IBM Developer

SKILLS NETWORK

{ width=200 height=200 }

Creating IBM Cloud Functions Entities

Estimated time needed: 45 minutes

In this lab you will begin creating IBM Cloud Functions entities! You will create and invoke actions and sequences, and you will be encouraged to explore the web action and API management capabilities provided by Cloud Functions. All this will familiarize you with how to start using IBM Cloud Functions yourself.

Objectives

After completing this lab, you will be able to:

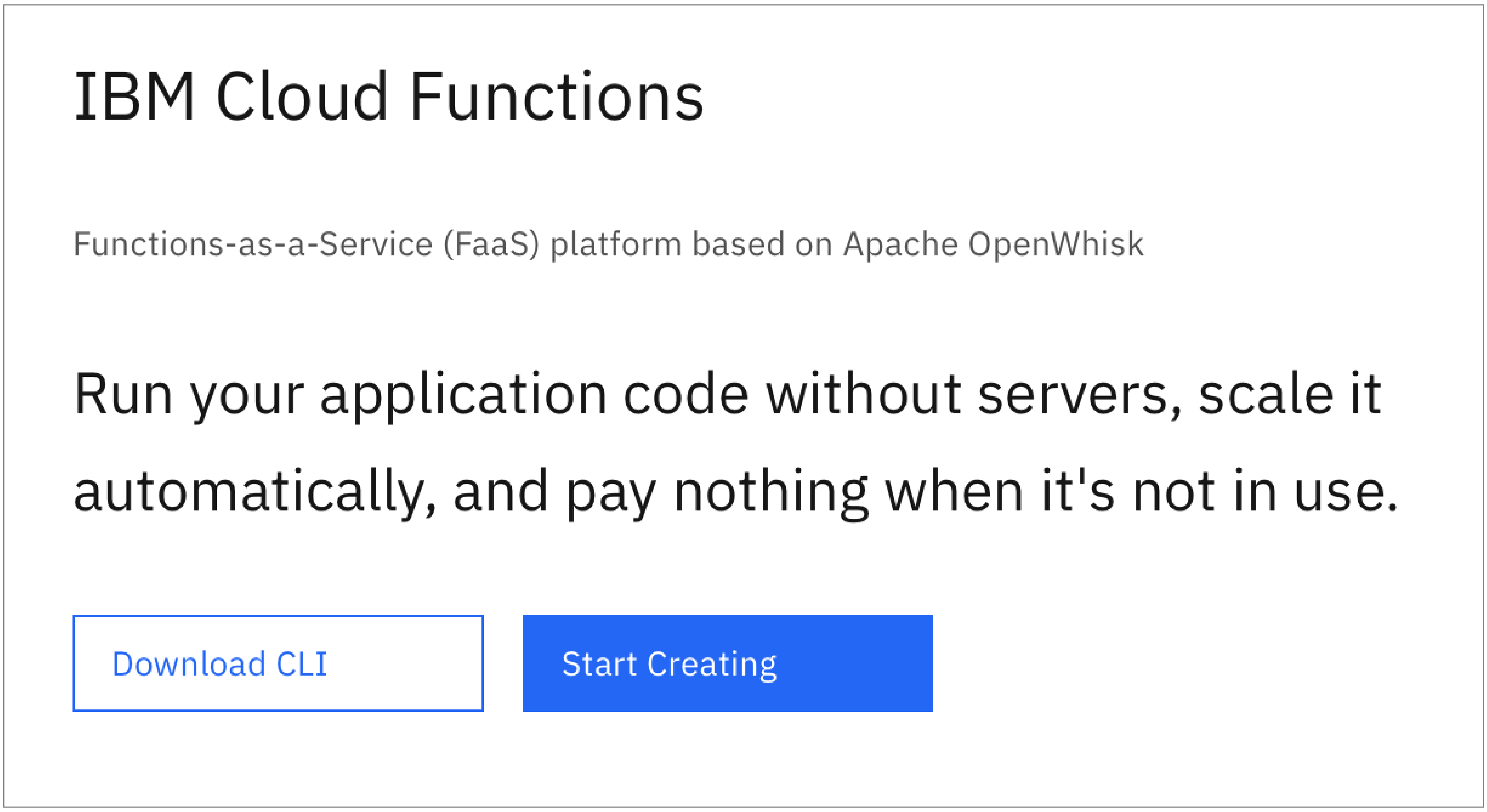
1. Create and invoke an action
2. Use custom Node.js code for an action
3. Create and invoke a sequence
4. Enable a web action and access an action using HTTP
5. Create an API for your action

Exercise 1 : Create and Invoke an Action

In this exercise, you will create and invoke your first IBM Cloud Functions action.

