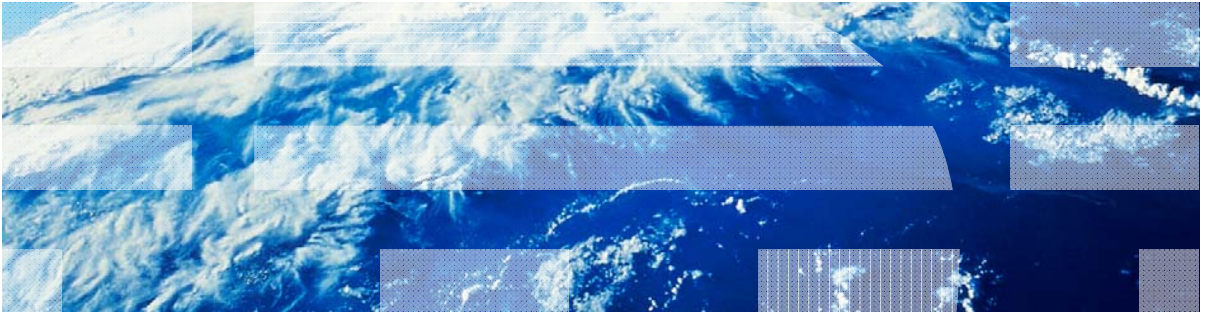


IBM Worklight V5.0.6 Getting Started

Invoking adapter procedures from client applications



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

- Invoking a procedure from the client application
 - The **invocationData** object - passing data to the server
 - The **options** object
- Invocation Result
- Exercise

Invoking a procedure from the application

- Worklight® applications can adapt procedures to communicate with any data source without being subjected to same-origin constraints.
- The first step to invoke an adapter procedure is to create an **invocationData** object:

```
var invocationData = {  
    adapter : 'RSSReader',  
    procedure : 'getFeedsFiltered',  
    parameters : [param1, param2, param3]  
};
```

Invoking a procedure from the application

WL.Client.invokeProcedure (**invocationData**, options)

- The ***invocationData*** object is used to provide invocation configuration and procedure parameters.
- It consists of a JSON block of properties.

```
var invocationData = {  
    adapter : 'RSSReader',  
    procedure : 'getFeedsFiltered',  
    parameters : [param1, param2, param3]  
};
```

Invoking a procedure from the application

WL.Client.invokeProcedure (**invocationData** options)

- The **invocationData** object is used to provide invocation configuration and procedure parameters
- It consists of a JSON block of parameters

adapter (mandatory)

A string that contains the name of the adapter as specified in the adapter <wl:adapter> element

```
var invocationData = {  
    adapter : 'RSSReader',  
    procedure : 'getFeedsFiltered',  
    parameters : [param1, param2, param3]  
};
```

Invoking a procedure from the application

WL.Client.invokeProcedure (**invocationData** options)

- The **invocationData** object is used to provide invocation configuration and procedure parameters
- It consists of a JSON block of parameters

procedure (mandatory)
Procedure name as defined
in the XML file

```
var invocationData = {  
    adapter : 'RSSReader',  
    procedure : 'getFeedsFiltered',  
    parameters : [param1, param2, param3]  
};
```

Invoking a procedure from the application

WL.Client.invokeProcedure (**invocationData** options)

- The **invocationData** object is used to provide invocation configuration and procedure parameters
- It consists of a JSON block of parameters

parameters (mandatory)

An array of parameters that are passed to the back-end JavaScript™ procedure.
Leave empty if no parameters are required.

```
var invocationData = {  
  adapter : 'RSSReader',  
  procedure : 'getfeedsfiltered',  
  parameters : [param1, param2, param3]  
};
```


Agenda

- Invoking a procedure from the client application
 - The **invocationData** object - passing data to the server
 - The **options** object
- Invocation Result
- Exercise

Invoking a procedure from the application

- The second step is to define failure and success behaviors in the **options** object.

```
var options = {  
    onSuccess : loadFeedsSuccess,  
    onFailure : loadFeedsFailure,  
    invocationContext: {}  
};
```

The Options Object

WL.Client.invokeProcedure (invocationData, options)

- The Options object must be passed for all asynchronous calls to the Worklight Server.

```
var options = {  
  onSuccess : loadFeedsSuccess,  
  onFailure : loadFeedsFailure,  
  invocationContext: {}  
};
```

