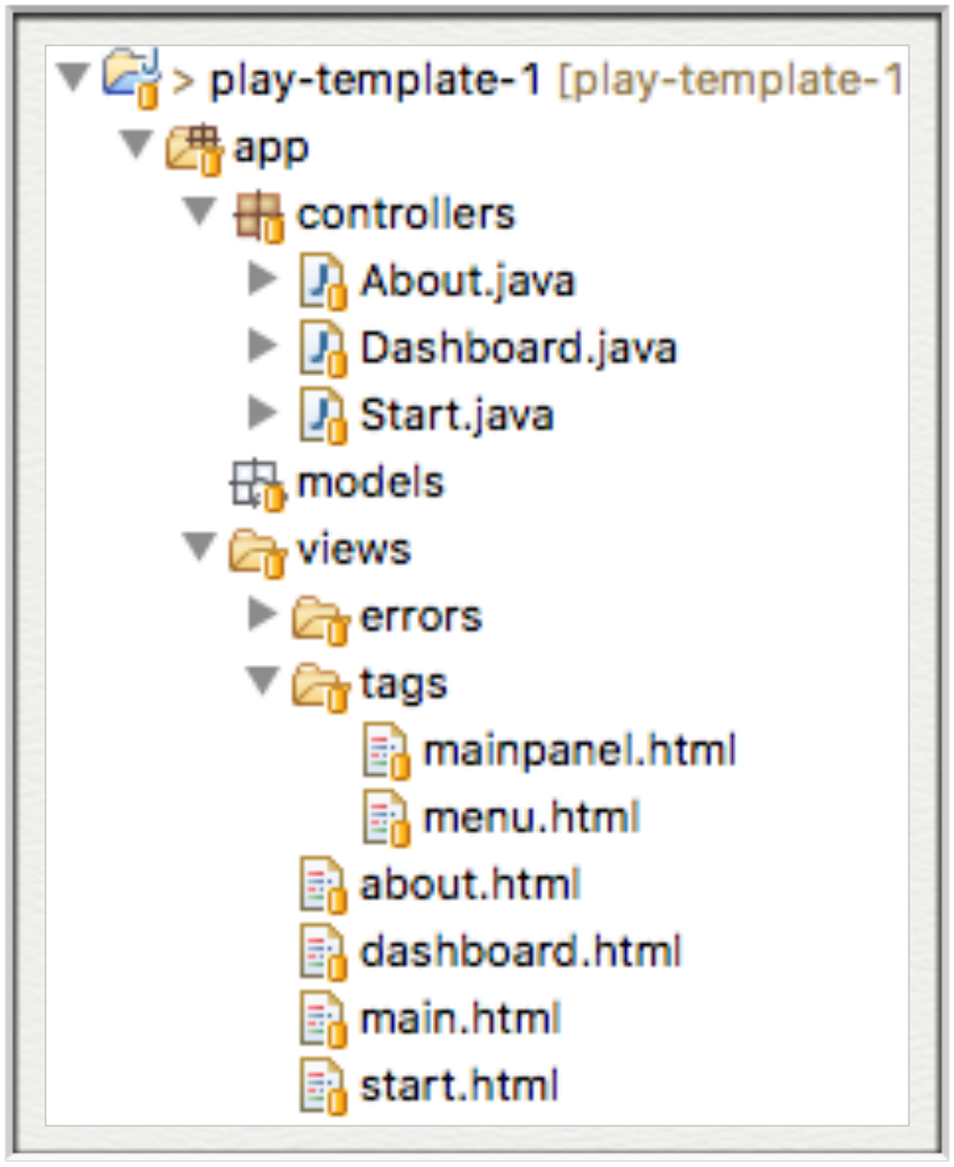
Back-end

Javascript Modules

- To structure an application coherently, the backend consists of separate Java classes declared in these files must be
    - exported by one file
    - imported by another
- In order to keep each module focused on a specific responsibility

