

Playlist 4 Review

- package.json
- README.md
- routes.js
- server.js

Routes

- controllers
 - about.js
 - accounts.js
 - dashboard.js
 - playlist.js
 - start.js

Controllers

- models
 - json-store.js
 - playlist-store.js
 - playlist-store.json
 - user-store.js
 - user-store.json

Models

- views
 - layouts
 - main.hbs
 - partials
 - addplaylist.hbs
 - addsong.hbs
 - listplaylists.hbs
 - listsongs.hbs
 - menu.hbs
 - welcomemenu.hbs
 - about.hbs
 - dashboard.hbs
 - index.hbs
 - login.hbs
 - playlist.hbs
 - signup.hbs
 - start.hbs

Views

routes.js

package.json
README.md
routes.js
server.js

▼ controllers
about.js
accounts.js
dashboard.js
playlist.js
start.js

▼ models
json-store.js
playlist-store.js
playlist-store.json
user-store.js
user-store.json

▼ views
▼ layouts
main.hbs

▼ partials
addplaylist.hbs
addsong.hbs
listplaylists.hbs
listsongs.hbs
menu.hbs
welcomemenu.hbs

about.hbs
dashboard.hbs
index.hbs
login.hbs
playlist.hbs
signup.hbs
start.hbs

```
'use strict';

const express = require('express');
const router = express.Router();

const start = require('./controllers/start');
const dashboard = require('./controllers/dashboard.js');
const playlist = require('./controllers/playlist.js');
const about = require('./controllers/about.js');
const accounts = require('./controllers/accounts.js');

router.get('/', accounts.index);
router.get('/login', accounts.login);
router.get('/signup', accounts.signup);
router.get('/logout', accounts.logout);
router.post('/register', accounts.register);
router.post('/authenticate', accounts.authenticate);

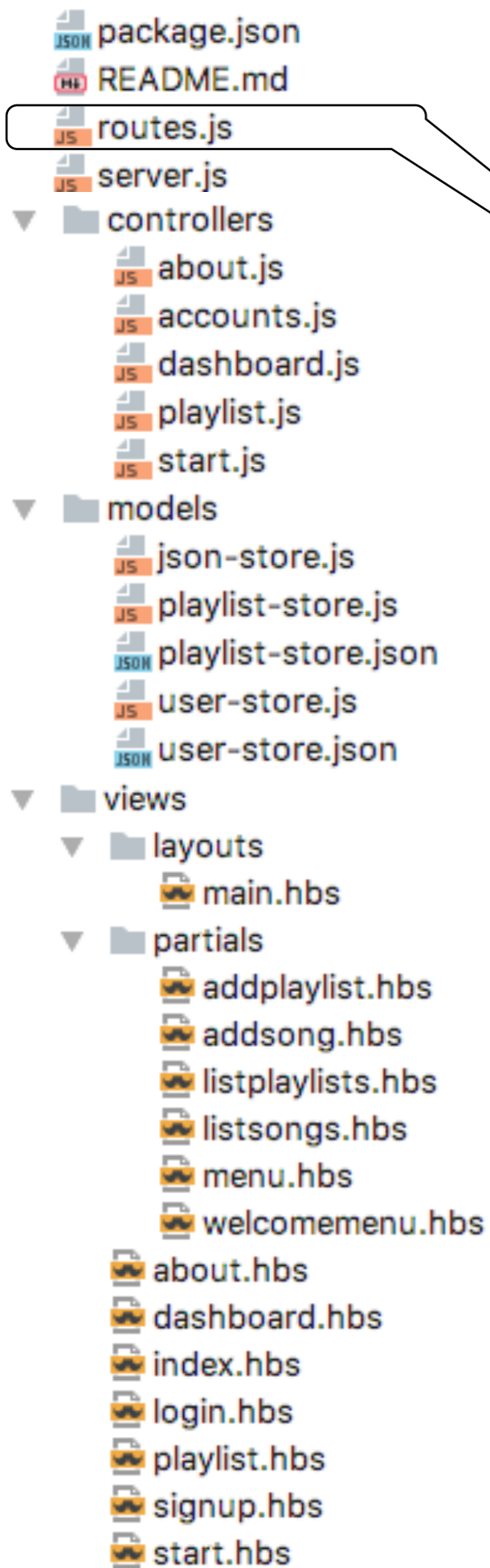
router.get('/dashboard', dashboard.index);
router.get('/dashboard/deleteplaylist/:id', dashboard.deletePlaylist);
router.post('/dashboard/addplaylist', dashboard.addPlaylist);

router.get('/playlist/:id', playlist.index);
router.get('/playlist/:id/deletesong/:songid', playlist.deleteSong);
router.post('/playlist/:id/addsong', playlist.addSong);

router.get('/about', about.index);

module.exports = router;
```

routes.js



```
'/' accounts.index
'/login' accounts.login
'/signup' accounts.signup
'/logout' accounts.logout
'/register' accounts.register
'/authenticate' accounts.authenticate

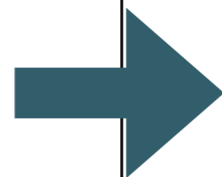
'/dashboard' dashboard.index
'/dashboard/deleteplaylist/:id' dashboard.deletePlaylist
'/dashboard/addplaylist' dashboard.addPlaylist

'/playlist/:id' playlist.index
'/playlist/:id/deletesong/:songid' playlist.deleteSong
'/playlist/:id/addsong' playlist.addSong

'/about' about.index
```

routes.js

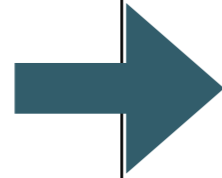
Accounts
Routes



```
'/'  
'/login'  
'/signup'  
'/logout'  
'/register'  
'/authenticate'
```

```
accounts.index  
accounts.login  
accounts.signup  
accounts.logout  
accounts.register  
accounts.authenticate
```

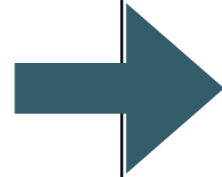
Dashboard
Routes



```
'/dashboard'  
'/dashboard/deleteplaylist/:id'  
'/dashboard/addplaylist'
```

```
dashboard.index  
dashboard.deletePlaylist  
dashboard.addPlaylist
```

Playlist
Routes



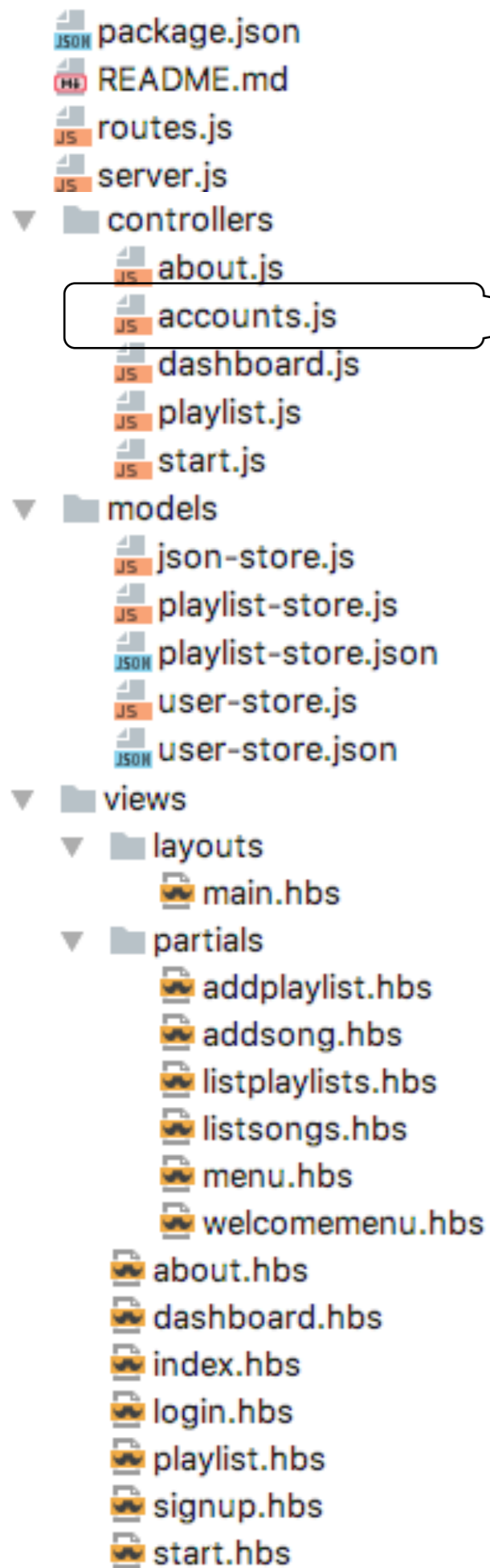
```
'/playlist/:id' playlist.index  
'/playlist/:id/deletesong/:songid'  
'/playlist/:id/addsong'
```

```
playlist.deleteSong  
playlist.addSong
```

```
'/about'
```

