CS454 Node.js & Angular.js

Cydney Auman
Albert Cervantes
CSULA

APIs & Intro to Node - Week 3

APIs

APIs (application programming interfaces) are a big part of web apps.

The four methods most commonly seen in Web APIs are:

GET - Asks the server to retrieve a resource

POST - Asks the server to create a new resource

PUT - Asks the server to edit/update an existing resource

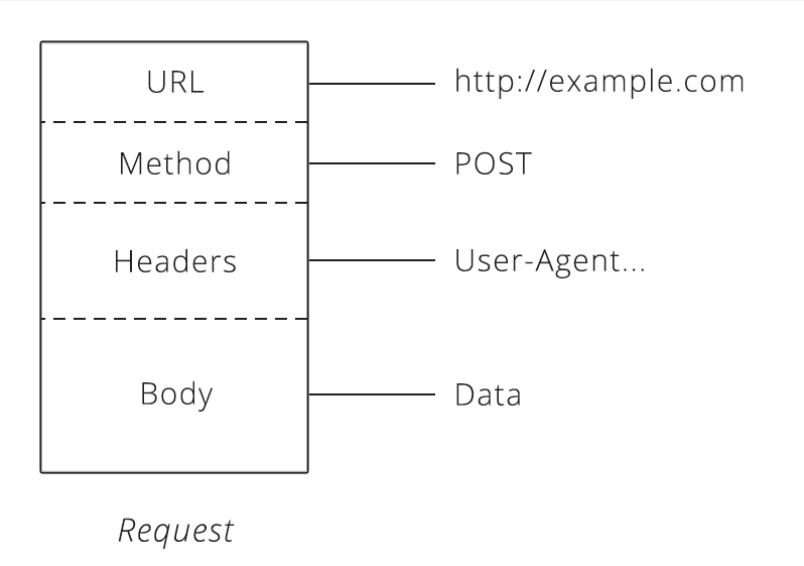
DELETE - Asks the server to delete a resource

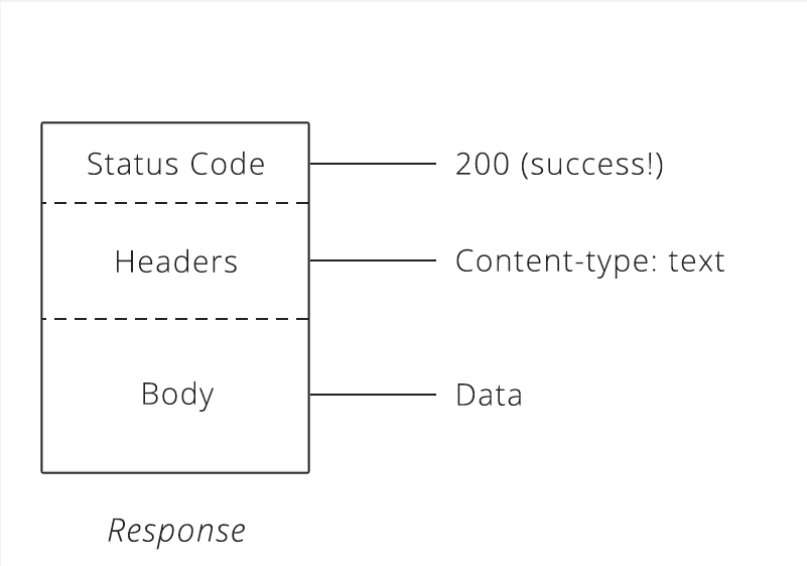
Web APIs - Req/Res

Communication in HTTP centers around a concept called the Request-Response Cycle.

The client sends the server a request to do something. The server, in turn, sends the client a response saying whether or not the server could do what the client asked.

Web APIs - Req/Res





JSON

APIs have adopted JSON as a format because it's built on the JavaScript, which is found everywhere on the web and usable on both the client side and server side of web apps.

JSON is a very simple format that has two pieces: keys and values. Keys represent an attribute about the object being described.

API Key Authentication

API Key authentication is a technique that overcomes the weakness of using shared credentials by requiring the API to be accessed with a unique key.

In this scheme, the key is usually a long series of letters and numbers that is distinct from the account owner's login password. The owner gives the key to the client, very much like a hotel gives a guest a key to a single room

What is Node?

Node.js is a platform. The creators of Node.js took javascript and allowed it to run on your machine. It’s built on Chrome's V8 JavaScript runtime.

The main idea is that Node.js uses an event-driven, non-blocking I/O model to remain lightweight and efficient

What Node Achieves

One of the more difficult problems in writing systems that communicate over a network is managing input and output — that is, the reading and writing of data to and from the network, the hard drive, DBs and other such devices.

Node was initially conceived for the purpose of making ***asynchronous*** I/O easy and convenient.

Where Node Excels

Any place where we can perform asynchronous I/O - reading and writing to network connections, reading/writing to the filesystem, and reading/writing to the database.

All of these are common tasks in web apps and execute very fast in Node.

- JSON APIs
- Web Apps
- Command Line Utilities/Apps
- Chat Apps

Modules

Node puts little functionality in the global scope.

If you want to access other built-in functionality, you have to ask the module system for it.

require

Node.js follows the CommonJS module system, and the builtin `require` function is the easiest way to include modules that exist in separate files.

The basic functionality of `require` is that it reads a javascript file, executes the file, and then proceeds to return the exports object.

Node Core Modules

File System

- Handles File I/O

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

HTTP/HTTPS

- Interfaces designed to support features of the http or https protocol.

```
var http = require('http');
var https = require('https');
```

module.exports

The basic functionality of `module.exports` is that it allows a variable, object, function, etc to be exported and used in another file.

```
module.exports.year = 2015;

module.exports.addTwo = function(n) {
  return n + 2;
};
```

NPM

npm (Node Package Manager) is the default package manager for Node.js.

npm is bundled and installed automatically with the node. npm manages dependencies for an application.

https://www.npmjs.org/

NPM Basics

```
npm init
```
// assist in creating the package.json.

```
npm install
```
// installs all modules in the package.json - installed in the node_modules directory.

```
npm install --save <module-name>
```
// installs the module by name and auto-magically adds it to package.json.

```
npm uninstall --save <module-name>
```
// removes the module by name and auto-magically removes it to package.json.