The function to be invoked on successful completion of the asynchronous call

The **response** typically contains the following properties:

- **invocationContext**: The **invocationContext** object that was originally passed in the options object, or undefined if no **invocationContext** object was passed
- **status**: The HTTP response status
- **invocationResult**: An object that contains the data that is returned by the invoked procedure, and additional information about the procedure invocation

The Options Object

WL.Client.invokeProcedure (invocationData, **options**)

- The Options object must be passed for all asynchronous calls to the Worklight Server

```
var options = {  
  onSuccess : loadFeedsSuccess,  
  onFailure : loadFeedsFailure,  
  invocationContext: {}  
};
```

The function to be invoked on failure

Includes both server-side errors, and client-side (such as server connection failure or timed out calls). The **response** typically contains the following properties:

- **invocationContext**
- **errorCode**: An error code string. All error codes that can be returned are defined as constants in the WL.ErrorCode object in worklight.js.
- **errorMessage**: This message is for developer use only, and is not displayed to the user. It is not translated to the user's language.
- **status**: The HTTP response status.

The Options object

WL.Client.invokeProcedure (invocationData, options)

- The Options object must be passed for all asynchronous calls to the Worklight Server.

```
var options = {  
  onSuccess : loadFeedsSuccess,  
  onFailure : loadFeedsFailure,  
  invocationContext: {}  
};
```

Optional parameter. An object that is returned to the success and failure handlers

The **invocationContext** object is used to preserve the context of the calling asynchronous service upon return from the service.

Invoking a procedure from the client application

- Bringing everything together

```
function loadFeeds(){  
    var invocationData = {  
        adapter : 'RSSReader',  
        procedure : 'getFeedsFiltered',  
        parameters : []  
    };  
};
```

```
WL.Client.invokeProcedure(invocationData,{  
    onSuccess : loadFeedsSuccess,  
    onFailure : loadFeedsFailure,  
});
```

```
function loadFeedsSuccess(result){  
    WL.Logger.debug("Feed retrieve success");  
}
```

```
function loadFeedsFailure(result){  
    WL.Logger.error("Feed retrieve failure");  
}
```

1. Invocation data
2. Invoking the procedure
3. Setting the success/failure handlers
4. Success handler
5. Failure handler

Agenda

- Invoking a procedure from the client application
 - The **invocationData** object - passing data to the server
 - The **options** object
- **Invocation Result**
- Exercise

Invocation result

- A JSON object that contains the data and additional information about the procedure invocation is returned.
- The Object is returned to a corresponding success/failure handler.

```
Items
  [ Object { pubDate="Sun. 26 Dec 2010 08:14:00 EDT",
    title="Stereoscopic copy-paste...third dimension (video)",
    link="http://www.engadget.com...at-clone-tool-into-the/" },
  Object { pubDate="Sun. 26 Dec 2010 05:38:00 EDT", title="Fisker
    raises Karma plu...back into consideration",
    link="http://www.engadget.com...price-to-95-900-brings/" },
  Object { pubDate="Sun. 26 Dec 2010 02:18:00 EDT", title="New
    solar machine could... for you future vehicle",
    link="http://www.engadget.com...-fuel-food-for-you-fut/" }, 37
  more... ]

errors      [ ]
info        [ ]
isSuccessful true
responseID  "23"
warnings    [ ]
```


Invocation Result

```

+ Items
    [ Object { pubDate="Sun, 26 Dec 2010 08:14:00 EDT",
      title="stereoscopic copy-paste...third dimension (video)",
      link="http://www.engadget.com...at-clone-tool-into-the/" },
    Object { pubDate="Sun, 26 Dec 2010 05:38:00 EDT", title="Riskier
      raises Karma plu...back into consideration",
      link="http://www.engadget.com...price-to-95-900-brings/" },
    Object { pubDate="Sun, 26 Dec 2010 02:18:00 EDT", title="New
      solar machine could... for you future vehicle",
      link="http://www.engadget.com...-fuel-food-for-you-fut/" }, 37
  more... ]
errors      [ ]
info        [ ]
isSuccessful true
responseID  "23"
warnings    [ ]

```

The *isSuccessful* property is set to:

- **true** if the procedure invocation succeeded (even if no data was retrieved)
- **false** otherwise

