**Создание приложения средней сложности для мониторинга условий выращивания растения**

Содержание

[Введение 2](#_Toc507370574)

[Аудитория 3](#_Toc507370575)

[Ограничения использования образовательной версии платформы ThingWorx 4](#_Toc507370576)

[Краткий обзор платформы ThingWorx 5](#_Toc507370577)

[ThingWorx Composer 5](#_Toc507370578)

[Mashup Builder 7](#_Toc507370579)

[Моделирование 10](#_Toc507370580)

[Подготовка контекста проекта 10](#_Toc507370581)

[Задание моделей данных 10](#_Toc507370582)

[Создание объектов на основе моделей данных 10](#_Toc507370583)

[Организация накопления и хранения данных 12](#_Toc507370584)

[Организация обработки данных 12](#_Toc507370585)

[Организация взаимосвязей и реализация бизнес-логики 14](#_Toc507370586)

[Организация пользовательских взаимодействий 15](#_Toc507370587)

[Создание мэшапа 15](#_Toc507370588)

[Добавление виджетов к мэшапу 15](#_Toc507370589)

[Связывание виджетов и сервисов 17](#_Toc507370590)

[Отладка приложения 21](#_Toc507370591)

[Составление перечня 21](#_Toc507370592)

[Тестирование приложения с помощью симулятора 21](#_Toc507370593)

# Введение

В этом проекте вы создадите приложение Интернета вещей средней сложности. Приложение будет получать информацию от двух источников: от системы сбора данных, которая измеряет влажность почвы, и из сервиса геоданных openweathermap.

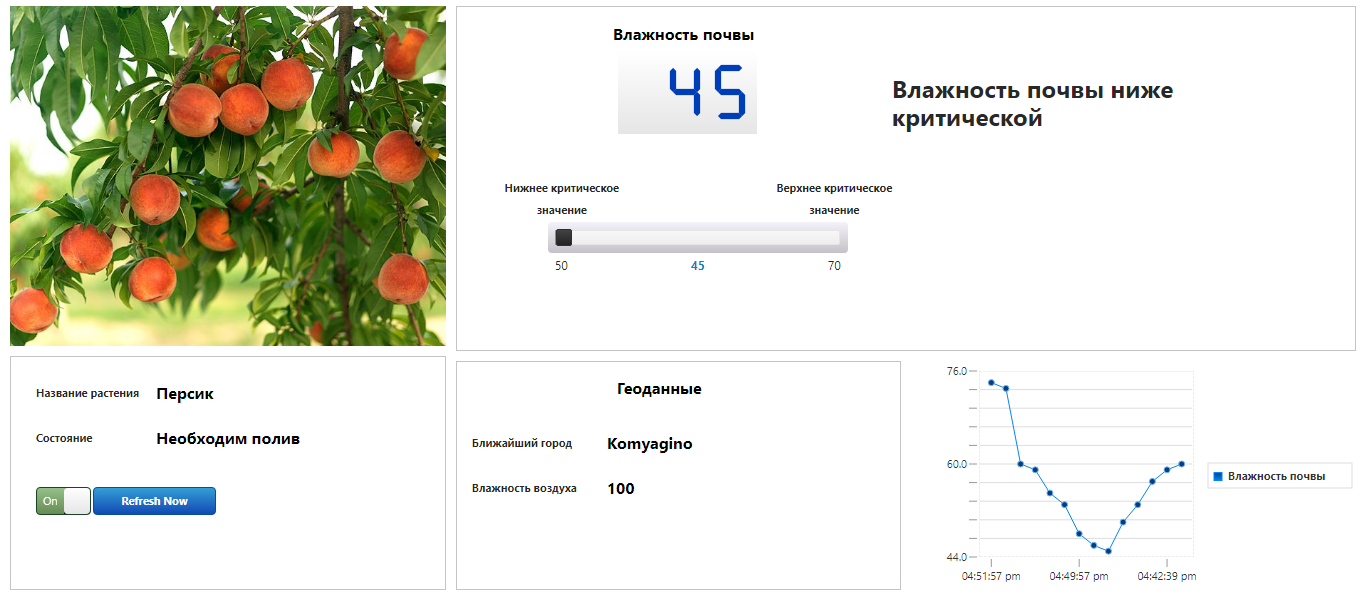
Полученная информация будет обрабатываться и затем выводиться в удобном виде на веб-странице. В итоге, на веб-странице пользователь будет видеть следующую информацию:

* изображение, название и состояние растения;
* текущее значение влажности почвы и ее положение относительно критических значений;
* график изменения влажности почвы от времени;
* название ближайшего города и влажность воздуха по заданному местоположению.

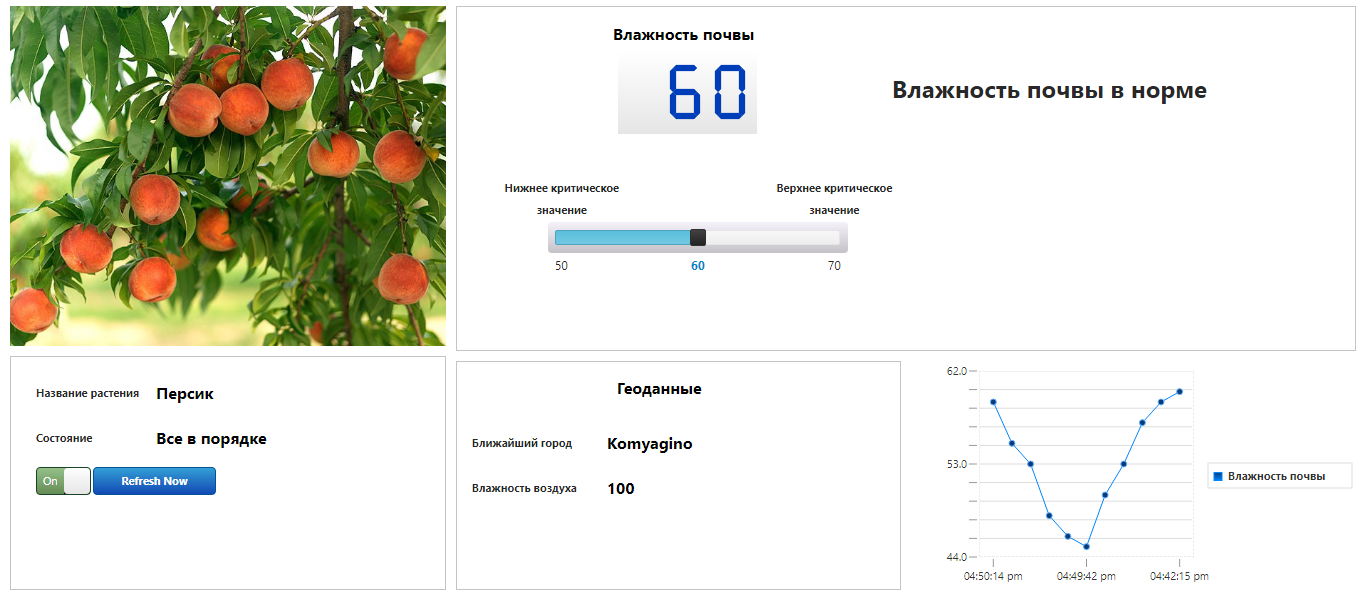
Приложение будем делать на платформе ThingWorx от компании PTC.

Вид итоговой веб-страницы.

