NodeJS

HyperionDev

Lecture – Housekeeping

- ❑ The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly.
  - ❑ Please review Code of Conduct (in Student Undertaking Agreement) if unsure
- ❑ No question is daft or silly - **ask them!**
- ❑ Q&A session at the end of the lesson, should you wish to ask any follow-up questions.
- ❑ Should you have any questions after the lecture, please schedule a mentor session.
- ❑ For all non-academic questions, please submit a query: www.hyperiondev.com/support

Lecture Objectives

1. Discover the benefits of Node.js
2. Learn to harness pre-built modules
3. Learn to create custom modules

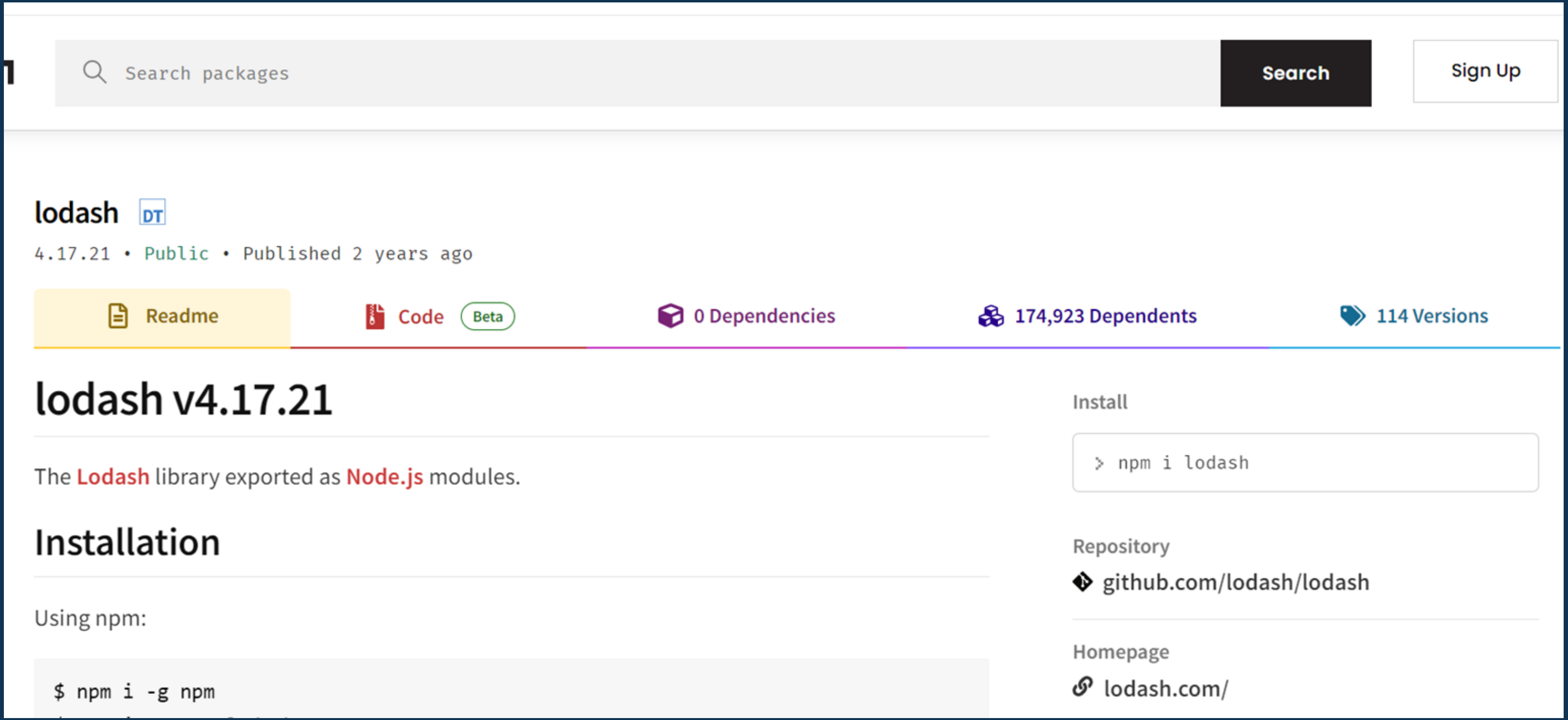
NodeJS

- Node.js is a JavaScript runtime environment that works outside a web browser.
- Enables server side scripting and provides an event-driven, non-blocking, I/O model making it a popular tool for any project.
- Built on top of Chrome V8 engine.
- Open source.
- A Node.js app runs in a single process, without creating a new thread for every request.
- While performing an I/O operation, Node.js does not block the thread waiting for the I/O operation to complete. It continues the operation right after a response comes from it.

Managing Node Packages

- ❑ NPM, Node Package Manager is used to manage, install and remove modules and libraries from your project.
- ❑ Installed along with Node.js.
- ❑ Module: A module is an encapsulated set of code with a specific function.
- ❑ To create your own module: Create a blank directory → Run ‘npm init’ → This will create a file ‘package.json’.
- ❑ Package.json contains metadata about your package.
- ❑ To install a package: Run ‘npm install <name-of-package>’ → As an aftereffect this command will generate a folder ‘node_modules’.
- ❑ Node.js stores all the installed libraries of the project to the ‘node_modules’ folder by default.
- ❑ An entry is also added to the ‘dependencies’ section of package.json.
- ❑ www.npmjs.com is an online repository of npm libraries.

Managing Node Packages



lodash DT

4.17.21 • Public • Published 2 years ago

lodash v4.17.21

The Lodash library exported as Node.js modules.

Installation

Using npm:

```
$ npm i -g npm
```

Install

```
> npm i lodash
```

Repository

github.com/lodash/lodash

Homepage

lodash.com/

NodeJS Example

```
const http = require('http');
const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

References

- ❏ https://nodejs.dev/en/learn/#an-example-nodejs-application
- ❏ https://nodejs.org/en/docs/guides/blocking-vs-non-blocking
- ❏ https://nodejs.org/en/docs/guides/event-loop-timers-and-nexttick
- ❏ https://nodejs.org/en/docs/guides

Questions and Answers

Thank You!