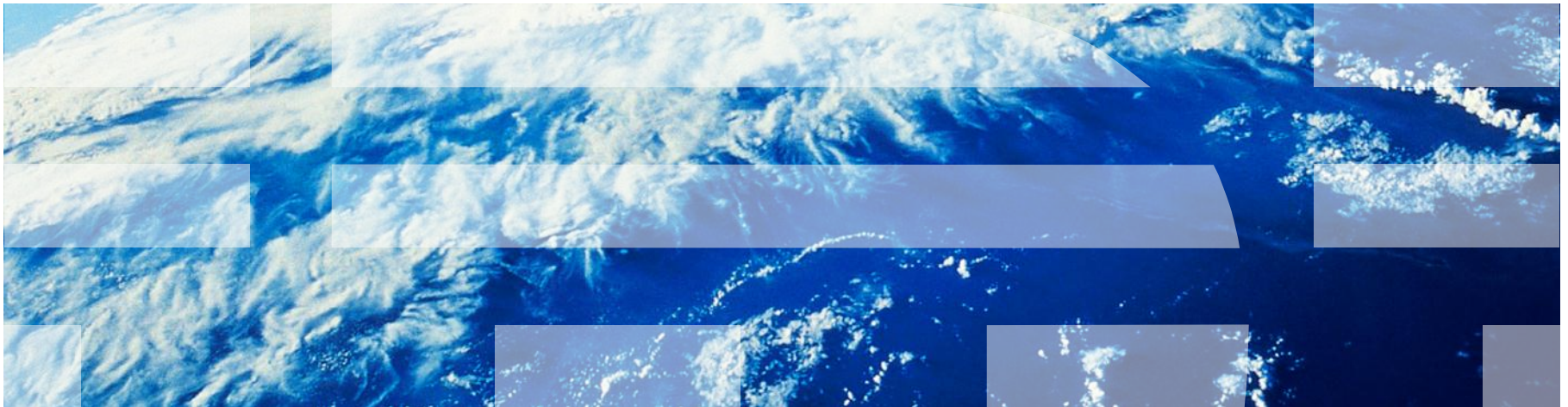


IBM Worklight Foundation V6.2.0 Getting Started

Building a multipage application



Trademarks

- IBM, the IBM logo, ibm.com, and Worklight are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “[Copyright and trademark information](#)” at www.ibm.com/legal/copytrade.shtml.
- Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.
- Other company products or service names may be trademarks or service marks of others.
- This document may not be reproduced in whole or in part without the prior written permission of IBM.

About IBM®

- See <http://www.ibm.com/ibm/us/en/>

Agenda

- Multipage application basics
- Loading an external HTML file
- Implementing page navigation with history
- Exercise

Multipage application basics

- A Worklight® hybrid application uses a single DOM model.
- A single DOM model means that you must never navigate between various HTML files by using hyperlinks or changing the `window.location` property.
- Instead, you must implement multipage interfaces by loading external HTML file content, and by using Ajax requests and injecting them into the existing DOM.
- This is because the main application HTML file loads the Worklight client-side JavaScript™ framework files, and when the browser navigates from one HTML file to another, the JavaScript context and loaded scripts are lost.
- Most JavaScript UI frameworks that are available today (for example, jQuery Mobile, Sencha Touch, Dojo Mobile) provide extensive APIs to achieve the required multipage navigation.
- In this module, learn how to build a multipage Worklight application by using built-in functionality only.
- **Important:** Do not use the built-in functionality that is described in this module if you are using JavaScript UI framework. Use the framework APIs instead.

Multipage application basics

- Applications with multiple pages can be built in two ways:

Single HTML file that contains all the application pages



A separate HTML file for each application page



Multipage application basics

- A single HTML file is the preferred model for simpler applications.
- The developer is responsible for showing the “current” page `<div>` and hiding the rest.
- However, large applications present a challenge:
 - The developer must take full responsibility for which DIVs are shown and which DIVs are hidden at any moment.
 - If you want to add some new content to a page, for example a table, you cannot load a prepared template. You must generate it manually.
 - A single large HTML file with many pages takes longer to load.
 - It is easy to get lost in a single HTML file that contains multiple pages. Separate files are easier to manage.