1. Go to the IBM Cloud Functions UI.

2. Click **Start Creating**.

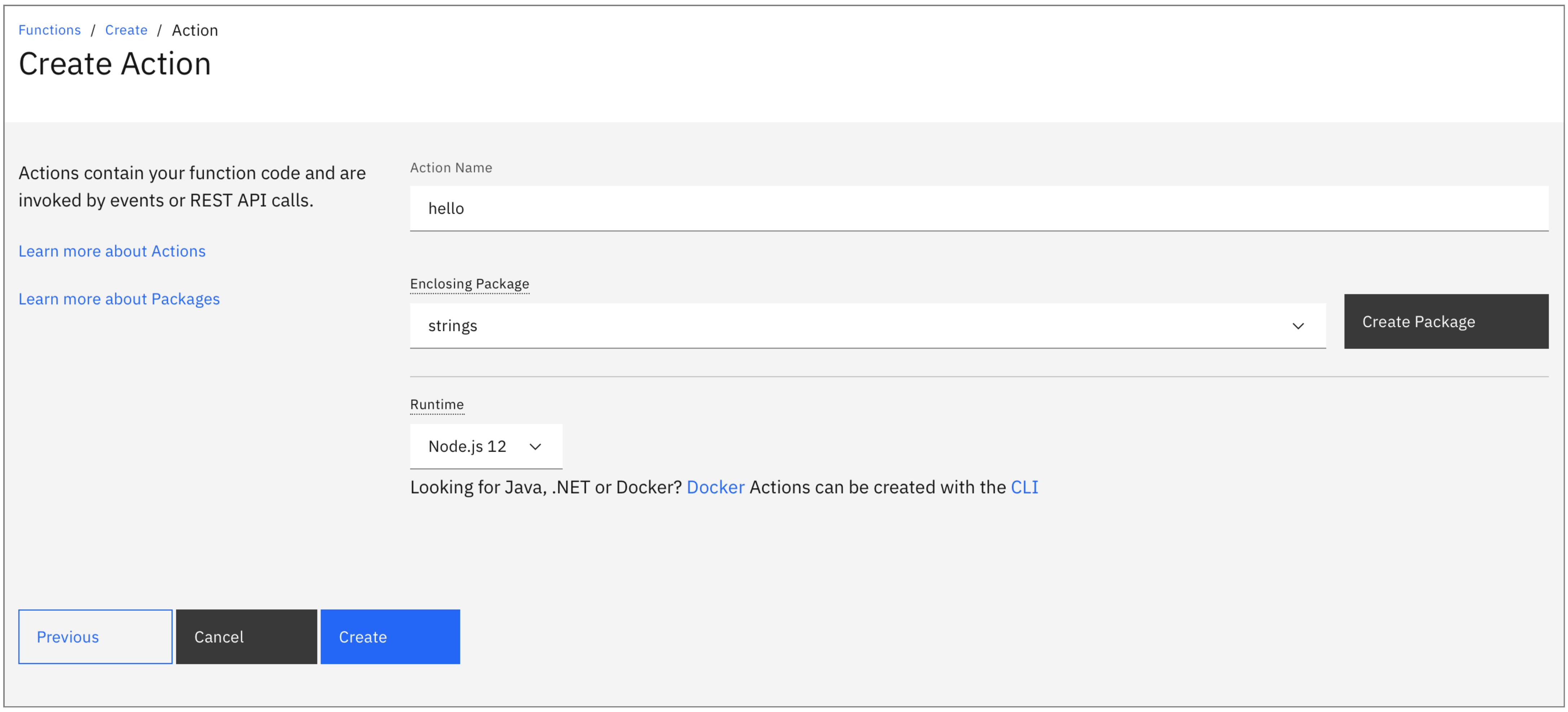


IBM Cloud Functions

Functions-as-a-Service (FaaS) platform based on Apache OpenWhisk

Run your application code without servers, scale it automatically, and pay nothing when it's not in use.

1. Click **Action** in order to create an action.
2. Name your action **hello**.
3. Click **Create Package** in order to put this action into a package.
4. Name the package **strings** since we'll make several actions that perform string manipulation.
5. Click **Create**.
6. Choose a Node.js runtime.
7. Click **Create**.



Create Action

Actions contain your function code and are invoked by events or REST API calls.

