

Todo List



todolist application

Todo List

Dashboard

About

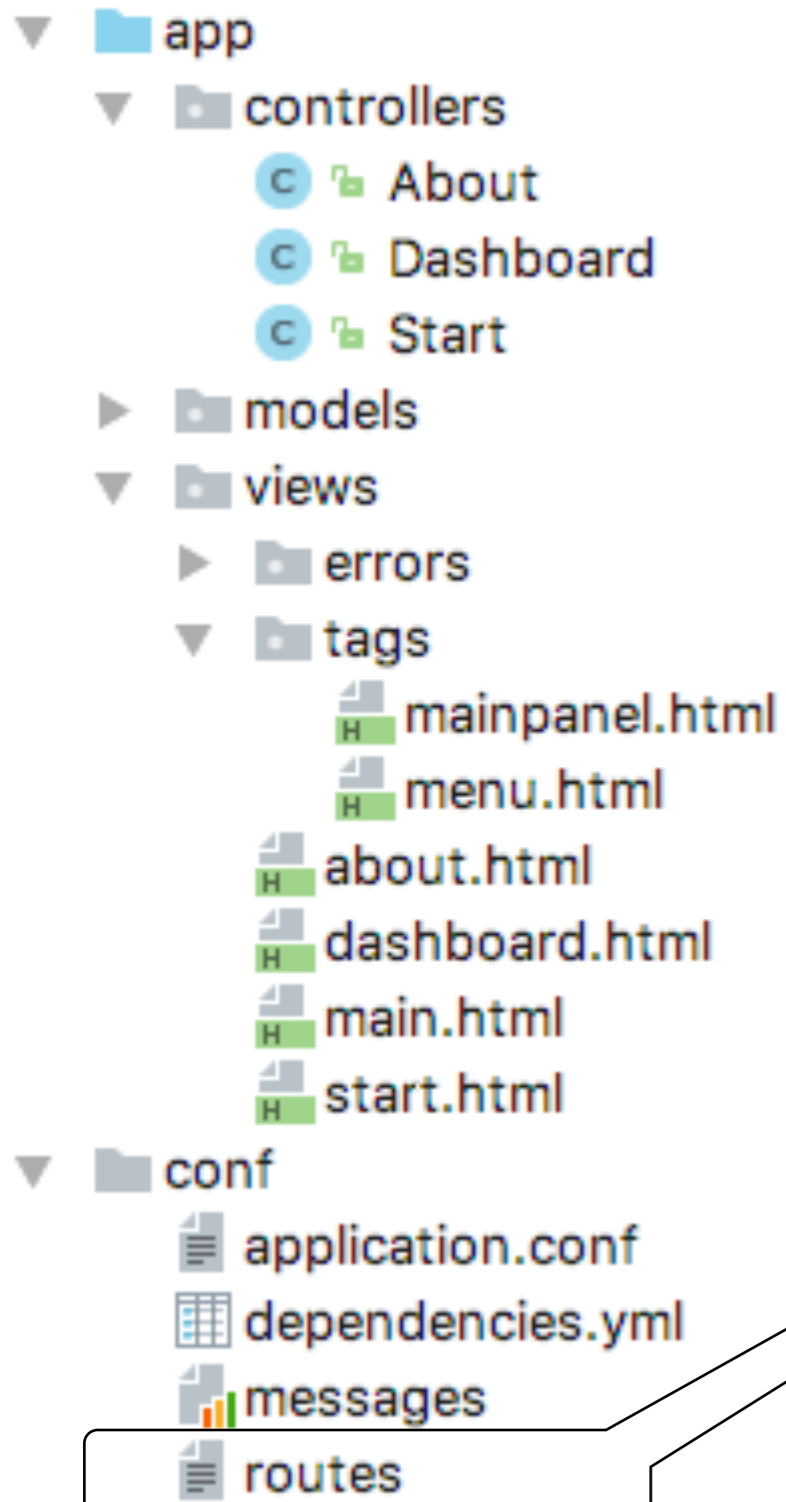
Todo List

- Make tea
- Go for snooze
- Make more tea

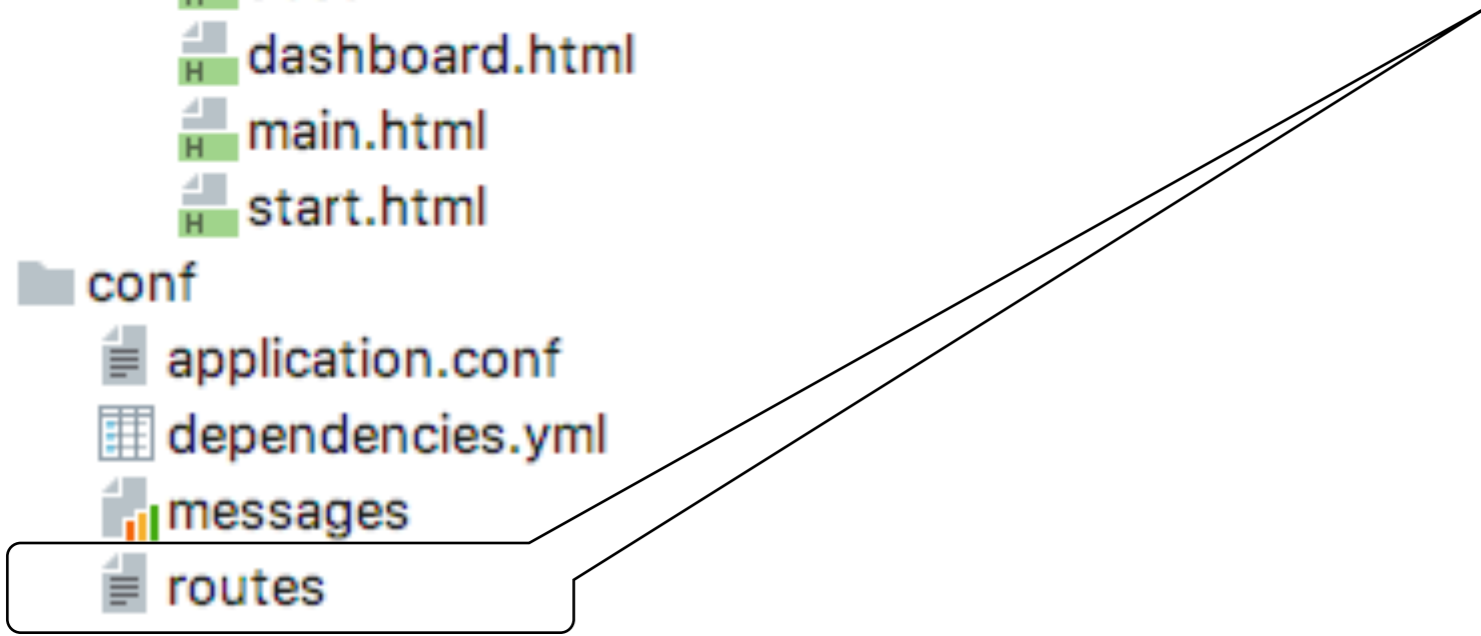
