Confidential Clients

Overview

When accessing a resource protected by the MobileFirst Platform framework, the MobileFirst Platform Foundation client SDK (for Cordova, iOS, Android and Windows) provide the tools to handle the security features.

Non-mobile clients that do not use the MobileFirst client SDK can also request protected resources, by acting as a **confidential client**.
For example, your backend server may need to request a protected resource, or use one of the MobileFirst Platform **REST APIs** such as **Push Notifications**.

Registered confidential clients can obtain a token to be used in all requests to the MobileFirst Server. This flow is based on the client credentials flow (https://tools.ietf.org/html/rfc6749#section-1.3.4) of the OAuth specification.

Registering the confidential client

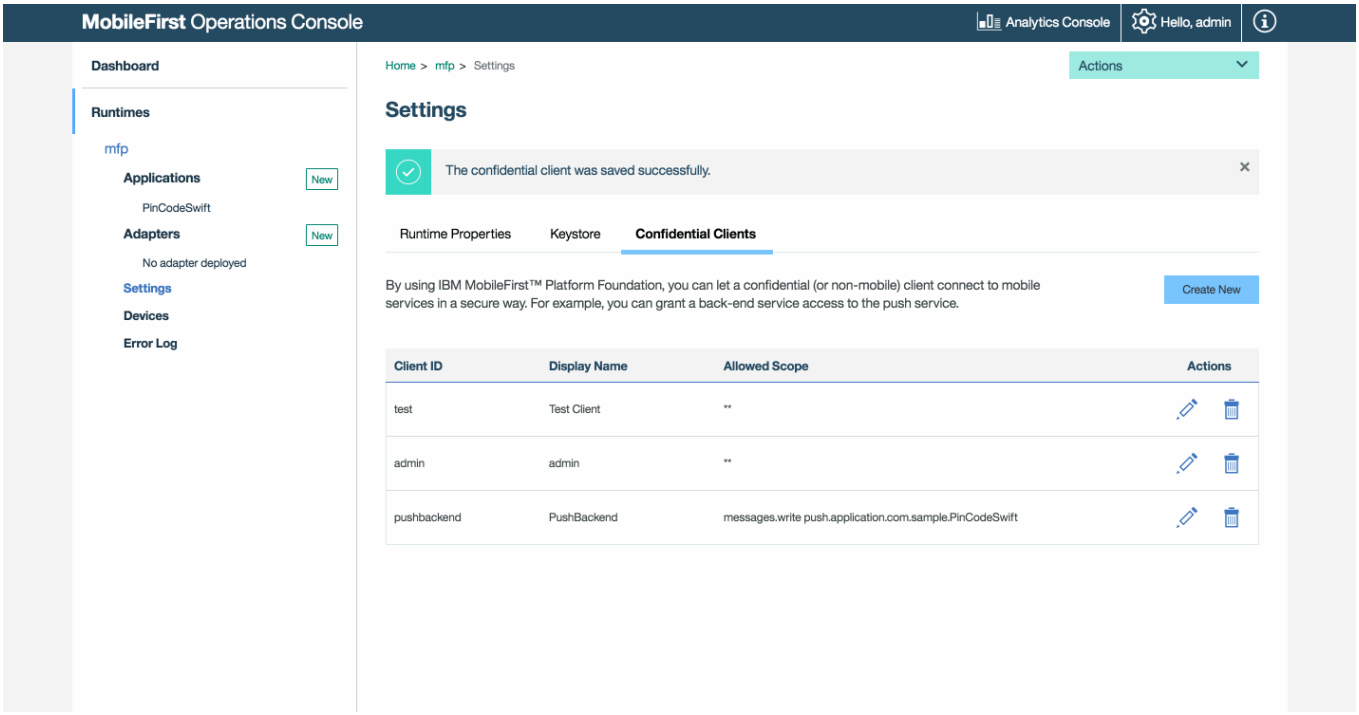
In the MobileFirst Operations Console, under **Settings → Confidential Clients**, click on **Create New** to add a new entry. You will need to provide the following:

- **Display Name**: A friendly display name that describes the confidential client, such as **Backend Node server**.
- **ID**: A unique identifier for the confidential client (can be considered as a "username").
- **Secret**: A private passphrase to authorize access from confidential client (can be considered as an API key).
- **Allowed Scope**: A confidential client using the above ID and Secret combination will automatically be granted the scope that is defined here (learn more about **Scopes** in the Authorization Concepts (../authorization-concepts/#scope) tutorial).

