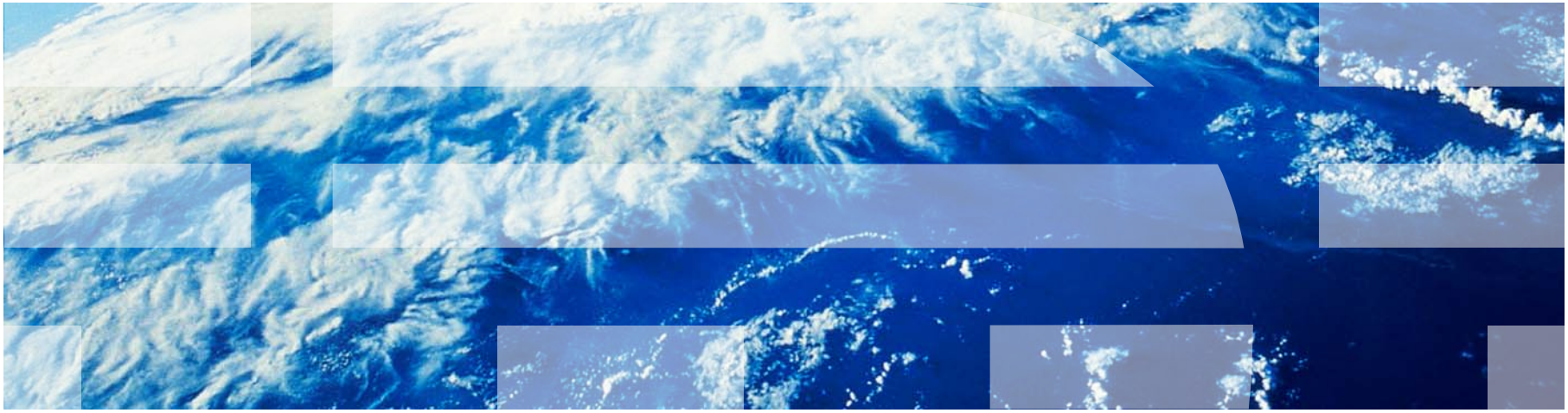
IBM Worklight V6.1.0
Getting Started

HTTP adapter – Communicating with HTTP back-end systems



7 March 2014

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

- What is it?
- How does it work?
- Creating the adapter
- Using SOAP
  - Creating SOAP-based service request
  - Service request invocation
  - Service discovery
- Back-end service discovery
- Exercise

What is it?

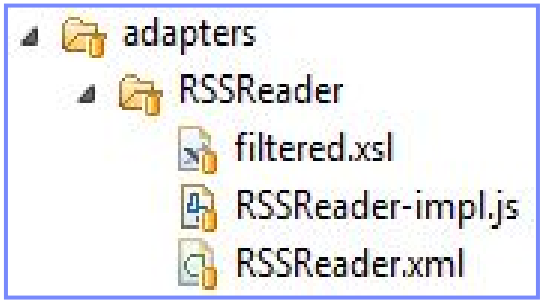
- A Worklight® HTTP adapter:
  - Works with RESTful and SOAP-based services.
  - Can read structured HTTP sources, for example RSS feeds.
  - Allows sending a GET or POST HTTP request and retrieves data from the response headers and body.
  - Is easily customizable with simple server-side JavaScript™.
  - Enables optional server-side filtering.
  - Retrieved data can be in XML, HTML, JSON, or plain text formats.

Agenda

- What is it?
- How does it work?
- Creating the adapter
- Using SOAP
  - Creating SOAP-based service request
  - Service request invocation
- Back-end service discovery
- Exercise

How does it work?

- The adapter is configured with XML.
- It uses XML to define the adapter properties and procedures.
- It uses JavaScript to create procedures.
- *Optional*: It uses XSL to filter received records and fields.

