



# Exercise - Set up a test API

10 minutes

Sandbox activated! Time remaining: **3 hr 11 min**

You have used 4 of 10 sandboxes for today. More sandboxes will be available tomorrow.

In Logic Apps, you can create custom connectors to call Web APIs. Those APIs must be published and must implement an OpenAPI definition.

Your company wants to build a workflow using Logic Apps that accesses the company's PrintFramer API. The API is hosted in Azure as a REST API.

In this unit, we'll deploy our test API to the Azure sandbox activated for this module.

## Host a test API in the Azure sandbox

To save time, let's start by running a script to host our test API in Azure. The script performs the following steps:

- Create an Azure App Service plan in the free tier
- Create a Web API in Azure App Service, configured for Git deployment from a local repo
- Set account-level deployment credentials for our app
- Configure Git locally
- Deploy our Web API to our App Service instance

1. Run the following `git clone` command in the Cloud Shell to clone the repo that contains the source for our app, as well as our setup script from GitHub.

Bash	Copy
<pre>git clone https://github.com/MicrosoftDocs/mslearn-logic-apps-and-custom-connectors.git</pre>	

This command clones the repo to the local storage in the sandbox.

2. Navigate into the repo folder locally by running the following `cd` command.


Bash	Copy
------	------

```
cd mslearn-logic-apps-and-custom-connectors
```

If you list the contents of the **ms-learn-print-framer-api** folder with the `ls` command, you'll see that the folder contains a file called **setup-exercise.sh** and a folder called **PrintFramerAPI**.

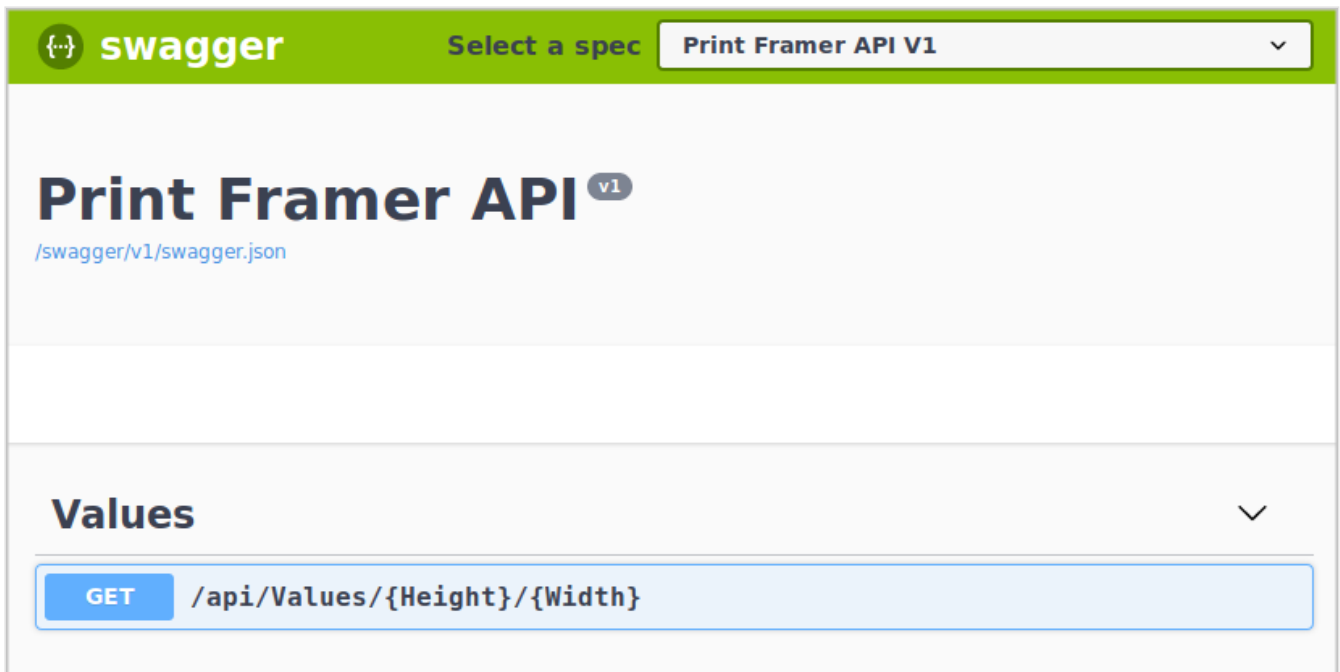
As its name suggests, `setup-exercise.sh` is the script we'll run to create our lab environment. **PrintFramerAPI** contains the app we want to deploy to Azure as a web app with a public API.

3. Run the following command in the Cloud Shell to execute the setup script.

Bash	 Copy
<pre>bash setup-exercise.sh</pre>	

The script takes a couple of minutes to run. When the script finishes, it will display two URLs that you can use to test the app deployment. Observe that during deployment, all dependencies needed for our app to run are automatically installed on the remote App Service at this point.

4. To test that our app deployed successfully, copy and paste the value for **Example URL** from the Cloud Shell output into your favorite browser. The browser displays the value `20.0`, which is the dollar value the API calculated for a frame with dimensions 6x7 inches. You can try other values by replacing `6` and `7` in the URL.
5. Keep note of this URL, because you'll need it in the next exercise.
6. To see what the API looks like, navigate in your browser to the URL that is displayed in the **Swagger URL** output shown in the Cloud Shell. The URL should look something like `https://PrintFramerAPIabcde.azurewebsites.net/swagger`. The browser displays the Swagger UI for our app, as shown in the following screenshot.



1. In the Swagger UI in your browser, select **Get** and then select **Try it out**. Enter a height and width, and then select **Execute**. Examine the response.
2. In the **Address** bar of your browser, replace **/swagger/index.html** with **/swagger/v1/swagger.json**. The browser displays the OpenAPI definition for the API.
3. Save the OpenAPI JSON content to a local file so we can use it in the next exercise.

In the next exercise, we'll create a custom connector from this OpenAPI JSON definition of our API and show how to use that connector in a Logic App. Remember to keep the OpenAPI JSON file you created as well as the URL for your app.

---

## Next unit: Access an API with a Logic App custom connector

[Continue >](#)

---

 English (United States)    Theme

[Previous Version Docs](#)   [Blog](#)   [Contribute](#)   [Privacy & Cookies](#)   [Terms of Use](#)   [Trademarks](#)

© Microsoft 2020

↻ Azure Cloud Shell

```
    }  
    return "not valid";  
  }  
}  
}  
rajani_net@Azure:~/mslearn-logic-apps-and-cu  
1 git clone https://github.com/Microsoft  
connectors.git
```