Examples of scopes:

- Protecting external resources (../protecting-external-resources) uses the scope `authorization.introspect`.
- Sending a Push Notification (../../notifications/sending-push-notifications) via the REST API uses the space-separated scope elements `messages.write` and `push.application.<applicationId>`.
- Adapters may be protected by a custom scope element, such as `accessRestricted`.
- The scope `*` is a catch-all scope, granting access to any requested scope.

Any scope can use the `*` character to replace any other valid character. For example `push.application.*` would match any `push.application.<applicationId>`.



Predefined confidential clients

The MobileFirst Platform Server comes with some predefined confidential clients:

test

The `test` client is only available in development mode. It allows you easily to test your resources.

- **ID**: `test`
- **Secret**: `test`
- **Allowed Scope**: `*` (any scope)

admin

The `admin` client is used internally by the MobileFirst Platform administration service.

push

The `push` client is used internally by the MobileFirst Platform push service.

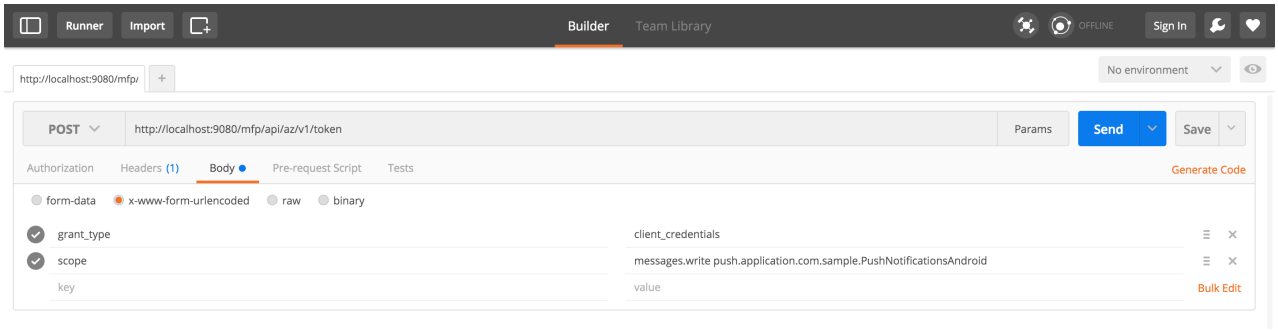
Obtaining an access token

A token can be obtained from MobileFirst Server's **token endpoint**.

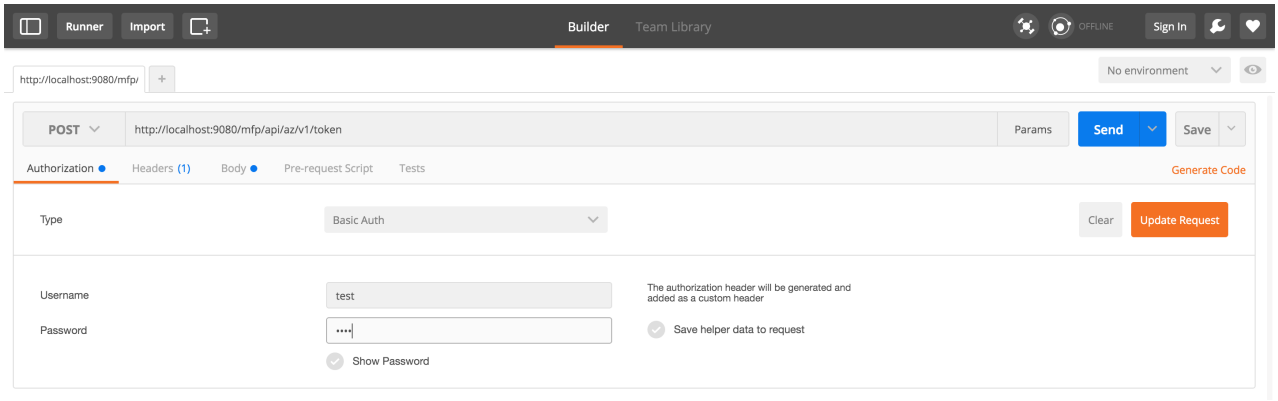
For **testing purposes**, you can use Postman as described below.
In a real setting, the bellow should be implemented in your backend logic, with the technology of your choice.

1. Make a **POST** request to **http(s)://[ipaddress-or-hostname]:[port]/[runtime]/api/az/v1/token**.
   For example: `http://localhost:9080/mfp/api/az/v1/token`
   - In a development environment, the MobileFirst Server uses a pre-existing "mfp" runtime.
   - In a production environment, replace the runtime value with your runtime name.
2. Set the request with a content-type of `application/x-www-form-urlencoded`.

3. Set the following two form parameters:
    - `grant_type`: `client_credentials`
    - `scope`: Use the scope protecting the resource.
      If you don't use a scope to protect your resource, use an empty string.



