

[Serverless Computing](https://cloud.google.com/products/serverless/) (https://cloud.google.com/products/serverless/)

[Cloud Functions](https://cloud.google.com/functions/) (https://cloud.google.com/functions/)

[Documentation](https://cloud.google.com/functions/docs/) (https://cloud.google.com/functions/docs/) [Guides](#)

# Your First Function: Node.js

This guide takes you through the process of writing a Cloud Function using the Node.js runtime. There are two types of Cloud Functions:

- An HTTP function, which you invoke from standard HTTP requests.
- A background function, which you use to handle events from your Cloud infrastructure, such as messages on a Cloud Pub/Sub topic, or changes in a Cloud Storage bucket.

The sample shows how to create a simple HTTP function.

**Learn more:** For more details, read about [HTTP functions](https://cloud.google.com/functions/docs/writing/#http_functions) (https://cloud.google.com/functions/docs/writing/#http\_functions) and [background functions](https://cloud.google.com/functions/docs/writing/#background_functions) (https://cloud.google.com/functions/docs/writing/#background\_functions).

## Guide structure

1. [Creating a GCP project using Cloud SDK](https://cloud.google.com/functions/docs/first-nodejs#creating_a_gcp_project_using_cloud_sdk)  
(https://cloud.google.com/functions/docs/first-nodejs#creating\_a\_gcp\_project\_using\_cloud\_sdk)
2. [Creating a function](https://cloud.google.com/functions/docs/first-nodejs#creating_a_function) (https://cloud.google.com/functions/docs/first-nodejs#creating\_a\_function)
3. [Specifying dependencies](https://cloud.google.com/functions/docs/first-nodejs#specifying_dependencies)  
(https://cloud.google.com/functions/docs/first-nodejs#specifying\_dependencies)
4. [Deploying your function](https://cloud.google.com/functions/docs/first-nodejs#deploying_the_function)  
(https://cloud.google.com/functions/docs/first-nodejs#deploying\_the\_function)
5. [Testing your function](https://cloud.google.com/functions/docs/first-nodejs#testing_the_function)  
(https://cloud.google.com/functions/docs/first-nodejs#testing\_the\_function)

## Creating a GCP project using Cloud SDK

1. [Sign in](https://accounts.google.com/Login) (https://accounts.google.com/Login) to your Google Account.

If you don't already have one, [sign up for a new account](https://accounts.google.com/SignUp)

(<https://accounts.google.com/SignUp>).

2. In the Cloud Console, on the project selector page, select or create a Cloud project.

★ **Note:** If you don't plan to keep the resources that you create in this procedure, create a project instead of selecting an existing project. After you finish these steps, you can delete the project, removing all resources associated with the project.

**GO TO THE PROJECT SELECTOR PAGE** ([HTTPS://CONSOLE.CLOUD.GOOGLE.COM/PROJECTSELECT](https://console.cloud.google.com/projectselect))

3. Make sure that billing is enabled for your Google Cloud project. [Learn how to confirm billing is enabled for your project](https://cloud.google.com/billing/docs/how-to/modify-project) (<https://cloud.google.com/billing/docs/how-to/modify-project>).

4. Enable the Cloud Functions API.

**ENABLE THE API** ([HTTPS://CONSOLE.CLOUD.GOOGLE.COM/FLOWS/ENABLEAPI?APIID=CLOUDFUN](https://console.cloud.google.com/flows/enableapi?apiid=CLOUDFUN))

5. [Install and initialize the Cloud SDK](https://cloud.google.com/sdk/docs/) (<https://cloud.google.com/sdk/docs/>).

6. Update and install `gcloud` components:

```
gcloud components update
```



★ Need a command prompt? You can use the [Google Cloud Shell](https://console.cloud.google.com/?cloudshell=true) (<https://console.cloud.google.com/?cloudshell=true>). The Google Cloud Shell is a command line environment that already includes the Google Cloud SDK, so you don't need to install it. The Google Cloud SDK also comes preinstalled on Google Compute Engine Virtual Machines.

