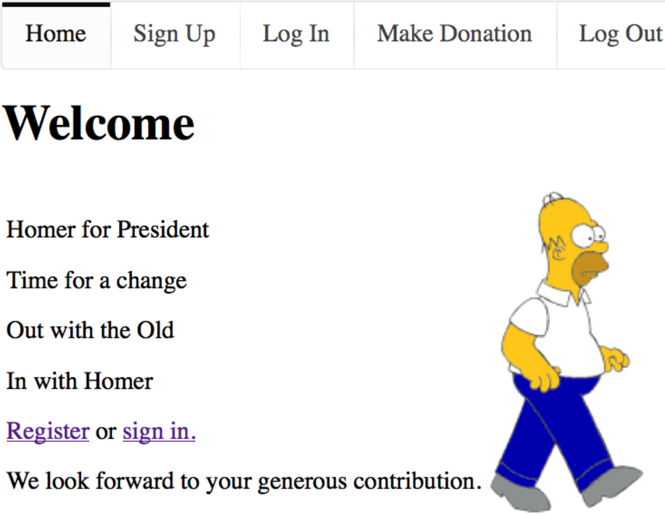
Donation
Dynamic Web App

Waterford Institute of Technology

March 19, 2016

John Fitzgerald

Welcome page



Welcome

Homer for President

Time for a change

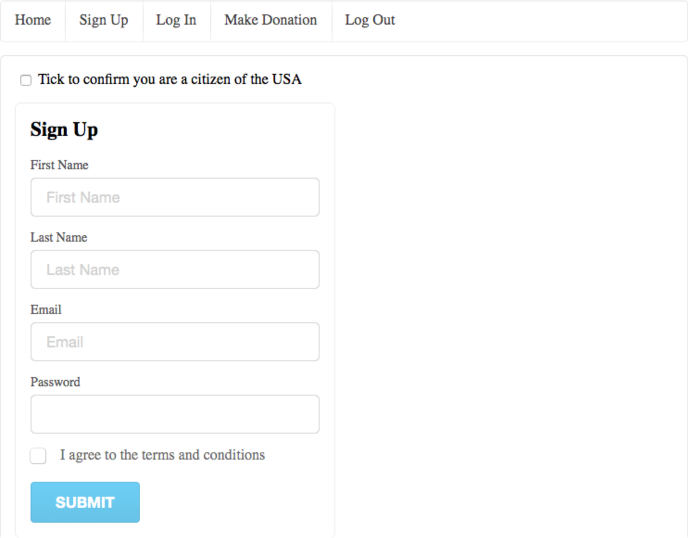
Out with the Old

In with Homer

Register or sign in.

We look forward to your generous contribution.

