**Node.js Workshop : npm Node Package manager and .gitignore file**

After completing this workshop the student is knows how to:

* What is npm
* how to install npm package globally
* using a package in your application
* how to uninstall a npm package
* how to add and populate project *.gitignore* file

**What is npm**

Official [npm Web page](https://www.npmjs.com/).

npm [documentation](https://docs.npmjs.com/)

**How to install npm package globally**

**Exercise 1. index.js and initialize npm project**

1. Create a new folder for these assignments. Save all your code in there.
2. Create a new file and name it to *index.js.* Make your new server to log *Hello world* in console. Then run your new server and make sure it works
3. Next you need to initialize your project by typing *npm init* in your terminal.

*This utility will walk you through creating a package.json file.*

*It only covers the most common items, and tries to guess sensible defaults.*

*See `npm help init` for definitive documentation on these fields*

*and exactly what they do.*

*Use `npm install <pkg>` afterwards to install a package and*

*save it as a dependency in the package.json file.*

1. Install [date-fns package](https://www.npmjs.com/package/date-fns) in your project. In your terminal run *npm i date-fns* – see the impact in *package.json* file

Edit your *index.js* file and add your newly installed npm package in it. Log current date and time in console.

const { format } = require('date-fns');

console.log(format(new Date(), 'dd-MM-yyyy\tHH:mm:ss'));

1. Install [nodemon](https://www.npmjs.com/package/nodemon) package in your project. In your terminal run *npm i nodemon -D* – see how *package.json* file content has changed. What is the role of -D parameter?
2. In your terminal run *npm run dev* and make sure your program works with no error messages.

Make some changes (e.g. add a new console.log()) to your *index.js* file and see your server is restarting after you have saved your changes.

1. Install [uuid](https://www.npmjs.com/package/uuid) package in your project. In your terminal run *npm i uuid* – see the impact in *package.json* file content

Edit your *index.js* file and add your newly installed npm package in it. Log generated uuid in console

* 1. const { format } = require('date-fns');
  2. const { v4: uuidv4 } = require('uuid');
  3. console.log(format(new Date(), 'dd-MM-yyyy\tHH:mm:ss'));
  4. console.log(uuidv4());
  5. console.log('Hello World!');

Familiarize yourself how to [uninstall npm packages and dependencies](https://docs.npmjs.com/uninstalling-packages-and-dependencies).

See *node\_modules* folder in your project directory. Adding npm packages to your project will expand your project size remarkably.

Your *package.json* holds important information about the project. It contains human-readable metadata about the project (like the project name and description) as well as functional metadata like the package version number and a list of dependencies required by the application. It's the package.json file that enables npm to start your project, run scripts, install dependencies, publish to the NPM registry, and many other useful tasks.

**How to make .gitignore for your project**

A .*gitignore* file is a plain text file that contains a list of all the specified files and folders from the project that Git should ignore and not track. Typically, a .*gitignore* file gets placed in the root directory of the repository. The root directory is also known as the parent and the current working directory. The root folder contains all the files and other folders that make up the project.

The types of files you should consider adding to a .gitignore file are any files that do not need to get committed. Folders generated by package managers, such as npm’s node\_modules folder. This is a folder used for saving and tracking the dependencies for each package you install locally.

**Exercise** - Create .*gitignore* file in your project root folder and add *node\_modules* folder in it.

Initialize a git repository and add all your project files to your repository. Make changes to your project files and make git commit to the modified files. See instructions…

<https://git-scm.com/docs/git-init>

<https://git-scm.com/docs/git-add>

<https://git-scm.com/docs/git-commit>