Lab 7-1: Send User Accounts Updates using an Event Hook

|  |  |
| --- | --- |
| **Objective** | Implement an Okta event hook to received updates on user account events |
| **Scenario** | Okta Ice would like to get notified whenever a user account is being update, for example, a user changed the password, or a user profile is updated. |
| **Duration** | 20 minutes |

**Get an NGrok Authorization token**

1. On your VM, on the **Windows Taskbar**, open a new browser window.
2. Navigate to http://www.ngrok.com.
3. If you do not already have an ngrok account, click the “Sign Up” button.
4. Fill in your name, email, a password, and click the checkbox indicating you are not a robot.
5. The ensuing screen under Item #2 shows your ngrok authorization token…
6. Open a command window, and enter the following at the following at the prompt: <screen shot with authToken>, and press Enter.
7. nGrok should respond with a message indicating that a .yml file has been created.

**Open the Project in Visual Studio 2019**

1. On your VM, on the **Windows Taskbar**, click the **Visual Studio 2019** icon.
2. Click **Open a Project or Solution**.
3. Navigate to **<wherever we decide to put it>,** and double-click the **NetCoreHooks.sln file.**

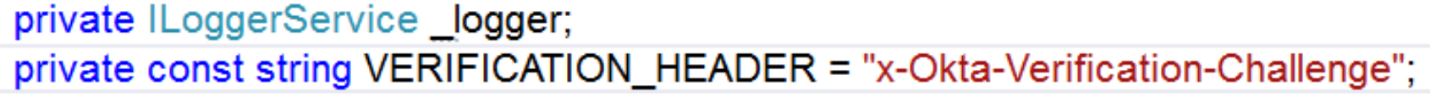
**Explore the EventController Class**

1. In the Visual Studio Solution Explorer window, open the Controllers folder. Double-click on EventController.cs.
2. Notice this controller contains 2 actions: **Get** and **Post**.
   1. The **Get** method verifies the hook by accepting a verification header, and sending the value back to Okta.
   2. The **Post** method uses the Http Request object to capture incoming Json. The method parses the Json using Linq-to-Json. It creates a VerificationResponse object and populates it with the parsed Json data.

**Implement the Verification Function**

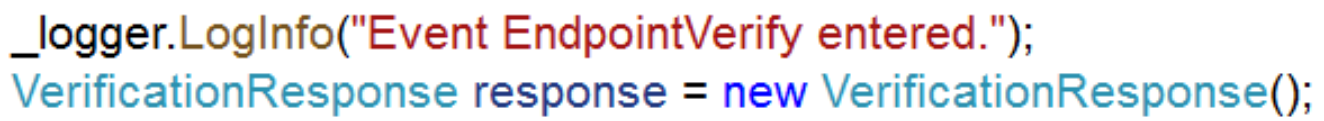
In the EventController class, locate the Get method.

Locate the constant declaration, VERIFICATION\_HEADER (around line 23Change the value of this constant

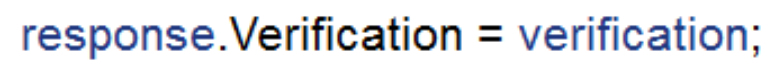


Locate the **Get** function (around line 75).

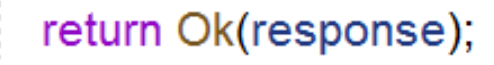
Just after the LogInfo call, instantiate the verification object (around line 78)…



Set the response.verification value (around line 87)…



Return 200 Ok value with verification object (around line 90)…



**Implement the Post (user-events) Function**

Locate the AccountEvents.json sample file in the <folder name> folder. This is the schema of the json file Okta will send in the Post execution.

Locate the **Post** function (around line 34).

Inside the function, parse the events information from the HTTP request…

A screenshot of a cell phone

Description automatically generated

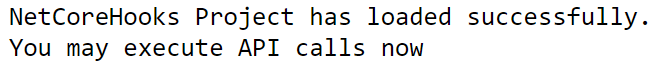
Navigate to the model folder

Locate the OktaEvents class.

Examine the overridden ToString() method. This is the value that will be printed to the log file on successful execution.

Execute the Hook Project

1. Run the NetCoreHooks application by clicking on the IIS Express button in the Visual Studio toolbar..
   1. **A close up of a logo

      Description automatically generated**
2. A browser window will open with this message…
   1. 

