Using MobileFirst application as a container for server-generated pages

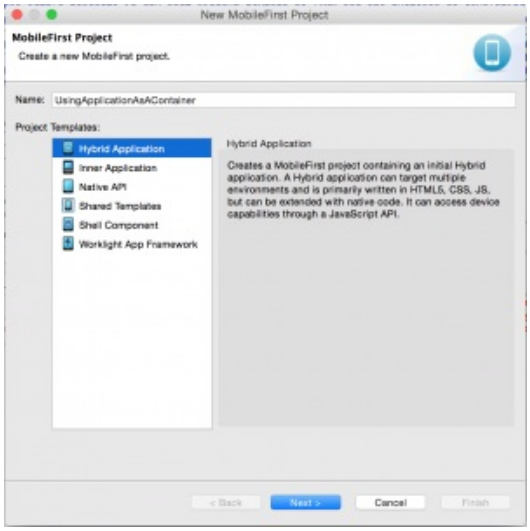
fork and edit tutorial (https://github.com/MobileFirst-Platform-Developer-Center/DevCenter/) | report issue (https://github.com/MobileFirst-Platform-Developer-Center/DevCenter/issues/new)

Migrating applications to IBM MobileFirst Platform Foundation

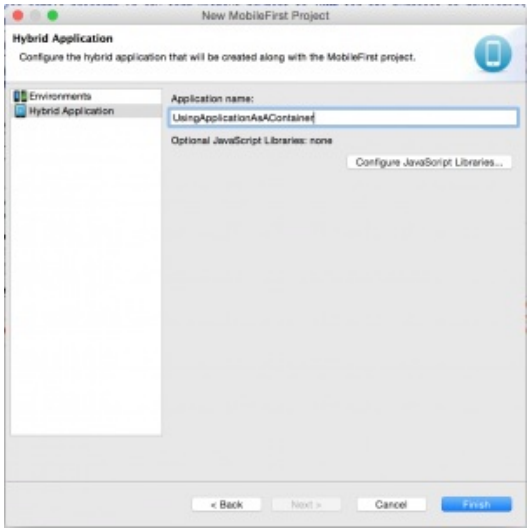
- By using mobile web technology, you can deploy applications to the widest variety of devices.
- The existence of public application stores, such as Apple iTunes and Google Play, changes the way applications are hosted and marketed. These changes make traditional methods of distribution less relevant.
- IBM MobileFirst Platform Foundation provides the solution to build cross-platform applications that can be distributed through the application stores by using the hybrid application programming model.
- In the hybrid model, developers typically package the application HTML, CSS, and JavaScript™ code as part of the application that is deployed to the application store.
- In this module, you see the remote loading of dynamic content capability, where the HTML, CSS, and JavaScript code is hosted externally from the natively packaged hybrid.

Creating IBM MobileFirst applications

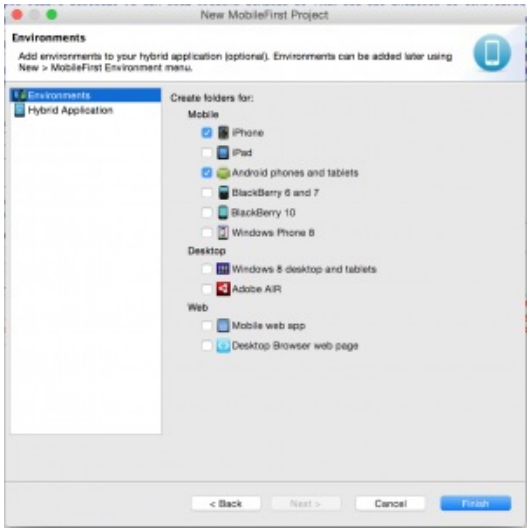
- A project and applications are created by using the IBM MobileFirst Project wizard.



