

The background of the image is a spiral-bound notebook with a light beige, textured cover. The spiral binding is visible on the left side, with the metal wire looping through a series of holes. The text is centered on the right side of the notebook cover.

Web App Development in Node.js and Express

CSE 264
J. Femister

Node.js

- Execution environment for Javascript on the server.
- Based on the V8 engine used in Chrome
- Single threaded
- Asynchronous event handling model
- Fast running and scalable
- Extended through modules
- Used mostly for creating web apps

Express

- Framework for web apps – builds on Node
- Implemented as Node module
- Unopinionated
- Offers:
 - Middleware
 - Routing
 - Subapplications
 - Conveniences (additions to req and res)

Using Express

- Installation
 - `npm install express`
- Load module into app.js
 - `var app= require("express")`

Setup Logging

- Install Morgan
 - `npm install morgan`
- Load the Morgan module
 - `var logger = require("morgan");`
- Register Morgan with Express
 - `app.use(logger("dev"));`

Setup Templates

- Install ejs
 - `npm install ejs`
- Setup Express to use EJS for views
 - `app.set("views",
path.resolve(__dirname, "views"));`
 - `app.set("view engine", "ejs");`
- Render Page
 - `res.render("index");`

Add Middleware

- A request is processed by each middleware function until a response is sent to the browser.

```
app.use( (req, res, next) => {  
  if (req.url === "/") {  
    next();  
  } else {  
    res.send("Must use the root!");  
  }  
});
```

- Note: Used app.use for logging and b-p

Setup Body Parsing

- Install body-parser
 - `npm install body-parser`
- Load module
 - `var bodyParser = require("body-parser");`
- Register with Express
 - `app.use(bodyParser.urlencoded({
extended: false }));`

Add Routing

- Routing matches a request method and url and processes it with a function:

```
app.get("/new-entry-pg", function(req, res) {  
  // process get request for new entry page  
});
```

```
app.post("/new-entry", function(req, res) {  
  // process post request for new entry  
});
```

Routing Examples

- `app.get("/simple/path", (req, res) => { ... })`
- `app.get("/users/:userid", (req, res) => {
 var userid =
 parseInt(req.params.userid); })`
- `app.get(/^\/users\/(\d+)$/, (req, res) => {
 var userid = parseInt(req.params[0]); })`
- `app.get("/search", (req, res) => {
 if (req.query.userid == "123") ... })`

Serving Static Files

- `app.use(express.static(path.resolve(__dirname, "public")));`
- Note that this is just another example of using middleware.

Start App

- Start App listening on port 3000 for HTTP requests:

```
app.listen(3000, () => {  
  console.log("App started");  
});
```