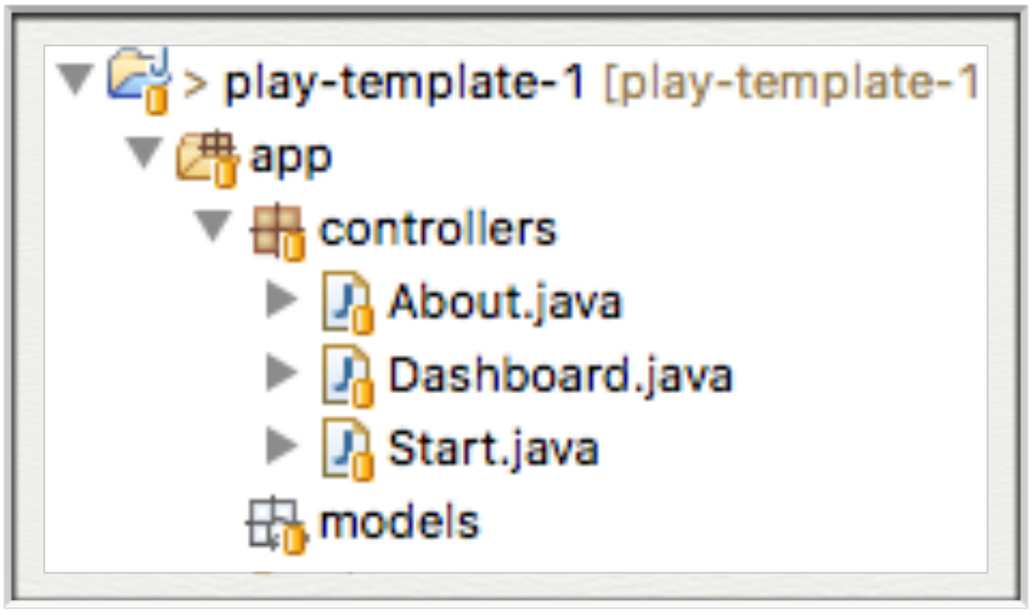
Application Structure



- App implements Routes + Model/View/Controller Architecture

- Back-end + Front-end collaborate to support structured, predictable application workflow

Back-end

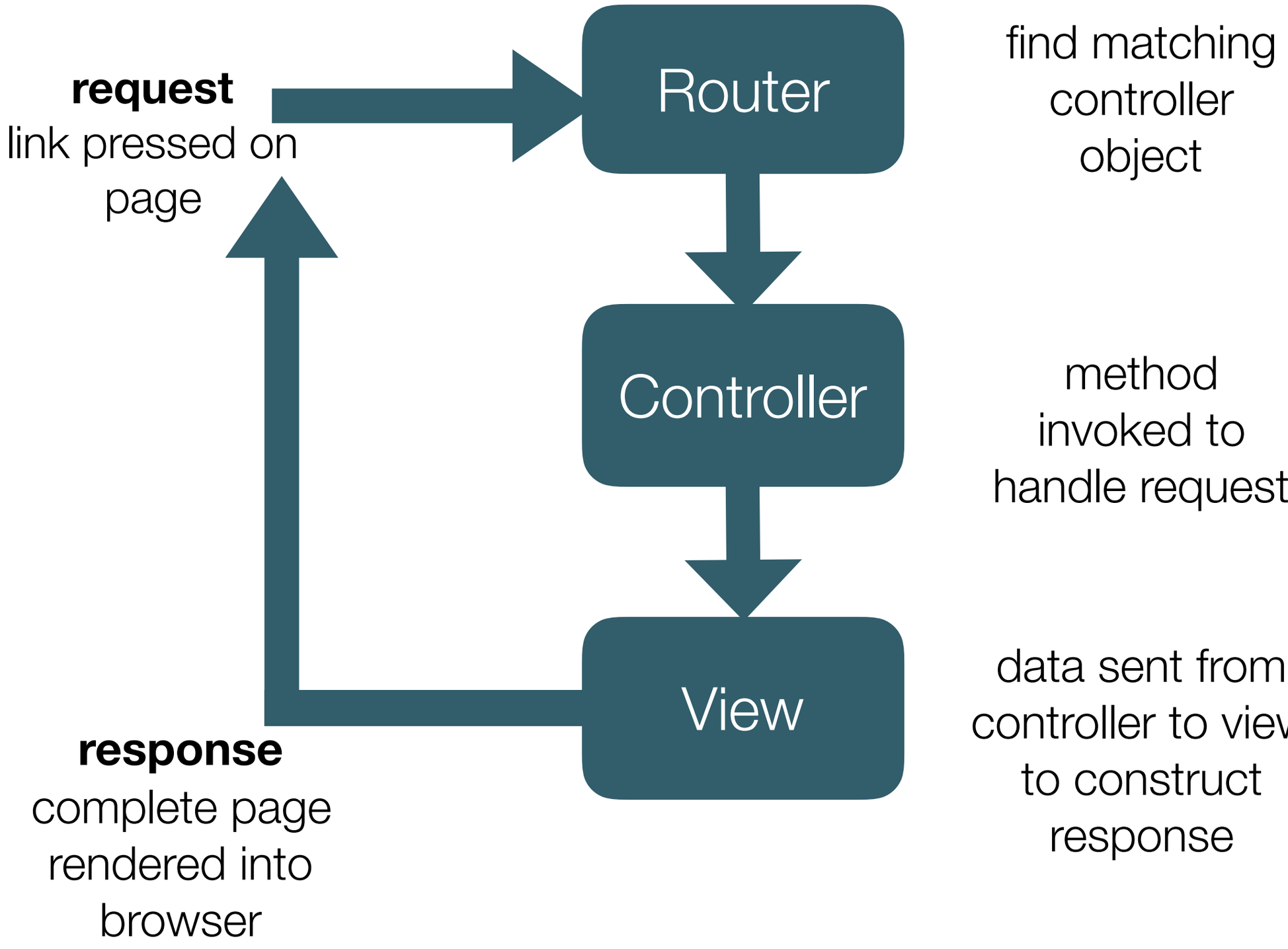


- All written in Java
- Consists of:
  - **Server** - main entry point
  - **Routes** - supported urls
  - **Controllers** - objects to handle the routes
  - **Config** - system wide config of application
- Will include **Models** later…