Title

Add Todo

routes



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

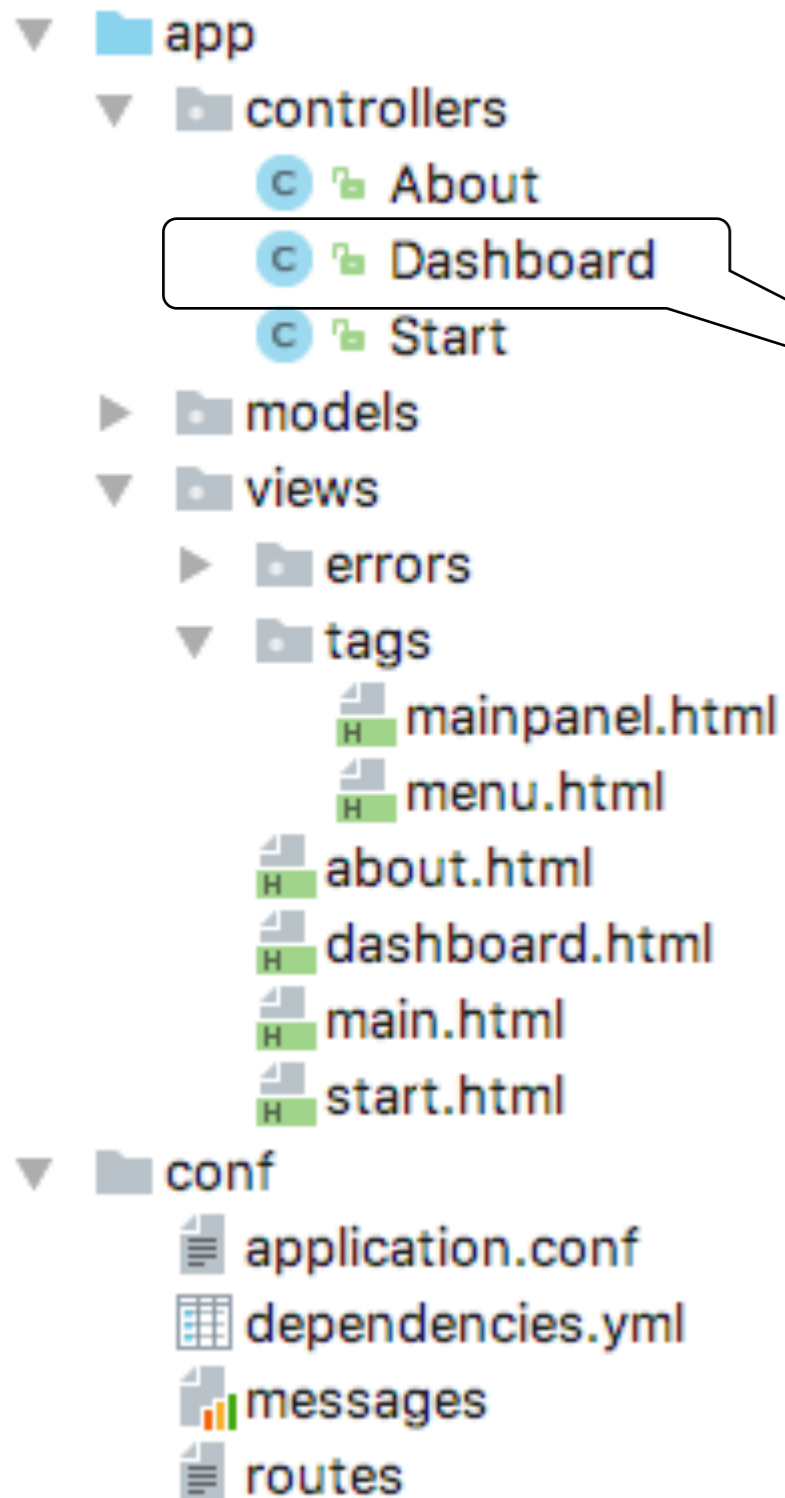


Dashboard routes

- ▼ app
 - ▼ controllers
 - Ⓢ About
 - Ⓢ Dashboard
 - Ⓢ Start
 - ▶ models
 - ▼ views
 - ▶ errors
 - ▼ tags
 - mainpanel.html
 - menu.html
 - about.html
 - dashboard.html
 - main.html
 - start.html
- ▼ conf
 - application.conf
 - dependencies.yml
 - messages
 - routes

