aicas GmbH / Fischertechnik

Sorting Line – Demo

Usage and Manual

Table of Contents

1) Preparation and Startup

Make sure that you have physical access to the Fischertechnik Sorting Line and its TXT controller (shown in Illustration 1). Make sure that you have a Notebook (or a Desktop PC) from which you will connect to the device and run the demo on Jamaica-IoT.

1.1. Power-On

Illustration 1: Fischertechnik TXT Controller

Press the on/off button and hold it (for at least a second) until the touch display turns on. The operating system and services need a little time (~ 30 sec.) until they are up and running.

1.2. Configure Network / WiFi (optional if not done yet)

For now we assume the network (LAN or WiFi) was configured in advance and is automatically set-up during system boot.

1.3. Connect to the Fischertechnik TXT Device

You need a remote terminal program on you PC. For Linux you can just use your standard Terminal with SSH. For Windows you explicitly need an (SSH) terminal client, e.g. Putty.

For Windows:

If you do not have Putty on your PC, please download and install it from:
https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

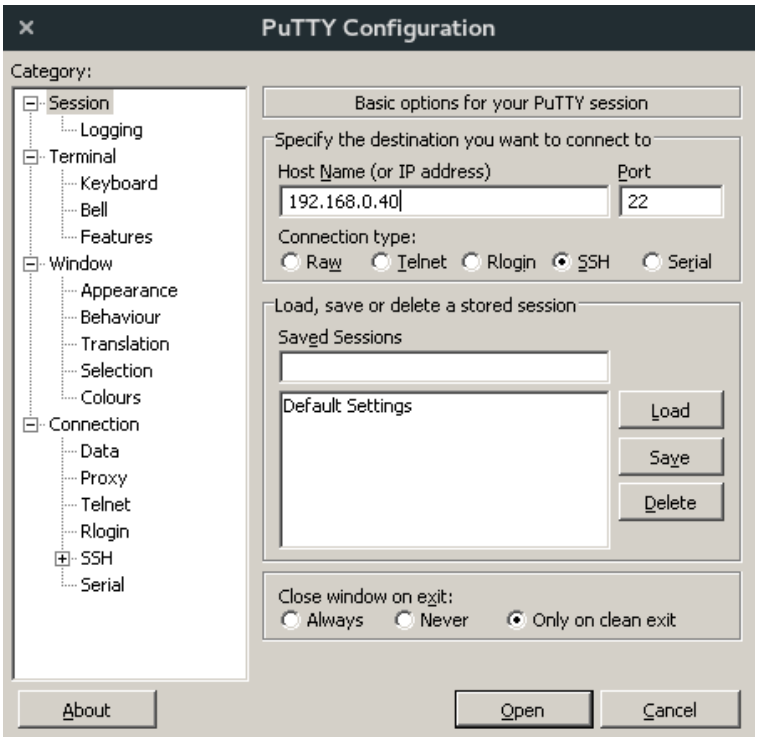


Illustration 2: Putty configuration

Start Putty → Enter the given IP address and press open → When requested please type the following user information:

Username: `root`
Password: `acc4txt%18`

For Linux:

Start a new Terminal (press Windows key on your keyboard, type “terminal”, press Enter) → Execute the following command:

`ssh root@192.168.0.40`

(expect a short delay of a couple of seconds ...)

Type as password: `acc4txt%18`

You terminal should be now connected to the TXT controller.

1.4. Mount the SD-Card

Jamaica-IoT as well as all the Demo components (bundles) are deployed on the SD-Card which is plugged into the TXT Controller.

The SD-Card must be initialized (mounted) to the OS file system. For that execute the following commands in the Terminal which you connected remotely to the TXT controller:

```
mount /dev/mmc… /mnt/sdcard1
mount /dev/mmc… /mnt/sdcard2
```

The deployed Jamaica-IoT framework can be then found under:

```
/mnt/sdcard2/opt/aicas/jiot
```

1.5. Start the Jamaica-IoT Framework

Execute the following commands in your remote terminal on the TXT controller:

```
cd /mnt/sdcard2/opt/aicas/jiot
```

```
./start
```

Wait until the Framework is up and running (i.e. there is no more new text coming out in the console). This might take up to 30 – 60 sec.

