JSON Web token Structure :-

* Header.

* Payload.

* signature

JWT Header :

→ Type of token

→ signing algorithm.

{
 "alg" : " HS256",
 "typ" : "JWT",
}

~~JWT~~ :

JWT PAYLOAD:-

→ Registered claims :- iss(issuen), exp(expination time), sub (subject), aud (audience).

যে ~~ই~~ information গুলো encrypte করা হয় সেগুলোই এখানে payload এর মধ্যে।

creation time, expine dunnation, ~~করে দেয় যেটি করা হলো~~

identy of user এগুলো এখন payload এ।

JSON WEb Token Signature:

→ JWT header + JWT payload base64 url এর মাধ্যমে encode হয় তার সাথে কোনো একটি secret add হয়। add হয় signature তৈরি হয়।

→ puting . all together.

∴ যে সবগুলো একসাথে encoding string তৈরি করে, সে string ঐ data কে represent করে।

Express Rest Api Package

Essential Package:-

* Express : The core backbone

* Body-parser: This is a node.js middleware for handling JSON, Raw, Text and URL encoded form data.

* Cookie-parser: Cookie কে prase করার জন্য cookie-parser ব্যবহার করতে হবে।

* Multer: This is a node.js middleware for handling multipart / form-data.

* Jsonwebtoken: Securely transmitting information between parties as a JSON object.

* My SQL Drive: To access a My SQL database with Node

* Mongo dB Driver: To 〃 〃 Mongo 〃 〃 〃

* Dotenv: Dotenv is a zero-dependency module that loads environment variables.

* Cors: CORS is a node.js package for providing a Connect / Express middleware that can be used to enable CORS with various options.

* Express-mongo-sanitize:- Sanitizes user-supplied data to prevent MongoDB Operator Injection.

* Express-rate-limit: একটা user থেকে প্রতি সেকেন্ড- কতগুলো req accepte করা হবে or manage করা হবে।

* Helmet: Helmet helps you secure your Express apps by setting various HTTP headers.

* HPP: Express middleware to protect against HTTP parameter Pollution attacks.

* Validator: A library of string validators and sanitizers.

* Xss-clean: Connect middleware to sanitize user input coming from POST body, Get queries and url params.

Rest Api Project structure

File Folder Structure :-

→ For Monolithic Application MVC (Model, view controller)

→ For Rest Api MC (Model, Controller).

Project structure.

* index.js

→ App.js → configure

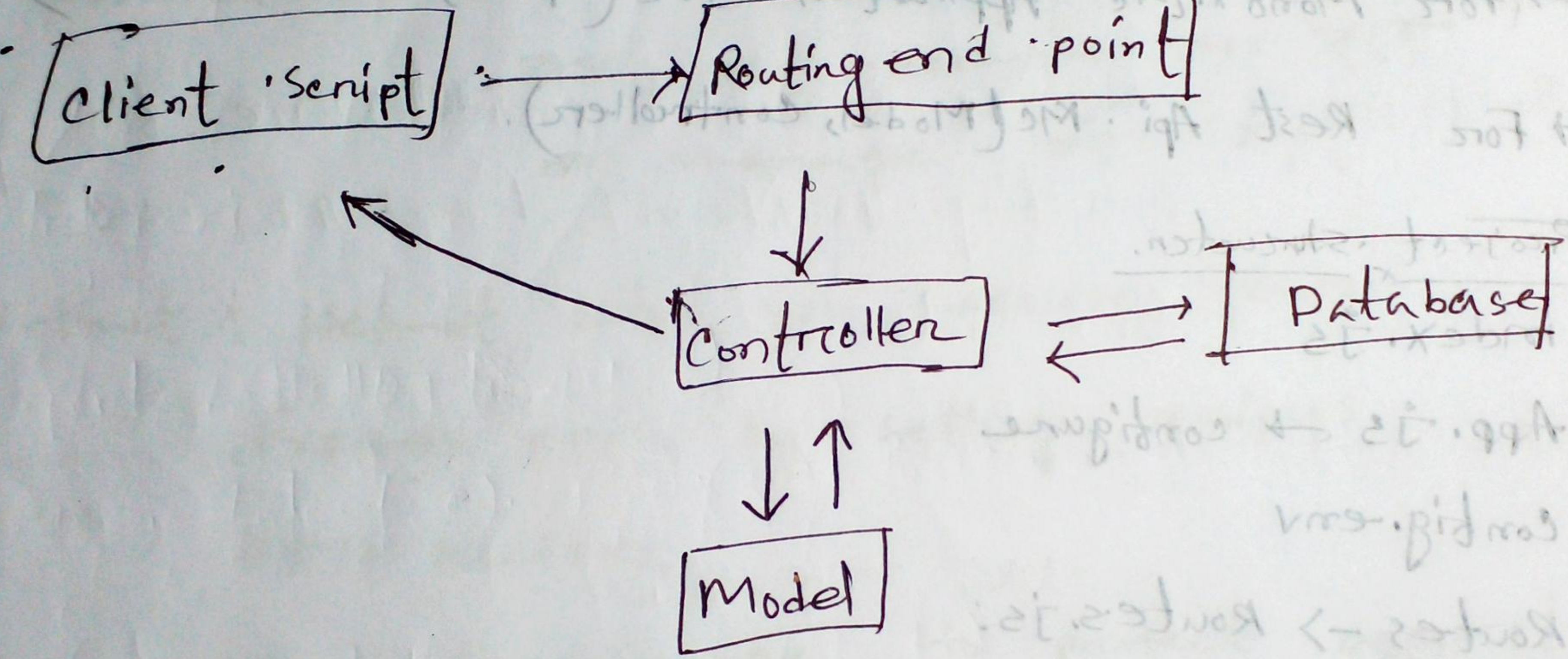
→ config.env

→ Routes -> Routes.js:

→ Controller -> Controller.js

→ Models -> Model.js

Project structer :-



Create My Express Rest Api Project Structer.

1. create a project folder.
2. Create Package.json
3. Install express
4. Install node package.
5. index.js
6. app.js
7. Create config.env
8. Create src directory
9. Create src -> controllers folder.
10. Create — src -> models folder.
11. Create src -> routes folder
12. " src -> middleware
13. " src -> helper.