Request/Response Lifecycle

1. **Request** - link pressed on page
2. **Router** - find matching controller object
3. **Controller** - method invoked to handle request
4. **View** - data sent from controller to view to construct response
5. **Response** - complete page rendered into browser

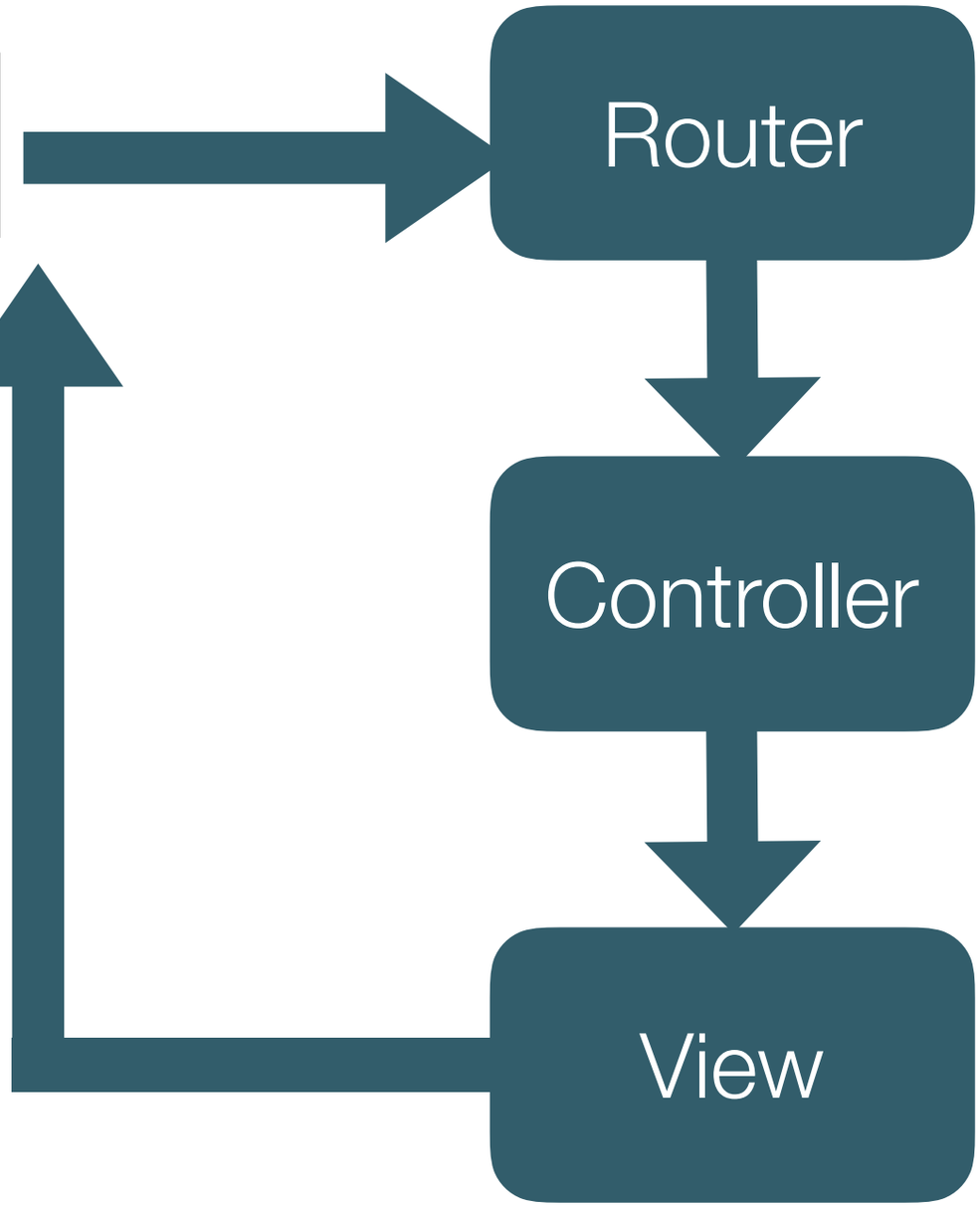
Router/Controller/View



