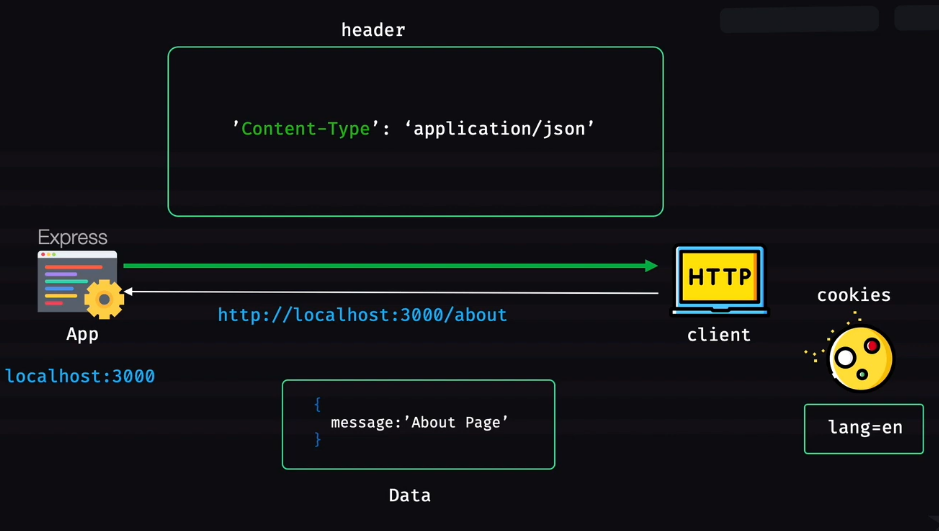
**Response Object**

Ajke amra response object niye discuss korbo. Response jinistha ashole ki?



Ekhane yellow iconta ekta http client (browser, mobile app, postman etc). ar left side e express application (jeta ashole server) localhost 3000 port e run korche. Client amader express app e ‘http://localhost:3000/about’ url e hit kore request korlo. Ekhon express app er task hocche ei request er upor kichu ekta kore ekta shothik response akare http client ke back deya. Request pawar shomoy jerokom bivinno header request er shathe chole ashe request er meta data hishebe ba request er characteristic hishebe. Eki vabe response deyar shomoyo response besh kichu header automatcially create hoye jay. Shei shathe amra manually response er header kichu jinish add kore dite pari. Response er headerta read kore amar client. Request er header ta read kori amra orthat express application. Header er moddhe thake bivinno properties like host, content-type etc. echara request er response deyar shomoy express app response er shathe kichu data o diye dite pare. Ei data bivinno format e hote pare like json, html, txt. But je format ei jak onno pashe client ke sheta accept korte parte hobe. Response deyar shomoy server chaile http client (ekhetre browser) e ekta cookie set kore dite pare. For e.g. lang=en name ekta cookie set kore deya jay.

Response represents the http response.

Ekhon amra express js er docs theke shikhbo.

**Res.app**

It works the same way as req.app does.

**Res.locals**

Res.locals dekhar jonno amar ekta view engine ba template system thakte hobe.

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>About</title>

  </head>

  <body>

    This is about page.</body></html>

‘views’ name ekta folder khule tate ‘pages’ name arekta folder khule about.ejs name ekta file niye kichu likhlam.Nodemon index kore localhost:3000/about url e hit korlam get method e. html ba ejs file er lekhata show kortese.

Ekhane index.js hocche server part. Ar about.ejs hocche view part. Amar controllergula ache index.js e. ekhon index.js theke amra jodi about.ejs e kono data pathate chai tokhon amra res.locals ta use korbo. Shekehtre amra res.render() er 2nd parameter ta use korbo. 2nd parameter e ekta object pathabo. Object e ja pathabo shetai amar oipashe locals hishabe use korte parbo.

Index.js …

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

*const* app = *express*();

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

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

  res.*render*("pages/about", {

    country: "Bangladesh",

  });

});

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>About</title>

  </head>

  <body>

    This is about page. <%= country %>

  </body>

</html>

Je object ta pathalam sheta res.locals er moddhe thake.

**Res.headersent**

Index.js …

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

*const* app = *express*();

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

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

  console.*log*(res.headersSent);

  res.*render*("pages/about", {

    country: "Bangladesh",

  });

  console.*log*(res.headersSent);

});

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

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

});

Ekhane res.render() kore ekta response pathay disi. Ekhon render er age ebong pore res.headersSent er value dekhlam. 1st er ta ashlo false. 2nd ta ashlo true. 1st time false ashlo cz oi shomoy header sent kora hoy nai. Tarpor response pathanor por res.headerSent er value ashlo true. Orthat res.headerSent er kaj hocche response pathano hoise kina check kora. Eki file e achi bole nije dekhte pacchi response pathano hoise kina. But real life amader application onek choto choto module e vag kora thakbe. kono ekta module jodi kono ekta jayga theke response pathaya dey ebong arekta module theke jodi amar check kora dorkar hoy onno module theke response chole geche kina tokhon ei jinishta amra use korte pari.

