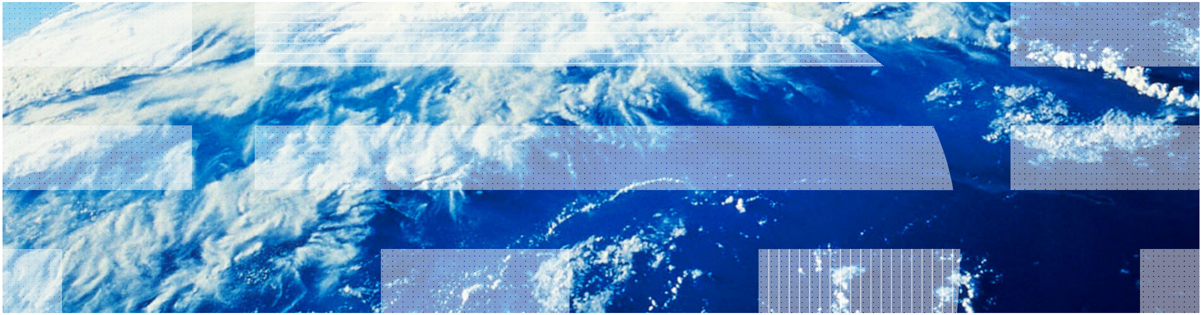
IBM Worklight V5.0.5
Getting Started

Module 7.7 – Offline Access



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

- Working in offline mode
- Active detection - using methods
- Passive detection - offline and online events
- Additional methods
- Foreground event
- Worklight heartbeat
- Samples

Working in offline mode

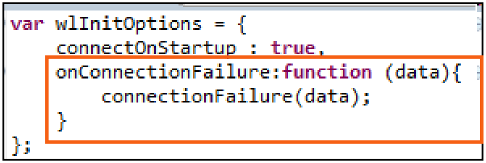
- By using IBM Worklight®, you can detect application connectivity failures and determine a course of action
- Apps going offline and online can be detected in two ways:
  - Explicitly, on invoking server-based procedures
  - Implicitly, by using JavaScript event listeners
- You can define custom application behavior for offline and online status

Agenda

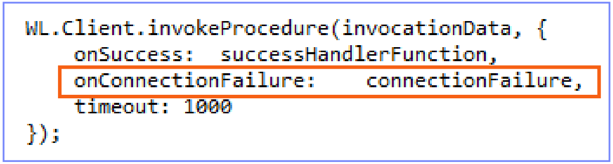
- Working in offline mode
- Active detection - using methods
- Passive detection - offline and online events
- Additional methods
- Foreground event
- Worklight heartbeat
- Samples

Explicit detection
Using methods

- Connectivity loss can be detected in two locations in your application code:
  - Application initialization – WL.Client.init() method, usually called from initOptions.js file
  - Adapter procedure invocation – WL.Client.invokeProcedure() method
- To add connectivity failure detection in either location, add the onConnectionFailure property and specify a callback function to be invoked if connectivity fails



```
var wlInitOptions = {
    connectOnStartup : true,
    onConnectionFailure:function (data){
        connectionFailure(data);
    }
};
```



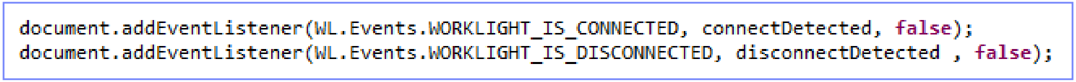
```
WL.Client.invokeProcedure(invocationData, {
    onSuccess:  successHandlerFunction,
    onConnectionFailure:    connectionFailure,
    timeout: 1000
});
```

Agenda

- Working in offline mode
- Active detection - using methods
- Passive detection - offline and online events
- Additional methods
- Foreground event
- Worklight heartbeat
- Samples