GET	/dashboard	Dashboard.index
-----	------------	-----------------

Dashboard Controller



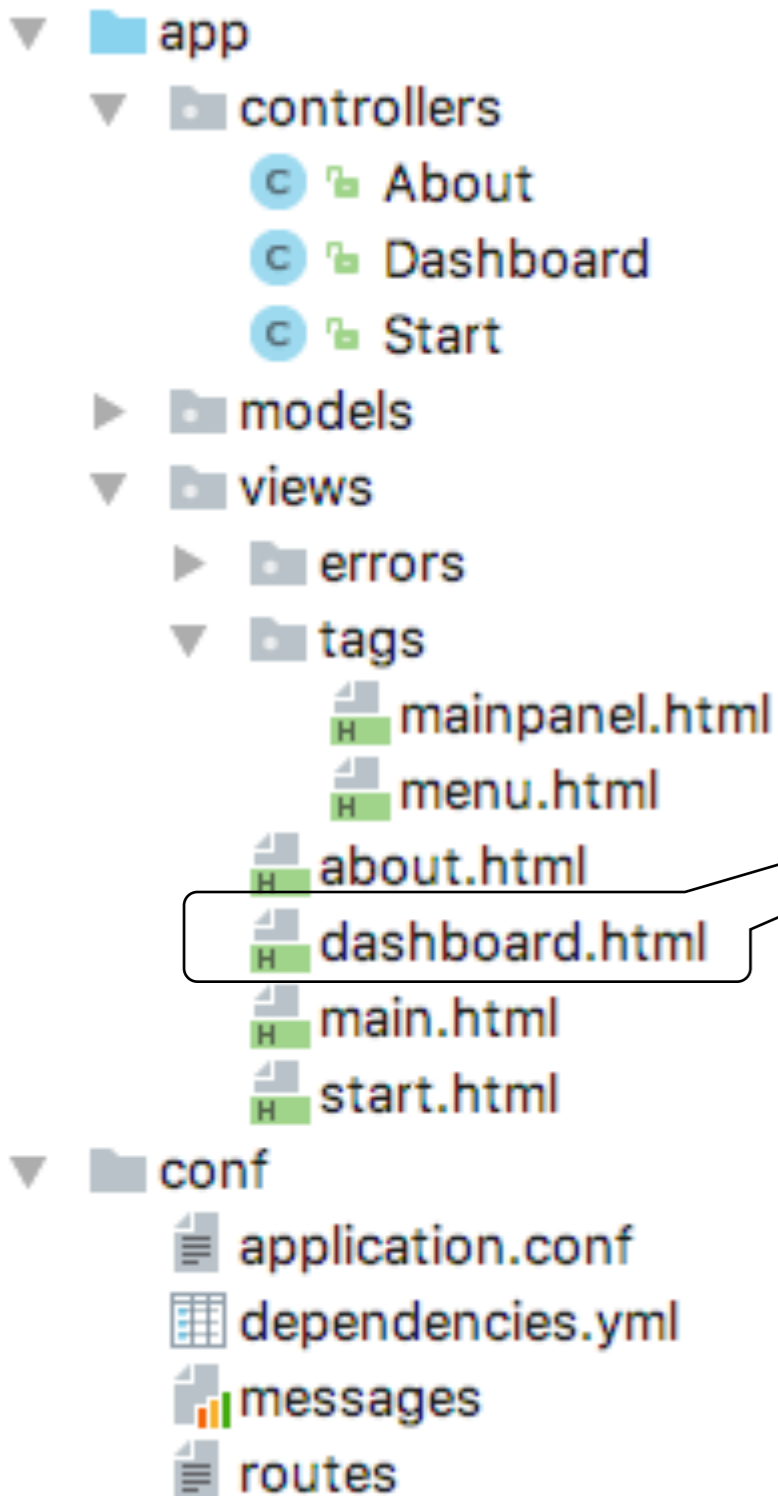
GET	/dashboard	Dashboard.index
-----	------------	-----------------

```
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");
    }
}
```

Dashboard View



```
#{extends 'main.html' /}  
#{set title:'Dashboard' /}  
  
#{menu id:"dashboard"/}  
  
<section class="ui raised segment">  
  <header class="ui header">  
    Todo List  
  </header>  
  <div class="ui bulleted list">  
    <div class="ui item"> Make tea</div>  
    <div class="ui item"> Go for snooze</div>  
    <div class="ui item"> Make more tea</div>  
  </div>  
</section>
```