Note: A Jamaica-IoT (OSGi) command line prompt (a shell) should appear. Press a couple times Enter to see a new prompt on each line. Type "`lb`" and press Enter to see the state of the installed components in the running Jamaica-IoT instance . If you can see the bundle list, than your framework is up and running.

2) Using the Sorting Line Demo

Once Jamaica-IoT is up and running you can install and start the Sorting Line Demo. The Demo consists of several dependent components (bundles), which have to be installed and started in a particular order, e.g. from the app store provided by aicas. Which bundles must be deployed and executed, and how this is done from the app store, depends on the actual demo scenario, and will be explained further below.

Note: After start-up, Jamaica-IoT may already have the required Demo bundles installed (from its own cache) and available from previous runs. This should be okay. Depending on their last state (stopped or running) the components might be automatically started again. If everything required is already there, you should see a message like:

```
-----------------------------------------------
-           aicas standard sorting            -
-----------------------------------------------
```

or

```
-----------------------------------------------
-           aicas multiple sorting            -
-----------------------------------------------
```

and after that some other output like:

```
waiting for a new object ...
```

Then, respectively, the single or the multiple object sorting program is up and ready for experiments.

However, you will still need to understand the precise scenarios and the respective components setup in order to play-through the whole Demo.

2.1. Sorting Line Demo Scenarios

Typically we tell a common story about how the Demo was created and designed to show our technical strengths and easy development with JamaicaVM and Jamaica-IoT.

Standard Sorting

In a first setup we show that we were able to implement Fischertechnik's original Sorting Line functionality with (Real-Time) Java. For that we implemented the logic in several Jamaica-IoT components (bundles) which are separately deployed and executed on top of the framework on the target device (the TXT controller).

This demo showcases only the standard Fischertechnik logic, which can sort only one object put on the line in a predefined and static order, i.e. white, red and blue (in line rotation direction).

Multiple Sorting

This scenario is motivated by reaching a *higher throughput* on the sorting line, by parallelizing the sorting procedure. However, because the Fischertechnik Sorting Line was not designed for such an use case, we may encounter some interference on the color sensor, if objects are put-in too close. Please make sure to have at least 3 – 5 cm distance between the colored objects when putting them one after the other on the sorting line.

Multiple Sorting with Exchangeable Sorting Logic

Here we explain, that the sorting logic (push-out mechanism in particular color order) is exchangeable. It is implemented by a service and provided by a bundle (component) which can be easily exchanged during runtime. Our Multiple Sorting program is clever enough to track for the latest deployed Sorting Logic Version and to automatically wire (hook-up) against it. In this case during runtime we simply update the bundle holding the particular sorting logic with another implementation for a different color order, e.g. Blue – White – Red (in line rotation direction). Note: Once put in the belt, for each object a sorting logic is queried and referenced. Therefore, in our example, a logic exchange holds for all new objects placed on the line after a particular logic update.

2.2. Running the different Scenarios

Here we assume that you are familiar with our app store, its dashboard and in particular with:

- how to connect to it (with your web browser)
- how to introspect, install and update application components in the app store
- how to introspect the configuration of running devices

- how to install, start and stop components on a particular device
- how to upload a new component version to the app store
- how to update a component on a device from the app store

For more information on that please refer to Chapter 3.

After getting warm with our app store and succeeding with the tasks mentioned above, you should know enough to be able to run our Demo and implement the mentioned scenarios.

(a) Standard Sorting

Open in your web browser the device configuration of the TXT controller in the app store front end.

Note: make sure that (if installed) the `AicasTxtMultipleSorting` component is stopped.

For the standard sorting the following components must be installed and started in the specified order:

1. `AicasTxtDriverBundle` – implements a basic Java driver for controlling the Sorting Line hardware (motor, color sensor, light barriers, valves, compressor)
2. `SmackContainerBundle` – provides XMPP connection services for the application components
3. `AicasTxtStandardSorting` – this is the actual standard sorting program

If these bundles are already installed on the device, it is okay, just make sure that they are started in the given order.

When you start these bundles in Jamaica-IoT on the TXT controller, finally a message should appear in your remote terminal like:

```
-----------------------------------------------
-           aicas standard sorting            -
-----------------------------------------------
... (some other output) ...
waiting for a new object ...
```

Done. You can start now putting objects on the belt. Show that if you put several objects in parallel only the first one is sorted out and then the belt stops. (original Fischertechnik functionality)

(b) Multiple Sorting

Open the device configuration of the TXT controller in the app store front end.

