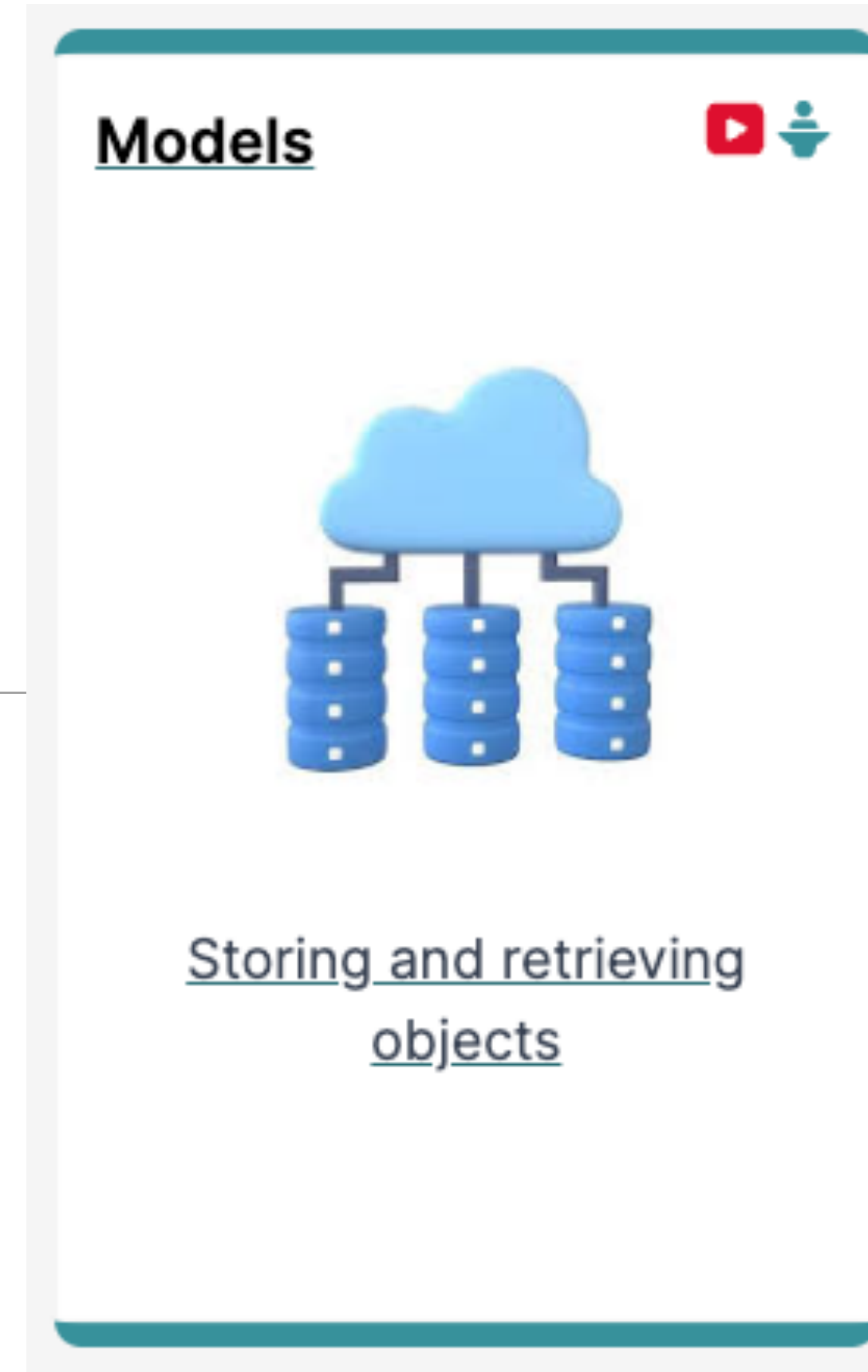
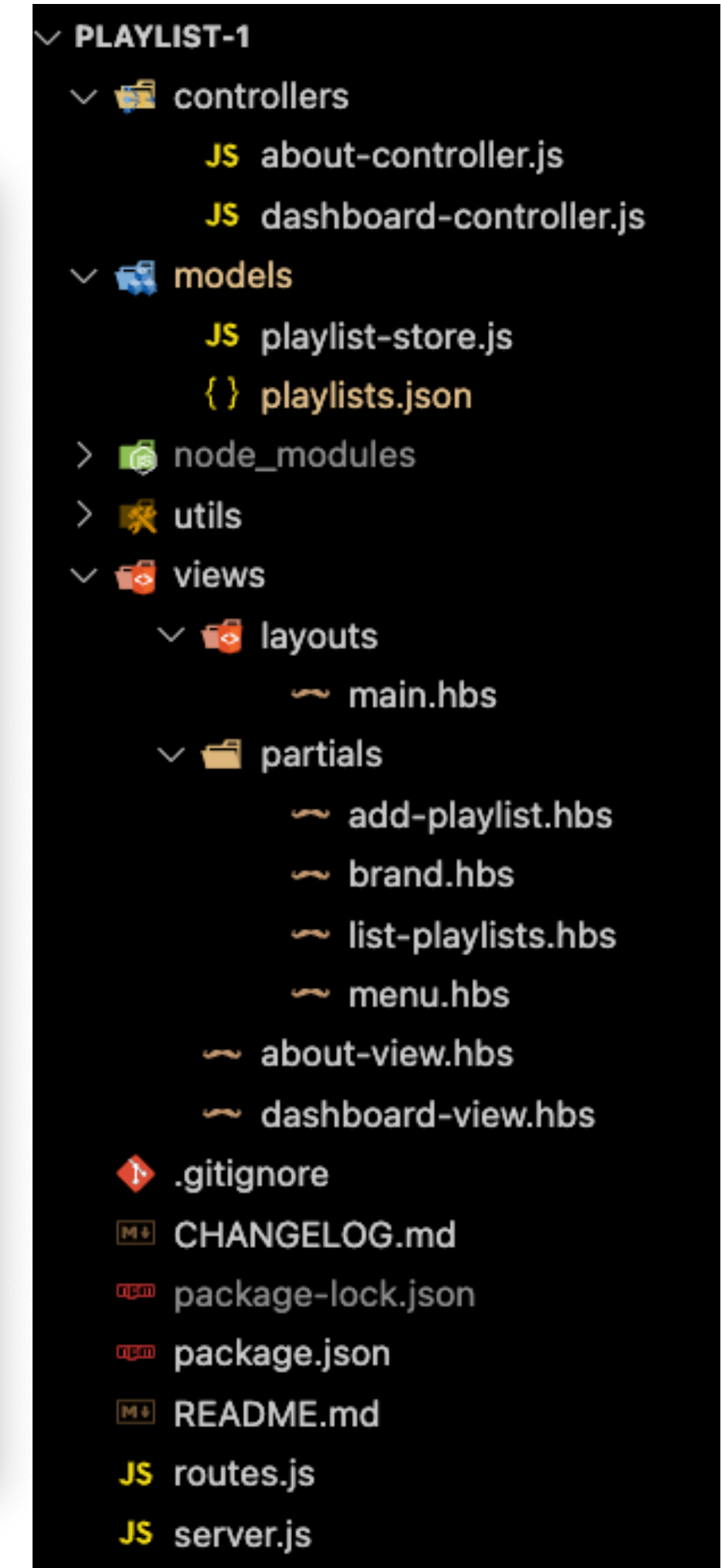