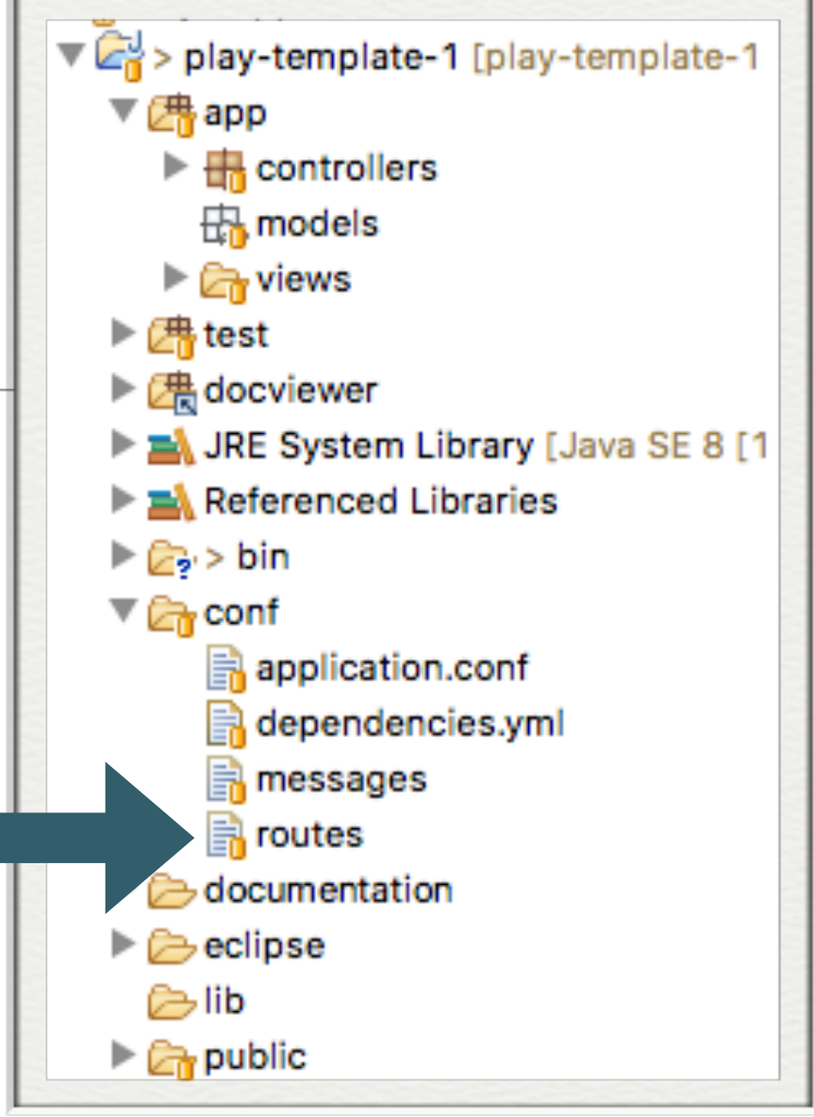
Request - link pressed on page

```
...
    <a id="dashboard" class="item" href="/dashboard"> Dashboard  </a>
    <a id="about" class="item" href="/about"> About </a>
...
```

- Requests defined in links in views:
  - href in <a> tags
  - href in Menus
  - href in Buttons
  - action links in forms



