**Application Object**

Ajke amra express js er application object ta niye kaj korbo. Express module ke require kore ene jokhon call kore dei tokhon ekta application ba app object pai. Ei app object tar moddhe ki ki methods or properties ache shegula niyei ajke kotha bolbo.

App.locals

First property holo app.locals. eta local variable application er moddhe save kore rakhe.

Index.js …

*const* express = *require*("express");

*const* app = *express*();

app.locals.title = "My App";

app.*get*("/", (*req*, *res*) *=>* {

  console.*log*(app.locals.title);

  console.*log*("This is home page.");

});

app.*listen*(3000, () *=>* {

  console.*log*("Listening on port 3000");

});

App.locals ekta object. Etar ekta property ‘title’ set kore dilam. Ei app.locals.title amader entire application er jekono jaygay access korte parbo. Eta amader application er local variable hoye gese. Nodemon index kore postman e giye get method e localhost:3000 hit korle console e “My App” pacchi.

Entire application mane index.js charao jekono external file eo app.locals.title access korte parbo. For e.g. ami app.get() er callback function ta ami bairer onno ekta file theke import kore anbo.

Handle.js …

*const* *handle* = (*req*, *res*) *=>* {

  console.*log*(req.app.locals.title);

  res.*send*("This is HOME page !");

};

module.exports = handle;

Index.js …

*const* express = *require*("express");

*const* handle = *require*("./handle");

*const* app = *express*();

app.locals.title = "My App";

app.*get*("/", handle);

app.*listen*(3000, () *=>* {

  console.*log*("Listening on port 3000");

});

Handle.js name ekta file create kore handle function ta likhe fellam. Then shetake export kore index.js e ene app.get() er callback banaya dilam. Ekhane app.locals.title req object er property hoye jay.

App.mountpath

The app.mountpath property contains one or more path patterns on which a sub-app was mounted.

Amra index.js e express() kore ekta app nisi. Amra chaile amader project e multiple app rakhte pari.

Index.js …

*const* express = *require*("express");

*const* app = *express*();

*const* admin = *express*();

app.*get*("/", (*req*, *res*) *=>* {

  res.*send*("This Home page...");

});

app.*listen*(3000, () *=>* {

  console.*log*("Listening on port 3000");

});

Amra duita express() function call korlam. Ekhane admin o ek dhoroner app. But ‘app’ ebong ‘admin’ duita completely separate object. Cz amra jani object reference type jinish. Amra chaile duita app ekta single projecter moddhei rakhte pari. For e.g. amra ekta project e frontend ebong backend ache. Front end er route gula ek rokom. ‘/admin’ er porer route gula sob backend.

Ekhon app.use(“/admin”, admin) kore bole dite pari kono client “/admin” url e hit korle take admin object ba app e pathiye deya hobe.

Index.js …

*const* express = *require*("express");

*const* app = *express*();

*const* admin = *express*();

app.*use*("/admin", admin);

app.*get*("/", (*req*, *res*) *=>* {

  res.*send*("This Home page...");

});

admin.*get*("/dashboard", (*req*, *res*) *=>* {

  console.*log*(admin.mountpath);

  res.*send*("Welcome to admin dashboard ");

});

app.*listen*(3000, () *=>* {

  console.*log*("Listening on port 3000");

});

Ei admin app er jonno admin.get() call kora hoise. So keu jodi “localhost:3000/admin/dashboard” --- ei url e hit kore tahole “Welcome to admin dashboard” pabe. Abar admin.mountpath property “/admin” return korche. Admin app er root path ta return korche.

Onno dike app.get() diye normal “localhost:3000/” url er response set kora ache.

Mount Event

The mount event is fired on a sub-app, when it is mounted on a parent app. The parent app is passed to the callback function.

Jokhon ami kono parent app e thakbo, oi parent app e thaka obosthay jodi kono sub app mounted hoy tokhoni mount event ta fire hobe.

Index.js …

*const* express = *require*("express");

*const* app = *express*();

*const* admin = *express*();

admin.*on*("mount", (*parent*) *=>* {

  console.*log*("Admin Mounted");

  console.*log*(parent);

});

admin.*get*("/dashboard", (*req*, *res*) *=>* {

  console.*log*(admin.mountpath);

  res.*send*("Welcome to admin dashboard ");

});

app.*get*("/", (*req*, *res*) *=>* {

  res.*send*("This Home page...");

});

app.*use*("/admin", admin);

app.*listen*(3000, () *=>* {

  console.*log*("Listening on port 3000");

});

Ekhon nodemon index kore server on korlei parent app ta console e dekhte parbo. Parent mane hocche main ‘app’.

Jokhoni subapp ta parent app er moddhe mount hobe tokhoni ‘mount’ event ta fire hobe. Ebong parent app ta event er callback function er moddhe passed hobe.

Ekhon different types of methods dekhbo.

App.all()

Index.js ….

*const* express = *require*("express");

*const* app = *express*();

app.*get*("/", (*req*, *res*) *=>* {

  res.*send*("Welcome to app home...");

});

app.*listen*(3000, () *=>* {

  console.*log*("Listening on port 3000");

});

Ekhon get method e localhost:3000 e hit korle o/p pai. But put ba post ba onno method e hit korle vul o/p ashe. But code e app.get() er bodle app.all() use korle jekono method e hit kori na keno o/p ashbe “Welcome to app home…”

App.disable()

App.disable() app er kono ekta setting ke disable kore dey. app.disable(name) --- ekhane name ki? Express er application er ekta settings table ache. Shekhane bivinno setting define kora ache. Ei setting gula ke app.disable() ebong app.enable() diye disable ba enable kora jay.

