

Web Development

Produced
by

Eamonn de Leastar (edeleastar@wit.ie)

Department of Computing, Maths & Physics
Waterford Institute of Technology

<http://www.wit.ie>

<http://elearning.wit.ie>



Waterford Institute of Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRCE



Status & Members List

Save Status Message


- If the user types something into the edit field, and presses 'change' - then the text entered will appear just above the email.

Spacebook Home Members **Profile** Logout

Homer Simpson's Profile

(homer@simpson.com)

Profile Image



Upload your file:
 No file chosen

Status Text

Enter text:

Current User Class

```
public class User extends Model
{
    public String firstName;
    public String lastName;
    public String email;
    public String password;

    public User(String firstName, String lastName, String email, String password)
    {
        this.firstName = firstName;
        this.lastName = lastName;
        this.email = email;
        this.password = password;
    }

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

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

Change 1

- Create a new member in the 'User' class to store the text - we will call it 'statusMessage'

```
public class User extends Model
{
    //...
    public String statusMessage;
    //...
}
```

Change 2

- Introduce a 'route' that maps the pressing of the change button to a method called 'changeText' in the HomeProfile controller:

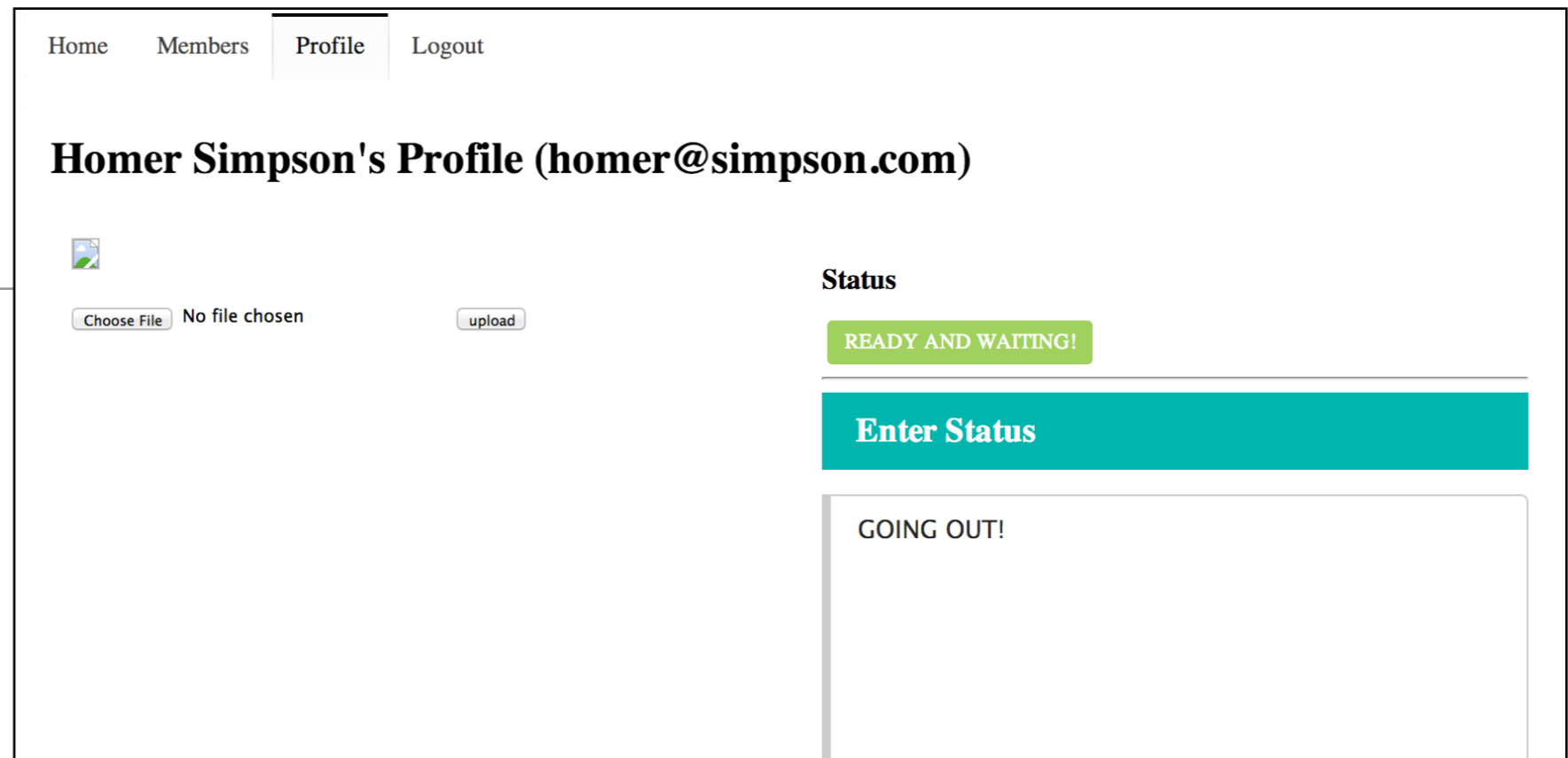
POST	/profile/changestatus	Profile.changeStatus
------	-----------------------	----------------------

Change 3

- Replace the `changeText` method in the `HomeProfile` class with the following method which:
 - determines who is currently logged in
 - reads the details about that user from the database into the user object
 - replaces the current status with the new status the user has just entered
 - saves this to the database
 - prints a log message
 - redraw the view