Router - find matching controller object



routes

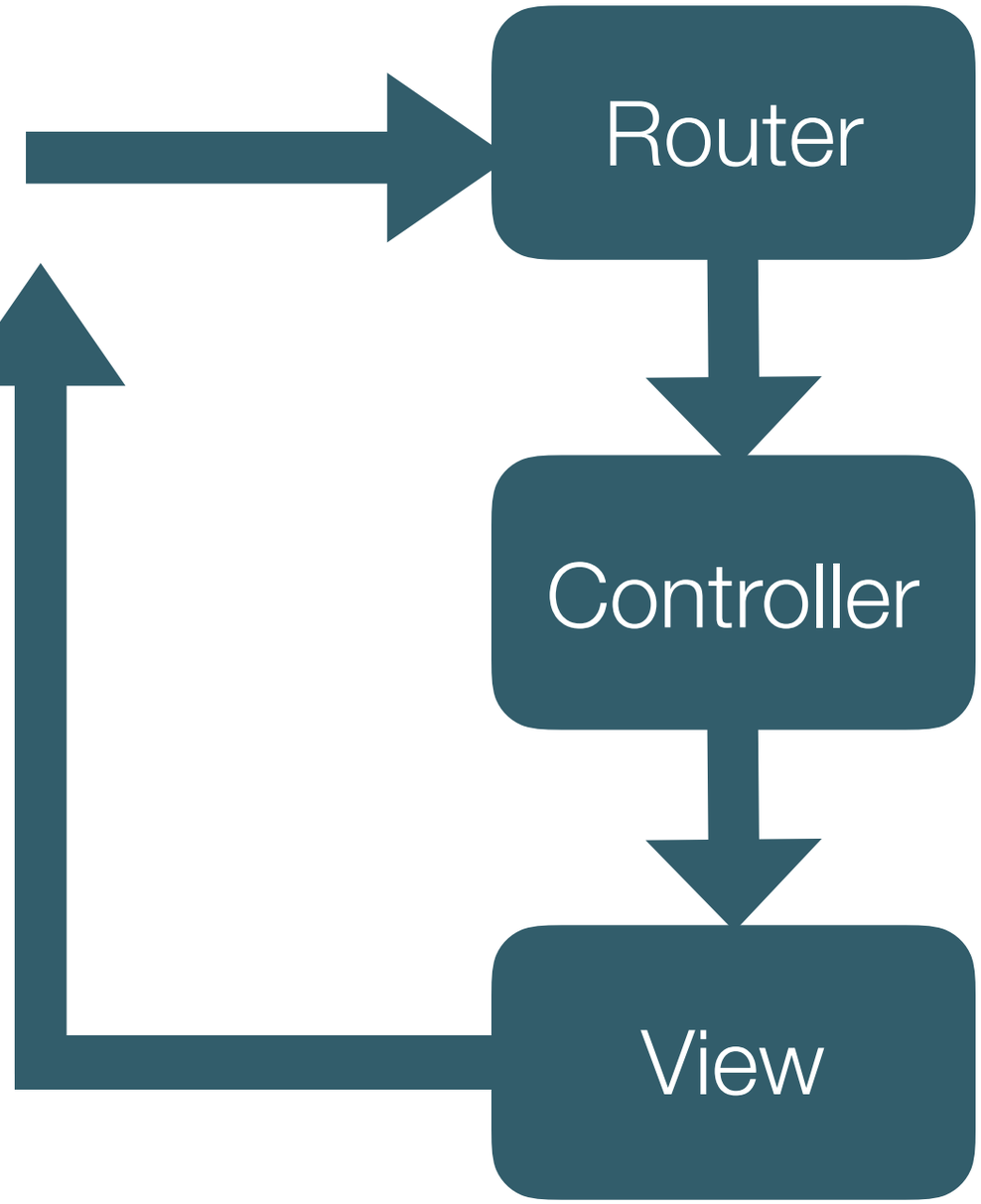
Mach the link patterns with the controler.method

```
...
GET     /                                       Start.index
GET     /dashboard                              Dashboard.index
GET     /about                                  About.index
...
```

Router Behaviour

If user selects these links…

…then call the corresponding controller methods



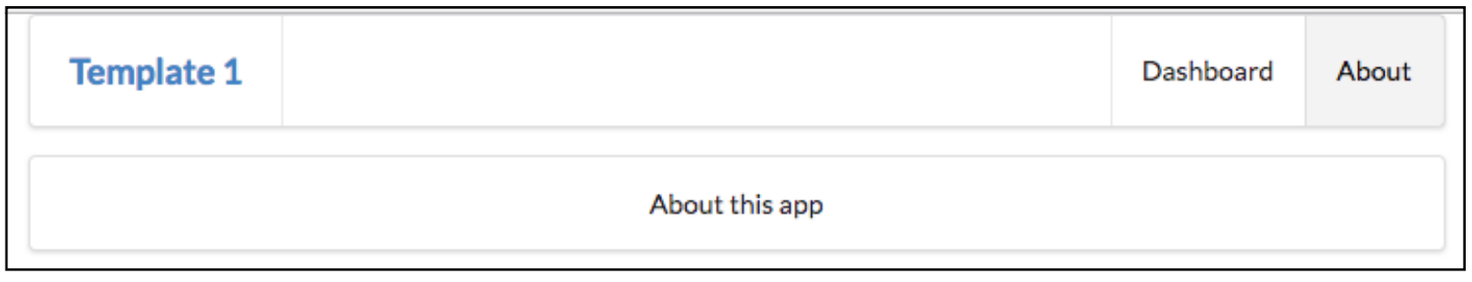

```
'/'          ⟶  Start.index

/dashboard'  ⟶  Dashboard.index

/about'      ⟶  About.index
```

