

Mobile Foundry

Getting Started Guide

Table of Contents

Draft 0.2	2
March 28 2018.....	2
Introduction.....	3
Mobile Foundry Application Flow	3
Creating Your First App	3
OEM information	3
Device Information.....	4
Further Reading.....	5

Introduction

Mobile Foundry is an exciting new release of Ayla's mobile applicaiton platform. It has been re-designed from the ground up to work with the latest Ayla Mobile SDK and provide the fastest IoT app development experience available.

Mobile applications used to provide an interface to connected devices all share many common tasks: Account management, device management, device control, scheduling, etc. While each application requires its own look and feel to distinguish it from others, the basic functionality still needs to be present and performant.

Mobile Foundry takes the approach that the underlying IoT infrastructure required by mobile applications can be handled by the framework, while developers are free to focus on user experience enhancements to make the application their own.

Mobile Foundry is a configuration-driven platform. Each application built with Mobile Foundry must have a configuration associated with it that defines the application flow, supported devices and information about what the supported devices do and how to interact with them.

Using Mobile Foundry, it is entirely possible to create a fully-functional mobile application that can interact with your own devices without writing a single line of code.

In addition to the stock screens and controls that ship with Mobile Foundry, developers are free to extend the framework to create new screens, controls or other functionality while still retaining the power and flexibility of defining the application through the configuration file.

Mobile Foundry Application Flow

Creating Your First App

The core functionality of Mobile Foundry is provided as a library which may be downloaded from Ayla's GitHub repository:

https://github.com/AylaNetworks/Android_Sepia

https://github.com/AylaNetworks/iOS_Sepia

Follow the instructions in the README file in the project to set up your application.

OEM information

Once your application is set up according to the instructions in the README file, you should have a configuration file that is being used by the application ("my_sepia_config.json" from the instructions, or a name of your choosing). This file will be read by the framework at application launch will contain all of the information necessary for the application to recognize and interact with your own devices.

The Mobile SDK requires some information regarding the OEM account and mobile application that is provided via the "SDKConfig" section in the configuration file. Update the appId and appSecret fields in this structure to match the values provided to your company by Ayla Networks:

```

"sdkConfig": {
  "appId": "sepiaapp-0dfc7900-id",
  "appSecret": "sepiaapp-0dfc7900-6s3Wn_kLZpbrV2ZomcCqK0EuIeQ",
  "allowMobileDSS": false,
  "allowOfflineUse": true,
  "serviceType": "Development",
  "serviceLocation": "USA",
  "consoleLogLevel": "Debug",
  "fileLogLevel": "Debug",
  "xPlatformID": "12345",
  "defaultNetworkTimeoutMs": 5000
}

```

Device Information

In addition to the OEM information for the mobile application, Mobile Foundry needs information about each of the device types that are supported by the application. This information is specified to Mobile Foundry in the "Devices" section of the configuration file. There should be exactly one entry in the Devices section for each type of device supported by the application.

Mobile Foundry matches devices in this section by the "oemModel" field. When the list of registered devices is fetched from the service, Mobile Foundry maps the oemModel of each device with an entry in the devices section of the configuration file. This allows Mobile Foundry to create different device class objects for each device type. Developers may extend the device classes with their own and map that class to their oemModel in this structure as well.

During development, it might be useful to have a "catch-all" device type that will be presented if no matching entry is found for a given oemModel. A wildcard "*" may be specified in the oemModel field for a device to indicate that this class should be used to represent devices that do not match any other oemModel in the devices array.

```

"devices": [
  {
    "class": "SepiaDevice",
    "icon": "ic_generic_device",
    "oemModel": "*",
    "detailScreen" : "device_details",
    "name": "generic_device"
  },
  {
    "class": "AylaSmartPlug",
    "icon": "ic_ayla_smart_plug",
    "oemModel": "smartplug1",
    "detailScreen" : "device_details",
    "managedProperties": [
      {
        "control": "generic_switch",
        "name": "outlet1",
        "notify": true,
        "schedule": true
      }
    ]
  }
]

```

```

},
{
  "class": "AylaEVBDevice",
  "name": "Ayla EVB",
  "detailScreen": "auto_device",
  "icon": "ic_ayla_evb",
  "oemModel": "leddevb",
  "scheduleScreen": "schedule",
  "ssidRegex": "Ayla-[0-9a-zA-Z]{12}",
  "managedProperties": [
    {
      "control": "led_light_blue",
      "name": "Blue_LED",
      "notify": true,
      "schedule": true
    },
    {
      "control": "led_light_green",
      "name": "Green_LED",
      "notify": true,
      "schedule": true
    },
    {
      "control": "blue_button",
      "name": "Blue_button",
      "notify": true,
      "schedule": false
    },
    {
      "control": "generic_number_control",
      "name": "decimal_out",
      "notify": "true",
      "schedule": "false"
    },
    {
      "control": "generic_number_control",
      "name": "decimal_in",
      "notify": "true",
      "schedule": "true"
    }
  ]
},

```

Details regarding the configuration file for Mobile Foundry can be found in the *Mobile Foundry Configuration Specification* document.

Further Reading

The overall architecture of Mobile Foundry can be found in the *Mobile Foundry Architecture* and *Mobile Foundry Overview* documents.

Details regarding the Mobile Foundry configuration JSON format can be found in the *Mobile Foundry Configuration Specification* document.

A guide to the user interface elements available in Mobile Foundry is also available, named *Mobile Foundry UI Reference*.

