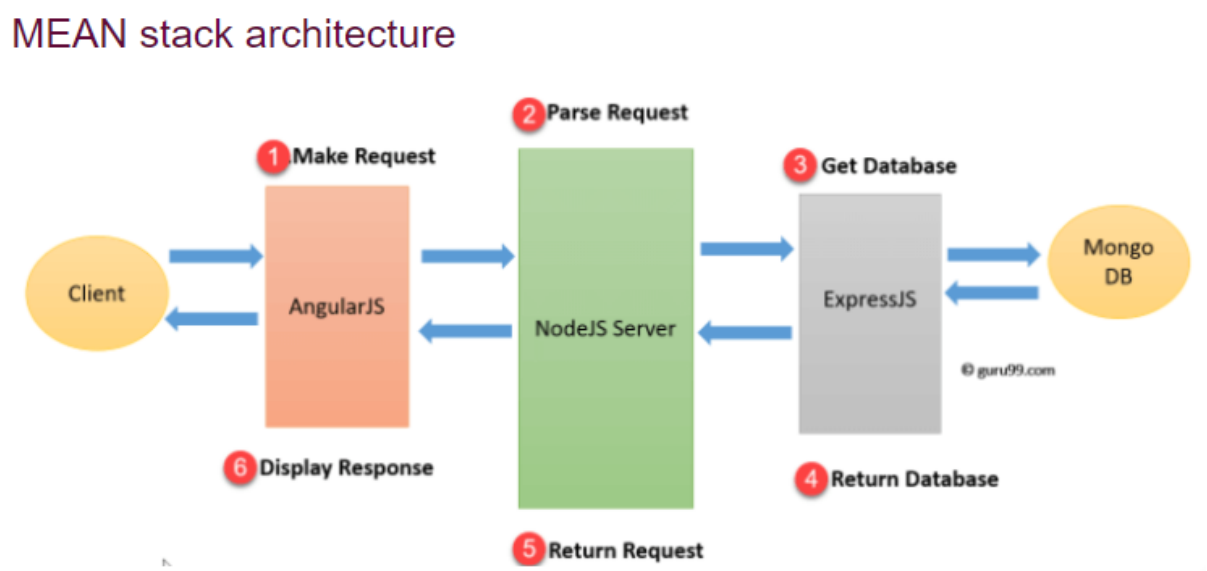
CORSO ANGULAR & NODEJS- THE MEAN STACK GUIDE 2021(https://www.javatpoint.com/mean-stack **,** https://itnext.io/crud-operation-using-mean-stack-7dfa2f51ec8c **)**



MEAN(MongoDB- Express- ANgular-NodeJS)
Express: is a Node framework which simplifies writing server-side code and Logic:
-bases on node , offers same Functionalities
_Middleware-based Funnel Requests through Functions
-Includes Routing, View- Rendering & More
MongoDB is a **NoSQL** Database wich stores “Documents” in “Collections” (Instead of ‘Records’ in ‘Tables’ as in SQL):
- Store Application Data(Users, Products,...)
-Enforces no Data Schema or Relations
-Easily connected to Node/Express Not to angular
Is a powerfull Database which can easily be integrated into a Node/Express Environment

WE BUILD SINGLE PAGE APPLICATION
In the single page we have one index.html. import/load(<script src=””>) Re-renders page this allow for instant re-rendering, instant user feedback and makes building engages

MEAN the big picture
client(Browser) :
-we use Angular
-presentation UI
-Single page Application is not necessarily served by Node Backend

Requests
Responses:
-Data
JSON Format, Ajax(Backgound)

server:

-we uses Node express and MongoDB
-Business Logic
-Persistent Data Storage
-Authentication Logic

ng new my-project --no-strict
Install Visual studio code :
install extensions Angular essentials and Material icon theme

Getting Started
Angular Frontend
NodeJS + Express BACKend
Handling Data With MongoDB
Enhancing the App
Image Upload
Data Pagination
Authentication
Authorization
Error Handling
Optimizations
Deployment

Getting The Most Out Of The Course
-Watch the videos -> at your speed, pause & rewind
-Code Along
-Use the course Resources -> Attached code & Links
-Ask in Q&A
-Help others in Q&A

The Angular Frontend- Understand the Basics
Module Introduction
-Understanding the folder structure
Understanding Angular Components
Adding our first Component
-Listening to Events
-Outputting Content declared in the component and display to view with interpolation where you want to see.
Getting User input: added FormsModule to import in app.module.ts

Two Ways of Connecting Node + Angular

1.Node App Serves Angular SPA
Node (Express) handles incoming requests
Requests targeting “/”path return Angular SPA

2.Two Separated Servers
Node (Express) handles incoming requests
Angular SPA served from separate static host

In both cases: Logically separated Apps

What is RESTful API or REST(Representational State Transfer)
Request - Server - Response

