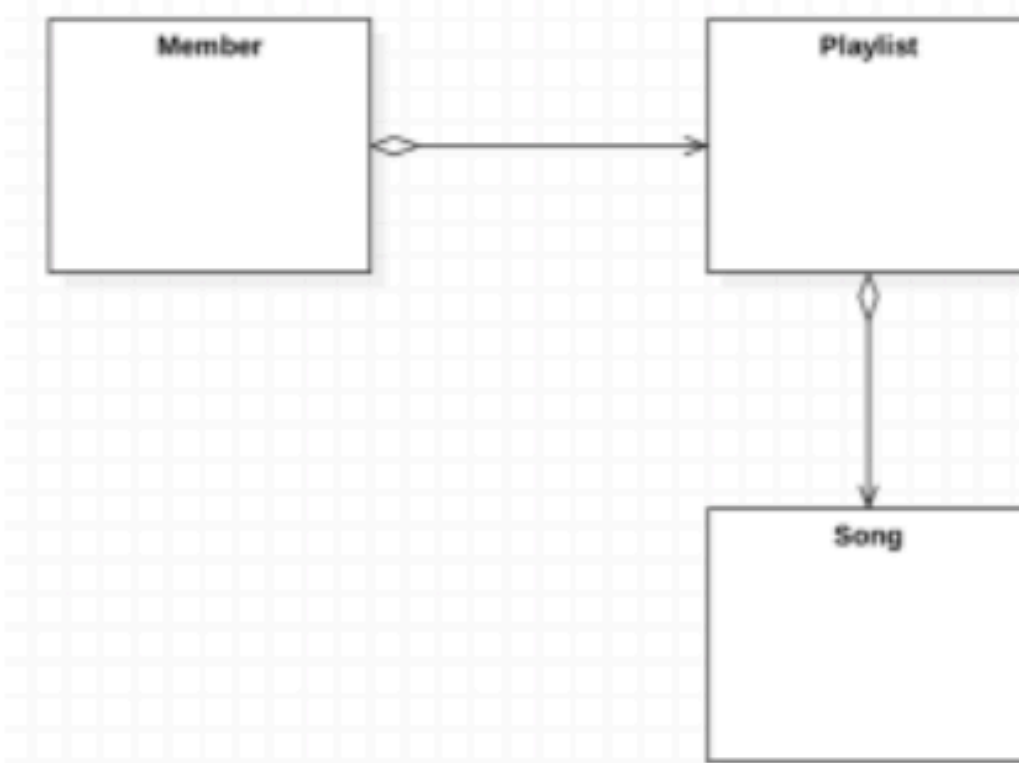
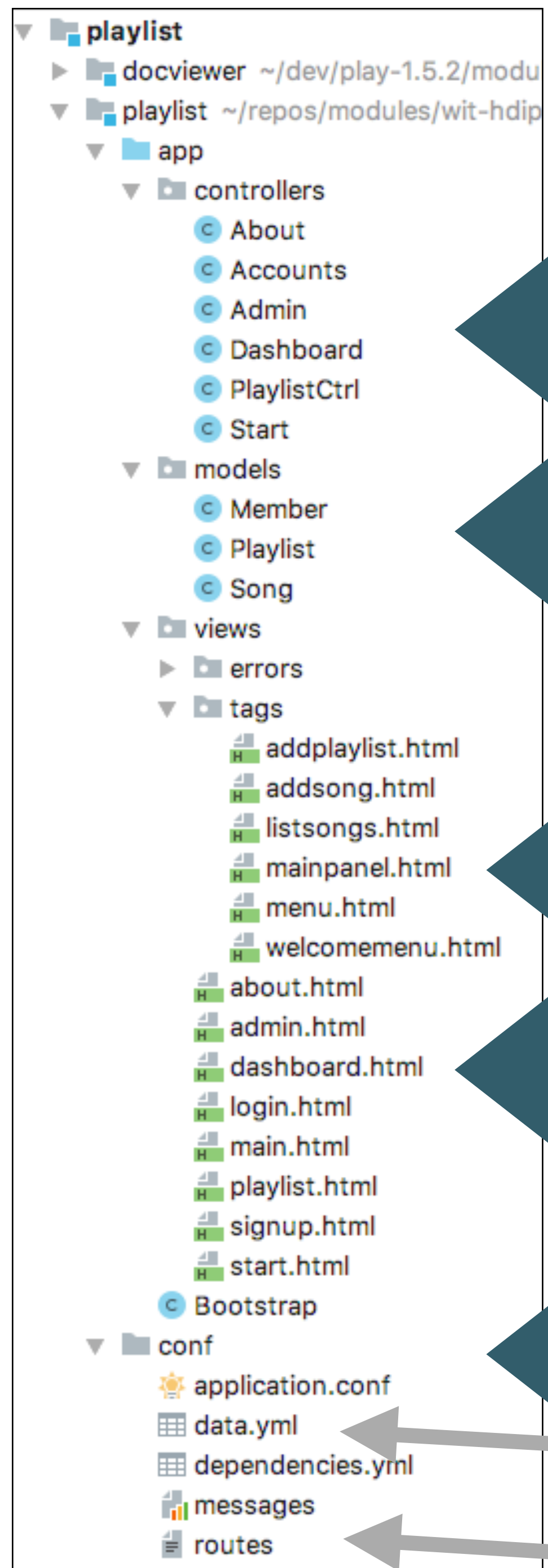