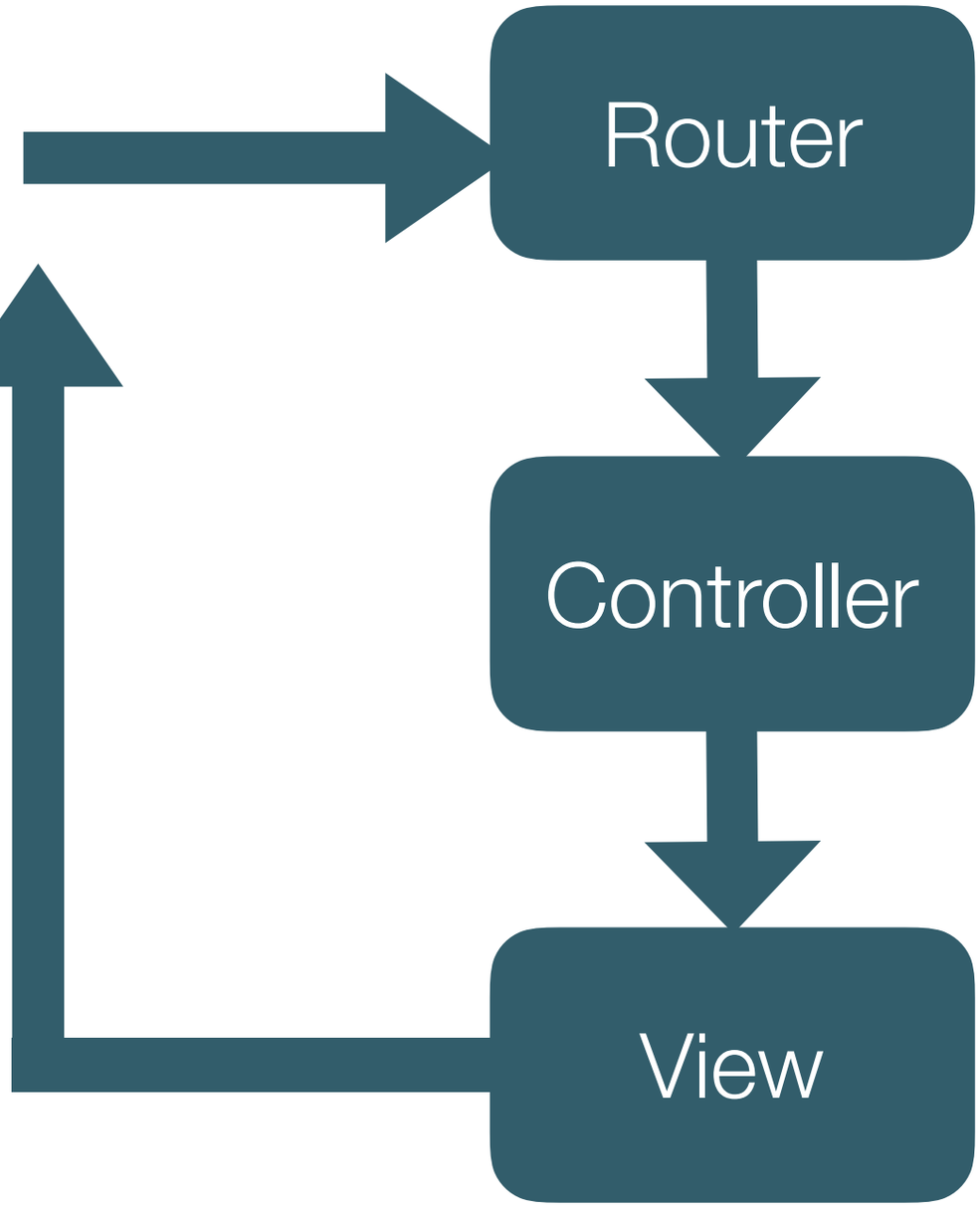
Controller method invoked to handle request

About.java

```
package controllers;

import play.*;
import play.mvc.*;
import java.util.*;
import models.*;

public class About extends Controller
{
  public static void index() {
    Logger.info("Rendering about");
    render ("about.html");
  }
}
```

The About controller





The ‘About’ controller index *action*

- Log a message to the console
- Render the ‘about’ view

```
public static void index() {
  Logger.info("Rendering about");
  render ("about.html");
}
```

about.js

Back-end + Front-End

about.html

```
public static void index() {
  Logger.info("Rendering about");
  render ("about.html");
}
```

