельзя. Используйте кнопку Copy Path (копировать путь) в области навигации при редактировании вложенного блока, чтобы получить путь к этому блоку.
	(<i>play</i>)	Воспроизвести целевой блок, как только он будет найден (например, видео).
	(<i>seek</i>)	Найти позицию в целевом блоке (например, временную позицию в видео).
<i>locateSpot</i>	<i>location</i>	Дать команду активному локатору найти указанный спот по его ID местоположения или имени (полный путь).
	(<i>isSpotPath</i>)	Если true, то местоположение указывается по полному пути к споту (разделенному точками), иначе по ID местоположения.
<i>reload</i>	(<i>reloadBrowser</i>)	Перезагрузить текущий блок. Может использоваться для принудительного обновления блока программно. Если <i>reloadBrowser</i> равно true, будет перезагружена вся веб-страница, а не только блок.
<i>scrollTo</i>	<i>x</i>	Прокрутка до указанной позиции (для работы необходим скроллер). Позиция X, 0...1 или неопределено (<i>undefined</i>).
	(<i>y</i>)	Вертикальная позиция, 0...1
	(<i>time</i>)	Продолжительность плавной прокрутки, в секундах. Jump-scroll, если не указано.
<i>seek</i>	<i>time</i>	Время поиска видео/аудио, в виде строки 'HH:MM:SS.fff'. Медиа должно находиться под синхронизатором, чтобы это свойство работало. См. также свойство <i>time</i> выше.

setCustomClasses	(classes)	Класс или классы, разделенные пробелами. Удаляет все пользовательские классы, если они пусты.
wakeUp	(timeout)	Включите питание и подключите спот. Приступайте после выполнения.

Visitor spot (гостевой спот)

Гостевой спот ([Visitor](#)) обычно подключается через сотовую или беспроводную сеть. Доступ к его свойствам осуществляется так же, как и к свойствам дисплей-спота, как описано выше.

Свойство	Тип	Описание
block	string	Полное имя блока по умолчанию, в виде "group/name".
connected	number	Число подключенных гостей.
playingBlock	string	Имя воспроизводимого блока контента. Может появиться из расписания.
priorityBlock	string	Полное имя приоритетного блока, в виде "group/name". Переопределяет блок, заданный по умолчанию.

- ◆ **ПРИМЕЧАНИЕ.** При использовании сбора данных о посетителях ([Visitor Data Collection](#)) и включении идентификации посетителя ([Visitor Identity](#)) для гостевого спота пользовательские сценарии становятся доступными для сопровождения каждого посетителя, что соответствует многим дополнительным свойствам, функциям и событиям, доступным для дисплей-спотов. Однако эти дополнительные возможности недоступны из задач или кнопок.

Location (местоположение)

Доступ к свойствам местоположения ([Location](#)) осуществляется так же, как и для дисплей-спота ([Display Spot](#)), как описано выше. Блок, связанный с местоположением, будет отображаться на гостевых спотах при переходе к этому месту с помощью локатора ([Locator](#)).

Свойство	Тип	Описание
block	string	Полное имя блока по умолчанию, в виде "group/name".
playingBlock	string	Имя текущего воспроизводимого корневого блока. Может происходить из расписания. Только чтение.
priorityBlock	string	Полное имя приоритетного блока, в виде "group/name". Переопределяет блок, заданный по умолчанию.

Spot Group (группа спотов)

Группа спотов имеет те же свойства *block* и *priorityBlock*, что и описанные выше. Блок, назначенный группе спотов (перетаскиванием или установкой свойств *block* или *priorityBlock*), применяется к спотам внутри этой группы, если он не переопределен. Таким образом, для назначения через объемлющую группу убедитесь, что споты внутри группы не имеют собственных назначений.

Task (задача)

[Задачи \(Tasks\)](#) и связанные с ними переменные области ([Realm Variables](#)) доступны в объекте область (Realm) следом за именем области. Затем для задачи выберите группу следом за именем группы и задачи. Или, для переменной области, выберите переменную, за которой следует имя переменной.