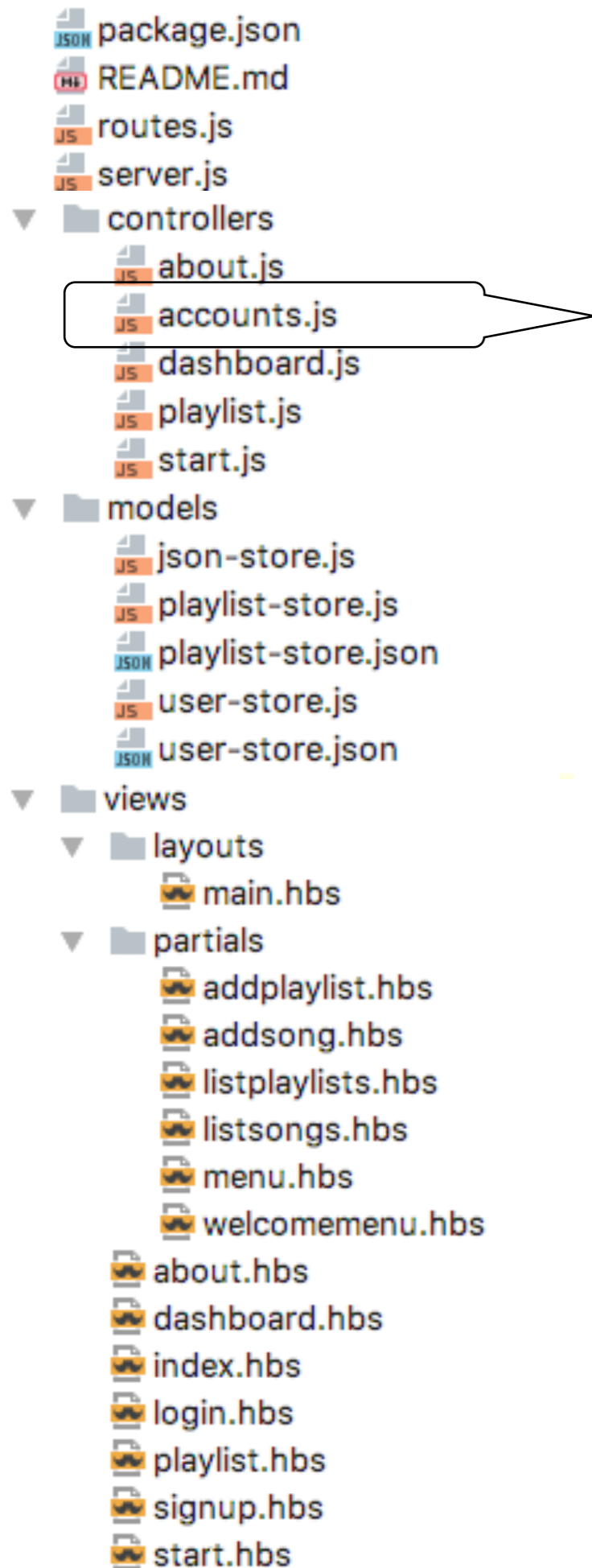
```
about.index
```

Accounts Routes



```
'/' accounts.index
'/login' accounts.login
'/signup' accounts.signup
'/logout' accounts.logout
'/register' accounts.register
'/authenticate' accounts.authenticate
```

Accounts Routes I



```
'/'  
'/login'  
'/signup'  
'/logout'
```

accounts.index
accounts.login
accounts.signup
accounts.logout

```
const accounts = {  
  
  index(request, response) {  
    const viewData = {  
      title: 'Login or Signup',  
    };  
    response.render('index', viewData);  
  },  
  
  login(request, response) {  
    const viewData = {  
      title: 'Login to the Service',  
    };  
    response.render('login', viewData);  
  },  
  
  signup(request, response) {  
    const viewData = {  
      title: 'Login to the Service',  
    };  
    response.render('signup', viewData);  
  },  
  
  logout(request, response) {  
    response.cookie('playlist', '');  
    response.redirect('/');  
  },  
  
  ...  
}
```

```

const accounts = {

  index(request, response) {
    const viewData = {
      title: 'Login or Signup',
    };
    response.render('index', viewData);
  },

  login(request, response) {
    const viewData = {
      title: 'Login to the Service',
    };
    response.render('login', viewData);
  },

  signup(request, response) {
    const viewData = {
      title: 'Login to the Service',
    };
    response.render('signup', viewData);
  },

  logout(request, response) {
    response.cookie('playlist', '');
    response.redirect('/');
  },

  ...
}

```

Playlist 4 Signup Login

Sign up or Log in...

Playlist 4 Signup Login

Log-in

Email

Password

[Login](#)

Playlist 4 Signup Login

Register

First Name Last Name

Email

Password

[Submit](#)


```

{{> welcomemenu }}

<section class="ui center aligned middle aligned segment">
  <p> Sign up or Log in... </p>
</section>

```

Playlist 4 Signup Login

Sign up or Log in...

```

{{> welcomemenu id='login' }}

<form class="ui stacked segment form" action="/authenticate" method="POST">
  <h3 class="ui header">Log-in</h3>
  <div class="field">
    <label>Email</label> <input placeholder="Email" name="email">
  </div>
  <div class="field">
    <label>Password</label> <input type="password" name="password">
  </div>
  <button class="ui blue submit button">Login</button>
</form>

```

Playlist 4 Signup Login

Log-in

Email

Password

[Login](#)

```

{{> welcomemenu id="signup"}}

<form class="ui stacked segment form" action="/register" method="POST">
  <h3 class="ui header">Register</h3>
  <div class="two fields">
    <div class="field">
      <label>First Name</label>
      <input placeholder="First Name" type="text" name="firstName">
    </div>
    <div class="field">
      <label>Last Name</label>
      <input placeholder="Last Name" type="text" name="lastName">
    </div>
  </div>
  <div class="field">
    <label>Email</label>
    <input placeholder="Email" type="text" name="email">
  </div>
  <div class="field">
    <label>Password</label>
    <input type="password" name="password">
  </div>
  <button class="ui blue submit button">Submit</button>
</form>