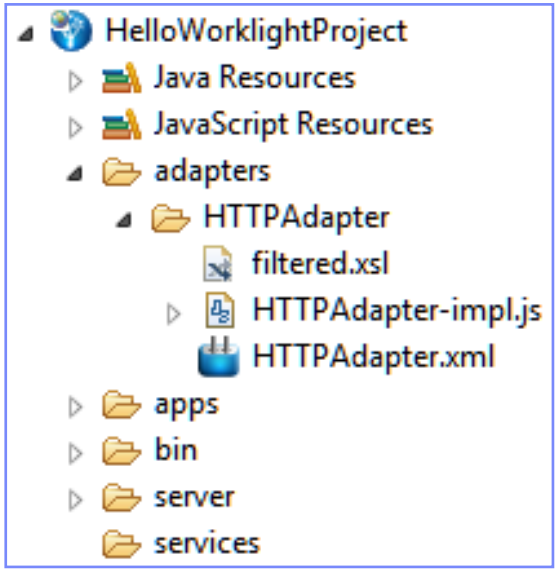


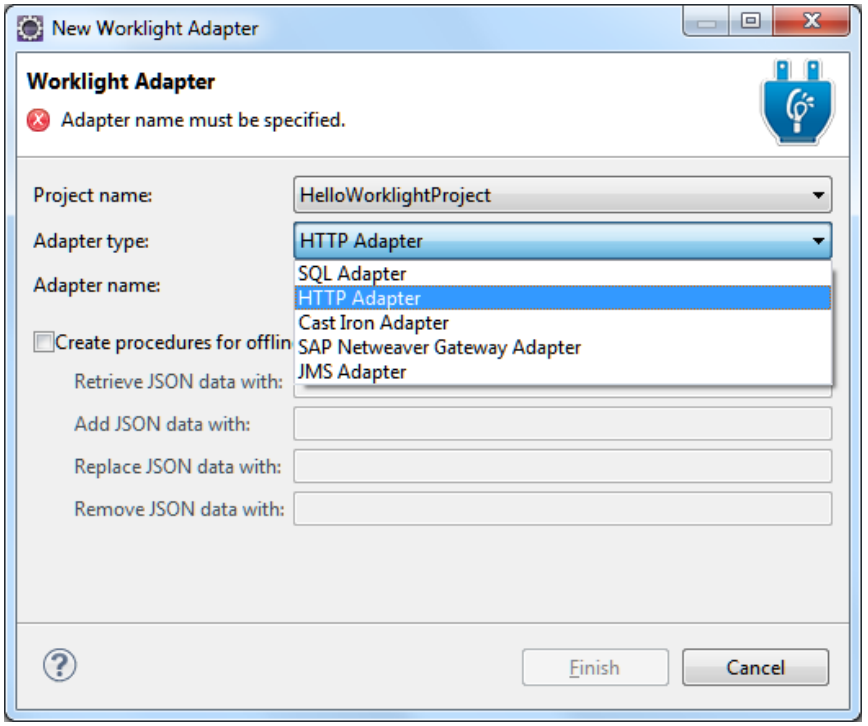
Agenda

- What is it?
- How does it work?
- Creating the adapter
- Using SOAP
  - Creating SOAP-based service request
  - Service request invocation
- Back-end service discovery
- Exercise

