Express

- Express is an abstraction layer on top of Node’s http-server
  - Similar to JQuery is to JavaScript
- Express simplifies implementation of tasks that otherwise will require significant effort using the **http** module
- What Express provides
  - **Extensions** - The basic request and response objects have extra functionality
  - **Middleware** – Instead of a single function handling the requests a stack of functions (middleware stack) is available. This allows organizing the processing in separate functions
  - **Routing** – Routing allow us to associate an URL and a HTTP method with some functionality
  - **Views** – Dynamic generation of HTML

Creating a Project in Node

- A Node project has a file called **package.json** providing information such as project's name, author, version, and dependencies (which modules your project relies on)
- You can create this file yourself or you can rely on **npm** init
- Example: Let's create a project
- To install Express and save it as dependency to package.json
  - npm install express –save
- After installing you will see a directory called **node_modules**
- **Example:** example1.js

Middleware

- Middleware is a function
- In Node a single function processes the request; using middleware the request can be processed by several functions.
- For example:
    - One function can do authentication
    - One function can do logging
- A request does not need to be processed by every middleware function (any of them could provide a response). If none provides a response the server will hang
- A middleware function can modify the request or response
- In **app = express()**, app is a function that goes through the set of functions that are part of middleware stack
- **app.use** allow us to add middleware functions to the middleware stack
- **Example:** middleware.js

Logger example

- We can log requests using a third party logger
- Installing morgan
  - npm install morgan –save
- writeHead is used with text/html
- **Example:** loggingHTML.js

Serving Static Files

- express.static – part of Express
  - Allow us to serve files
- path
  - built-in module we use to generate a cross-platform (Windows, Mac, Linux) path
- **Example:** servingFiles.js

Additional Functionality to request/response

- **Express** expands the request and response objects
- request.ip → ip address
- request.get → to obtain HTTP headers
- request.status → to set status code
- request.send
- response.redirect
  - Redirects to a particular site
- response.sendFile
  - To send a file
- response.json → sending s JSON response
- **Example:** additionalFunc.js

HTTP Verbs/Methods

- An HTTP request has a method/verb associated with it
- HTTP Methods
  - **GET**
    - Gets a resource
    - Most common method used
    - Idempotent (executing many times does not cause server change)
  - **POST**
    - Generates a change of server state (e.g., you bought an item)
    - Non-idempotent
  - **PUT**
    - To update or change
    - Idempotent
  - **DELETE**
    - To remove a resource
    - Idempotent
  - **PATCH**
    - Relatively new
    - Can be use to update

HTTP Verbs/Methods

- You can use **Express** to handle different HTTP verbs
- Download the **curl** application so you are able to generate http requests with different methods/verbs
  - https://curl.haxx.se/download.html
- **Example:** httpMethods.js
- You can issue the requests as follows
  - GET → curl http://localhost:8001
  - POST → curl –X POST http://locahost:8001
  - PUT → curl –X PUT http://locahost:8001
  - DELETE → curl –X DELETE http://locahost:8001

Routing

- Routing - Mapping an URI and HTTP verb to a request handler
- In **Express** you specify routes using strings and can specify them as regular expressions
- **Example:** routing.js

Dynamic Generation of HTML

- View/templating engines – Allow you to generate dynamic HTML
- EJS (Embedded JavaScript) engine is a templating engine that compiles/generates HTML for you
- EJS is a superset of HTML
- Files with the .ejs extension are placed in a folder where Express can locate them.
- To install ejs
  - npm install ejs --save
- Interpolate variables in template file by using:

  <%= variableName %>
- Inclusion of ejs file in another by using:

  <% fileNameWithoutEJSExtension %>   // Notice no = in <%
- **Example:** dynamicHTML.js, templates/welcome.ejs

Retrieving Query Arguments

- We can use request.query.<ARGUMENT_NAME> to retrieve arguments provided in the URL
- **Example:** formGet.html, queryArguments.js, templates/courseInfo.js

Retrieving values associated with POST

- The body-parser module allows you to retrieve parameters submitted using post by using request.body.<PARAMETER_NAME>
- To install
  - npm install body-parser --save
- **Example:** formPost.html, postParameters.js, templates/courseInfo.js

References

- Express in Action

  Writing, building, and testing Nodes.js applications

  Evan M. Hahn

  April 2016 , ISBN 9781617292422