Invocation Result

```

+ Items      [ Object { pubDate="Sun, 26 Dec 2010 08:14:00 EDT",
              title="stereoscopic copy-paste...third dimension (video)",
              link="http://www.engadget.com...at-clone-tool-into-the/" },
              Object { pubDate="Sun, 26 Dec 2010 05:38:00 EDT", title="Risker
              raises Karma plu...back into consideration",
              link="http://www.engadget.com...price-to-95-900-brings/" },
              Object { pubDate="Sun, 26 Dec 2010 02:18:00 EDT", title="New
              solar machine could... for you future vehicle",
              link="http://www.engadget.com...-fuel-food-for-you-fut/" }, 37
              more... ]
errors       [ ]
info         [ ]
isSuccessful true
responseID   "23"
warnings     [ ]

```

Records

An array of records that are retrieved from the back-end system. Each array entry is an object that contains the fields of the single record. If the procedure call returns no records, an empty array is returned. If not specified otherwise in the XSL file, default property name is *recordSet*.

Invocation Result

Items	<pre>[Object { pubDate="Sun, 26 Dec 2010 08:14:00 EDT", title="stereoscopic copy-paste...third dimension (video)", link="http://www.engadget.com...at-clone-tool-into-the/" }, Object { pubDate="Sun, 26 Dec 2010 05:38:00 EDT", title="Risker raises Karma plu...back into consideration", link="http://www.engadget.com...price-to-95-900-brings/" }, Object { pubDate="Sun, 26 Dec 2010 02:18:00 EDT", title="New solar machine could... for you future vehicle", link="http://www.engadget.com...-fuel-food-for-you-fut/" }, 37 more...]</pre>
errors	[]
info	[]
isSuccessful	true
responseID	"23"
warnings	[]

Error/Warn/Info Messages

Optional arrays of strings that contain messages. If no messages are provided, these objects are NULL.

Invocation result

```
function loadFeedsSuccess(result) {
    WL.Logger.debug("Feed retrieve success");
    if (result.invocationResult.Items.length>0) this.displayFeeds result.invocationResult.Items;
}
```



Items

```
[ Object { pubDate="Sun, 26 Dec 2010 08:14:00 EDT", title="stereoscopic
copy-paste...third dimension (video)", link="http://www.engadget.com...at-
clone-tool-into-the/" }, Object { pubDate="Sun, 26 Dec 2010 05:38:00 EDT",
title="Fisker raises Karma plu...back into consideration",
link="http://www.engadget.com...price-to-95-900-brings/" }, Object { pubDate="Sun,
26 Dec 2010 02:18:00 EDT", title="New solar machine could... for you future
vehicle", link="http://www.engadget.com...fuel-food-for-you-fut/" }, 37 more... ]
0 Object { pubDate="Sun, 26 Dec 2010 08:14:00 EDT", title="stereoscopic
copy-paste...third dimension (video)", link="http://www.engadget.com...at-
clone-tool-into-the/" }
1 Object { pubDate="Sun, 26 Dec 2010 05:38:00 EDT", title="Fisker raises Karma
plu...back into consideration", link="http://www.engadget.com...price-
to-95-900-brings/" }
2 Object { pubDate="Sun, 26 Dec 2010 02:18:00 EDT", title="New solar machine
could... for you future vehicle", link="http://www.engadget.com...fuel-
food-for-you-fut/" }
3 Object { pubDate="Sat, 25 Dec 2010 23:59:00 EDT", title="Nikon patent app
detail.... videographers rejoice", link="http://www.engadget.com...and-
electronic-zoom-v/" }
4 Object { pubDate="Sat, 25 Dec 2010 22:33:00 EDT", title="Sanyo Eneloop Pedal Juice
battery pack preview", link="http://www.engadget.com...e-battery-
pack-preview/" }
```

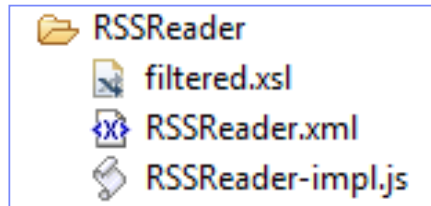
Agenda

- Invoking a procedure from the client application
 - The **invocationData** object - passing data to the server
 - The **options** object
- Invocation Result
- Exercise

