Modul 3

Node JS & REST API

Purwadhika
Startup and Coding School

Outline

- What is Node JS?
- Why Node JS?

What is Node JS

“Node JS is an open-source and cross-platform Javascript runtime environment that executes Javascript outside of a web browser”.

- Node JS is NOT a programming language.
- It is a runtime which allows Javascript to run on the server.
- Previously, Javascript was designed as a scripting language that runs in the browser.

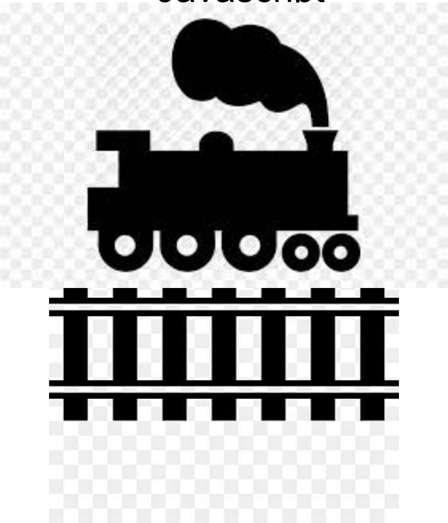
Browser

Server

Browser

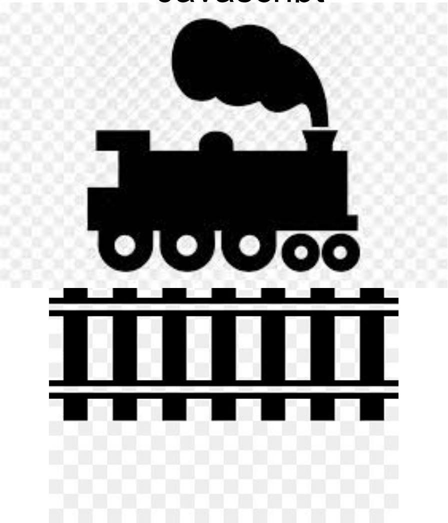
Server

Javascript



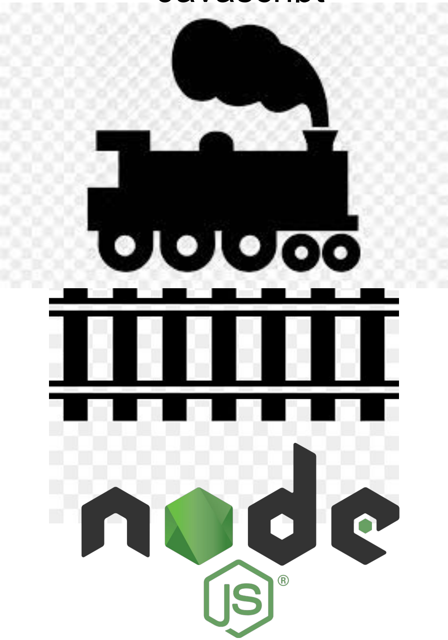
Browser

Javascript

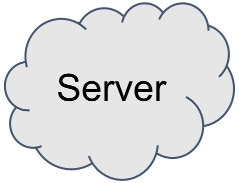


Server

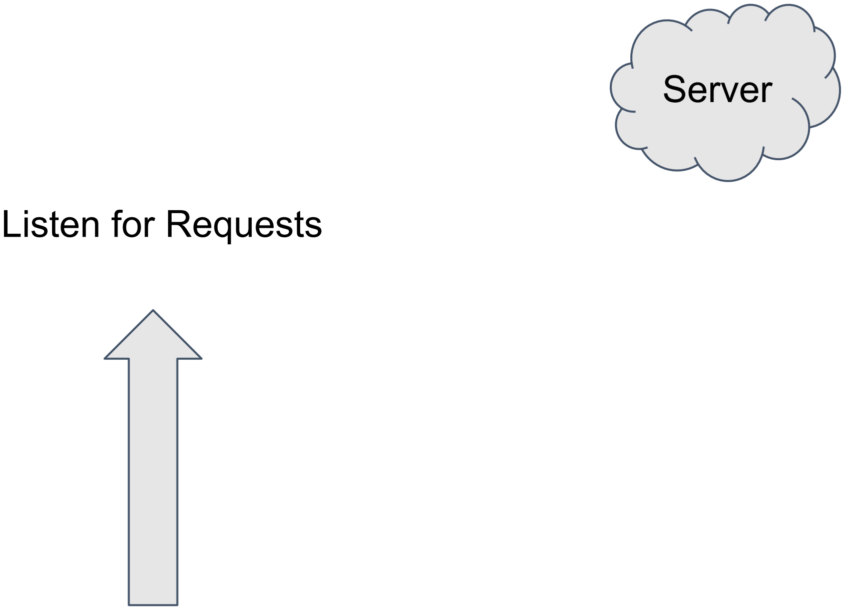
Javascript



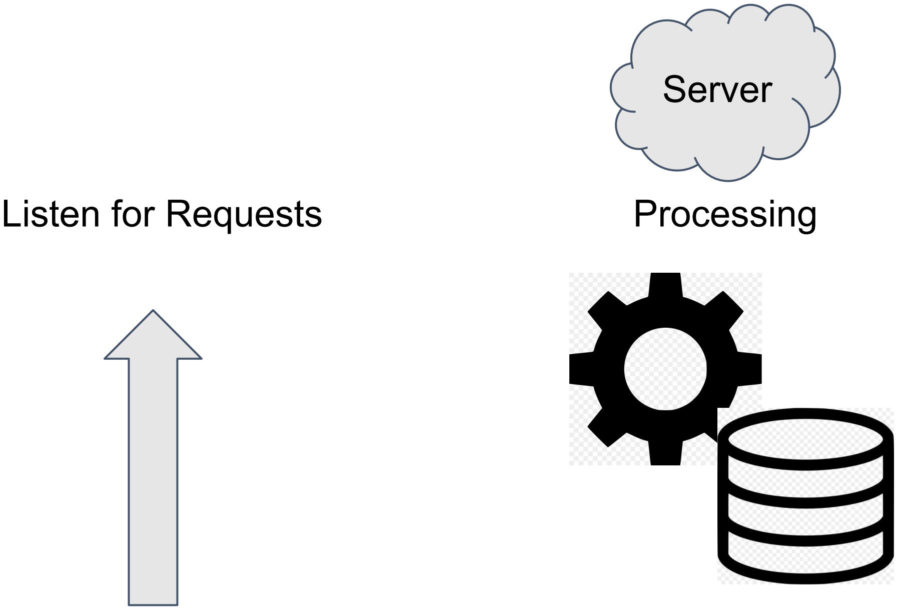
In Web Development



Server



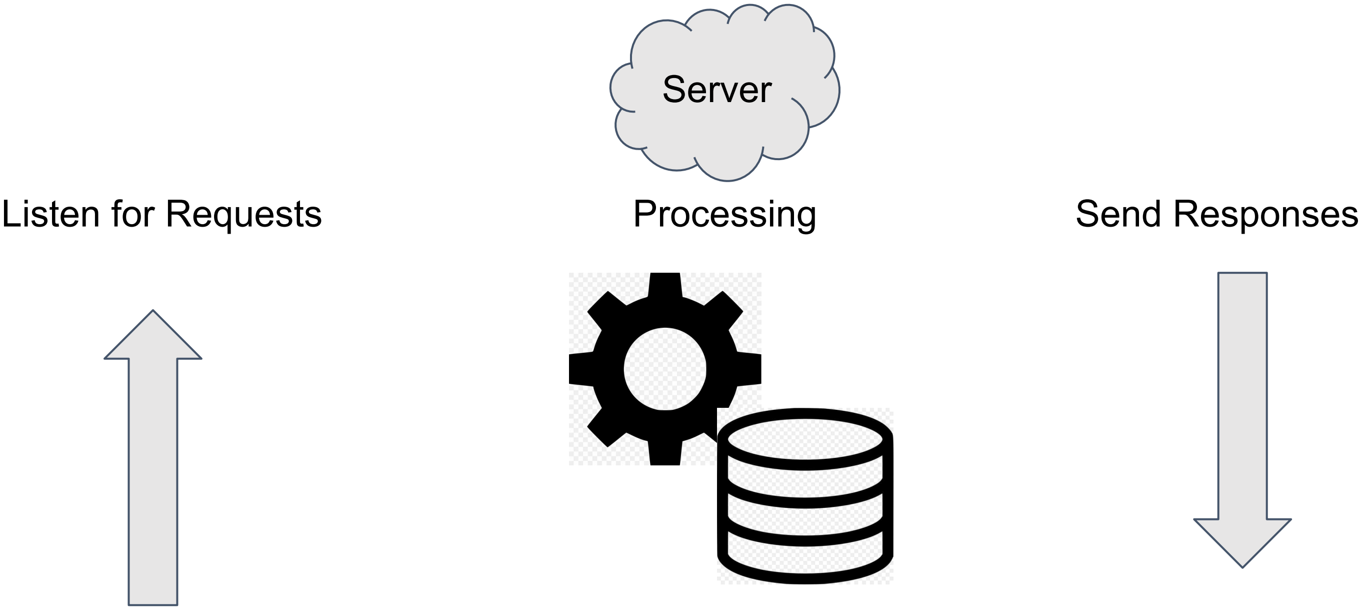