Свойство запуска (*running*) задачи истинно во время выполнения задачи. Установите для этого свойства значение «истина» (*true*), чтобы запустить задачу непосредственно от кнопки или другой задачи. Вы также можете установить значение «ложь» (*false*) для немедленного завершения задачи.

- ◆ Если задача имеет условие переключения [Trigger Condition](#), оно будет оцениваться даже при использовании этого метода, и задача будет запущена только в том случае, если условие истинно (true).

Для переменной Realm свойство value содержит ее текущее значение, которое будет иметь тип, указанный для переменной. Вы можете установить это значение из кнопки, слайдера, текстового ввода или задачи (Button, Slider, Text Input или Task).

При программировании задачи в пределах одной области используйте оператор "[set realm variable](#)" для установки значения. Читать значение можно только с помощью

`VarName.value`

где *VarName* - имя переменной realm. Вам нужно использовать только полное имя (т.е. начинающееся с объекта корневого уровня Realm) при обращении к переменной области из элементов управления панели или задач в других областях.

Канал Modbus

Устройства управления, добавленные в модули входа/выхода [Modbus I/O Modules](#) на странице Manage. Единственное свойство канала Modbus – это его значение. Если канал является входом, то значение будет доступно только для чтения. Если канал установлен на «Analog», то значение будет числом в диапазоне 0 ... 1, в противном случае значение будет логическим.

- ◆ Если вы присваиваете псевдоним каналам Modbus (Alias name), то можно получить доступ к этим каналам по имени (IO name). Это бывает полезным, когда нужно присвоить каналу не номер, а понятное имя, описывающее его функцию.

Art-Net/DMX

Чтобы обратиться к световому прибору, сконфигурированному, как описано в разделе "[Art-Net/DMX-512 Lighting](#)", сначала выберите «Artnet», затем имя светового прибора, а затем требуемое свойство или функцию. Большинство каналов имеют свойство *value*, которое, как правило, стандартизировано до диапазона 0 ... 1, независимо от своего фактического разрешения. Диапазон обычно указан под выбранным свойством.

Если вы установите значение свойства, будет применена установленная для прибора скорость затухания по умолчанию [Default Fade Rate](#). Если же вместо этого вы выбрали функцию *fadeTo*, вы можете указать как целевое значение, так и скорость затухания.

Для некоторых источников света с распространёнными функциями, такими как цветовые каналы RGB или функциями панорамирования / наклона, может быть предусмотрен дополнительный параметр с именем *multi-Func*, позволяющий устанавливать несколько значений с помощью одного оператора задачи.

Некоторые свойства могут принимать дискретные значения, которые задаются строкой. Для свойств, которые имеют и дискретное состояние, и соответствующее числовое значение, сначала выберите дискретное состояние, затем (с помощью второго оператора do) установите соответствующее значение.

WATCHOUT

Управляет подсистемой WATCHOUT, добавленной в [WATCHOUT Clusters](#) на странице Manage.

Свойство	Тип	Описание
auxTimeline	timeline	Дополнительные шкалы (auxiliary timelines) шоу. Если имеются.
input	number	Текущее значение входа (как задано в Blocks).
connected	boolean	True, если кластер подсоединён, false, если кластер находится оффлайн.
layerConditions	number	Комбинированное числовое значение включенных условий слоя (Combined numeric value of enabled layer conditions).
playing	boolean	True, если воспроизводится основная временная шкала (main timeline).
ready	boolean	True, если кластер полностью действующий, false, если кластер busy (занят) (например, загружает файлы).

showName	string	Имя текущего загруженного шоу. Например, имя шоу для дисплей-компьютера (без расширения ".watch" в названии файла), или полный путь (с расширением ".watch") для продакшн-компьютера.
standBy	boolean	True, если кластер находится в режиме ожидания.
time	time	Текущее положение во времени на основной временной шкале. В задачах выражается в миллисекундах.