```

Playlist 4 Signup Login

Register

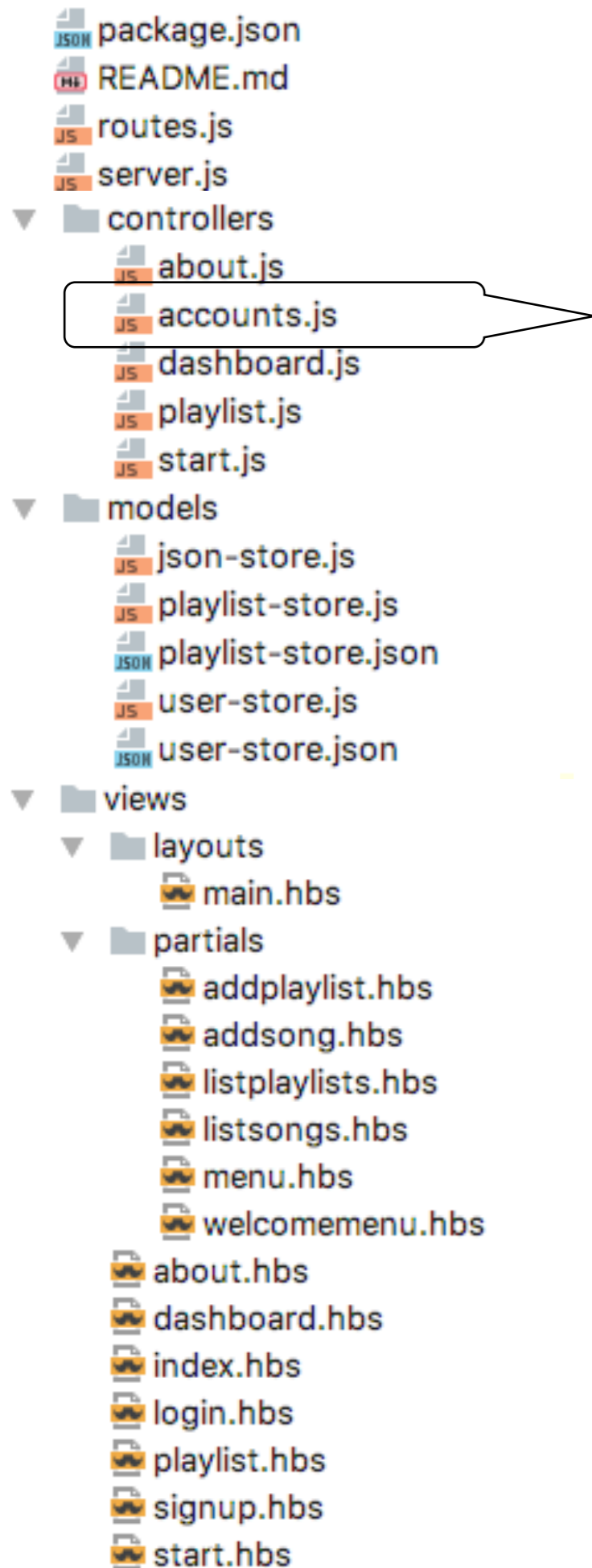
First Name Last Name

Email

Password

[Submit](#)

Accounts Routes II

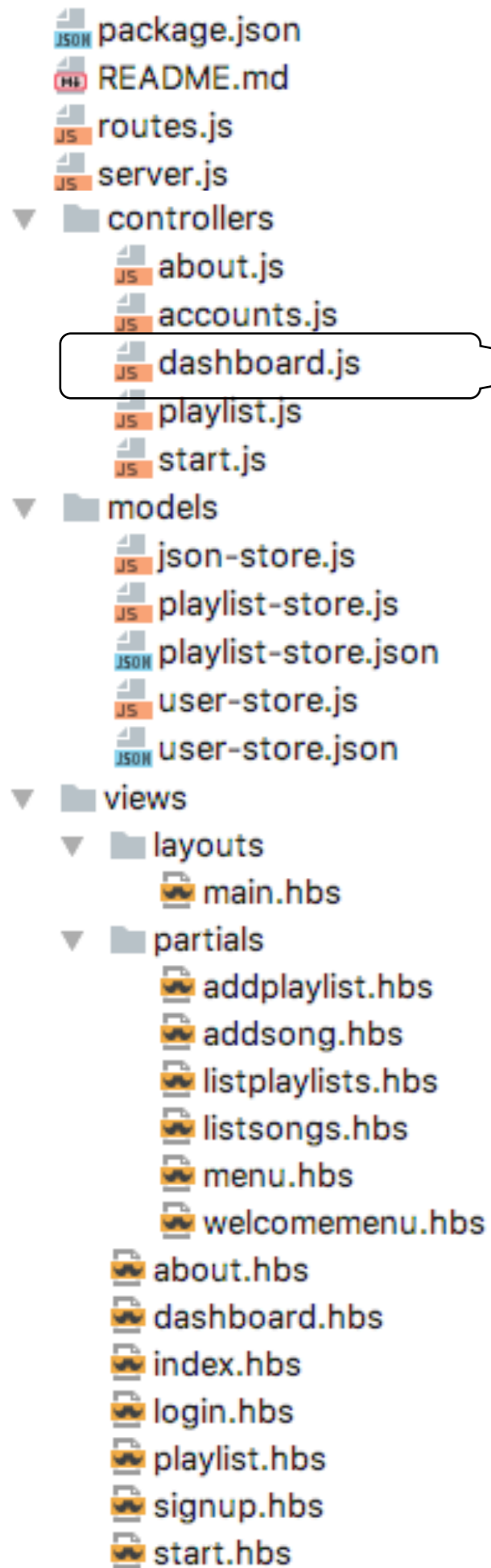


```
'/register'  
'/authenticate'
```

```
accounts.register  
accounts.authenticate
```

```
const accounts = {  
  ...  
  register(request, response) {  
    const user = request.body;  
    user.id = uuid();  
    userstore.addUser(user);  
    logger.info(`registering ${user.email}`);  
    response.redirect('/');  
  },  
  authenticate(request, response) {  
    const user = userstore.getUserByEmail(request.body.email);  
    if (user) {  
      response.cookie('playlist', user.email);  
      logger.info(`logging in ${user.email}`);  
      response.redirect('/dashboard');  
    } else {  
      response.redirect('/login');  
    }  
  },  
  ...  
}
```

Dashboard Routes

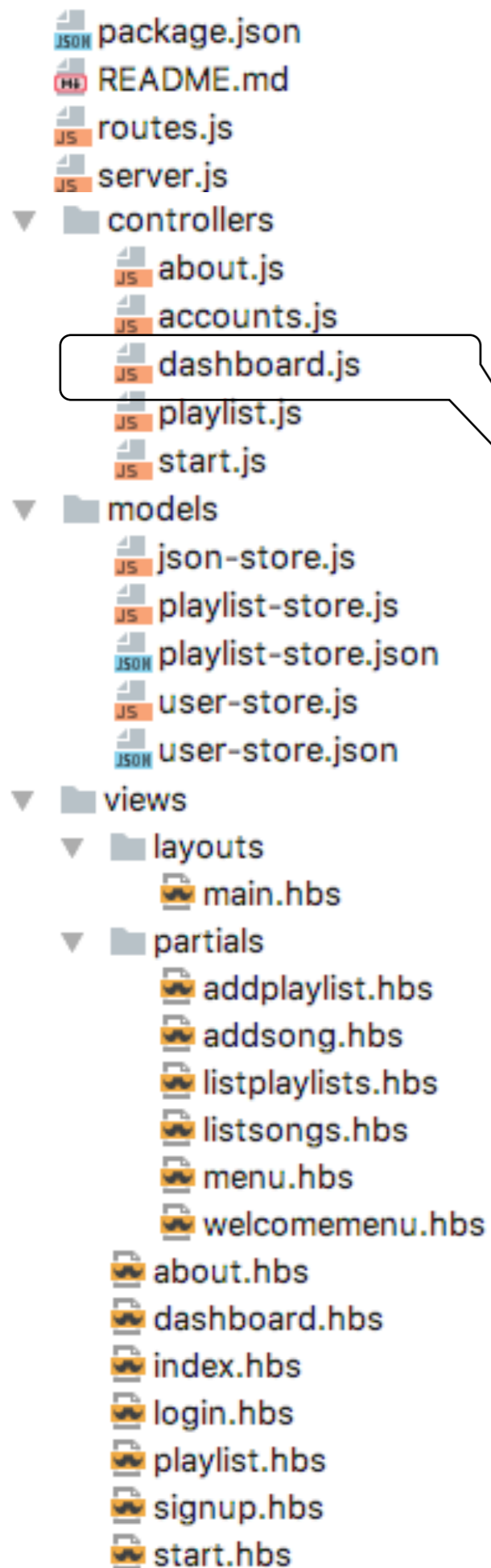


```
'/dashboard' dashboard.index
'/dashboard/deleteplaylist/:id' dashboard.deletePlaylist
'/dashboard/addplaylist' dashboard.addPlaylist
```