Creating the adapter

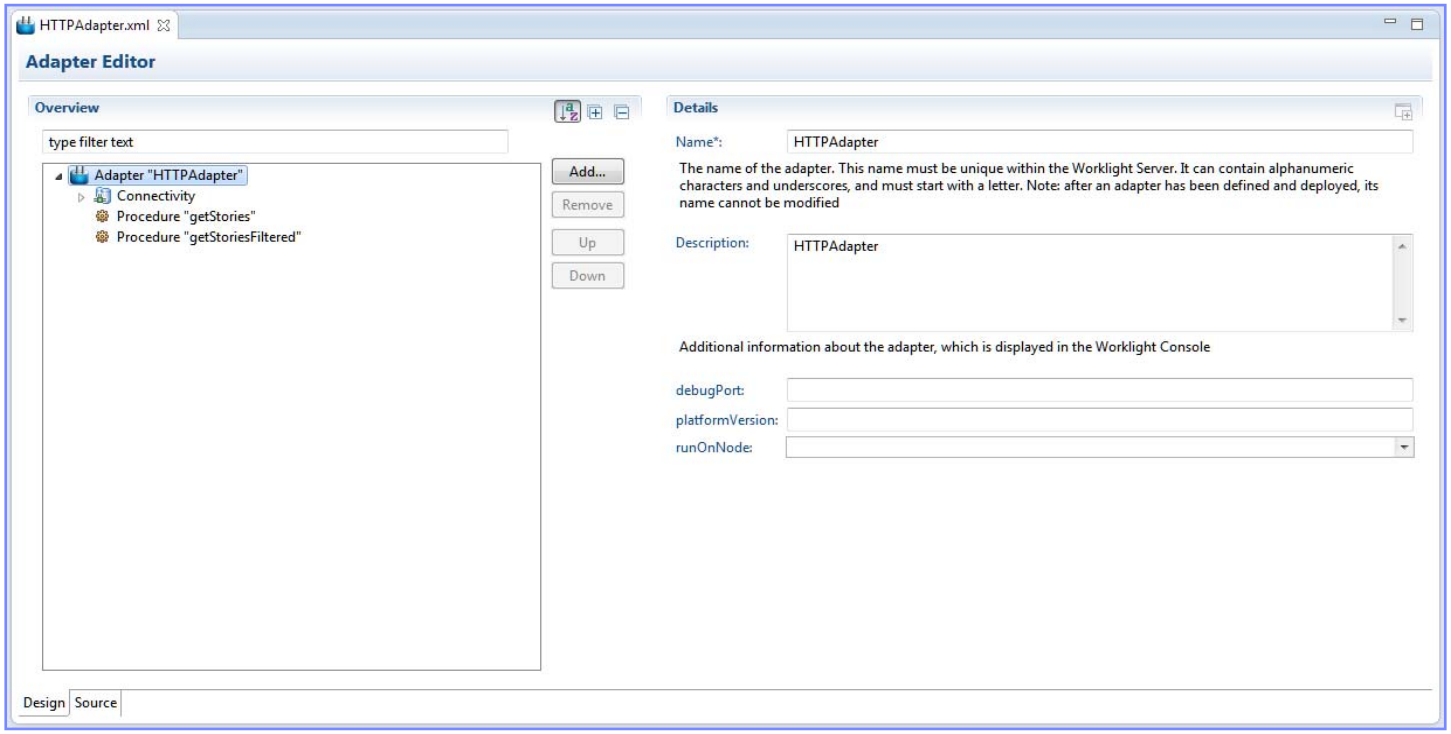
- In Worklight Studio, create a Worklight Adapter.
  - Choose the HTTP Adapter type.
  - A standard HTTP adapter structure is created:





Creating the adapter – continued
Adapter XML editor

- Settings and metadata are stored in the adapter XML file.
- You can use either the Design or Source editor to modify the adapter XML file.



Creating the adapter – continued
XML file: connectivity settings

- To edit the adapter XML file, you must:
  - Set the protocol to HTTP or HTTPS.
  - Set the HTTP domain to domain part of HTTP URL.
  - Set the TCP Port.

```
<connectivity>
    <connectionPolicy xsi:type="http:HTTPConnectionPolicyType">
        <protocol>http</protocol>
        <domain>rss.cnn.com</domain>
        <port>80</port>
        <!-- Following properties used by adapter's key manager
        <sslCertificateAlias></sslCertificateAlias>
        <sslCertificatePassword></sslCertificatePassword>
        -->
    </connectionPolicy>
    <loadConstraints maxConcurrentConnectionsPerNode="2" />
</connectivity>
```