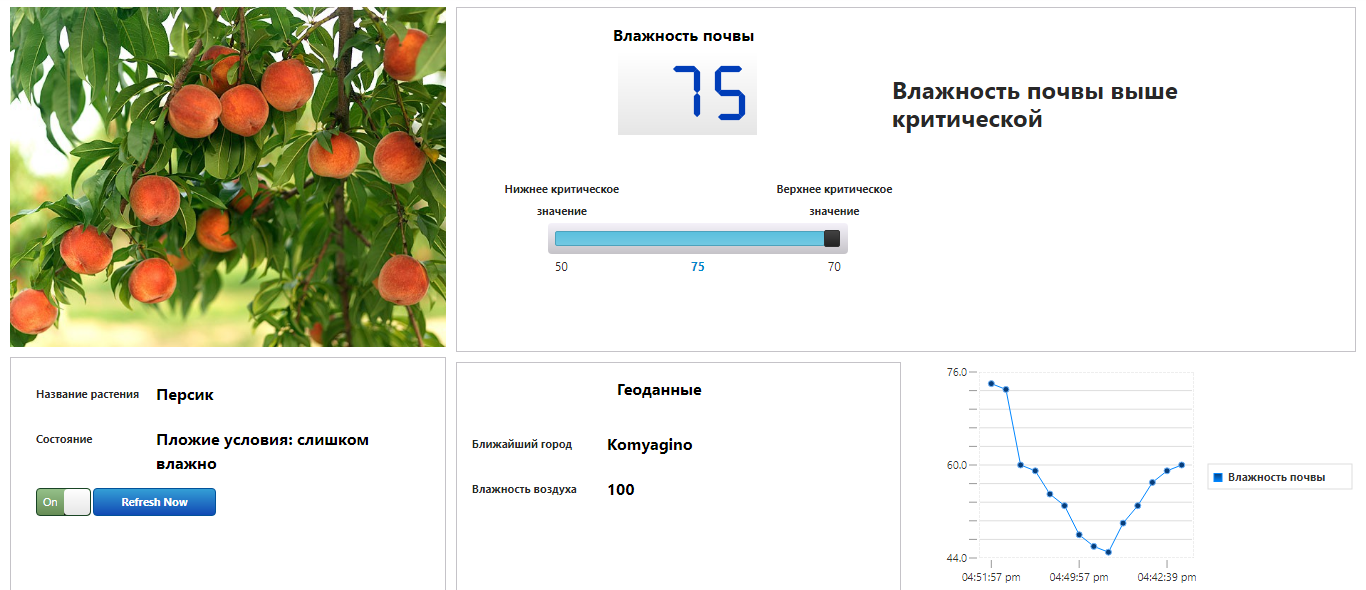
1. **При влажности почвы меньше нижней критической**

****

1. **При влажности почвы в норме**

****

1. **При влажности почвы больше верхней критической**

****

## Аудитория

Этот проект направлен на всех, кто заинтересован в более детальном знакомстве с платформой ThingWorx.

Чтобы успешно пройти проект, нужно быть опытным пользователем ПК и обладать базовыми навыками программирования: знать язык Java Script хотя бы на начальном уровне, желательно иметь представление об XML.

Рекомендуемый возраст для успешного самостоятельного выполнения проекта – 16+

## Ограничения использования образовательной версии платформы ThingWorx

Образовательная версия платформы:

* не поддерживает функции импорта/экспорта данных. С любыми объектами можно работать только на том аккаунте, на котором они создавались. Их нельзя выгрузить и использовать на другом сервере платформы или через другого пользователя на этом сервере.
* позволяет пользователю работать только со своими объектами и не позволяет работать с объектами других пользователей. Все объекты, созданные одним пользователем, не будут видны другим пользователям.

Если вам недостаточно возможностей образовательной версии платформы, вы можете использовать [другие варианты](https://www.technovision.ru/store/%D0%9F%D0%9E-ThingWorx-c26468015).

Примечание: при моделировании объектов на образовательной версии платформы ThingWorx (Educator Extension) к названию объектов автоматически будет добавлено через подчеркивание имя пользователя в системе. Например, MyThing\_user22.

# Получение доступа к платформе ThingWorx

Для получения доступа к платформе необходимо зарегистрироваться с использованием кода доступа. Для этого:

1. Перейдите по ссылке <https://www.thingworxacademic.com/>
2. Под надписью New User Registration в поле Access Code вставьте код **seuFUAC4** и нажмите **Register**.
3. Введите ваши личные данные **на английском языке**:

* First Name – имя
* Last Name – фамилия
* I am studying… - напишите **IoT**
* University or Organization - образовательное учреждение

Если не учитесь / не знаете, какое учреждение писать, напишите **Other**

* Email – напишите вашу почту
* Create a username… - придумайте логин, **используйте латинские буквы**
* Create a password… - придумайте пароль
* Поставьте галочку напротив **I agree…**

1. Нажмите **Register Now**.  
   Появится личный кабинет, через который можно заходить в платформу.

Кроме того, в личном кабинете на вкладке **Academic Resources** вам доступны полезные образовательные ресурсы, включая бесплатные онлайн-курсы от университета Интернета вещей (IoT University).

1. Нажмите **Launch ThingWorx**.
2. В появившемся окне введите логин и пароль, которые вы указали при регистрации.
3. Дождитесь загрузки платформы Thingworx.

В дальнейшем можете заходить в платформу по адресу

<https://academic-educatorsextension.portal.ptc.io/Thingworx/Composer/>, используя ваш логин и пароль.

# Краткий обзор платформы ThingWorx

**ThingWorx** – это ведущая платформа Интернета вещей, которая позволяет связывать между собой устройства, создавать приложения Интернета вещей и оптимизировать операции.

**ThingWorx Foundation** – основа платформы ThingWorx. Он интегрирован со всеми остальными компонентами платформы. В дальнейшем будет более подробно рассмотрена именно компонента ThingWorx Foundation.

ThingWorx Foundation имеет две основные компоненты – ThingWorx Composer и Mashup Buider.

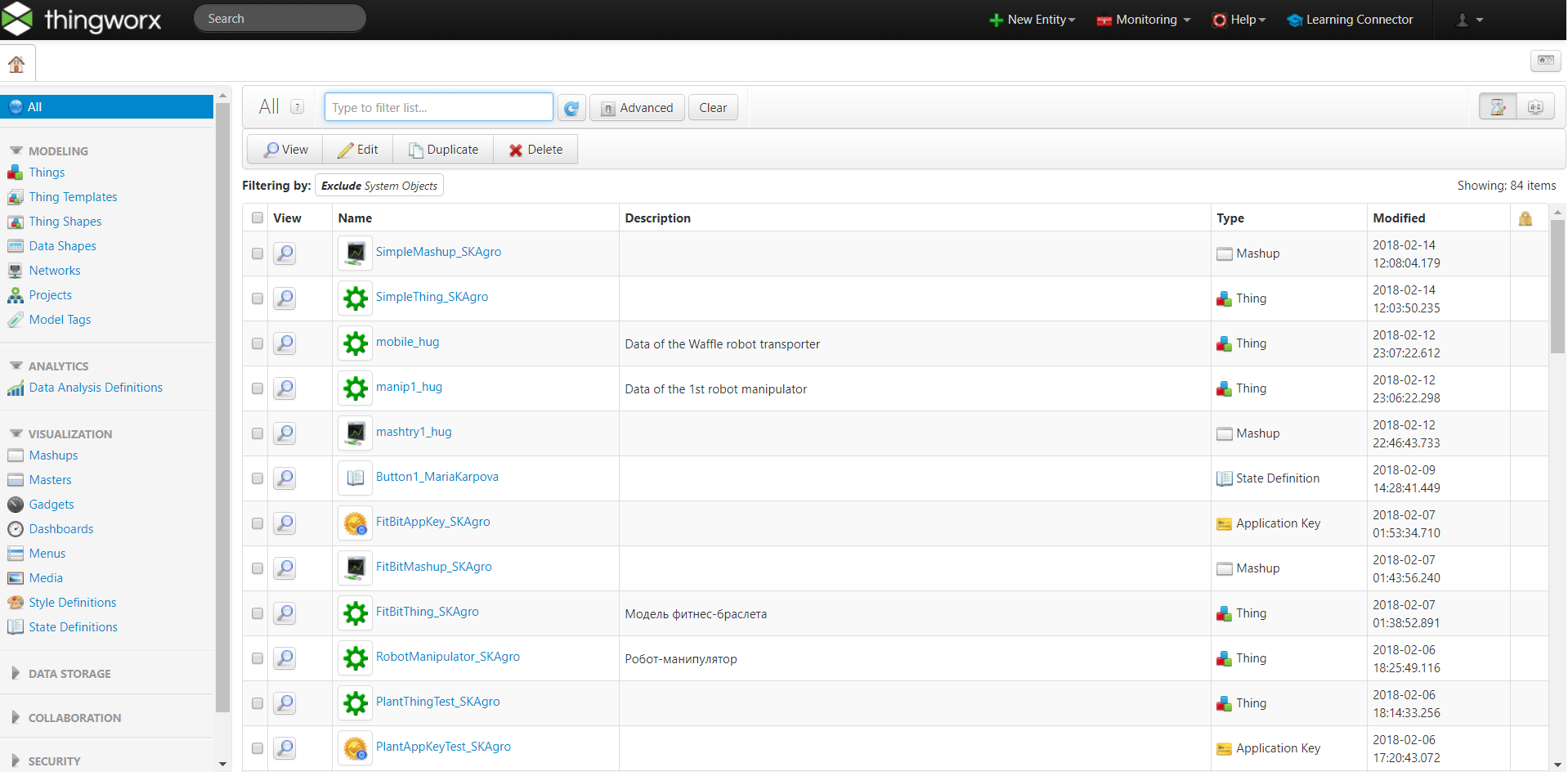
• **ThingWorx Composer** – среда моделирования решений Интернета вещей.

