Background

NodeJS has a managed packages environment called npm. A package is a functional NodeJS module with versioning, documentation, dependencies (in the form of other packages), and more. npm is being constantly updated with new packages and new versions of existing packages.

In order to identify vulnerabilities in certain packages, the following (rough) process takes place:

1. User provides name of package for analysis
2. We fetch the overall set of dependencies of given package
3. We compare the set of dependencies with a managed set of vulnerable packages
4. We identify remediation paths (upgrades and/or patches for vulnerable packages)
5. Users choose their preferred remediation action and we apply it

Exercise

In this exercise we will focus on on stages 1 and 2 for a package that is already published on npmjs.com

Your task is to design and implement a web service that, given a name of an npm package, returns its dependent packages to be used by the consequent stage (3).

You can obtain package data through the npm registry:
`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

1. Look at the inner "dependencies" object for analysis of first-order dependencies.
2. There are currently over 650K packages on npmjs.com, and the number is growing all the time.
3. The packages update from time to time, just as their dependencies.
4. What makes a good web service? API, architecture, data storage, low latency, scalability, monitoring, you name it :)

Implementation

1. A working web application that, given a name of an npm published package, returns the set of dependencies for said package.
2. Present the dependencies in a tree view.
3. Account of asynchronous fetching of dependencies as you see fit.
4. Cache relevant data so that repeated requests resolve with minimum latency.

Good luck, and enjoy!