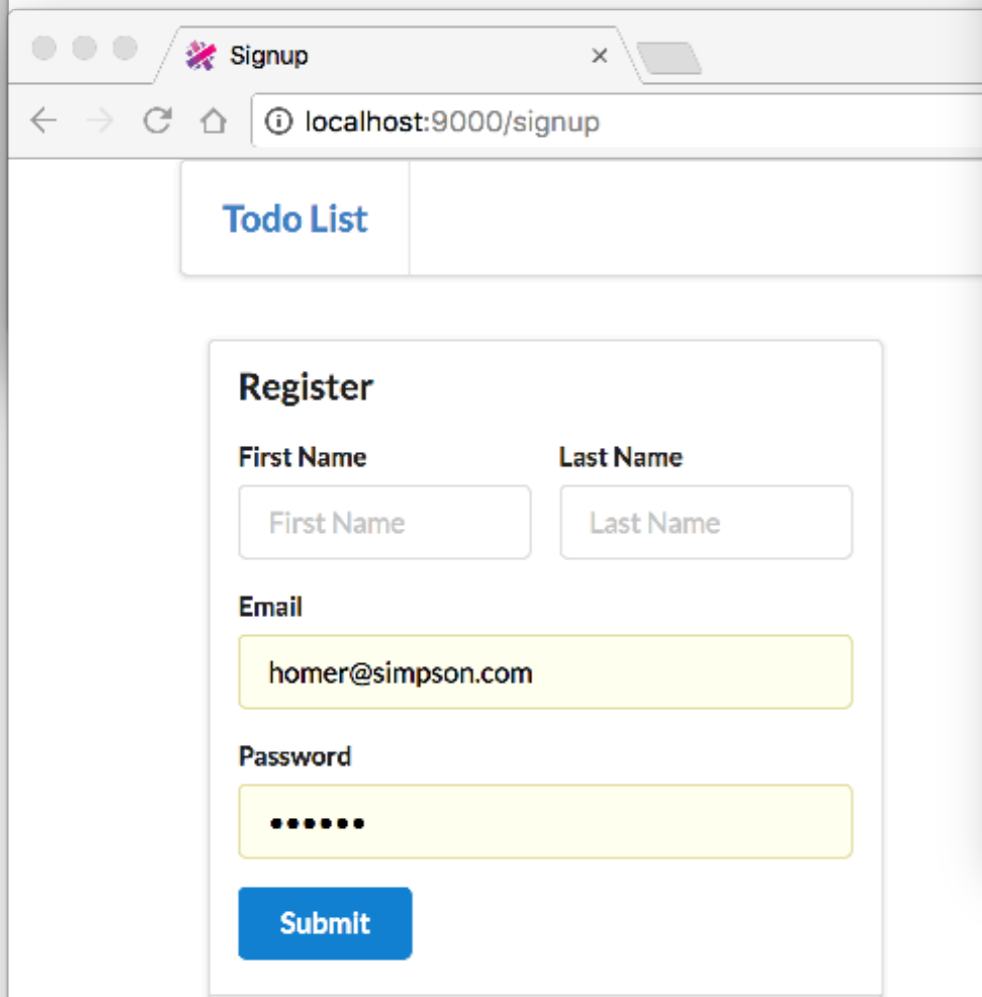
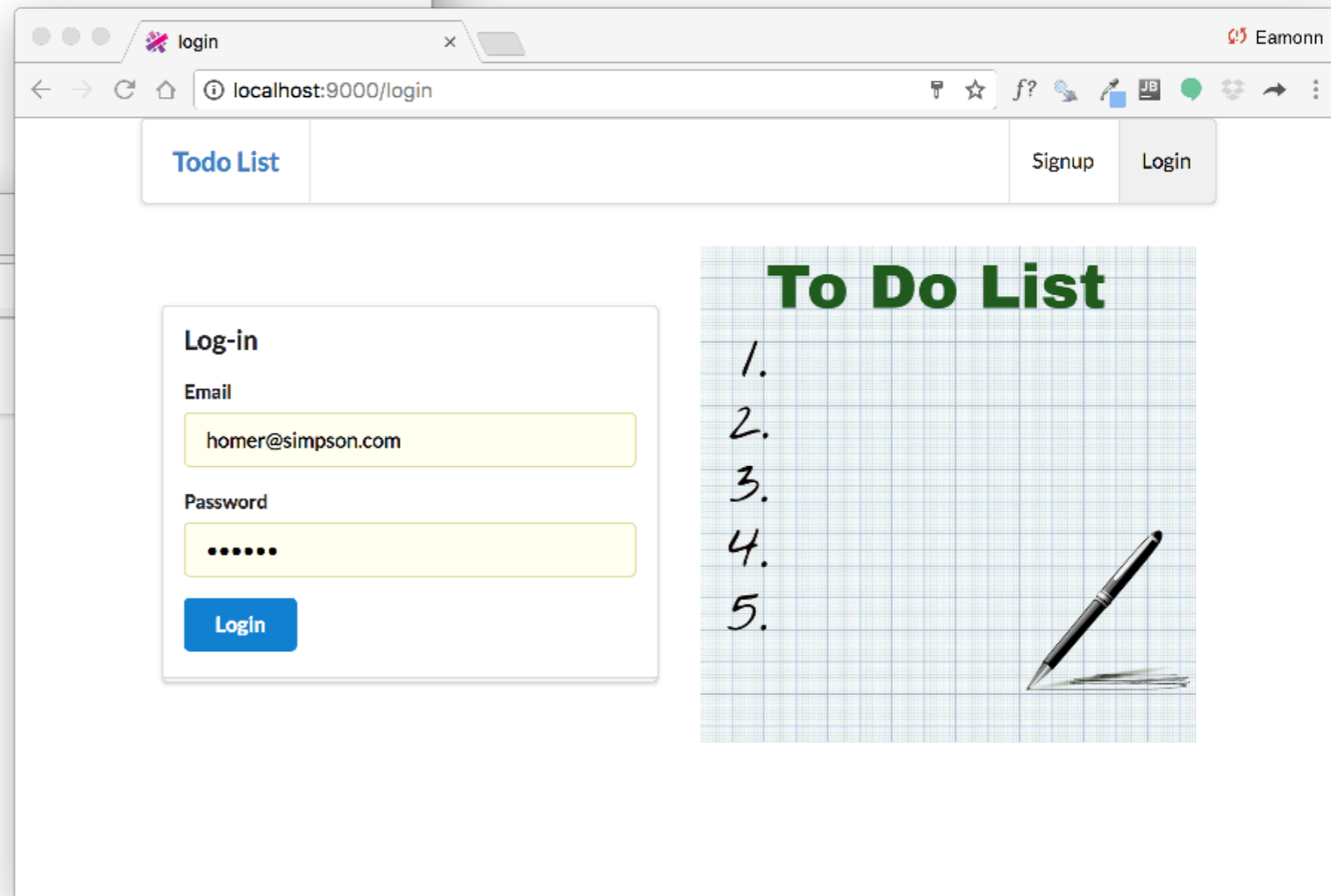