• **Mashup Builder** – “Drag and drop” инструмент для создания интерактивных веб-интерфейсов, информационных панелей («дэшбордов») и рабочих пространств для совместной работы.

***Замечание****. В ThingWorx Foundation есть и другие компоненты. Они будут подробно рассмотрены в других документах.*

## ThingWorx Composer

ThingWorx Composer – веб-среда моделирования решений Интернета вещей, которая позволяет эффективно работать с данными, представляя их в виде отдельных цифровых объектов (Entities).



Объекты платформы (Entities) сгруппированы по следующим разделам:

* **Modeling** – объекты для моделирования решений Интернета вещей: цифровые модели физических устройств, шаблоны, тэги и др.
* **Analytics** (аналитика) – содержит одну компоненту для экспорта данных в ThingWorx Analytics
* **Visualization** - объекты для визуализации данных через веб-интерфейс: веб-страницы, дэшборды, стили и др.
* **Data Storage** – объекты для хранения данных: таблицы, потоки данных и др.
* **Collaboration** – объекты для совместной работы: блоги и вики.
* **Security** – объекты для работы с безопасностью: пользователи, группы пользователей, ключи для связи с внешними устройствами и др.
* **System** - системные объекты: подсистемы, ресурсы и др.

Сейчас многие объекты физического мира собирают и отправляют в Интернет огромное количество информации. В связи с этим появляется смысл говорить об отдельных объектах цифровой действительности – **«цифровых двойниках»** (Digital Twins) – виртуальных моделях физических объектов, которые объединяют в себе информацию о соответствующих им физических объектах. Любая физическая вещь может быть представлена в информационном мире с помощью одной или нескольких виртуальных вещей, однако виртуальная вещь может существовать и без какой бы то ни было соответствующей ей физической вещи.

В платформе ThingWorx такие «цифровые двойники» называются **Things (вещи)**. Они находятся в разделе **Modeling**.

Создание приложения Интернета вещей на платформе ThingWorx состоит из трех основных этапов:

1. Создание виртуальных вещей и организация взаимодействия между ними. Этот пункт соответствует разделу **Modeling**.
2. Визуализация необходимых данных на веб-странице. Этот пункт соответствует разделу **Visualization**.
3. Организация взаимодействия виртуальных вещей с физическими. Этот пункт частично соответствует разделу **Security**, частично **Modeling**.

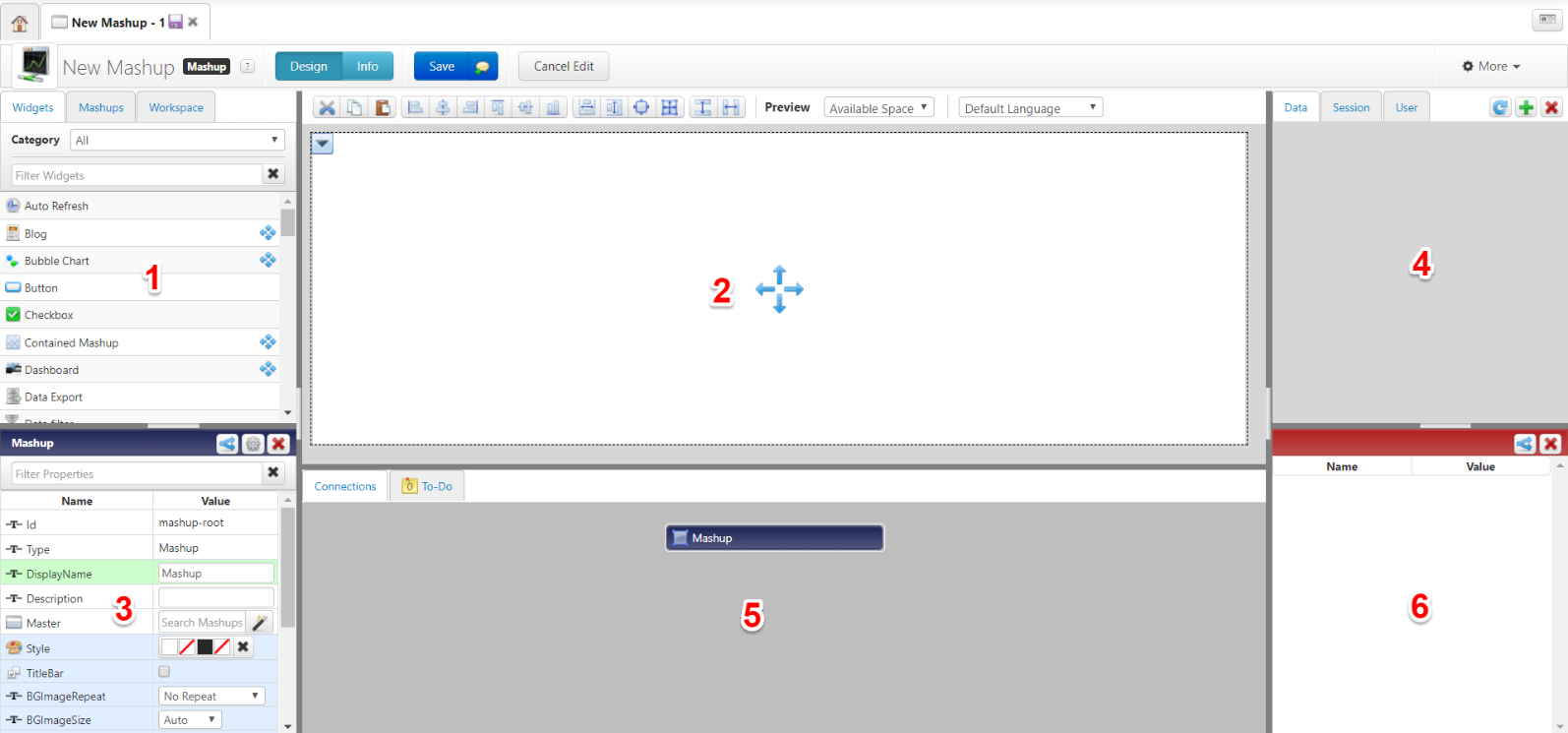
В результате получается система, в которой поведение физических объектов задается поведением их цифровых двойников, а вся необходимая информация отображается и задается на веб-странице.