Todo List

Todo List

- Make tea
- Go for snooze
- Make more tea

Title

Title

Add Todo

“Static” Todo List Segment

Todo List

- Make tea
- Go for snooze
- Make more tea

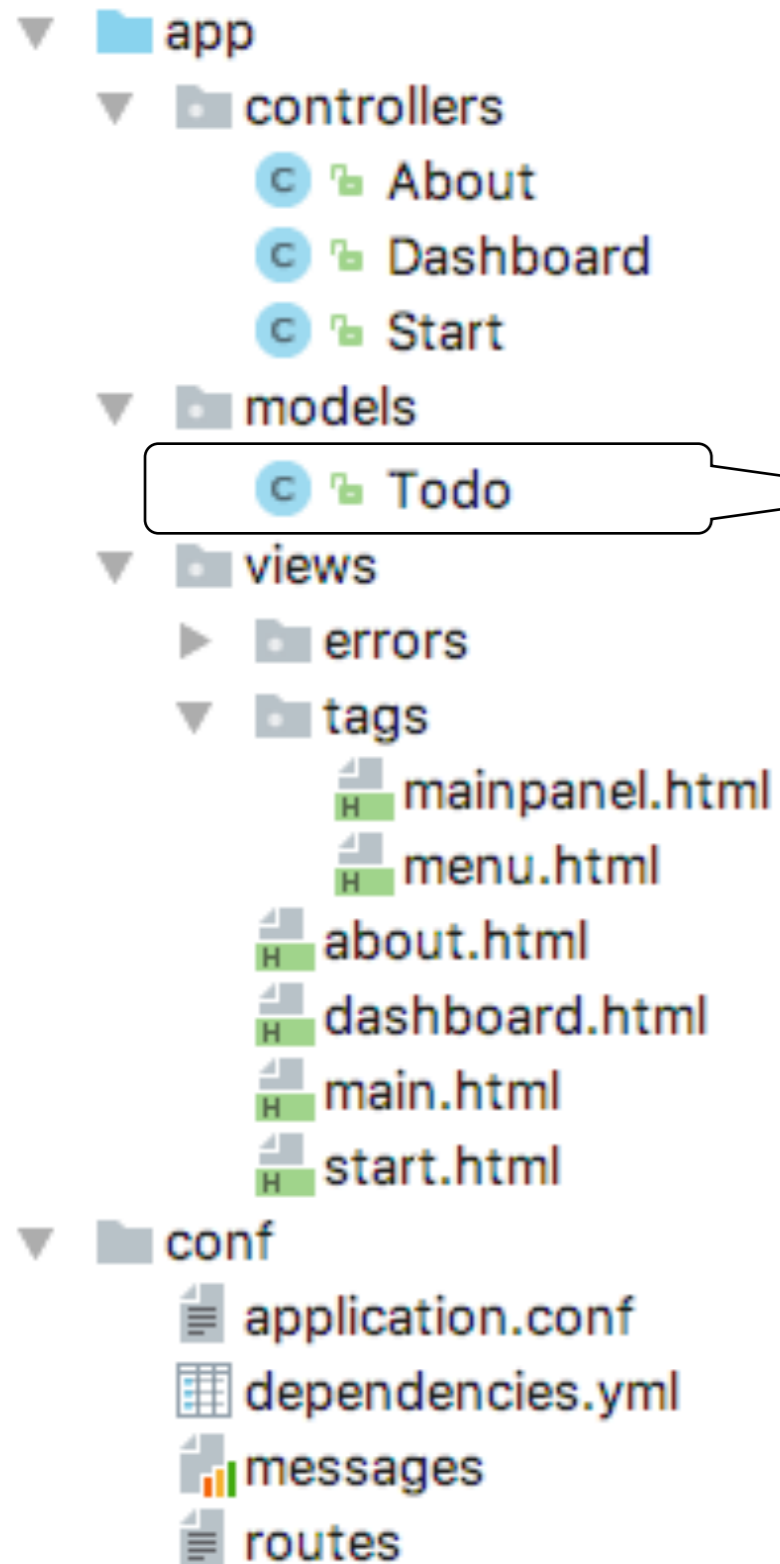
```
...  
<section class="ui raised segment">  
  <header class="ui header">  
    Todo List  
  </header>  
  <div class="ui bulleted list">  
    <div class="ui item"> Make tea</div>  
    <div class="ui item"> Go for snooze</div>  
    <div class="ui item"> Make more tea</div>  
  </div>  
</section>  
...
```

Add Todo List Form

Title


```
...  
<form class="ui stacked segment form" action="/dashboard/addtodo" method="POST">  
  <div class="field">  
    <label>Title</label>  
    <input placeholder="Title" type="text" name="title">  
  </div>  
  <button class="ui blue submit button">Add Todo</button>  
</form>
```