Creating the adapter – continued
XML file: procedures declaration

- Edit the adapter XML file.
- Declare the required procedures.

```
<wl:adapter name="RSSReader"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:wl="http://www.worklight.com/integration"
    xmlns:http="http://www.worklight.com/integration/http">

    <displayName>RSSReader</displayName>
    <description>RSSReader</description>
    <connectivity>
        <connectionPolicy xsi:type="http:HTTPConnectionPolicyType">
            <protocol>http</protocol>
            <domain>rss.cnn.com</domain>
            <port>80</port>
            <!-- Following properties used by adapter's key manager
            <sslCertificateAlias></sslCertificateAlias>
            <sslCertificatePassword></sslCertificatePassword>
            -->
        </connectionPolicy>
        <loadConstraints maxConcurrentConnectionsPerNode="2" />
    </connectivity>

    <procedure name="getStories"/>
    <procedure name="getStoriesFiltered"/>

</wl:adapter>
```

Creating the adapter – continued
JS file: procedures implementation

- Procedures are implemented in the adapter JavaScript file.
- The service URL is used for procedure invocation.
- Some parts of the URL are constant; for example, http://example.com/. They are declared in the XML file.
- Other parts of the URL can be parameterized; that is, substituted at run time by parameter values provided to the Worklight procedure.
- URL parts that can be parameterized are:
  - Path elements
  - Query string parameters
  - Fragments

- See the IBM Worklight user documentation for advanced options for adapters, such as cookies, headers, and encoding.

Creating the adapter – continued
JS file: procedures implementation

- The procedure name in the JavaScript file must be the same as in the XML file.
- The required invocation parameters are `method`, `path`, and `returnedContentType`
- The procedure can be parameterized at run-time.

```
<connectivity>
    <connectionPolicy xsi:type="http:HT
        <protocol>http</protocol>
        <domain>rss.cnn.com</domain>
        <port>80</port>
    </connectionPolicy>
    <loadConstraints maxConcurrentConne
</connectivity>

<procedure name="getStories"/>
<procedure name="getStoriesFiltered"/>
```

The procedure XML file

```
function getStories(interest) {
    path = getPath(interest);

    var input = {
        method : 'get',
        returnedContentType : 'xml',
        path : path
    };


    return WL.Server.invokeHttp(input);
}
```

The procedure JavaScript file

Creating the adapter – continued
JS file: procedures implementation

- To invoke a HTTP request, use the `WL.Server.invokeHttp` method.
- Provide an input parameters object.
- You must specify:
  - The HTTP method: `GET` or `POST`.
  - The returned content type: `XML`, `JSON`, `HTML`, or `plain`.
  - The service path.
  - The query parameters (optional).
  - The request body (optional).
  - The transformation type (optional).

```
function getStoriesFiltered(interest) {
    path = getPath(interest);

    var input = {
        method : 'get',
        returnedContentType : 'xml',
        path : path,
        transformation : {
            type : 'xslFile',
            xslFile : 'filtered.xsl'
        }
    };

    return WL.Server.invokeHttp(input);
}
```

- For a complete list of invocation options, see the IBM Worklight user documentation.

Creating the adapter – continued
XSL transformation filtering

- XSL transformation can be applied to the received data.
- It can be used to filter received data.
- To apply, specify the transformation options in the procedure invocation input parameters.

```
function getStoriesFiltered(interest) {
    path = getPath(interest);

    var input = {
        method : 'get',
        returnedContentType : 'xml',
        path : path,
        transformation : {
            type : 'xslFile',
            xslFile : 'filtered.xsl'
        }
    };

    return WL.Server.invokeHttp(input);
}
```

Agenda

- What is it?
- How does it work?
- Creating the adapter
- Using SOAP
  - Creating SOAP-based service request
  - Service request invocation
- Back-end service discovery
- Exercise

Creating a SOAP-based service request

- The `WL.Server.invokeHttp` method can be used to create a SOAP envelope, which can be sent directly.
  - To invoke a SOAP-based service in an HTTP adapter, you must encode the SOAP XML envelope within the request body.
  - Encoding XML within JavaScript is simple using E4X, which is officially part of JavaScript 1.6.
  - This technology can be used to encode any type of XML document, not only SOAP envelopes.

- If you receive error messages for SOAP envelopes, disable the JavaScript validator.
  - Click **Project > Properties > Builders** and clear **JavaScript Validator**.