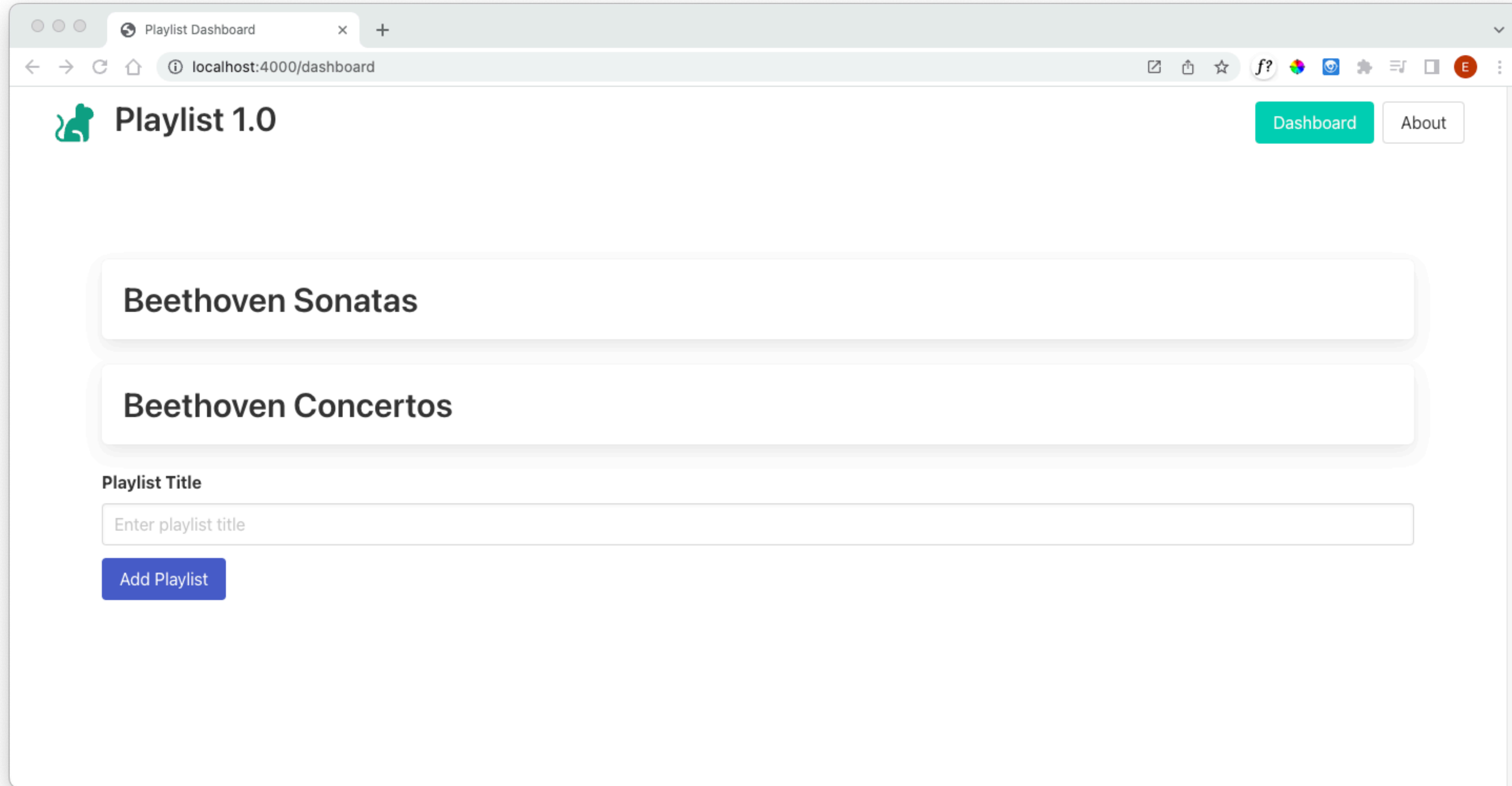
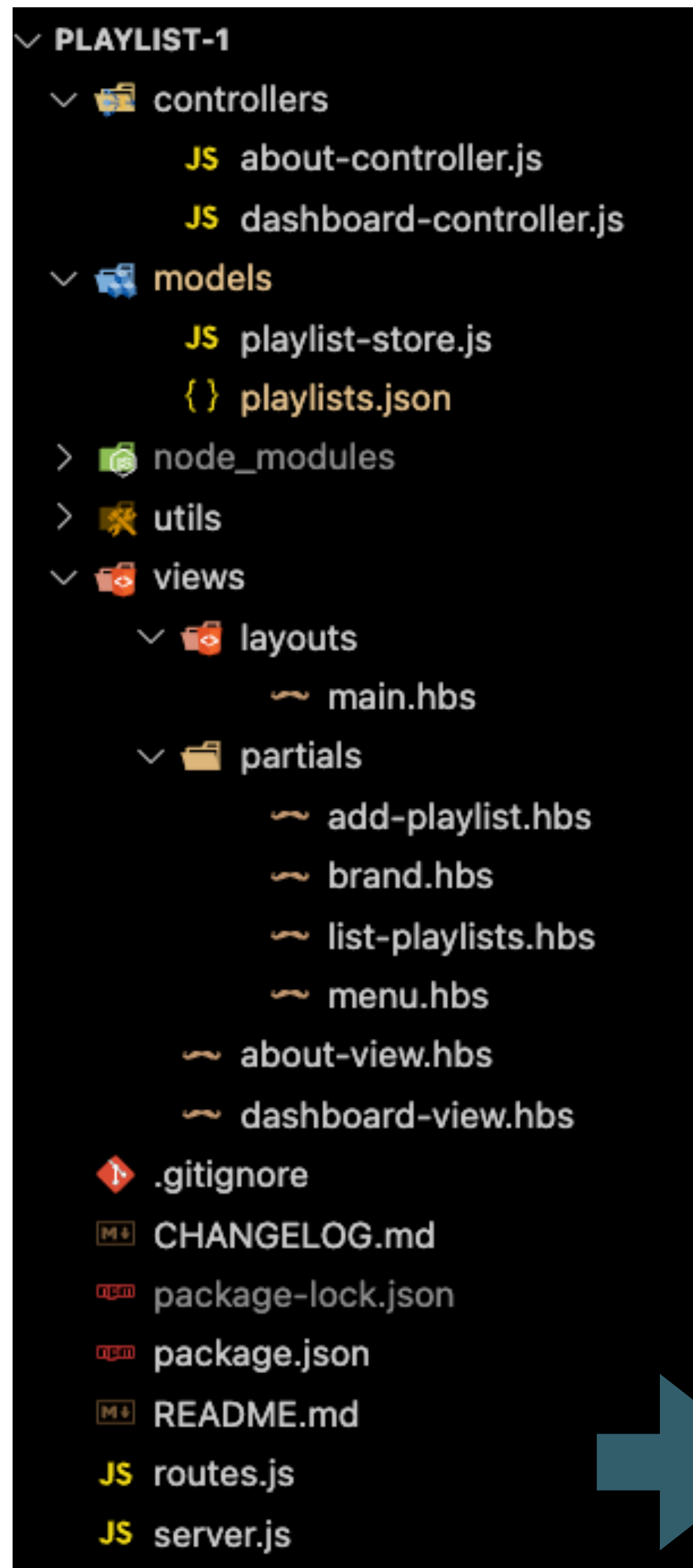