Sign up page



□ Tick to confirm you are a citizen of the USA

Sign Up

First Name

First Name

Last Name

Last Name

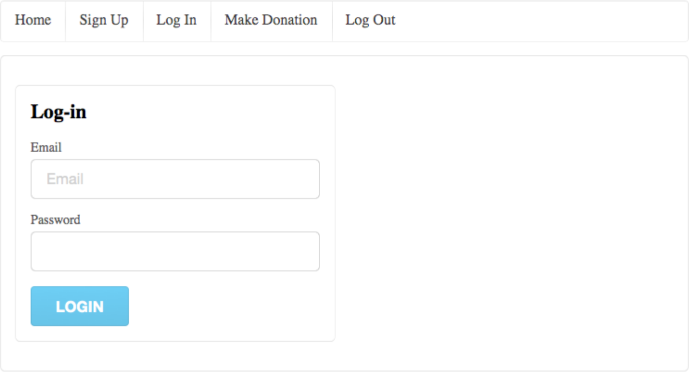
Email

Email

Password

□ I agree to the terms and conditions

Sign in page



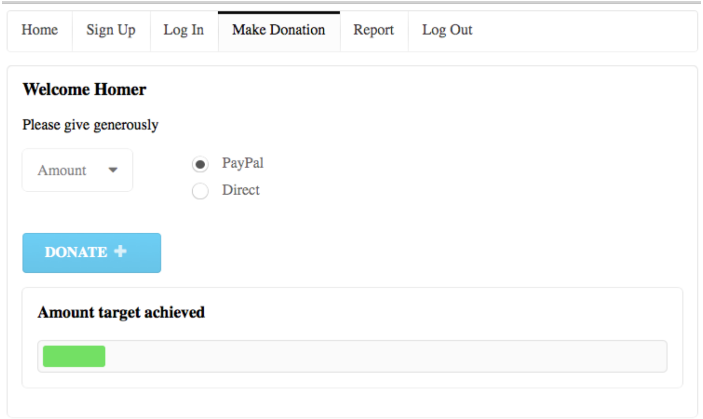
Log-in

Email

Email

Password

Making a Donation



Welcome Homer

Please give generously

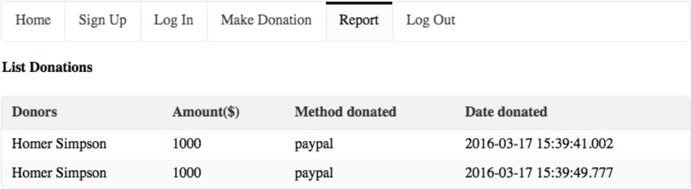
Amount ▾

● PayPal

○ Direct

Amount target achieved

List of Donations

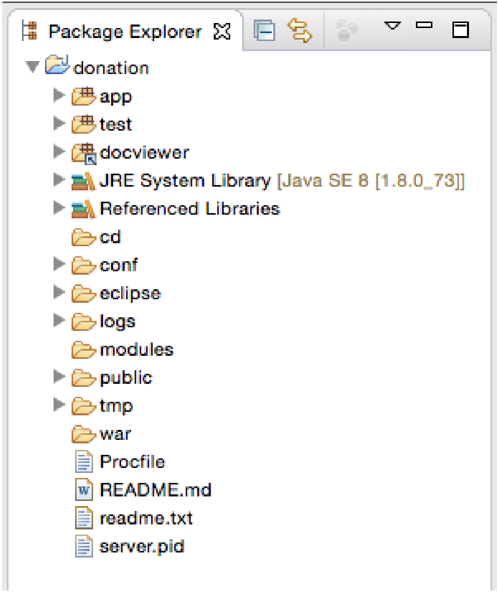


List Donations

Donors	Amount($)	Method donated	Date donated
Homer Simpson	1000	paypal	2016-03-17 15:39:41.002
Homer Simpson	1000	paypal	2016-03-17 15:39:49.777

Application Structure

- app
- conf
- public

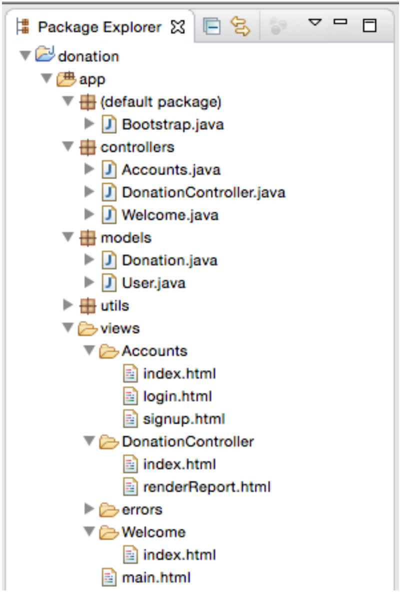


Package Explorer

- donation
  - app
  - test
  - docviewer
  - JRE System Library [Java SE 8 [1.8.0_73]]
  - Referenced Libraries
  - cd
  - conf
  - eclipse
  - logs
  - modules
  - public
  - tmp
  - war
  - Procfile
  - README.md
  - readme.txt
  - server.pid

app

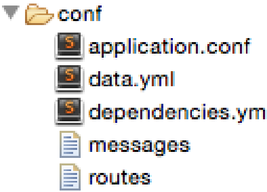
- Java
  - default
  - controllers
  - model
- Html/template
  - views



```
Package Explorer
▼ donation
  ▼ app
    ▼ (default package)
      ▶ Bootstrap.java
    ▼ controllers
      ▶ Accounts.java
      ▶ DonationController.java
      ▶ Welcome.java
    ▼ models
      ▶ Donation.java
      ▶ User.java
    ▶ utils
    ▼ views
      ▼ Accounts
          index.html
          login.html
          signup.html
      ▼ DonationController
          index.html
          renderReport.html
      ▶ errors
      ▼ Welcome
          index.html
        main.html
```

conf

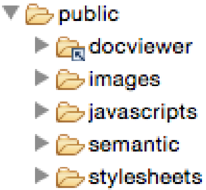
- application
- data
- routes
- dependencies



Public

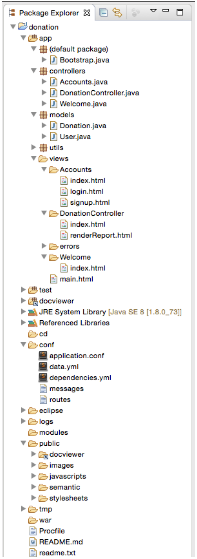
Static resources

- CSS
- Images
- JavaScript libraries



Play Project

- Remaining features generated as part build process and generally not of interest to programmer.
- Focus on maintaining:
  - app
  - conf
  - public
  - test

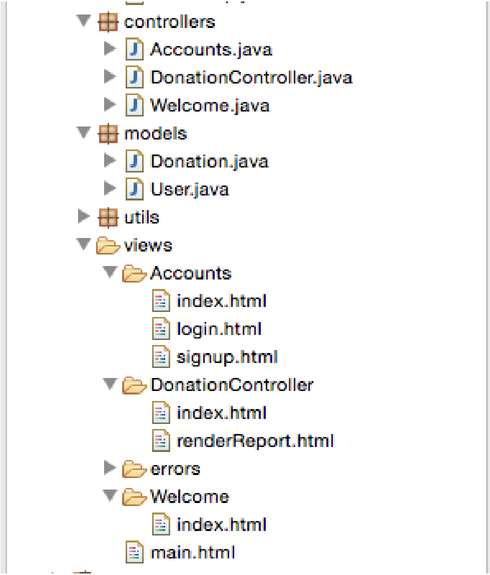


Package Explorer

```
donation
  app
    (default package)
      Bootstrap.java
    controllers
      Accounts.java
      DonationController.java
      Welcome.java
    models
      Donation.java
      User.java
    utils
    views
      Accounts
        index.html
        login.html
        signup.html
      DonationController
        index.html
        renderReport.html
      errors
      Welcome
        index.html
      main.html
  test
  docviewer
  JRE System Library [Java SE 8 [1.8.0_73]]
  Referenced Libraries
  cd
  conf
    application.conf
    data.yml
    dependencies.yml
    messages
    routes
  eclipse
  logs
  modules
  public
    docviewer
    images
    javascripts
    semantic
    stylesheets
  tmp
  war
  Procfile
  README.md
  readme.txt
```