## Mashup Builder

Mashup Builder (конструктор мэшапов) – “drag-and-drop” инструмент для создания графических веб-интерфейсов приложений Интернета вещей - мэшапов. Конструктор мэшапов позволяет объединять данные из совершенно разных источников и создавать интерактивные веб-интерфейсы на их основе.

**Структура конструктора мэшапов**

Окно конструктора мэшапов состоит из шести областей



|  |  |  |
| --- | --- | --- |
|  | **Область** | **Описание** |
| 1 | Виджеты | Во вкладке виджеты (Widgets) расположен список всех доступных графических элеметов платформы – виджетов. |
| 2 | Рабочая область (Canvas / «канва») | Область, на которую способом «drag-and-drop» перемещаются виджеты и размещаются в соответствии и желаемым внешним видом пользовательского интерфейса. Затем, виджеты рабочей области также способом «drag-and-drop» связываются с данными из ThingWorx Composer. |
| 3 | Параметры виджетов | В этой области отображаются и задаются параметры выбранного виджета. Например, здесь можно задать размеры, точное положение или стиль виджета. |
| 4 | Данные | Во вкладке данные (Data) добавляются данные из ThingWorx Composer и затем связываются с виджетами на рабочей области мэшапа. Например, можно добавить цифровую модель – Thing - и стандартный сервис для получения данных – GetProperties – и с помощью этого сервиса выдавать данные цифровой модели на мэшапе. |
| 5 | Связи | На этой области в удобном виде показываются все связи между данными и виджетами. Например, если нажатие на кнопку – виджет Button – вызывает сервис SetProperties, то на этой области вы увидите связь, идущую от события Clicked виджета Button к сервису SetProperties. |
| 6 | Параметры данных | Здесь показываеются параметры выбранного элемента данных. Это может быть полезно, если вы, например, хотите, чтобы один сервис запустился только когда закончится выполнение другого сервиса. Для такого сценария в этой области есть событие ServiceInvokeCompleted. |

# Моделирование

В этом разделе мы создадим модель приложения в среде разработки ThingWorx Composer.

## Подготовка контекста проекта

*Создание «проекта», тегов модели: словаря и терминов*

Создадим новый объект «проект» (Project) для объединения всех объектов, которые относятся к данному проекту.

1. На вкладке **Home** (Домой), разделе **Modeling** (Моделирование), выберите **Projects** и нажмите **+** справа.
2. В поле **Name** напишите **PlantMonitoringProject** и нажмите **Save**.

## Задание моделей данных

*Создание шаблонов вещей (Thing Templates), образов данных (Thing Shapes) и их наполнение: создание свойств, сервисов, событий, подписок*

В этом проекте никаких моделей данных не требуется. Все объекты будут созданы на основе стандартных шаблонов. В частности, модель растения будет создана на основе стандартного шаблона GenericThing.

## Создание объектов на основе моделей данных

*Создание вещей и добавление к ним свойств*

1. **Создание вещи**

Создадим цифровую модель растения (Thing).

1. На вкладке **Home** (Домой), разделе **Modeling** (Моделирование), выберите **Things** (Вещи) и нажмите **New** (Новая).
2. Введите имя **PlantThing** в поле Name.
3. В поле **Project** выберите проект **PlantMonitoringProject**.
4. В поле **ThingTemplate** напишите **generic** и выберите общий шаблон **GenericThing**.
5. **Создание «потока значений» (Value Stream)**

Теперь создадим объект «поток значений» (Value Stream) для логирования значений влажности почвы.

1. На вкладке **Home** (Домой), разделе **Data Storage** (Хранение данных), выберите **Value Streams** и нажмите **+** справа.
2. В появившемся окне (Choose Template) оставьте параметры по умолчанию (Value Stream) и нажмите **Choose**.
3. Введите **PlantThingValueStream** в поле Name.
4. В поле **Project** выберите проект **PlantMonitoringProject**.
5. Нажмите **Save** вверху.
6. **Задание свойств вещи**

Зададим свойства цифровой модели горшка с растением. Для этого:

1. Перейдите на вкладку **Properties**.
2. Нажмите **+Add My Property.**
3. Добавьте следующие свойства. **При создании всех свойств ставьте галочку напротив параметра Persistent** (раздел Aspects в правой части экрана).
   1. **PlantName** – название растения, тип **String.**
   2. **SoilMoisture –** влажность почвы, тип **Number.**
   3. **SoilMoistureLower –** нижнее критическое значение влажности почвы, тип **Number.**
   4. **SoilMoistureUpper –** верхнее критическое значение влажности почвы, тип **Number.**
   5. **SoilMoistureStatus –** состояние влажности почвы:

0 – норма, -1 – ниже критической, 1 – выше критической, тип **Integer**.

* 1. **SoilMoistureStatusMessage** – сообщение о состоянии влажности почвы (“Влажность почвы в норме” / “Влажность почвы ниже критической” / “Влажность почвы выше критической”), тип **String.**
  2. **PlantStatusMessage** – сообщение о состоянии растения (“Все в порядке” / “Необходим полив” / “Плохие условия: слишком влажно”), тип **String**.
  3. **ImageURL** - URL изображения растения для показа на веб-странице, тип **String**.
  4. **Latitude\_geo** – широта, тип **String.**
  5. **Longitude\_geo –** долгота, тип **String.**
  6. **CityName\_geo** – ближайший город (данные с сервиса openweathermap), тип **String.**
  7. **Humidity\_geo –** влажность воздуха (данные с сервиса openweathermap), тип **String.**

## Организация накопления и хранения данных

*Организация хранения данных в потоке значений (Value Stream) / в таблице данных (Data Table) / в информационной таблице (InfoTable)*

Организуем логирование значений влажности почвы в потоке значений (Value Stream).

1. Откройте вещь **PlantThing** и перейдите во вкладку **Properties**.
2. Перейдите к редактированию свойства **SoilMoisture** (карандаш слева от свойства) и поставьте галочку у **Logged**.
3. Нажмите **Done**.
4. Перейдите во вкладку **General Information** вещи.
5. В поле Value Stream напишите **plantmonitoring** и выберите **PlantMonitoringValueStream** во всплывающем окне Search Results.
6. Нажмите **Save** вверху экрана.

## Организация обработки данных

*Создание сервисов, задание программного кода*

1. **Создание сервиса для определения состояния растения по влажности почвы**

Создадим сервис, в котором в зависимости от значения влажности почвы - свойство SoilMoisture - будут меняться следующие параметры:

* состояние влажности почвы – свойство SoilMoistureStatus;
* сообщение о состоянии влажности почвы – свойство SoilMoistureStatusMessage
* сообщение об общем состоянии растения – свойство PlantStatusMessage.