**Deploy the Hook Project**

1. Run the NetCoreHooks application by clicking on the IIS Express button in the Visual Studio toolbar..
2. Wait until the console tab displays the message **Started HookprojectApplication**.
3. Leave IntelliJ opened.
4. Close any existing command prompt windows inside the VM.
5. Open a new command prompt inside your VM.
6. Enter the following command to start **ngrok:**

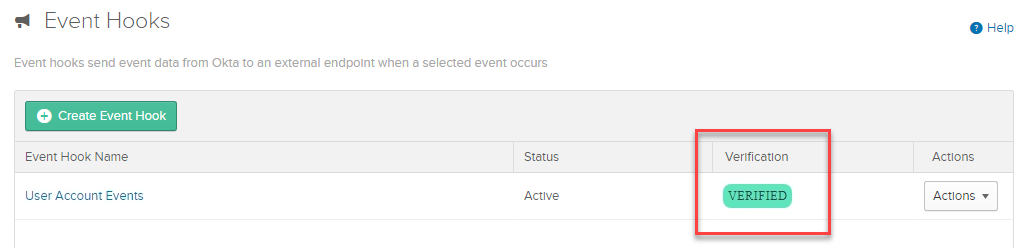
**ngrok http 8080**

**Create and Verify the Event Hook**

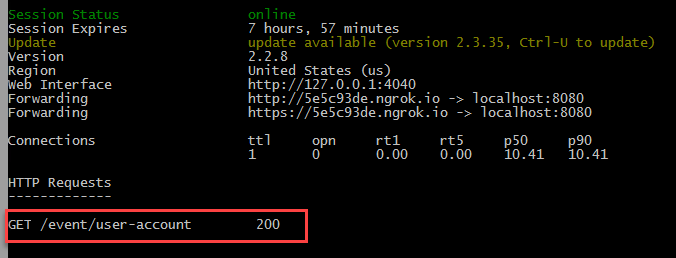
1. Sign into your Okta org as **okta.service**.
2. In the Admin console, click **Workflow** -> **Event Hooks**.
3. Click **Create Event Hook**.
4. Complete the fields as follows:

|  |  |
| --- | --- |
| **Field** | **Value** |
| Name | **User Account Events** |
| URL | **https://<your\_ngrok-address>/event/user-account**  for example, https://*a59d8d5a.ngrok.io*/event/user-account |
| Authentication field | **x-api-key** |
| Authentication secret | **Tra!nme4321** |
| Subscribe to events | **User’s Okta profile updated**  **User’s Okta password reset by an admin**  **User’s Okta password updated** |

1. Click **Save and Continue**.
2. On the Verify Endpoint Ownership window, click **Verify**.
3. The User Account Events hook is created and verified:

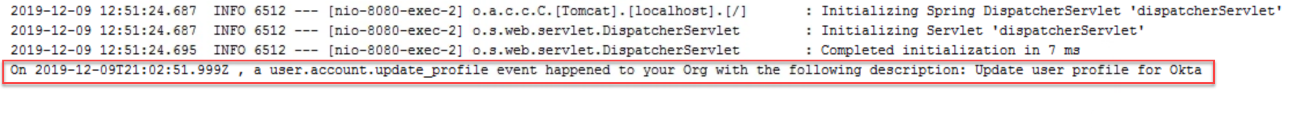


1. Go back to the ngrok command line window, you should see a new GET request:

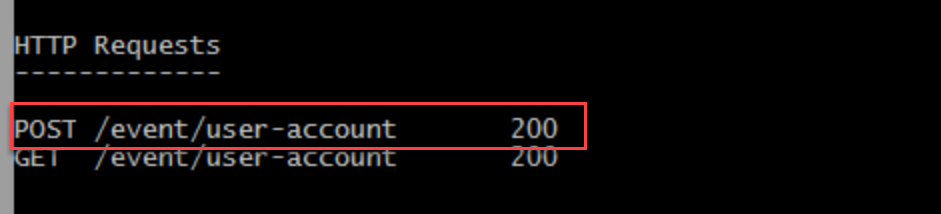


**Test the Event Hook**

1. In the Admin console, click **Directory** -> **People**.
2. Select **Kay West** and click the **Profile** tab.
3. Click **Edit**.
4. Update the Middle name field with **O'Reilly** and click **Save**.
5. In IntelliJ, the event message displayed in the console:



1. In the ngrok command line window, you should see a new POST request:



Go back to IntelliJ and click **Run -> Stop ‘HookprojectApplication’.**