Donation

- 2 models
- 3 controllers
- 6 views

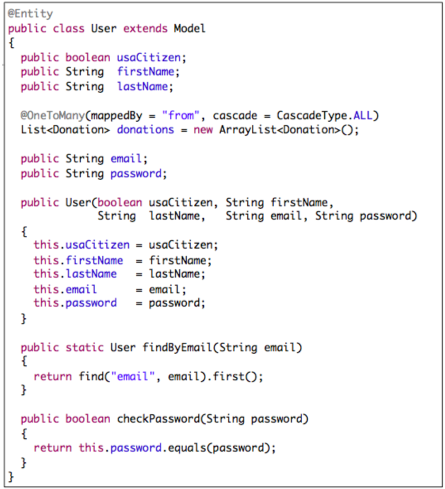


```
▼ controllers
    ▶ Accounts.java
    ▶ DonationController.java
    ▶ Welcome.java
▼ models
    ▶ Donation.java
    ▶ User.java
▶ utils
▼ views
    ▼ Accounts
          index.html
          login.html
          signup.html
    ▼ DonationController
          index.html
          renderReport.html
    ▶ errors
    ▼ Welcome
          index.html
      main.html
```

Models

User

- Represents user
- Has list donations
- Candidate implicit, i.e. *Homer*

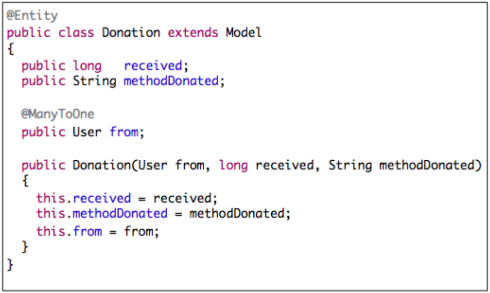


```
@Entity
public class User extends Model
{
  public boolean usaCitizen;
  public String  firstName;
  public String  lastName;

  @OneToMany(mappedBy = "from", cascade = CascadeType.ALL)
  List<Donation> donations = new ArrayList<Donation>();

  public String email;
  public String password;

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

Models

Donation

- Individual donation
- Donation method
- Donor

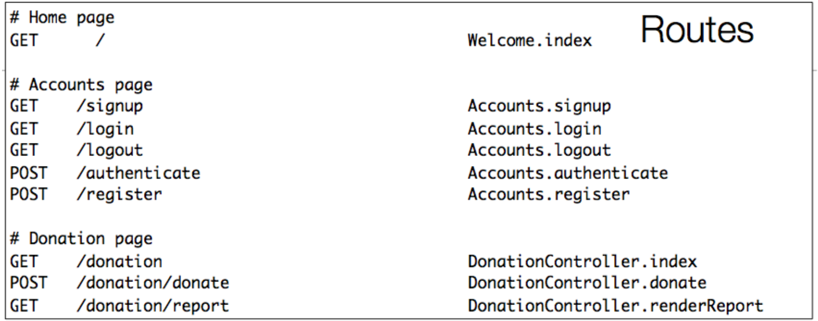


```
@Entity
public class Donation extends Model
{
  public long   received;
  public String methodDonated;

  @ManyToOne
  public User from;

  public Donation(User from, long received, String methodDonated)
  {
    this.received = received;
    this.methodDonated = methodDonated;
    this.from = from;
  }
}
```

Routes

Associate views with Java controller methods (actions)

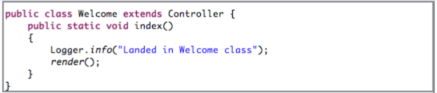
- Key to understanding how the views relate to the controllers
- Route patterns are used in views:
  - By html <a> tags (links)
  - By form submit buttons
  - To associate these with actions



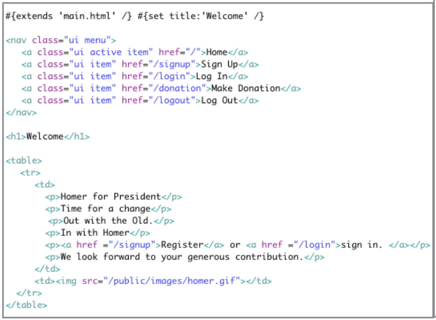
Routes

```
# Home page
GET     /                                            Welcome.index

# Accounts page
GET     /signup                                      Accounts.signup
GET     /login                                       Accounts.login
GET     /logout                                      Accounts.logout
POST    /authenticate                                Accounts.authenticate
POST    /register                                    Accounts.register

# Donation page
GET     /donation                                    DonationController.index
POST    /donation/donate                             DonationController.donate
GET     /donation/report                             DonationController.renderReport
```

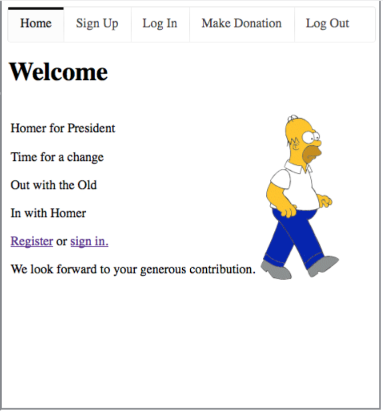
Welcome *Controller - View*



```
public class Welcome extends Controller {
    public static void index()
    {
        Logger.info("Landed in Welcome class");
        render();
    }
}
```



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

<nav class="ui menu">
   <a class="ui active item" href="/">Home</a>
   <a class="ui item" href="/signup">Sign Up</a>
   <a class="ui item" href="/login">Log In</a>
   <a class="ui item" href="/donation">Make Donation</a>
   <a class="ui item" href="/logout">Log Out</a>
</nav>

<h1>Welcome</h1>

<table>
   <tr>
      <td>
         <p>Homer for President</p>
         <p>Time for a change</p>
          <p>Out with the Old.</p>
         <p>In with Homer</p>
         <p><a href ="/signup">Register</a> or <a href ="/login">sign in. </a></p>
         <p>We look forward to your generous contribution.</p>
      </td>
      <td><img src="/public/images/homer.gif"></td>
   </tr>
</table>
```



Welcome

Homer for President

Time for a change

Out with the Old

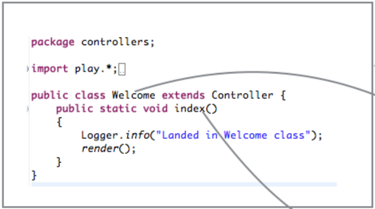
In with Homer

Register or sign in.

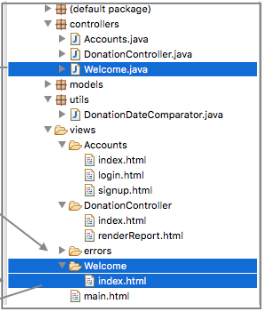
We look forward to your generous contribution.

Welcome Controller

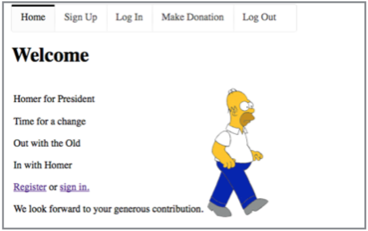
Welcome View



```
package controllers;

import play.*;

public class Welcome extends Controller {
    public static void index()
    {
        Logger.info("Landed in Welcome class");
        render();
    }
}
```



```
▶ (default package)
▼ controllers
    ▶ Accounts.java
    ▶ DonationController.java
    ▶ Welcome.java
▶ models
▼ utils
    ▶ DonationDateComparator.java
▼ views
    ▼ Accounts
        index.html
        login.html
        signup.html
    ▼ DonationController
        index.html
        renderReport.html
    ▶ errors
    ▼ Welcome
        index.html
    main.html
```



Welcome

Homer for President

Time for a change

Out with the Old

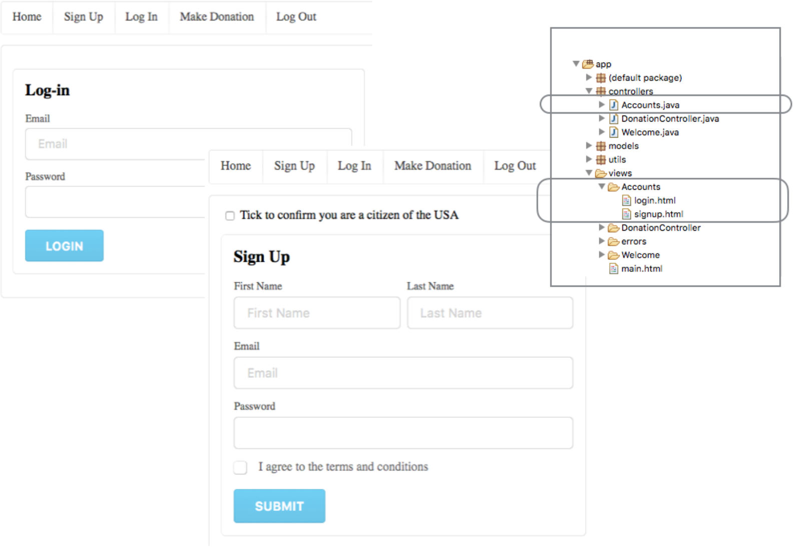
In with Homer

Register or sign in.

We look forward to your generous contribution.

Accounts Controller

Accounts Views



Log-in

Email

Email

Password

□ Tick to confirm you are a citizen of the USA

Sign Up

First Name | Last Name

First Name | Last Name

Email

Email

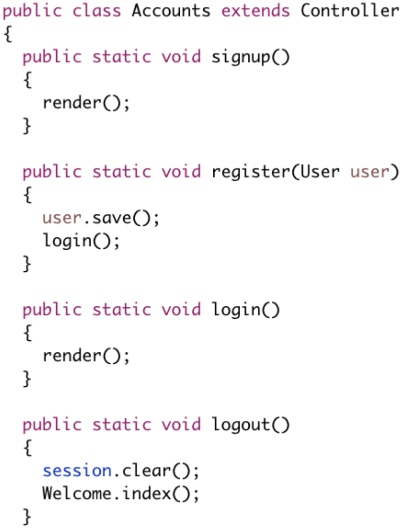
Password

□ I agree to the terms and conditions

```
▼ app
  ▶ (default package)
  ▼ controllers
    ▶ Accounts.java
    ▶ DonationController.java
    ▶ Welcome.java
  ▶ models
  ▶ utils
  ▼ views
    ▼ Accounts
        login.html
        signup.html
    ▶ DonationController
    ▶ errors
    ▶ Welcome
      main.html
```