Note: make sure that the `AicasTxtStandardSorting` component is stopped.

For the multiple sorting the following components must be installed and started in the specified order:

1. `AicasTxtDriverBundle` – implements a basic Java driver for controlling the Sorting Line hardware (motor, color sensor, light barriers, valves, compressor)
2. `SmackContainerBundle` – provides XMPP connection services for the application components

3. `AicasTxtSortingAPI` – An API bundle for the exchangeable sorting logic

4. `AicasTxtSortingLogic` – a particular sorting logic implementation
5. `AicasTxtMultipleSorting` – this is the actual standard sorting program

Note: the 4-th bundle (`AicasTxtSortingLogic`) can be implemented and provided within the app store (and the TXT device respectively) by one of the following JAR packages:

- `AicasTxtDefaultSortingLogic.jar`
- `AicasTxtBWRSortingLogic.jar`
- `AicasTxtRWBSortingLogic.jar`

For our actual use case make sure that the `AicasTxtDefaultSortingLogic.jar` has been loaded into the app store and installed on the device as `AicasTxtSortingLogic.`

If everything is started in correct order in your remote terminal you should see an output like:

```
-----------------------------------------------
-           aicas multiple sorting            -
-----------------------------------------------
... (some other output) ...

waiting for a new object …
```

Done. You can start now putting multiple objects in parallel on the belt. Make sure that there is a distance of at least 3 – 5 cm between each subsequent object, otherwise you may have problems with the color detection and the pushout mechanisms. The program sorts all objects on the line in parallel and terminates the line rotation (motor) after the last object has been pushed out.

(c) Multiple Sorting with Logic Exchange

Assuming that you have successfully established and presented the previous multiple sorting demo scenario, now you can exchange the sorting logic during runtime.

For that, perform the following steps while the multiple sorting program is running:

1. Open the app store dashboard and list the available components
2. Update the `AicasTxtSortingLogic` component by uploading another JAR file for its implementation, e.g. `AicasTxtBWRSortingLogic.jar` (Blue – White – Red sorting)
3. Go to the device list and introspect the configuration of the TXT device
4. Press “Update” on the `AicasTxtSortingLogic` bundle

Now the new logic has been installed and registered at the TXT controller device.

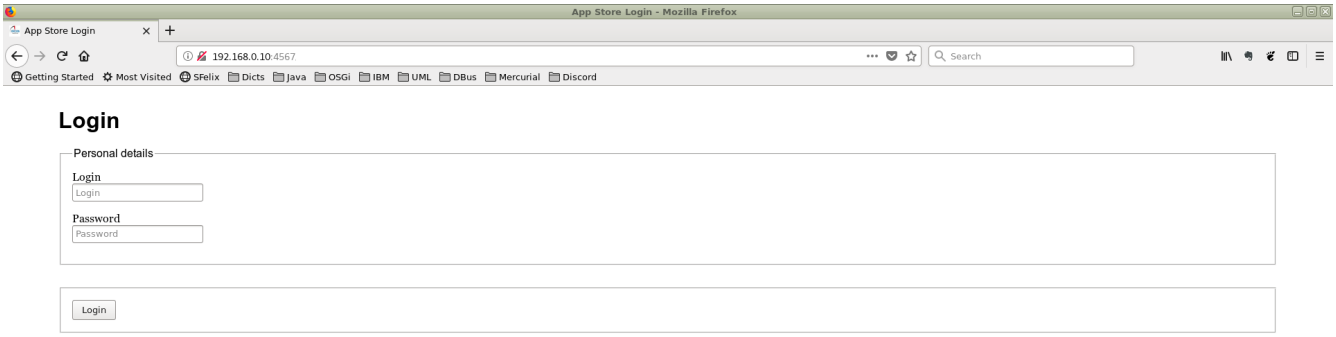
Note: Once put in the belt, for each object a sorting logic is queried and referenced. Therefore, in our example, a logic exchange holds for all new objects placed on the line after a particular logic update.

