

Background

Developers working with NodeJS use packages in their code. A package is a functional NodeJS module that includes versioning, documentation, dependencies (in the form of other packages), and more. NodeJS has a managed packages environment called npm. npm is regularly updated with new packages and new versions of existing packages.

Snyk scans NodeJS packages to identify and assist developers in remediating vulnerabilities prior to merging their code back with its project. In order for Snyk to identify these vulnerabilities in certain packages, this is what happens:

1. The user provides the name of the package for analysis.
2. We fetch the overall set of dependencies from the relevant package manager, for the given package.
3. We compare the set of dependencies that we retrieve with our database of vulnerable packages in order to identify whether any of the dependencies are vulnerable.
4. For any package that we identify as vulnerable, we then list all available remediation paths (upgrades and/or patches for vulnerable packages) for the user.
5. The user chooses their preferred remediation actions from the list, and we apply them by creating a PR for the relevant repository.

Exercise

This exercise focuses on stages 1 and 2 above, for a package that is already published on [npmjs.com](https://www.npmjs.com).

Your task is to design and implement a web service. This service should return the full package dependency tree based on a given package name (user input), which we could then later use for stage 3.

You can obtain package data through the npm registry using the following URL format:
`https://registry.npmjs.org/<package_name>/<version_or_tag>`

For example: `https://registry.npmjs.org/express/latest` or `https://registry.npmjs.org/async/2.0.1`

Things to consider

- There are currently over 1M packages on npmjs.com, and the number is growing all the time.
- The packages update from time to time, just as their dependencies do too.
- Consider these important factors that can make or break a great web service: API, architecture, data storage, low latency, scalability, monitoring, you name it :)

Implementation

1. Create a working application that, given the name of a published npm package, returns the entire set of dependencies for the package.
2. Present the dependencies in a tree view.
3. We require tests. It's up to you what style and how exhaustive these are.
4. Account for asynchronous fetching of dependencies as you see fit.
5. Consider caching relevant data so that repeated requests resolve with minimum latency.
6. Please use Git and, preferably, GitHub for your implementation. When using GitHub, please invite the **snyk-exercise-review** user to your repository if you wish to make it private.

Good luck, and enjoy!