Playlist 5 - Sessions

Playlist Sessions



Introduce session management into the Playlist 4 Application



← Controllers

← Models

← Partials

← Views

← Configuration

← • Seed Data

← • Routes

Playlist 5

Entity Class: Member

```
@Entity
public class Member extends Model
{
    public String firstname;
    public String lastname;
    public String email;
    public String password;

    @OneToMany(cascade = CascadeType.ALL)
    public List<Playlist> playlists = new ArrayList<Playlist>();

    public Member(String firstname, String lastname, String email, String password)
    {
        this.firstname = firstname;
        this.lastname = lastname;
        this.email = email;
        this.password = password;
    }

    public static Member findByEmail(String email)
    {
        return find("email", email).first();
    }

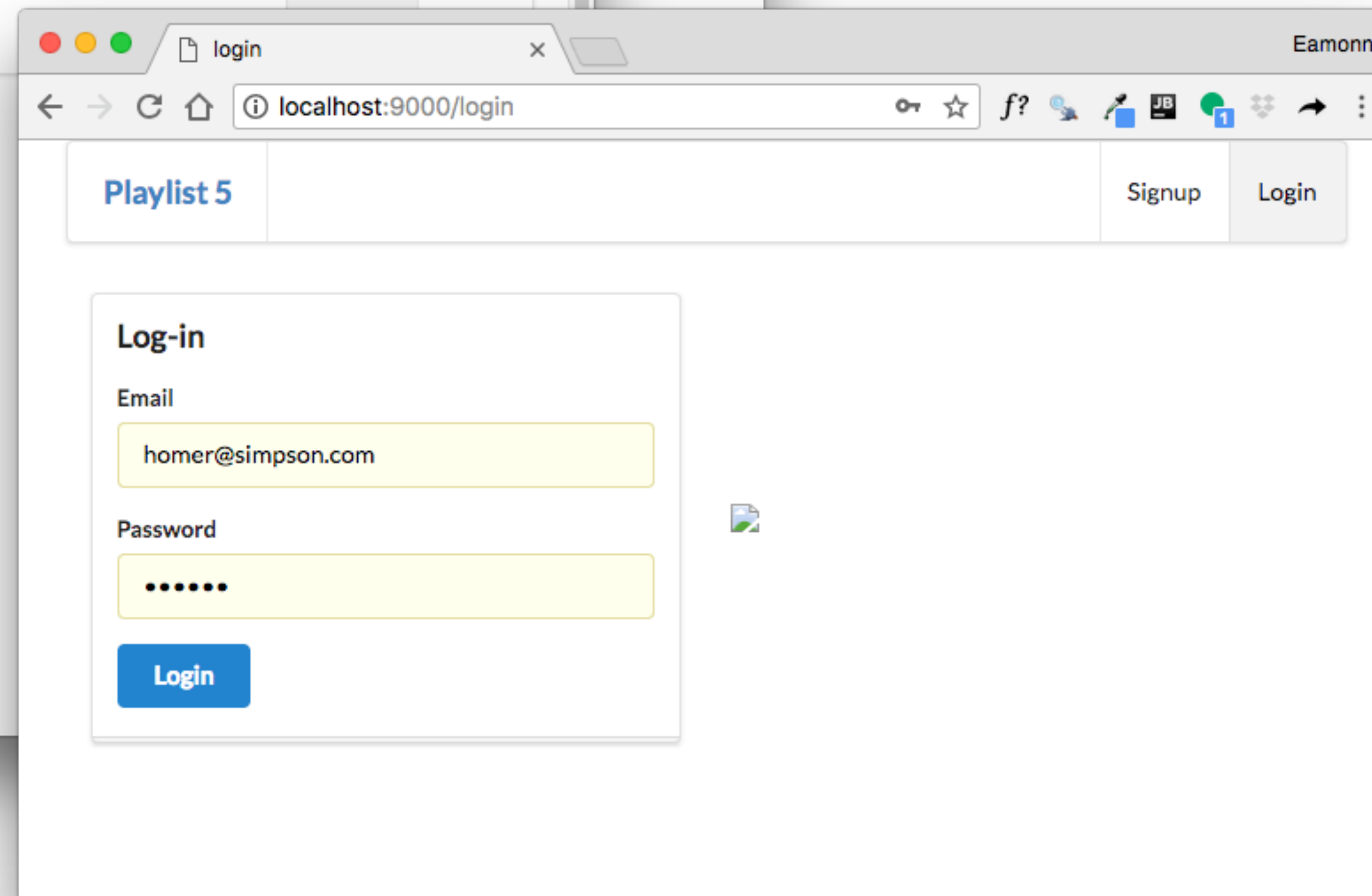
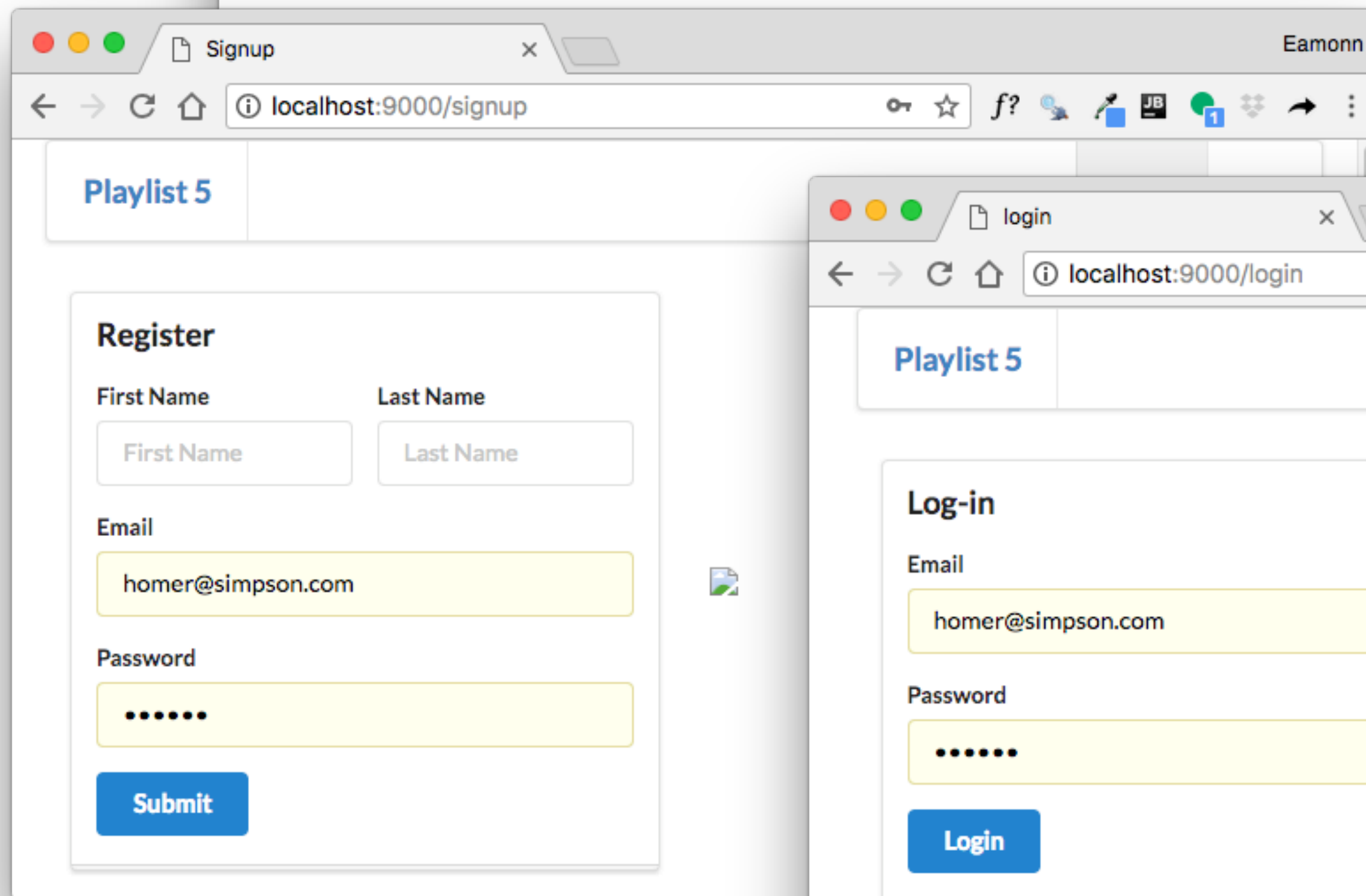
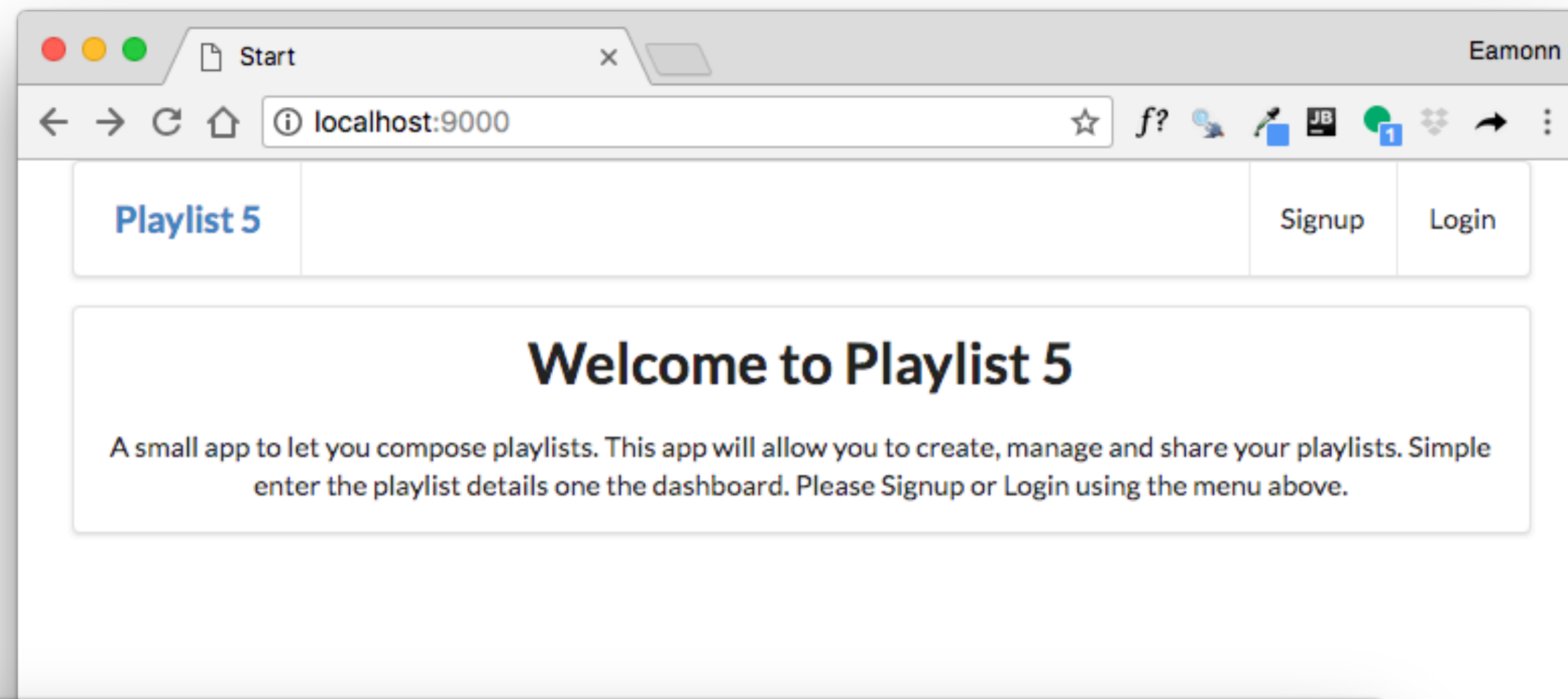
    public boolean checkPassword(String password)
    {
        return this.password.equals(password);
    }
}
```