Controller

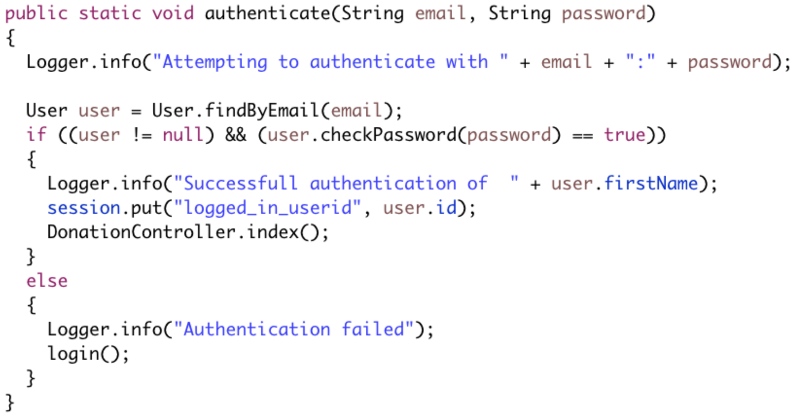
Accounts



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

  public static void register(User user)
  {
    user.save();
    login();
  }

  public static void login()
  {
    render();
  }

  public static void logout()
  {
    session.clear();
    Welcome.index();
  }
```

Controller

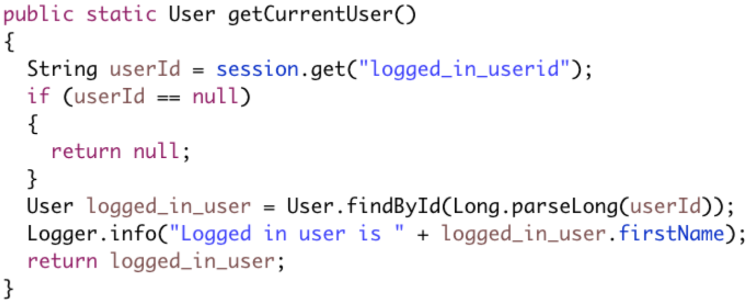
Accounts



```
public static void authenticate(String email, String password)
{
  Logger.info("Attempting to authenticate with " + email + ":" + password);

  User user = User.findByEmail(email);
  if ((user != null) && (user.checkPassword(password) == true))
  {
    Logger.info("Successfull authentication of  " + user.firstName);
    session.put("logged_in_userid", user.id);
    DonationController.index();
  }
  else
  {
    Logger.info("Authentication failed");
    login();
  }
}
```

Controller

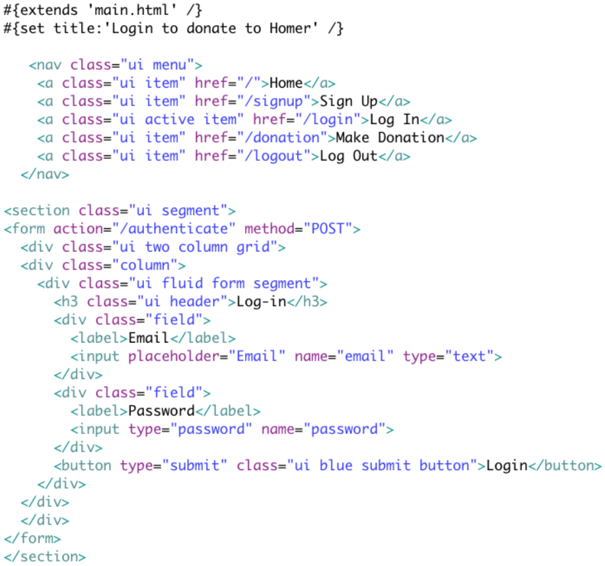
Accounts



```
public static User getCurrentUser()
{
  String userId = session.get("logged_in_userid");
  if (userId == null)
  {
    return null;
  }
  User logged_in_user = User.findById(Long.parseLong(userId));
  Logger.info("Logged in user is " + logged_in_user.firstName);
  return logged_in_user;
}
```

Views

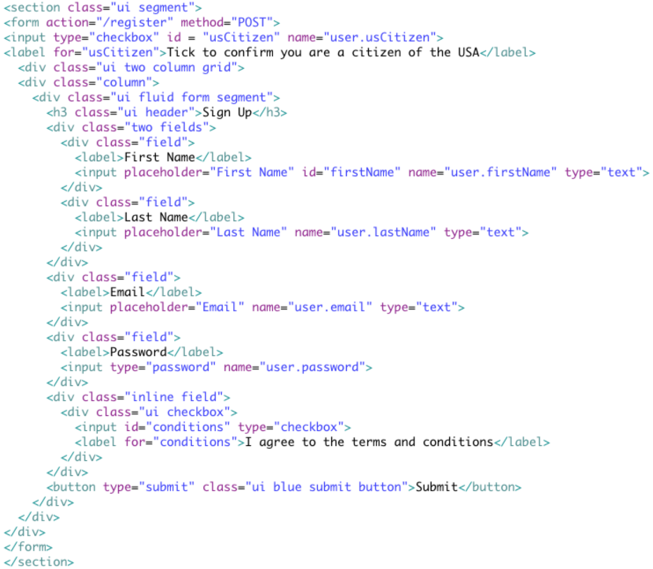
Accounts