- A project might host multiple applications. However, in this module, you use only one app: **UsingApplicationAsAContainer**

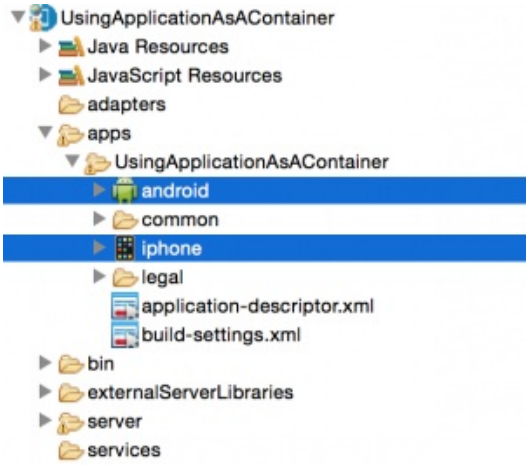


- Optionally the target environments can be set on the MobileFirst Project wizard or it could be done at a later time



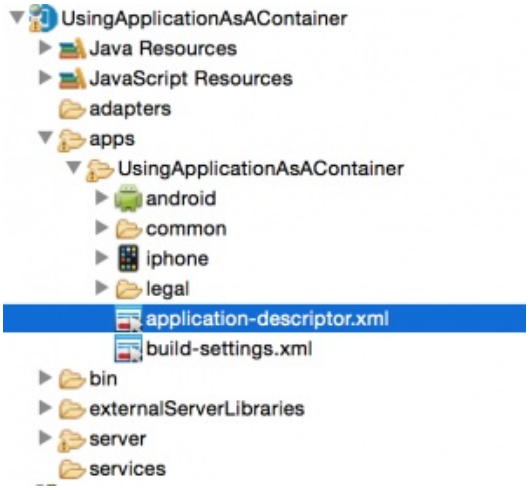
IBM MobileFirst environments

- IBM MobileFirst Foundation provides a model for organizing the application project structure for each targeted environment (for example, Android, iPhone, iPad).
- These targeted environments are selected through the MobileFirst Environment wizard.



IBM MobileFirst Common environment

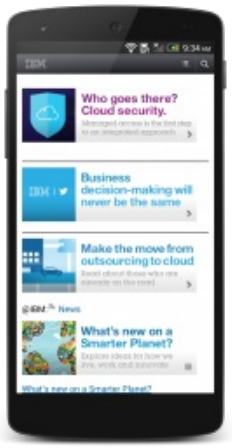
- The simplest way to use IBM MobileFirst apps as containers for server generated pages is through the Common environment
- Open the `application-descriptor.xml` file and edit the `mainFile` tag to point to "`http://m.ibm.com`"



```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<application xmlns="http://www.worklight.com/application-descriptor" id="UsingApplicationAsAContainer" platformVersion="6.
3.0.00.20141003-1438">
   <displayName>UsingApplicationAsAContainer</displayName>
   <description>UsingApplicationAsAContainer</description>
   <author>
      <name>application's author</name>
      <email>application author's e-mail</email>
      <homepage>http://mycompany.com</homepage>
      <copyright>Copyright My Company</copyright>
   </author>
   <mainFile>http://m.ibm.com</mainFile>
   <features/>
   <thumbnailImage>common/images/thumbnail.png</thumbnailImage>
   <iphone bundleId="com.UsingApplicationAsAContainer" version="1.0">
      <worklightSettings include="false"/>
      <security>
         <encryptWebResources enabled="false"/>
         <testWebResourcesChecksum enabled="false" ignoreFileExtensions="png, jpg, jpeg, gif, mp4, mp3"/>
      </security>
   </iphone>
   <android version="1.0">
      <worklightSettings include="false"/>
      <security>
         <encryptWebResources enabled="false"/>
         <testWebResourcesChecksum enabled="false" ignoreFileExtensions="png, jpg, jpeg, gif, mp4, mp3"/>
         <publicSigningKey/>
         <packageName/>
      </security>
   </android>
</application>
```

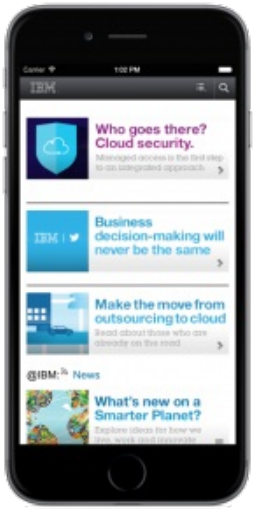
Running your application on the Android emulator

- Build the Android environment, and then deploy the application. Right-click the generated Android project and click **Run As** > **Android Application**.
- You can see that http://m.ibm.com is rendered in your Android emulator.



Running your Application on the iOS

- Deploy the application to your iOS emulator. Right-click the IOS environment, and click **Run As** > **Xcode project**.



Sample application

Click to download (http://public.dhe.ibm.com/software/products/en/MobileFirstPlatform/docs/v630/UsingWorklightApplicationAsAContainerProject.zip) the Studio project.