Models



Playlist 1





- Four “routes” defined for the application:

- “/”
- “/dashboard”
- “/dashboard/addplaylist”
- “/about”

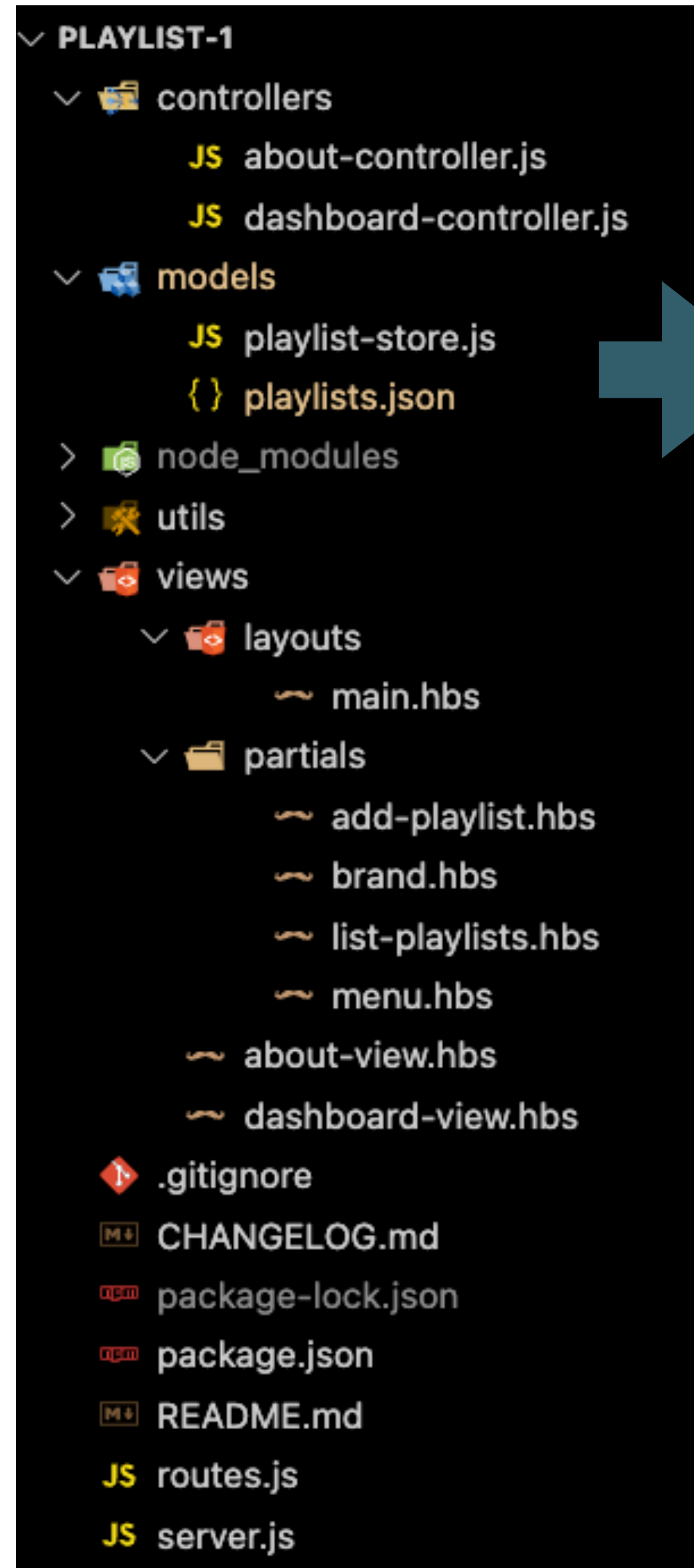
routes.js

```
import express from "express";
import { dashboardController } from "../controllers/dashboard-controller.js";
import { aboutController } from "../controllers/about-controller.js";

export const router = express.Router();

router.get("/", dashboardController.index);
router.get("/dashboard", dashboardController.index);
router.post("/dashboard/addplaylist", dashboardController.addPlaylist);
router.get("/about", aboutController.index);
```

