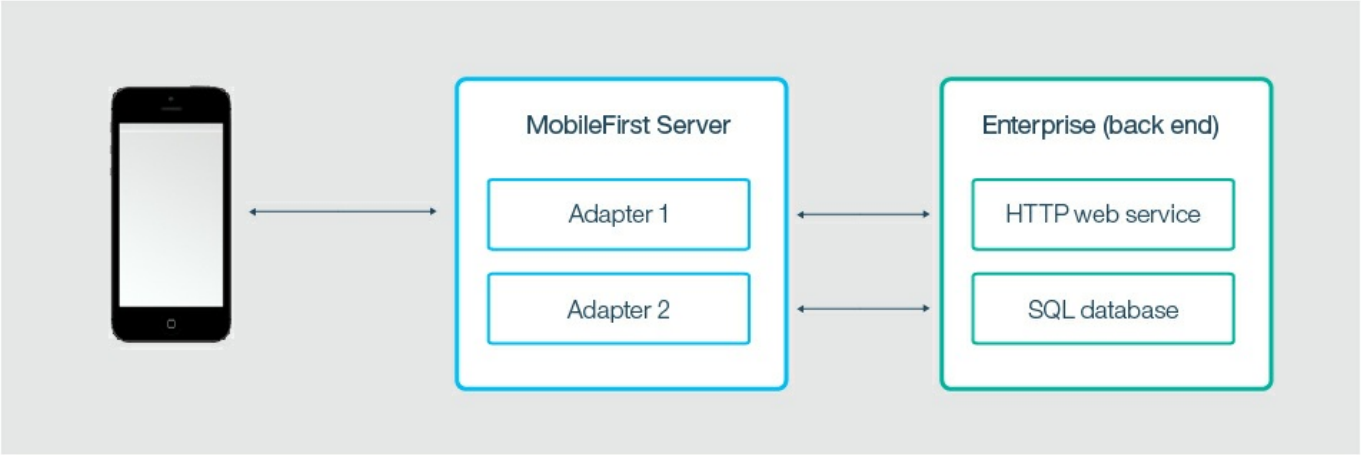
Adapters overview

Overview

This tutorial presents how to develop server-side code (adapters) that is used to transfer and retrieve information from back-end systems to client applications and cloud services. MobileFirst Server processes the information and handles security. You can write adapters in JavaScript or Java.

This tutorial covers the following topics:

- Benefits of using adapters
- JavaScript adapters
- Java adapters
- Creating adapters
- Deploying adapters
- Testing adapters
- Tutorials to follow next



Benefits of using adapters

Universality

- Adapters support multiple integration technologies and back-end information systems.

Read-only and transactional capabilities

- Adapters support read-only and transactional access modes to back-end systems.

Fast development

- Adapters use simple XML syntax and are easily configured with JavaScript API or Java API.

Security

- Adapters use flexible authentication facilities to create connections with back-end systems.
- Adapters offer control over the identity of the connected user.

Transparency

- Data that is retrieved from back-end applications is exposed in a uniform manner, regardless of the adapter type.

Benefits specific to Java adapters

- Ability to fully control the URL structure, the content types, the request and response headers, content and encoding
- Easy and fast development and testing by using MobileFirst Studio or the command-line interface (CLI)
- Ability to test the adapter without MobileFirst Studio or CLI, by using a 3rd-party tool such as Postman
- Easy and fast deployment to a running MobileFirst Server instance with no compromise on performance and no downtime
- Security integration with the MobileFirst security model with no additional customization, by using simple annotations in the source code

JavaScript adapters

JavaScript adapters provide templates for connection to various back-ends, such as HTTP, SQL, Cast Iron, SAP JCo, and SAP Netweaver. JavaScript adapters also provide a service discovery wizard, which you can use to autogenerate adapters for connecting to WSDL services and more.

> Learn more about JavaScript adapters (../javascript-adapters/)

Java adapters

Java adapters expose a full REST API to the client and are written in Java. This type of adapters is based on the JAX-RS specification (https://jax-rs-spec.java.net/nonav/2.0-rev-a/apidocs/index.html).
In Java adapters, it is up to the developer to define the returned content and its format, as well as the URL structure of each resource.