Для этого:

1. Выберите вещь **PlantThing**.
2. Выберите **Services** (сервисы) из раздела **Entity Information** (Свойства записи).
3. Нажмите **+Add My Service**.
4. В поле **Name** введитеназвание сервиса **CheckPlantStatus.**
5. В поле **Description** напишите “Сервис для определения состояния влажности почвы и общего состояния растения”.
6. В поле **Script** напишите:

if ((me.SoilMoisture > me.SoilMoistureLower) && (me.SoilMoisture < me.SoilMoistureUpper))

{

me.SoilMoistureStatus=0;

me.SoilMoistureStatusMessage="Влажность почвы в норме";

me.PlantStatusMessage="Все в порядке";

}

else if (me.SoilMoisture < me.SoilMoistureLower)

{

me.SoilMoistureStatus=-1;

me.SoilMoistureStatusMessage="Влажность почвы ниже критической";

me.PlantStatusMessage="Необходим полив";

}

else if (me.SoilMoisture > me.SoilMoistureUpper)

{

me.SoilMoistureStatus=1;

me.SoilMoistureStatusMessage="Влажность почвы выше критической";

me.PlantStatusMessage="Пложие условия: слишком влажно";

}

1. Нажмите **Done.**
2. **Создание сервиса для получения геоданных**

В этом пункте создадим сервис, который будет по заданному местоположению определять ближайший город и влажность почвы в этой точки. Данные будут браться с сервиса openweathermap.

Прежде чем создать сервис, нужно получить доступ к сервису openweathermap. Для этого:

1. Зайдите на сайт <http://openweathermap.org>.
2. Зайдите во вкладку API.
3. Перейдите по ссылке API key в тексте (предложение “To access the API you need to sign up for an API key if you are on a free or paid plan” вверху страницы).
4. Нажмите Sign up и зарегистрируйтесь.
5. Зайдите во вкладку API Keys.
6. Скопируйте ключ.

Этот ключ нужно будет использовать в запросе, формируемом в Thingworx.

Теперь приступим к созданию сервиса.

1. Перейдите во вкладку **Services**.
2. Нажмите **+Add My Service**.
3. В поле **Name** напишите **getWeatherData.**
4. В поле **Description** напишите «Сервис для получения геоданных».
5. В поле **Script** (скрипт) напишите:

var params = {

url:

"http://api.openweathermap.org/data/2.5/weather?lat="+me.Latitude\_geo+"&lon="+

me.Longitude\_geo+"&type=accurate&mode=xml&APPID=**<ключ доступа к сервису OpenWeatherMap>"**

/\* STRING \*/,

timeout: 60 /\* NUMBER \*/

};

var xmlPage = Resources["ContentLoaderFunctions"].LoadXML(params);

me.CityName\_geo = xmlPage.city.@name;

me.Humidity\_geo = [xmlPage.humidity.@value](mailto:xmlPage.humidity.@value);

Вместо **<ключ доступа к сервису OpenWeatherMap>** напишите ваш ключ из п. 1).

1. Нажмите **Done** (готово).
2. Нажмите **Save** вверху экрана.

## Организация взаимосвязей и реализация бизнес-логики

*Создание событий, создание подписок, реализация правил*

1. **Создание подписки**

Сделаеем подписку, которая будет срабатывать при каждом изменении значения влажности почвы и запускать сервис CheckPlantStatus, который мы сделали в предыдущем пункте.

1. Выберите **Subscription** (подписки) из раздела **Entity Information** (Свойства записи).
2. Нажмите **+Add My Subscription**.
3. Поле **Source** оставьте пустым.
4. В поле **Event** выберите событие **DataChange.**
5. В появившемся поле **Property** напишите **soil** и выберите **SoilMoisture.**
6. Поставьте галочку напротив **Enabled** (включить подписку).
7. Кликните на поле **Script**.
8. Перейдите во вкладку **Me** и затем во вкладку **Services**
9. В строке для фильтра напишите **checkpl** и выберите сервис **CheckPlantStatus.**
10. Код для выполнения сервиса **CheckPlantStatus** появится в поле **Script**.
11. Нажмите **Done**.
12. Нажмите **Save** вверху экрана.

# Организация пользовательских взаимодействий

В этом разделе мы создадим пользовательский интерфейс на конструкторе мэшапов Mashup Builder.

Мэшап («сборка») – веб-страница для визуализации данных приложения Интернета вещей. Мэшап может объединять данные из нескольких совершенно разных источников.

## Создание мэшапа

Сделаем мэшап для вывода информации о растении. Вид итогового мэшапа показан в описании проекта в начале этого документа. Для того, чтобы создать мэшап:

1. На вкладке **Home** (Домой), разделе **Visualization** (Визуализация), выберите **Mashups** (Мэшапы) и нажмите **New** (Новый).
2. Оставьте предопределенные настройки (Mashup: Page, Layout Options: Responsive) и нажмите **Done**.
3. Нажмите кнопку **Info** (Информация) вверху экрана.
4. Введите имя **PlantMonitoringMashup** в поле Name.
5. В поле **Project** выберитепроект **PlantMonitoringProject.**
6. Нажмите **Save** вверху экрана.
7. Нажмите **Edit** (Редактировать) вверху экрана, затем нажмите **Design**.

## Добавление виджетов к мэшапу

Добавим виджеты на рабочу область мэшапа (Canvas).

1. На вкладке **Виджеты** (Widgets) (в левой части экрана), отфильтруйте или найдите виджет **Layout** (Разметка).
2. Перетащите его с вкладки Widgets (Виджеты) на рабочую область в центре экрана. Оставьте предопределенные настройки (Layout: Horizontal, Options: 2 Columns, галочки у Sidebar не ставить) и нажмите **Done** (Готово).
3. Выберите левый прямоугольник на рабочей области и уменьшите его ширину.
4. Перетащите ещё один виджет **Layout** (Разметка) с вкладки Widgets (Виджеты) на левый прямоугольник рабочей области. Выберите вертикальное расположение и оставьте настройки расположения по умолчанию (Two rows, галочки у header и footer не ставить).
5. Перетащите ещё один виджет **Layout** (Разметка) с вкладки Widgets (Виджеты) на правый прямоугольник рабочей области. Выберите вертикальное расположение и оставьте настройки расположения по умолчанию (Two rows, галочки у header и footer не ставить).
6. Увеличьте размер верхних прямоугольников на рабочей области по вертикали.
7. Перетащите ещё один виджет **Layout** (Разметка) с вкладки Widgets (Виджеты) на правый нижний прямоугольник рабочей области. Оставьте предопределенные настройки (Layout: Horizontal, Options: 2 columns, галочки у Sidebar не ставить) и нажмите **Done** (Готово).
8. На вкладке **виджеты** (Widgets) отфильтруйте или найдите виджет **Image**.
9. Перетащите его на верхний левый квадрат на рабочей области.

Виджет Image будет использоваться для показа изображения растения.

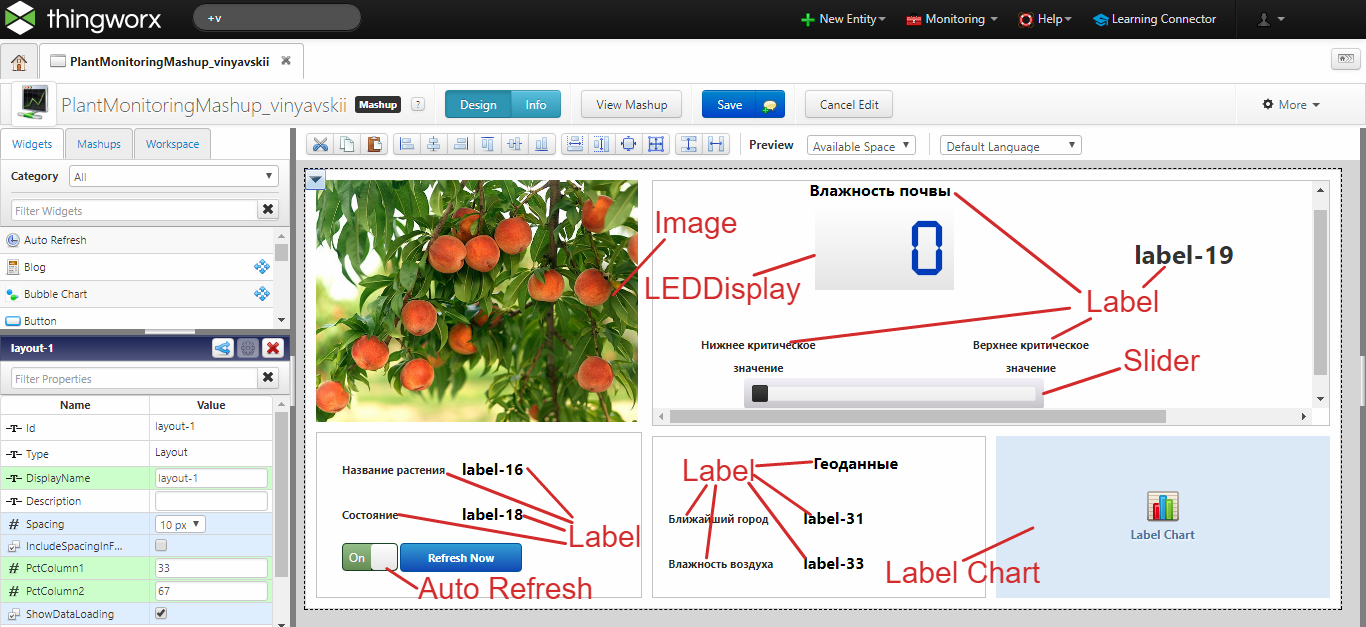
1. На вкладке **виджеты** (Widgets) отфильтруйте или найдите виджет **Label Chart.**
2. Перетащите его на нижний левый квадрат на рабочей области.