```
#{extends 'main.html' /}
#{set title:'Login to donate to Homer' /}

   <nav class="ui menu">
    <a class="ui item" href="/">Home</a>
    <a class="ui item" href="/signup">Sign Up</a>
    <a class="ui active item" href="/login">Log In</a>
    <a class="ui item" href="/donation">Make Donation</a>
    <a class="ui item" href="/logout">Log Out</a>
  </nav>

<section class="ui segment">
<form action="/authenticate" method="POST">
  <div class="ui two column grid">
  <div class="column">
    <div class="ui fluid form segment">
      <h3 class="ui header">Log-in</h3>
      <div class="field">
        <label>Email</label>
        <input placeholder="Email" name="email" type="text">
      </div>
      <div class="field">
        <label>Password</label>
        <input type="password" name="password">
      </div>
      <button type="submit" class="ui blue submit button">Login</button>
    </div>
  </div>
  </div>
</form>
</section>
```

Views

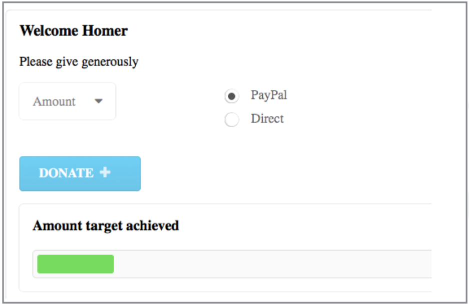
Accounts



```
<section class="ui segment">
<form action="/register" method="POST">
<input type="checkbox" id = "usCitizen" name="user.usCitizen">
<label for="usCitizen">Tick to confirm you are a citizen of the USA</label>
  <div class="ui two column grid">
  <div class="column">
    <div class="ui fluid form segment">
      <h3 class="ui header">Sign Up</h3>
      <div class="two fields">
        <div class="field">
          <label>First Name</label>
          <input placeholder="First Name" id="firstName" name="user.firstName" type="text">
        </div>
        <div class="field">
          <label>Last Name</label>
          <input placeholder="Last Name" name="user.lastName" type="text">
        </div>
      </div>
      <div class="field">
        <label>Email</label>
        <input placeholder="Email" name="user.email" type="text">
      </div>
      <div class="field">
        <label>Password</label>
        <input type="password" name="user.password">
      </div>
      <div class="inline field">
        <div class="ui checkbox">
          <input id="conditions" type="checkbox">
          <label for="conditions">I agree to the terms and conditions</label>
        </div>
      </div>
      <button type="submit" class="ui blue submit button">Submit</button>
    </div>
  </div>
</div>
</form>
</section>
```

Donation Controller

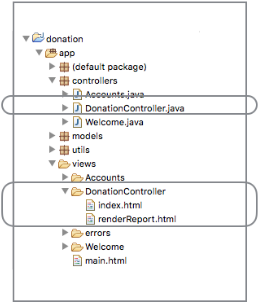
Donation Views

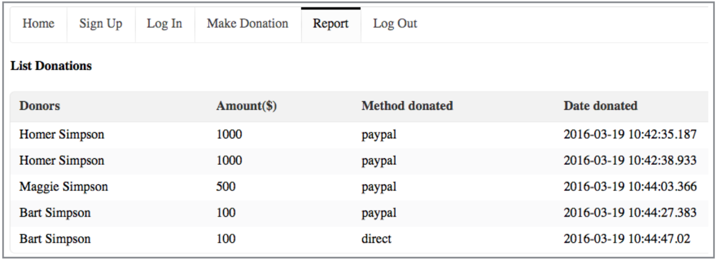


Welcome Homer

Please give generously

Amount target achieved



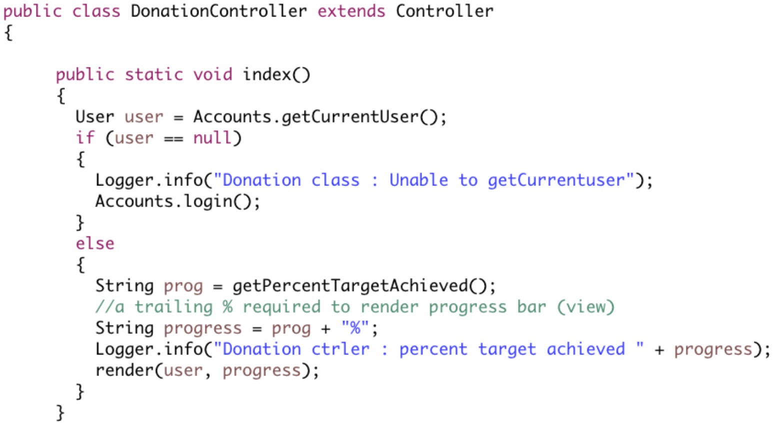


List Donations

Donors	Amount($)	Method donated	Date donated
Homer Simpson	1000	paypal	2016-03-19 10:42:35.187
Homer Simpson	1000	paypal	2016-03-19 10:42:38.933
Maggie Simpson	500	paypal	2016-03-19 10:44:03.366
Bart Simpson	100	paypal	2016-03-19 10:44:27.383
Bart Simpson	100	direct	2016-03-19 10:44:47.02

Donation Controller

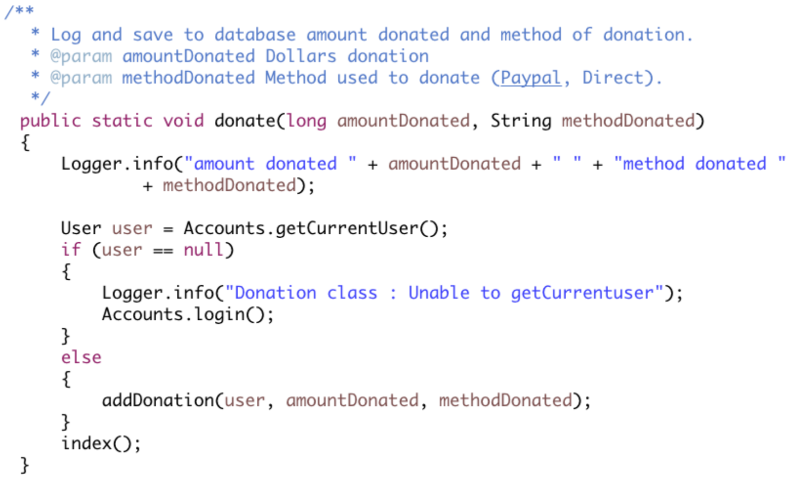
Source code (1)



