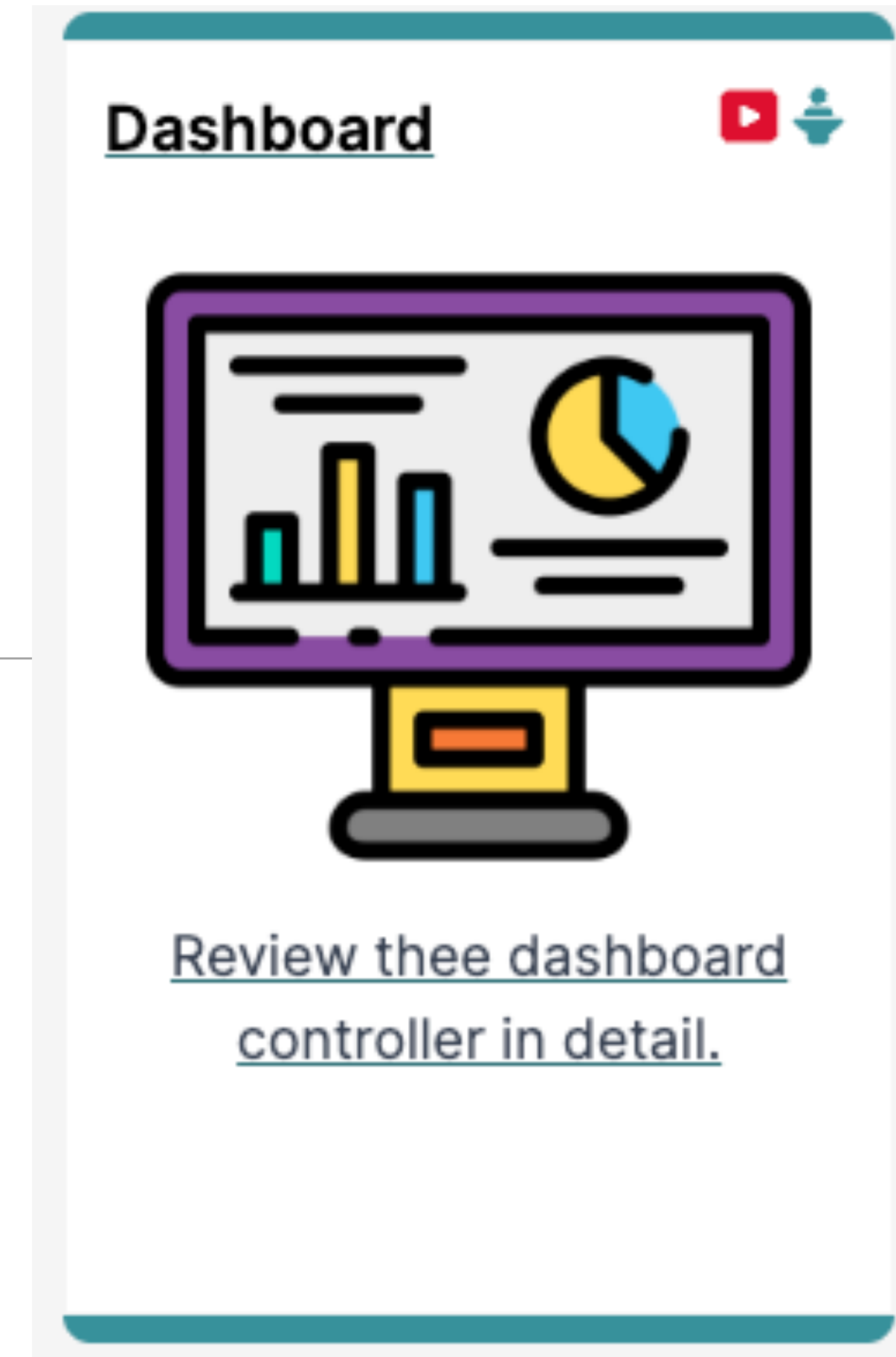