Dashboard Routes I

'/dashboard'

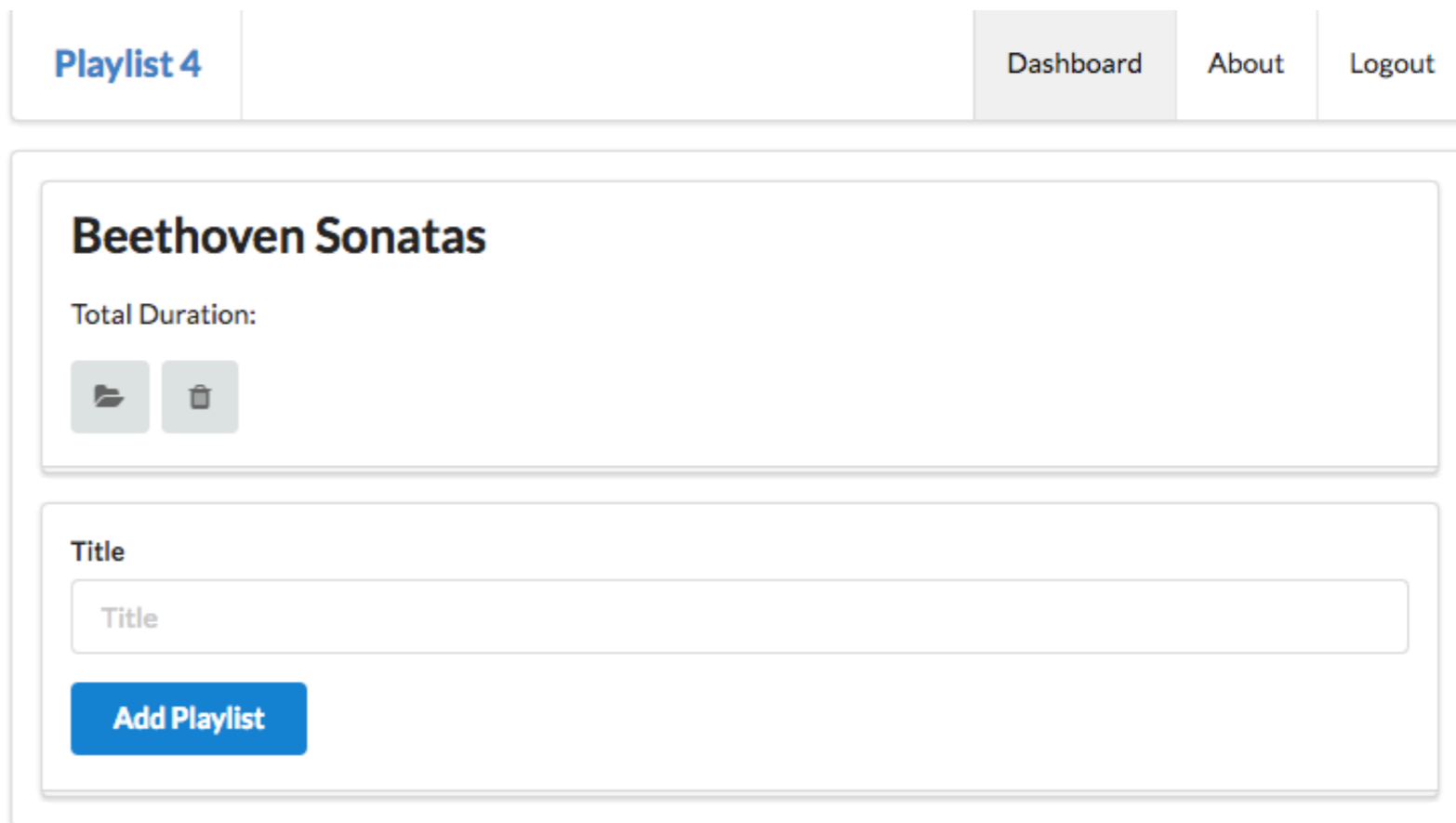
dashboard.index



```
const dashboard = {
  index(request, response) {
    logger.info('dashboard rendering');
    const loggedInUser = accounts.getCurrentUser(request);
    const viewData = {
      title: 'Playlist Dashboard',
      playlists: playlistStore.getUserPlaylists(loggedInUser.id),
    };
    logger.info('about to render', playlistStore.getAllPlaylists());
    response.render('dashboard', viewData);
  },
  ...
};

module.exports = dashboard;
```

Dashboard Index



```
const dashboard = {
  index(request, response) {
    logger.info('dashboard rendering');
    const loggedInUser = accounts.getCurrentUser(request);
    const viewData = {
      title: 'Playlist Dashboard',
      playlists: playlistStore.getUserPlaylists(loggedInUser.id),
    };
    logger.info('about to render', playlistStore.getAllPlaylists());
    response.render('dashboard', viewData);
  },
  ...
};

module.exports = dashboard;
```

dashboard.hbs

```
{{> menu id="dashboard"}}  
  
<section class="ui segment">  
  {{> listplaylists}}  
  {{> addplaylist}}  
</section>
```

listplaylists.hbs

```
{{#each playlists}}  
  <section class="ui stacked segment">  
    <h2 class="ui header">  
      {{title}}  
    </h2>  
    <p> Total Duration: {{duration}} </p>  
    <a href="/playlist/{{id}}" class="ui icon button">  
      <i class="icon folder open"></i>  
    </a>  
    <a href="/dashboard/deleteplaylist/{{id}}" class="ui icon button">  
      <i class="icon trash"></i>  
    </a>  
  </section>  
{{/each}}
```



addplaylist.hbs

```
<form class="ui stacked segment form" action="/dashboard/addplaylist" method="POST">  
  <div class="field">  
    <label>Title</label>  
    <input placeholder="Title" type="text" name="title">  
  </div>  
  <button class="ui blue submit button">Add Playlist</button>  
</form>
```

Playlist 4 Dashboard About Logout

Beethoven Sonatas

Total Duration:

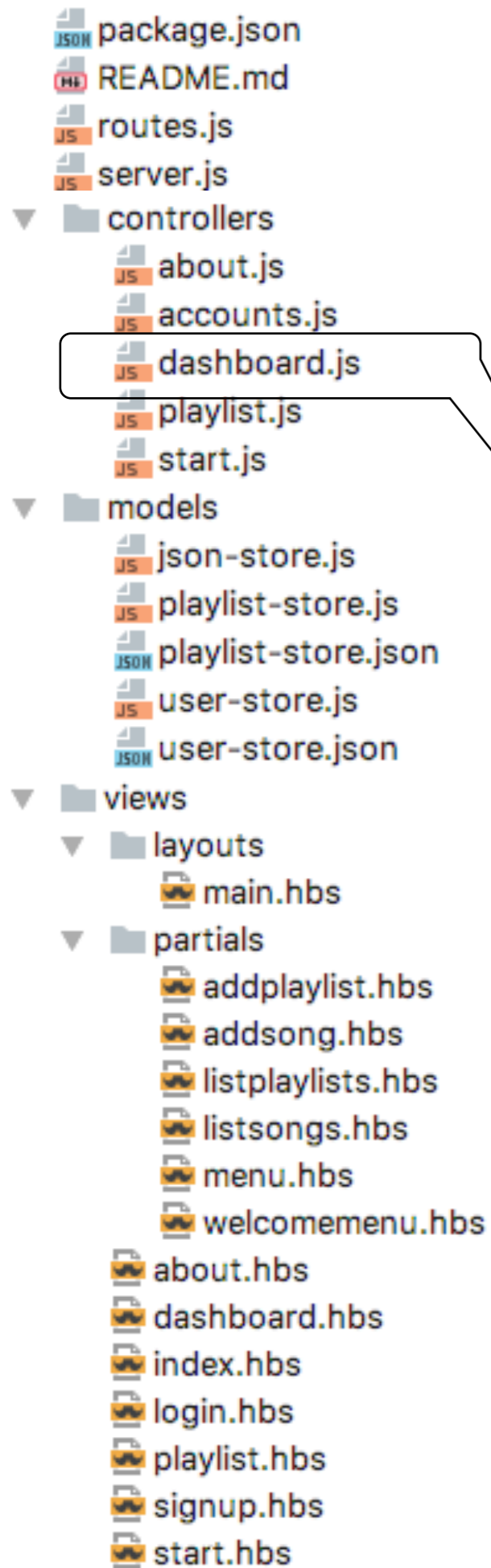
 

Title

Add Playlist

Dashboard Views

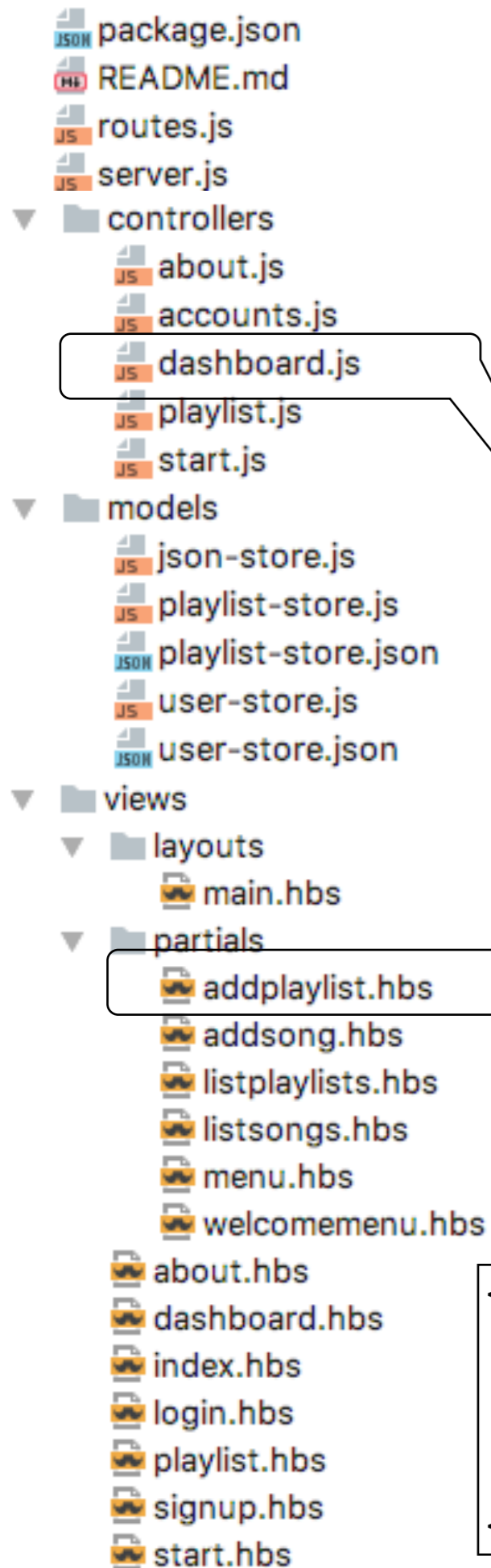
Dashboard Routes II



```
'/dashboard/deleteplaylist/:id'    dashboard.deletePlaylist
```

```
const dashboard = {  
  ...  
  deletePlaylist(request, response) {  
    const playlistId = request.params.id;  
    logger.debug(`Deleting Playlist ${playlistId}`);  
    playlistStore.removePlaylist(playlistId);  
    response.redirect('/dashboard');  
  },  
  ...  
};
```

Dashboard Routes III



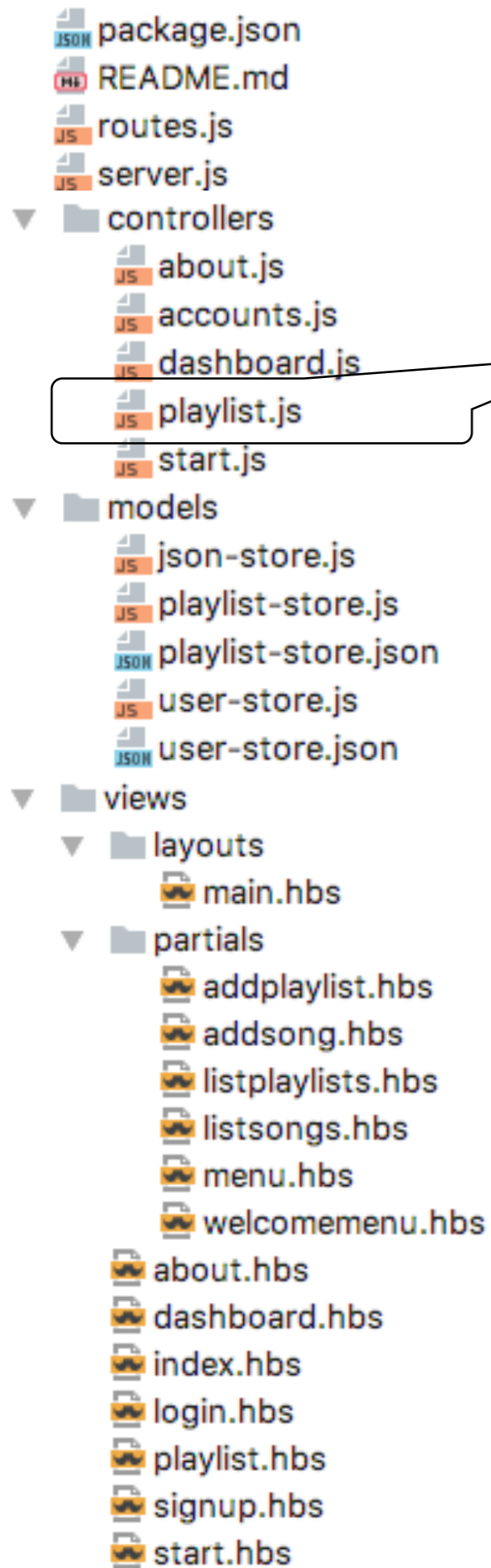
`'/dashboard/addplaylist'`

`dashboard.addPlaylist`

```
const dashboard = {  
  ...  
  addPlaylist(request, response) {  
    const loggedInUser = accounts.getCurrentUser(request);  
    const newPlayList = {  
      id: uuid(),  
      userid: loggedInUser.id,  
      title: request.body.title,  
      songs: [],  
    };  
    logger.debug('Creating a new Playlist', newPlayList);  
    playlistStore.addPlaylist(newPlayList);  
    response.redirect('/dashboard');  
  },  
};
```

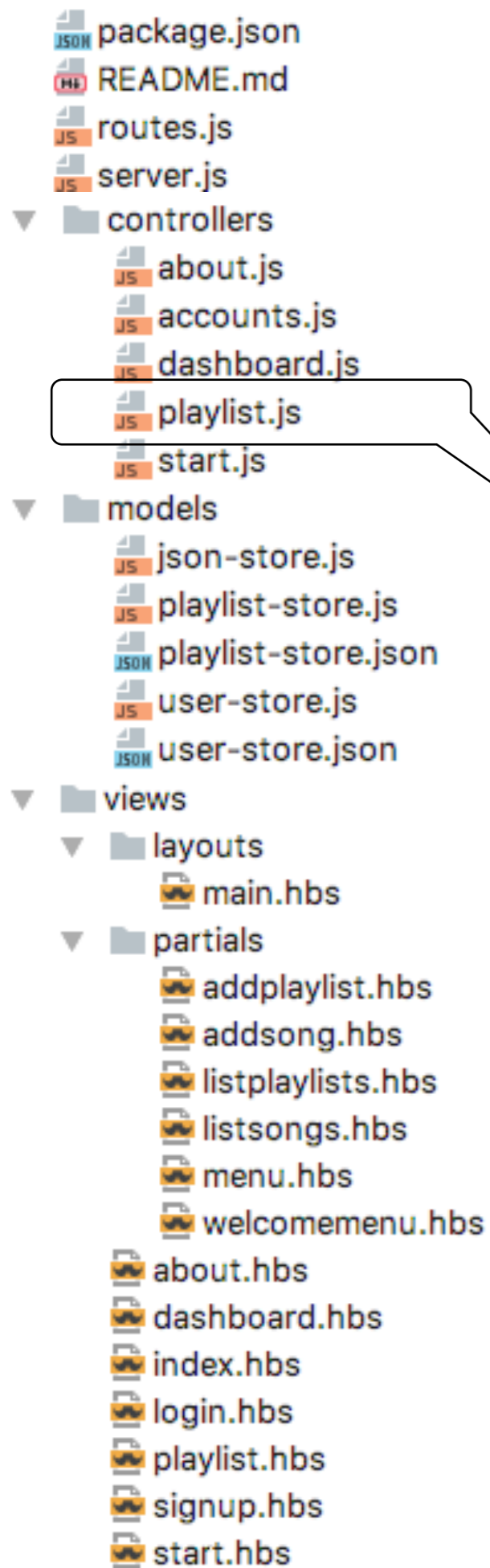
```
<form class="ui stacked segment form" action="/dashboard/addplaylist" method="POST">  
  <div class="field">  
    <label>Title</label>  
    <input placeholder="Title" type="text" name="title">  
  </div>  
  <button class="ui blue submit button">Add Playlist</button>  
</form>
```