RESTful APIs are Stateless Backends
RESTfull Server(API)
/users /posts / products
GET, POST(response with JSON data request AJAX),DELETE GET, POST,DELETE GET, POST,DELETE

Client

Adding the Node Backend

create folder **backend** inside projet localhhost://4000 Frontend , localhhost://3000 Backend
C:\xampp\htdocs\AngularProject\AngularTourOfProducts\angularEcommerce\backend
create file **server.js** inside src project
C:\xampp\htdocs\AngularProject\AngularTourOfProducts\angularEcommerce\client\server.js

To launch server node server.js localhhost://3000
inside serve.js

```
const http = require('http');
const server = http.createServer((req, res) => {
  res.end('This is my first response');   // res.head()
});

server.listen(process.env.PORT || 3000);
```

Install the dependencies in the projet

> npm init per il package.lock.json
>npm install per installare le dependencies

Adding the Express framwork

Install Express framework of nodeJs
npm install --save express

Improving the server.js Code

In the folder backend add app.js file

```
const express = require('express');   //
const app = express();                // execute the express package
app.use((req, res, next) => {
  console.log('First middleware');
  next();     //senza questa il server non manda la risposta, fa si che dopo la risposta continua
a girare
});
app.use((req, res, next) => {
  res.send('Hello from response3');
```

```
});
module.exports = app;
```

Il middleware è il software che risiede tra un sistema operativo e le applicazioni eseguite in esso, per consentire la comunicazione e la gestione dei dati.

install nodemon npm install --save-dev nodemon
nodemon is a tool that helps develop node. js based applications by automatically restarting the node application when file changes in the directory are detected. ... nodemon is a replacement wrapper for node

In the server.js analiza port value insert by parseInt
ParseInt(val,port)
https://developer.mozilla.org/it/docs/Web/JavaScript/Reference/Global_Objects/parseInt
agguinge questo al file package.json "e2e": "ng e2e", **"start:server"**: "**nodemon server.js" o "start" : "nodemon app.js"** per lanciare il server con **start:server** oppure **npm run star**t invece di sempre **nodemon server.js**
run **npm run start:server**

fetch initial Categories

inapp.js localhost:3000/categories

```
app.use("/categories", (req, res, next) => {
  const categories = [
    {
      id: 1,
      categoryName: "Bevande"
    },
    {
      id: 2,
      categoryName: "Igiene personale"
    }
  ];
  res.status(200).json({
    message: 'Categories fetched succesfull',
    categories: categories
  });
  next();
});
```

Using the Angular HTTP Client

Understanding CORS(Cross-Origin Ressource Sharing)

client (localhost:3000) - server (localhost:3000)
client (localhost:4000) - server (localhost:3000)
Il **Cross-Origin Resource Sharing** (CORS (en-US)) è un meccanismo che usa header HTTP addizionali per indicare a un browser che un'applicazione Web in esecuzione su

un'origine (dominio) dispone dell'autorizzazione per accedere alle risorse selezionate da un server di origine diversa. Un'applicazione web invia una cross-origin HTTP request quando richiede una risorsa che ha un'origine (protocollo, dominio e porta) differente dalla propria. CORS Error è un meccanismo

Getting Started with Angular Material

npm install - save @angular/material
ng add @abgular/material

Adding the Post Backend Point(app.post() invece di app.use() gestito da middleware)

nella sezione precedente abbiamo utilizzato il **middleware**(Insieme di software che fungono da intermediari fra strutture e programmi informatici, permettendo loro di comunicare a dispetto della diversità dei protocolli o dei sistemi operativi.) per agguingere alcune intestazione(setHeader) per risolvere l'errore CORS. In questo si come non abbiamo ancora un data base per archiviare i dati, la richiesta post a un corpo quindi hanno dati allegati e dobbiamo estrarre quei dati per ora installiamo il pachetto body-parser.

```
app.use((req, res, next) => {
   res.setHeader("Access-control-Allow-Origin", "*");

   res.setHeader(
      "Access-Control-Allow-Headers",
      "Origine, X-Requested-With, Content-Type, Accept");
   res.setHeader("Access-Control-Allow-Methods",
      "GET, POST, PATCH, DELETE, OPTIONS");
});
```

Body Parser

Non abbiamo ancora un database, quindi non possiamo archiviarli. Per ora, non possiamo memorizzare, quindi non saremo in grado di elaborarli e recuperarli, ma possiamo verificare se il trasferimento dei dati su quel percorso funziona. Quindi, pubblicheremo semplicemente i post che abbiamo ricevuto con console.log.app.post("/api/posts", (req, res, next) => {
   console.log();
});La richiesta di post ha un corpo, quindi hanno dati allegati e dobbiamo estrarre quei dati. Ora, per estrarre i dati, dobbiamo installare un pacchetto aggiuntivo che aggiunge un pratico middleware che collegheremo alla nostra app express. Estrarrà automaticamente i dati della richiesta in arrivo e li aggiungerà come nuovo campo a quell'oggetto richiesta per accedervi comodamente. Useremo il seguente comando per installare il pacchetto:
npm install --save body-parser
Si tratta di un pacchetto node express che può essere utilizzato come middleware express. Il parser del corpo fa esattamente ciò che suggerisce il nome. Analizza i corpi delle richieste in entrata, estrae i dati delle richieste perché quello sarà un flusso di dati e li converte in un oggetto dati.