playlist-store



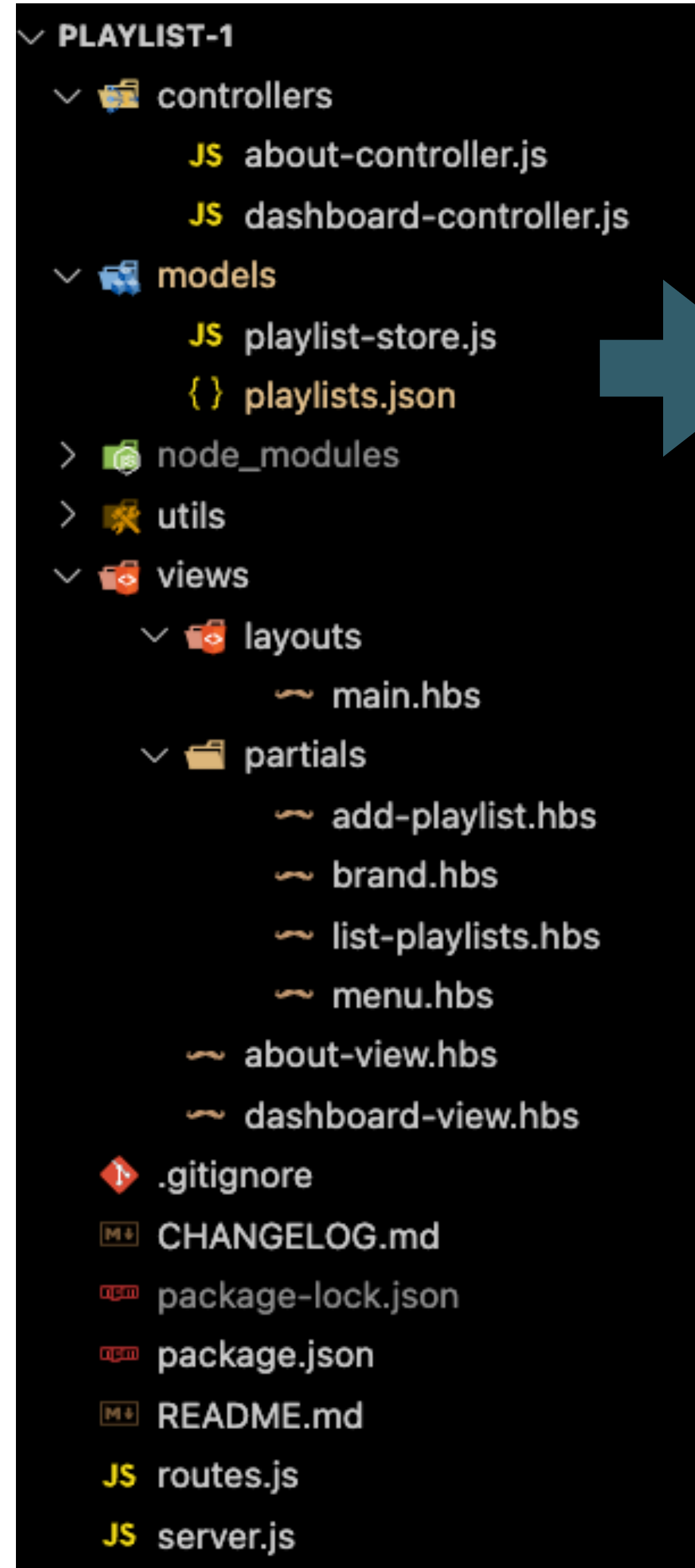
```
import { v4 } from "uuid";
import { createStore } from "../utils/store-utils.js";

const db = createStore("playlists");

export const playlistStore = {
  async getAllPlaylists() { ... },
  async addPlaylist(playlist) { ... },
  async getPlaylistById(id) { ... },
  async deletePlaylistById(id) { ... },
  async deleteAllPlaylists() { ... },
};
```

