

44-563: Unit 14

Developing Web Applications and Services

Includes

- Review of Code Organization
- Client Review Visit Friday
- Client Final Presentations Next Friday
- See Finals Schedule in Course Site

Review: Code Organization

Open the reference project in VS Code.

Where:

1. Is the sample **data**?
2. Are the **models** that define the properties of all our resources?
3. Are the **controllers** that provide the logic to handle requests?
4. Are the **views** (dynamically-generated page parts)?
5. Is the public static content (unchanging content for client side HTML/CSS/JS)?
6. Do we **route** requests?
7. Do we specify the user authorization & authentication **config**?
8. Where do we keep **utils** (such as the seeder and logger?)

What is this? What extension?

- config
- controllers
- data
- logs
- models
- node_modules
- public
- routes
- utils
- views
- ⚙ .env
- ≡ .env.example
- ⦿ .eslinttrc.json
- 📄 .gitignore
- JS app.js**
- 📄 LICENSE.md
- {} package-lock.json
- {} package.json
- 📖 README.md

```
{ } puppies.json x
1  [
2    {
3      "_id": 1,
4      "name": "Lexie",
5      "breed": "Cavachon",
6      "age": 1,
7      "parents": [
8        {
9          "parentName": "Mom",
10         "parentBreed": "Bichon Frise",
11         "parentAge": 5
12       },
13       {
14         "parentName": "Dad",
15         "parentBreed": "Cavalier King Charles",
16         "parentAge": 6
17       }
18     ]
19   },
```

What is this? What extension?

- config
- controllers
- data
- logs
- models
- node_modules
- public
- routes
- utils
- views
- ⚙ .env
- ≡ .env.example
- ⦿ .eslintrc.json
- 📄 .gitignore
- JS app.js
- 📄 LICENSE.md
- {} package-lock.json
- {} package.json
- 📖 README.md

```
JS puppy.js x
1  const mongoose = require('mongoose')
2
3  const PuppySchema = new mongoose.Schema({
4    _id: { type: Number, required: true },
5    name: {
6      type: String,
7      required: true,
8      default: 'Lexie'
9    },
10   breed: {
11     type: String,
12     required: false,
13     default: 'Cavachon'
14   },
15   age: {
16     type: Number,
17     required: false,
18     default: 1
19   },
20   parents: {
21     type: Array,
22     required: false,
23     default: [{ parentName: 'Mom', parentBreed: 'Bichon Frise', parentAge: 5 },
24               { parentName: 'Dad', parentBreed: 'Cavalier King Charles', parentAge: 6 } ]
25   }
26 })
27 module.exports = mongoose.model('Puppy', PuppySchema)
```

What is this? What extension?


- config
- controllers
- data
- logs
- models
- node_modules
- public
- routes
- utils
- views
- ⚙ .env
- ≡ .env.example
- ⚙ .eslintrc.json
- 📄 .gitignore
- JS app.js**
- 📄 LICENSE.md
- {} package-lock.json
- {} package.json
- 📖 README.md

```
JS index.js x
1  /**
2   * @index.js - manages all routing
3   *
4   * router.get when assigning to a single request
5   * router.use when deferring to a controller
6   *
7   * @requires express
8   */
9
10 const express = require('express')
11 const LOG = require('../utils/logger.js')
12
13 LOG.debug('START routing')
14 const router = express.Router()
15
16 // Manage top-level request first
17 router.get('/', (req, res, next) => {
18   LOG.debug('Request to /')
19   res.render('index.ejs', { title: 'Puppies!' })
20 })
21
22 // Defer path requests to a particular controller
23 router.use('/about', require('../controllers/about.js'))
24 router.use('/puppy', require('../controllers/puppy.js'))
25
26 LOG.debug('END routing')
27 module.exports = router
28
```

- config
- controllers
- data
- logs
- models

What is this? What extension?

```
js logger.js x
1  const winston = require('winston')
2  const fs = require('fs')
3  const path = require('path')
4
5  const logDir = 'logs'
6
7  if (!fs.existsSync(logDir)) {
8    fs.mkdirSync(logDir)
9  }
10
11 const logger = new (winston.Logger)({
12   level: 'debug',
13   transports: [
14     new (winston.transports.Console)({ json: false, timestamp: true }),
15     new winston.transports.File({ filename: path.join(logDir, '/debug.log'), json: false })
16   ],
17   exceptionHandlers: [
18     new (winston.transports.Console)({ json: false, timestamp: true, humanReadableUnhandledException: true }),
19     new winston.transports.File({ filename: path.join(logDir, '/debug.log'), json: false, humanReadableUnhandledException: true })
20   ],
21   exitOnError: false
22 })
23
24 module.exports = logger
```

 package-lock.json

 package.json

 README.md

What is this? What extension?