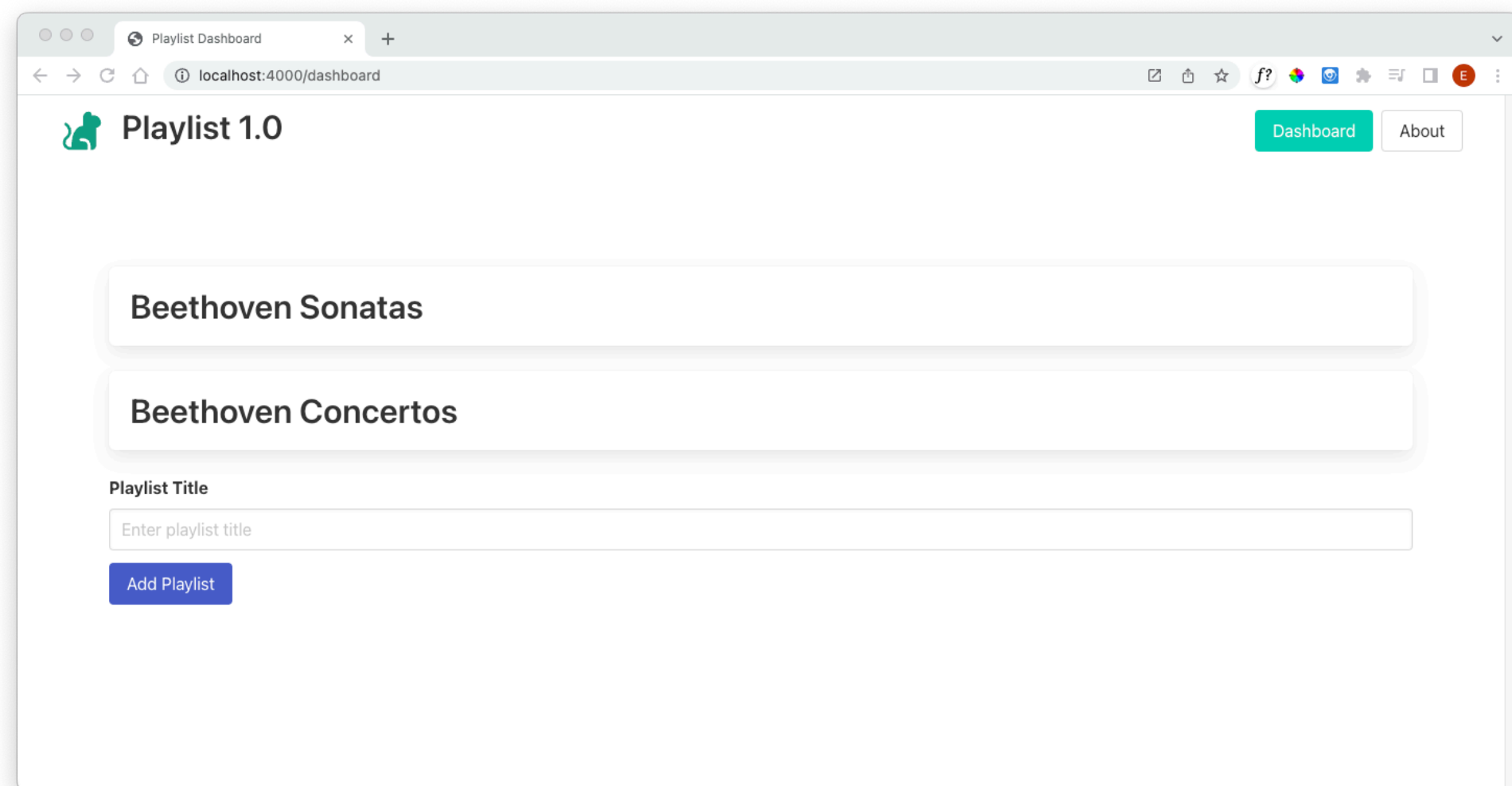# Dashboard Controller