Exercise

Displaying the Engadget Feed

- The application uses the HTTP Adapter created in module 5.2
- You can use the **Worklight Client API Basics** (Module 4) as a reference



Exercise

Displaying the Engadget Feed

- The purpose of the **RSSReader** adapter is to have a **getFeedsFiltered** procedure
- The **getFeedsFiltered** procedure returns a record set called **Items** with the following fields:
 - **title**
 - **link**
 - **pubDate**

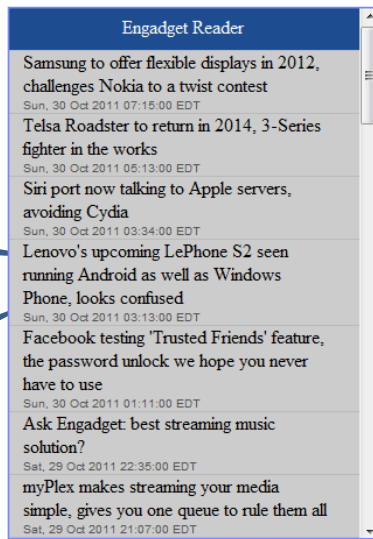
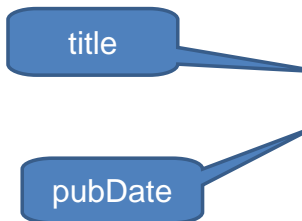
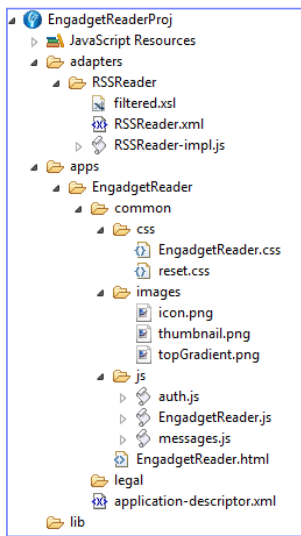
Exercise

Displaying the Engadget Feed

- By using the **WL.Client.invokeProcedure** API, invoke the **getFeedsFiltered** procedure
- Use **Firebug** or a similar tool to inspect the request and the JSON response structure
- Parse the returned **Items** records and display them in your application
- Each item requires a title and a publication date

Exercise

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



Check yourself questions

- The invocationData object must contain following properties:
 - "adapter" – mandatory, "procedure" – mandatory, "parameters" – mandatory
 - "adapter" – mandatory, "procedure" – mandatory, "parameters" – optional
 - "adapter" – mandatory, "procedure" – optional, "parameters" – optional
 - "adapter" – optional, "procedure" – optional, "parameters" – optional
- Assuming that it is not overridden manually, the isSuccessful property of the procedure invocation results true if:
 - The Worklight server is not accessible
 - The Back-end system that adapter tries to connect to is not accessible
 - Internet is not available on a mobile device
 - The specified procedure does not exist in the adapter
- How can the developer declare what to do when a response from Worklight adapter is received?
 - The developer is not required to declare it. The client side framework code does everything automatically
 - By specifying the callback function names in adapter XML file
 - By specifying the callback function names in onSuccess and onFailure properties of an options object
 - By adding the callback function names as invocation parameters

Check yourself questions

- The invocationData object must contain following properties
 - "adapter" – mandatory, "procedure" – mandatory, "parameters" – mandatory
 - "adapter" – mandatory, "procedure" – mandatory, "parameters" - optional
 - "adapter" – mandatory, "procedure" – optional, "parameters" - optional
 - "adapter" – optional, "procedure" – optional, "parameters" - optional
- Assuming that it is not overridden manually, the isSuccessful property of the procedure invocation results true if:
 - The worklight server is not accessible
 - The back-end system that the adapter tries to connect to is not accessible
 - Internet is not available on a mobile device
 - The specified procedure does not exist in the adapter
- How can developer declare what to do when a response from Worklight adapter is received
 - The developer is not required to declare it, client side framework code does everything automatically
 - By specifying the callback function names in adapter XML file
 - By specifying the callback function names in onSuccess and onFailure properties of an options object
 - By adding the callback function names as invocation parameters

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/en/us> sections entitled "Cookies, Web Beacons and Other Technologies" and "Software Products and Software-as-a-Service".

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