Express e case sensitivity by default off thake. Ekhon eta ke amra on ba enable korbo.

Index.js …

*const* express = *require*("express");

*const* app = *express*();

app.*enable*("case sensitive routing");

app.*disable*("case sensitive routing");

app.*all*("/home", (*req*, *res*) *=>* {

  res.*send*("Welcome to app home...");

});

app.*listen*(3000, () *=>* {

  console.*log*("Listening on port 3000");

});

Ekhane first e enable korlam shekhtre postman e ‘localhost:3000/Home’ likhe hit korle o/p pacchilam na. then disable korlam. o/p pelam.

App.get()

App.get() amra already use korsi.

App er kono kichu ekta set kore sheta jodi get korte chai tahole app.get() use korte pari. Set korar jonno app.set() ache.

App er jekono setting ba amar custom setting set korar jonno app.set() use korbo. App.get() diye sheta access korte parbo.

Index.js …

*const* express = *require*("express");

*const* app = *express*();

app.*set*("food", "pizza");

app.*all*("/", (*req*, *res*) *=>* {

  console.*log*(app.*get*("food"));

  res.*send*("Welcome to app home...");

});

app.*listen*(3000, () *=>* {

  console.*log*("Listening on port 3000");

});

Console e ‘pizza’ asche.

App.param()

Amader route e bivinno parameter thakte pare jemonta amra react router dom practice korar shomoy dekhsilam. For e.g. “localhost:3000/user/5” ---- ekhane 5 no user er info pete chacchi. Ekhane 5 hocche parameter. So etar jonno app.get() er path ta hobe emon “/user/:id” orthat ami ‘id’ diye parameter ta ke define korlam. App.param() kono ekta particular param name er upor base kore kono ekta kaj kore. Majhkhane boshe jay.

Index.js …

*const* express = *require*("express");

*const* app = *express*();

app.*param*("id", (*req*, *res*, *next*, *id*) *=>* {

*const* user = {

    userId: id,

    person: "kalim",

  };

  req.userDetails = user;

*next*();

});

app.*get*("/user/:id", (*req*, *res*) *=>* {

  console.*log*(req.userDetails);

  res.*send*("Welcome to user page...");

});

app.*listen*(3000, () *=>* {

  console.*log*("Listening on port 3000");

});

Ekhane amar param er name holo ‘id’. So eta app.param() er first parameter e dite hobe. Then ekta callback function ba middleware dite hobe. So express route er moddhe jodi ‘id’ namer kono param pay tahole tar jonno ei callback function ta call kore. Ei callback function er moddhe amra chaile req object modify korte pari, database theke kono ekta data ene ekta kaj korte pari. Ekhane lokkhonio app.param() er callback function ta app.get() er function er age execute hoy. So app.param() er CB ta ekta middleware. Middleware is kinda interceptor. Orthat requester jawar age kichu ekta kora. Kore app.get() e pathay deya. Ekarone middleware er last e ekta next function call kore dite hoy. Next() call korar ortho middleware er kaj shesh. Ekhon porer ba ashol jaygay jao.

App.route()

Jodi amader route path ta boro complex kichu hoy ebong ei eki route path e bivinno method e hit korte hoy tahole route path ta bivinno method er jonno bar bar na likhe ekbar likhar system kore dey app.route().

Index.js …

*const* express = *require*("express");

*const* app = *express*();

app

  .*route*("/myapp/home")

  .*get*((*req*, *res*) *=>* {

    res.*send*("Welcome to home get.");

  })

  .*post*((*req*, *res*) *=>* {

    res.*send*("Welcome to home post.");

  })

  .*put*((*req*, *res*) *=>* {

    res.*send*("Welcome to home put");

  })

  .*delete*((*req*, *res*) *=>* {

    res.*send*("Welcome to home delete.");

  });

app.*listen*(3000, () *=>* {

  console.*log*("Listening on port 3000");

});

App.engine()

Ekhon ekta template engine install korbo. Etokkhon amra rest api er moto json response dicchilam ba text ba raw kono response dicchilam. Kintu ekhon ekta template response dibo. Mane kono ekta html. Amader server theke onek shomoy html o return korte hoy jodi amra express js ke ekta MVC framework hishebe use korte chai.

Npm install ejs kore ejs name ekta template engine install kortesi.

Index.js …

*const* express = *require*("express");

*const* app = *express*();

app.*set*("view engine", "ejs");

app.*get*("/about", (*req*, *res*) *=>* {

  res.*render*("pages/about");

});

app.*listen*(3000, () *=>* {

  console.*log*("Listening on port 3000");

});

About.ejs ….

<!DOCTYPE *html*>

<html *lang*="en">

  <head>

    <meta *charset*="UTF-8" />

    <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge" />

    <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0" />

    <title>Document</title>

  </head>

  <body>

    <div>Hello, I am Turza</div>

  </body>

</html>

Ekhon ejs template engine ta ke amader application e attach korar jonno index.js e app.set() er maddhome ‘view engine’ setting ta ke update korlam. Ekhon app.get() er moddhe res.send na diye res.render() use korbo. Ebong shetar moddhe ekta random template file er path ‘pages/about’ diye dibo. Ekhetre express by default amader project e views name ekta folder er moddhe pages name ekta folder er moddhe about.ejs name ekta template file khujbe. So ami nije theke agei nirdishto folder er moddhe nirdishto file create korlam. Ekhon server start kore browser e localhost:3000/about url e hit korle about.ejs content dekhte pacchi.