При выборе auxTimeline необходимо выбрать или ввести имя дополнительной временной шкалы, которая затем предоставляет доступ к следующим дополнительным свойствам (sub-properties):

Свойство	Тип	Описание
playing	boolean	True, если дополнительная шкала воспроизводится, false, если она на паузе или остановлена.
stopped	boolean	True, если дополнительная шкала остановлена, false, если она воспроизводится
time	time	Текущее положение во времени на дополнительной временной шкале. В задачах выражается в миллисекундах.

В дополнение к вышеуказанным свойствам кластеры WATCHOUT предоставляют следующие функции, которые могут быть вызваны из задач:

Функция	Параметр	Описание
connect		Попытаться подключиться к кластеру. Продолжать после завершения.
disconnect		Отключиться от кластера.
gotoControlCue	name	Имя управляющего сигнала (control cue) для быстрого перехода.
	reverseOnly	Искать только назад во времени для указанного управляющего сигнала.
	timeline	Имя временной шкалы для управления (это основная шкала, если имя не указано).
gotoTime	time	Время перехода (указано в миллисекундах).
	timeline	Имя временной шкалы для управления (это основная шкала, если имя не указано).
load	showName	Загрузка шоу с задержкой выполнения задачи до завершения загрузки. Указывает полный путь при управлении продакшн-компьютером, иначе просто показывает имя. Всегда будет загружаться, даже если это шоу уже загружено.
pause	(timeline)	Приостановка временной шкалы. Название временной шкалы, не указано для основной шкалы.
play	(timeline)	Воспроизведение временной шкалы. Параметр – имя временной шкалы, не указано для основной шкалы
reset		Остановка всех временных шкал.
setInput	name	Установить значение названного входа.
	value	Значение для установки
	(rate)	Скорость перехода, в мс.
setStandBy	standby	Установить состояние ожидания для кластера.
	(rate)	Скорость перехода, в миллисекундах.
stop	(timeline)	Остановить временную шкалу. Название временной шкалы, не указано для основной шкалы.
sleep		Отсоедините, выключите и обесточьте все дисплей-компьютеры в кластере.
wakeUp	(timeout)	Проснуться и подключить все дисплей-компьютеры кластера. Приступайте после выполнения.

Schedule (расписание)

Каждый [Block Scheduler](#) (планировщик блоков) раскрывает группу/имя своего текущего блока как свойство объекта верхнего уровня Schedule (расписание). Использование этого свойства в качестве цели для Reference Block, позволяет вам запланировать часть композиции, которая становится независимой запланированной "зоной".

Network (сеть)

Элементы управления добавляются в [Network Devices](#) на странице Manage. При использовании без драйвера устройства доступно лишь свойство *connected* (в виде TCP, только для чтения) и следующие функции:

Функция	Параметр	Описание
enableWakeOnLAN		Подготавливает это устройство к действию по команде wakeOnLAN (запоминает его MAC-адрес).
sendBytes	toSend	Отправляет массив необработанных байтов данных.
sendText	toSend	Текст для отправки на удаленное устройство.
	(eoln)	Конец строки. По умолчанию – возврат каретки.
wakeOnLAN	(macAddress)	Отправляет wake-on-LAN на это устройство или на указанный MAC-адрес, выраженный в виде строки шестнадцатеричных байтов, разделенных двоеточиями (HEX bytes), или в виде массива из 6 чисел.

Если есть драйвер устройства, он обычно предоставляет несколько свойств более высокого уровня (например, включение/выключение питания или выбор входа при управлении видеопроектором). Он показывает или не показывает вышеуказанные функции. В большинстве случаев драйвер также управляет соединением с устройством, пытаясь постоянно повторно подключаться, пока не будет установлено соединение.

14. АВТОРИЗАЦИЯ И РОЛИ

Blocks имеет фиксированный набор ролей, которые могут быть назначены пользователям. Назначение ролей можно сделать в разделе Manage > Users (см. ["Users"](#)). Для работы в редакторе Blocks пользователь должен осуществить вход в систему. При доступе к споту вход в систему необязателен и обычно не используется (см. ["Role Required to Use Spot"](#)).