Action Name: hello

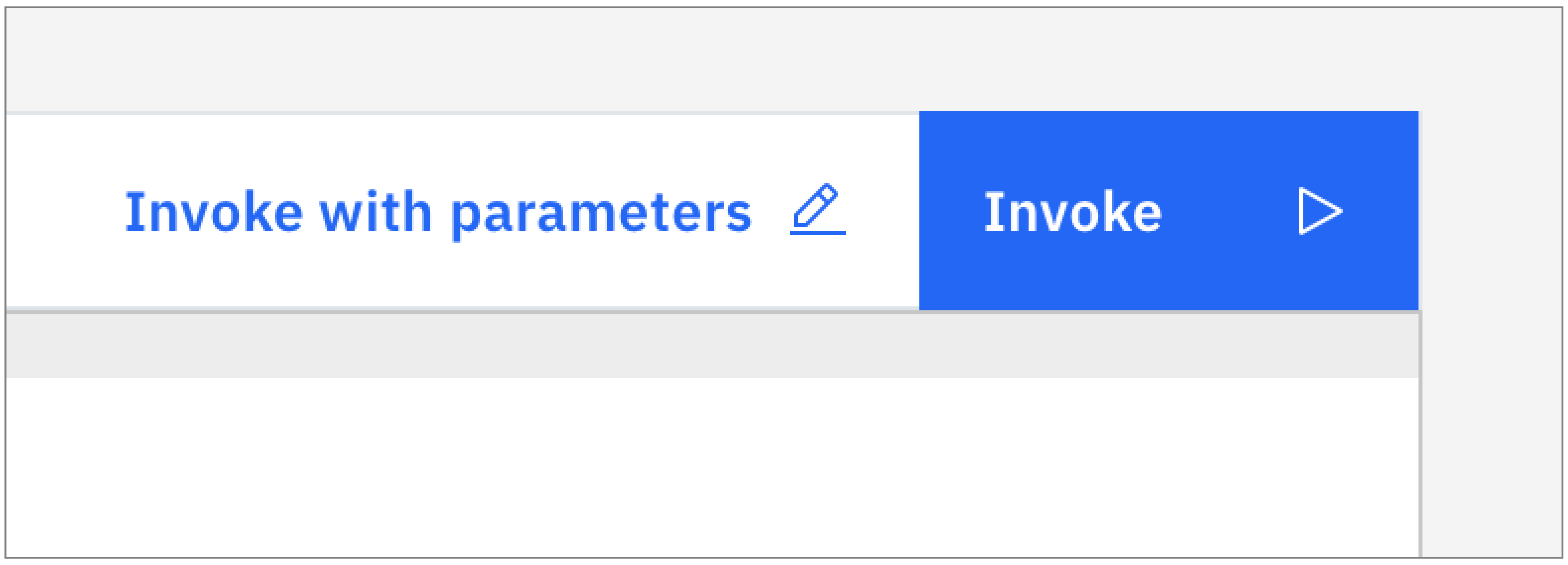
Enclosing Package: strings

Runtime: Node.js 12

Looking for Java, .NET or Docker? Docker Actions can be created with the CLI

Now you have created an action! Your newly created action should be pre-populated with Node.js code to return a hello world message.

1. Click **Invoke**.

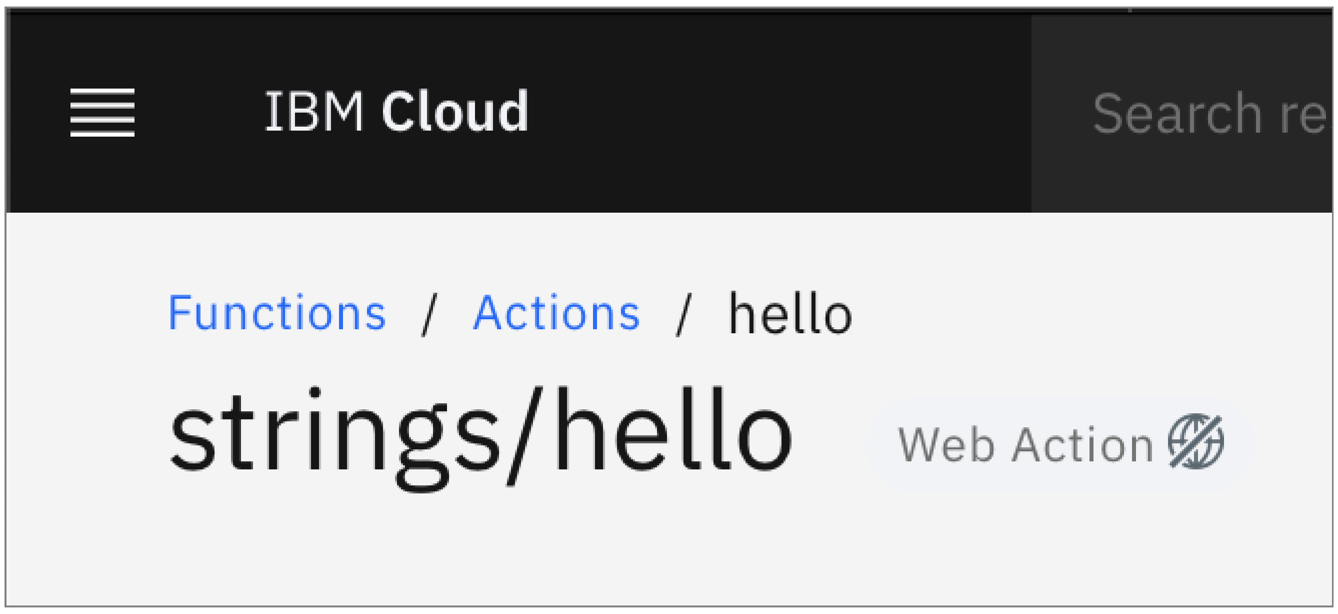


The activations should appear in a panel on the right. Wait for a second and the results should appear within your activation.