Creating a SOAP-based service request – continued

- Use JavaScript to create a SOAP Envelope.

```
var request =
    <soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    "http://schemas.xmlsoap.org/soap/envelope/">
        <soap:Body>
            <CelsiusToFahrenheit xmlns="http://tempuri.org/">
                <Celsius>{celsiusTemp}</Celsius>
            </CelsiusToFahrenheit>
        </soap:Body>
    </soap:Envelope>;
```

Creating a SOAP-based service request – continued

- It is possible to insert Java™ code and variables into SOAP XML. It is evaluated at run-time.

```
var request =
<soap:Envelope
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <messageHeader>
            <originatingIP>
                { WL.Server.configuration["local.IPAddress"] }
            </originatingIP>
            <requestTimestamp>
             { new Date().toLocaleString() }
            </requestTimestamp>
        </messageHeader>

        <ConversionRate xmlns="http://www.webserviceX.NET/">
            <FromCurrency>USD</FromCurrency>
            <ToCurrency>EUR</ToCurrency>
        </ConversionRate>
    </soap:Body>
</soap:Envelope>;
```

Service request invocation

- The `WL.Server.invokeHttp(options)` method is used to invoke a request for a SOAP service.

- The Options object must include the following properties:
  - A method – usually `POST`.
  - A returnedContentType – usually `XML`.
  - A path – a service path.
  - A body:
    - content – SOAP XML as a string
    - contentType

```
var options = {
    method : 'post',
    returnedContentType : 'xml',
    path : '/webservices/tempconvert.asmx',
    body: {
        content: request.toString(),
        contentType: 'text/xml; charset=utf-8'
    }
};
```

Service request invocation – continued

- Full SOAP-based service invocation procedure example:

```
function temperatureConvertor(celsiusTemp) {
    // Use USD and EUR as fromCurrency and toCurrency variables. Case is impo

    var request =
        <soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        "http://schemas.xmlsoap.org/soap/envelope/">
            <soap:Body>
                <CelsiusToFahrenheit xmlns="http://tempuri.org/">
                    <Celsius>{celsiusTemp}</Celsius>
                </CelsiusToFahrenheit>
            </soap:Body>
        </soap:Envelope>;

    var input = {
        method : 'post',
        returnedContentType : 'xml',
        path : '/webservices/tempconvert.asmx',
        body: {
            content: request.toString(),
            contentType: 'text/xml; charset=utf-8'
        }
    };

    var result = WL.Server.invokeHttp(input);

    return result.Envelope.Body;
}
```