Виджет Label Chart будет использоваться для показа графика изменения влажности почвы от времени.

1. На вкладке **виджеты** (Widgets) отфильтруйте или найдите виджет **Panel** (панель)**.**
2. Перетащите его на нижний левый квадрат на рабочей области.

Виджет Panel – это “подложка” под все остальные виджеты области.

1. Соберите мэшап, как на картинке, перетащив остальные виджеты на рабочую область.



Для этого:

* 1. Перетащите оставшиеся виджеты на рабочую область (их названия подписаны на картинке красным).
  2. Кликните на виджет Label для отображения слова **“Влажность почвы”**.
  3. В области с параметрами виджета (левая нижняя область экрана) найдите поле **Text**, напишите там текст **Влажность почвы** и нажмите **Enter**.
  4. Ниже в области с параметрами виджета найдите поле **Style**, нажмите на крестик справа, затем на волшебную палочку.
  5. В появившемся окне выберите стиль текста из списка или нажмите **+Custom** и создайте свой.
  6. Выполните пункты (b.) – (e.) для всех виджетов **Label**, у которых хотите изменить текст и/или стиль текста.
  7. Кликните на виджете **Auto Refresh** (левый нижний прямоугольник рабочей области).
  8. В области с параметрами виджета (левая нижняя область экрана) найдите поле **RefreshInterval** (интервал между обновлениями в секундах), напишите там **3** и нажмите **Enter**.

## Связывание виджетов и сервисов

Для отображения информации на виджетах нужно связать их с данными из ThingWorx Composer. Для этого:

1. **Связывание с данными вещи PlantThing**
2. На вкладке **Data** (правая часть экрана), нажмите на зеленый "плюс".
3. В появившемся окне в поле **Select Entity** начните писать имя вещи, которую вы создали (**PlantMonitoringThing**) и выберите ее из списка результатов поиска (Search Results).
4. В поле **Select Services**, напишите **getproperties** для фильтра сервиса получения свойств GetProperties**.**
5. Нажмите на голубую стрелку, указывающую вправо, чтобы выбрать сервис получения свойств **GetProperties.**
6. Поставьте галочку у **MashupLoaded?** (запускать сервис при загрузке мэшапа) и нажмите **Done**.
7. В правой части экрана разверните красный заголовок, нажав на знак **+** слева от **GetProperties**.
8. Привяжите виджет **LEDDisplay**:
   1. Перетащите **SoilMoisture** под красным сервисным заголовком “GetProperties” на виджет **LEDDisplay** на правом верхнем прямоугольнике рабочей области.
   2. Выберите **Data** как цель привязки (Binding Target) на всплывающем окне.
9. Привяжите виджет **LEDDisplay**:
   1. Перетащите **SoilMoisture** под красным сервисным заголовком “GetProperties” на виджет **LEDDisplay** на правом верхнем прямоугольнике рабочей области.
   2. Выберите **Data** как цель привязки (Binding Target) на всплывающем окне.
10. Привяжите виджет **Label** для отображения названиярастения:
    1. Перетащите **PlantName** под красным сервисным заголовком “GetProperties” на виджет **Label** (предназначенный для отображения названия растения) на левом нижнем прямоугольнике рабочей области.
    2. Выберите **Text** как цель привязки (Binding Target) на всплывающем окне.
11. Аналогично привяжите остальные виджеты **Label**:
    1. Виджет **Label** для отображения состояния растения свяжите со свойством **SoilMoistStatusMessage**.
    2. Виджет **Label** для отображения состояния влажности почвы свяжите со свойством **PlantStatusMessage**.
12. Привяжите виджет **Slider**:
    1. Перетащите **SoilMoisture** под красным сервисным заголовком “GetProperties” на виджет **Slider** на правом верхнем прямоугольнике рабочей области.
    2. Выберите **Value** как цель привязки (Binding Target) на всплывающем окне.
    3. Перетащите **SoilMoistureLower** под красным сервисным заголовком “GetProperties” на виджет **Slider** на правом верхнем прямоугольнике рабочей области.
    4. Выберите **Minimum** как цель привязки (Binding Target) на всплывающем окне.
    5. Перетащите **SoilMoistureUpper** под красным сервисным заголовком “GetProperties” на виджет **Slider** на правом верхнем прямоугольнике рабочей области.
    6. Выберите **Maximum** как цель привязки (Binding Target) на всплывающем окне.
13. Привяжите виджет **AutoRefresh**:
    1. Кликните на виджете **AutoRefresh** на рабочей области.
    2. Наведите на стрелочку в левом верхнем углу виджета.
    3. Выберите событие **Refresh** и перетащите его на красный сервисный заголовок **GetProperties** на вкладке Data (правая часть экрана).
14. **Связывание с данными из «потока значений» (Value Stream)**
15. На вкладке **Data** (правая часть экрана)нажмите плюс справа от черного блока **Things\_PlantThing.**
16. В появившемся окне в поле **Select Services**, напишите **queryprop** для фильтра сервиса получения свойств **QueryPropertyHistory.**
17. Нажмите на голубую стрелку, указывающую вправо, чтобы выбрать сервис получения свойств QueryPropertyHistory**.**
18. Поставьте галочку у **MashupLoaded?** (запускать сервис при загрузке мэшапа) и нажмите **Done**.
19. В правой части экрана разверните красный заголовок, нажав на знак **+** слева от **QueryPropertyHistory** (если не нажато).
20. Перетащите **All Data** на виджет **LabelChart** (правый нижний прямоугольник рабочей области).
21. Выберите **Data** как цель привязки (Binding Target) на всплывающем окне.
22. Измените свойства виджета **LabelChart** (левая нижняя часть экрана):
    1. В поле **ChartType** выберите **Line/Marker.**
    2. В поле **XAxisField** выберите **timestamp.**
    3. В поле **XAxisFormat** напишите **hh:mm:ss a.**
    4. В поле **XAxisLabelType** выберите **Date/Time**.
    5. В поле **DataField1** выберите **SoilMoisture.**
    6. В поле **DataLabel1** напишите **Температура почвы.**
23. Привяжите виджет **AutoRefresh** к сервису QueryPropertyHistory:
    1. Кликните на виджете **AutoRefresh** на рабочей области.
    2. Наведите на стрелочку в левом верхнем углу виджета.
    3. Выберите событие **Refresh** и перетащите его на красный сервисный заголовок **QueryPropertyHistory** на вкладке Data (правая часть экрана).
24. Нажмите **Save** вверху экрана.
25. Нажмите **Cancel Edit** вверху экрана.
26. Нажмите **View Mashup** вверху экрана.