Exercise 2 : Create a Second Action

In this exercise you will create a second action that performs a slightly more advanced task.

1. Click **Actions** in the breadcrumbs on the top right to return to the actions page. When it loads, you should see your **strings** package with your **echo** action in it.



2. Click **Create** and then **Action** to create another action.
3. Name your action **capitalize**.
4. Select **strings** as the enclosing package.
5. Choose a Node.js runtime.
6. Click **Create**.
7. Use the following code for the main function in the code editor.

```
function main(params) {
  return { message: params.message.toUpperCase() };
}
```

8. Click **Save**.
9. Click **Invoke with parameters**.
10. Put the following in the input box:

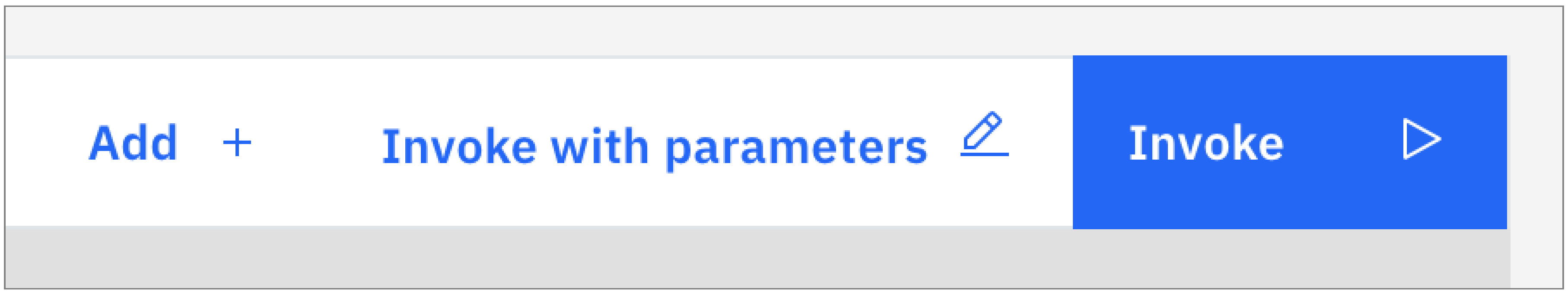
```
{
  "message": "I'm hungry!"
}
```

11. Click **Apply**.
12. Click **Invoke**.
13. Check the activation results to see your message capitalized.

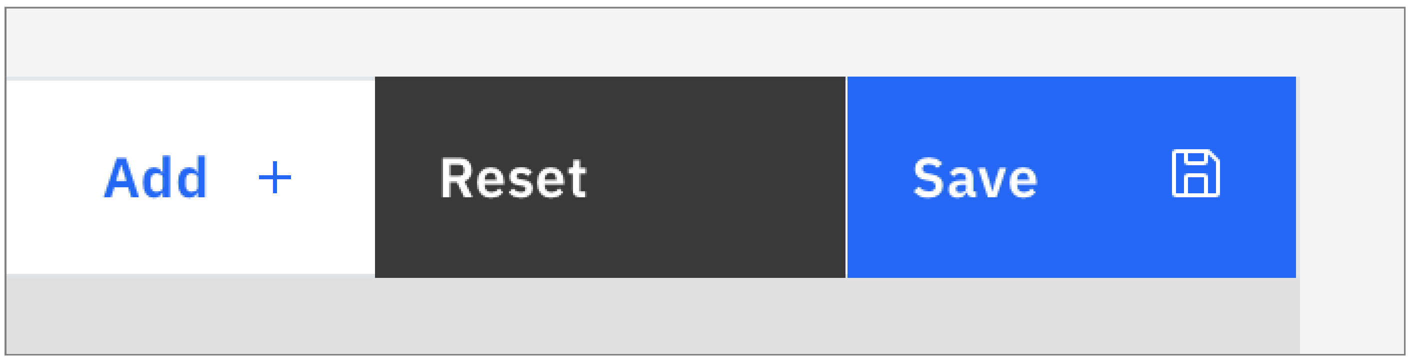
Exercise 3 : Create and Invoke a Sequence

In this exercise you will create a sequence using your two previously created actions. You'll then invoke the sequence.

1. Click **Actions** in the breadcrumbs on the top right to return to the actions page. When it loads, you should see your **strings** package with your **hello** action in it.
2. Click **Create** and then **Sequence**.
3. Name the sequence **hello_capitalize**.
4. Select **strings** as the enclosing package.
5. Select an existing action for this sequence and choose **strings / hello**.
6. Click **Create**.
7. Click **Add +** to add another action to this sequence.