> Learn more about Java adapters (../java-adapter/)

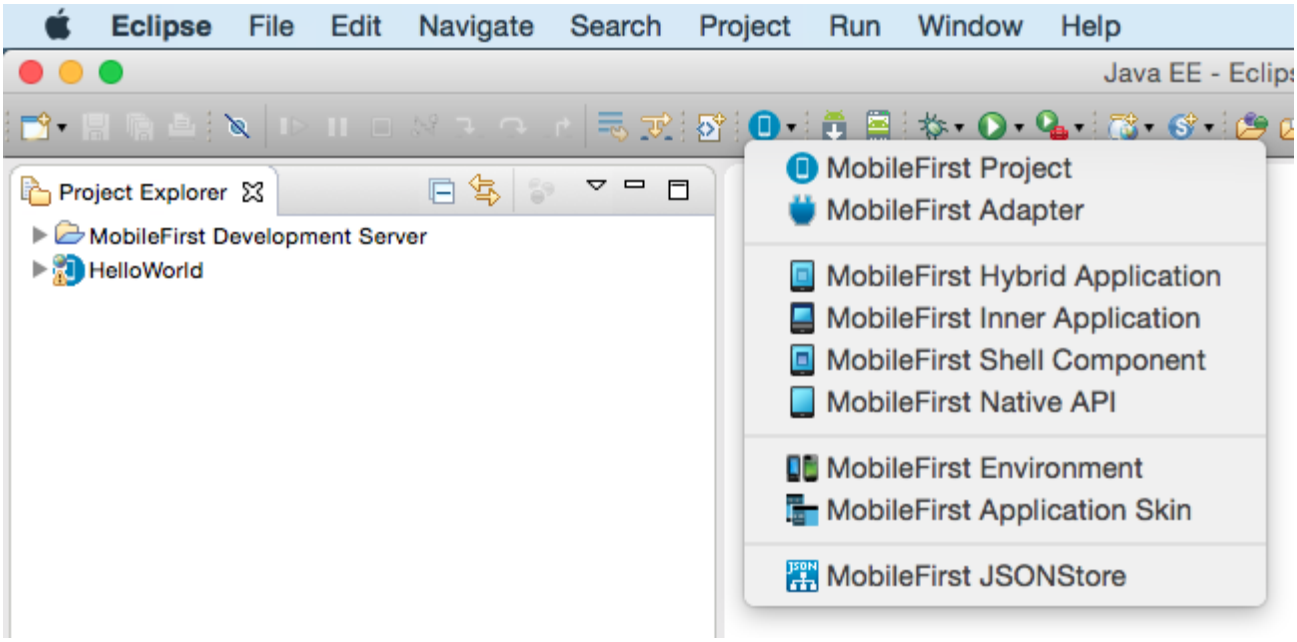
Creating adapters

CLI

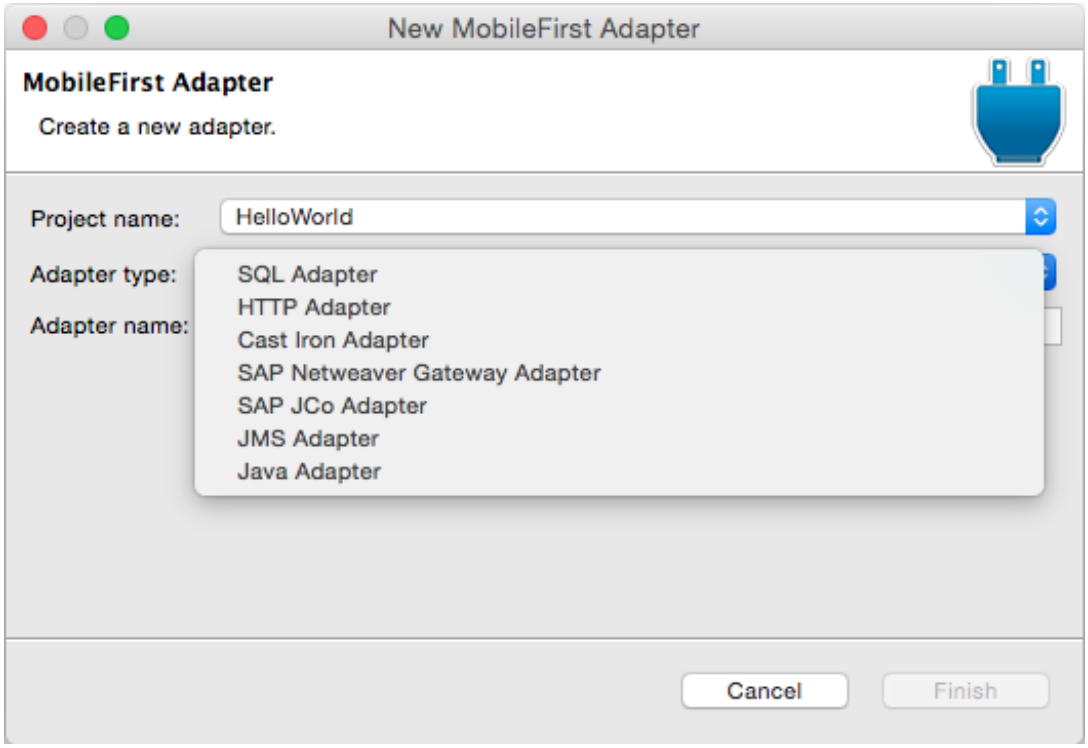
From the project's directory, use `mfp add adapter` and follow the interactive instructions.