Todo Model



```
package models;

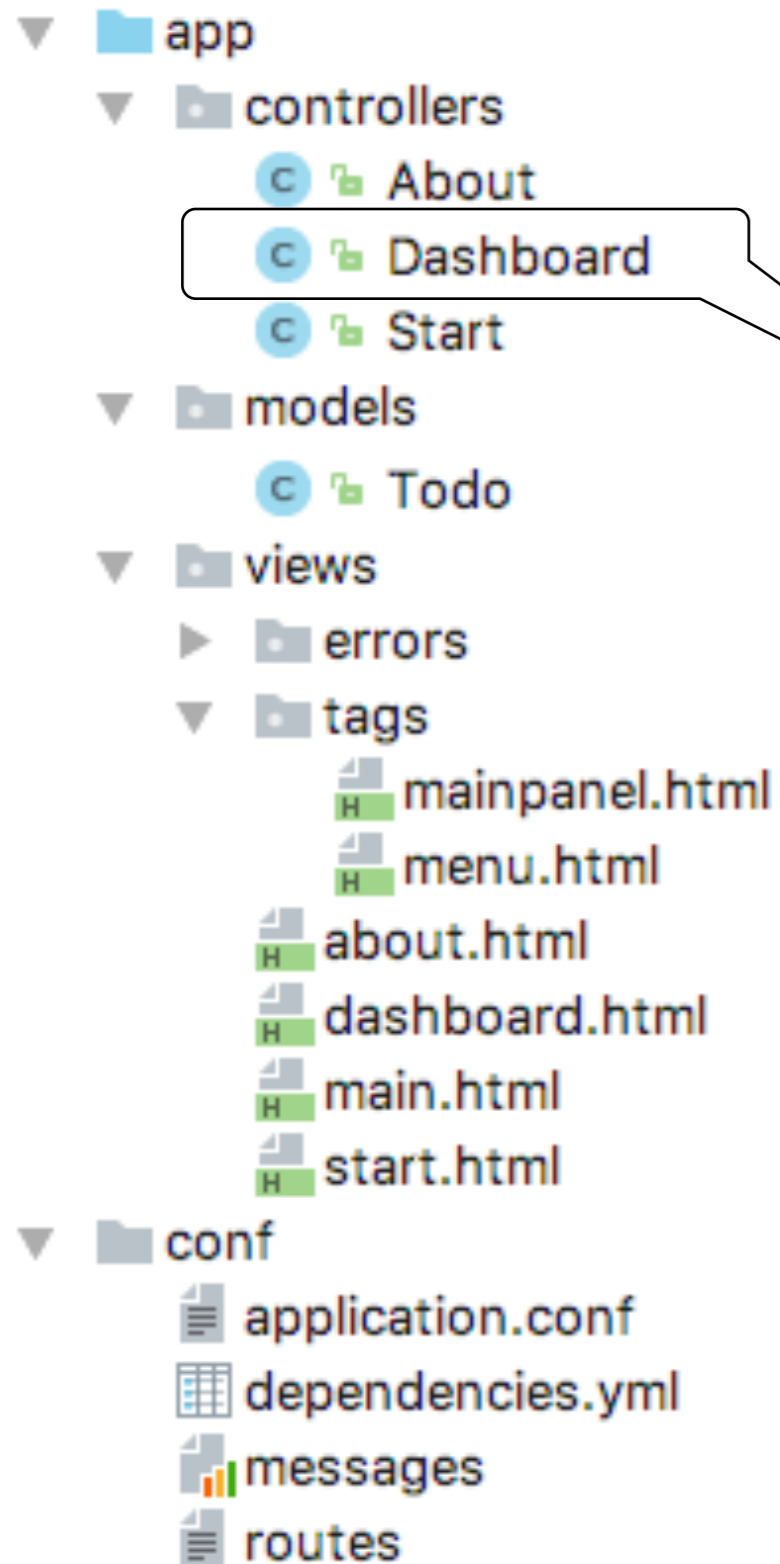
import play.db.jpa.Model;

import javax.persistence.Entity;

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

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

Revised Dashboard Controller



GET	/dashboard	Dashboard.index
POST	/dashboard/addtodo	Dashboard.addTodo

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

    public static void addTodo(String title)
    {
        Todo todo = new Todo(title);
        todo.save();
        Logger.info("Addint Todo" + title);
        redirect("/dashboard");
    }
}
```

Dashboard Controller - List Todos

GET	/dashboard	Dashboard.index
-----	------------	-----------------

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

```
...
<section class="ui raised segment">
  <header class="ui header">
    Todo List
  </header>
  <div class="ui bulleted list">
    #{list items: todolist, as: 'todo'}
    <div class="ui item"> ${todo.title} </div>
  </div>
</section>
...
```

Todo List

- Make tea
- Go for snooze
- Make more tea

“Dynamic” Todo List Segment

Todo List

- Make tea
- Go for snooze
- Make more tea

```
List<Todo> todolist = Todo.findAll();  
render("dashboard.html", todolist);
```

```
...  
<section class="ui raised segment">  
  <header class="ui header">  
    Todo List  
  </header>  
  <div class="ui bulleted list">  
    <div class="ui item"> ${todo.title} </div>  
  </div>  
</section>  
...
```