SOAP envelope

Options

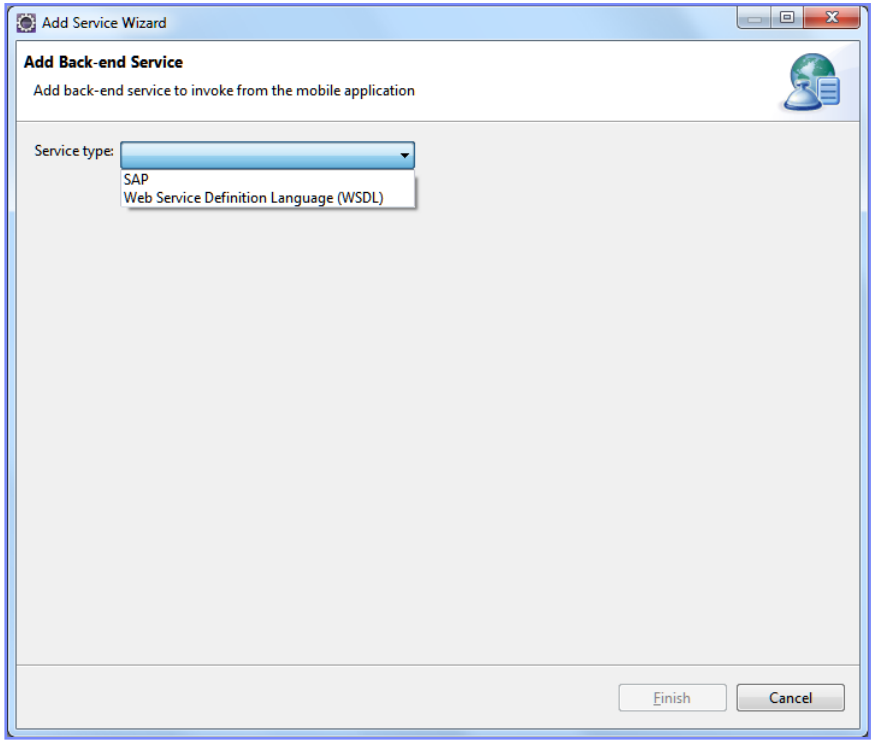
Request invocation

Agenda

- What is it?
- How does it work?
- Creating the adapter
- Using SOAP
  - Creating SOAP-based service request
  - Service request invocation
- Back-end service discovery
  - Exercise

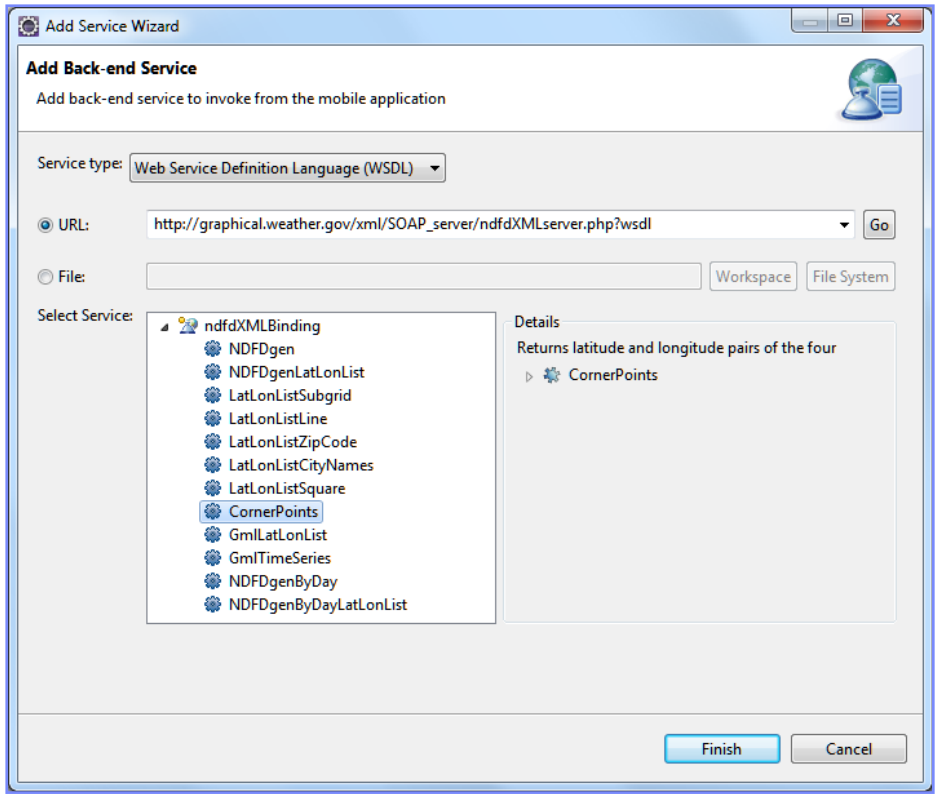
Back-end service discovery

- If you are developing HTTP adapters for SOAP (or SAP) services, it is possible to reduce development time by using the Discover Back-end Services tool to auto-generate the adapter with procedures based on the provided WSDL.
- Right-click on the **services** folder in a Worklight project and choose **Discover back-end services**.
- Select the type of service, **SAP** or **SOAP**:



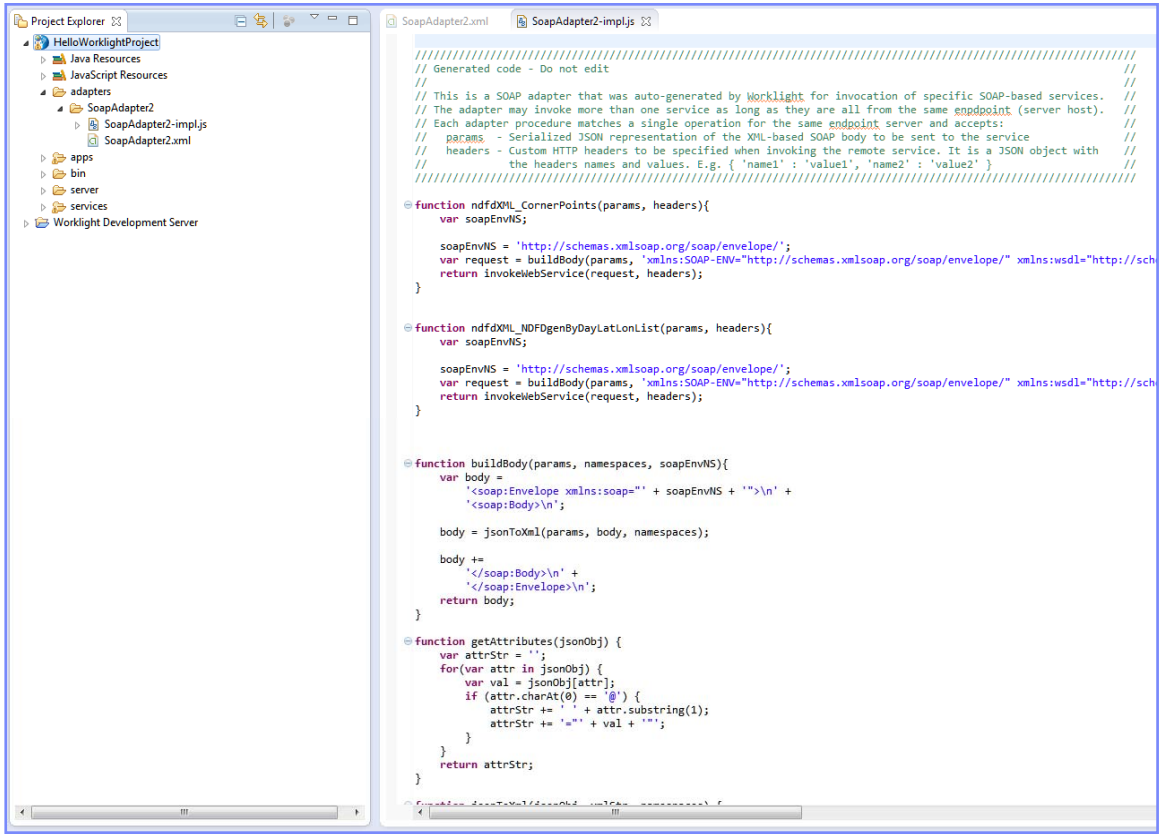
Back-end service discovery – continued

- Add the services location to use and select the service you'd like to add to the adapter.
- This should be repeated for each service you'd like to add; it will be added to the same adapter.



Back-end service discovery – continued

- The end result is an auto-generate adapter:



Agenda

- What is it?
- How does it work?
- Creating the adapter
- Using SOAP
  - Creating SOAP-based service request
  - Service request invocation
- Back-end service discovery
- Exercise