В браузере откроется новая вкладка с веб-страницей приложения.

**Проверка связей**

В этом пункте проверим правильную работу связей между всеми компонентами приложения.

1. **Проверка связи между ThingWorx Composer и Mashup Builder**
2. Откройте вещь **PlantThing** и перейдите во вкладку **Properties**.
3. Нажмите **Set** справа от свойства **PlantName**.
4. В появившемся окне введите название растения **Персик** и нажмите **Set**.
5. Нажмите **Set** справа от свойства **SoilMoistureLower**.
6. В появившемся окне введите значение **50** и нажмите **Set**.
7. Нажмите **Set** справа от свойства **SoilMoistureUpper**.
8. В появившемся окне введите значение **70** и нажмите **Set**.
9. Нажмите **Set** справа от свойства **SoilMoisture**.
10. В появившемся окне введите значение **60** и нажмите **Set**.
11. Откройте вкладку в браузере с веб-страницей приложения и убедитесь, что значения, которые вы задали, изменились.
12. **Проверка правильной работы сервисов**
13. Перейдите во вкладку **Properties**.
14. Нажмите **Set** справа от свойства **Latitude\_geo**.
15. В появившемся окне введите название растения **56** и нажмите **Set**.
16. Нажмите **Set** справа от свойства **Longitude\_geo**.
17. В появившемся окне введите значение **38** и нажмите **Set**.
18. Перейдите во вкладку **Services**.
19. Нажмите **Test** справа от сервиса **getWeatherData**.
20. В появившемся окне нажмите **Execute Service** внизу.

При успешном выполнении сервиса появится надпись **Results**.

1. Нажмите **Close.**
2. Снова перейдите во вкладку **Properties**.
3. Нажмите на голубую стрелочку справа от Value для обновления значений свойств.
4. Убедитесь, что свойство **CityName\_geo** приняло значение **Komyagino**, асвойство **Humidity\_geo** – какое-то число от 0 до 100.

# Отладка приложения

В этом разделе будет произведена отладка приложения при помощи симулирования внешних источников данных.

## Составление перечня

В этом разделе в дальнейшем будет показано, как составлять «перечень» для отладки приложения.

## Тестирование приложения с помощью симулятора

Протестирует работу приложения, используя в качестве внешнего источника данных программу-симулятор ThingWorx Academic Simulator.

Для тестирования необходимо будет:

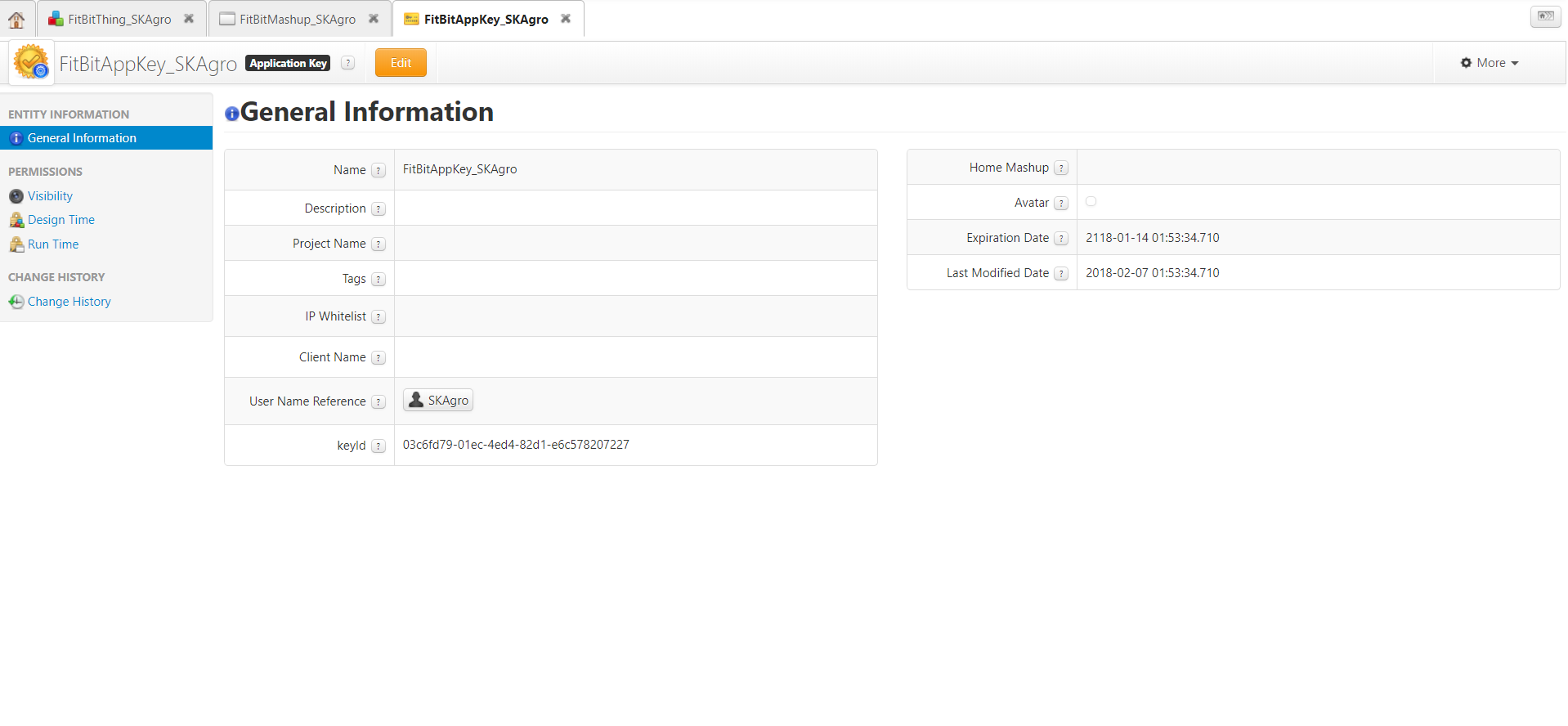
1. в платформе создать ключ приложения (Application Key) для связи программы ThingWorx Academic Simulator с приложением;
2. скачать и настроить программу ThingWorx Academic Simulator, сгенерировать с помощью нее данные, отправить их в приложение и убедиться, что приложение работает корректно.
3. **Создание ключа приложения**

Для обеспечения безопасности передачи данных, в ThingWorx используются ключи приложения (Application Keys). Чтобы подключиться к приложению, источник данных должен «знать» такой ключ.

Для создания ключа приложения:

1. В поисковой строке в верхней части экрана напишите **“+a”**и выберите команду**+Application Key**.
2. В поле **Name**в появившемся окне напишите **FitBitAppKey**.
3. В поле**User Name Reference**начните писать ваш логин и выберите появившийся объект.
4. Нажмите **Save**вверху экрана.

После сохранения в поле **keyId**появится ключ.