Studio

1. In Eclipse, click the MobileFirst icon that is located in the toolbar and select **MobileFirst Adapter**.



2. Select a MobileFirst project and an adapter type. This generates the template of the adapters.



3. Select an adapter type and type an adapter name. Applications use this name to access the adapter.
4. Click **Finish**.

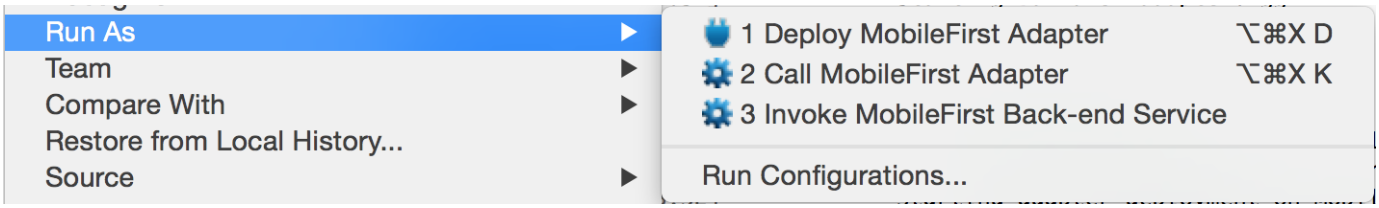
Deploying adapters

CLI

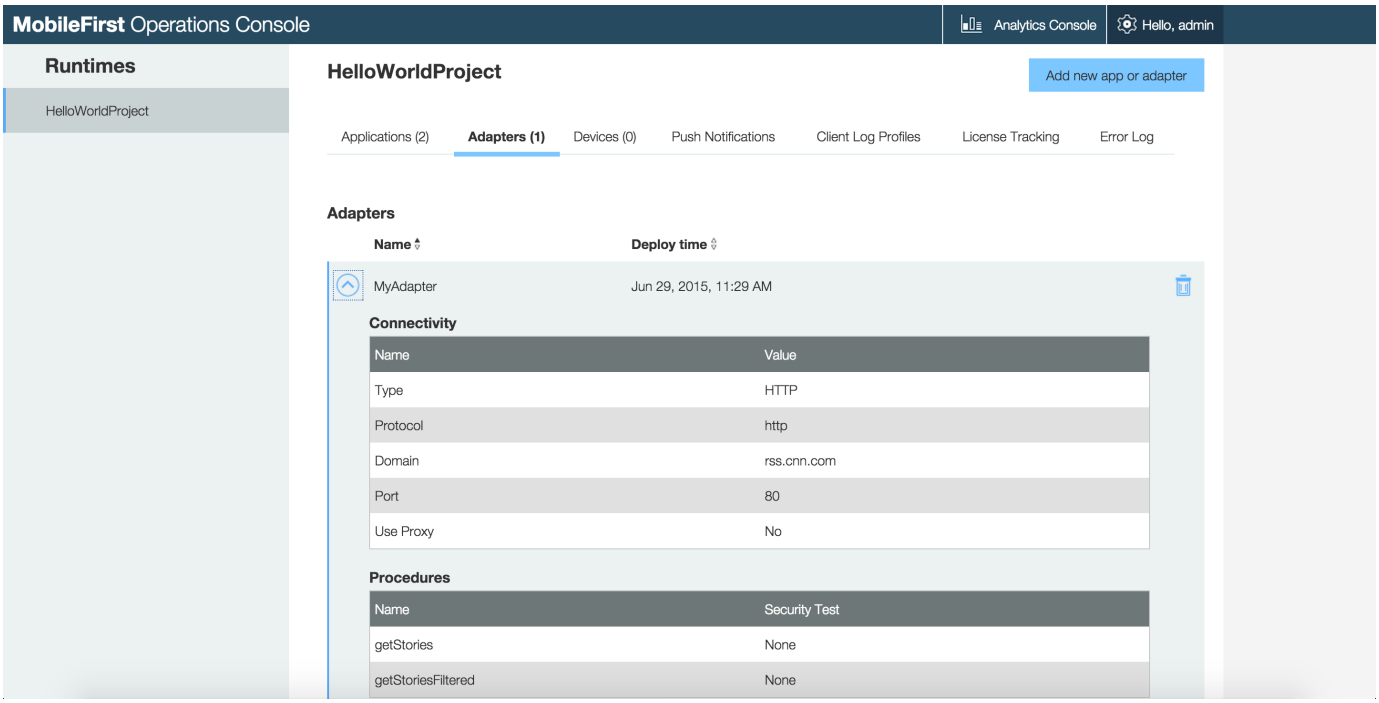
In the terminal, go to the adapter's directory: `$ cd adapters/TestAdapt/`.
Use `mfp push` to build and deploy the current adapter.