Dashboard Controller - Add Todos

Title

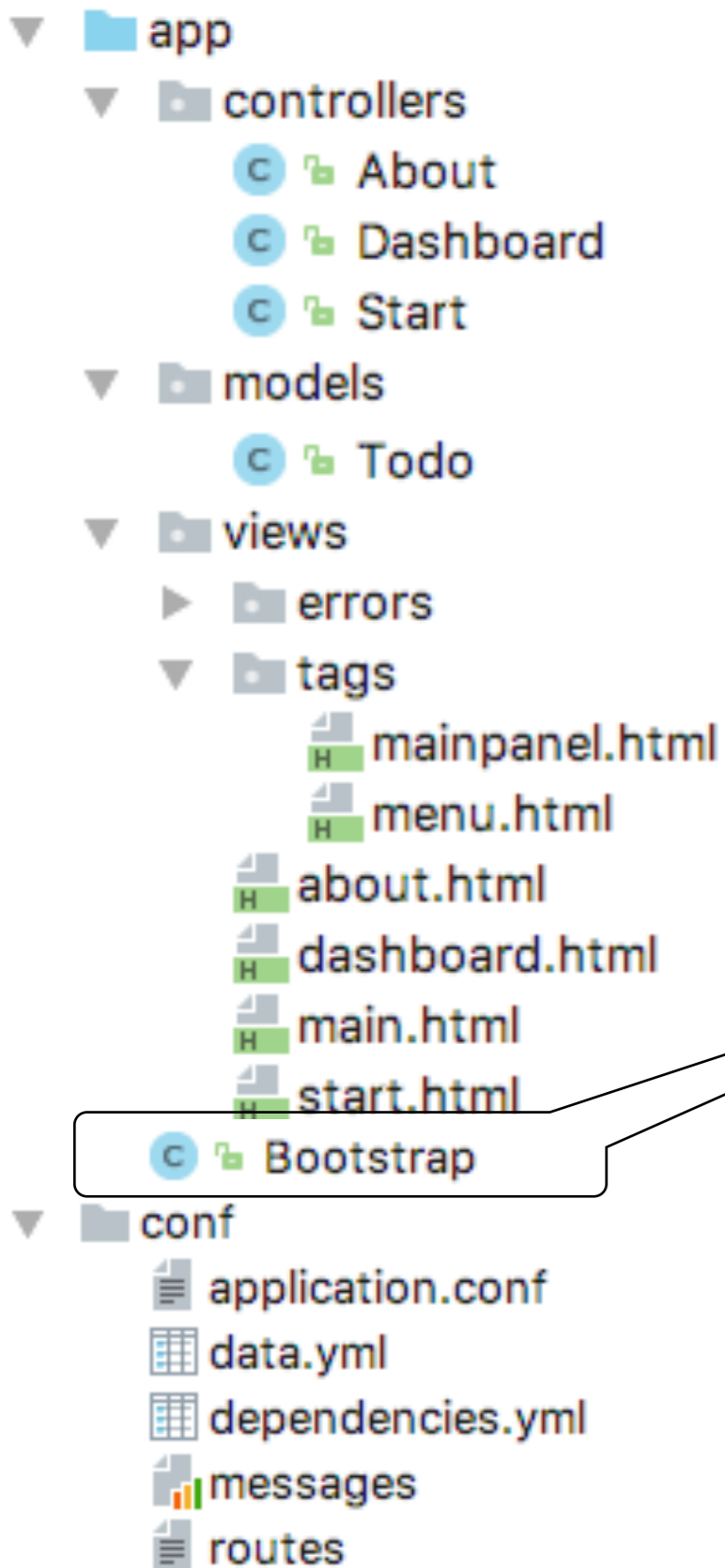
Add Todo

```
...  
<form class="ui stacked segment form" action="/dashboard/addtodo" method="POST">  
  <div class="field">  
    <label>Title</label>  
    <input placeholder="Title" type="text" name="title">  
  </div>  
  <button class="ui blue submit button">Add Todo</button>  
</form>
```

POST	/dashboard/addtodo	Dashboard.addToDo
------	--------------------	-------------------

```
public class Dashboard extends Controller  
{  
  ...  
  public static void addToDo(String title)  
  {  
    Todo todo = new Todo(title);  
    todo.save();  
    Logger.info("Addint Todo" + title);  
    redirect("/dashboard");  
  }  
}
```