Lavorare con MongoDB

-Introduzione
-cos'è MongoDB?
-Confronto tra SQL e NoSQL
-Collegamento di Angular a un database
-Configurazione di MongoDB
-Utilizzo di MongoDB Atlas e IP Whitelist
-Aggiunta di mangusta(Mongoose).
Mongoose è un ottimo strumento che potrebbe funzionare anche con dati così non strutturati
npm install --save mongoose invece di npm install --save mongodb
questo strumento ci permetterà di connetterci al nostro MongoDB e anche di interagire con esso per memorizzare dati e recuperare dati.
-Understand Mongose Schema & Models
-Creating a Product instance

Instal Angular Material
ng add @angular/material
to updare the version run npm install --save @angular/matrial@8 --save -exact

Output content (Events)
Local Reference: imposta un valore d'ingresso manualmente nel html e lo recupera nel modello dopo il click puoi stampa nel html.

```
newPost = '';
  public onAddPost(postInput: HTMLTextAreaElement) {
          this.newPost = postInput.value;
   }
<mat-form-field>
     <textarea matInput rows="6"  #postInput></textarea>
   </mat-form-field>
   <button mat-raised-button color="primary" (click)="onAddPost(postInput)">Save Post</button>
<p>{{newPost}}</p>
```

Two way data binding
La rilegatura a due vie combina **l'impostazione e la lettura del valore**. Si chiama associazione a due vie perché ha un flusso bidirezionale. Useremo **ngMode**l che è una direttiva e **ascolteremo l'input** dell'utente. Emette i dati e memorizza anche i nuovi dati in quell'area di testo o output

Connessione di Angular all'endpoint API
.Install MongoDB download MongoDB for windows
Install NoSQLBooster for MongoDB

Connection to Data base to shell (access from the MongoDB shell)
>mongo "mongodb+srv://cluster0.87qg8.mongodb.net/admin" --username sandrine
> enter password:sandrine2021
> use admin
>show users

how to create #responsive #nav #bar using #angular #material in 10 second
ng g @angular/material:material-nav --name=main-nav

Adding image Upload to our App

Adding file input button
<button mat-stroked-button type="submit" (click)="filePicker.click()">pick image</button>
<input type="file" #filePicker/>
in the css hidden input file and when we click to the button is as we click to input

Schema for API Developement

Post

- /api/post/list <- GET = invia una lista di posts / non riceve parametri
- /api/post/create <- POST = invia un esito (true/false) / riceve un oggetto codificato in JSON tipo Post
- /api/post/update <- PUT = invia tl post aggiornato / Riceve un oggetto codificato in JSON tipo Post
- /api/post/delete <-DELETE = invia un esito (true/false) / Riceve un ID di un Post

Scheme for API Development

Post
- /api/post/list <- GET = Invia una lista di Posts/ Non riceve parametri
- /api/post/create <- POST = Invia un esito (true/false) / Riceve un oggetto codificato in JSON tipo Post
- /api/post/update <- PUT = Invia il Post aggiornato / Riceve un oggetto codificato in JSON tipo Post
- /api/post/delete <- DELETE = Invia un esito (true/false) / Riceve un ID di un Post

Category
- /api/category/list <- GET = Invia una lista di Categories / Non riceve parametri
- /api/category/create <- POST = Invia un esito (true/false) / Riceve un oggetto codificato in JSON tipo Category
- /api/category/update <- PUT = Invia il Category aggiornato / Riceve un oggetto codificato in JSON tipo Category
- /api/category/delete <- DELETE = Invia un esito (true/false) / Riceve un ID di un Category

Video
- /api/video/list <- GET = Invia una lista di Video / Non riceve parametri
- /api/video/create <- POST = Invia un esito (true/false) / Riceve un oggetto codificato in JSON tipo Video
- /api/video/update <- PUT = Invia il Video aggiornato / Riceve un oggetto codificato in JSON tipo Video
- /api/video/delete <- DELETE = Invia un esito (true/false) / Riceve un ID di un Video

Products

- /api/product/list <- GET = Invia una lista di Products / Non riceve parametri
- /api/product/create <- POST = Invia un esito (true/false) / Riceve un oggetto codificato in JSON tipo Product
- /api/product/update <- PUT = Invia il Product aggiornato / Riceve un oggetto codificato in JSON tipo Product
- /api/product/delete <- DELETE = Invia un esito (true/false) / Riceve un ID di un Product

online api post request
insomnia

Full stack Mean with Angular

(https://itnext.io/crud-operation-using-mean-stack-7dfa2f51ec8c)