Studio

1. Select an adapter to deploy.
2. Right-click the adapter and select **Run As > Deploy MobileFirst Adapter**.



MobileFirst Studio archives the adapter code and deploys it to the MobileFirst Server instance.
You can see the deployed adapter in the MobileFirst Console (../../hello-world/introduction-to-mobilefirst-platform-operations-console/).



Testing adapters

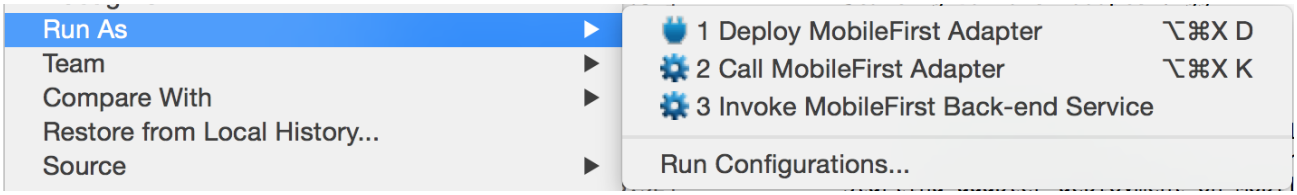
CLI

To run a procedure test, make sure your adapter is built and deployed, then use `$ mfp adapter call`.
Follow the interactive instructions.