Listen for Requests



Server

Listen for Requests

Processing



Server

Listen for Requests

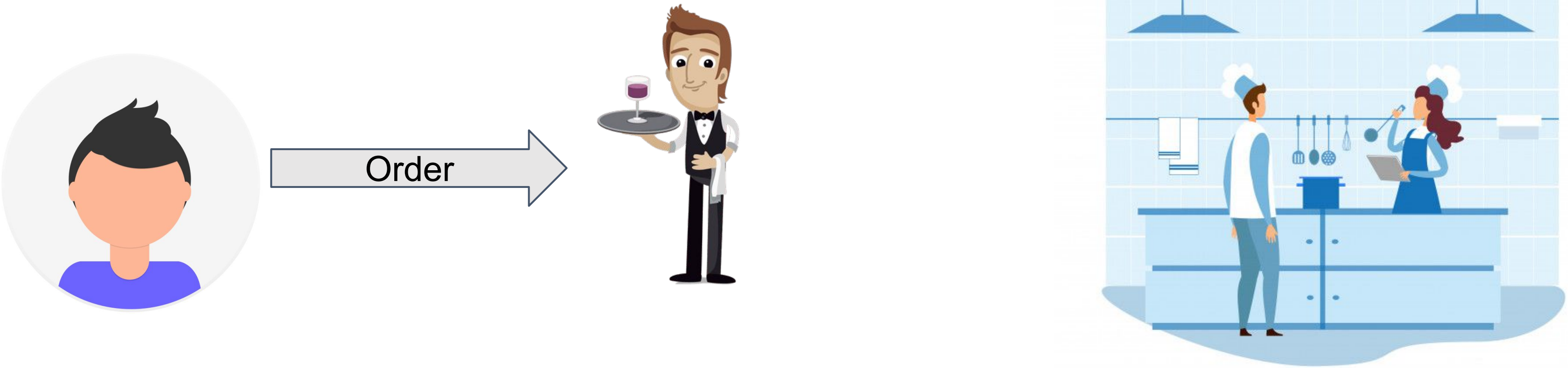
Processing

Send Responses

Server - Side JavaScript

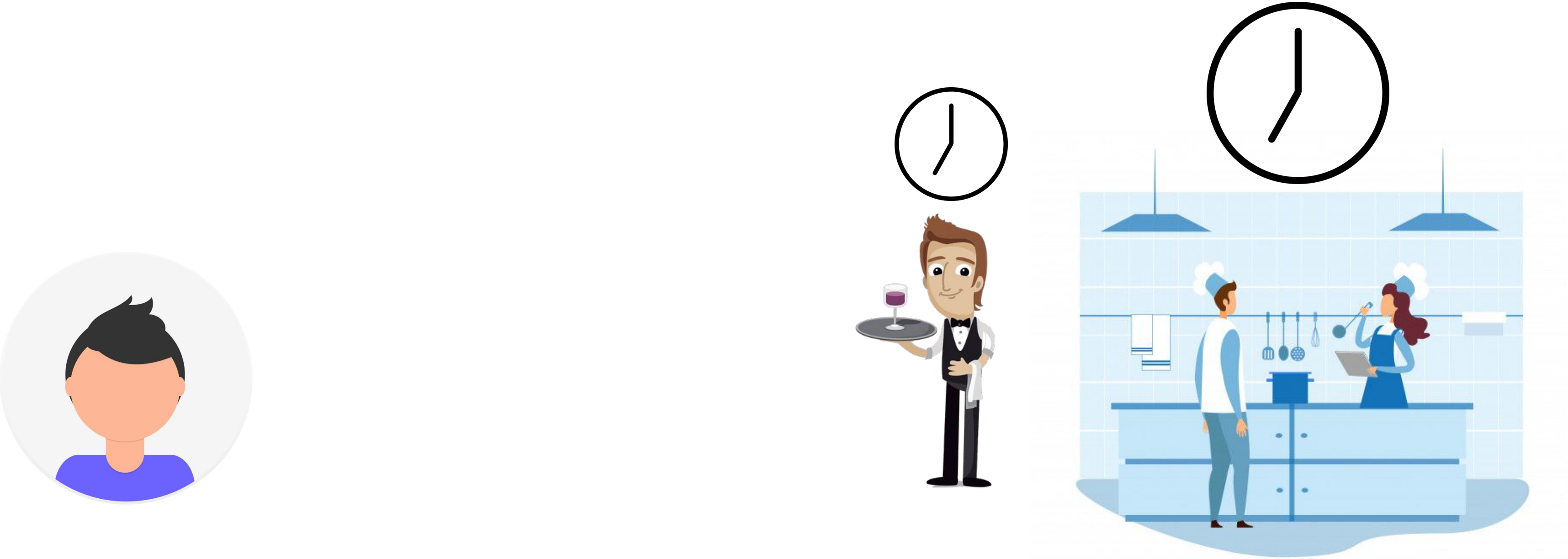
- Node JS allows us to use the JavaScript language to create and develop a server-side application.
- Node JS uses an event-driven, non-blocking i/o model, asynchronous architecture