Ekhon res object er kichu method dekhbo.

**Res.end()**

Res.send() ebong res.end() er moddhe difference hoche send er shomoy ami ekta data dite pari. Ar res.end() deya mane response ta close kore deya.

Res.send() holo with data response shesh kore deya.

Res.end() holo without data response shesh kore deya.

**Res.json()**

Jehetu rest api er khetre json response pathanota khub common shehetu etar maddhome amra object kingba array jekono jinish dite pari. Json.stringify kora jay emon shob jinish dite pari.

Index.js …

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

*const* app = *express*();

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

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

  res.*json*({

    food: "chicken burger",

  });

});

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

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

});

Ekhane res.json er data hishebe ekta object disi. Ekhon server start kore localhost:3000/about url e get method e hit korle postman er response section e {“food”: “chiken burger”} json type er data dekhte pacchi. Eki shathe response er header content-type ‘application/json’ set kore dise. Orthat amar client er kache response ashche application/json. So res.json() duta kaj kortese. 1. Response er header e application/json kore ditese. 2. Ami je object disilam sheta ke json.stringify kore ultimately response diye dicche.

**Res.status()**

Ami jokhon server theke client er kache response pathacchi tokhon amar bivinno shomoy status pathate hote pare (error hole 400, server error hole 500, forbidden hole 403 etc). res.status() call korle response shesh hoy na. so res.status() er por amake res.end() kortei hobe. Res.status() diye muloto response er status ke set kora hoy. Ekhon ami jodi ekta method diyei status set korte chai ebong response end o kore dite chai tahole res.sendStatus() use korte hobe. For e.g. res.sendStatus(400) call korle client ke 400 status pathabe ebong 400 er corresponding header value (which is problem in request) pathabe.

Index.js

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

*const* app = *express*();

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

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

  res.*sendStatus*(400);

});

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

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

});

Server on kore url e get method e hit korle postman er response section e ‘bad request’ lekha ashtese. Ebong dan pashe upore status code 400 show kortese.

Res.sendStatus(200) pathale ‘ok’ lekha ashtese. Ei method ta status discche, response end korche ebong shetar corresponding text pathay ditese.

**Res.format()**

Res.format() request er shathe ekta content negotiation kore. Request er shathe jodi kono rokom accept header ashe like request e jodi client bole dey she omok dhoroner content-type accept kore ebong server end e jodi res.format() use kori she automatically content negotiate korbe. kore dekhbe client ki chacche ebong she onujayi response dibe.

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

*const* app = *express*();

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

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

  res.*format*({

    "text/plain": () *=>* {

      res.*send*("Hello plain text");

    },

    "text/html": () *=>* {

      res.*render*("pages/about", { country: "Bangladesh" });

    },

    "application/json": () *=>* {

      res.*json*({ message: "About postman" });

    },

*default*: () *=>* {

      res.*status*(406).*send*("Not acceptable");

    },

  });

});

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

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

});

Ekhane res.format() er parameter hishabe ekta object dibo. Shei object er ekekta property hobe ekekta content-type. Propertygular value hobe ekta kore function. Shei functiongular moddhe response send ba render korbo. ami ekta default response o pathate parbo. Jodi kono content-type match na kore tahole ei default response jabe.

Jokhoni client theke request ashbe server request er header er ‘Accept’ check korbe. jodi ‘Accept’ mention kora thake ebong amar res.format() er object er property gular moddhei kono ekta hoy tahole she correspondingly response dibe. Jodi ‘Accept’ e \*/\* thake tahole Prothom property ta kaj korbe. konotai match na korle fallback hishabe default ta kaj korbe. jodi header e ‘accept’ na thake taholeo first property ta kaj korbe.

**Res.cookie()**

Res.cookie() er first parameter e cookie name ebong 2nd parameter e value gula dite parbo.

Index.js …

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

*const* app = *express*();

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

  res.*cookie*("cheese", "cheddar");

  res.*end*();

});

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

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

});

Res.cookie set kore res.end kore dilam . server on kore url e hit korle postman er request ebong response dui jaygate ei cookie ta dekhte pacchi.

**Res.location()**

Index.js…

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

*const* app = *express*();

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

  res.*location*("/test");

  res.*end*();

});

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

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

});

Server on kore url e hit korle postman er response section er header e location name ekta property pacchi jar value holo ‘/test’

**Res.redirect()**

Index.js …

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

*const* app = *express*();

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

  res.*send*("This is redirected test page.");

});

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

  res.*redirect*("/test");

  res.*end*();

});

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

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

});

Server on kore localhost:3000/about url e hit korlam. Er fole amar route ta redirected hoye ‘/test’ route e chole gelo. /test route e giye ‘this is redirected….’ Print korlo.

**Res.get() & res.set()**

Index.js …

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

*const* app = *express*();

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

  res.*set*("Moncho", "ADcodes");

  console.*log*(res.*get*("Moncho"));

  res.*end*();

});

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

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

});

/about url e hit kore response header e giye dekhalam ‘Moncho’ name ekta property ache. Echara console e ‘ADcodes’ lekha uthe ache.