- Simple Store for “playlist” objects
- Provides methods for:
 - Add
 - Get by ID
 - Delete by ID
 - Get All
 - Delete All

playlist-store



```
import { v4 } from "uuid";
import { initStore } from "../utils/store-utils.js";

const db = initStore("playlists");

export const playlistStore = {
  async getAllPlaylists() {
    await db.read();
    return db.data.playlists;
  },

  async addPlaylist(playlist) {
    await db.read();
    playlist._id = v4();
    db.data.playlists.push(playlist);
    await db.write();
    return playlist;
  },

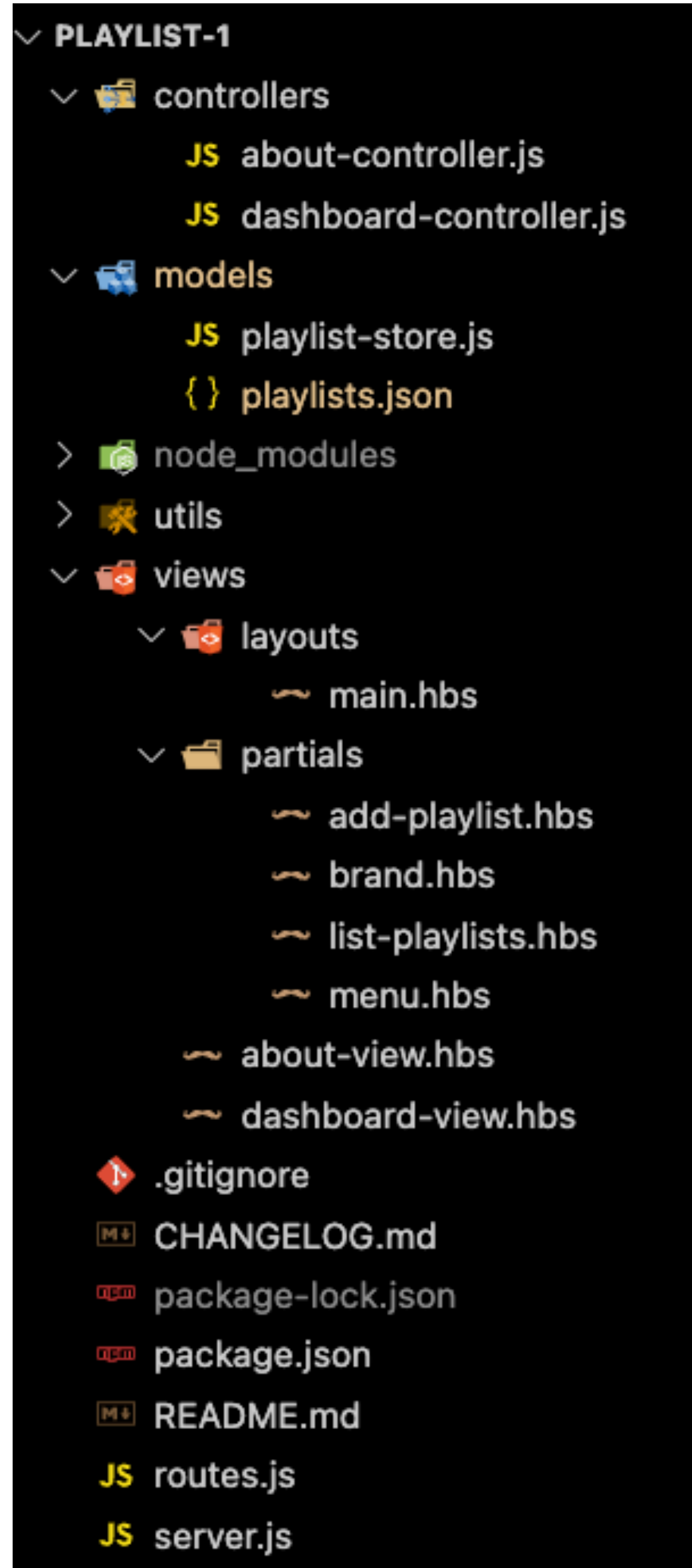
  async getPlaylistById(id) {
    await db.read();
    const list = db.data.playlists.find((playlist) => playlist._id === id);
    return list;
  },

  async deletePlaylistById(id) {
    await db.read();
    const index = db.data.playlists.findIndex((playlist) => playlist._id === id);
    db.data.playlists.splice(index, 1);
    await db.write();
  },

  async deleteAllPlaylists() {
    db.data.playlists = [];
    await db.write();
  },
};
```

- Implementation uses a file-based database
- This stores the objects in a JSON file
 - playlists.json
- Use this module as a 'black box'
- Postpone understanding this component for the moment, regard as a 'black box' storage

dashboard-controller



```
import { playlistStore } from "../models/playlist-store.js";

export const dashboardController = {
  async index(request, response) {
    const viewData = {
      title: "Playlist Dashboard",
      playlists: await playlistStore.getAllPlaylists(),
    };
    console.log("dashboard rendering");
    response.render("dashboard-view", viewData);
  },

  async addPlaylist(request, response) {
    const newPlaylist = {
      title: request.body.title,
    };
    console.log(`adding playlist ${newPlaylist.title}`);
    await playlistStore.addPlaylist(newPlaylist);
    response.redirect("/dashboard");
  },
};
```