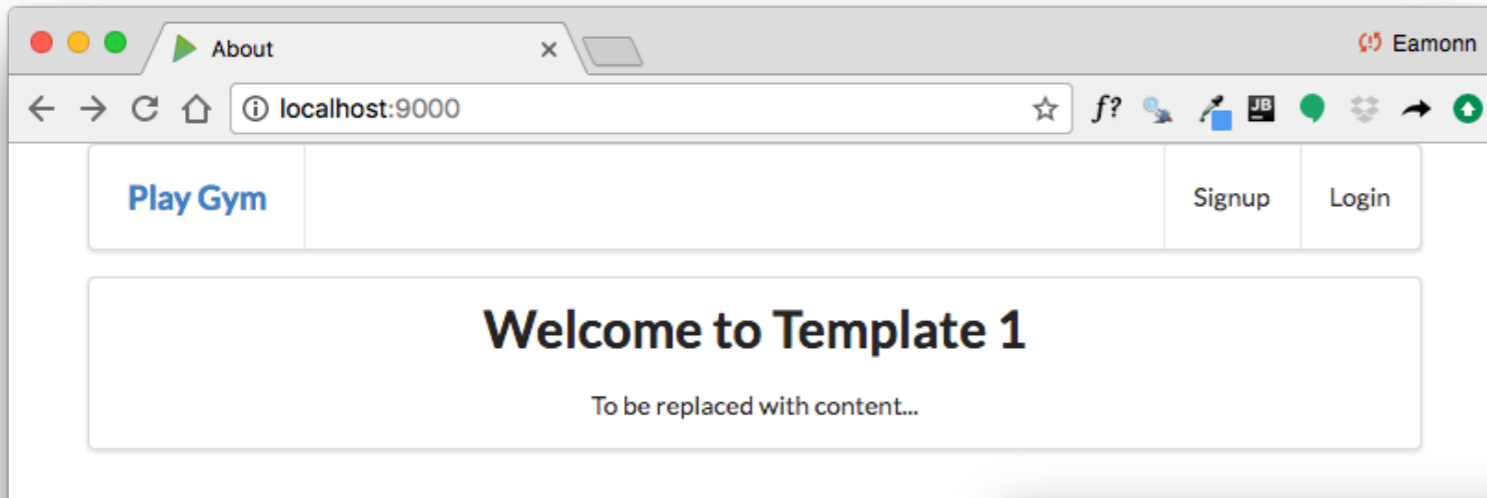