Playlist Routes



```
'/playlist/:id'           playlist.index  
'/playlist/:id/deletesong/:songid' playlist.deleteSong  
'/playlist/:id/addsong'  playlist.addSong
```

Playlist Routes I



`'/playlist/:id'`

`playlist.index`

```
const playlist = {  
  index(request, response) {  
    const playlistId = request.params.id;  
    logger.debug('Playlist id = ', playlistId);  
    const viewData = {  
      title: 'Playlist',  
      playlist: playlistStore.getPlaylist(playlistId),  
    };  
    response.render('playlist', viewData);  
  },  
  ...  
};
```

```
{{> menu}}
```

```
<section class="ui center aligned middle aligned segment">  
  <h2 class="ui header">  
    {{playlist.title}}  
  </h2>  
  {{> listsongs}}  
  {{> addsong}}  
</section>
```

playlist.hbs

Playlist View

listsongs.hbs

```
<table class="ui table">  
  <thead>  
    <tr>  
      <th>Song</th>  
      <th>Artist</th>  
      <th></th>  
    </tr>  
  </thead>  
  <tbody>  
    {{#each playlist.songs}}  
    <tr>  
      <td>  
        {{title}}  
      </td>  
      <td>  
        {{artist}}  
      </td>  
      <td>  
        <a href="/playlist/{{../playlist.id}}/deletesong/{{id}}" class="ui icon button">  
          <i class="icon trash"></i>  
        </a>  
      </td>  
    </tr>  
    {{/each}}  
  </tbody>  
</table>
```

Playlist 4

Beethoven Sonatas

Total Duration:



Title

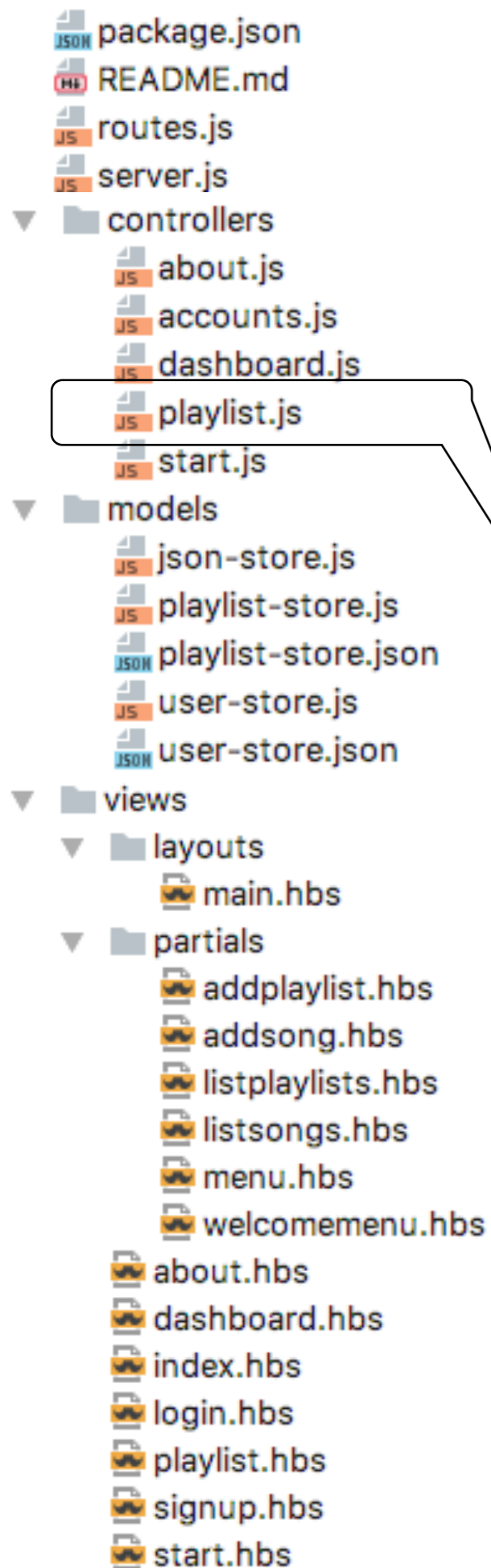
Title

Add Playlist

addsong.hbs

```
<form class="ui stacked segment form" action="/playlist/{{playlist.id}}/addsong" method="POST">  
  <div class="two fields">  
    <div class="field">  
      <label>Title</label>  
      <input placeholder="Title" type="text" name="title">  
    </div>  
    <div class="field">  
      <label>Artist</label>  
      <input placeholder="Artist" type="text" name="artist">  
    </div>  
  </div>  
  <button class="ui blue submit button">Add Song</button>  
</form>
```

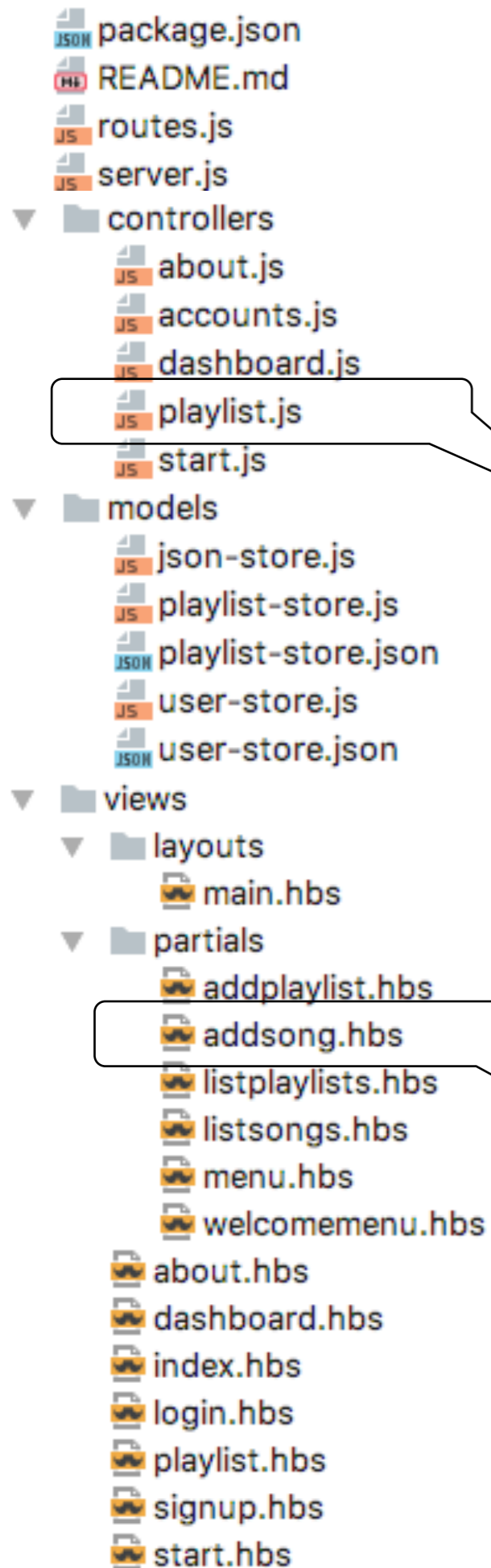
Playlist Routes II



```
'/playlist/:id/deletesong/:songid' playlist.deleteSong
```

```
const playlist = {  
  ...  
  deleteSong(request, response) {  
    const playlistId = request.params.id;  
    const songId = request.params.songid;  
    logger.debug(`Deleting Song ${songId} from Playlist ${playlistId}`);  
    playlistStore.removeSong(playlistId, songId);  
    response.redirect('/playlist/' + playlistId);  
  },  
  ...  
}
```