8. Click **Select Existing** and choose **strings / capitalize**.
9. Click **Add**.
10. Click **Save** to save the sequence.



11. Click **Invoke with parameters**.
12. Put the following in the input box:

```
{
  "text": "Testing my sequence."
}
```

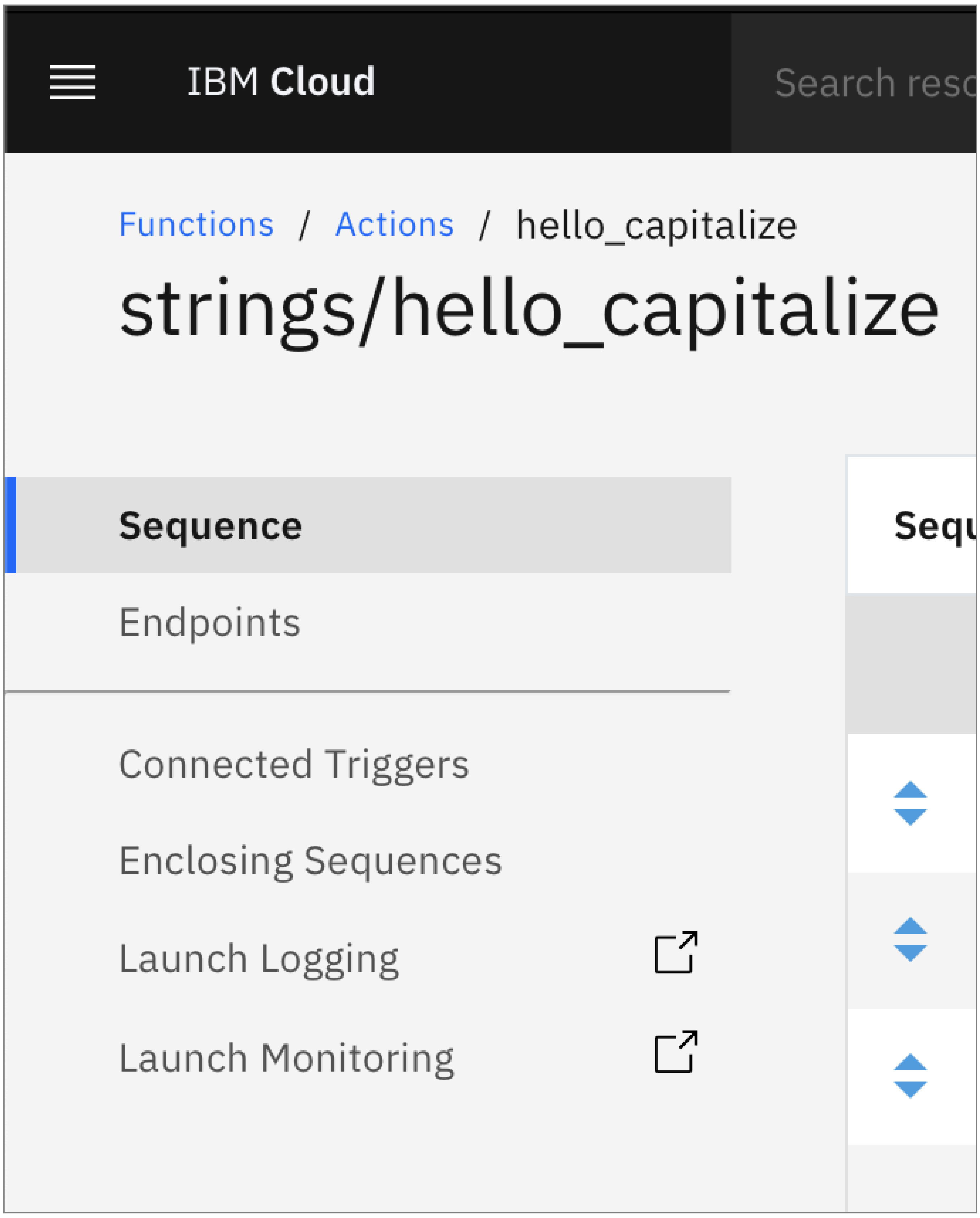
13. Click **Apply**.
14. Click **Invoke**.
15. Check the activation record and ensure that the results include this message: "TESTING MY SEQUENCE."

Great job! You passed the provided input to your sequence. This input was echoed, and that output was passed to the capitalize action, which capitalized the whole thing.

Exercise 4 : Enable Web Actions

In this exercise you will enable a web action so that your action can be called using HTTP.

