Install Node.js

Install CLI using cmd

Install VSCode

To create new angular project use CMD: ng new mean-course

Cmd to run server: ng server

Stop server: cltr+C

File usability:

Package.json : used for dependencies

E2e: end to end conenections

Angular.json: angular cli

SRC >> APP : where our main app goes

Index.htm file works as a single page application

String manipulation and property binding using value object.

Two-way binding.

Now install angular material to use some readymade forms and other components;

To install in latest angular CLI: ng add @angular/material

To install in older one’s like less than version 6: npm install –save @angular/material

Directives to manipaulate in the dom. Dynamically looping using structural directives.

Example: \*ngFor=”let post of posts”

<mat-accordion multi="true">

  <mat-expansion-panel \*ngFor="let post of posts">

    <mat-expansion-panel-header>

      {{ post.title }}

    </mat-expansion-panel-header>

    <p>{{ post.content }}/p>

  </mat-expansion-panel>

</mat-accordion>

Create interface to define certain rules in case of declarations.

Import those interface and use to limit different variable and also use in events as generic types.

Service is a class which let inject angular to components and centralize some tasks.

Dependency injection. Constructor

Rxjs >> Observable

“ng serve” gives us a development server.

Two ways of connecting Node+Angular: Node App Serves Angular SPA, Two Separated Servers

RESTful API: Representational State Transfer. Serverside app

To run js file in the node server run cmd: node name\_of\_the\_JS\_file

To install Express run: npm install –save express

Call next() if you’re not sending a response using middlewears .

Add an extra package which will help to not run node server again and again to run js files. To install run: npm install --save nodemon

To show dependency visible through the project register the package in “package.json” file under “scripts” section

"start:server": "nodemon server.js"

To run the project now use : npm run start:server

To run this package you must import “debug package”

const debug = require("debug")("node-angular");

CORS >> Cross-Origin Resource Sharing : This CORS error occurs when you’re not in the same server.

We need to disable the CORS error so that our API can be run by all types of servers and clients.

Use Headers to solve problems

app.use((req, res, next) => {

  res.setHeader("Access-Control-Allow-Origin", "\*");

  res.setHeader(

        "Access-Control-Allow-Header",

        "Origin, X-Requested-With, Content-Type, Accept"

  );

  res.setHeader(

    "Access-Control-Allow-Methods",

    "GET, POST, PATCH, DELETE, OPTIONS"

  );

  next();

});

To parse parse incoming request install body-parser package and run: npm install -- save body-parser

Install MongoDB Atlas which is a cloud DB.

Create user: Saif001 | PW: by5Ky6p0PypzxFyE | IP: 103.85.233.181

Add a package to help us to connect MongoDB cloud DB. For that we’ll use “mongoose”. “mongoose” uses Schemas. To install run: npm install –save mongoose. ID’s are created by mongoose by default.

\*\*\*\*\*TypeScript: lower case ‘string’ , Node.js or JS: Upper case ‘String’

Go to MongoDB bin to look at the DB through powershell.

* Run: mongo "mongodb+srv://cluster0-bzdu4.mongodb.net/test" --username Saif001
* Enter password
* Then direct to your folder. Ex: use node-angular
* Then to see collections run: show collections
* Run: db.collection\_name.find() - to find the documents inside collection
* Run: help - to see the cmds

Angular router to add multiple pages.

\*\*\*\*\*\*\*\*\*Client side vs Server side Routing

Elvis (?) Operator… this operator force html to check if there’s any request or fill with nothing

Creating a filepicker to upload image by hiding the visibility of input

<div>

      <button mat-stroked-button type="button" (click)="filePicker.click()">Pick Image</button>

      <input type="file" #filePicker>

    </div>

input[type="file"] {

  visibility: hidden;

}

Reactive form approach to upload images

1. Remove NgModel and use ReactiveFormsModule.
2. import { FormGroup, FormControl, Validators } from '@angular/forms';

To export file in server we will use a package name multer. To install it run : npm install –save multer

Now, It’s time to add pagination. To do that use “MatPaginatorModule” from angular material

Creating Authntication like login and signup using template driven approach by using FormsModule.

Install a package for Authenticate unique validation of emails using an extended pachage of mongoose .