- 2 “Actions”:
- List all Playlists
- Add new Playlists

dashboard-controller: index

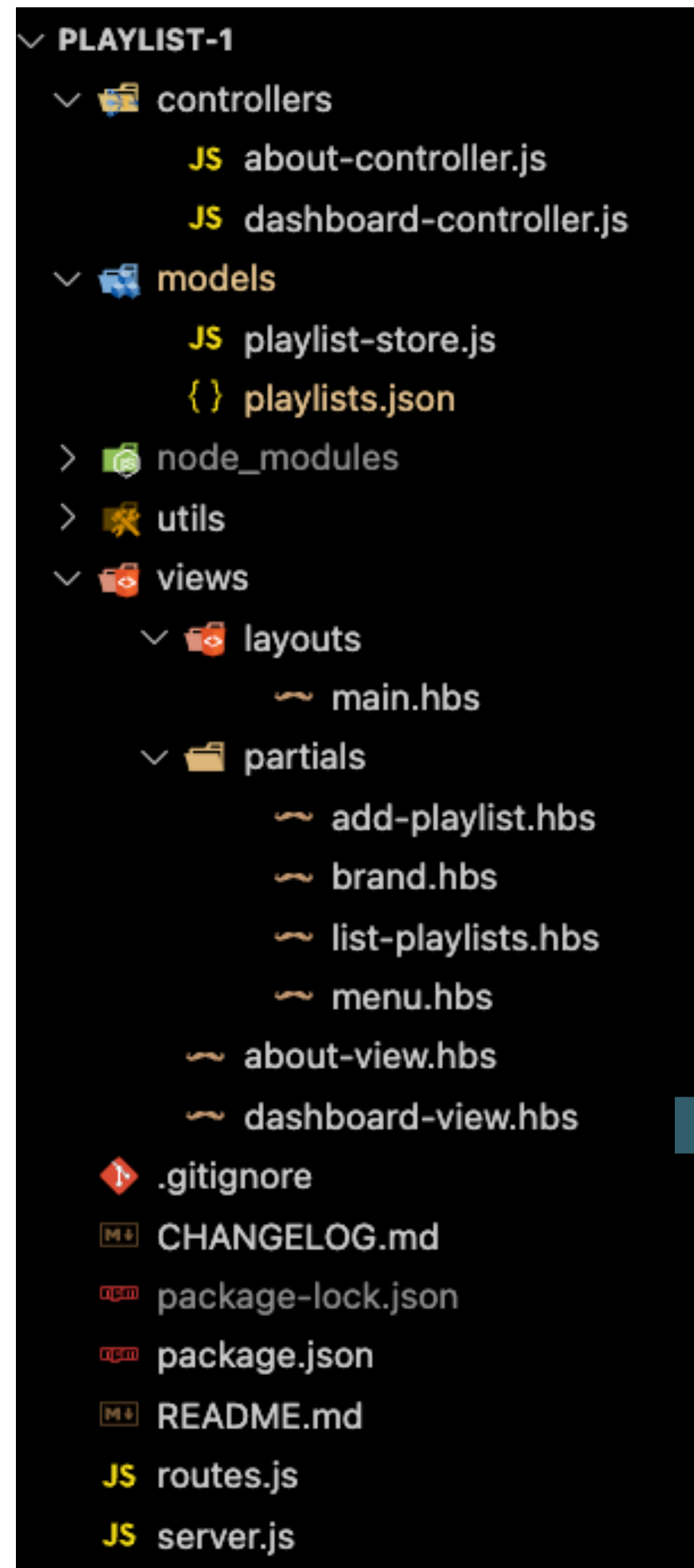
```
async index(request, response) {  
  const viewData = {  
    title: "Playlist Dashboard",  
    playlists: await playlistStore.getAllPlaylists(),  
  };  
  console.log("dashboard rendering");  
  response.render("dashboard-view", viewData);  
},
```

- Create a **viewData** object to be send to the view
- Include the page title
- Include all known playlists, retrieved from the playlist-store
- Pass viewData to the view

dashboard-controller: addPlaylist

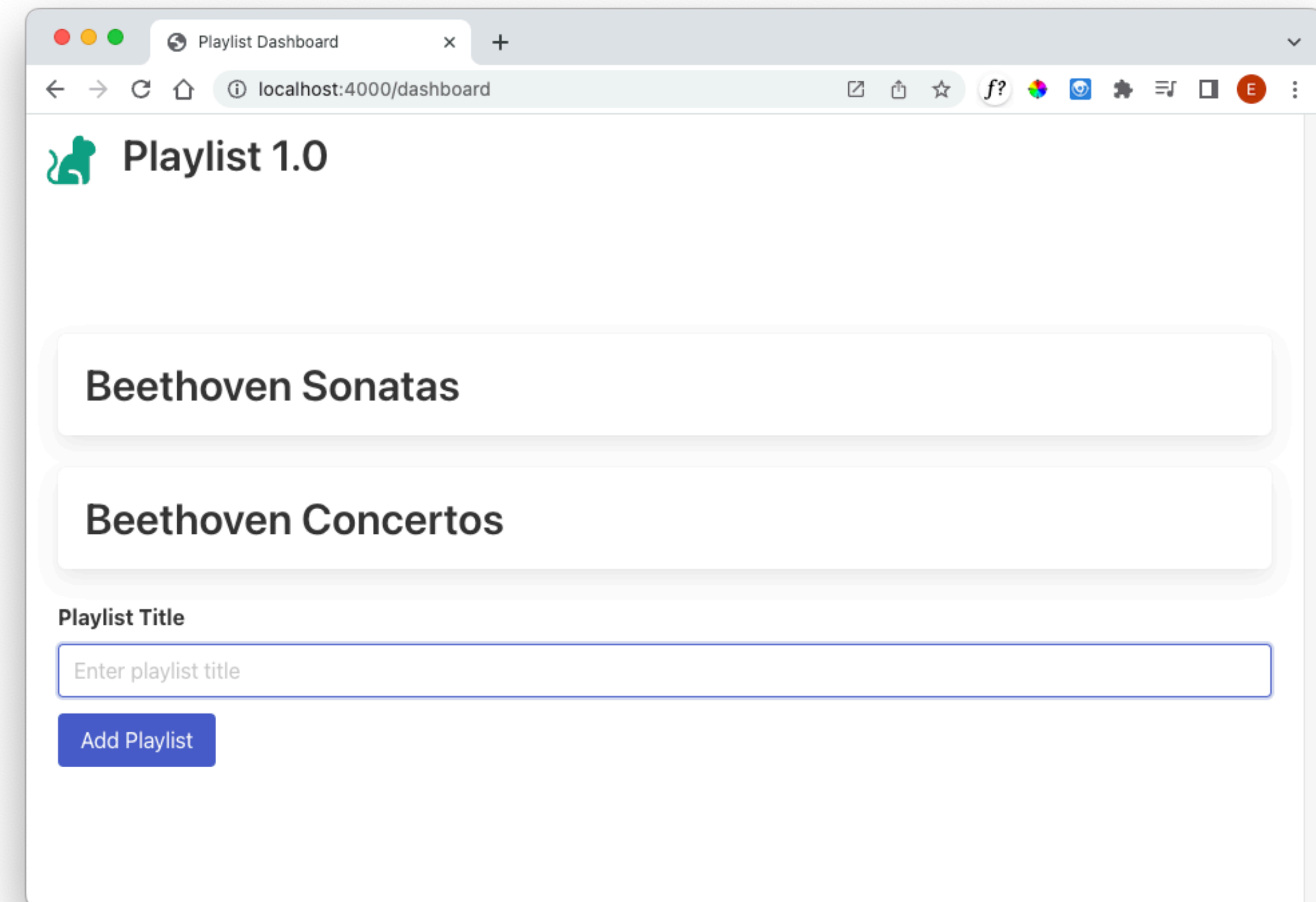
```
async addPlaylist(request, response) {  
  const newPlaylist = {  
    title: request.body.title,  
  };  
  console.log(`adding playlist ${newPlaylist.title}`);  
  await playlistStore.addPlaylist(newPlaylist);  
  response.redirect("/dashboard");  
},
```

