

Netlify Serverless Functions Cheat Sheet

By @jamesqquick - jamesqquick.com

Create Your First Serverless Function

Create a JavaScript file inside of a *functions* directory that exports a `handler` function.



The name of your Serverless Function file will be the name of the serverless function itself.

Serverless Functions take 3 parameters

Event	Context	Callback
request information (body, httpMethod, etc.)	information about invocation, function, and environment	function to call in non-async handlers to send a response

Handlers can be either **synchronous** or **asynchronous**...

Synchronous handlers must call the `callback` function with an error as the first parameter (if applicable), and the second an object that includes HTTP `statusCode` and `body`.

```
exports.handler = (event, context, callback) => {
  return callback(null{
    statusCode: 200,
    body: JSON.stringify({msg: '👍'})
  })
}
```

Asynchronous handlers must be tagged as `async` and **do not need** to call the callback function. It's fine to remove the `callback` parameter if you aren't using it.



You'll most likely run asynchronous JavaScript, so I always use **Asynchronous Handlers**.

```
exports.handler = async(event, context) => {
  return {
    statusCode: 200,
    body: JSON.stringify({msg: '👍'})
  }
}
```

If you need to access the request JSON body, you have to parse it with

`JSON.parse()`.

```
const body = JSON.parse(event.body)
```

Running Functions Locally

Install the Netlify CLI...

```
npm install -g netlify-cli
```

and create a Netlify.toml configuration file to specify where directory your serverless functions are in.

```
[build]
  functions = "functions"
```

Run `netlify dev` and your functions will be available at **localhost:8888/.netlify/functions/<function_name>**.

Simplify this path by adding a redirects configuration in your `Netlify.toml` file.

```
[[redirects]]
  from = "api/*"
  to = ".netlify/functions/:splat"
  status = 200
```

Extra Tips



Use Environment Variables ([Full Video](#))

Often times, you'll have secret credentials (API keys, API secrets, etc.) that **you don't want to check into your source code**. For these private credentials, use **environment variables**. When running locally, install the `dotenv` package with `npm install dotenv`.

Create a `.env` in your repo with your credentials.

```
SECRET_KEY=<Some secret key>
```

but make sure you ignore this file in your `.gitignore` file

```
.env
```

Lastly, require the `dotenv` packages and grab the variables you need.

```
require('dotenv').config()  
const secretKey = process.env.SECRET_KEY
```

Link Your Local Site to Netlify

After you've created your site in Netlify, you can link the local repository to it from the command line with `netlify link`. Netlify will then **inject the environment variables** from the dashboard directly into your locally running functions. This way you don't need to manage environment variables locally.

🔥 Focus on Code Reusability

You will quickly find yourself repeating code through different Serverless Functions. Create a `utils` directory and include any code that might be useful to share.

🔥 Simplify Return Data

When returning body data, I hate having to type `JSON.stringify()` each time. Use this helper function instead.

```
const formattedReturn = (statusCode, body) => {  
  return {  
    statusCode,  
    body: JSON.stringify(body);  
  }  
}
```

Then use it in your function.

```
exports.handler = async(event, context) => {  
  return formattedReturn(200, {msg: '👍'});  
}
```