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

#{menu id:"about"/}

<section class="ui center aligned middle aligned segment">
  <p>
    About this app
  </p>
</section>
```

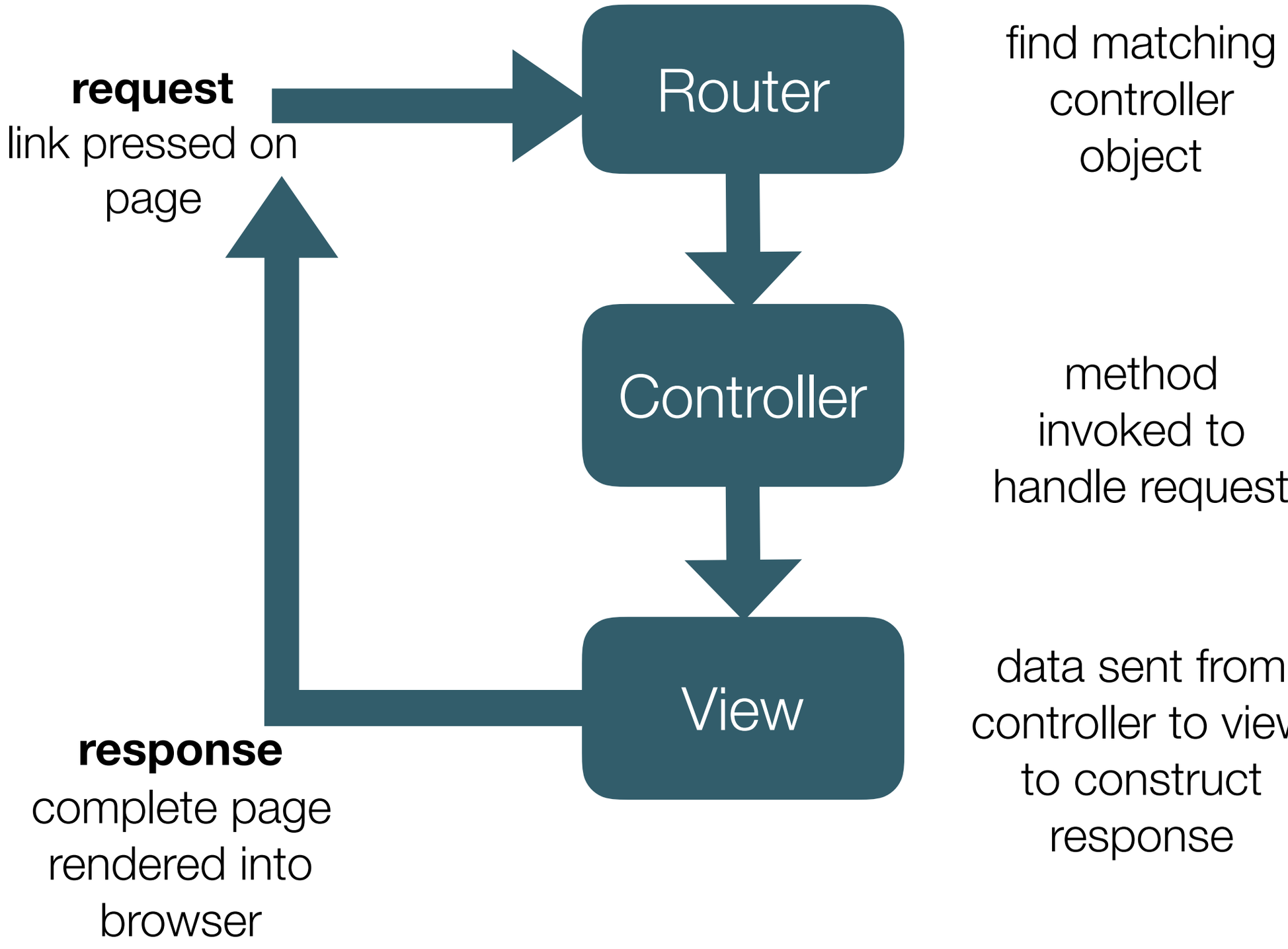
menu.html

```
<nav class="ui menu">
  <header class="ui header item"> <a href="/"> Template 1 </a></header>
  <div class="right menu">
    <a id="dashboard" class="item" href="/dashboard"> Dashboard  </a>
    <a id="about" class="item" href="/about"> About </a>
  </div>
</nav>

<script>
  $("#${_id}").addClass("active item");
</script>
```

main.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>#{get 'title' /}</title>
    <script type="text/javascript" src=
    <script type="text/javascript" src=
    <link type="text/css"  href="https:
  </head>
  <body>
    <section class="ui container">
      #{doLayout /}
    </section>
  </body>
```

Router/Controller/View



Dashboard Controller

```
package controllers;

import play.Logger;
import play.mvc.Controller;

public class Dashboard extends Controller
{
  public static void index() {
    Logger.info("Rendering Dashboard");
    render ("dashboard.html");
  }
}
```

