

# Working with UI frameworks

## Overview

Designing and implementing the UI of an application is an important part of the development process.

Writing custom CSS style for each component from scratch can deliver a high level of customization, but also requires a large amount of resources.

Sometimes it is better to use the existing JavaScript™ UI frameworks.

This module describes how to develop MobileFirst applications with two JavaScript UI frameworks jQuery Mobile and Dojo Mobile, by using the wizards that are provided in MobileFirst Studio.

This tutorial covers the following topics:

- Using jQuery in a MobileFirst application
- Working with jQuery Mobile
- Working with Dojo Mobile
- Working with Ionic framework

## Using jQuery in a MobileFirst application

jQuery is a fast and concise JavaScript framework that simplifies HTML document flow, event handling, animating, and Ajax interactions for rapid web development.

The MobileFirst client-side framework uses the jQuery library for internal functionality. By default, the `$` character is assigned to the internal jQuery in the main HTML file of the application, in the HEAD element:

```
<script>window.$ = window.jQuery = WLJQ;</script>
```

If the application does not require jQuery, or if you want to use a different version of jQuery, you can remove this line.

For more information, see jQuery (<http://jquery.com/>).



## Working with jQuery Mobile

jQuery Mobile is a touch-optimized web framework for smartphones and tablets.

jQuery Mobile requires jQuery to run.

The required jQuery Mobile components are in the `.js` and `.css` files provided by jQuery Mobile. More style images can optionally be added.

- When you create a new application, you can add the jQuery Mobile files from the **Configure JavaScript Libraries** dialog. You can use either the compressed jQuery Mobile `.zip` file or the extracted folder.
- You can then select the required files. Select at least the `images` folder and the `jquery-mobile-*.css` and `jquery-mobile-*.js` files.
- When you create the application, a `jqueryMobile` folder is added to contains these files. The main HTML file of the application is created with references to the `.css` and `.js` files and with a basic jQuery Mobile template.

## Basic structure of a screen

The basic structure of a screen in a jQuery Mobile application is as follows:

```

<div data-role="page"><br />
  <div data-role="header"> ... </div><br />
  <div data-role="content"> ... </div><br />
>
  <div data-role="footer"> ... </div><br />
</div>

```

- The header and footer elements are optional.
- The page is rendered by a jQuery Mobile at run time.
- All the required styles are automatically applied.

## WYSIWYG editor

A WYSIWYG editor is provided for jQuery Mobile widgets for developer convenience.



By using jQuery Mobile, you can create mobile application screens with only a few lines of code.

```

<div data-role="page"><br />
  <div data-role="header"><br />
    <h2>My Application</h2><br />
  </div><br />
  <div data-role="content"> This is the application body </div><br />
>
</div>

```



For more information, see the jQuery Mobile demos (<http://jquerymobile.com/demos>).