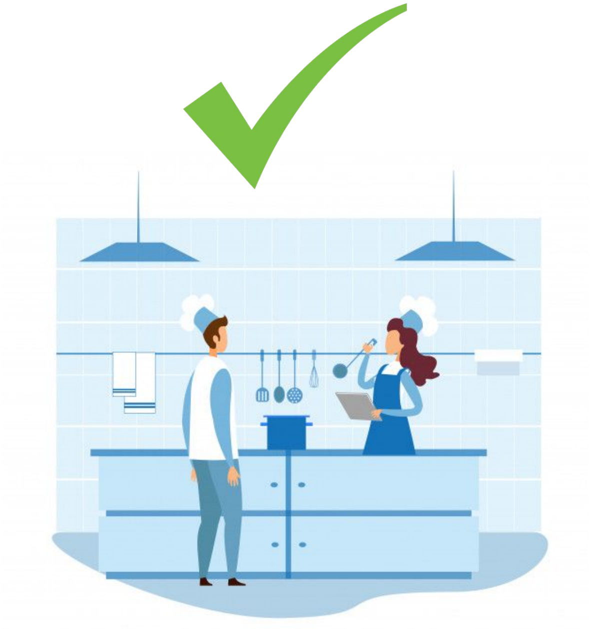
Case 1

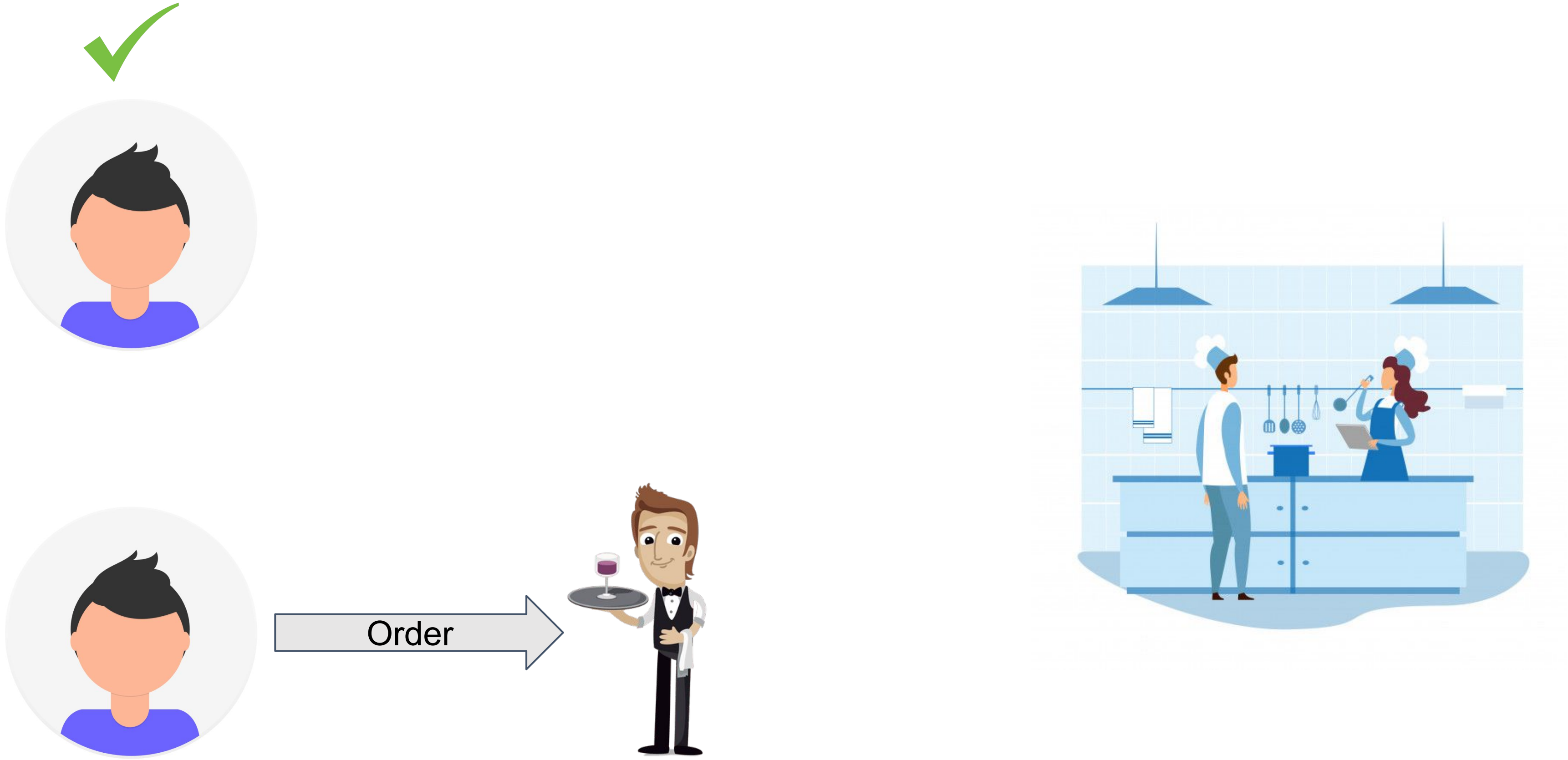


Order

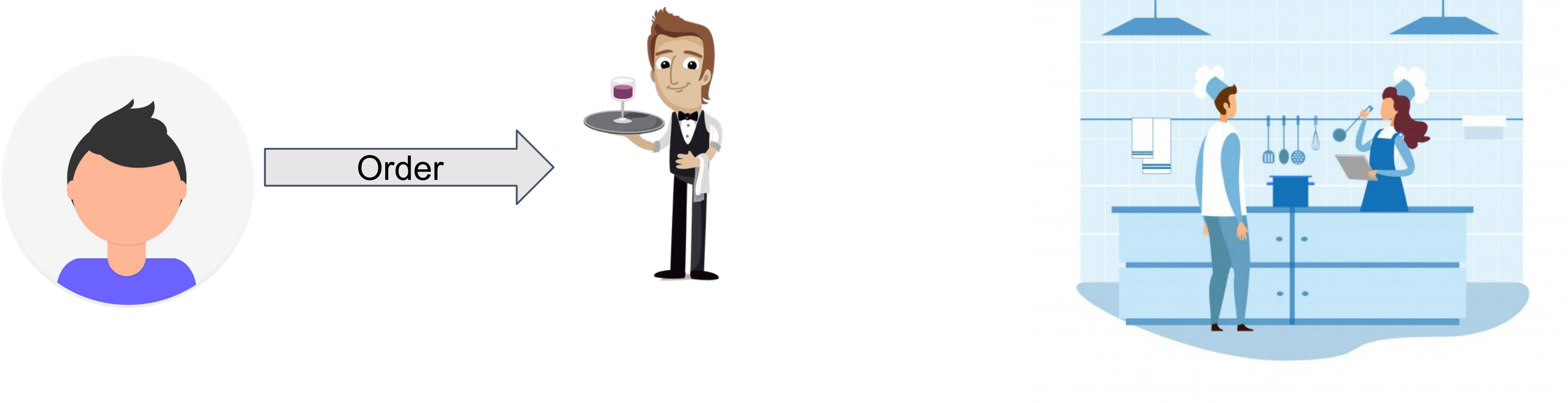
Order





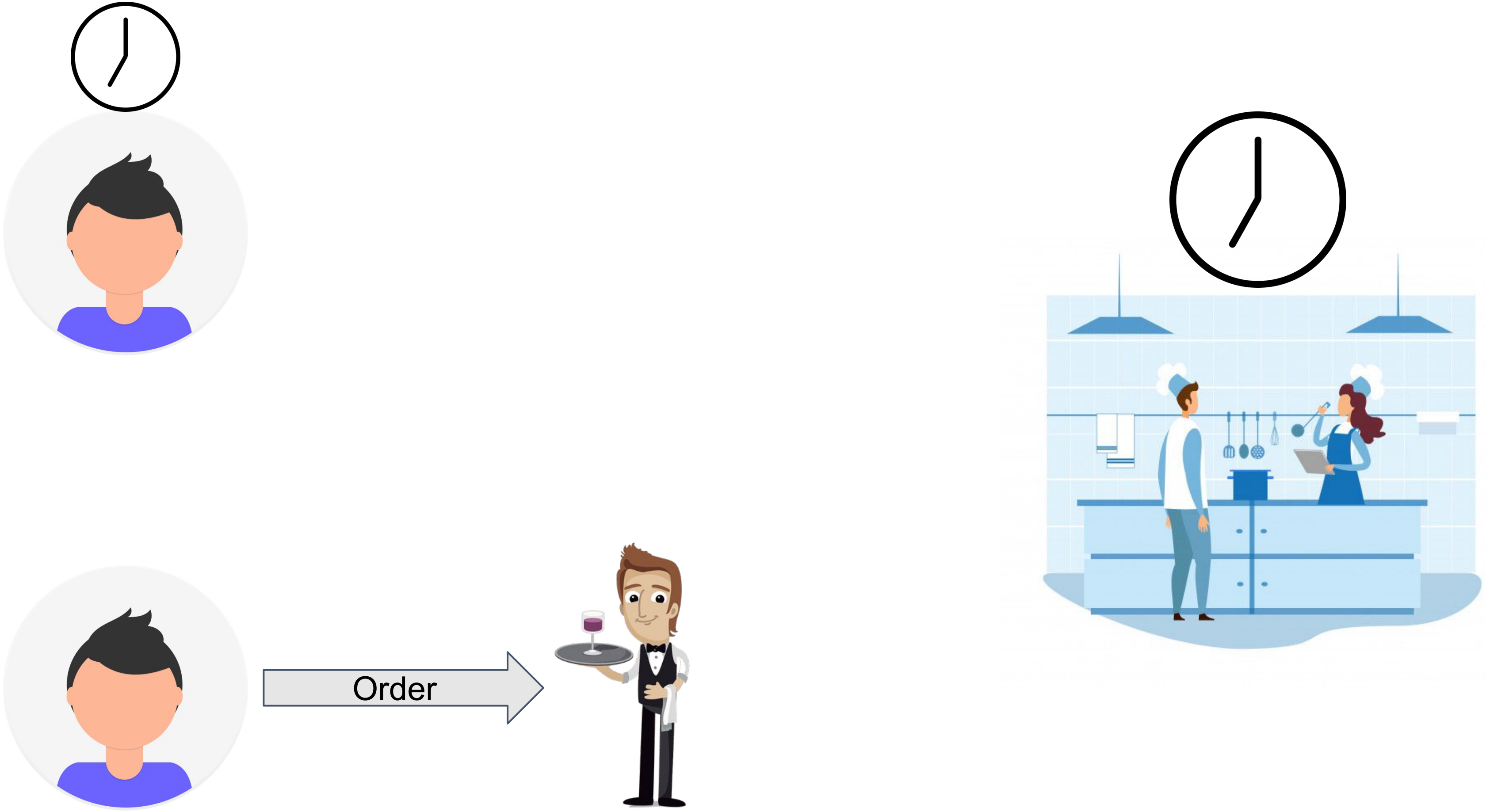

Order

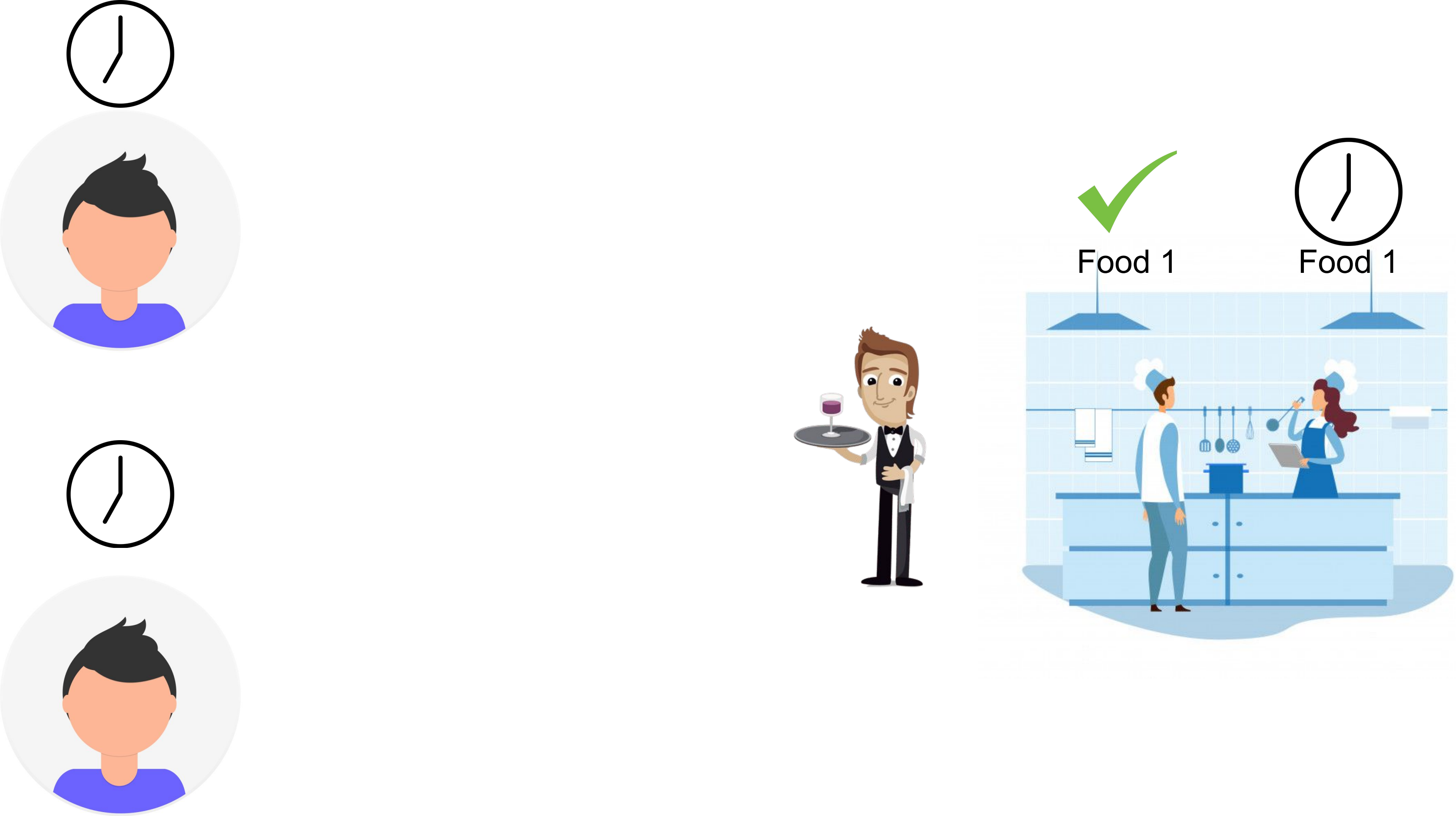
Case 2



Order

Order


Order



Purwadhika
Startup and Coding School

