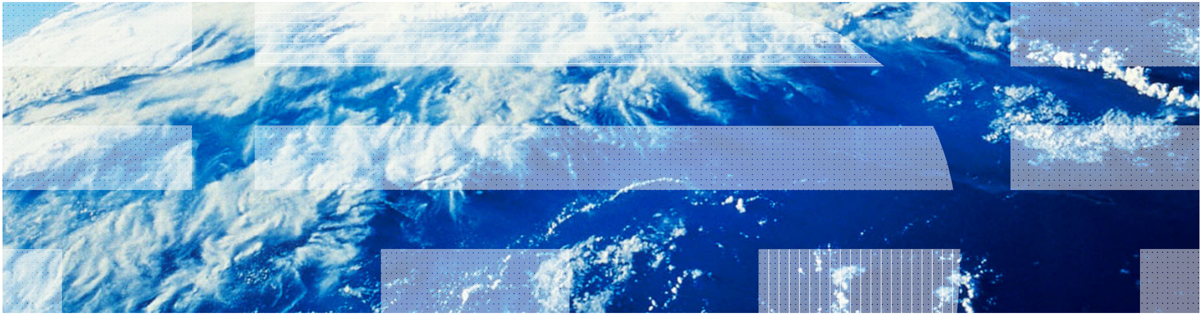
# IBM Worklight V5.0.5
# Getting Started

## Module 7.5 – Building a multi page application



18 January 2013

# Trademarks

- IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Worklight is a trademark or registered trademark of Worklight, an IBM Company. 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

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

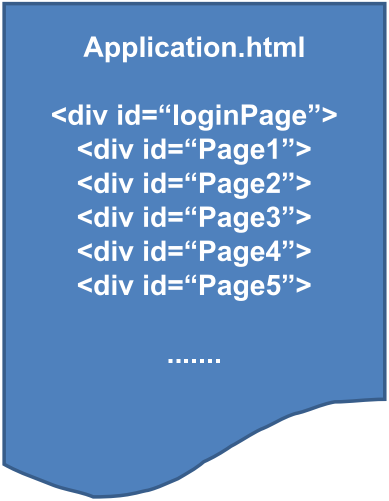
# *Multi page application basics*

- IBM Worklight® hybrid application uses a single DOM model
- Single DOM model means that you should never navigate between various HTML files by using hyperlinks or changing **window.location** property
- Instead, multi page interfaces should be implemented by loading external HTML file content using Ajax requests and injecting it into existing DOM
- This is due to a fact that main application HTML file loads Worklight client side JavaScript framework files and once browser navigates from one HTML file to another the JavaScript context and loaded scripts are lost
- Most JavaScript UI frameworks (e.g jQuery Mobile, Sencha Touch, Dojo Mobile) available today provide an extensive APIs to achieve required multi page navigation
- In this module, you learn how to build a multi-page Worklight application by using built-in functionality only
- Important – you should not use the built-in functionality that is described in this module if you are using JavaScript UI framework. You should use this framework APIs instead.

# *Multi page application basics*

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

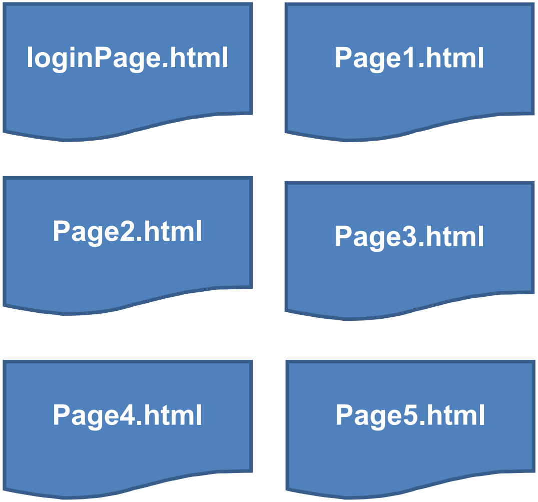
Single HTML file containing all application pages



Application.html

```
<div id="loginPage">
  <div id="Page1">
  <div id="Page2">
  <div id="Page3">
  <div id="Page4">
  <div id="Page5">

  .......
```

A separate HTML file for each application page



loginPage.html

Page1.html

Page2.html

Page3.html

Page4.html

Page5.html

# *Multi page application basics*

- A single HTML file is the preferred model for simpler applications
- 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 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 but must generate it manually.
  - Single large HTML files with lots of pages take longer to load.
  - It is easy to get lost in a single HTML file containing multiple pages. Separate files are easier to manage.