1. From the **string/hello_capitalize** sequence page, click **Endpoints** in the navigation menu.



IBM Cloud

Search reso

Functions / Actions / hello_capitalize

strings/hello_capitalize

Sequence

Endpoints

Connected Triggers

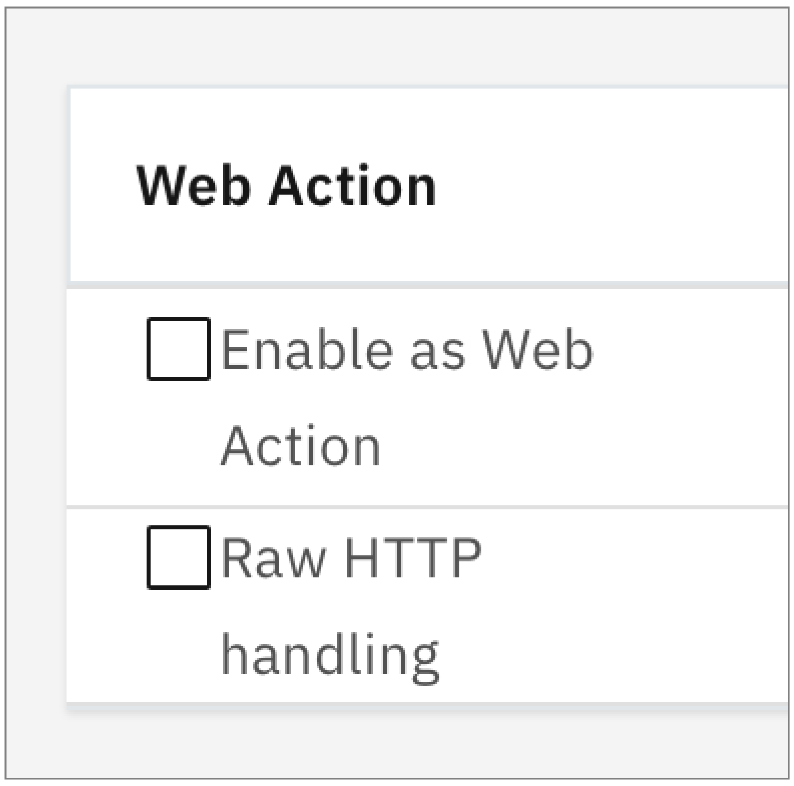
Enclosing Sequences

Launch Logging

Launch Monitoring

Sequ

2. Click the checkbox for **Enable as Web Action**.



Web Action

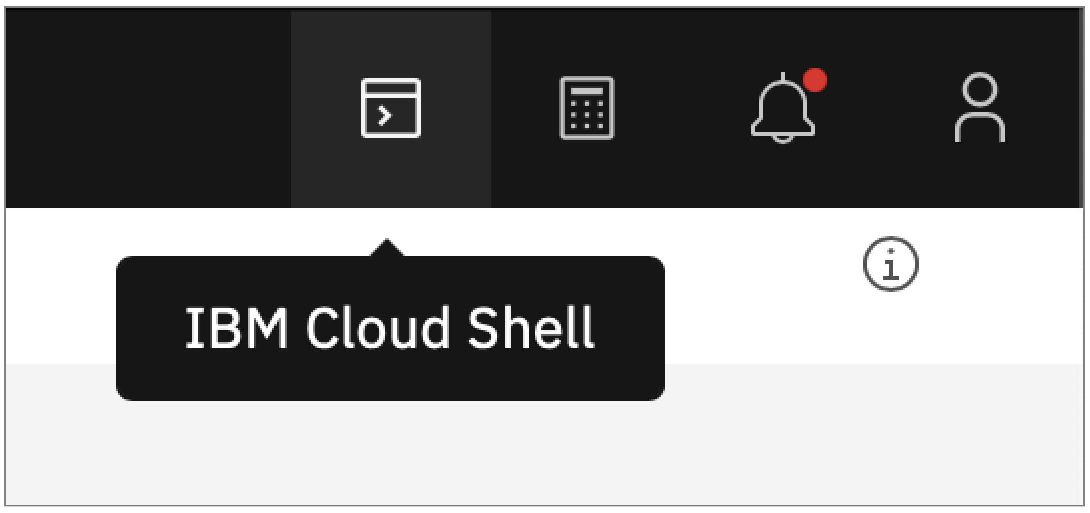
□ Enable as Web Action

□ Raw HTTP handling

3. Click **Save**.

4. Copy the `curl` command located on this page.

