Introduction to NodeJS

Check Your Progress

- Do you know how Bootstrap Grid works? **…YES**
- Have you created your homepage with Bootstrap? **…YES**
- Have you created a basic HTTP server with NodeJS and served one of two possible files based on query strings parameters? **…YES**
- Do you know how to extract values from a URL string? **…YES**
- Have you connect NodeJS server to your database (SQL or noSQL) **…YES**
- Have you been able to send a table contents as a JSON string? **…YES**
- Have you made at least two code commits to your repositories? **…YES**

What Is NodeJS?

- Node.js is a platform built on Google Chrome's JavaScript Engine
- Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications
- NodeJS runs Javascript code outside the browser
- Node.js uses an event-driven, non-blocking I/O model

Features of NodeJS

- **Asynchronous and Event Driven**
- **Very Fast**
- **Single Threaded but Highly Scalable**
- **No Buffering**
- **MIT License**

Where to use NodeJS

- Following are the areas where Node.js is proving itself as a perfect technology partner.
- I/O bound Applications
- Data Streaming Applications
- Data Intensive Real-time Applications (DIRT)
- JSON APIs based Applications
- Single Page Applications

Node Package Manager (NPM)

- provides two main functionalities –
  - Online repositories for node.js packages/modules which are searchable on search.nodejs.org
  - Command line utility to install Node.js packages, do version management and dependency management of Node.js packages.
- By default, NPM installs any dependency in the local mode.
  - npm install express
- Locally deployed packages are accessible via require() method.
- To install globally
  - npm install express –g
- Note that you can uninstall, update and search modules using NPM

Using Package.json

- package.json is present in the root directory of any Node application/module and is used to define the properties of a package.
- Attributes of package.josn
  - **name** − name of the package
  - **version** − version of the package
  - **description** − description of the package
  - **homepage** − homepage of the package
  - **author** − author of the package
  - **contributors** − name of the contributors to the package
  - **dependencies** − list of dependencies. NPM automatically installs all the dependencies mentioned here in the node_module folder of the package.
  - **repository** − repository type and URL of the package
  - **main** − entry point of the package
  - **keywords** − keywords

Callaback & Events

- A callback function is called at the completion of a given task
- Node makes heavy use of callbacks
- Node.js is a single-threaded application, but it can support concurrency via the concept of **event** and **callbacks**
- In an event-driven application, there is generally a main loop that listens for events, and then triggers a callback function when one of those events is detected.
- callback functions are called when an asynchronous function returns its result, whereas event handling works on the observer pattern.

Blocking code Example

```
var fs = require("fs");

var data = fs.readFileSync('input.txt');

console.log(data.toString());
console.log("Program Ended");
```

Non Blocking Code Example

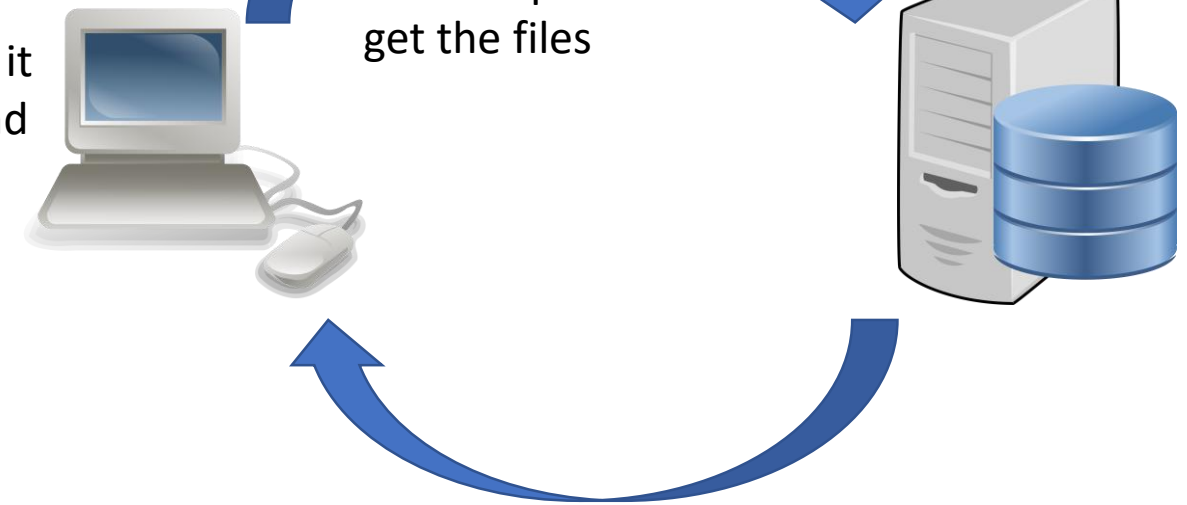
```
var fs = require("fs");

fs.readFile('input.txt', function (err, data) {
   if (err) return console.error(err);
   console.log(data.toString());
});

console.log("Program Ended");
```