- Create a **newPlaylist** object, containing the 'title' string passed from the form.
- Add the new playlist to the playlist store
- Redirect to the “/dashboard” route



dashboard-view.hbs

```
{{> menu active="dashboard"}}  
  
<section class="section">  
  {{> list-playlists}}  
  {{> add-playlist}}  
</section>
```



dashboard-view.hbs

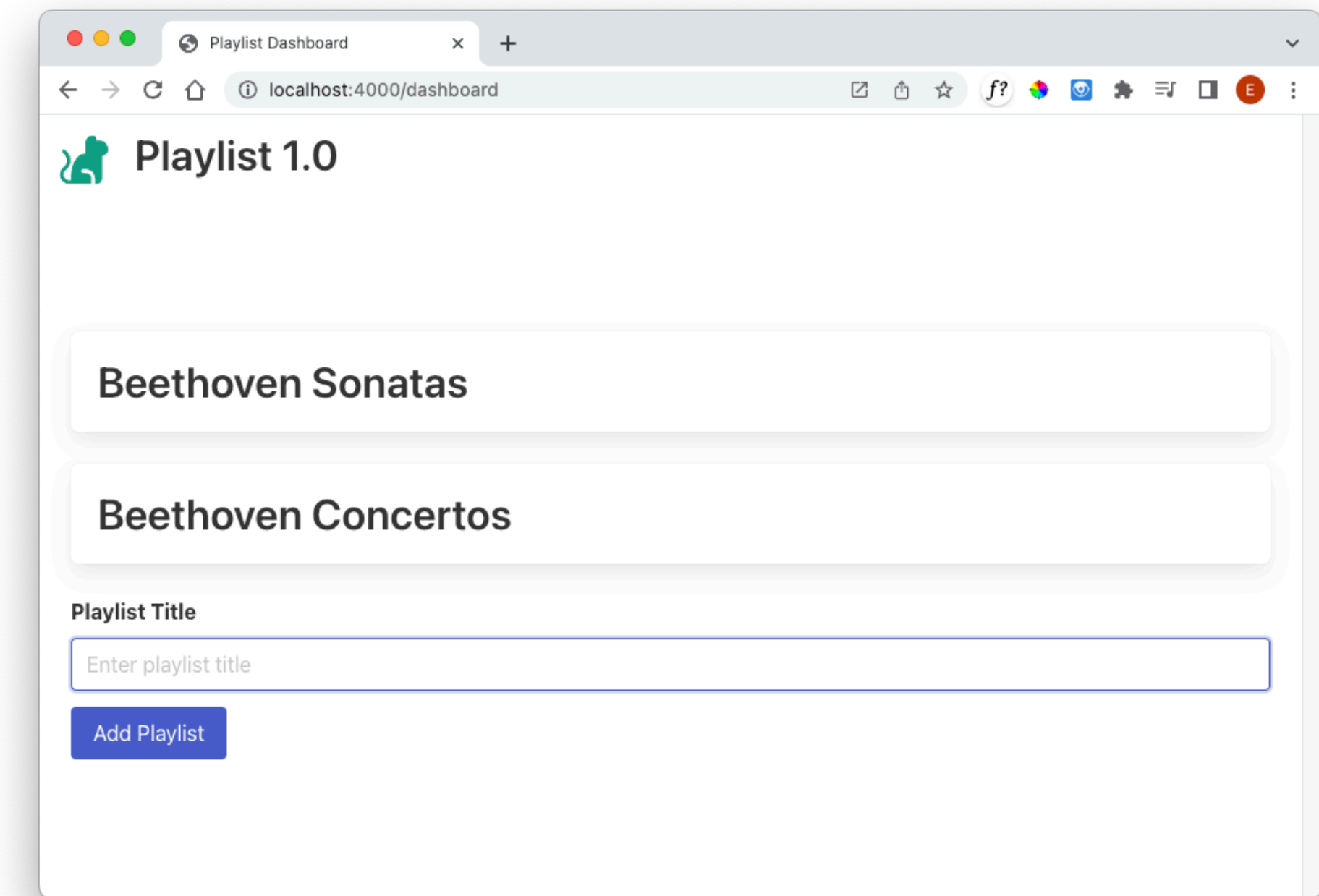
```
{{> menu active="dashboard"}}  
  
<section class="section">  
  {{> list-playlists}}  
  {{> add-playlist}}  
</section>
```

list-playlists.hbs

```
{{#each playlists}}  
<div class="box box-link-hover-shadow">  
  <h2 class="title">  
    <a href="/playlist/{{_id}}">  
      {{title}}  
    </a>  
  </h2>  
</div>  
{{/each}}
```

```
<form action="/dashboard/addplaylist" method="POST">  
  <div class="field">  
    <label class="label">Playlist Title</label>  
    <input class="input" type="text" placeholder="Enter playlist title" name="title">  
  </div>  
  <button class="button is-link">Add Playlist</button>  
</form>
```

add-playlist.hbs



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

- Iterate through each playlist in viewData.playlists

list-playlists.hbs

```
{{#each playlists}}
<div class="box box-link-hover-shadow">
  <h2 class="title">
    {{title}}
  </h2>
</div>
{{/each}}
```

