**GCP – HOL -Session 4**

# **Writing Your Web Service with Node.js**

## Before you begin

* [Install Node.js LTS](https://nodejs.org/en/download/).

## Key points

* You can use dependencies by listing them in your [package.json](https://docs.npmjs.com/files/package.json) file. See [Specifying Dependencies](https://cloud.google.com/appengine/docs/standard/nodejs/specifying-dependencies) for more information.
* App Engine starts your application by running npm start.
* Your server must listen to the port specified by the process.env.PORT environment variable.
* You need an [app.yaml](https://cloud.google.com/appengine/docs/standard/nodejs/configuring-your-app-with-app-yaml) file to deploy your service to App Engine.

## Creating a server to listen for HTTP requests

The core of your web service is the HTTP server. The sample code in this HOL session uses the [Express.js](https://expressjs.com/) framework to handle HTTP requests, but you are free to use a web framework of your choice.

1. Create a new folder called my-nodejs-service for your Node.js service.
2. Navigate to the folder in your terminal, and create a package.json file by running npm init.
3. Add Express as a dependency by running:

npm install express

Confirm that Express appears in your package.json file's dependencies field. Here's an example:

{  
  ...  
  "dependencies": {  
    "express": "^4.16.3"  
  }  
  ...  
}

1. Add a start script to your package.json file:

"scripts": {  
  "start": "node server.js"  
}

1. Create a file called server.js in the same folder and add the following code:

[appengine/building-an-app/build/server.js](https://github.com/GoogleCloudPlatform/nodejs-docs-samples/blob/3bb14ef7c23305613bbfe04f03d3b83f6a120e1a/appengine/building-an-app/build/server.js)

[VIEW ON GITHUB](https://github.com/GoogleCloudPlatform/nodejs-docs-samples/blob/3bb14ef7c23305613bbfe04f03d3b83f6a120e1a/appengine/building-an-app/build/server.js)

const express = require('express');  
const app = express();  
  
app.get('/', (req, res) => {  
  res.send('Hello from App Engine!');  
});  
  
// Listen to the App Engine-specified port, or 8080 otherwise  
const PORT = process.env.PORT || 8080;  
app.listen(PORT, () => {  
  console.log(`Server listening on port ${PORT}...`);  
});

This is a very basic web server - it responds to all GET requests to the root path ('/') with the text "Hello from App Engine!" Note the last four lines, where the server is set to listen to the port specified by process.env.PORT, an [environment variable](https://cloud.google.com/appengine/docs/standard/nodejs/runtime#environment_variables) set by the App Engine runtime. If your server isn't set to listen to this port, it will not receive requests.

Notice that if process.env.PORT is not set, port 8080 is used as a default. This is necessary for testing your app locally, because process.env.PORT doesn't get set during local runs - it is only set when your app is running on App Engine. You can use whichever port you prefer for testing, but this guide uses 8080.

## Running the server locally

To run the server locally:

1. Run npm start in your terminal. This will run your server.js file.
2. Point your web browser to [http://localhost:8080](http://localhost:8080/).

You should see a page with the text "Hello from App Engine!"

## Creating the app.yaml file

An app.yaml file specifies settings for your App Engine service's runtime environment. Your service will not deploy without this file.

1. In your my-nodejs-service folder, create a file called app.yaml.
2. Add the following contents:

[appengine/building-an-app/build/app.yaml](https://github.com/GoogleCloudPlatform/nodejs-docs-samples/blob/master/appengine/building-an-app/build/app.yaml)

[VIEW ON GITHUB](https://github.com/GoogleCloudPlatform/nodejs-docs-samples/blob/master/appengine/building-an-app/build/app.yaml)

runtime: nodejs10

This is a minimal configuration file, indicating to App Engine the version of the Node.js runtime. The app.yaml file can also specify network settings, scaling settings, and more. For more information, see the[app.yaml reference](https://cloud.google.com/appengine/docs/standard/nodejs/config/appref).

At this point, you should have a file structure like the following:

my-nodejs-service/  
  app.yaml  
  package.json  
  server.js

# **Deploying Your Web Service**

This section of the HOL session shows how to use the gcloud tool to deploy your Node.js web service on App Engine.

## Before you begin

1. [Create a GCP project with an App Engine app](https://cloud.google.com/appengine/docs/standard/nodejs/building-app/creating-project).
2. [Write a Node.js web server ready to deploy on App Engine](https://cloud.google.com/appengine/docs/standard/nodejs/building-app/writing-web-service).
3. [Install Cloud SDK](https://cloud.google.com/sdk/docs/), which provides the gcloud command-line tool. Ensure gcloud is configured to use the Google Cloud Platform project you want to deploy to.

## Key points

* Use gcloud app deploy and gcloud app browse to deploy and view your service.

## Deploying your service

In your my-nodejs-service folder, where the app.yaml file is located, run the following command in your terminal:

gcloud app deploy

Your project files are then packaged and uploaded to Google Cloud Storage.

**Note:** Files listed in your [**.gcloudignore**](https://cloud.google.com/sdk/gcloud/reference/topic/gcloudignore) file will not be uploaded. If you don't have one, a **.gcloudignore** file is created to ensure that your **node\_modules** folder is not uploaded.Node.js modules are installed in the cloud as listed in your package.json and package-lock.json files and your service is started by using npm start.

## Viewing your service

To quickly launch your browser and access your web service at https://**YOUR\_PROJECT\_ID**.appspot.com, use the following command:

gcloud app browse

## Services and versions

You've just created and deployed a service on App Engine. You can specify the name of your service in the app.yaml file. If the name is omitted, it is treated as default. The first service you deploy must be the default service.

You can update your service at any time by running the gcloud app deploy command again. Each time you deploy, a new version is created and traffic is automatically routed to the latest version.

To confirm that your service has been created and a version has been deployed:

1. View your App Engine services in the GCP Console:

[VIEW SERVICES](https://console.cloud.google.com/appengine/services)

You should see one service listed, named default. The default service is publicly accessible at the following URL:

https://**YOUR\_PROJECT\_ID**.appspot.com

1. View your versions:

[VIEW VERSIONS](https://console.cloud.google.com/appengine/versions)

You should see one timestamped version listed, corresponding to your deployment.