Case 1 vs Case 2

Synchronous

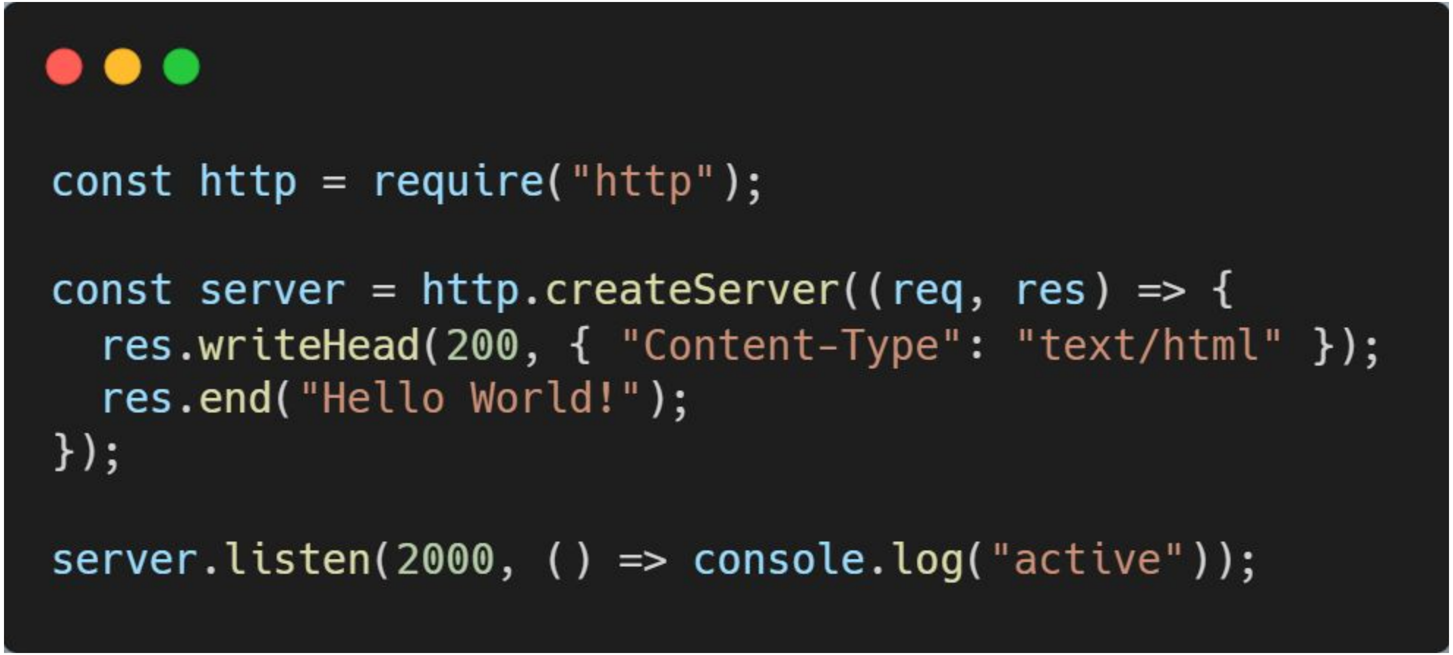
- The first order must be done for the waiter to continue with the next order
- The first order BLOCKS the progress of the second order
- This restaurant needs more waiters to be more efficient

Asynchronous

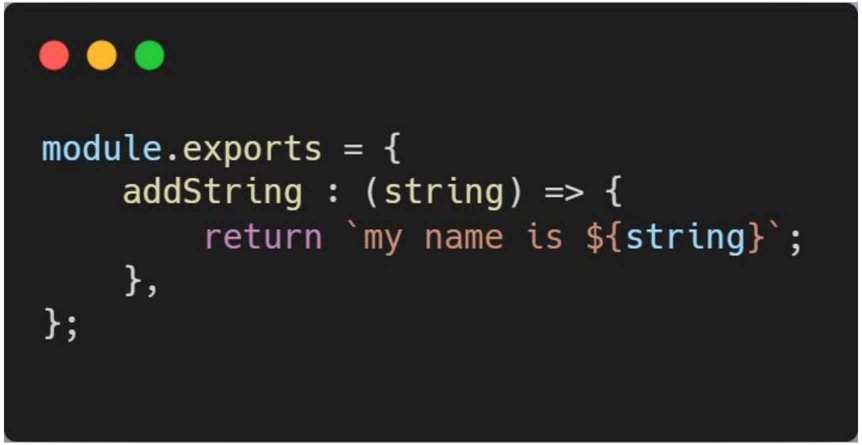
- While the first order is cooking, the waiter collects the next order.
  - Non-blocking
- When the kitchen is finished cooking, the waiter collects the food and delivers it to the customer.
  - Event-Driven (Event-queue)

Modules

- Modules are essentially JavaScript Libraries
- Which are a set of functions to be included in your Node JS application
- Node JS has 3 types of modules
  - Built-in modules which require no downloads and are readily available
  - NPM modules which are downloaded from NPM.
  - Your own module.
- Modules are added in your Node JS applications by using the require() function.



