

Playlist 5 - Sessions



Introduce Signup/Login

Start

localhost:9000

Playlist 5

Signup Login

Welcome to Playlist 5

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.

login

localhost:9000/login

Playlist 5

Signup Login

Log-in

Email

Password

Login

Signup

localhost:9000/signup

Playlist 5

Signup Login

Register

First Name

Last Name

Email

Password

Submit

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");
    }
}
```

Log-in

Email

homer@simpson.com

Password

•••••

Login

Register

First Name

First Name

Last Name

Last Name

Email

homer@simpson.com

Password

•••••

Submit

Initial Accounts
Routes + Controller

Register

First Name

Last Name

Email

Password

Submit

Signup Member: Register

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

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

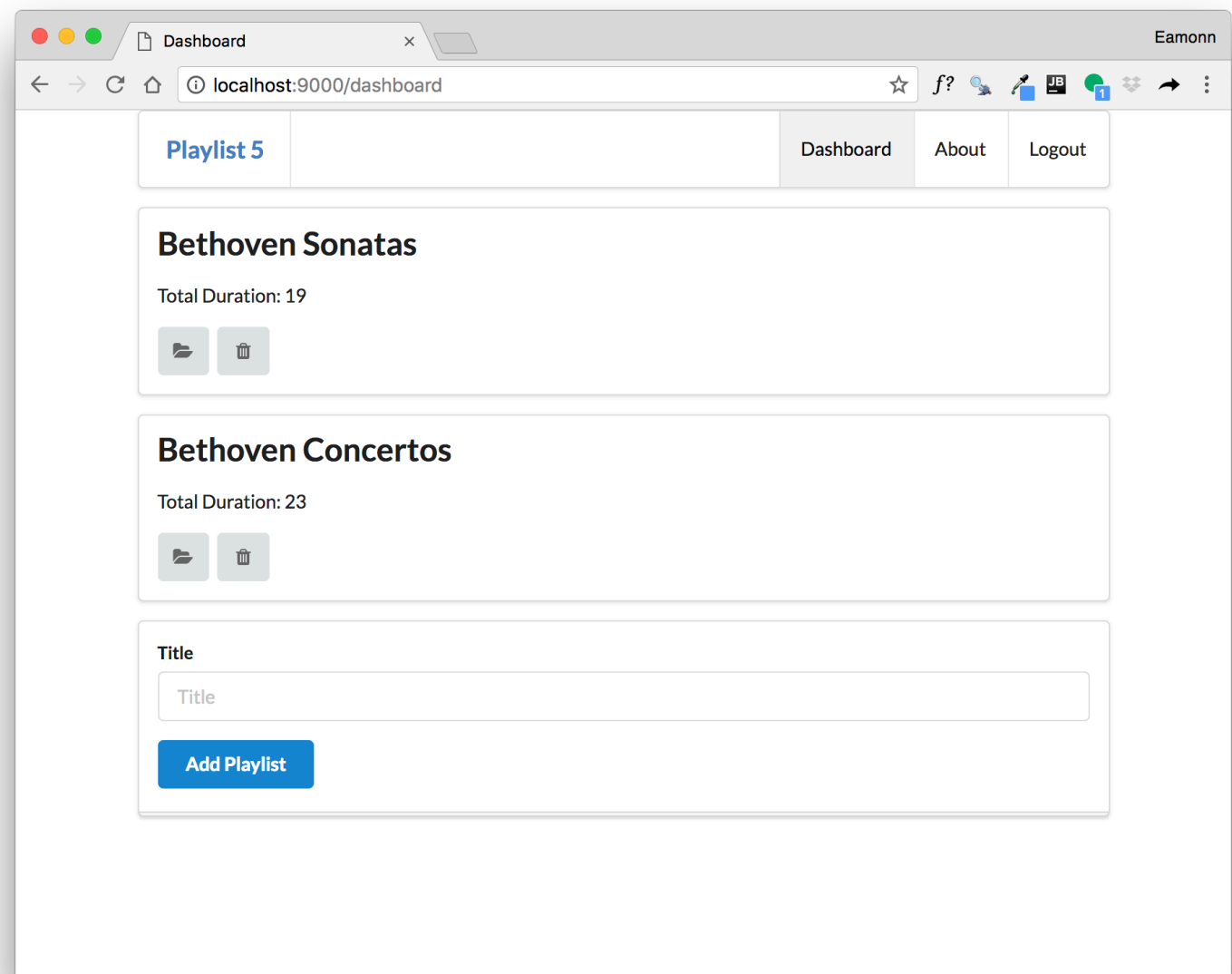
Return the member (or null)

Revised Dashboard Index

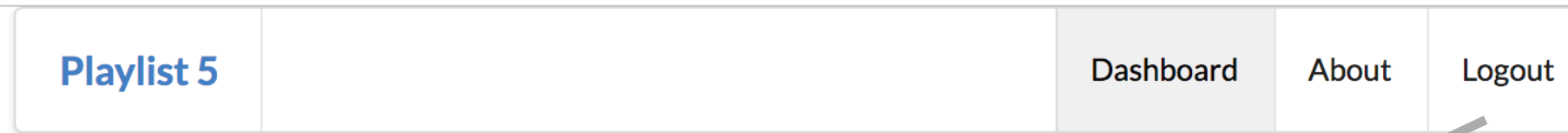
Retrieved the
logged in member

```
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);
    }
    ...
}
```

Send the
member's playlist
to the view



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 web browser window with a tab titled "Dashboard". The address bar displays "localhost:9000/dashboard". The page content includes a navigation bar with "Playlist 5", "Dashboard", "About", and "Logout". Below the navigation bar, there are two sections: "Bethoven Sonatas" with a total duration of 19, and "Bethoven Concertos" with a total duration of 23. Each section has a folder icon and a trash icon. At the bottom of the page, there is a "Title" input field.

The Chrome DevTools Application tab is open, showing a table of cookies. The table has columns: Name, Value, Domain, Path, Expires..., Size, HTTP, Secure, and SameSite. The first row shows a cookie named "PLAY_SESSION" with a value "3edda631b40926bac7c435313050b60171943fa1-lo...", domain "localhost", and path "/".

Name	Value	Domain	Path	Expires...	Size	HTTP	Secure	SameSite
PLAY_SESSION	3edda631b40926bac7c435313050b60171943fa1-lo...	localhost	/	1969-1...	74			

Logout clears the Cookie

The screenshot shows a web browser window with a single tab titled 'Start'. The address bar displays 'localhost:9000'. The web application has 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 explaining the app's purpose: '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.'

The browser's developer tools are open, showing the 'Application' tab. The left sidebar lists 'Manifest', 'Service Workers', and 'Clear storage' under 'Application', and 'Local Storage', 'Session Storage', 'IndexedDB', 'Web SQL', and 'Cookies' under 'Storage'. The 'Cookies' section is expanded, showing a single cookie for 'http://localhost:9000'. The main table in the 'Application' tab is empty, with columns for Name, Value, Domain, Path, Expires..., Size, HTTP, Secure, and SameSite.

At the bottom, the 'Console' tab is active, showing a single log entry with a blue arrow icon. The console toolbar includes a 'top' button, a 'Filter' input, 'Default levels', and a checked 'Group similar' checkbox.