Todo Sessions

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
GET /login
```

```
Accounts.signup
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

Email

Password

Log-in

Email

Password

Initial Accounts
Routes + Controller

Todo Model

```
@Entity
public class Todo extends Model
{
    public String title;

    public Todo(String title)
    {
        this.title = title;
    }
}
```

```
Todo(t1):
    title: Make tea
```

```
Todo(t2):
    title: Go for snooze
```

```
Todo(t3):
    title: Make more tea
```

Member Model

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

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

    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);
    }
}
```

```
Todo(t1):
  title: Make tea

Todo(t2):
  title: Go for snooze

Todo(t3):
  title: Make more tea

Member(m1):
  firstname: homer
  lastname: simpson
  email: homer@simpson.com
  password: secret
  todolist:
  - t1
  - t2
```

```

Todo(t1):
  title: Make tea

Todo(t2):
  title: Go for snooze

Todo(t3):
  title: Make more tea

Member(m1):
  firstname: homer
  lastname: simpson
  email: homer@simpson.com
  password: secret
  todolist:
  - t1
  - t2

```

```
SELECT * FROM TODO;
```

ID	TITLE
1	Make tea
2	Go for snooze
3	Make more tea

(3 rows, 2 ms)

```
SELECT * FROM MEMBER;
```

ID	EMAIL	FIRSTNAME	LASTNAME	PASSWORD
1	homer@simpson.com	homer	simpson	secret

(1 row, 3 ms)

```
SELECT * FROM MEMBER_TODO;
```

MEMBER_ID	TODOLIST_ID
1	1
1	2

(2 rows, 1 ms)

Auto

- jdbc:h2:mem:play
- member
- member_todo
- todo
- information_schema
- Sequences
- Users

H2 1.4.193 (2016-10-31)

```

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

  @OneToMany(cascade = CascadeType.ALL)
  public List<Todo> todolist = new ArrayList<Todo>();
}

```

```

@Entity
public class Todo extends Model
{
  public String title;
}

```

Database

Register

First Name

Last Name

Email

Password

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 Actions

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

    public static void addTodo(String title)
    {
        Member member = Accounts.getLoggedInMember();
        Todo todo = new Todo(title);
        member.todolist.add(todo);
        member.save();
        Logger.info("Adding Todo" + title);
        redirect("/dashboard");
    }

    public static void deleteTodo(Long id, Long todoId)
    {
        Member member = Member.findById(id);
        Todo todo = Todo.findById(todoId);
        member.todolist.remove(todo);
        member.save();
        todo.delete();
        Logger.info("Deleting " + todo.title);
        redirect("/dashboard");
    }
}
```

Dashboard index()

[Todo List](#) [Dashboard](#) [About](#) [Logout](#)

homer simpson's Todo List

Todo	
Make tea	Delete
Go for snooze	Delete

Title


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

Only send Members Todo list to view

Dashboard addTodo()



homer simpson's Todo List

Todo	
Make tea	Delete
Go for snooze	Delete

Title

Add Todo

```
public static void addTodo(String title)
{
    Member member = Accounts.getLoggedInMember();
    Todo todo = new Todo(title);
    member.todoList.add(todo);
    member.save();
    Logger.info("Adding Todo" + title);
    redirect("/dashboard");
}
```

Add todo to the member list

Dashboard deleteTodo()

Dashboard navigation: [Dashboard](#) | [About](#) | [Logout](#)

homer simpson's Todo List

Todo
Make tea Delete
Go for snooze Delete

Title


```
public static void deleteTodo(Long id, Long todoId)
{
    Member member = Member.findById(id);
    Todo todo = Todo.findById(todoId);
    member.todoList.remove(todo);
    member.save();
    todo.delete();
    Logger.info("Deleting " + todo.title);
    redirect("/dashboard");
}
```

Remove the todo from the members list

Accounts logout() action

Todo List

Dashboard

About

Logout

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 tab titled 'Dashboard' at the URL 'localhost:9000/dashboard'. The page content includes a navigation bar with 'Dashboard', 'About', and 'Logout' links. Below this is a section titled 'homer simpson's Todo List' containing two items: 'Make tea' and 'Go for snooze', each with a red 'Delete' button. A 'Title' input field is also visible.

The Chrome DevTools 'Application' tab is open, displaying a table of cookies. The 'PLAY_SESSION' cookie is selected, showing its value and other attributes.

Name	Value	Do...	Path	Exp...	Size	HTTP	Secure	SameS...
PLAY_SESSION	1813c9cb72f6f7b3954da2ce827b1fdee94972cc-logged_in_Mem...	loc...	/	Ses...	73			

Logout clears the Cookie

The image shows a web browser window with a single tab titled 'About'. The address bar displays 'localhost:9000'. The page content includes a 'Todo List' link, 'Signup' and 'Login' buttons, and a large central box with the text 'Welcome to Template 1' and 'To be replaced with content...'. Below the browser window, the Chrome DevTools 'Application' tab is open. The left sidebar shows the 'Storage' section expanded to 'Cookies', with a sub-entry for 'http://localhost:9000'. The main content area of the DevTools tab displays the message 'This site has no cookies.' in a large, bold, grey font.