Playlist Routes III



```
'/playlist/:id/deletesong/:songid' playlist.deleteSong
```

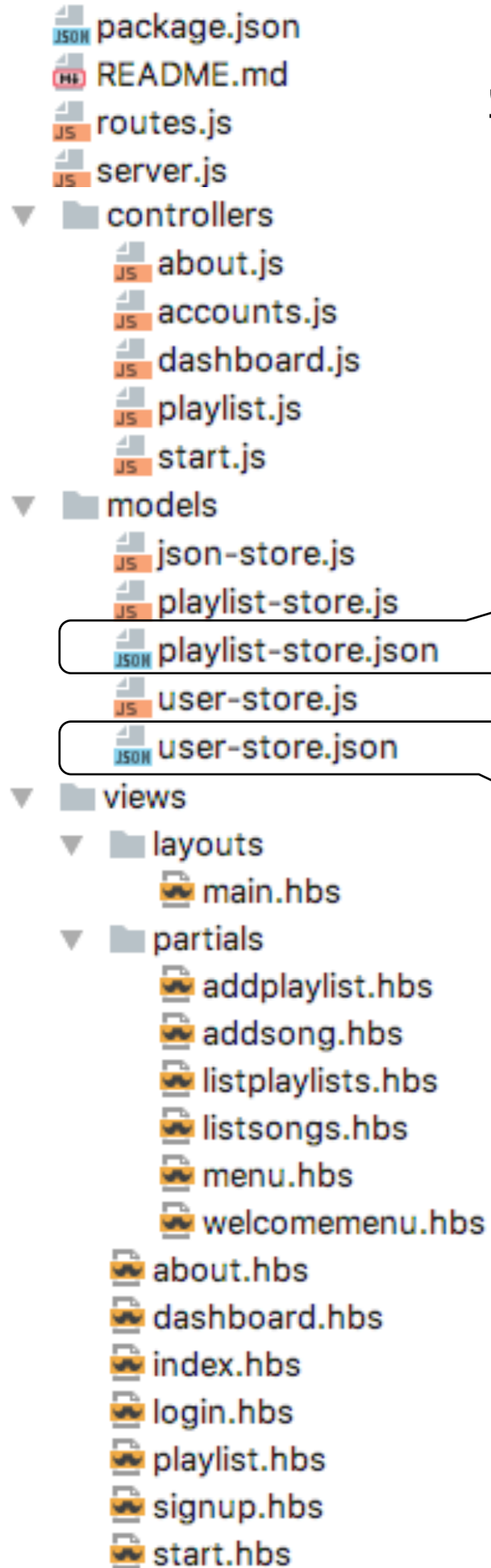
```
const playlist = {  
  ...  
  addSong(request, response) {  
    const playlistId = request.params.id;  
    const playlist = playlistStore.getPlaylist(playlistId);  
    const newSong = {  
      id: uuid(),  
      title: request.body.title,  
      artist: request.body.artist,  
    };  
    playlistStore.addSong(playlistId, newSong);  
    response.redirect('/playlist/' + playlistId);  
  },  
  ...  
}
```

```
<form class="ui stacked segment form" action="/playlist/{{playlist.id}}/addsong" method="POST">  
  <div class="two fields">  
    <div class="field">  
      <label>Title</label>  
      <input placeholder="Title" type="text" name="title">  
    </div>  
    <div class="field">  
      <label>Artist</label>  
      <input placeholder="Artist" type="text" name="artist">  
    </div>  
  </div>  
  <button class="ui blue submit button">Add Song</button>  
</form>
```

- package.json
- README.md
- routes.js
- server.js
- controllers
 - about.js
 - accounts.js
 - dashboard.js
 - playlist.js
 - start.js
- models
 - json-store.js
 - playlist-store.js
 - playlist-store.json
 - user-store.js
 - user-store.json
- views
 - layouts
 - main.hbs
 - partials
 - addplaylist.hbs
 - addsong.hbs
 - listplaylists.hbs
 - listsongs.hbs
 - menu.hbs
 - welcomemenu.hbs
 - about.hbs
 - dashboard.hbs
 - index.hbs
 - login.hbs
 - playlist.hbs
 - signup.hbs
 - start.hbs

Models

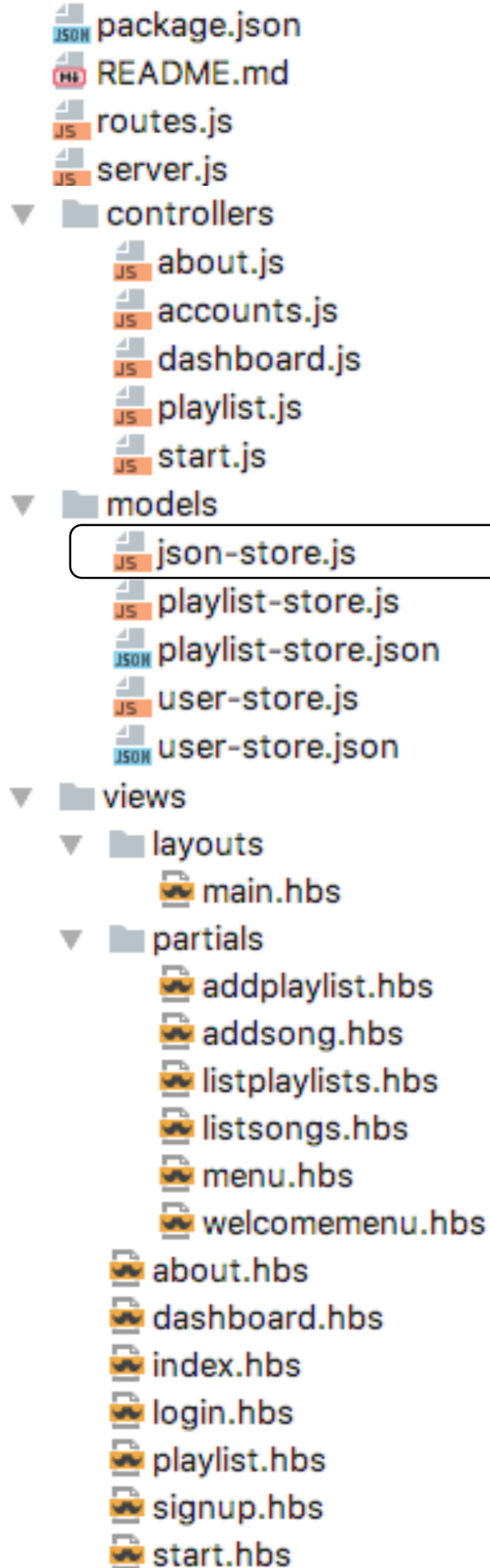
JSON files



```
{
  "playlistCollection": [
    {
      "id": "b76204cc-515a-46ff-b1dc-4efecb8d6ae2",
      "userid": "2b6f0989-7b7f-4a38-ad26-aa06b922d751",
      "title": "Beethoven Sonatas",
      "songs": [
        {
          "id": "072d6417-6d26-4a78-a562-d3f1b78f4313",
          "title": "Sonata No 1",
          "artist": "Beethoven"
        },
        {
          "id": "194d0ecd-dd62-4d6c-83a9-096275026455",
          "title": "Sonata No 2",
          "artist": "Beethoven"
        }
      ]
    }
  ]
}
```

```
{
  "users": [
    {
      "firstName": "homer",
      "lastName": "simpson",
      "email": "homer@simpson.com",
      "password": "secret",
      "id": "3ad52697-6d98-4d80-8273-084de55a86c0"
    },
    {
      "firstName": "marge",
      "lastName": "simpson",
      "email": "marge@simpson.com",
      "password": "secret",
      "id": "2b6f0989-7b7f-4a38-ad26-aa06b922d751"
    }
  ]
}
```