Bootstrap

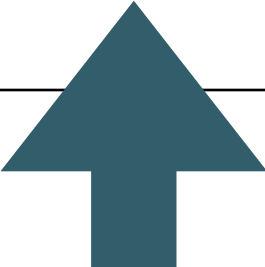


```
import java.util.List;

import play.*;
import play.jobs.*;
import play.test.*;

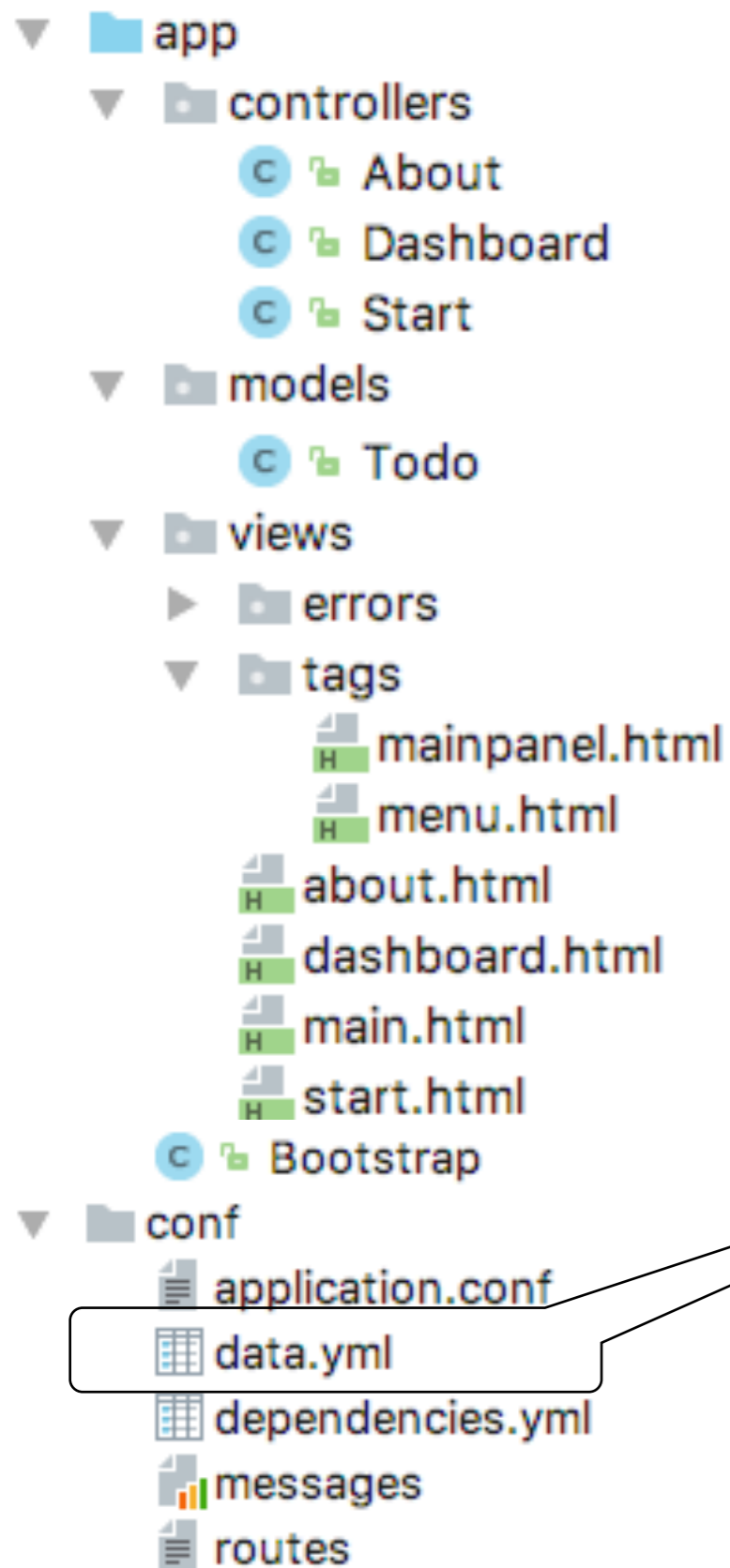
import models.*;

@OnApplicationStart
public class Bootstrap extends Job
{
    public void doJob()
    {
        Fixtures.loadModels("data.yml");
    }
}
```



This method executes every time application starts/restarts

Bootstrap



```
Todo(t1):  
  title: Make tea  
  
Todo(t2):  
  title: Go for snooze  
  
Todo(t3):  
  title: Make more tea
```



Todo Objects loaded
by Bootstrap

<http://localhost:9000/@db>

Login

Saved Settings:

Setting Name:

Driver Class:

JDBC URL:

User Name:

Password:

localhost:9000/@db

The screenshot shows a database client interface with a toolbar at the top containing icons for undo, redo, auto-commit, and other settings. The main area is divided into a left sidebar with a tree view and a central SQL editor. The tree view shows the following structure:

- jdbc:h2:mem:play
 - todo
 - information_schema
 - Sequences
 - Users
 - H2 1.4.193 (2016-10-31)

The SQL editor contains the query: `SELECT * FROM TODO`. Below the editor, the results are displayed in a table format:

```
SELECT * FROM TODO;
```

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

Below the table, the text "(3 rows, 5 ms)" indicates the execution statistics. An "Edit" button is located at the bottom of the results area.