data.yml

```
Member(m1):
  firstname: homer
  lastname: simpson
  email: homer@simpson.com
  password: secret

Member(m2):
  firstname: marge
  lastname: simpson
  email: marge@simpson.com
  password: secret
```

Introduce Signup/Login



Signup Form

Register

First Name

Last Name

Email

Password

```
<div class="ui two column grid basic middle aligned segment">
  <div class="column">
    <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>
  </div>
  <div class="column">
    
  </div>
</div>
```

Login Form

Log-in

Email

homer@simpson.com

Password

••••••••

Login

```
<div class="ui two column middle aligned grid basic segment">
  <div class="column">
    <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>
  </div>
  <div class="column">
    
  </div>
</div>
```

GET	/signup	Accounts.signup
GET	/login	Accounts.login

```
public class Accounts extends Controller
{
    public static void signup()
    {
        render("signup.html");
    }

    public static void login()
    {
        render("login.html");
    }
}
```

Register

First Name	Last Name
<input type="text" value="First Name"/>	<input type="text" value="Last Name"/>

Email

Password

Log-in

Email

Password

Initial Accounts Routes + Controller

Register

First Name

Last Name

Email

Password

Signup Member: Register

```
...  
<form class="ui stacked segment form" action="/register" method="POST">  
...  
</form>
```

POST	/register	Accounts.register
------	-----------	-------------------

```
public class Accounts extends Controller  
{  
...  
    public static void register(String firstname, String lastname, String email, String password)  
    {  
        Logger.info("Registering new user " + email);  
        Member member = new Member(firstname, lastname, email, password);  
        member.save();  
        redirect("/");  
    }  
...  
}
```


Log-in

Email

homer@simpson.com

Password

.....