7. Prepare your development environment.

**GO TO THE NODE.JS SETUP GUIDE** ([HTTPS://CLOUD.GOOGLE.COM/NODEJS/DOCS/SETUP](https://cloud.google.com/nodejs/docs/setup))

## Creating a function

1. Create a directory on your local system for the function code:

LINUX OR MAC OS X

WINDOWS

```
mkdir ~/gcf_hello_world
cd ~/gcf_hello_world
```

2. Create an `index.js` file in the `gcf_hello_world` directory with the following contents:

```
functions/helloworld/index.js
(https://github.com/GoogleCloudPlatform/nodejs-docs-
samples/blob/master/functions/helloworld/index.js)
```

GOOGLECLOUDPLATFORM/NODEJS-DOCS-SAMPLES/BLOB/MASTER/FUNCTIONS/HELLOWORLD/INDEX.JS)

[FEEDBACK \(#\)](#)

```
const escapeHtml = require('escape-html');

/**
 * HTTP Cloud Function.
 *
 * @param {Object} req Cloud Function request context.
 *                More info: https://expressjs.com/en/api.html#req
 * @param {Object} res Cloud Function response context.
 *                More info: https://expressjs.com/en/api.html#res
 */
exports.helloHttp = (req, res) => {
  res.send(`Hello ${escapeHtml(req.query.name || req.body.name || 'World')}!`);
};
```

This example function takes a name supplied in the HTTP request and returns a greeting, or "Hello World!" when no name is supplied.

## Specifying dependencies

Dependencies in Node.js are managed with [npm](https://docs.npmjs.com/) and expressed in a metadata file called [package.json](https://docs.npmjs.com/files/package.json).

1. Create a `package.json` file in the `gcf_hello_world` directory, either manually or by running the command:

```
npm init
```

2. Add the function's dependency, in this case the `escape-html` package, to your `package.json` file, either manually or by running the command:

```
npm install escape-html
```

Your `package.json` file should contain a section like the following:

```
{  
  ...  
  "dependencies": {  
    "escape-html": "^1.0.3"  
  }  
  ...  
}
```

**Learn more:** For more details, read about [specifying dependencies](https://cloud.google.com/functions/docs/writing/specifying-dependencies-nodejs) (<https://cloud.google.com/functions/docs/writing/specifying-dependencies-nodejs>).

## Deploying the function

To deploy the function with an HTTP trigger, run the following command in the `gcf_hello_world` directory:

```
gcloud functions deploy helloHttp --runtime nodejs8 --trigger-http
```

**Learn more:** For more details, read about [deploying Cloud Functions](https://cloud.google.com/functions/docs/deploying/) (<https://cloud.google.com/functions/docs/deploying/>).

## Testing the function

1. When the function finishes deploying, take note of the `httpsTrigger.url` property or find it using the following command:

```
gcloud functions describe helloHttp
```

It should look like this:

```
https://GCP_REGION-PROJECT_ID.cloudfunctions.net/helloHttp
```

2. Visit this URL in your browser. You should see a "Hello World!" message.

Try passing a name in the HTTP request, for example by using the following URL:

```
https://GCP_REGION-PROJECT_ID.cloudfunctions.net/helloHttp?name=NAME
```

You should see the message "Hello *NAME*!"

## Viewing logs

### Using the command-line tool

Logs for Cloud Functions are viewable in the Stackdriver Logging UI, and via the `gcloud` command-line tool.

To view logs for your function with the `gcloud` tool, use the `logs read` (<https://cloud.google.com/sdk/gcloud/reference/functions/logs/read>) command, followed by the name of the function:

```
gcloud functions logs read helloHttp
```

The output should resemble the following:

LEVEL	NAME	EXECUTION_ID	TIME_UTC	LOG
D	helloHttp	rvb9j0axfclb	2019-09-18 22:06:25.983	Function execution started
D	helloHttp	rvb9j0axfclb	2019-09-18 22:06:26.001	Function execution took 19

**Note:** There is typically a slight delay between when log entries are created and when they show up in Stackdriver.

### Using the Logging dashboard

You can also view logs for Cloud Functions from the [Cloud Console](#) ([https://console.cloud.google.com/project/\\_/logs?service=cloudfunctions.googleapis.com](https://console.cloud.google.com/project/_/logs?service=cloudfunctions.googleapis.com)).

**Learn more:** For more details, read about [writing, viewing, and responding to logs](#) (<https://cloud.google.com/functions/docs/monitoring/logging>).

---

*Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](#) (<https://creativecommons.org/licenses/by/4.0/>), and code samples are licensed under the [Apache 2.0 License](#) (<https://www.apache.org/licenses/LICENSE-2.0>). For details, see our [Site Policies](#) (<https://developers.google.com/terms/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.*

*Last updated November 15, 2019.*