Studio

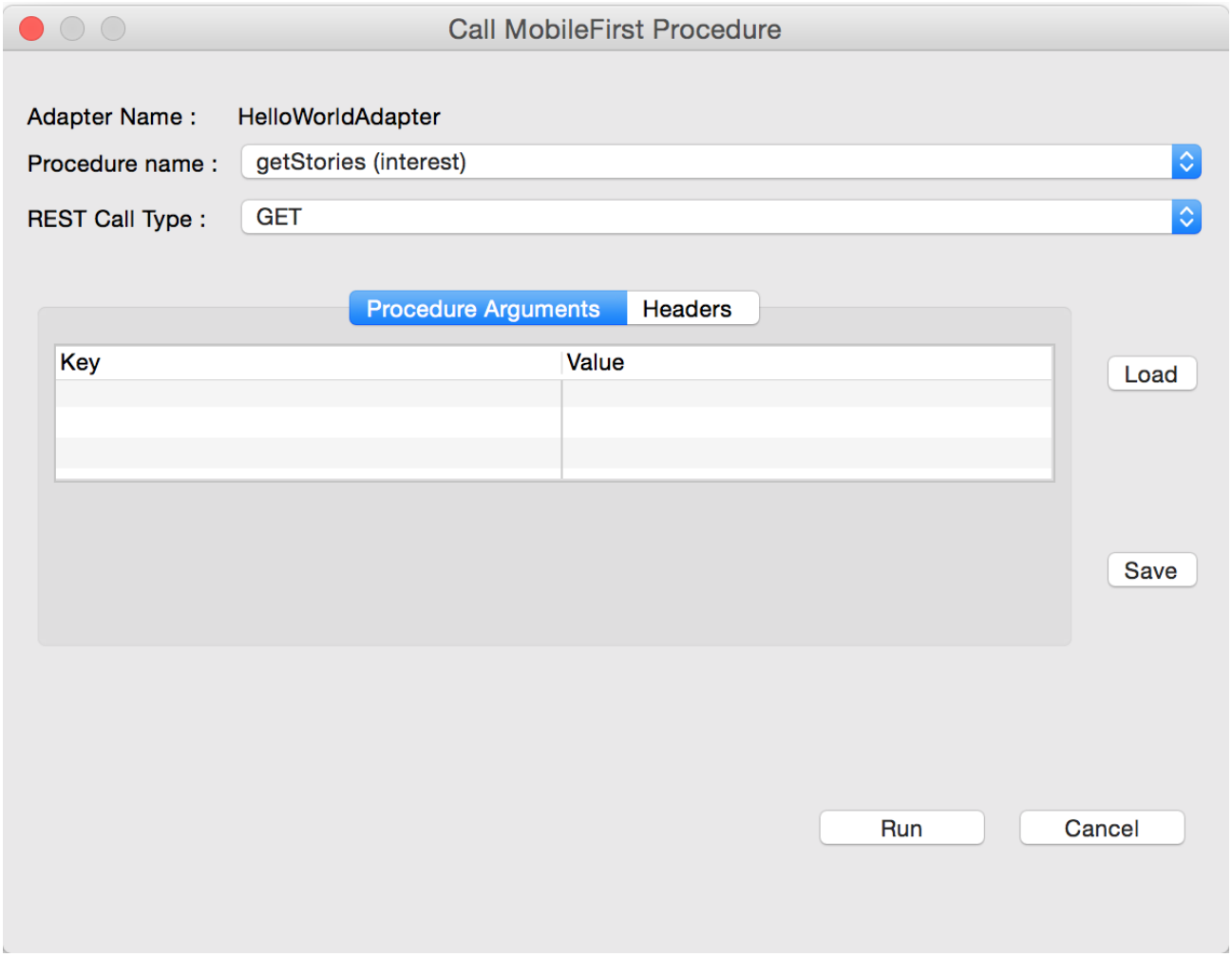
To test adapter procedures, you can use MobileFirst Studio.
To run a procedure test:

1. Select **Run As > Call MobileFirst Procedure**.

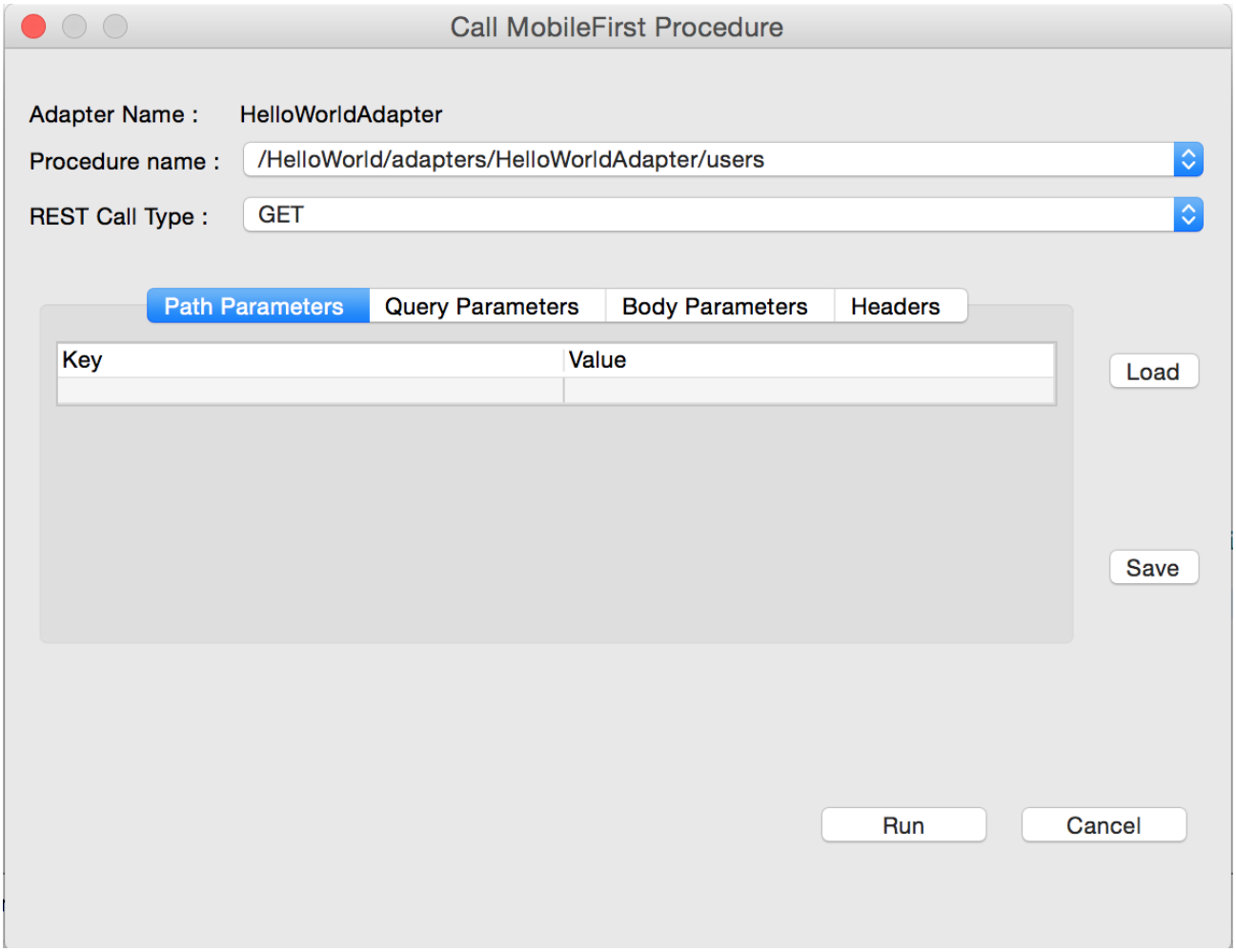


2. Select the procedure that you want to test.
3. Enter key values and click **Run**.

JavaScript adapters



Java adapters



In Postman

MobileFirst adapters are available via a REST interface. This means that if you know the URL of a resource/procedure, you can use HTTP tools such as Postman to test requests and pass URL parameters, path parameters, body parameters or headers as you see fit.

The URL to access your adapter procedure is:
`/{project-context}/adapters/{adapter-name}/{procedure-name}`

When using Java adapters, parameters can be passed in the URL, body, form, etc, depending on how you configured your adapter.

When using JavaScript adapters, parameters are passed as `params=[a,b,c,d]`, in other words, a JavaScript procedure receives only **one** parameter called params which needs to be an array of ordered, unnamed values. This parameter can either be in the URL (`GET`) or in the body (`POST`) using `Content-Type: application/x-www-form-urlencoded`.

If your resource is protected by a security test, the request prompts you to provide a valid authorization header. Note that by default, MobileFirst uses a simple security scope even if you did not specify any. So unless you specifically disabled security, the endpoint is always protected.