**npm install --save mongoose-unique-validator**

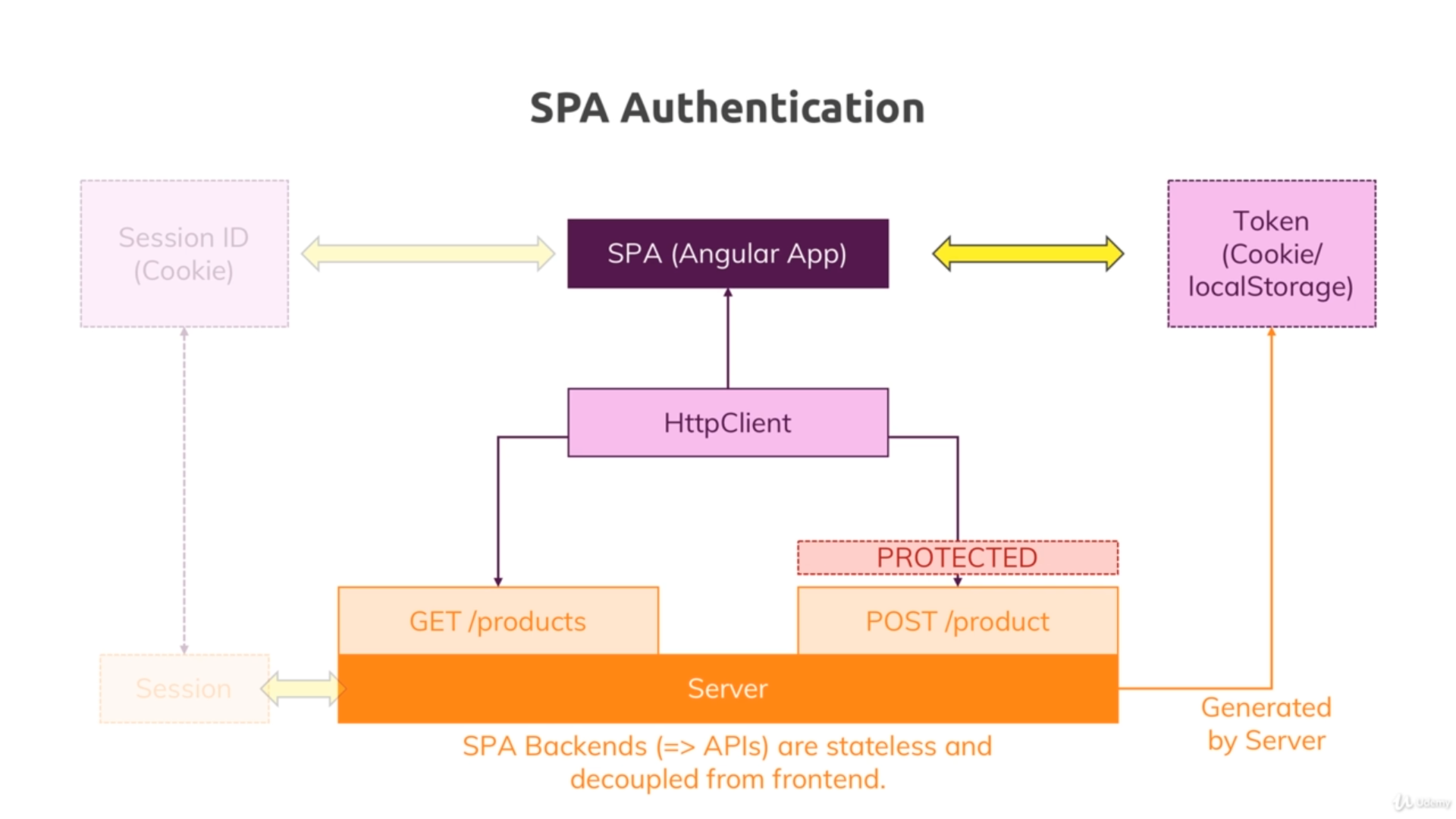
**to use the plugin simply import**

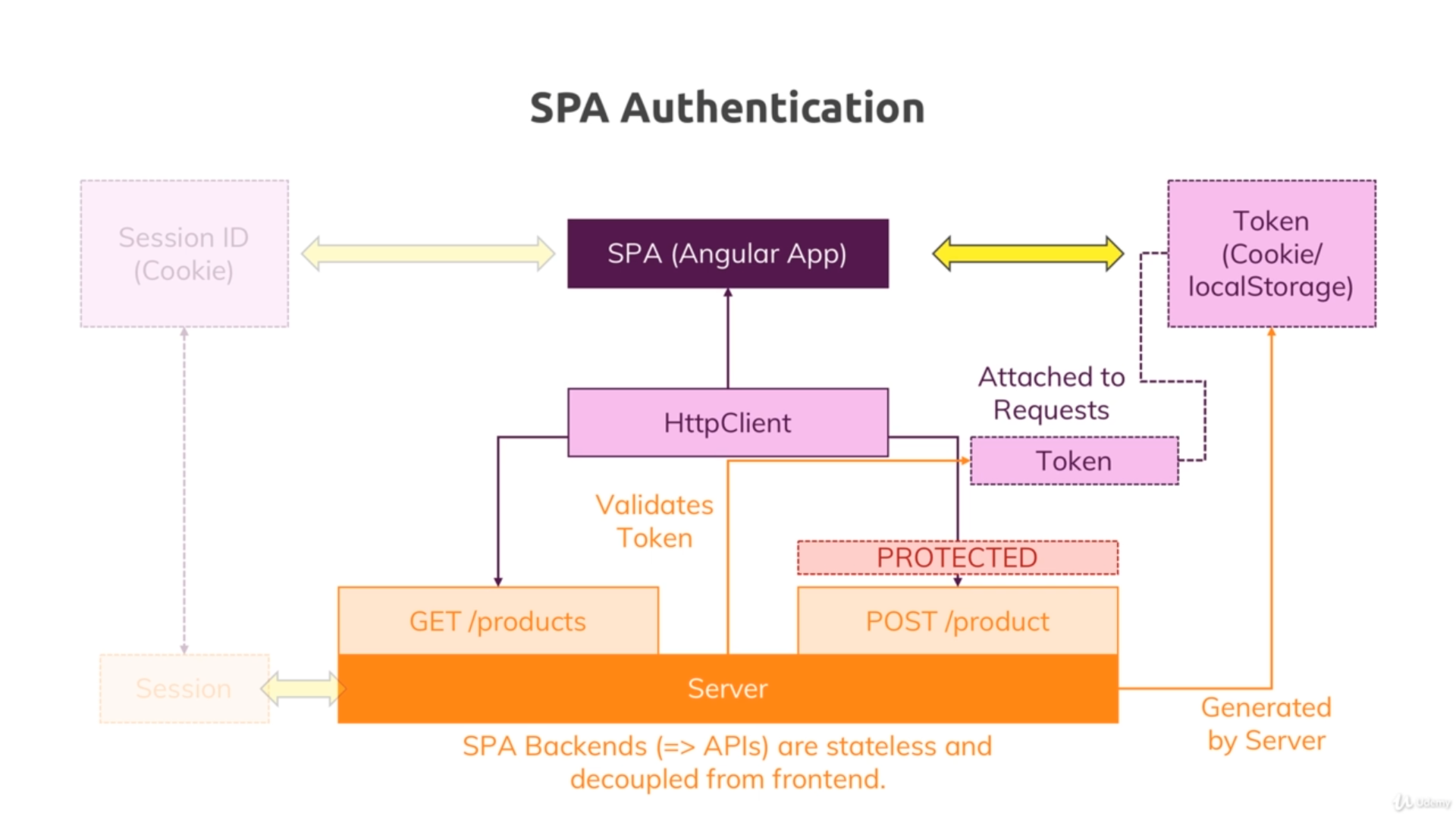
const uniqueValidator = require('mongoose-unique-validator');

userSchema.plugin(uniqueValidator);

Install bcrypt to encrypt the password data.

npm install --save bcrypt





We need to create JSON Web token

To create web token there’s a package name “jsonwebtoken”. To install it run: npm install –save jsonwebtoken

**When database connection failed locally change your IP address**

**Run this to upgrade from css to scss.**

ng config schematics.@schematics/angular:component.styleext scss

Deploying the app is server which has two ways:

