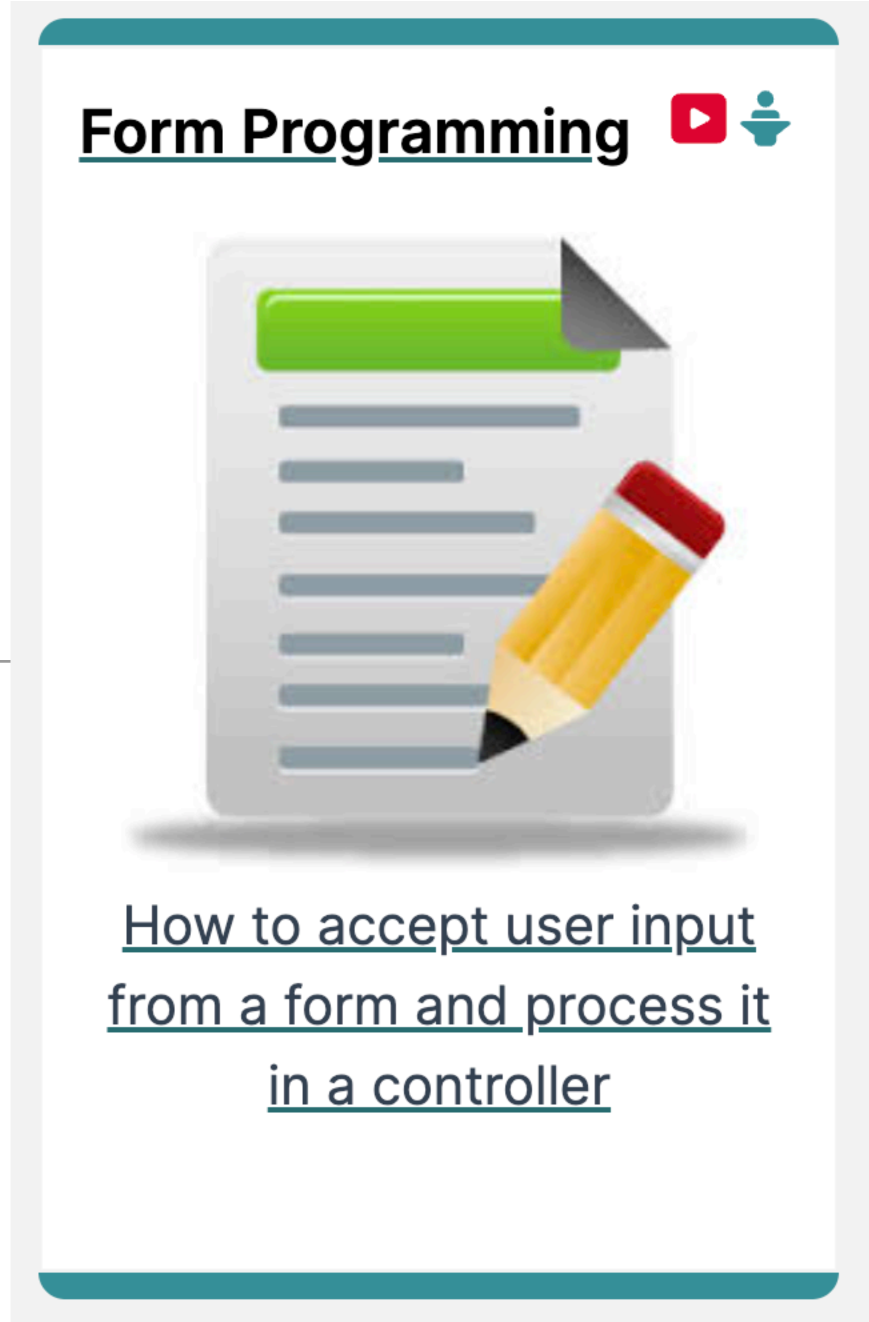
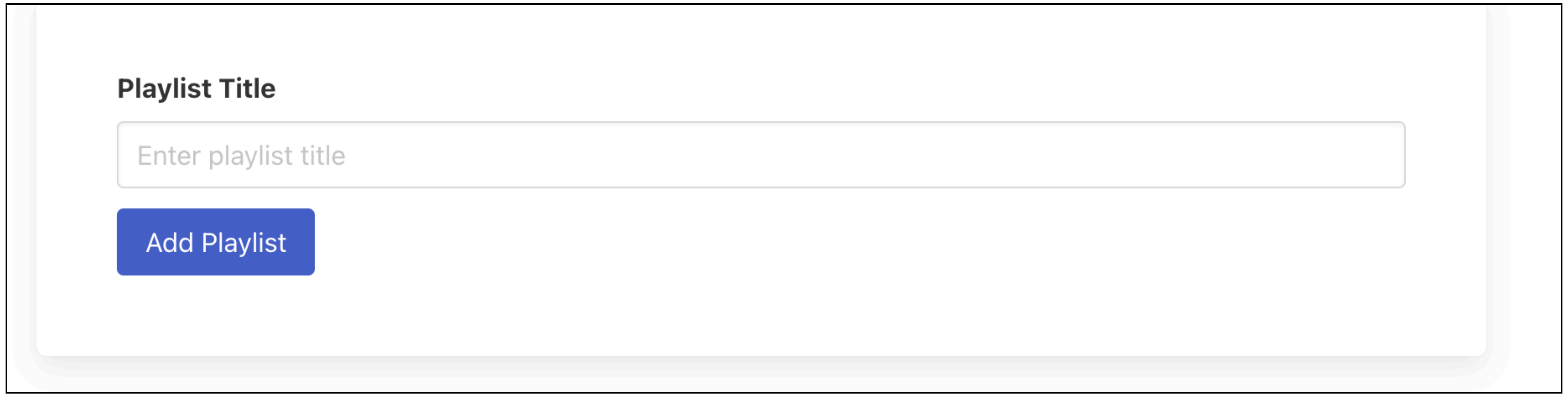
Form Programming



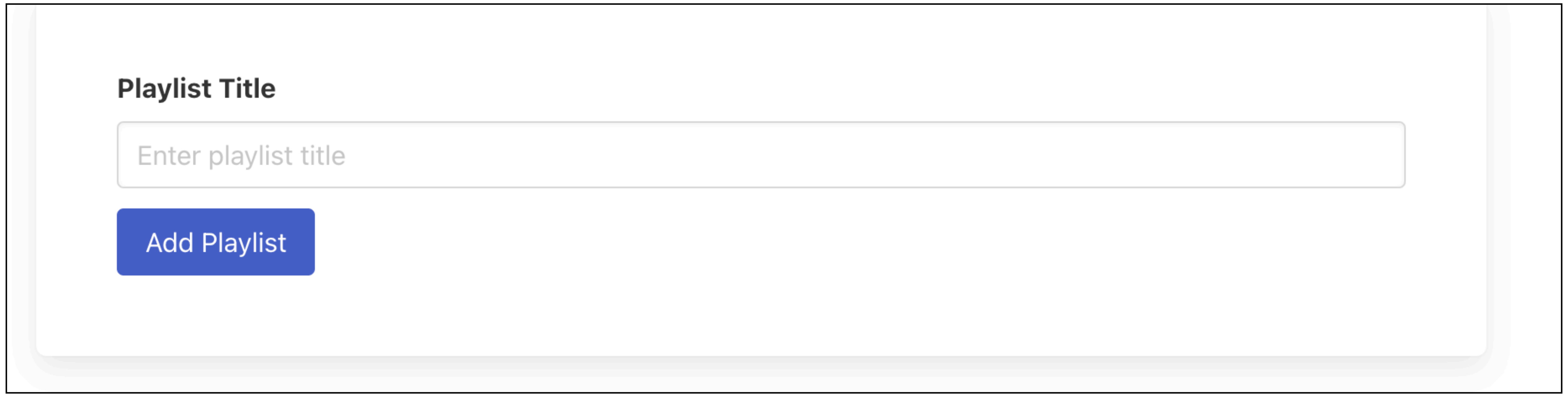
Form Programming

How to accept user input from a form and process it in a controller

Add a new Playlist



Add a new Playlist



```
<form class="section box" 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>
```

action → **"/dashboard/addplaylist"**

Field Value → **name="title"**

action → "/dashboard/addplaylist"