Multipage application basics

- Building a rich dynamic application with multiple pages can be easier with dynamic pages loading.
- You can use built-in jQuery APIs to dynamically load, update, and insert DOM elements in your application.
- You can insert HTML pages with CSS and JavaScript as needed.
- You can implement navigation history.
- JavaScript code can be run when pages are loaded or unloaded.
- In the following slides, you learn how to implement a simple multipage navigation interface.

Multipage application basics

- **Note:** When you implement multipage navigation in the **Windows Phone 8** environment, you **must** change the URL each time you use the **jQuery** `load()` function.
- Add `/www/default/` at the beginning of the URL path string.
- For example:

```
$("#pagePort").load("pages/MainPage.html", function() {
    (currentPage.init);
});
```

Change the URL path string to:

```
$("#pagePort").load("/www/default/pages/MainPage.html", function() {
    (currentPage.init);
});
```

- The example project for this training module demonstrates in detail how you can perform this task.

Multipage application basics

- **Note:** When you implement multipage navigation in the **Windows Phone 8** environment, and jQuery Mobile is used with the `changePage()` API method, a jQuery Mobile defect prevents it from working.
- To overcome the defect, consult the changes that need to be made in the .js file of jQuery Mobile, as described in the following Stack Overflow question:
- <http://stackoverflow.com/questions/17965560/ibm-worklight-v-5-0-6-cant-navigate-multipages-on-windows-phone-7-5-environme>

Agenda

- Multipage application basics
- Loading an external HTML file
- Implementing page navigation with history
- Exercise

Loading an external HTML file

- An external HTML file is a segment of HTML code that can be injected into any location in the existing DOM.
- A single HTML file can contain multiple HTML elements hierarchy.
- You can include JavaScript by using the `<script>` tag.
- You can include CSS files by using the `<link>` tag.
- You can inject it into the parent element, usually `<div>`, but it is not mandatory.
- You implement it by using the jQuery `$().load()` API method.

Loading an external HTML file

- To load an HTML file, use the following syntax:

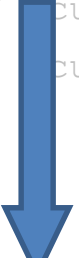
```
$(containerSelector).load(filePath, callbackFunction);
```

- `containerSelector` – jQuery CSS selector of element to host the loaded content.
- `filePath` – Relative path to an HTML file. Always relative to the main HTML file.
- `callbackFunction` - a JavaScript function to run when the loading completes.

Loading an external HTML file

■ Example

```
$("#pagePort").load(path + "pages/MainPage.html", function() {  
    $.getScript(path + "js/MainPage.js", function() {  
        if (currentPage.init) {  
            currentPage.init();  
        }  
    });  
});
```



```
<!-- This is a placeholder for dynamic page content -->  
<div id="pagePort"></div>
```

- This code loads the **MainPage.html** file and inserts its content into the `pagePort` `<div>` element.
- JavaScript and CSS references from the **MainPage.html** are loaded into the DOM.

Agenda

- Multipage application basics
- Loading an external HTML file
- **Implementing page navigation with history**
- Exercise

Implementing page navigation with history

- By using the technique previously described to load an external HTML file, you can implement a navigation interface with history.

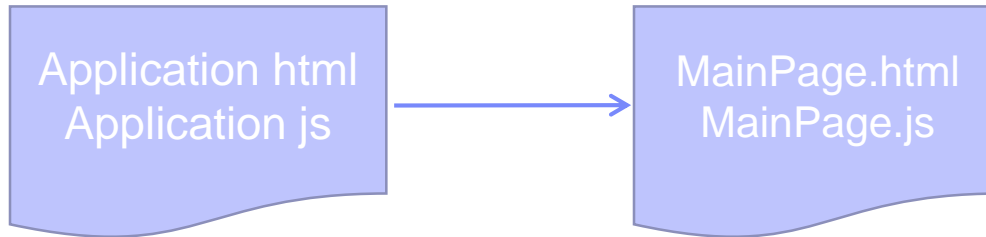


- You must keep the navigation history as a stack in an array object.
- You must also keep a reference to a currently loaded page functionality by using a JavaScript object variable.

```
var pagesHistory = [];
```

```
var currentPage = {};
```