```
public class DonationController extends Controller
{

    public static void index()
    {
      User user = Accounts.getCurrentUser();
      if (user == null)
      {
        Logger.info("Donation class : Unable to getCurrentuser");
        Accounts.login();
      }
      else
      {
        String prog = getPercentTargetAchieved();
        //a trailing % required to render progress bar (view)
        String progress = prog + "%";
        Logger.info("Donation ctrler : percent target achieved " + progress);
        render(user, progress);
      }
    }
```

Donation Controller

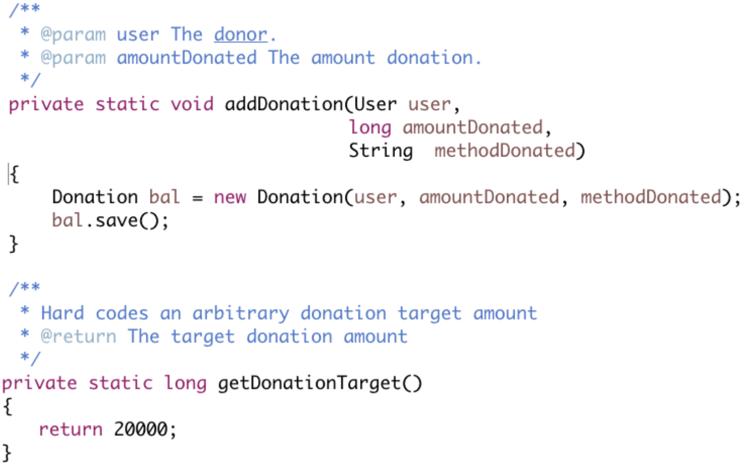
Source code (2)



```
/**
  * Log and save to database amount donated and method of donation.
  * @param amountDonated Dollars donation
  * @param methodDonated Method used to donate (Paypal, Direct).
  */
 public static void donate(long amountDonated, String methodDonated)
 {
     Logger.info("amount donated " + amountDonated + " " + "method donated "
             + methodDonated);

     User user = Accounts.getCurrentUser();
     if (user == null)
     {
         Logger.info("Donation class : Unable to getCurrentuser");
         Accounts.login();
     }
     else
     {
         addDonation(user, amountDonated, methodDonated);
     }
     index();
 }
```

Donation Controller

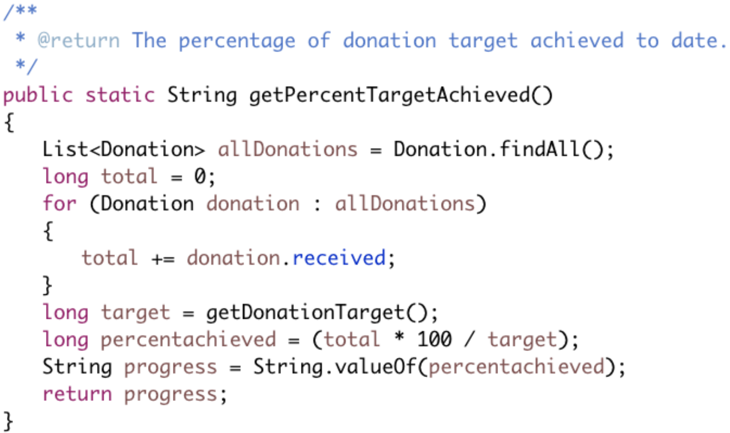
Source code (3)



```
 /**
  * @param user The donor.
  * @param amountDonated The amount donation.
  */
 private static void addDonation(User user,
                                 long amountDonated,
                                 String  methodDonated)
 {
     Donation bal = new Donation(user, amountDonated, methodDonated);
     bal.save();
 }