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

Router matches to action to a controller method

This method will be called when the ‘Submit’ button is pressed

action → "/dashboard/addplaylist"

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

controller method to create a new playlist

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

playlistController

Create a playlist object

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

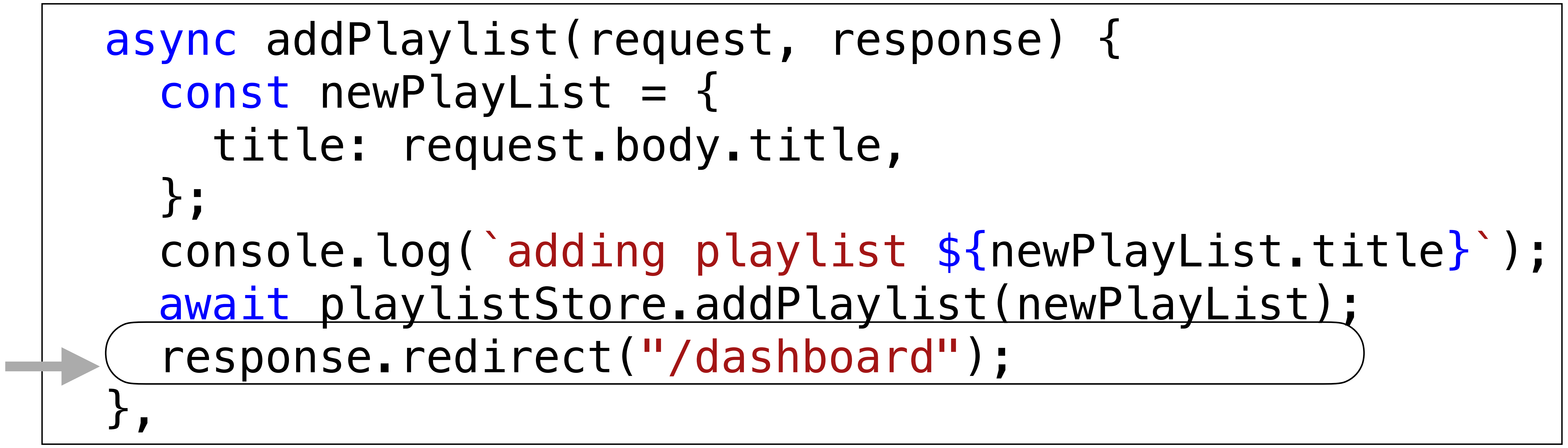
playlistController

Add to Playlist Store

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

playlistController

Refresh the Dashboard (which will now contain the new playlist)



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

playlistStore

playlistStore method to add a new playlist

```
const db = initStore("playlists");

export const playlistStore = {

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

  ...
}
```

playlistStore

Initialise Json file playlists.json

```
const db = initStore("playlists");

export const playlistStore = {

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

  ...
}
```

playlistStore

Load playlists from json file

```
const db = initStore("playlists");

export const playlistStore = {

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

  ...
}
```

playlistStore

Add unique ID to playlist →

```
const db = initStore("playlists");

export const playlistStore = {

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

  ...
}
```

playlistStore

Append to playlists array →

```
const db = initStore("playlists");

export const playlistStore = {

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

  ...
}
```