Implicit detection Offline and online events

- Each time the Worklight framework attempts to access the Worklight Server, it might detect that the application has switched from offline to online status or vice versa
- In both cases, JavaScript events are fired:
  - WL.Events.WORKLIGHT_IS_DISCONNECTED event is fired when connectivity to the Worklight Server fails
  - WL.Events.WORKLIGHT_IS_CONNECTED event is fired when connectivity to the Worklight Server is restored
- You can add event listeners to these events and specify the callback functions to handle them



```
document.addEventListener(WL.Events.WORKLIGHT_IS_CONNECTED, connectDetected, false);
document.addEventListener(WL.Events.WORKLIGHT_IS_DISCONNECTED, disconnectDetected , false);
```

- Note that both WL.Events.WORKLIGHT_IS_DISCONNECTED and WL.Events.WORKLIGHT_IS_CONNECTED are namespace constants, not strings

Agenda

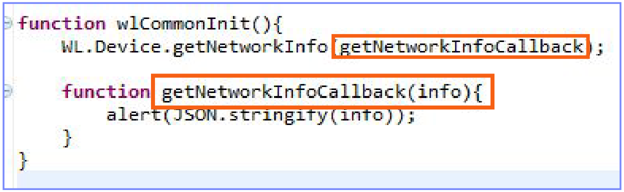
- Working in offline mode
- Active detection - using methods
- Passive detection - offline and online events
- Additional methods
- foreground event
- Worklight Heartbeat
- Samples

Additional methods

- Additional methods are provided by the Worklight framework to simplify online and offline development:
  - WL.Client.connect (options) – Attempt to establish a connection to the Worklight Server and return to online mode. `options` is an object containing the following keys:
    - onSuccess – Callback function to invoke when server connection is established
    - onFailure – Callback function to invoke when server connection fails
    - timeout – Number of milliseconds to wait for the server response before failing with request timeout

Additional methods

- WL.Device.getNetworkInfo() method is available for iOS and Android environments.
- A callback must be specified as a function parameter



```
function wlCommonInit(){
    WL.Device.getNetworkInfo(getNetworkInfoCallback);

    function getNetworkInfoCallback(info){
        alert(JSON.stringify(info));
    }
}
```

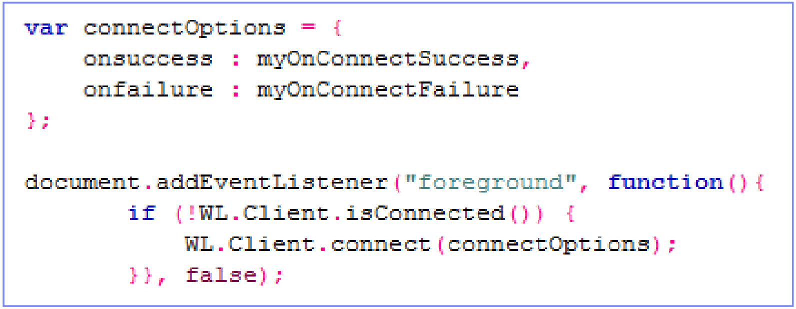
- A callback receives an object with the following properties:
  - isAirplaneMode – true/false
  - carrierName – string (for example, AT&T or VERIZON)
  - telephonyNetworkType – string (for example, UMTS or GPRS)
  - isRoaming – true/false
  - networkConnectionType – mobile/WIFI
  - ipAddress – string
  - isNetworkConnected – true/false

Agenda

- Working in offline mode
- Active detection - using methods
- Passive detection - offline and online events
- Additional methods
- Foreground event
- Worklight heartbeat
- Samples

Foreground event

- When the Worklight Application returns to foreground, a foreground JavaScript event is fired
- You can add a listener for this event and specify the callback function that handles it
- This technique is particularly useful to restart connectivity on the device when the user has left the application. On returning to the app, the user expects the app to connect.



```
var connectOptions = {
    onsuccess : myOnConnectSuccess,
    onfailure : myOnConnectFailure
};

document.addEventListener("foreground", function(){
        if (!WL.Client.isConnected()) {
            WL.Client.connect(connectOptions);
        }}, false);
```