5. Open the IBM Cloud Shell at the top right of the cloud console.



6. Run `ibmcloud iam oauth-tokens` to list your IAM tokens for authentication.

7. Copy the bearer token. You can ignore the word "bearer" and just copy the token.

8. Plug the token into the `curl` command that you copied.

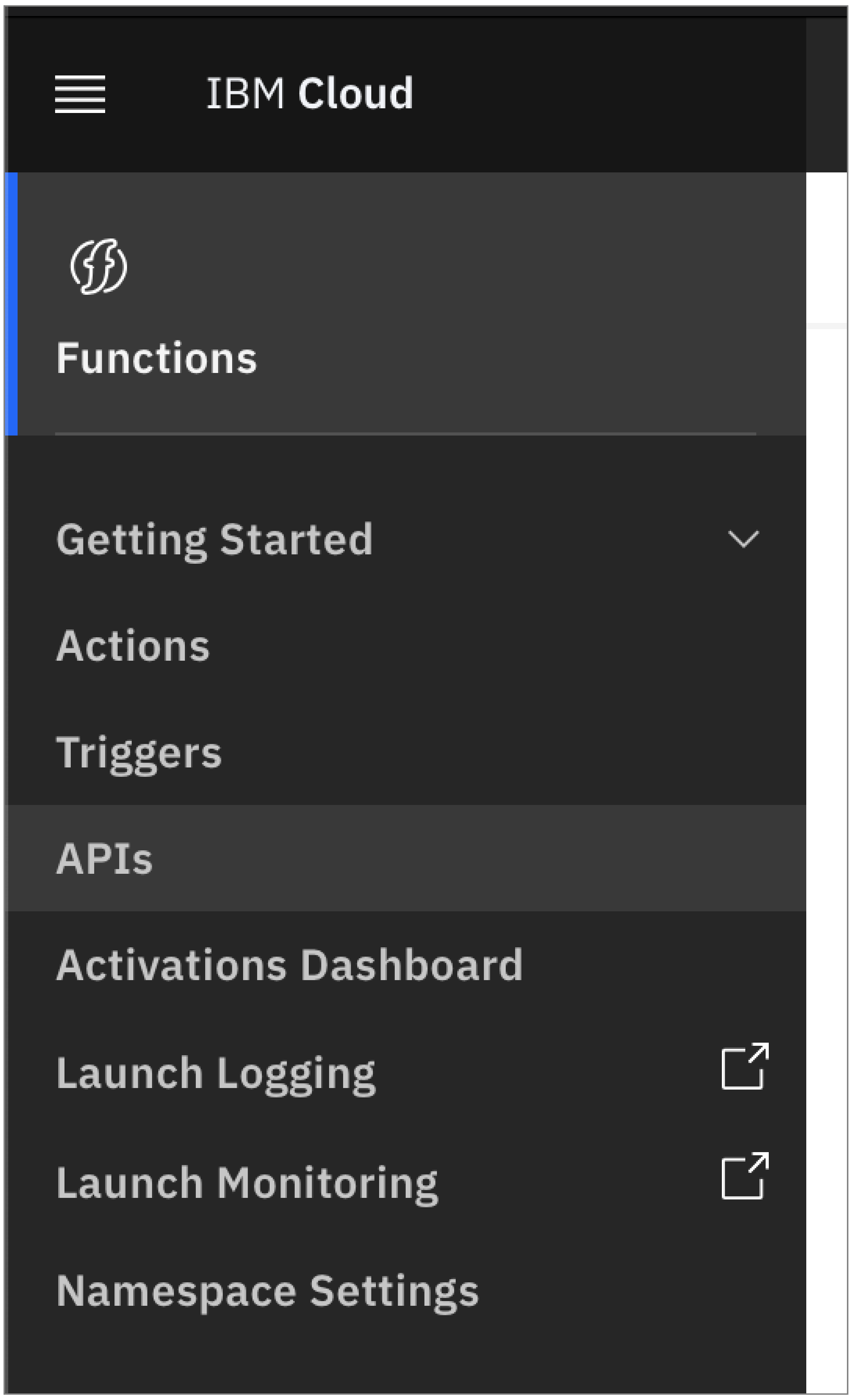
9. In the Cloud Shell, run the `curl` command with your bearer token plugged in.

Exercise 5 : API Management

In this exercise you will create an API to better manage and invoke your actions.