Доступны следующие роли пользователя:

Роль	Описание
Admin	Полный контроль над всеми аспектами Blocks, включая пользователей и их роли.
Manager	Полный контроль над всеми аспектами Blocks, кроме пользователей и их ролей.
Creator	Может создавать и редактировать контент всех видов.
Editor	Может редактировать существующий контент.
Contributor	Может заменять контент в блоках верхнего уровня: текстовом и медиа.
Staff	Зарегистрированный пользователь, который не может ничего редактировать (может быть использовано для контроля доступа).

Каждая роль представляет собой подмножество вышестоящей роли. А каждая высшая роль обладает всеми возможностями низших ролей, плюс свои возможности. То же самое относится и к использованию этих ролей для настройки ограничений доступа, как описано ниже. В нижеследующей таблице описаны разрешения редактора Blocks для каждой роли.

Функции	Admin	Manager	Creator	Editor	Contributor
Пользователь: добавить / удалить / настроить	✓				
Управляемые объекты: добавить / настроить	✓	✓			
Управляемые объекты: удалить / переименовать	✓	✓			
Спот: добавить / настроить / переназначить	✓	✓			
Спот: удалить / переименовать	✓	✓			
Добавить расписание	✓	✓	✓		
Дублировать/редактировать блок	✓	✓	✓	✓	
Добавить, переименовать, редактировать блок	✓	✓	✓		
Создание и изменение шаблона блока и защиты блока	✓	✓	✓		
Удалить блок	✓	✓	✓		
Редактировать расписание (schedule)	✓	✓	✓	✓	
Удалить/переименовать расписание	✓	✓			
Редактирование/замена контента корневого медиа/текстового блока	✓	✓	✓	✓	✓
Добавить/настроить/редактировать задачу/группу/переменную	✓	✓			
Удаление/переименование задачи/группы/переменной	✓	✓			

Настраиваемые ограничения доступа

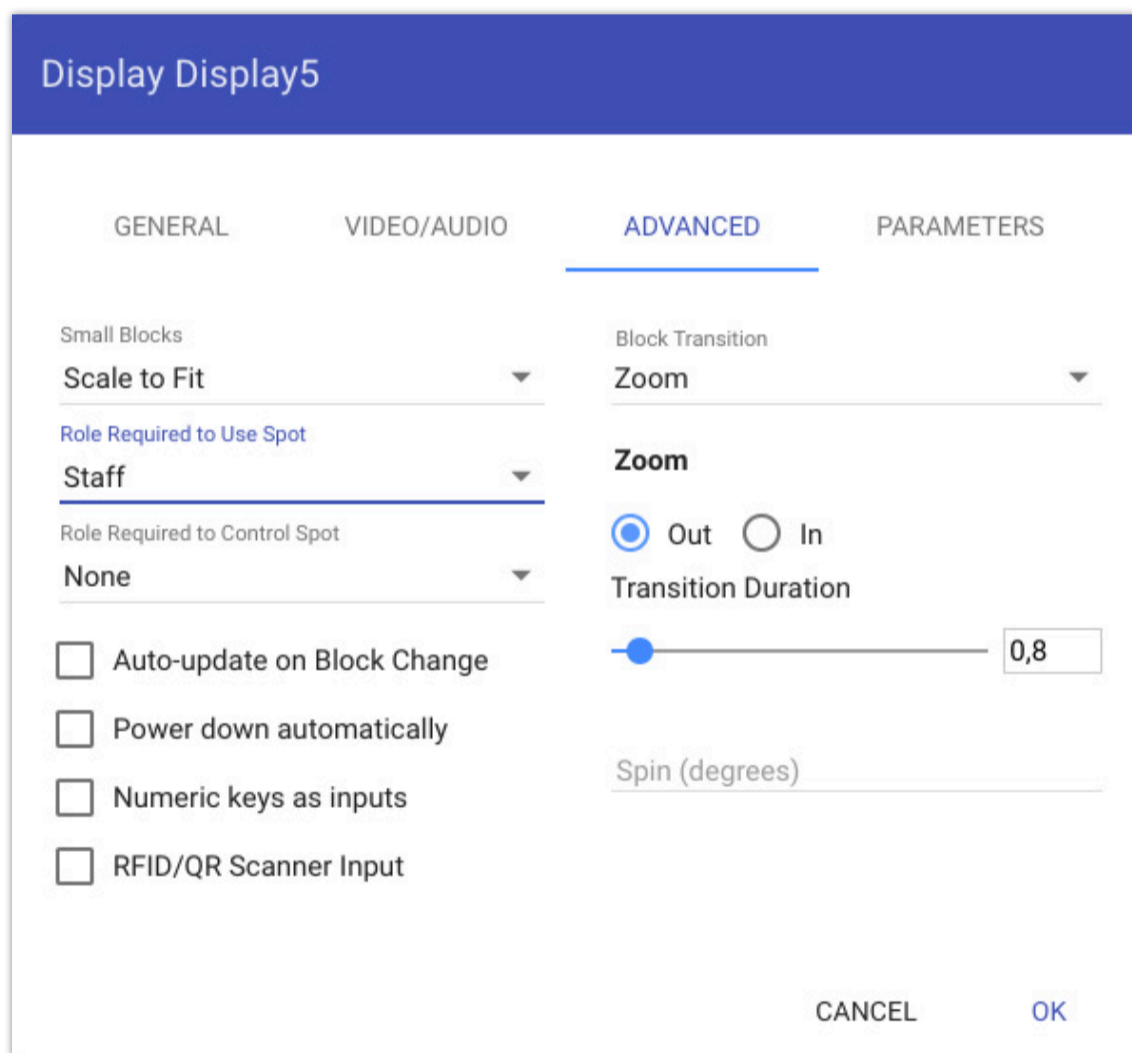
В дополнение к фиксированным ограничениям редактора, перечисленным выше, возможно также настроить специфичные для объекта ограничения для различных функций.