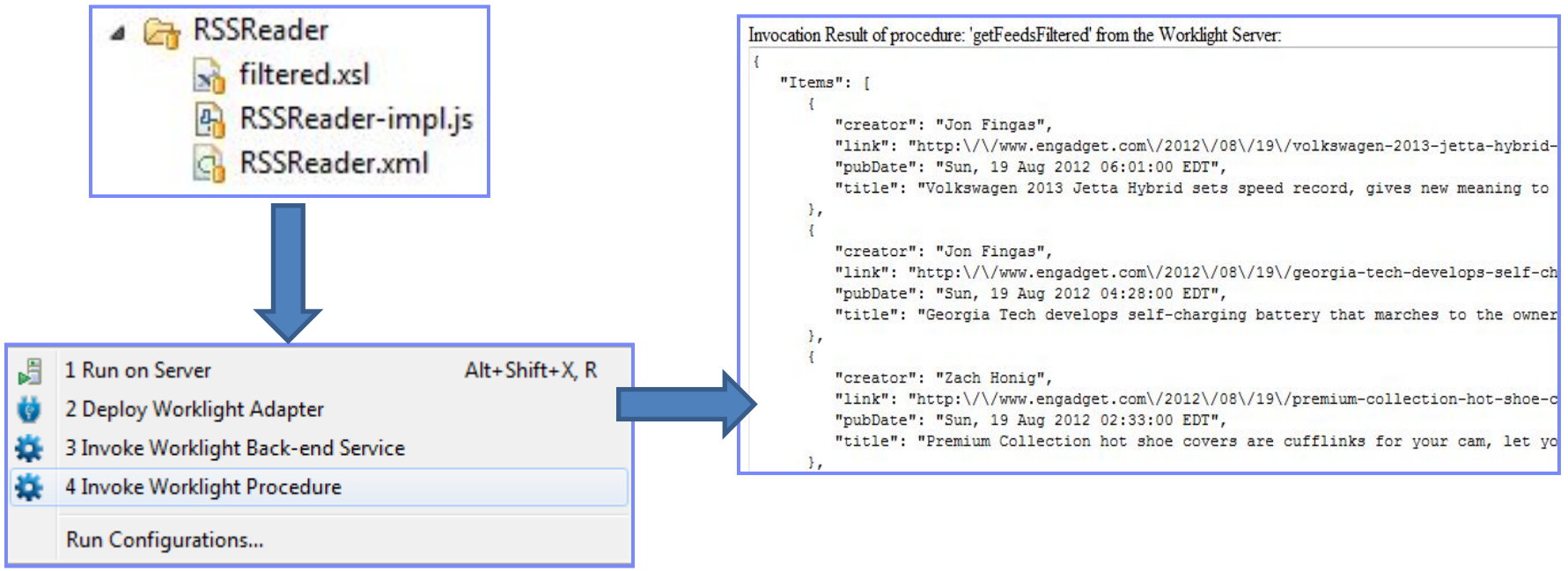
Exercise

Engadget RSS Reader

- Create a Worklight HTTP adapter called **RSSReader**.
- Connect to the Engadget RSS feed at http://engadget.com/rss.xml
- Declare and implement a `getFeeds` procedure, which retrieves the RSS feed.
- Declare and implement a `getFeedsFiltered` procedure, which does the same as `getFeeds`, but returns only the `title`, the `creator`, the `pubDate`, and the `link` fields.
- Deploy the adapter and use Worklight Studio to test your procedures as described in this module.

Exercise
Solution

- 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

- HTTP adapters can be used to:
  - Work with RESTful services.
  - Work with SOAP services.
  - Issue GET and POST requests.
  - All of the above.
- What format of data can the HTTP adapter retrieve and automatically parse?
  - XML
  - JSON
  - Plain text
  - All of the above
- Can you use the HTTP adapter with non-standard HTTP ports?
  - You must use port 80 for HTTP and port 443 for HTTPS.
  - You can use any port for HTTP but only port 443 for HTTPS.
  - You must use port 80 for HTTP but can use any port for HTTPS.
  - It is possible to use any port for both HTTP and HTTPS.

Check yourself questions

- HTTP adapters can be used to:
  - Work with RESTful services.
  - Work with SOAP services.
  - Issue GET and POST requests.
  - All of the above.
- What format of data can the HTTP adapter retrieve and automatically parse?
  - XML
  - JSON
  - Plain text
  - All of the above
- Can you use the HTTP adapter with non-standard HTTP ports?
  - You must use port 80 for HTTP and port 443 for HTTPS.
  - You can use any port for HTTP but only port 443 for HTTPS.
  - You must use port 80 for HTTP but can use any port for HTTPS.
  - It is possible to use any port for both HTTP and HTTPS.

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 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