Agenda

- Working in offline mode
- Active detection - using methods
- Passive detection - offline and online events
- Additional methods
- Foreground event
- Worklight heartbeat
- Samples

Worklight heartbeat

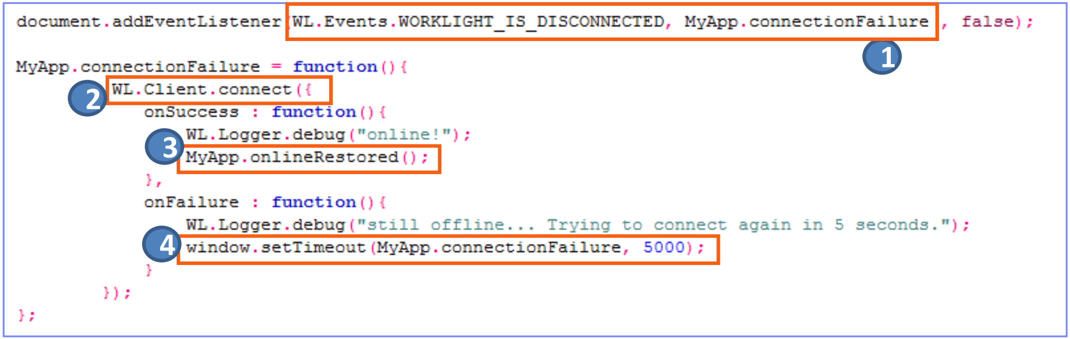
- Worklight heartbeat pings the server at specified intervals to verify connectivity
- You can use Heartbeat periodically to make sure that the application remains connected to the server
- Both WL.Events.WORKLIGHT_IS_CONNECTED and WL.Events.WORKLIGHT_IS_DISCONNECTED events can be fired by heartbeat in designated cases
- You can specify the heartbeat interval using theWL.Client.setHeartBeatInterval(intervalSeconds) API
- Worklight heartbeat usage can be seen in the sample code for this module

Agenda

- Working in offline mode
- Active detection - using methods
- Passive detection - offline and online events
- Additional methods
- Foreground event
- Worklight heartbeat
- Samples

Periodic Connectivity Test Sample

- The following sample shows an offline and online detection mechanism



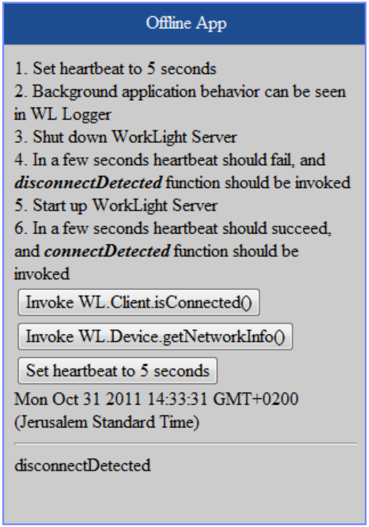
```
document.addEventListener(WL.Events.WORKLIGHT_IS_DISCONNECTED, MyApp.connectionFailure, false);

MyApp.connectionFailure = function(){
        WL.Client.connect({
            onSuccess : function(){
                WL.Logger.debug("online!");
                MyApp.onlineRestored();
            },
            onFailure : function(){
                WL.Logger.debug("still offline... Trying to connect again in 5 seconds.");
                window.setTimeout(MyApp.connectionFailure, 5000);
            }
        });
};
```

1. An event listener for a WL.Events.WORKLIGHT_IS_DISCONNECTED event is added to the document. MyApp.connectionFailure () is invoked when the event fires
2. WL.Client.connect () tries to establish a server connection
3. If connection is successfully established MyApp.onlineRestored () is invoked
4. If connection fails, a timeout is set for five seconds to invoke MyApp.connectionFailure () again