Функция	Предоставляется с помощью
Просмотр спота Display или Visitor	настройки спота “Role Required to Use” (роль, необходимая для использования).
Увидеть блок в редакторе	настройки блока или группы “Role Required to See” (роль, необходимая для просмотра).
Назначить (перетащить) споту блок или расписание	настройки спота или вмещающей группы “Role Required to Control” (роль, необходимая для контроля).

Как обсуждалось выше, роли образуют правильные подмножества. Следовательно, если вы выберете роль Creator в качестве ограничения для доступа, вы неявно также предоставите доступ пользователям Manager и Admin.

Роль, необходимая для использования спота (required to Use a Spot)

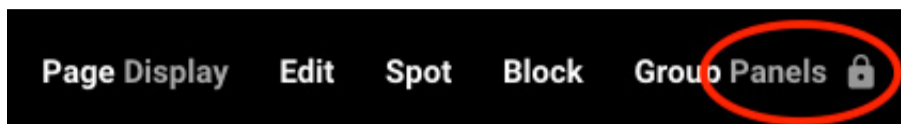
Вы можете ограничить доступ к дисплей-споту (Display) или гостевому споту (Visitor) с помощью ролей. Это делается через параметр “Role Required to Use Spot”, который находится на вкладке Advanced настроек спота.



Если вы выберете здесь какую-либо роль, кроме *None* (нет), потребуется вход в систему, прежде чем спот отобразит свой контент. Таким образом, эта настройка полезна только для устройств отображения, имеющих клавиатуру для ввода учетных данных. Этот параметр предназначен для спотов дисплеев (Display) или гостя (Visitor), используемых в качестве панелей управления для объектов системы с ограниченным доступом. Кнопка или другой элемент управления, прикрепленный к объектам системы с ограниченным доступом, не действуют, если только спот не был зарегистрирован пользователем с соответствующей ролью настройкой “Роль, необходимая для управления”, см. ниже.