Start Controller

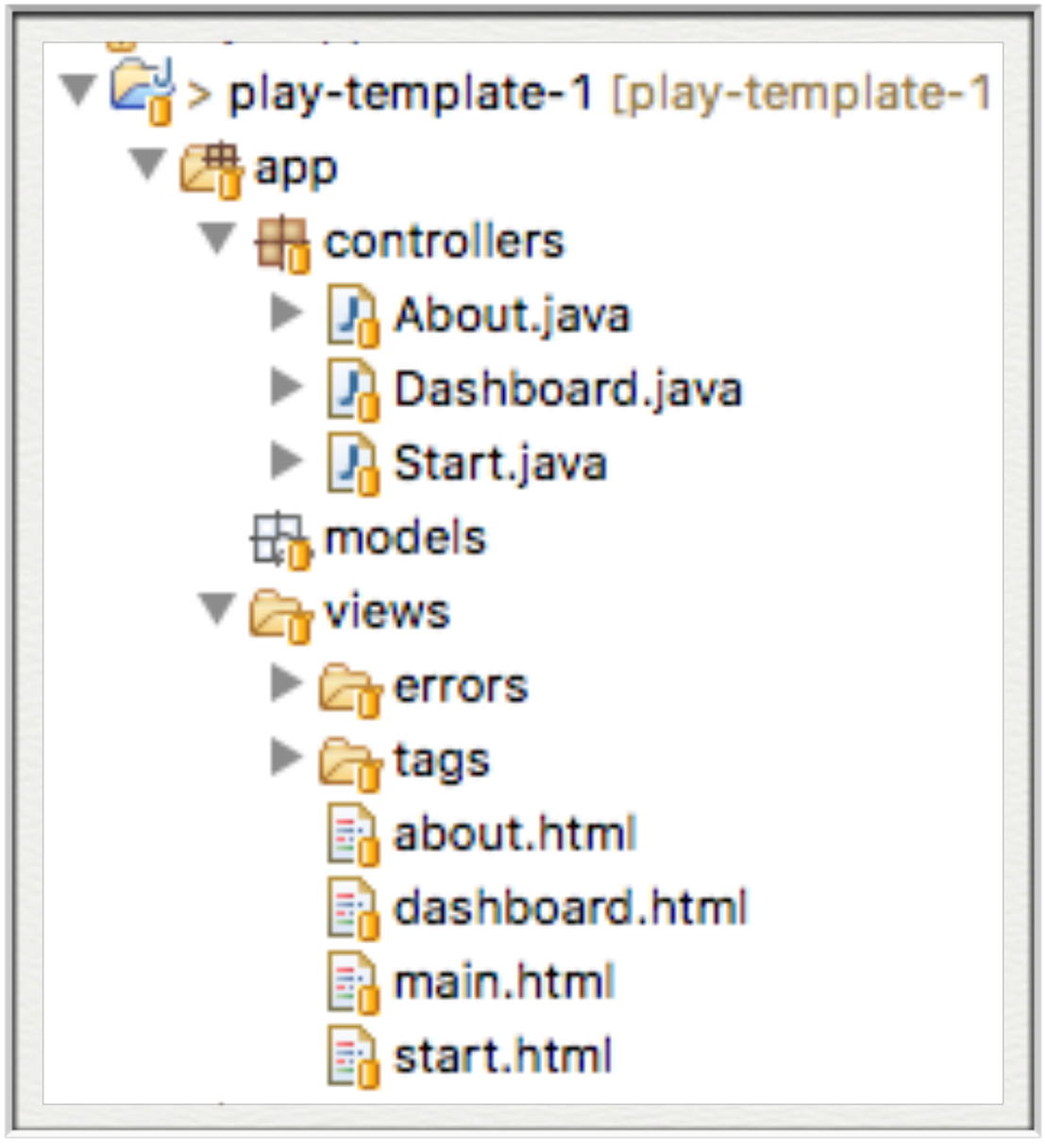
```
package controllers;

import play.Logger;
import play.mvc.Controller;

public class Start extends Controller
{
  public static void index() {
    Logger.info("Rendering Start");
    render ("start.html");
  }
}
```

Application Structure

3 Controllers

which will render

3 matching views



routes/controllers

routes.js

```
...
GET     /                   Start.index
GET     /dashboard          Dashboard.index
GET     /about              About.index
...
```

Start.java

```
package controllers;
…

public class About extends Controller
{
  public static void index() {
    Logger.info("Rendering about");
    render ("about.html");
  }
}
```

Dashboard.java

```
package controllers;
…

public class Dashboard extends Controller
{
  public static void index() {
    Logger.info("Rendering Dashboard");
    render ("dashboard.html");
  }
}
```

About.java

```
package controllers;
…

public class Start extends Controller
{
  public static void index() {
    Logger.info("Rendering Start");
    render ("start.html");
  }
}
```

controllers/views

```
package controllers;
…

public class About extends Controller
{
  public static void index() {
    Logger.info("Rendering about");
    render ("about.html");
  }
}
```

start.html

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

#{menu id:"start"/}

<section class="ui center aligned middle aligned segment">
  <h1 class="ui header">
    Welcome to Template 1
  </h1>
  <p>
    To be replaced with content...
  </p>
</section>
```

```
package controllers;
…

public class Dashboard extends Controller
{
  public static void index() {
    Logger.info("Rendering Dashboard");
    render ("dashboard.html");
  }
}
```

dashboard.html

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

#{menu id:"dashboard"/}

<section class="ui segment">
  #{mainpanel/}
</section>
```

```
package controllers;
…

public class Start extends Controller
{
  public static void index() {
    Logger.info("Rendering Start");
    render ("start.html");
  }
}
```

about.html

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

#{menu id:"about"/}

<section class="ui center aligned middle aligned segment">
  <p>
    About this app
  </p>
</section>
```