# *Multi page 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
- HTML pages with CSS and JavaScript can be inserted on the fly.
- Possible to implement navigation history
- JavaScript code can be executed when pages are loaded or unloaded
- In the following slides you will learn and implement a simple multi page navigation interface

# *Agenda*

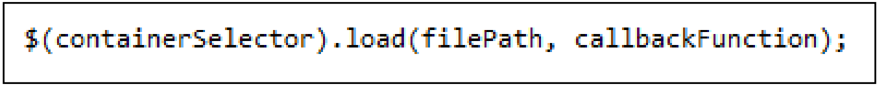
- Multi page application basics
- Loading an external HTML file
- Implementing page navigation with history
- Exercise

# *Loading an external HTML file*

- 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 element hierarchy
- May include JavaScript by using <script> tag
- May include CSS files by using <link> tag
- Injected into parent element, usually <div>, but not mandatory
- Implemented using jQuery **$().load()** API

# *Loading an external HTML file*

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

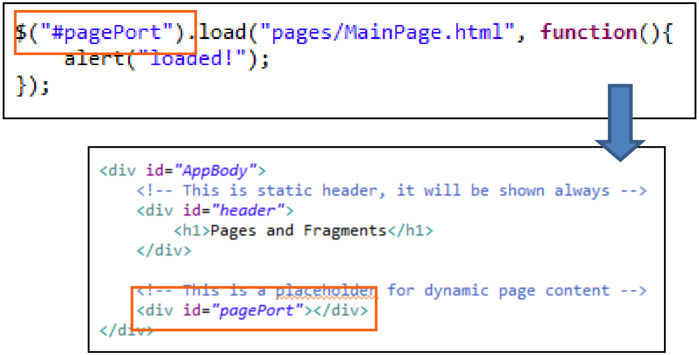


```
$(containerSelector).load(filePath, callbackFunction);
```

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

# Loading an external HTML file

- Example



```
$("#pagePort").load("pages/MainPage.html", function(){
    alert("loaded!");
});
```

```
<div id="AppBody">
    <!-- This is static header, it will be shown always -->
    <div id="header">
        <h1>Pages and Fragments</h1>
    </div>

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

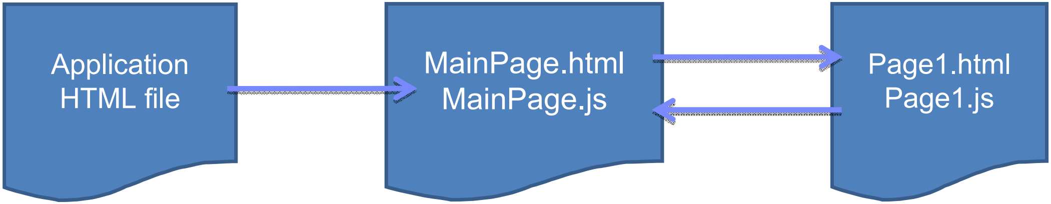
- This code loads **MainPage.html** file and inserts its content into the **pagePort** <div> element
- JavaScript and CSS from **MainPage.html** are loaded to DOM
- **alert("loaded!")** is executed when **MainPage.html** load completes

# *Agenda*

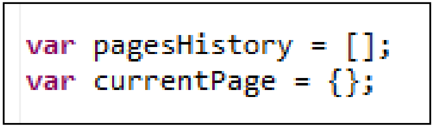
- Multi page application basics
- Loading an external HTML file
- Implementing page navigation with history
- Exercise

# *Implementing page navigation with history*

- By using the technique described previously, you will implement a navigation interface with history in the following slides.

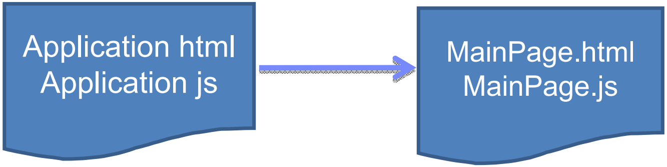


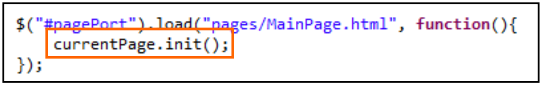
- You will keep the navigation history as a stack in an array object
- You will also keep a reference to a currently loaded page functionality by using a JavaScript object variable



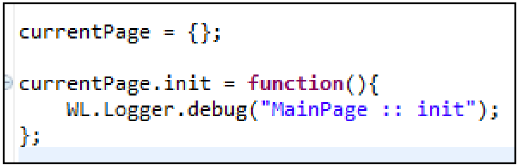
```
var pagesHistory = [];
var currentPage = {};
```

# *Implementing page navigation with history – step 1*





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



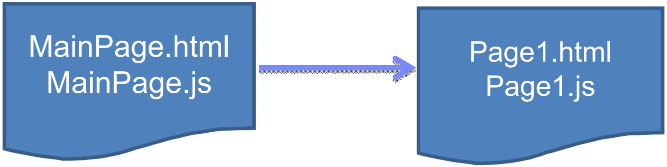
```
currentPage = {};