Worklight Heartbeat Sample

- The sample for this training module can be found in the Getting Started page of the IBM Worklight documentation website at https://www.ibm.com/developerworks/mobile/worklight/getting-started/



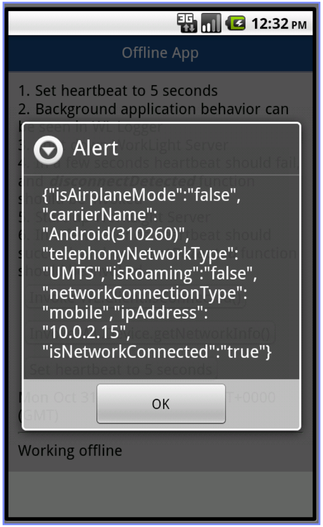
Offline App

1. Set heartbeat to 5 seconds
2. Background application behavior can be seen in WL Logger
3. Shut down WorkLight Server
4. In a few seconds heartbeat should fail, and ***disconnectDetected*** function should be invoked
5. Start up WorkLight Server
6. In a few seconds heartbeat should succeed, and ***connectDetected*** function should be invoked

Invoke WL.Client.isConnected()
Invoke WL.Device.getNetworkInfo()
Set heartbeat to 5 seconds

Mon Oct 31 2011 14:33:31 GMT+0200 (Jerusalem Standard Time)

disconnectDetected



Offline App

1. Set heartbeat to 5 seconds
2. Background application behavior can

Alert

{"isAirplaneMode":"false", "carrierName": "Android(310260)", "telephonyNetworkType": "UMTS","isRoaming":"false", "networkConnectionType": "mobile","ipAddress": "10.0.2.15", "isNetworkConnected":"true"}

Working offline

Check yourself questions

- What is a best way to keep an active server session alive?
  - Alert the user that the session is about to expire and to perform some action to keep it alive
  - Invoke any Worklight adapter procedure in the background, ignoring the response
  - Set heartbeat interval to a value lower than session timeout defined on Worklight Server
  - There is no way to keep the session alive from the client, the server is the only entity that can control this
- What happens when adapter procedure invocation fails because of inability to reach the Worklight server?
  - Client side framework tries to invoke the procedure until it succeeds
  - onFailure callback specified in invocation options is invoked. The developer should use it to treat server connectivity issues
  - onFailure callback specified in invocation options is invoked. In addition WL.Events.WORKLIGHT_IS_DISCONNECTED event is fired. The developer should use either of them to treat server connectivity issues
  - The client side framework shows an error message with a retry button
- What information cannot be retrieved using WL.Device.getNetworkInfo API?
  - The current IP address of the device
  - The telephony type currently in use (for example, GSM or UMTS)
  - Whether the device is currently roaming
  - The phone number of the device

Check yourself questions

- What is a best way to keep an active server session alive?
  - Alert the user that the session is about to expire and to perform some action to keep it alive
  - Invoke any Worklight adapter procedure in the background, ignoring the response
  - [✓ Set heartbeat interval to a value lower than session timeout defined on Worklight Server]
  - There is no way to keep the session alive from the client, the server is the only entity that can control this
- What happens when adapter procedure invocation fails because of inability to reach the Worklight server?
  - Client side framework tries to invoke the procedure until it succeeds
  - onFailure callback specified in invocation options is invoked. The developer should use it to treat server connectivity issues
  - [✓ onFailure callback specified in invocation options is invoked. In addition WL.Events.WORKLIGHT_IS_DISCONNECTED event is fired. The developer should use either of them to treat server connectivity issues]
  - The client side framework shows an error message with a retry button
- What information cannot be retrieved using WL.Device.getNetworkInfo API?
  - The current IP address of the device
  - The telephony type currently in use (for example, GSM or UMTS)
  - Whether the device is currently roaming
  - [✓ The phone number of the device]

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