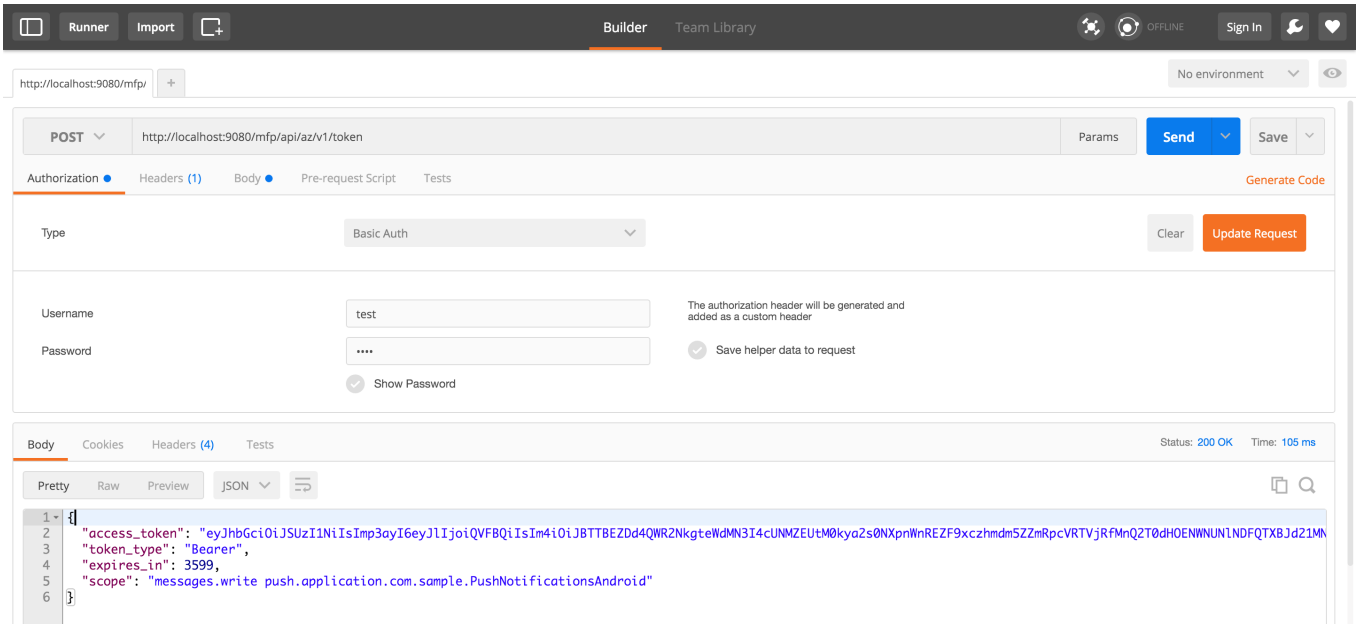
4. The request should be authenticated using Basic Authentication (https://en.wikipedia.org/wiki/Basic_access_authentication#Client_side). Use your confidential client's **ID** and **secret**.



   Outside of Postman, if using the **test** confidential client, you should have the **HTTP header** set to `Authorization: Basic dGVzdDp0ZXN0` (`test:test` encoded using **base64**).

The response for this request will contain a `JSON` object, including the **access token** and its expiration time (1 hour).

```
{
  "access_token": "eyJhbGciOiJSUzI1NiIsImp ...",
  "token_type": "Bearer",
  "expires_in": 3599,
  "scope": "sendMessage accessRestricted"
}
```



Using the access token

From here on, requests can be made to the desired resources by adding the **HTTP header**: `Authorization: Bearer eyJhbGciOiJSUzI1NiIsImp ...`, replacing the access token by the one you extracted from the previous JSON object.

Possible responses

In addition to the normal responses that your resource may generate, there are a few responses to look out for, generated by the MobileFirst Platform server:

Bearer

An HTTP **401** response status with the HTTP header `WWW-Authenticate : Bearer` means that no token was found on the `Authorization` header of the original request.

invalid_token

An HTTP **401** response status with the HTTP header `WWW-Authenticate: Bearer error="invalid_token"` means that the token that was sent is **invalid** or **expired**.

insufficient_scope

An HTTP **403** response status with the HTTP header `WWW-Authenticate : Bearer error="insufficient_scope", scope="scopeA scopeB"` means that the token found in the original request did not match the **scope required by this resource**. The header also includes the scope it expected.

When making a request, if you do not know which scope is required by the resource, `insufficient_scope` is the way to determine the answer.
For example, request a token with an empty string (`""`) as the scope value and make a request to the resource. Then, you can extract the required scope from the 403 response and request a new token for this scope.