Server Development

Master in Electrical and Electronic Engineering

luis.marcelino@ipleiria.pt
2020/11/17

POLITÉCNICO DE LEIRIA | ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

Overview

- NPM
- Environment
- SDKs

NPM

NPM

- package.json : project manifest in the JSON format
  - Name, description and author
  - Dependencies
  - Information about its unique source control
  - Tasks configuration

init

- npm command to create a package.json file
- Options can be inserted by answering questions:
    - The project's name,
    - The project's initial version,
    - The project's description,
    - The project's entry point (meaning the project's main file),
    - The project's test command (to trigger testing with something like Standard)
    - The project's git repository (where the project source can be found)
    - The project's keywords (basically, tags related to the project)
    - The project's license

```
npm init
```

Dependencies

- Objects where key is the name of the package and value is the acceptable version to install
- dependencies:
    - Packages required by your application in production
- devDependencies:
    - Packages that are only needed for local development and testing
- NPM will download dependencies and devDependencies listed in package.json that meet the semantic version requirements listed

```
npm install <package-name> [--save-prod]
npm install <package-name> --save-dev
```

install

- The npm install without arguments install the dependencies in the local node_modules folder
- In global mode (ie, with -g or --global appended to the command), installs as a global package
- With the --production flag (or when the NODE_ENV environment variable is set to production), npm will not install modules listed in devDependencies
- Install supports tarball files or URLs
    - npm install <git remote url>

```
npm install git+ssh://git@github.com:npm/cli.git#v1.0.27
```

Running tasks

- The file package.json can define named scripts
- To execute them use the run-script command

package.json

```
"scripts": {
 ...
    "lint": "eslint src/js"
 ...
}
```

```
npm run lint
```

environment

Development Environment

- Development variables: external to our (node.js) application
    - reside in the operating system or in the container of the application that is running.
- Variable can be set in the shell
    - export NODE_ENV="development"
- Accessed in node.js using the process object
    - process.env.NODE_ENV

.env

- Dotenv is a module that loads environment variables from a .env file into process.env.
- Storing configuration in the environment separate from code
- Install and require the dotenv module (as early as possible)
  - require('dotenv').config()
- File should NOT be added to repository

nodemon

- Restart the node server on file changes
- Install globally to use the nodemon command
  - It may require administrative privileges
- Starts the “main” entry point
- Networked environments may need to use the “legacyWatch”
  - --legacy-watch or -L
- Start script can be placed in “package.json”
  - "start": "nodemon index.js -L -e ejs,js,css,html,jpg,png,scss"

```
npm install nodemon -g
```

SDK

Software Development Kits

- Node.js is able to run most Javascript modules
  - Abstract from HTTP requests
- Available from NPM / YARN
- Multiple functionalities:
  - Authentication
  - Storage
  - Resources management
  - ...

Firebase SDK

- Firebase JavaScript SDK
  - There is a Firebase AdminSDK
  - npm install --save firebase
  - Add the core Firebase SDK
  - Add the “Firebase products” to be used
  - Create a web “Firebase app”
  - Use the credentials in your server
  - Read or write data from/to firebase

References

https://docs.npmjs.com/specifying-dependencies-and-devdependencies-in-a-package-json-file

https://nodesource.com/blog/an-absolute-beginners-guide-to-using-npm/

https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs

https://www.digitalocean.com/community/tutorials/workflow-nodemon