Login

Login Member: Authenticate

POST /authenticate

Accounts.authenticate

```
...  
<form class="ui stacked segment form" action="/authenticate" method="POST">  
...
```

```
public class Accounts extends Controller  
{  
    public static void authenticate(String email, String password)  
    {  
        Logger.info("Attempting to authenticate with " + email + ":" + password);  
  
        Member member = Member.findByEmail(email);  
        if ((member != null) && (member.checkPassword(password) == true)) {  
            Logger.info("Authentication successful");  
            session.put("logged_in_Memberid", member.id);  
            redirect("/dashboard");  
        } else {  
            Logger.info("Authentication failed");  
            redirect("/login");  
        }  
    }  
}
```

Login Member: Authenticate

Find Member in DB

If member found &
passwords match

Store members id in
the Session object

Render Dashboard

Back to login

```
public class Accounts extends Controller
{
    public static void authenticate(String email,
                                   String password)
    {
        Member member = Member.findByEmail(email);

        if ((member != null) &&
            (member.checkPassword(password) == true)) {

            session.put("logged_in_Memberid", member.id);

            redirect ("/dashboard");
        } else {
            redirect("/login");
        }
    }
}
```

Utility Method: getLoggedInMember()

```
public class Accounts extends Controller
{
    ...
    public static Member getLoggedInMember()
    {
        Member member = null;
        if (session.contains("logged_in_Memberid")) {
            String memberId = session.get("logged_in_Memberid");
            member = Member.findById(Long.parseLong(memberId));
        } else {
            login();
        }
        return member;
    }
    ...
}
```

Read the session object, and locate the member id

Find Member in database

Return the member

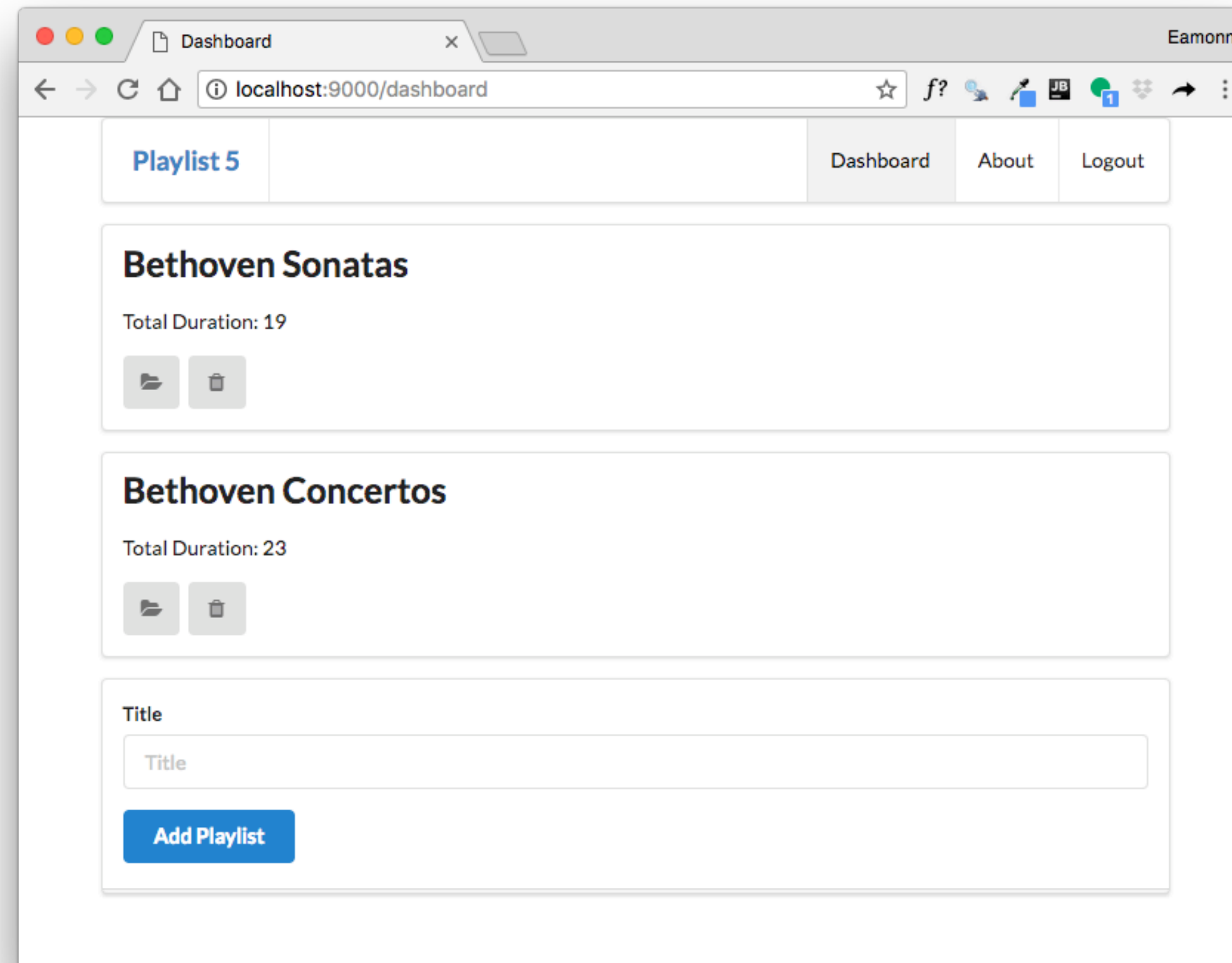
If no member found, switch to login screen