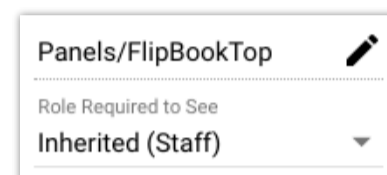
Роль, необходимая для просмотра блока в редакторе

Вы можете регулировать, кто имеет право видеть определенные блоки в редакторе. Скрывая блоки от некоторых пользователей, вы уменьшаете сложность и вероятность ошибок, показывая только блоки, определяемые ролью вошедшего в систему пользователя. Вы можете указать роль, необходимую для просмотра блоков, либо на уровне группы, либо для каждого отдельного блока. Таким образом, если блоки, которые являются "запрещенными" для некоторых ролей, уже сгруппированы, установите ограничение на уровне группы. Для этого выберите группу в меню в верхней части страницы Display и щёлкните символ замка рядом с именем группы.



Появится диалоговое окно, позволяющее указать минимальную роль, необходимую для просмотра блоков этой группы в редакторе. После установки такие блоки будут скрыты от пользователей с более низкой ролью.

В качестве альтернативы можно задать минимальную роль, необходимую для просмотра блока в самом блоке. Эта настройка доступна только на самом верхнем уровне блока. По умолчанию блоки наследуют роль, заданную на уровне группы, на что указывает параметр "Inherited", показанный на рисунке справа. Однако, вы можете использовать настройку уровня блока, чтобы переопределить уровень группы, сделав его более ограничительным или более доступным.



Роль, необходимая для управления

Blocks обеспечивает детальный контроль над тем, кто и чем может управлять, на основе ролей, назначенных пользователям. Ограничения на управление задаются для каждого объекта и представлены значками замков. Эти замки регулируют внешнее управление, исходящее от кнопок и других элементов управления, организованных на спотах. Изначально роль, необходимая для управления всеми объектами, установлена на None, это означает, что для управления объектами не требуется никакой конкретной роли. Эта разрешающая настройка соответствует поведению предыдущих версий Blocks и гарантирует, что функции управления будут работать так же, как и раньше.

Ограничения управления задаются ролями, перечисленными выше, но существует ещё псевдороль, называемая *Identified Spot (идентифицированный спот)*. Эта псевдороль не требует входа в систему и предоставляется всем спотам Display и Visitor (гость) в системе, кроме спота Visitor по умолчанию, как назначено настройками [serverRoot-Redirect](#). Этот параметр обеспечивает некоторую безопасность, поскольку посетители, использующие имя спота Visitor, заданное по умолчанию, не могут управлять такими объектами. Только споты, идентифицирующие себя по ID, известному системе, или использующие явный URL-адрес спота Visitor, считаются идентифицированными спотами (Identified Spot). Следовательно, это полезная настройка по умолчанию для всех объектов, не предназначенных для управления случайными посетителями (или хакерами).

Все роли, кроме None и *Identified Spot* требуют, чтобы управлял объектом лишь зарегистрированный пользователь. Кроме того, вошедший в систему пользователь для управления объектом должен иметь указанную роль или более высокую роль. Поведение при входе в систему должно запускаться параметром "[Role Required to Use a Spot](#)" (роль, необходимая для использования спота), описанным выше. Если пользователь с требуемой ролью не вошел в систему, кнопка или другой элемент управления не будут работать, и будет зарегистрировано сообщение об ошибке.

Конечно, вы не предоставите «опасные» функции управления в общедоступных спотах для посетителей, это явное ограничение обеспечивает дополнительную защиту от злонамеренных попыток контролировать функции системы. Оно действует как внутренний барьер, проверяя все попытки, поступающие извне, следя за тем, чтобы они подчинялись правилам доступа, определенным для каждого объекта.


- ◆ ПРИМЕЧАНИЕ. Ограничение "Role Required to Control" (роль, необходимая для управления) применяется только к попыткам управления извне (например, из спотов). Внутреннее управление в Blocks (задачи и сценарии) не подпадает под эти ограничения.

Символ замка, указывающий на применяемые ограничения:

🔒 Ограничение управления не применяется.