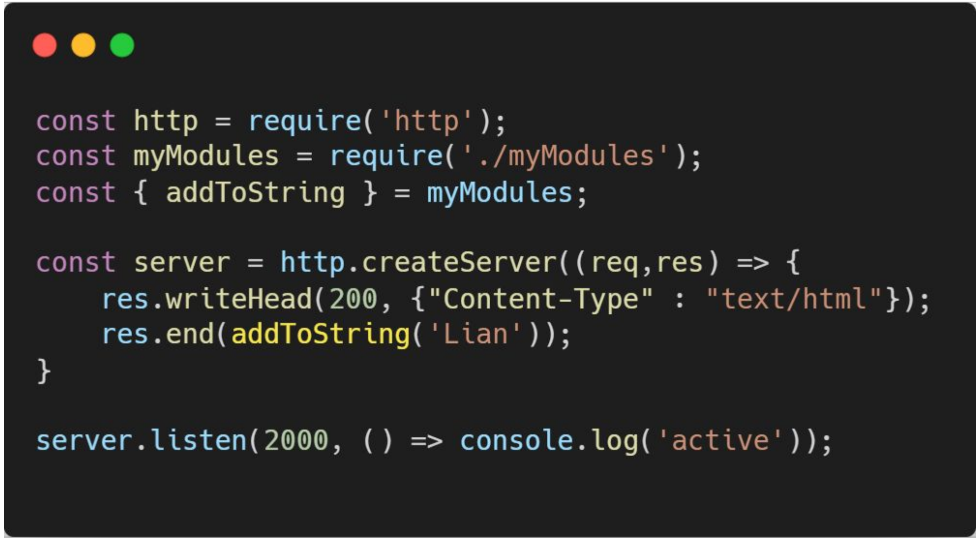
```javascript
const http = require("http");

const server = http.createServer((req, res) => {
  res.writeHead(200, { "Content-Type": "text/html" });
  res.end("Hello World!");
});

server.listen(2000, () => console.log("active"));
```



```
module.exports = {
    addString : (string) => {
        return `my name is ${string}`;
    },
};
```

myModules.js

index.js



```
const http = require('http');
const myModules = require('./myModules');
const { addToString } = myModules;

const server = http.createServer((req,res) => {
    res.writeHead(200, {"Content-Type" : "text/html"});
    res.end(addToString('Lian'));
}

server.listen(2000, () => console.log('active'));
```

Purwadhika
Startup and Coding School

NPM

- NPM or Node Package Manager comes with Node JS
- The NPM registry has become the center of Javascript code sharing
  - With more than one million package becoming the largest software registry in the world.
- Basically NPM allows you to install modules in your project
  - Modules are additional Javascript libraries.
  - A package contains all the files you need for a module.
- NPM will create an additional folder called node_modules
  - Which will contain all the packages
  - Future libraries downloaded will be inside of this folder

Package.json

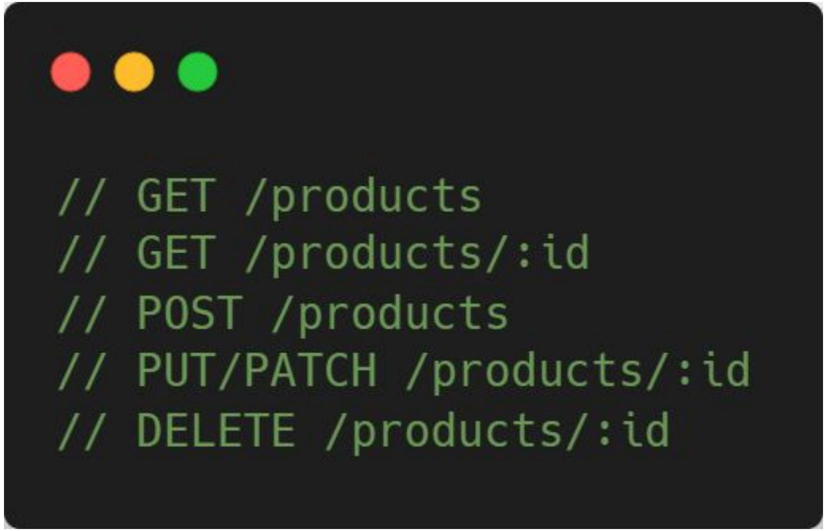
- This file contains various metadata relevant to the project
- This file is used to give information to NPM that allows it to identify project as well as its dependencies.

REST API

- REST or Representational State Transfer is an API architecture style
    - Basically it is a set of rules developers follow when creating an API.

- REST breaks down a transaction into smaller modules
    - Where each module addresses a particular part of the transaction

- Modularity creates a more flexible and scalable API



```
// GET /products
// GET /products/:id
// POST /products
// PUT/PATCH /products/:id
// DELETE /products/:id
```

- A REST based API makes use of the following functions:
  - **GET** − Provides read-only access to a resource.
  - **POST** − Creates a new resource.
  - **DELETE** − Removes a resource.
  - **PUT/PATCH** − Updates an existing resource

Modul 3

Node JS & REST API

Purwadhika
Startup and Coding School