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

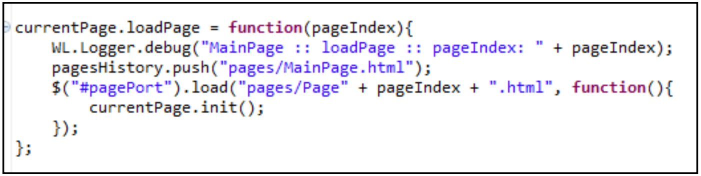
1. On application start **MainPage.html** is loaded from the application code and injected into **#pagePort** div

2. ***MainPage.js*** *is loaded as a part of* ***MainPage.html***
3. ***currentPage*** object is declared
4. ***currentPage.init*** function is declared

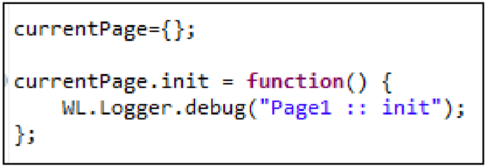
5. When **MainPage.html** loading completes, ***currentPage.init*** method is called

# 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", function(){
        currentPage.init();
    });
};
```



```
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

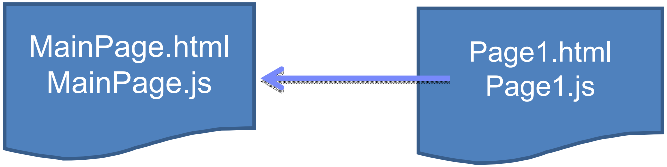
*2.* ***Page1.js*** *is loaded as a part of* ***Page1.html***

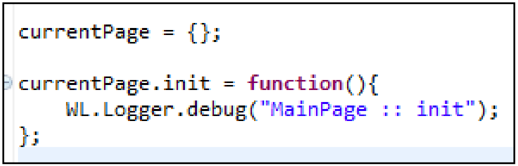
*3.* ***currentPage*** object is declared (overriding the old one)

*4.* ***currentPage.init*** function is declared

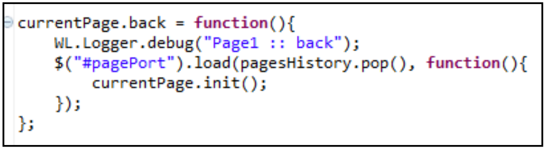
5. When ***Page1.html*** loading completes, a new ***currentPage.init*** method is called