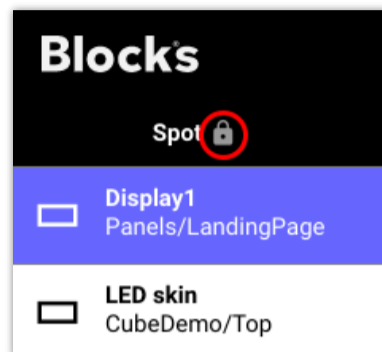
🔒 Ограничение управления применяется.

🔒 Применяется более строгое ограничение управления по сравнению с родительским объектом или группой.

 Применяется менее строгое ограничение контроля по сравнению с родительским объектом или группой

Споты

Роль, необходимая для управления (“Role Required to Control”), контролирует все свойства спотов, такие как *block*, *priorityBlock* и *power*. Роль может быть установлена в разделе ADVANCED в настройках спота. Если не переопределено, роль наследуется от вмещающей группы спотов (как указано в настройках группы) или на верхнем уровне – от общей настройки “Role Required to Control”, доступной через символ замка над перечнем спотов.














Управляемые объекты

Все управляемые объекты, найденные на странице Manage, имеют настройки ограничения управления, на что указывают символы замков. Настройки замка, отображаемые в списке управляемых объектов, определяют ограничения по умолчанию для каждой категории (кластеров WATCHOUT, модулей ввода / вывода Modbus, устройств TCP / UDP и световых приборов Art-Net / DMX).

При выборе категории отображаются ее объекты, и для каждого из этих объектов можно переопределить настройку, унаследованную от категории. Наконец, для некоторых объектов (таких как выходы Modbus) вы можете переопределить эту настройку для каждого отдельного выхода.

На иллюстрации, показанной ниже, ограничения применяются к кластерам WATCHOUT и к модулям ввода-вывода Modbus на уровне категории. Модуль “Netio” Modbus предъявляет более строгие требования к роли, о чём свидетельствует знак «плюс» на символе замка. Все его выходные каналы имеют одинаковые требования, кроме канала 103, который является менее ограничивающим.

Чтобы увидеть требование к роли, заданное на каждом уровне, и изменить эту настройку, щёлкните на символ замка на нужном уровне.

Blocks						MANAGE	Edit	Add	
Users	● f	192.168.2.32	YES			Module Name * Netio			
Licenses	● g	192.168.2.34	YES			IP Address * 192.168.2.24			
WATCHOUT Clusters		● Netio	192.168.2.24	YES					
Instagram Feeds		Channel	Direction	Type	Alias	Modbus Channel Number * 103			
Modbus I/O Modules		101	Out	Digital	Power1		Alias (under 'IO') Power3		
TCP/UDP Devices		102	Out	Digital	Power2		Signal Direction <input checked="" type="radio"/> Output <input type="radio"/> Input		
Art-Net/DMX Lighting		103	Out	Digital	Power3		Signal Type <input checked="" type="radio"/> Digital <input type="radio"/> Analog		
		104	Out	Digital	Power4				

Задачи и переменные области

Аналогичная иерархия ограничений управления применяется к задачам и переменным области.

Ограничение верхнего уровня здесь указывается на уровне области, как определяется символом замка рядом с названием области в верхней части окна. Этот параметр определяет ограничение, применяемое

к переменным области и группам задач, если оно не переопределено. Все параметры, применяемые на уровне группы, наследуются задачами в этой группе, если они не переопределены конкретной задачей.

Для задач это ограничение определяет, кто может запускать задачу, устанавливая её свойство запуска (*running*) с помощью кнопки. Это не влияет на нормальный запуск задач, как указано в настройке *Trigger* для каждой задачи..

Аналогично, для переменных области определяется, кто может устанавливать значение переменной с помощью кнопки, движка или другого элемента управления. Это не влияет на способность задания переменных области изнутри (например, из операторов задачи).

- ◆ Поскольку задачи редко предназначены для прямого запуска анонимными посетителями, обычно целесообразно установить требование роли уровня Realm (область) на более строгое значение, чем *None*. При необходимости это требование можно смягчить для отдельных задач или переменных.