Revised Dashboard Index

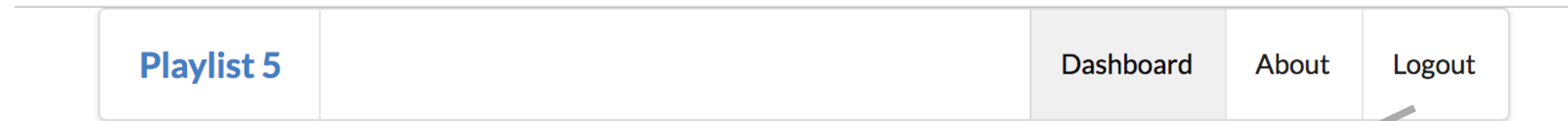
Retrieved the
logged in member

Send the
member's playlist
to the view

```
public class Dashboard extends Controller
{
    public static void index()
    {
        Logger.info("Rendering Dashboard");
        Member member = Accounts.getLoggedInMember();
        List<Playlist> playlists = member.playlists;
        render("dashboard.html", playlists);
    }
    ...
}
```



Accounts logout() action



GET /logout Accounts.logout

```
public static void logout()  
{  
    session.clear();  
    redirect ("/");  
}
```

Clear the session
Member id will no longer be found

Inspect the Session Cookie in Chrome

The screenshot shows a Chrome browser window with a dashboard page. The dashboard has a navigation bar with "Playlist 5", "Dashboard", "About", and "Logout". Below the navigation bar are two sections: "Bethoven Sonatas" with a total duration of 19, and "Bethoven Concertos" with a total duration of 23. Each section has a trash icon. The developer tools are open to the "Application" tab, showing a table of cookies. The table has columns for Name, Value, Domain, Path, Expire..., Size, HTTP, Secure, and Same... The first row shows a cookie named "PLAY_SESSION" with a long alphanumeric value, domain "localh...", and path "/".

Name	Value	Domain	Path	Expire...	Size	HTTP	Secure	Same...
PLAY_SESSION	bb066f6aba69f65a449dce74297f68b2ca3cc75...	localh...	/	1969-...	73			

Logout clears the Cookie

The screenshot shows a web browser window with the URL `localhost:9000`. The page displays a header with a "Playlist 5" link and "Signup" and "Login" buttons. The main content area features a "Welcome to Playlist 5" heading and a paragraph: "A small app to let you compose playlists. This app will allow you to create, manage and share your playlists. Simple enter the playlist details one the dashboard. Please Signup or Login using the menu above."

Below the browser window, the Chrome DevTools Application tab is open, showing the Cookies table for `http://localhost:9000`. The table is empty, indicating that cookies have been cleared.

Name	Value	Domain	Path	Expire...	Size	HTTP	Secure	Same...
------	-------	--------	------	-----------	------	------	--------	---------