JavaScript HTTP Adapter

Overview

By using IBM MobileFirst Platform Foundation HTTP adapters, you can send GET or POST HTTP requests and retrieve data from the response headers and body. HTTP adapters work with RESTful and SOAP-based services, and can read structured HTTP sources such as RSS feeds.

You can easily customize HTTP adapters with simple server-side JavaScript code. For example, you could set up server-side filtering if necessary. The retrieved data can be in XML, HTML, JSON, or plain text format.

The adapter is configured with XML to define the adapter properties and procedures.
Optionally, it is also possible to use XSL to filter received records and fields.

Prerequisite: Make sure to read the JavaScript Adapters (../) tutorial first.

The XML File

The XML file contains settings and metadata.
To edit the adapter XML file, you must:

- Set the protocol to HTTP or HTTPS.
- Set the HTTP domain to the domain part of HTTP URL.
- Set the TCP Port.

Declare the required procedures below the connectivity element:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<mfp:adapter name="JavaScriptHTTP"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:mfp="http://www.ibm.com/mfp/integration"
    xmlns:http="http://www.ibm.com/mfp/integration/http">

    <displayName>JavaScriptHTTP</displayName>
    <description>JavaScriptHTTP</description>
    <connectivity>
        <connectionPolicy xsi:type="http:HTTPConnectionPolicyType">
            <protocol>https</protocol>
            <domain>mobilefirstplatform.ibmcloud.com</domain>
            <port>443</port>
            <connectionTimeoutInMilliseconds>30000</connectionTimeoutInMilliseconds>
            <socketTimeoutInMilliseconds>30000</socketTimeoutInMilliseconds>
            <maxConcurrentConnectionsPerNode>50</maxConcurrentConnectionsPerNode>
        </connectionPolicy>
    </connectivity>

    <procedure name="getFeed"/>
    <procedure name="getFeedFiltered"/>

</mfp:adapter>
```

JavaScript implementation

A service URL is used for procedure invocations. Some parts of the URL are constant; for example, http://example.com/.
Other parts of the URL can be parameterized; that is, substituted at run time by parameter values that are provided to the MobileFirst procedure.

The following URL parts can be parameterized.

- Path elements
- Query string parameters
- Fragments

> See the topic about "The connectionPolicy element of the HTTP adapter" in the user documentation for advanced options for adapters, such as cookies, headers, and encoding.

To call an HTTP request, use the `WL.Server.invokeHttp` method.
Provide an input parameter object, which must specify:

- The HTTP method: `GET`, `POST`, `PUT`, `DELETE`
- 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 getFeed() {
  var input = {
      method : 'get',
      returnedContentType : 'xml',
      path : "feed.xml"
  };


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

> See the topic about "WL.Server.invokeHttp" in the user documentation for a complete list of options.

XSL transformation filtering

You can also apply XSL transformation to the received data, for example to filter the data.
To apply XSL transformation, create a **filtered.xsl** file next to the JavaScript implementation file.

You can then specify the transformation options in the input parameters of the procedure invocation. For example:

```
function getFeedFiltered() {

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

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

> For more information on XSL transformation, refer to the user documentation.

Creating a SOAP-based service request

You can use the `WL.Server.invokeHttp` method to create a **SOAP** envelope, which can be sent directly.

To call 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: just use **E4X**, which is officially part of JavaScript 1.6. You can use this technology to encode any type of XML document, not only SOAP envelopes.

Use JavaScript to create a SOAP Envelope. It is possible to insert JavaScript code and variables into SOAP XML. Such additional code 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>
          <GetCitiesByCountry xmlns="http://www.webserviceX.NET">
            <CountryName>{countryName}</CountryName>
          </GetCitiesByCountry>
        </soap:Body>
      </soap:Envelope>;
```

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

The Options object must include the following properties:

- A `method` property: usually `POST`
- A `returnedContentType` property: usually `XML`
- A `path` property: a service path
- A `body` property: `content` (SOAP XML as a string) and `contentType`

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

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

Sample adapter

Click to download (https://github.com/MobileFirst-Platform-Developer-Center/Adapters/tree/release80/) the Adapters Maven project.

Sample usage

- Use either Maven or MobileFirst Developer CLI to build and deploy the JavaScriptHTTP adapter (../../creating-adapters/).
- To test or debug an adapter, see the testing and debugging adapters (../../testing-and-debugging-adapters) tutorial.