playlistStore

Write playlist array to json file

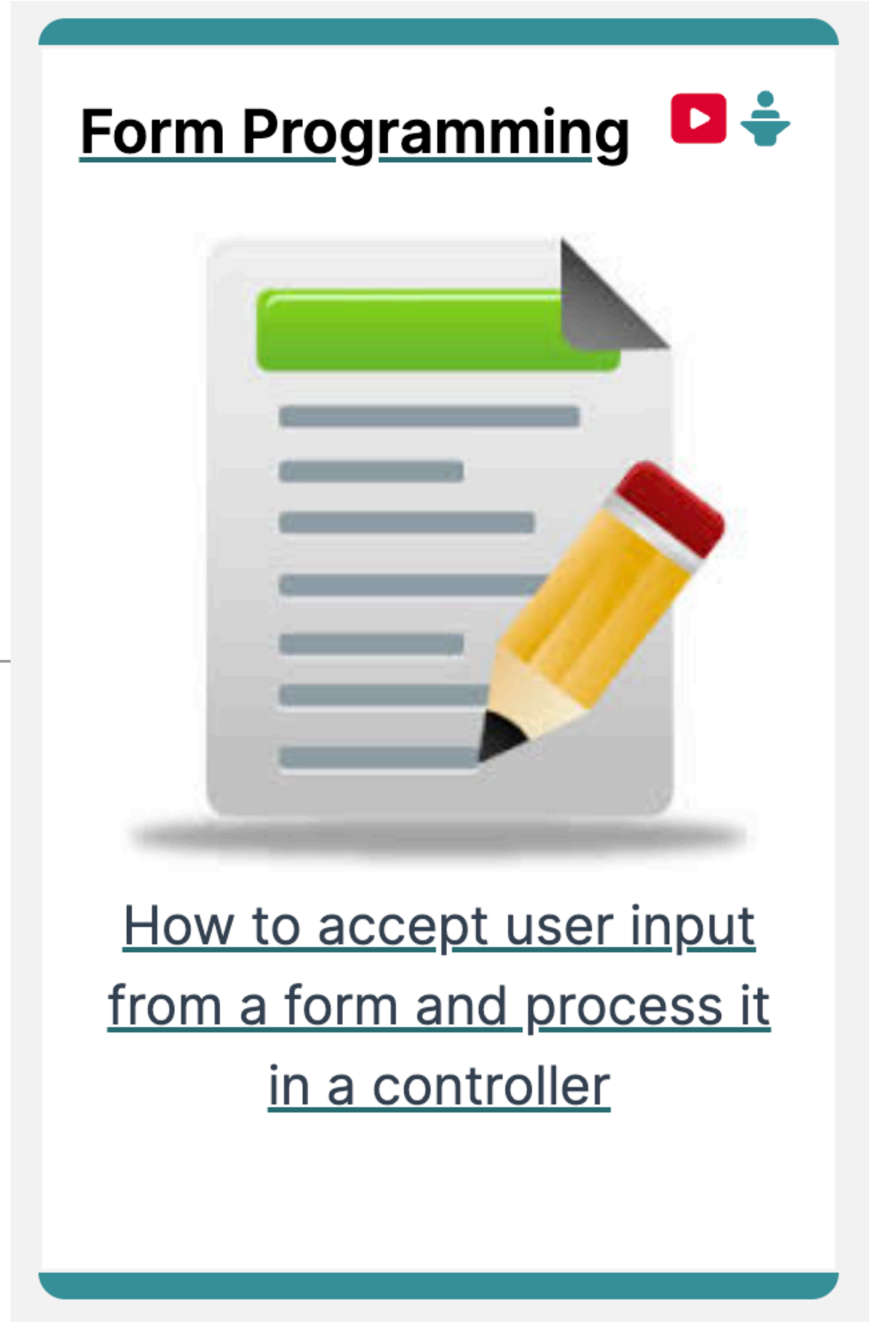
```
const db = initStore("playlists");

export const playlistStore = {

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

  ...
}
```

playlistStore

```
const db = initStore("playlists");

export const playlistStore = {

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

  ...
}
```

Return newly added playlist → return playlist;

Form Programming



Form Programming

How to accept user input from a form and process it in a controller