

# 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 FHORT LÁIRCE

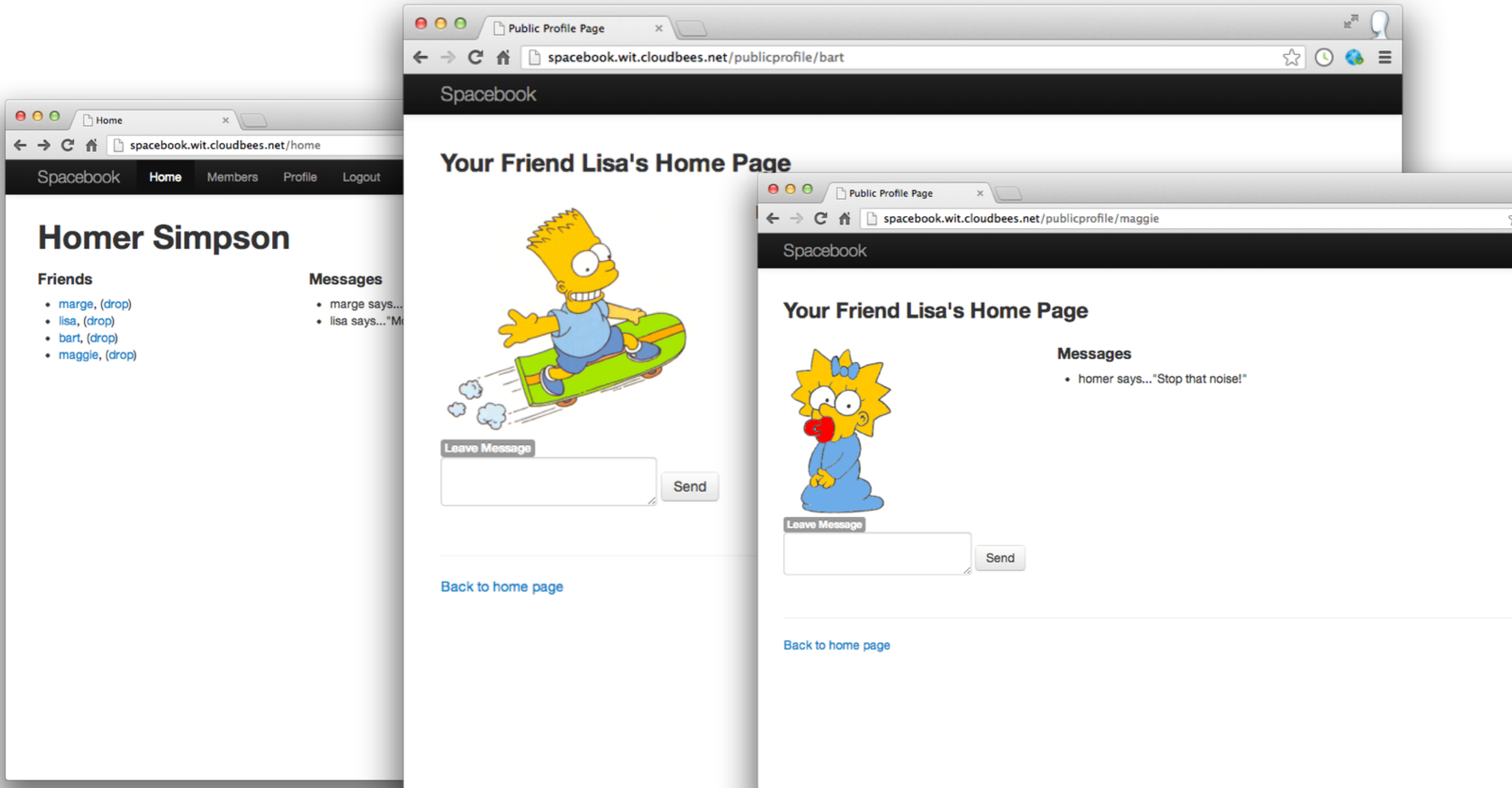


# Lab07: Exercises

---

# Exercise 1

- Currently marge and lisa appear on the home page as 'friends'. Add bart and maggie. Make sure that, when visiting the pages, the correct images appear.



# Exercise 1 Solution

---

- No change necessary to application, apart from introducing new links in home page:

```
<div class="ui list">
  <div class="item">
    <i class="right triangle icon"></i> <a href="/publicprofile/marge">marge</a>, (<a href="drop/marge">drop</a>)
  </div>
  <div class="item">
    <i class="right triangle icon"></i> <a href="/publicprofile/lisa">lisa</a>, (<a href="drop/lisa">drop</a>)
  </div>
  <div class="item">
    <i class="right triangle icon"></i> <a href="/publicprofile/bart">bart</a>, (<a href="drop/bart">drop</a>)
  </div>
  <div class="item">
    <i class="right triangle icon"></i> <a href="/publicprofile/maggie">maggie</a>, (<a href="drop/maggie">drop</a>)
  </div>
</div>
```

## HomeProfile/index.html

```
<div class="ui list">
  <div class="item">
    <i class="right triangle icon"></i> <a href="/publicprofile/marge">marge</a>, (<a href="drop/marge">drop</a>)
  </div>
  <div class="item">
    <i class="right triangle icon"></i> <a href="/publicprofile/lisa">lisa</a>, (<a href="drop/lisa">drop</a>)
  </div>
  <div class="item">
    <i class="right triangle icon"></i> <a href="/publicprofile/bart">bart</a>, (<a href="drop/bart">drop</a>)
  </div>
  <div class="item">
    <i class="right triangle icon"></i> <a href="/publicprofile/maggie">maggie</a>, (<a href="drop/maggie">drop</a>)
  </div>
</div>
```

## controllers/PublicProfile.java

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

  public static void visit(String name)
  {
    Logger.info("Just visiting the page for " + name);
    render(name);
  }
}
```

## PublicProfile/visit.html

```
#{extends 'main.html' /}
#{set title:'Spacebook' /}

<nav class="ui menu">
  <a class="ui active item" href="/home">Back to Home</a>
</nav>

<section class="ui raised segment">
  <div class="ui small header"> ${name}'s Profile</div>
  <section class="ui two column grid segment">
    <div class="row">
      <div class="column">
        <div class="ui medium image">
          
        </div>
      </div>
      <div class="column">
        <h2> Messages </h2>
        <ul>
          <li>homer says..."What time is dinner?"</li>
          <li>lisa says..."Where is my saxophone?"</li>
          <li>homer says..."Where are you?"</li>
        </ul>
      </div>
    </div>
  </section>
</section>
```

# Exercise 2

- Visit the members page - which shows a list of members + a 'follow' link. This is the source of members/index.html here:
- Note that each of the links is blank, and should be unresponsive (try them). Using step 4 of this lab as a guide, implement links in the above such that whenever the 'follow' link is selected, we log to the console the name of the member we want to follow.
- Eg - the CLI (DOS or Command prompt) might show:
- This will require that you create a new method in Java, and implement a call to the logger (as in step 4). NB: This exercise will also require a changes to the 'conf/routes' file. Specifically, you will need to introduce a new route to support the follow method, very similar to the 'visit' route we have implemented earlier

```
<div class="ui list">
  <div class="item">
    <i class="right triangle icon"></i> marge [<a href="home.html">fo
  </div>
  <div class="item">
    <i class="right triangle icon"></i> bart [<a href="home.html">fol
  </div>
  <div class="item">
    <i class="right triangle icon"></i> lisa [<a href="home.html">fol
  </div>
  <div class="item">
    <i class="right triangle icon"></i> maggie [<a href="home.html">f
  </div>
</div>
```

```
08:34:23,768 INFO ~ Just visiting the page for maggie
08:37:24,417 INFO ~ Just visiting the page for marge
08:37:27,320 INFO ~ Just visiting the page for lisa
08:37:30,430 INFO ~ Just visiting the page for bart
08:37:33,511 INFO ~ Just visiting the page for maggie
08:56:10,652 INFO ~ Following marge
08:56:14,085 INFO ~ Following bart
08:56:16,802 INFO ~ Following maggie
```

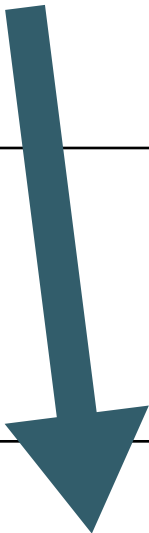
# Exercise 2

## Solution (1)

---

- Introduce 'follow' link...

```
<div class="ui list">
  <div class="item">
    <i class="right triangle icon"></i> marge [
```



```
<div class="ui list">
  <div class="item">
    <i class="right triangle icon"></i> marge [
```

# Exercise 2 Solution (2)

---

- Introduce 'follow' action...

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

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



# Exercise 2 Solution (3)

---

- Introduce 'follow' route...

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

# Exercise 2 Summary

• §

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

marge [[follow](/members/follow/marge)]

GET /members/follow/{name}

Members.follow

```
public class Members extends Controller
{
    public static void index()
    {
        render();
    }
    public static void follow(String name)
    {
        Logger.info("Following " + name);
    }
}
```

```
08:34:23,766 INFO ~ Just visiting the page for maggie
08:37:24,417 INFO ~ Just visiting the page for marge
08:37:27,320 INFO ~ Just visiting the page for lisa
08:37:30,430 INFO ~ Just visiting the page for bart
08:37:33,511 INFO ~ Just visiting the page for maggie
08:56:10,652 INFO ~ Following marge
08:56:14,085 INFO ~ Following bart
08:56:16,802 INFO ~ Following maggie
```

# Exercise 3

---

- If you get Exercise 2 working, then turn your attention back to the list of friends on the home page. What should the 'drop' link do? Implement something similar to the last exercise: i.e. have pressing 'drop' log a message to the console.
- Do this as follows:
  - Create a method in Home class called "drop"
  - it should take a single string as a parameter
  - it should log the string and a message to the console
  - a route should be set up in conf/routes to 'map' this route
  - edit the home/index.html to generate the correct link.
- If all goes according to plan, clicking on the links should cause log message to appear. However, the user interface will take you to a blank page each time. Can you change this? Try calling "index();" in the drop method.

# Exercise 3 Solution (1)

---

- Create a method in Home class called "drop"
  - it should take a single string as a parameter
  - it should log the string and a message to the console

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

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

# Exercise 3 Solution (1)

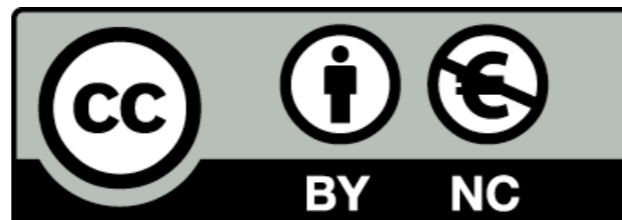
---

- a route should be set up in conf/routes to 'map' this route

GET	/home/drop/{name}	Home.drop
-----	-------------------	-----------

- edit the home/index.html to generate the correct link.

```
<a href="/publicprofile/marge">marge</a>, (<a href="/home/drop/marge">drop</a>)  
...
```



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/>