Dashboard

Review thee dashboard controller in detail.

```javascript
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");
  },
};
```

Import playlistStore to access model

Define & export dashboardController object

index Action

addPlaylist Action

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



export const dashboardController = {



  async index(request, response) {
    ...
  },


  async addPlaylist(request, response) {
    ...
  },

};
```

Invoked when "/dashboard" route triggered

Create object to be sent to view

Log to console for debug purposes

Render "dashboard-view.hbs" passing viewData

```javascript
async index(request, response) {


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




  console.log("dashboard rendering");


  response.render("dashboard-view", viewData);

},
```

Retrieve all playlists from store

Invoked when
"/dashoard/addplayist" triggered

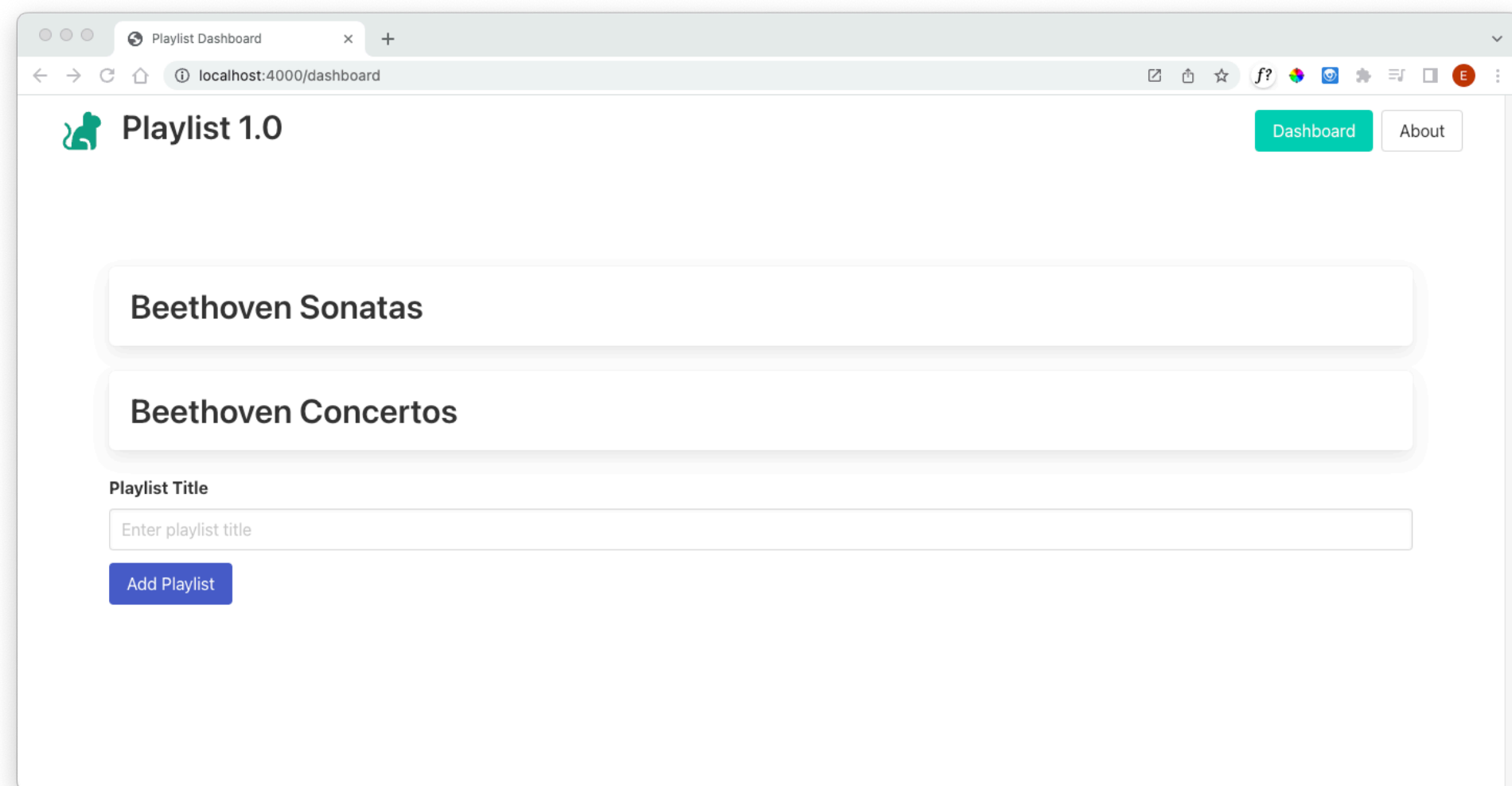Create a playlist object, using
the title passed in the request

Log to console for debug
purposes

Add new playlist object to the
datastore

Redirect to "/dashboard" route

```javascript
async addPlaylist(request, response) {


  const newPlaylist = {
    title: request.body.title,
  };


  console.log(`adding playlist ${newPlaylist.title}`);


  await playlistStore.addPlaylist(newPlaylist);


  response.redirect("/dashboard");

},
```

```javascript
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");
  },
};
```

Dashboard Controller

Dashboard

Review thee dashboard controller in detail.