- config
- controllers
- data
- logs
- models
- node_modules
- public
- routes
- utils
- views
- ⚙ .env
- ≡ .env.example
- ⦿ .eslintrc.json
- 🔒 .gitignore
- JS app.js**
- 📄 LICENSE.md
- {} package-lock.json
- {} package.json
- 📖 README.md

```
JS seeder.js x
1 // set up a temporary (in memory) database
2 const Datastore = require('nedb')
3 const LOG = require('../utils/logger.js')
4 const puppies = require('../data/puppies.json')
5 const users = require('../data/users.json')
6 const userController = require('../controllers/user')
7
8 module.exports = (app) => {
9   LOG.info('START seeder.')
10   const db = {}
11   db.puppies = new Datastore()
12   db.users = new Datastore()
13
14   db.puppies.loadDatabase()
15   db.users.loadDatabase()
16
17   // insert the sample data into our data stores
18   db.puppies.insert(puppies)
19   db.users.insert(users.data)
20
21   // initialize app.locals (these objects will be available to our controllers)
22   app.locals.puppies = db.puppies.find(puppies)
23   LOG.debug(`${puppies.length} puppies seeded`)
24
25   app.locals.users = db.users.find(users)
26   LOG.debug(`${users.length} users seeded`)
27
28   users.forEach((user) => {
29     userController.newUser(user)
30   })
31
32   LOG.info('END Seeder. Sample data read and verified.')
33 }
```

What is this? What extension?

- config
- controllers
- data
- logs
- models
- node_modules
- public
- routes
- utils
- views
- ⚙ .env
- ≡ .env.example
- ⚙ .eslintrc.json
- 📄 .gitignore
- JS app.js**
- 📄 LICENSE.md
- {} package-lock.json
- {} package.json
- 📖 README.md

```
JS puppy.js x
1  const express = require('express')
2  const api = express.Router()
3  const Model = require('../models/puppy.js')
4  const LOG = require('../utils/logger.js')
5  const find = require('lodash.find')
6  const remove = require('lodash.remove')
7  const notfoundstring = 'puppies'
8  const passport = require('../config/passportConfig.js')
9
10 // RESPOND WITH JSON DATA -----
11
12 // GET all JSON
13 api.get('/findall', passport.isAuthenticated, (req, res) => {
14   res.setHeader('Content-Type', 'application/json')
15   const data = req.app.locals.puppies.query
16   res.send(JSON.stringify(data))
17 })
18
19 // GET one JSON by ID
20 api.get('/findone/:id', passport.isAuthenticated, (req, res) => {
21   res.setHeader('Content-Type', 'application/json')
22   const id = parseInt(req.params.id, 10) // base 10
23   const data = req.app.locals.puppies.query
24   const item = find(data, { _id: id })
25   if (!item) { return res.end(notfoundstring) }
26   res.send(JSON.stringify(item))
27 })
```

What is this? What extension?

- config
- controllers
- data
- logs
- models
- node_modules
- public
- routes
- utils
- views
- ⚙ .env
- ≡ .env.example
- ⦿ .eslintrc.json
- 📄 .gitignore
- JS app.js**
- 📄 LICENSE.md
- {} package-lock.json
- {} package.json
- 📖 README.md

```
29 // RESPOND WITH VIEWS -----
30
31 // GET to this controller base URI (the default)
32 api.get('/', passport.isAuthenticated, (req, res) => {
33   res.render('puppy/index.ejs')
34 })
35
36 // GET create
37 api.get('/create', passport.isAuthenticated, (req, res) => {
38   LOG.info(`Handling GET /create${req}`)
39   const item = new Model()
40   LOG.debug(JSON.stringify(item))
41   res.render('puppy/create.ejs',
42     {
43       title: 'Create puppy',
44       layout: 'layout.ejs',
45       puppy: item
46     })
47 })
```