As a result, new objects put on the line would be sorted with the new sorting logic. The logic update happened seamlessly during runtime (at best while the motor and the line still rotate some other objects) without interruption of the device and its sorting program.

3) App Store Usage

3.1. Connect to the App Store

We assume the Fischertechnik TXT device and the Jamaica-IoT framework are up and running. For more information on how to do that, please refer to Chapter 1.



Open a web browser on your laptop / desktop PC.

Go to address: `192.168.0.10:4567`

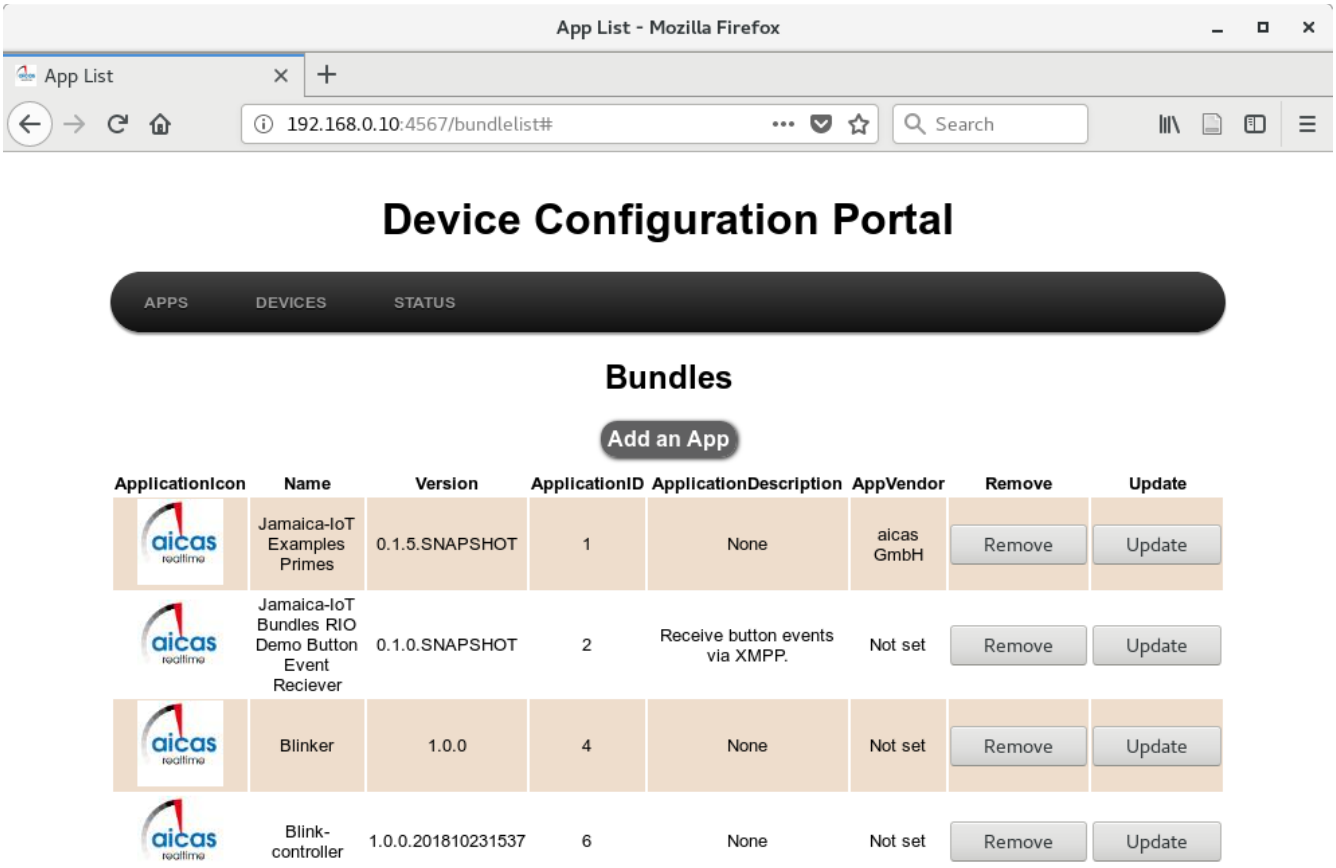
Type as login information:

Login: `aicas`

Password: `aicas`

After pressing “`Login`” you should connect to the main page of the app store.