Implementing page navigation with history – step 1



```

$("#pagePort").load("pages/MainPage.html",
function() {
    if (currentPage.init) {
        currentPage.init();
    }
});
  
```

```

currentPage = {};
currentPage.init = function() {
    WL.Logger.debug("MainPage :: init");
};
  
```

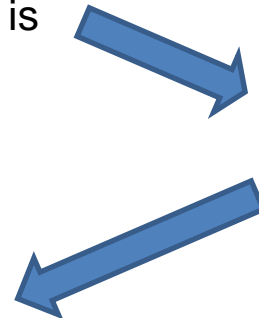
1. On application start, **MainPage.html** is loaded from the application code and injected into `#pagePort` div.

2. **MainPage.js** is loaded as a part of `MainPage.html`.

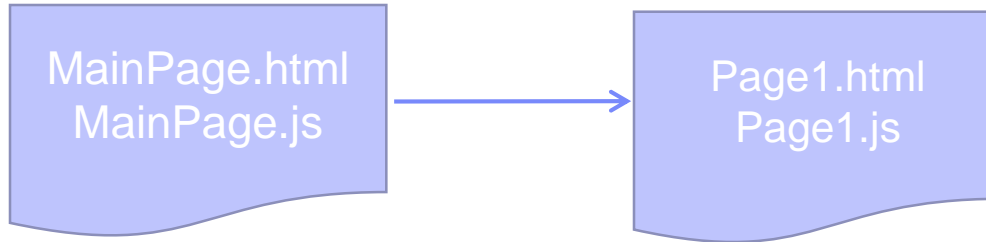
3. The `currentPage` object is declared.

4. The `currentPage.init` function is declared.

5. When the **MainPage.html** loading completes, the `currentPage.init` method is called.



Implementing page navigation with history – step 2



```

currentPage.loadPage = function(pageIndex){
WL.Logger.debug("MainPage :: loadPage :: pageIndex:
" + pageIndex);
pagesHistory.push("pages/MainPage.html");
$("#pagePort").load("pages/Page" + pageIndex +
".html");
};

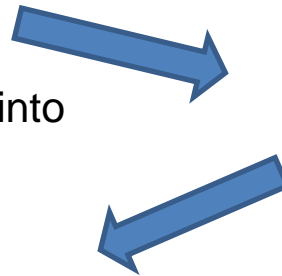
```

```

currentPage = {};
currentPage.init = function(){
    WL.Logger.debug("page1 :: init");
};

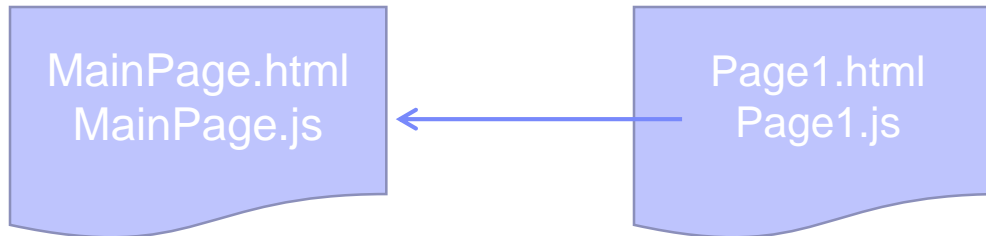
```

1. **MainPage.html** is pushed into **pagesHistory** stack.
2. **Page1.html** is loaded and injected into `#pagePort` div.
6. When the loading of **Page1.html** completes, a new `currentPage.init` method is called.



3. **Page1.js** is loaded as a part of **Page1.html**.
4. The `currentPage` object is declared and overrides the old one.
5. The `currentPage.init` function is declared.

Implementing page navigation with history – step 3



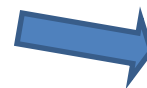
```

currentPage = {};
currentPage.init = function() {
    WL.Logger.debug("MainPage :: init");
};
  
```

```

currentPage.back = function() {
    WL.Logger.debug("Page1 :: back");
    $("#pagePort").load(pagesHistory.pop(),
        function() {
            if (currentPage.init) {
                currentPage.init();
            }
        });
};
  
```

3. **MainPage.js** is loaded as a part of **MainPage.html**.
4. The `currentPage` object is declared and overrides the old one.
5. The `currentPage.init` function is declared.