views

▸ about

▸ account

▸ puppy

<> 404.ejs

<> about_details.ejs

<> about.ejs

<> contact.ejs

<> footer.ejs

<> header.ejs

<> index.ejs

<> layout.ejs

<> layout.ejs x

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="utf-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
7   <title>CGE</title>
8   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/boot
9     crossorigin="anonymous">
10  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/boot
11    crossorigin="anonymous">
12  <link rel="stylesheet" href="../../css/main.css">
13  <script src="https://code.jquery.com/jquery-3.1.1.slim.min.js">
14  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js
15    crossorigin="anonymous"></script>
16  <link rel="apple-touch-icon" sizes="180x180" href="../../images
17  <link rel="icon" type="image/png" sizes="192x192" href="../../i
18  <link rel="icon" type="image/png" sizes="32x32" href="../../ima
19  <link rel="manifest" href="../../images/manifest.json">
20  <meta name="msapplication-TileColor" content="#ffffff">
21  <meta name="msapplication-TileImage" content="../../images/ms-i
22  <meta name="theme-color" content="#ffffff">
23 </head>
24
25 <body>
26   <% include header.ejs %>
27   <div class="container">
28     <div class="body">
29       <%- body %>
30     </div>
31   </div>
32   <% include footer.ejs %>
33 </body>
34
35 </html>
```

What is this? What extension?

views

about

account

puppy

<> 404.ejs

<> about_details.ejs

<> about.ejs

<> contact.ejs

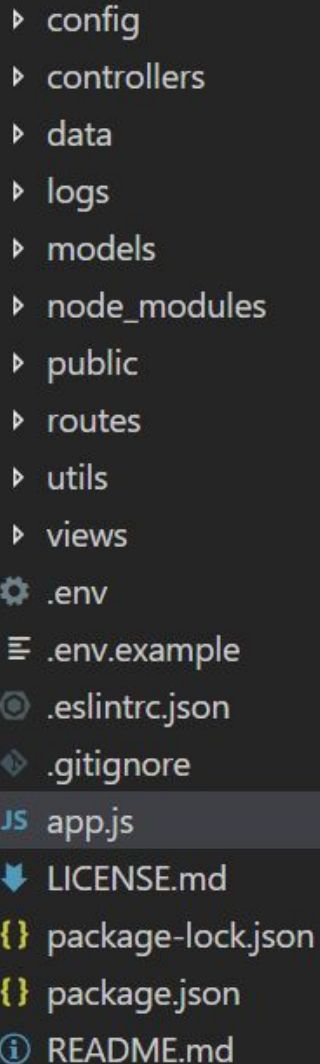
<> footer.ejs

<> header.ejs

<> index.ejs

<> layout.ejs

```
<> 404.ejs x
1  <br>
2  <br>
3  <div >
4    <h1>404</h1>
5    <p> Page Not Found </p>
6  </div>
7
```

- config
- controllers
- data
- logs
- models
- node_modules
- public
- routes
- utils
- views
- ⚙ .env
- ≡ .env.example
- 🔍 .eslintrc.json
- 📄 .gitignore
- JS app.js
- 📄 LICENSE.md
- {} package-lock.json
- {} package.json
- 📖 README.md

- All these may appear on the final.
- Understand the **roles** of the different types of files
- Recognize the **language** / **extension** the file should have
- Recognize what the code **does**
- Write small, specific, well-commented functions
- Understand basic control events

Sample final questions

```
api.get('/findall', function(request, response){  
  
    res.setHeader('Content-type', 'application/json');  
  
    var data = req.app.locals.estimates.query;  
  
    res.send(JSON.stringify(data));  
  
});
```

1. What is the purpose of this method?
2. What type of file is it in (M-V-C)?
3. What folder of the JCE does it go in?
4. What is the file extension of the file?
5. What is the method name? How many arguments are there?
6. What is the second argument?
7. The file has an error - why does it not work?
8. The file is signed. How would you fix it?

**Wed: Prepare for
Client Visit**

Plan for visit

1. Decide how to use your client visit.
2. Will you show in small groups, informally?
3. Will you have a few people lead the presentation - and then break into small groups?

Fri:
Client Visit