3.2. Introspect, Install and Update Components in the App Store



On the Menu Bar please press on `Apps` / `Bundles`.

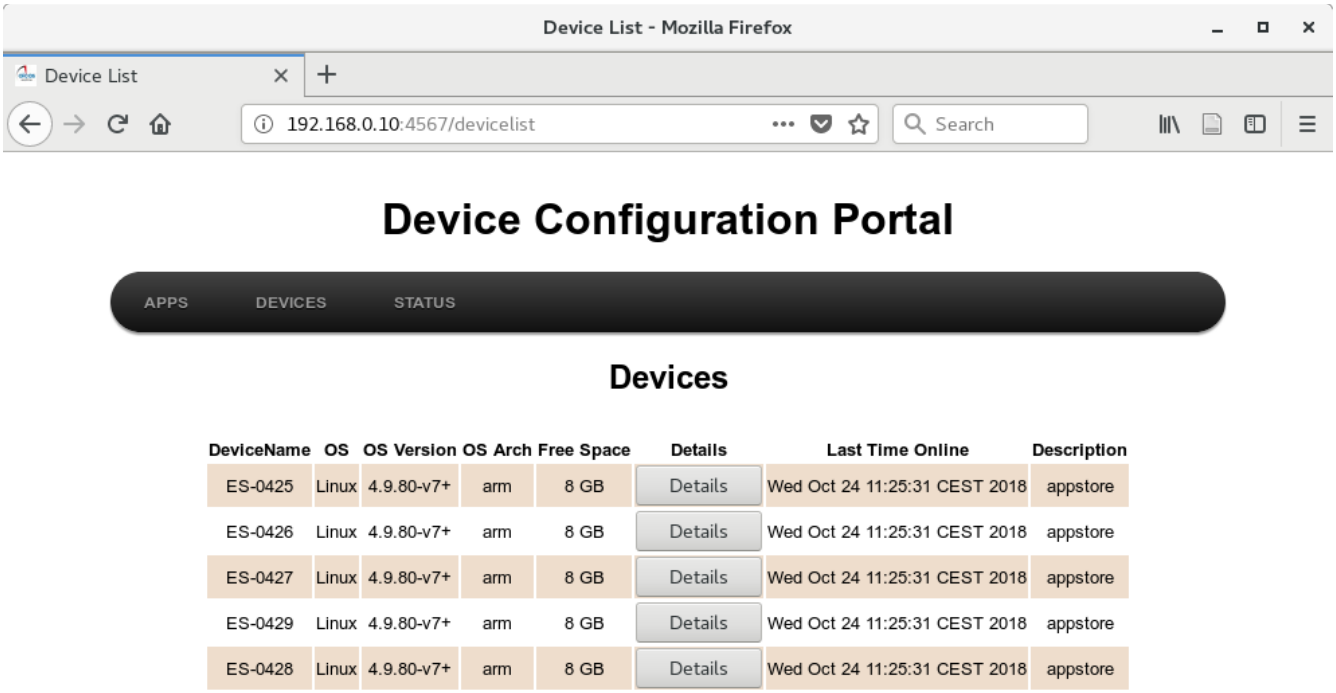
You should see the installed bundles in the App Store.

You can directly Remove or Update each installed bundle. The latter allows you to upload a new bundle JAR file from your local hard disk, which is then managed by the App Store in the background.

You can upload and install new components in the App Store by pressing “`Add an App`”.

After upload new components should appear in the App Store list.