1. Return to the main IBM Cloud Functions page.

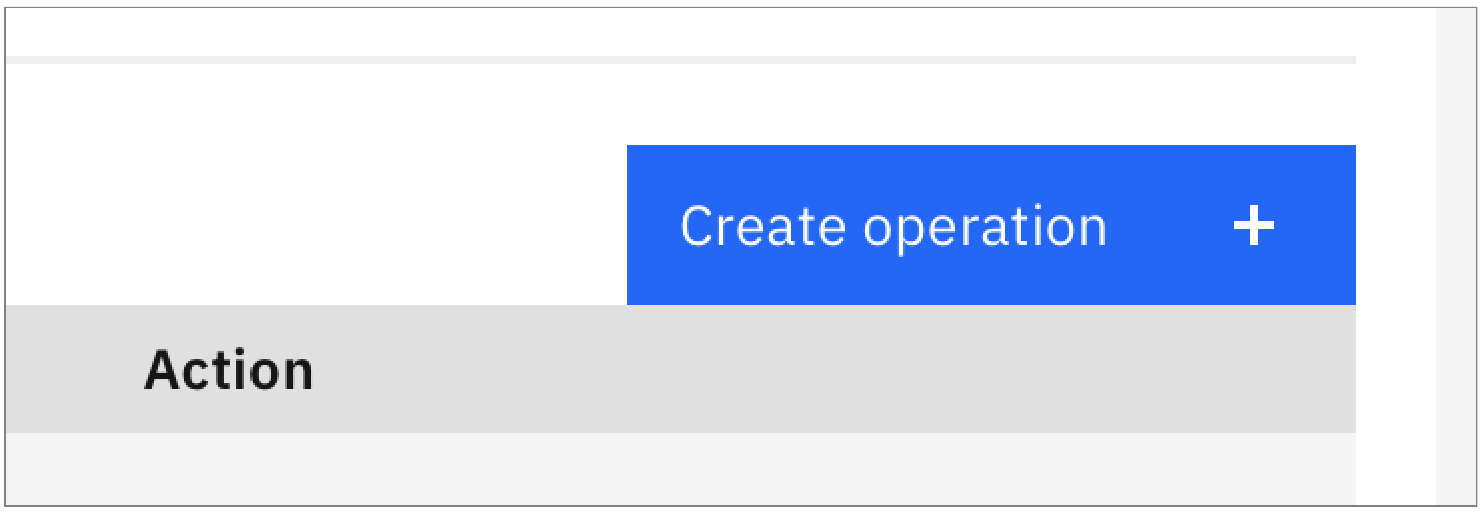
2. Click **APIs**.



3. Click **Create API**.

4. For API name, put **test**. This will also populate the base path.

5. Click **Create operation**.



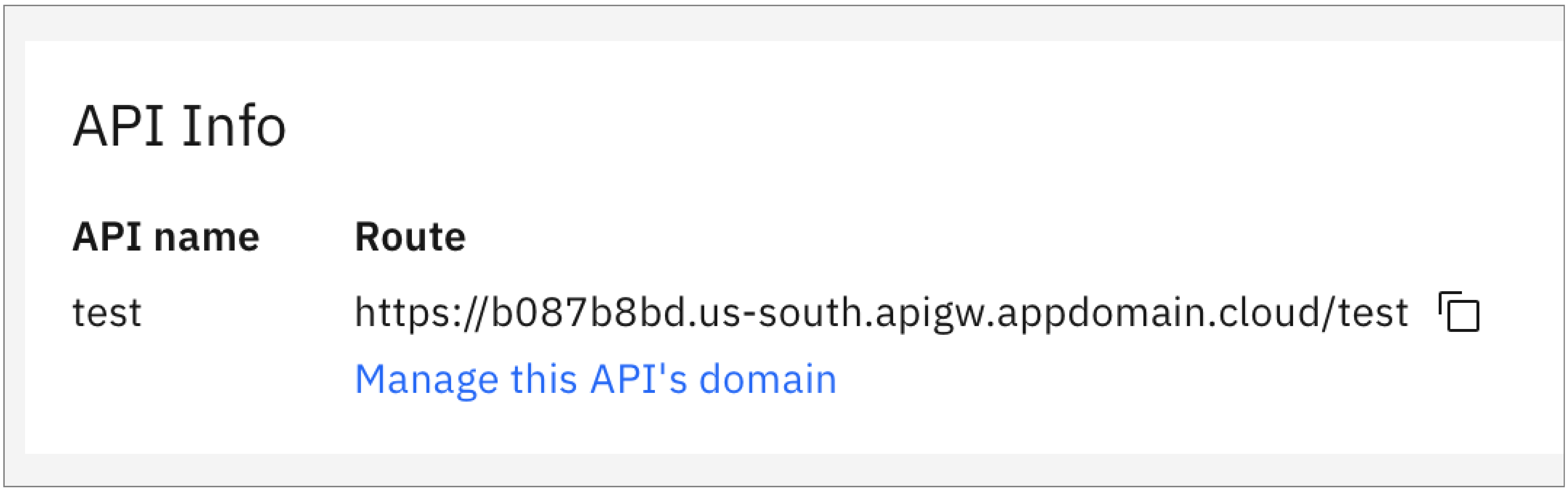
6. Put **/hello_capitalize** as the path.

7. Use **GET** for the verb, **strings** for the package**, and choose **hello_capitalize** for the action.

8. Click **Create**.

9. Scroll to the bottom and click **Create**.

10. Copy the API route. This is the domain and the base path.


API Info

API name	Route
test	https://b087b8bd.us-south.apigw.appdomain.cloud/test Manage this API's domain

11. Paste the route into your browser and add `/hello_capitalize` to it. This should look like `https://<domain>/test/hello_capitalize`.

You should see your capitalized hello world message in your browser!

Conclusion

In this lab you created your first actions and sequence! You created two actions that can be independently invoked, and you also joined those actions into a sequence that you invoked. You then enabled a web action and invoked the action using HTTP in a terminal. Finally, you created an API to better manage and invoke your actions.

Author(s)

Alex Parker