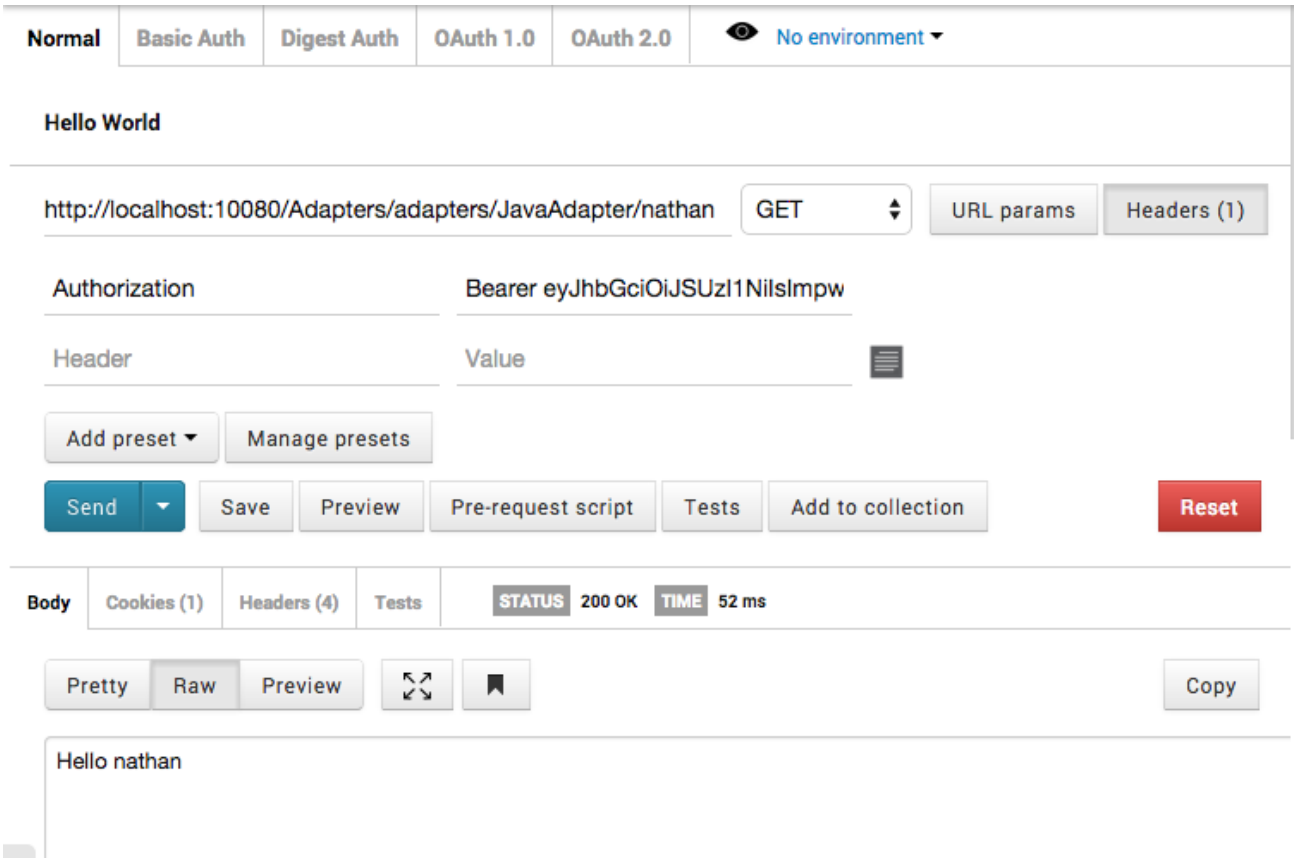
For you to work around this during your development stage, the development version of the MobileFirst server includes a test token endpoint.

To receive a Test Token, make an HTTP `POST` request to `http(s)://{server_ip}:{server_port}/{project_name}/authorization/v1/testtoken` with your HTTP client (Postman).

You receive a JSON object with a temporary valid authorization token:

```
{
"Authorization": "Bearer eyJhbGciOiJSUzI1NiIsImpwayI6eyJhbGciOiJSU0EiLCJleHAiOiJBUUFCIi
wibW9kIjoiQU0wRGQ3eEFkdjZILXlnTDdyOHFDTGRFLTNJMm2FPZUlxb2UtckpBMHVadXcyckh
oWFozV1ZDZUtlelJWY0NPWXNRTi1tUUswbWZ6NV8zby1ldjBVWXdYa1NPd0JCbDFFaHFJd1Z
Ed09pZWcySk1HbDBFWHNQWmZrTlpJLUhVNG9NaWktVHJOTHp..."
}
```


In your next requests to your adapter endpoints, add an HTTP header with the name "`Authorization`" and the value you received previously (starting with `Bearer`).
The security framework now skips any security challenges protecting your resource.



> For more information about JavaScript and Java adapters, see the topic about "MobileFirst adapters overview" in the user documentation.

Tutorials to follow next

Follow the tutorials in the server-side development section (../) to learn more about HTTP, SQL, Cast Iron, and JMS JavaScript adapters, Java adapters, invoking adapter procedures from hybrid and native applications, advanced adapter usage and more.