3.3. Introspect Running Devices

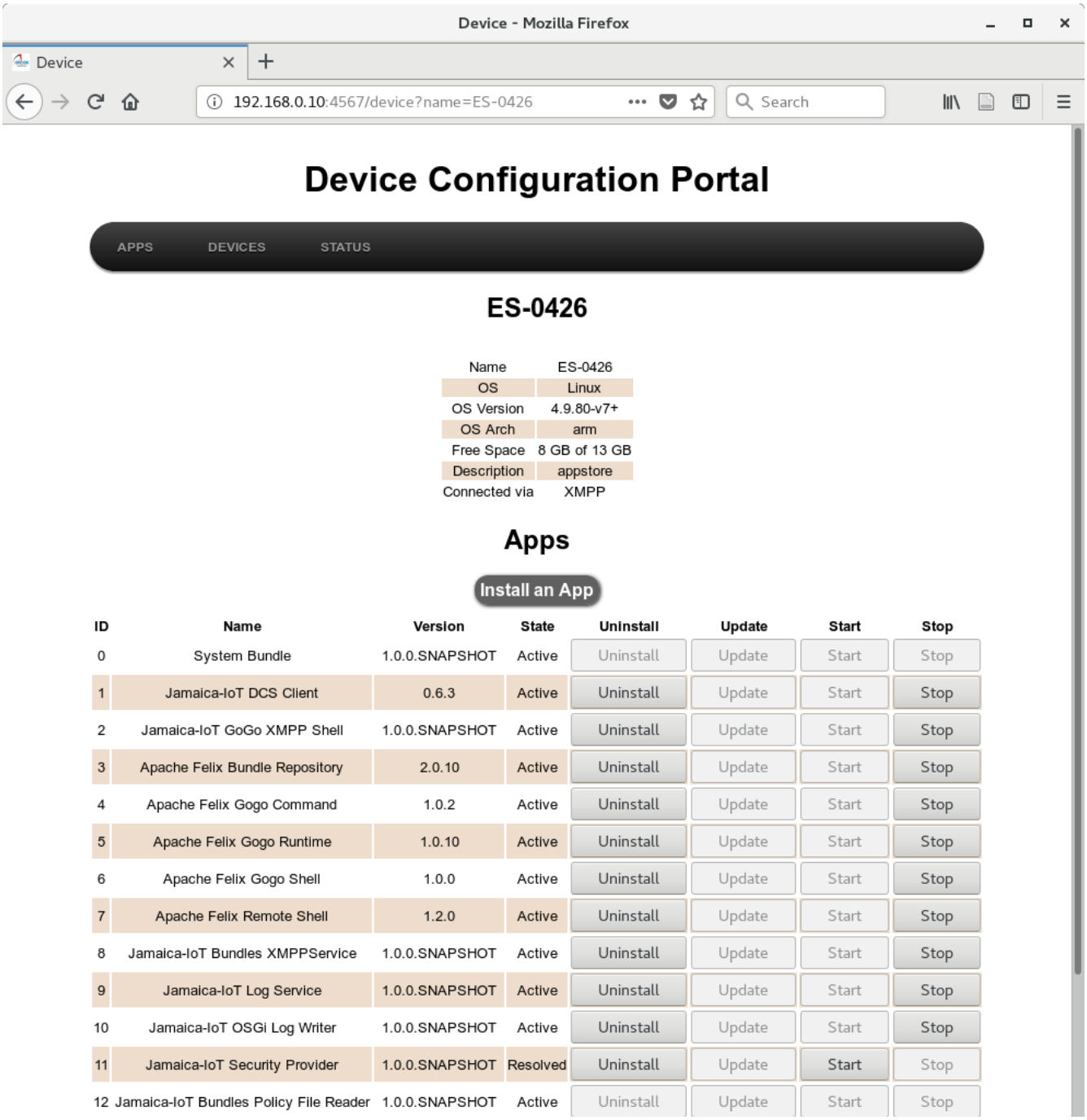


On the Menu Bar press “`Devices`”.

You should see a list of devices which are actually managed by the App Store.

For each device you can introspect its actual configuration by pressing “`Details`”.

3.4. Manage Device Configuration



By listing the "`Details`" of a device you can introspect its currently installed components and their states. A list of components appears with respective deployment and life cycle control action buttons for each component.