# *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(){
        currentPage.init();
    });
};
```

1. *Previous html file name is popped from the **pagesHistory** stack (**MainPage.html**)*
2. It is loaded and injected into **#pagePort** div

3. ***MainPage.js** is loaded as a part of **MainPage.html***
4. ***currentPage*** object is declared (overriding the old one)
5. ***currentPage.init*** function is declared

6. When **MainPage.html** loading completes, ***currentPage.init*** method is called

# Agenda

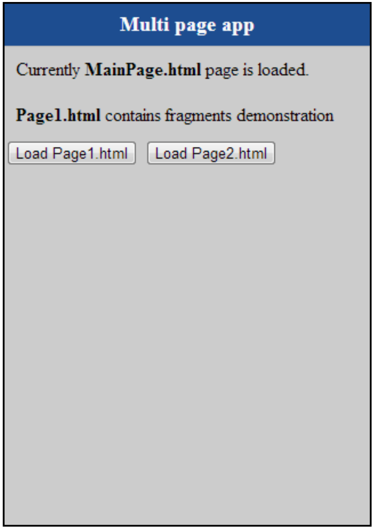
- Multi page application basics
- Loading an external HTML file
- Implementing page navigation with history
- Exercise

# *Exercise*

- Create a simple application with 3 pages as follows:
  - Application HTML file should contain only one <div> element that will be used as a container for loaded pages
  - **MainPage.html** – A file that will be loaded and displayed automatically on application start
    - **MainPage.html** must have two buttons – load page1 and load page2
    - Clicking on 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**

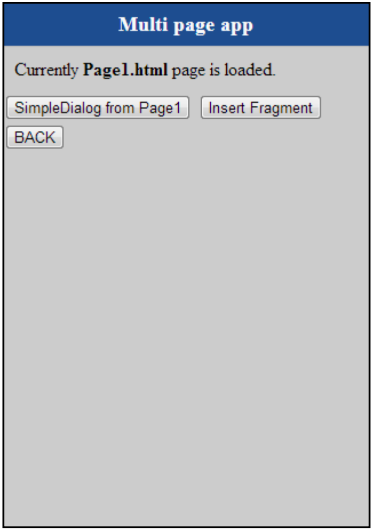
# *Exercise*

- The sample for this training module can be found in the Getting Started page of the IBM® Worklight® documentation website at http://www.ibm.com/mobile-docs

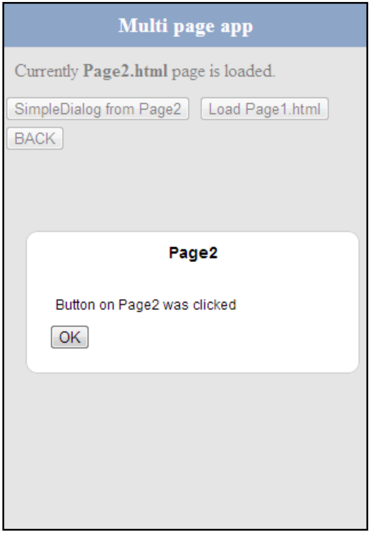


Currently **MainPage.html** page is loaded.

**Page1.html** contains fragments demonstration



Currently **Page1.html** page is loaded.



Currently **Page2.html** page is loaded.

**Page2**

Button on Page2 was clicked

# 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:

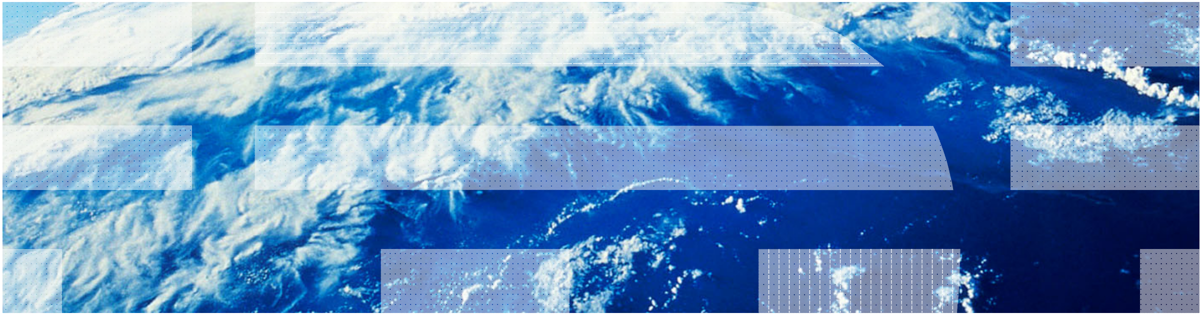
- This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing 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.

# Support and comments

- For the entire IBM Worklight 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 forums at:
    - https://www.ibm.com/developerworks/mobile/mobileforum.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



18 January 2013