## Working with Dojo Mobile

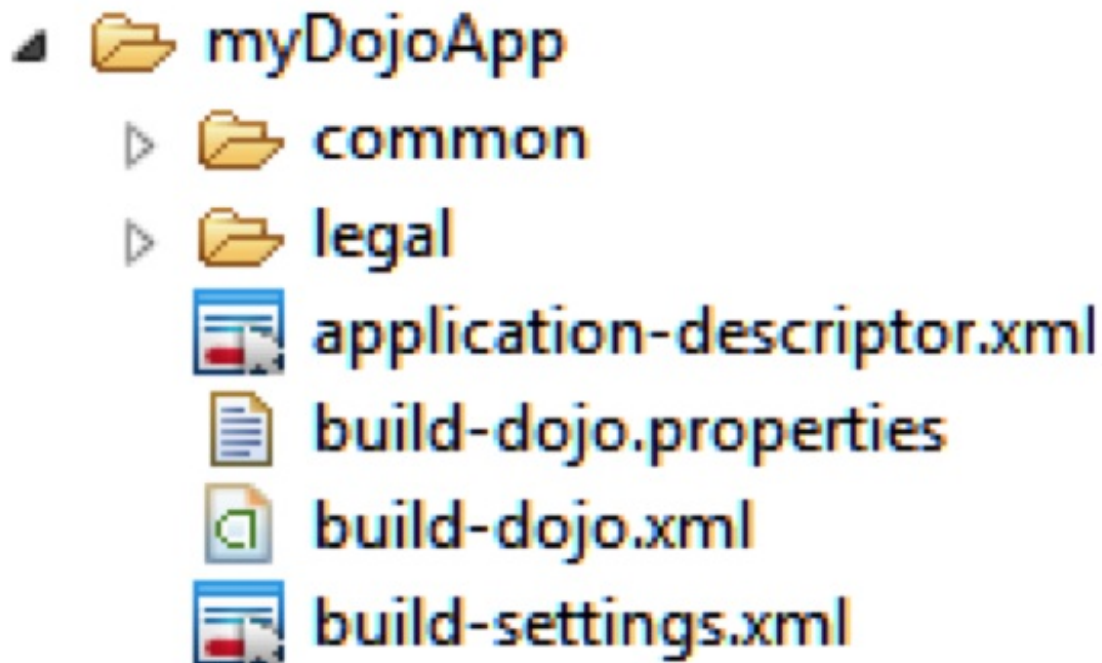
Dojo Mobile can be used to rapidly develop mobile web applications on iPhone, iPod Touch, iPad, Android, and BlackBerry touch devices, with the appearance of the native device.

Dojo Mobile is part of the Dojo Toolkit, which is developed and maintained by the Dojo Foundation. For more information about the Dojo Toolkit, see the Dojo Toolkit (<http://dojotoolkit.org/documentation>) website.

MobileFirst Studio includes a Dojo Toolkit package and tools that you can use to develop your mobile web applications.

- When you create a new application, you can add the Dojo Mobile files from the **Configure JavaScript Libraries** dialog. Select the **Add Dojo Toolkit** check box.
- The project is initialized with mobile-optimized layers and resources. A new Dojo Library project is created with a full Dojo Toolkit to supplement the mobile layers.

## Deployment configuration with Ant



MobileFirst Studio automatically uses the `build-dojo.xml` file to specify which parts of the Dojo Toolkit package, such as compressed layers and resources, must be deployed with your application. By default, all the resources from the `www` folder are deployed.

**Note:** The `build-dojo.xml` file stores the configuration that must be deployed to run a basic Dojo Mobile application. You can modify the file to add more layers to deploy with the application.

## Custom layers

To create custom layers, you can use the full Dojo Toolkit that is part of the Dojo Library project.

- Update the `build-dojo.properties` file to indicate where to pick up the Dojo Toolkit files from.
- Update the `dojo.workspaceRoot` property value to pick up the files from another directory in your project, or from another project in your workspace: `dojo.workspaceRoot=/myProject/dojo`
- Use the `dojo.root` property instead if the directory is not in your workspace:  
`dojo.root=C\:/dojo`

## Deploying to a device

When development is done, the used resources must be in the application (either copied loosely or as layers) to run properly.

To configure the application to use local resources only, toggle the **Provide Library Resources** action in the console and redeploy.



## Working with Ionic framework

The Ionic framework (<http://ionicframework.com/>) utilizes Angular in order to create a powerful SDK suited to develop rich and robust applications. Ionic offers a library of mobile-optimized HTML, CSS and JS components, gestures, and tools for building highly interactive apps.

An overview of using the Ionic Framework with MobileFirst Hybrid Mobile Applications ([file:///home/travis/build/MFPSTemplates/DevCenter/\\_site/blog/2015/04/16/an-overview-of-using-the-ionic-framework-with-mobilefirst-hybrid-mobile-applications/](file:///home/travis/build/MFPSTemplates/DevCenter/_site/blog/2015/04/16/an-overview-of-using-the-ionic-framework-with-mobilefirst-hybrid-mobile-applications/)).