A. ФАЙЛОВАЯ СТРУКТУРА СЕРВЕРА

Сервер Blocks хранит свои данные, а также ваши медиафайлы и другие настройки в папках. Для использования Blocks вам не нужно знать, как это выглядит, но иногда бывает полезно понимать, где что лежит, например, если вы хотите переместить что-либо с одного сервера Blocks на другой.

По умолчанию Blocks хранит свои данные в каталоге с именем "PIXILAB-Blocks-root" в домашнем каталоге учетной записи пользователя, под которой запущен сервер.

PIXILAB-Blocks-root	Корневой каталог данных Blocks, расположенный в пользовательской директории.
config.yml	Конфигурационный файл сервера (см. " Server Configuration Options ").
logs	Файлы журналов регистрации, полезные при устранении ошибок сервера.
model	Здесь настройки страниц Manage, Task, Spots и Schedules.
Core	Внутренние данные сервера, однозначно идентифицирующие сервер Blocks.
ModbusModules	Содержит один файл на модуль Modbus, добавленный на стр. Manage.
NetworkDevices	Содержит один файл для каждого модуля сети, добавленного на стр. Manage
Schedules	Содержит все блоки Schedule
Main	Каталог, содержащий группу "Main" (по каталогу на группу).
Entrance.json	Файл, описывающий расписание "Entrance" (вход) для основной группы "Main".
Spot	Файл, описывающий конфигурацию всех спотов.
Task	Каталог, содержащий все ваши области (по одному файлу на область).
Users	Файл с информацией обо всех пользователях, добавленных через страницу Manage.
Licenses-XXXX	Дополнительная информация о лицензии. Не удалять и не предоставлять общий доступ.
WOClusters	Содержит один файл для каждого кластера WATCHOUT, добавленного на странице Manage.
public	Каталог, содержащий общедоступные (http) файлы.
block	Здесь все созданные вами блоки (кроме расписаний - Schedules).
Main	Группа блоков с именем "Main".
Saltkrakan	Папка блока с именем "Saltkrakan".
media	В этой директории - медиафайлы, используемые в этом блоке.
Spec.pixi	Данные, описывающие структуру блока.
script	
driver	Файлы драйверов для сетевых устройств.
user	Пользовательские скрипты.
feed	Скрипты подачи базы данных.

Перенос конфигурации сервера

Вы можете переносить как целые группы блоков, так и отдельные блоки, копируя блок или всю папку группы с одного сервера Blocks на другой. Вам не нужно перезагружать сервер после добавления на него новых блоков. Просто перезагрузите страницу Display в редакторе, чтобы появились новые блоки.

Аналогичным образом вы можете переносить расписания, сетевые устройства, модули modbus, задачи и драйверы устройств, копируя эти каталоги или файлы с одного сервера Blocks на другой. В этом случае перед добавлением файлов необходимо завершить работу сервера, а затем перезапустить его.

Резервное копирование сервера

Наконец, чтобы сделать резервную копию всей конфигурации сервера Blocks, просто скопируйте каталог "PIXILAB-Blocks-root" в безопасное место. Чтобы потом восстановить эту резервную копию, выйдите из сервера и замените каталог "PIXILAB-Blocks-root" на сохраненный ранее каталог. Подробнее о резервном копировании [здесь](#).

В. СЦЕНАРИИ И ДРАЙВЕРЫ УСТРОЙСТВ

В дополнение к программированию на основе задач, описанному ранее в этом руководстве, Blocks также поддерживает расширенное программирование сценариев с использованием языка программирования [TypeScript](#). Скрипты можно использовать для работы с очень сложными сценариями, такими как игровые движки, где как правило, требуется традиционное программирование.

Подробнее о требованиях и вариантах использования сценариев можно узнать в [PIXILAB Wiki](#).

Разработка драйверов устройств

Язык программирования TypeScript также используется для разработки пользовательских драйверов устройств, о которых говорилось ранее в этом руководстве. Такая разработка требует аналогичных навыков программирования и также описана в [PIXILAB Wiki](#).