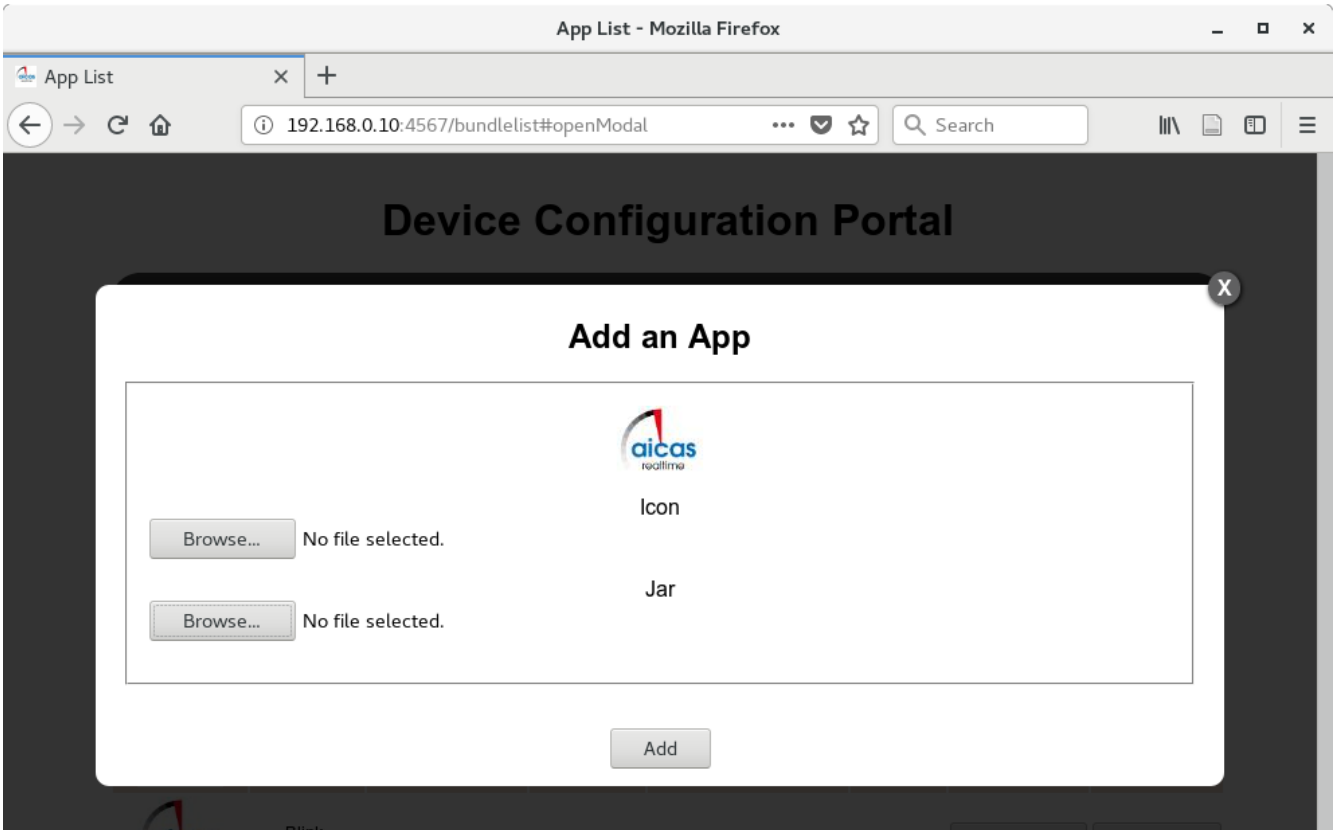
`Uninstall` – removes the component from the actual device
`Update` – updates the component on the device with the newest version available in the app store

`Start` – starts the component
`Stop` – stops the component

For our Fischertechnik Demo make sure that the following components are installed and available on the TXT device:

```
AicasTxtDriverBundle
SmackContainerBundle
AicasTxtSortingAPI
AicasTxtSortingLogic
AicasTxtStandardSorting
AicasTxtMultipleSorting
```

3.5. Upload/Update a Component in the App Store



Press `Apps` → `Bundles` on the Menu Bar.

You can add a new or update an existing component by pressing the respective buttons.

Consequently, the presented "`Add an App`" window pops-up, where you can specify a new component Icon image and a JAR file.

