

# Playlist Controller

---

# Introducing a new view

---

- Typically need 3 things:
  - Controller
  - View
  - Route

controller:  
playlist.js

```
'use strict';

const logger = require('../utils/logger');
const playlistCollection = require('../models/playlist-store.js');

const playlist = {
  index(request, response) {
    const viewData = {
      title: 'Playlist',
    };
    response.render('playlist', viewData);
  },
};

module.exports = playlist;
```

view:  
playlist.hbs

```
{{> menu}}

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

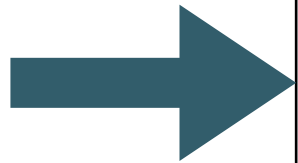
route:  
routes.js

```
router.get('/playlist/:id', playlist.index);
```

# Playlist Controller

---

- In order to display the correct playlist, we need to extract the id from the url.



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

- Model-View-Controller Pattern
- Controllers:
- Model Playlists stored in JSON
- Views:

#### back-end +

`.env`

`.gitignore`

`.jscsrc`

`controllers/about.js`

`controllers/dashboard.js`

`controllers/start.js`

`models/playlist-store.js`

`models/playlist-store.json`

`package.json`

`README.md`

`routes.js`

`server.js`

`utils/logger.js`

#### front-end +

`assets`

`views/about.hbs`

`views/dashboard.hbs`

`views/layouts/main.hbs`

`views/partials/listsongs.hbs`

`views/partials/menu.hbs`

`views/start.hbs`

# Revised Playlist-Store Model

---

- playlistStore is an object with 2 methods:
- retrieve all playlists from store.
- retrieve a specific playlist by it's id from the store.

```
const playlistStore = {  
  playlistCollection: require('./playlist-store.json').playlistCollection,  
  getAllPlaylists() {  
    return this.playlistCollection;  
  },  
  getPlaylist(id) {  
    return _.find(this.playlistCollection, { id: id });  
  },  
};
```

*Dont worry about how  
this is implemented for  
the moment!*

# Revised playlist Controller

---

- Retrieve playlist ID in request.
- Get playlist matching this id and insert into viewData.
- Pass **viewData** to the view (containing the selected playlist).

```
const playlistStore = require('../models/playlist-store');

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);
  },
};
```

# Revised Playlist View

```
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);  
}
```

Beethoven Sonatas	
Song	Artist
Piano Sonata No. 3	Beethoven
Piano Sonata No. 7	Beethoven
Piano Sonata No. 10	Beethoven

```
<table class="ui table">  
  <thead>  
    <tr>  
      <th>Song</th>  
      <th>Artist</th>  
    </tr>  
  </thead>  
  <tbody>  
    {{#each playlist.songs}}  
      <tr>  
        <td>  
          {{title}}  
        </td>  
        <td>  
          {{artist}}  
        </td>  
      </tr>  
    {{/each}}  
  </tbody>  
</table>
```

listsongs.hbs

playlist.hbs

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



# Revised dashboard Controller

---

- Require a link to **playlistStore**.
- Pass **viewData** to the view (containing all the playlists).

```
const playlistStore = require('../models/playlist-store');

const dashboard = {
  index(request, response) {

    logger.info('dashbaord rendering');

    const viewData = {
      title: 'Playlist Dashboard',
      playlists: playlistStore.getAllPlaylists(),
    };

    logger.info('about to render ', playlistStore.getAllPlaylists());
    response.render('dashboard', viewData);
  },
};
```