1. **Скачивание и настройка программы ThingWorx Academic Simulator**
2. Скачайте программу по адресу <http://apps.ptc.com/schools/software/ThingWorxAcademicSimulatorSetup.exe>

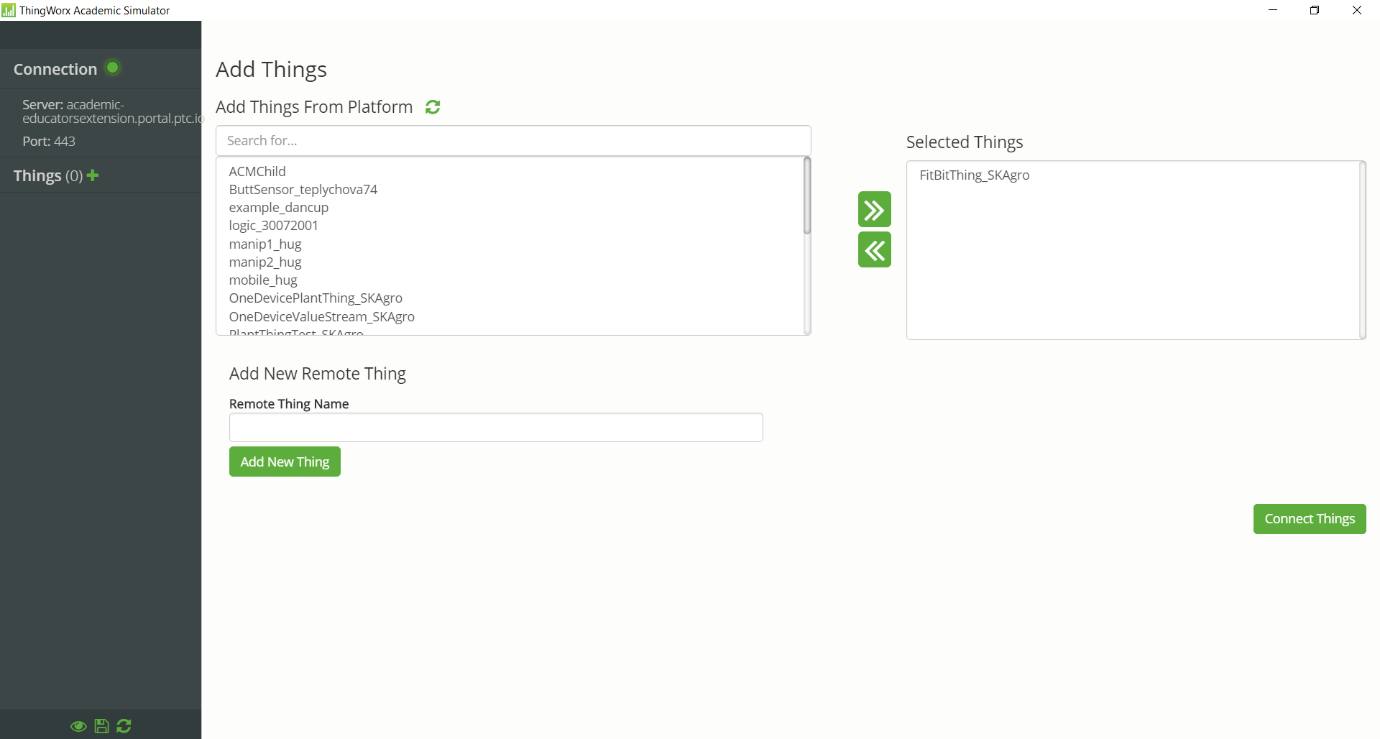
и установите.

1. Для работы в режиме симуляции выберите **Simulate Device** при запуске.
2. В поле **Server URL** ведите адрес сервера ThingWorx, на котором вы работаете (адрес до /Thingworx, **не включая** /Thingworx) **без протокола** (http://, https://).

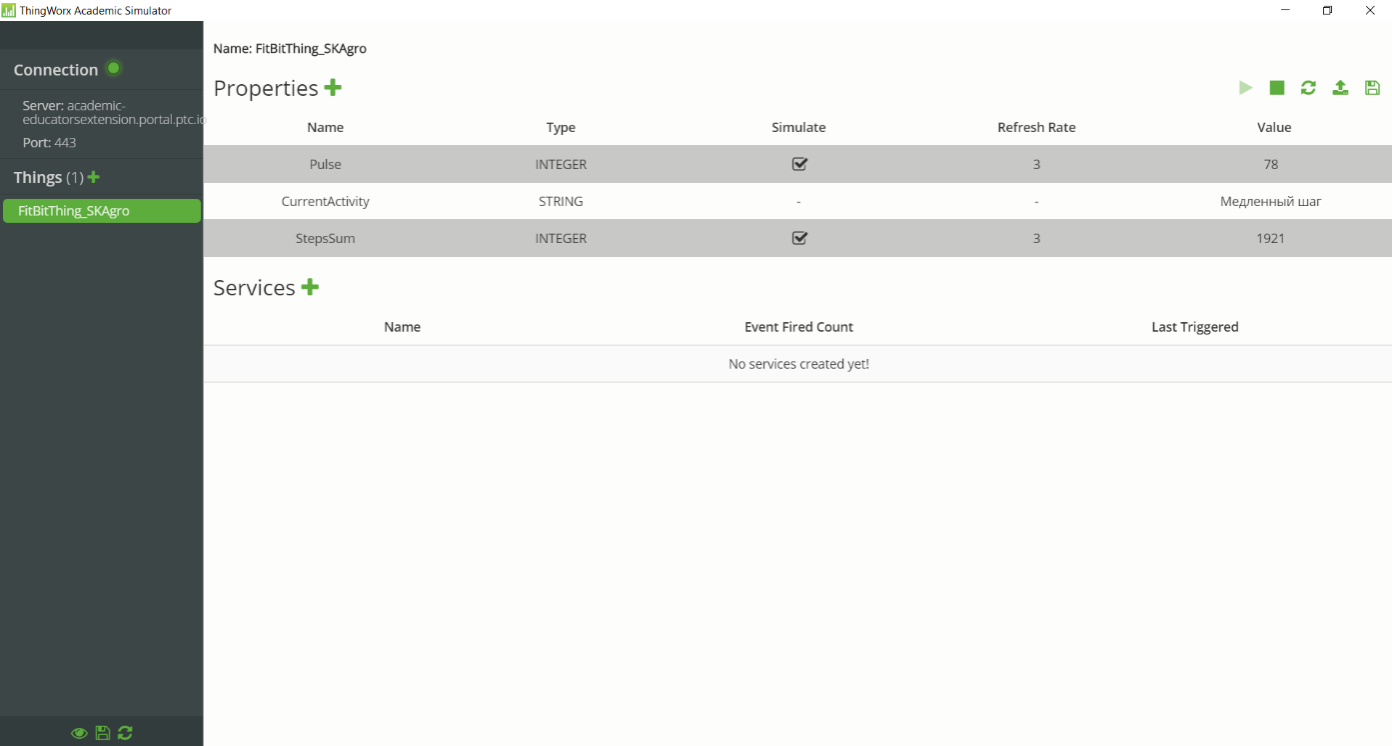
При работе с учебной версией платформы вводите

*academic-educatorsextension.portal.ptc.io.*

1. В поле **Application Key** введите ключ: код в поле keyId ключа, созданного в предыдущем пункте.
2. В поле **Port** введите **443**, если используется протокол **https** (в адресе сервера платформы). Если используется протокол **http**, в поле Port напишите **80**.
3. Нажмите **Connect to ThingWorx**.
4. В появившемся окне нашмите **+** справа от Things.
5. Выберите вещь **PlantThing** из списка и нажмите **>>** справа.

****

1. Нажмите **Connect Thing**.
2. Выберите свойство **SoilMoisture.**
3. Выберите тип симуляции (Simulation Type) **Random**.
4. В поле **Minimum Value** напишите 10.
5. В поле **Maximum Value** напишите 100.
6. В поле **Refresh Rate** напишите **5** (интервал между обновлениями данных в секундах).
7. Поставьте галочку **Include in Simulation** и нажмите **Save Property**.
8. Нажмите вверху.

****

1. Откройте мэшап и убедитесь, что данные приходят и приложение функционирует в соответствии с «перечнем» для отладки.