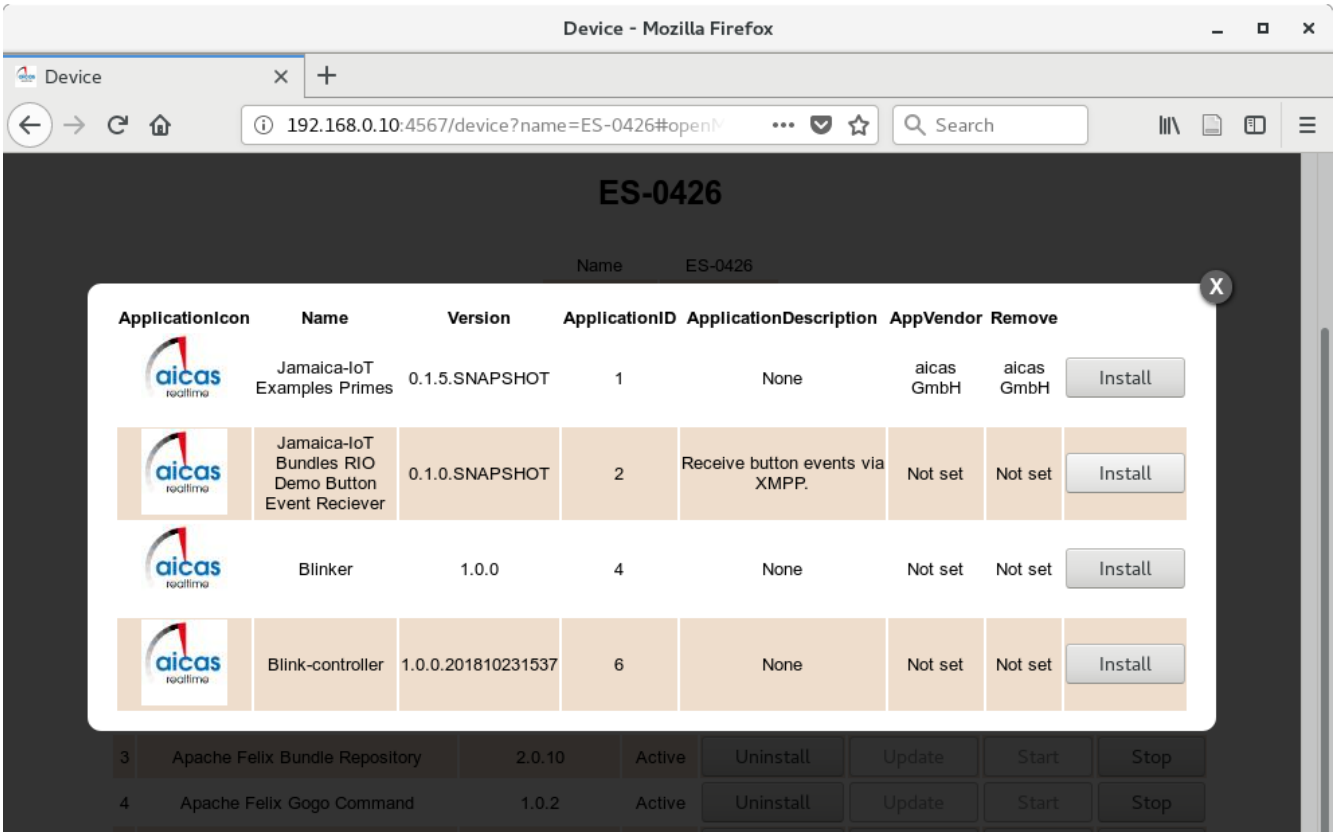
Please, leave the icon empty, if you do not have a special custom one.

Click on “Browse” for the JAR file to select a new component file and then press “Add”.

The new component should be installed in the App Store and visible in the components list.

Note: For the Fischertechnik Demo the component update procedure in the app store is important. As already explained, for the AicasTxtSortingLogic there are three different versions available as JAR-files (AicasTxtDefaultSortingLogic.jar, AicasTxtBWRSortingLogic.jar, AicasTxtRWBSortingLogic.jar). They implement different versions of the sorting logic. Our third demo scenario requires to update the actual AicasTxtSortingLogic version first in the app store and then on the TXT device during runtime.

3.6. Install/Update a Component on a Device



Go back to the devices list and the detailed configuration of a device.

You can install a component from the App Store on that Device by pressing “Install an App”.
The presented window pups-up where you can select an available component from the App Store to be installed on the device by pressing “Install”.
Your new component should then appear in the device configuration and its component list.

Update a Bundle on a Device

For updating bundles with their newest version from the App Store:

- Open the Device Configuration (`Devices → Details`)
- Press “`Update`” on the respective component
- the update is automatically performed in the background

Last steps are particularly important for the logic update scenario in our Fischertechnik Demo.