- In Node Application, any async function accepts a callback as the last parameter and a callback function accepts an error as the first parameter.

```
fs.readFile('input.txt', function (err, data) {
   if (err) return console.error(err);
   console.log(data.toString());
});
```

Why CSS and images No longer works

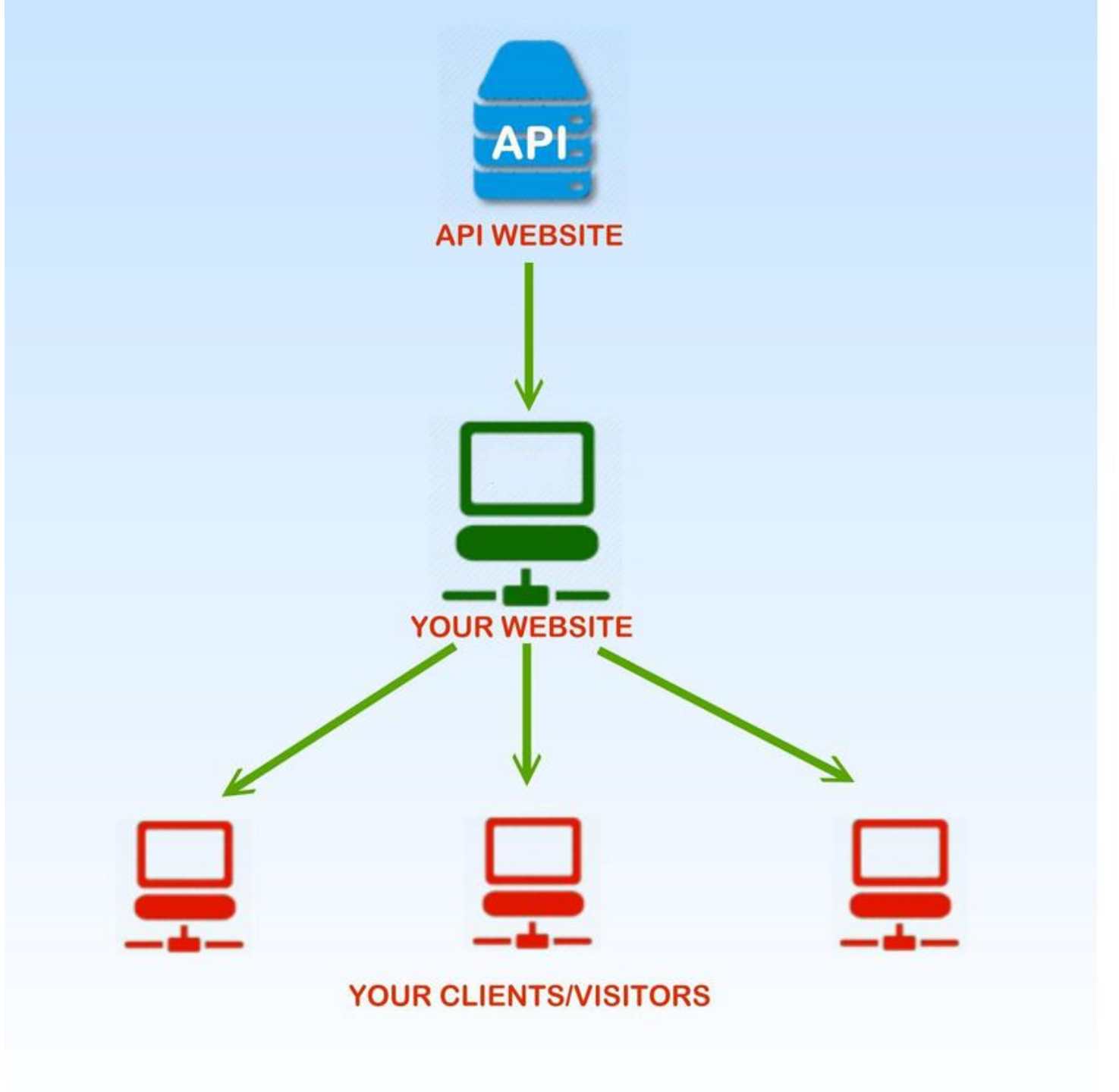
Web Server Explained



Code Demonstration

Why making a REST?

- REST is any interface between systems using HTTP to obtain data and generate operations on those data in all possible formats, such as XML and JSON.
- **Separation between the client and the server**
- **Visibility, reliability and scalability**.
- **The REST API is always independent of the type of platform or languages m**aking your server communicate with any device (mobile, desktop, tv, web…etc)
- Easily integrate your server with other applications
- Ability to expose your API to public, so data can be easily shared, allowing more opportunities for growth and revenue


API
API WEBSITE
YOUR WEBSITE
YOUR CLIENTS/VISITORS