1. The previous HTML file name is popped from the **PagesHistory** stack (**MainPage.html**).
2. It is loaded and injected into `#pagePort` div.
6. When **MainPage.html** loading completes, the `currentPage.init` method is called.

Agenda

- Multipage application basics
- Loading an external HTML file
- Implementing page navigation with history
- **Exercise**

Exercise

- Create a simple application with three pages:
 - The application HTML file must contain only one `<div>` element as a container for loaded pages.
 - **MainPage.html** is a file that is loaded and displayed automatically on application start.
 - **MainPage.html** must have two buttons – **load page1** and **load page2**.
 - Clicking each button loads the corresponding page.
 - Both **Page1.html** and **Page2.html** must include a button that raises a `WL.SimpleDialog`.
 - Both **Page1.html** and **Page2.html** must have a back button.
 - **Page1.html** must have a button that dynamically loads and adds an HTML fragment to a predefined placeholder (`<div>`).
 - **Page2.html** must have a button that loads **Page1.html**.

Exercise

You can find the sample for this training module in the Getting Started page of the IBM Worklight Foundation documentation website at <http://www.ibm.com/mobile-docs>.

The image displays three sequential screenshots of a mobile application interface, each titled "Multi-page application".

- Left Screenshot:** Shows the initial state where the "MainPage.html" page is loaded. It includes a text block stating "Page1.html contains fragments demonstration" and two buttons: "Load Page1.html" and "Load Page2.html".
- Middle Screenshot:** Shows the state after "Page2.html" is loaded. It features a "SimpleDialog from Page2" button, a "Load Page1.html" button, and a "BACK" button.
- Right Screenshot:** Shows a modal dialog box titled "Page2" overlaid on the application. The dialog contains the text "Button on Page2 was clicked" and an "OK" button.

Notices

- Permission for the use of these publications is granted subject to these terms and conditions.
- This information was developed for products and services offered in the U.S.A.
- IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.
- IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
 - IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.
- For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:
 - Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan
- **The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.
- This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.
- Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.
- IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.
- Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:
 - IBM Corporation
Dept F6, Bldg 1
294 Route 100
Somers NY 10589-3216
USA

- Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.
- The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.
- Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

COPYRIGHT LICENSE:

- This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.
- Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:
 - © (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs.
© Copyright IBM Corp. _enter the year or years_. All rights reserved.

Privacy Policy Considerations

- IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.
- Depending upon the configurations deployed, this Software Offering may use session cookies that collect session information (generated by the application server). These cookies contain no personally identifiable information and are required for session management. Additionally, persistent cookies may be randomly generated to recognize and manage anonymous users. These cookies also contain no personally identifiable information and are required.
- If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent. For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at <http://www.ibm.com/privacy> and IBM's Online Privacy Statement at <http://www.ibm.com/privacy/details> the sections entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at <http://www.ibm.com/software/info/product-privacy>.

Support and comments

- For the entire IBM Worklight Foundation documentation set, training material and online forums where you can post questions, see the IBM website at:
 - <http://www.ibm.com/mobile-docs>
- **Support**
 - Software Subscription and Support (also referred to as Software Maintenance) is included with licenses purchased through Passport Advantage and Passport Advantage Express. For additional information about the International Passport Advantage Agreement and the IBM International Passport Advantage Express Agreement, visit the Passport Advantage website at:
 - <http://www.ibm.com/software/passportadvantage>
 - If you have a Software Subscription and Support in effect, IBM provides you assistance for your routine, short duration installation and usage (how-to) questions, and code-related questions. For additional details, consult your IBM Software Support Handbook at:
 - <http://www.ibm.com/support/handbook>
- **Comments**
 - We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this document. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.
 - For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.
 - When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state.
 - Thank you for your support.
 - Submit your comments in the IBM Worklight Developer Edition support community at:
 - <https://www.ibm.com/developerworks/mobile/worklight/connect.html>
 - If you would like a response from IBM, please provide the following information:
 - Name
 - Address
 - Company or Organization
 - Phone No.
 - Email address

Thank You