```
public static void changeStatus(String profiletext)
{
    String userId = session.get("logged_in_userid");
    User user = User.findById(Long.parseLong(userId));
    user.statusText = profiletext;
    user.save();
    Logger.info("Status changed to " + profiletext);
    index();
}
```

Change 4



```
<div class="ui column form segment">
  <h2 class="ui small header"> Status</h2>
  <div class="ui green block label">
    <p> ${user.statusText} </p>
  </div>
  <hr>
  <form action="/profile/changestatus" method="post">
    <h3 class="ui inverted teal block header">Enter Status</h3>
    <textarea class="ui field" name="profiletext"> </textarea>
    <input class="ui blue button" type="submit" name="submit" value="Change" />
  </form>
</div>
```

views/Profile/index.html

Display Members List

```
<section class="ui segment">
  <h2 class="ui header">SpaceBook's Members</h2>
  <div class="ui list">
    <div class="item">
      <i class="right triangle icon"></i> marge [<a href="home.html">follow</a>]
    </div>
    <div class="item">
      <i class="right triangle icon"></i> bart [<a href="home.html">follow</a>]
    </div>
    <div class="item">
      <i class="right triangle icon"></i> lisa [<a href="home.html">follow</a>]
    </div>
    <div class="item">
      <i class="right triangle icon"></i> maggie [<a href="home.html">follow</a>]
    </div>
  </div>
</section>
```

Home **Members** Profile Logout

SpaceBook's Members

- ▶ marge [[follow](#)]
- ▶ bart [[follow](#)]
- ▶ lisa [[follow](#)]
- ▶ maggie [[follow](#)]

```
public class Members extends Controller
{
  public static void index()
  {
    render();
  }

  public static void follow (String name)
  {
    Logger.info("Following " + name);
    index();
  }
}
```

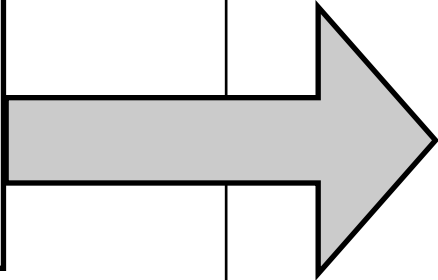
- List is 'hard coded'. i.e. never changes.

Reading Members from Database

- Ask User class to 'findAll' members
- Returns a list of all members in database
- Pass this to the view

```
public class Members extends Controller
{
    public static void index()
    {
        render();
    }

    public static void follow(String name)
    {
        Logger.info("Following " + name);
    }
}
```



```
public static void index()
{
    List<User> users = User.findAll();
    render(users);
}
```

Rendering the list in the View

```
<div class="ui list">
  #{list items:users, as:'user'}
  <div class="item">
    <i class="right triangle icon"></i> ${user.firstName} ${user.lastName} </a>
  </div>
  #{/list}
</div>
```

- Write a loop in HTML!
- Not really, the loop is written in the play templating language
- Close in syntax and semantics to the java 'foreach' loop
- However, remains challenging to read and interpret

*“for all of the user objects in the users list, visit each one and call it ‘user’.
With each of these user objects, create a element to contain the user’s first name and last name”*

Home

Members

Profile

Logout

SpaceBook's Members

- ▶ Homer Simpson
- ▶ Marge Simpson

```
<div class="ui list">
  #{list items:users, as:'user'}
  <div class="item">
    <i class="right triangle icon"></i> ${user.firstName} ${user.lastName} </a>
  </div>
  #{/list}
</div>
```

- Now, would like to reintroduce 'follow' link...

Home

Members

Profile

Logout

SpaceBook's Members

- ▶ Homer Simpson ([follow](#))
- ▶ Marge Simpson ([follow](#))

Follow Link

```
<div class="ui list">
  #{list items:users, as:'user'}
  <div class="item">
    <i class="right triangle icon"></i> ${user.firstName} ${user.lastName} <a href="/members/follow/${user.id}"> (follow) </a>
  </div>
#{/list}
</div>
```

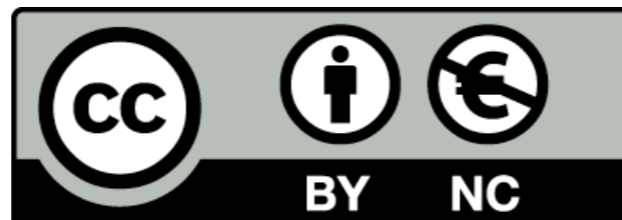
GET	/members/follow/{id}	Members.follow
-----	----------------------	----------------

```
public static void follow(Long id)
{
  Logger.info("Following " + id);
  index();
}
```

Follow Database Lookup

```
public static void follow(Long id)
{
    User userToFollow = User.findById(id);
    Logger.info("Following " + userToFollow.firstName);
    index();
}
```

- Use the id to search the database
- Get the user object from the database
- Log this to the console to make sure it works



Except where otherwise noted, this content is licensed under a Creative Commons Attribution-NonCommercial 3.0 License.

For more information, please see <http://creativecommons.org/licenses/by-nc/3.0/>