Beethoven Sonatas

Beethoven Concertos

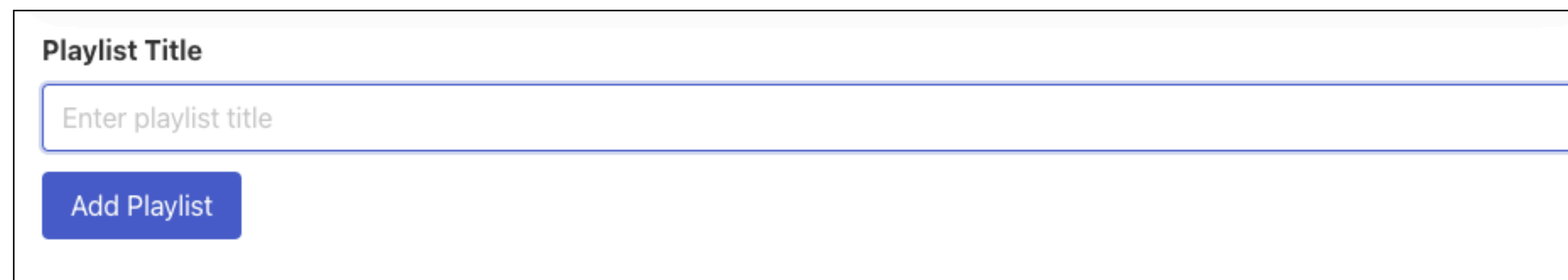
- Display the title of each playlist

Add Playlist

```
<form action="/dashboard/addplaylist" method="POST">
  <div class="field">
    <label class="label">Playlist Title</label>
    <input class="input" type="text" placeholder="Enter playlist title" name="title">
  </div>
  <button class="button is-link">Add Playlist</button>
</form>
```

add-playlist.hbs

- Simple form to accept a string (playlist title)
- Send the title to the route “/dashboard/addplaylist”



Playlist Title

Add Playlist

Add Playlist

```
<form action="/dashboard/addplaylist" method="POST">
  <div class="field">
    <label class="label">Playlist Title</label>
    <input class="input" type="text" placeholder="Enter playlist title" name="title">
  </div>
  <button class="button is-link">Add Playlist</button>
</form>
```

add-playlist.hbs

addplaylist route

```
router.post("/dashboard/addplaylist", dashboardController.addPlaylist);
```

routes.js

Playlist Title

Add Playlist

Add Playlist

```
<form action="/dashboard/addplaylist" method="POST">
  <div class="field">
    <label class="label">Playlist Title</label>
    <input class="input" type="text" placeholder="Enter playlist title" name="title">
  </div>
  <button class="button is-link">Add Playlist</button>
</form>
```

add-playlist.hbs

addPlaylist
controller action

```
async addPlaylist(request, response) {
  const newPlaylist = {
    title: request.body.title,
  };
  console.log(`adding playlist ${newPlaylist.title}`);
  await playlistStore.addPlaylist(newPlaylist);
  response.redirect("/dashboard");
},
```

dashboard-controller.js

Playlist Title

Add Playlist

Add Playlist

```
async addPlaylist(request, response) {  
  const newPlaylist = {  
    title: request.body.title,  
  };  
  console.log(`adding playlist ${newPlaylist.title}`);  
  await playlistStore.addPlaylist(newPlaylist);  
  response.redirect("/dashboard");  
},
```

dashboard-controller.js

```
{  
  "file": "./models/playlists.json",  
  "playlists": [  
    {  
      "title": "Beethoven Sonatas",  
      "_id": "17666bb5-522b-40c6-8432-9754b9ff4411"  
    },  
    {  
      "title": "Beethoven Concertos",  
      "_id": "a8f44b6f-28f3-406e-87da-78c2364edb78"  
    }  
  ]  
}
```

playlists.json

Beethoven Sonatas

Beethoven Concertos

Playlist Title

Enter playlist title

Add Playlist

Controller / Model

```
import { playlistStore } from "../models/playlist-store.js";

export const dashboardController = {
  async index(request, response) {
    const viewData = {
      title: "Playlist Dashboard",
      playlists: await playlistStore.getAllPlaylists(),
    };
    console.log("dashboard rendering");
    response.render("dashboard-view", viewData);
  },

  async addPlaylist(request, response) {
    const newPlaylist = {
      title: request.body.title,
    };
    console.log(`adding playlist ${newPlaylist.title}`);
    await playlistStore.addPlaylist(newPlaylist);
    response.redirect("/dashboard");
  },
};
```

Beethoven Sonatas

Beethoven Concertos

Playlist Title

Enter playlist title

Add Playlist

```
import { v4 } from "uuid";
import { initStore } from "../utils/store-utils.js";

const db = initStore("playlists");

export const playlistStore = {
  > async getAllPlaylists() { ...
  },
  > async addPlaylist(playlist) { ...
  },
  > async getPlaylistById(id) { ...
  },
  > async deletePlaylistById(id) { ...
  },
  > async deleteAllPlaylists() { ...
  },
};
```

Models