json-store.js



```
'use strict';

const low = require('lowdb');
const fileAsync = require('lowdb/lib/file-async');

class JsonStore {
  constructor(file, defaults) {
    this.db = low(file, { storage: fileAsync, });
    this.db.defaults(defaults).value();
  }

  add(collection, obj) {
    this.db.get(collection).push(obj).last().value();
  }

  remove(collection, obj) {
    this.db.get(collection).remove(obj).value();
  }

  removeAll(collection) {
    this.db.get(collection).remove().value();
  }

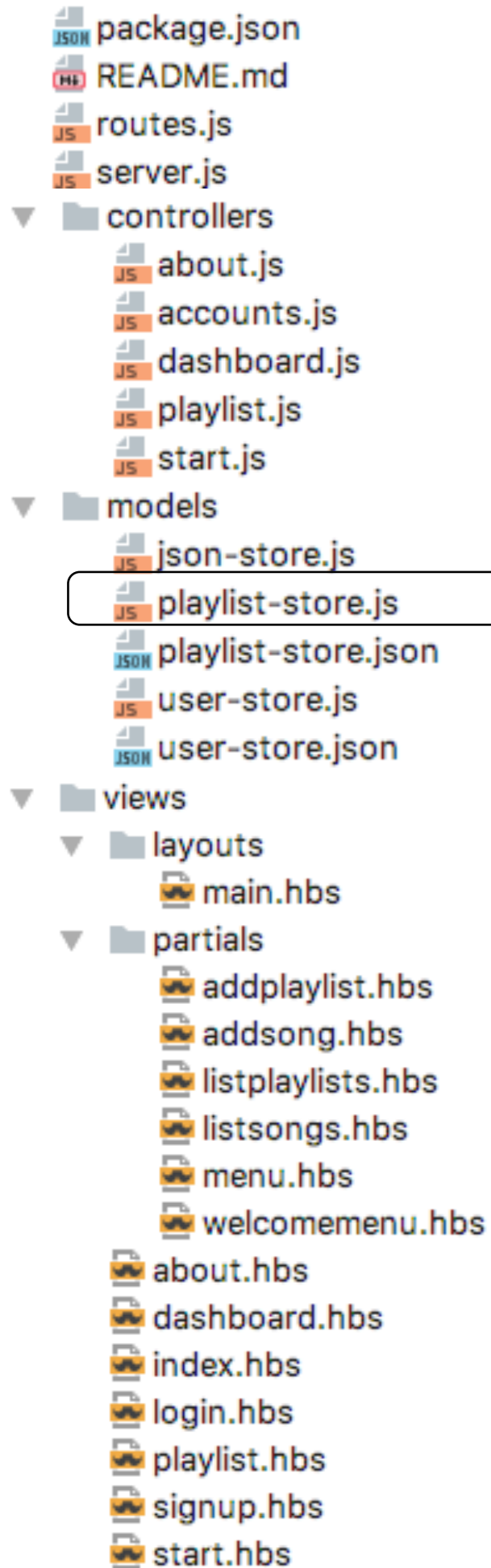
  findAll(collection) {
    return this.db.get(collection).value();
  }

  findOneBy(collection, filter) {
    const results = this.db.get(collection).filter(filter).value();
    return results[0];
  }

  findByIds(collection, ids) {
    return this.db.get(collection).keyBy('id').at(ids).value();
  }

  findBy(collection, filter) {
    return this.db.get(collection).filter(filter).value();
  }
}

module.exports = JsonStore;
```

playlist-store.js

```
'use strict';

const _ = require('lodash');
const JsonStore = require('./json-store');

const playlistStore = {

  store: new JsonStore('./models/playlist-store.json', { playlistCollection: [] }),
  collection: 'playlistCollection',

  getAllPlaylists() {
    return this.store.findAll(this.collection);
  },

  getUserPlaylists(userid) {
    return this.store.findBy(this.collection, { userid: userid });
  },

  getPlaylist(id) {
    return this.store.findOneBy(this.collection, { id: id });
  },

  addPlaylist(playlist) {
    this.store.add(this.collection, playlist);
  },

  removePlaylist(id) {
    const playlist = this.getPlaylist(id);
    this.store.remove(this.collection, playlist);
  },

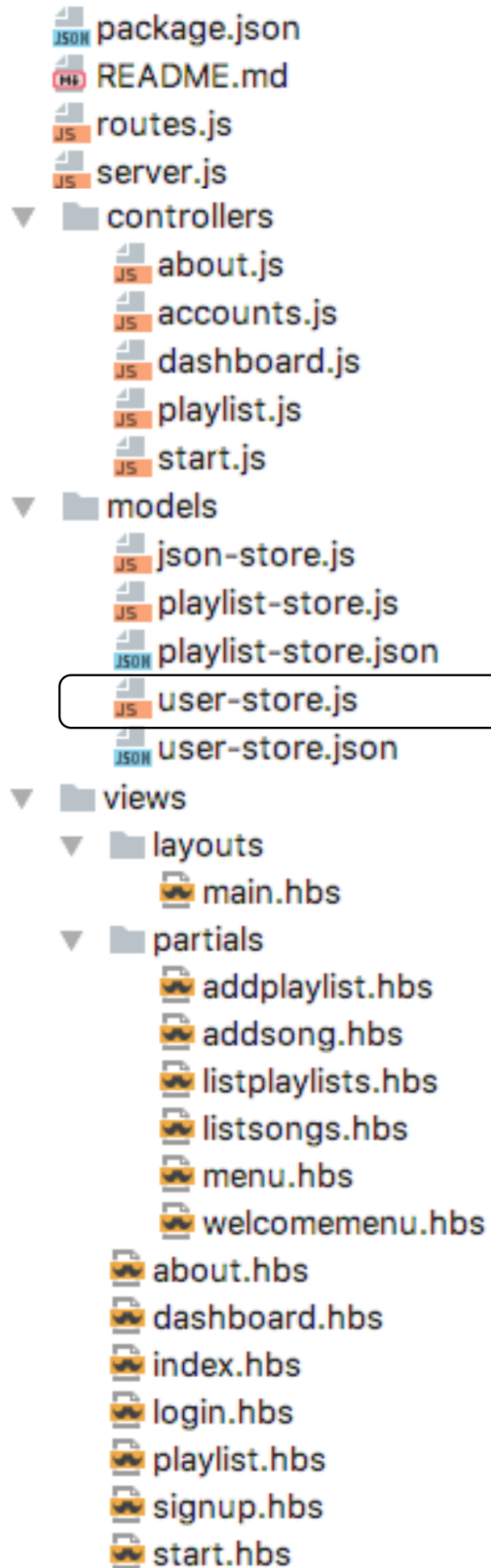
  removeAllPlaylists() {
    this.store.removeAll(this.collection);
  },

  addSong(id, song) {
    const playlist = this.getPlaylist(id);
    playlist.songs.push(song);
  },

  removeSong(id, songId) {
    const playlist = this.getPlaylist(id);
    const songs = playlist.songs;
    _.remove(songs, { id: songId });
  },
};

module.exports = playlistStore;
```

user-store.js



```
'use strict';

const _ = require('lodash');
const JsonStore = require('./json-store');

const userStore = {

  store: new JsonStore('./models/user-store.json', {users: []}),
  collection: 'users',

  getAllUsers() {
    return this.store.findAll(this.collection);
  },

  addUser(user) {
    this.store.add(this.collection, user);
  },

  getUserById(id) {
    return this.store.findOneBy(this.collection, { id: id });
  },

  getUserByEmail(email) {
    return this.store.findOneBy(this.collection, { email: email });
  },
}

module.exports = userStore;
```