 /**
  * Hard codes an arbitrary donation target amount
  * @return The target donation amount
  */
private static long getDonationTarget()
{
   return 20000;
}
```

Donation Controller

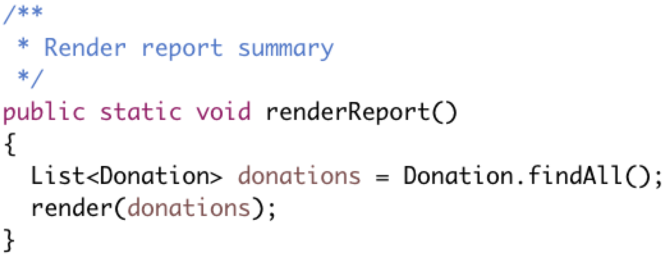
Source code (4)



```
/**
 * @return The percentage of donation target achieved to date.
 */
public static String getPercentTargetAchieved()
{
    List<Donation> allDonations = Donation.findAll();
    long total = 0;
    for (Donation donation : allDonations)
    {
        total += donation.received;
    }
    long target = getDonationTarget();
    long percentachieved = (total * 100 / target);
    String progress = String.valueOf(percentachieved);
    return progress;
}
```

Donation Controller

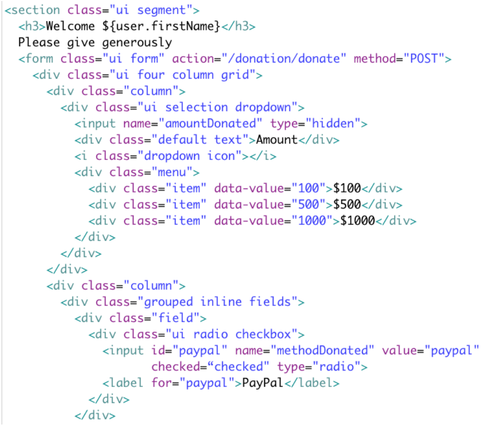
Source code (5)



```java
/**
 * Render report summary
 */
public static void renderReport()
{
  List<Donation> donations = Donation.findAll();
  render(donations);
}
```

Donation Views

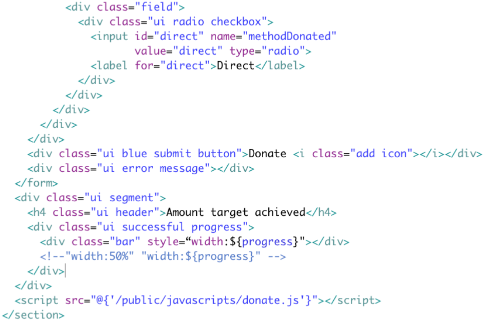
Make Donation (1)



```
<section class="ui segment">
  <h3>Welcome ${user.firstName}</h3>
  Please give generously
  <form class="ui form" action="/donation/donate" method="POST">
    <div class="ui four column grid">
      <div class="column">
        <div class="ui selection dropdown">
          <input name="amountDonated" type="hidden">
          <div class="default text">Amount</div>
          <i class="dropdown icon"></i>
          <div class="menu">
            <div class="item" data-value="100">$100</div>
            <div class="item" data-value="500">$500</div>
            <div class="item" data-value="1000">$1000</div>
          </div>
        </div>
      </div>
      <div class="column">
        <div class="grouped inline fields">
          <div class="field">
            <div class="ui radio checkbox">
              <input id="paypal" name="methodDonated" value="paypal"
                     checked=“checked" type="radio">
              <label for="paypal">PayPal</label>
            </div>
          </div>
```

Donation Views

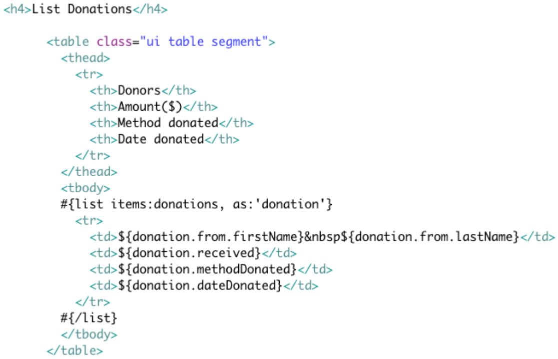
Make Donation (2)



```
          <div class="field">
            <div class="ui radio checkbox">
              <input id="direct" name="methodDonated"
                     value="direct" type="radio">
              <label for="direct">Direct</label>
            </div>
          </div>
        </div>
      </div>
    </div>
    <div class="ui blue submit button">Donate <i class="add icon"></i></div>
    <div class="ui error message"></div>
  </form>
  <div class="ui segment">
    <h4 class="ui header">Amount target achieved</h4>
    <div class="ui successful progress">
      <div class="bar" style=“width:${progress}"></div>
      <!--"width:50%" "width:${progress}" -->
    </div>
  </div>
  <script src="@{'/public/javascripts/donate.js'}"></script>
</section>
```

Donation Views

Report



```
<h4>List Donations</h4>

      <table class="ui table segment">
        <thead>
          <tr>
            <th>Donors</th>
            <th>Amount($)</th>
            <th>Method donated</th>
            <th>Date donated</th>
          </tr>
        </thead>
        <tbody>
        #{list items:donations, as:'donation'}
          <tr>
            <td>${donation.from.firstName}&nbsp${donation.from.lastName}</td>
            <td>${donation.received}</td>
            <td>${donation.methodDonated}</td>
            <td>${donation.dateDonated}</td>
          </tr>
        #{/list}
        </tbody>
      </table>
```

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

eLearning
support unit