

Todo Review

todolist app

- ▼ **todolist** ~/repos/modules/wek
 - ▶ **.idea**
 - ▼ **app**
 - ▼ **controllers**
 - Ⓢ **About**
 - Ⓢ **Admin**
 - Ⓢ **Dashboard**
 - Ⓢ **Start**
 - ▼ **models**
 - Ⓢ **Todo**
 - ▼ **views**
 - ▶ **errors**
 - ▼ **tags**
 - mainpanel.html
 - menu.html
 - about.html
 - admin.html
 - dashboard.html
 - main.html
 - start.html
 - Ⓢ **Bootstrap**
 - ▼ **conf**
 - application.conf
 - data.yml
 - dependencies.yml
 - messages
 - routes

[Todo List](#) [Dashboard](#) [About](#)

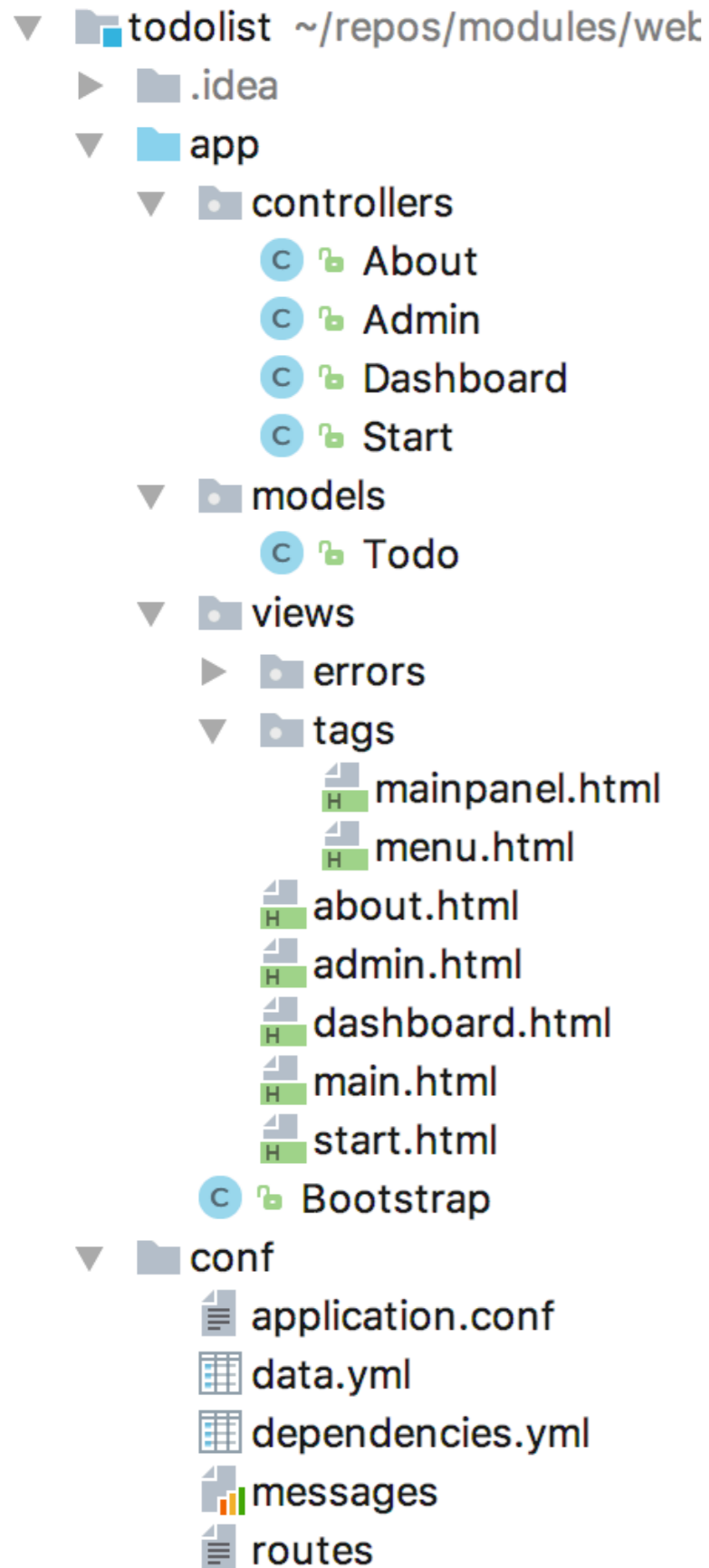
Todo List

Todo	
Make tea	Delete
Go for snooze	Delete
Make more tea	Delete

Title

[Add Todo](#)

Bootstrap Model Seeding



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

```
Todo(t1):
    title: Make tea

Todo(t2):
    title: Go for snooze

Todo(t3):
    title: Make more tea
```

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

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

Dashboard View

Todo List

Dashboard

About

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

#{menu id:"dashboard"/}

<section class="ui raised segment">
  <header class="ui header">
    Todo List
  </header>
  <table class="ui fixed table">
    <thead>
      <tr>
        <th>Todo</th>
        <th></th>
      </tr>
    </thead>
    <tbody>
      <div class="ui bulleted list">
        #{list items:todolist, as:'todo'}
        <tr>
          <td> ${todo.title} </td>
          <td> <a href="/dashboard/deletetodo/${todo.id}" class="ui tiny red button"> Delete </a> </td>
        </tr>
      </div>
    </tbody>
  </table>

</section>

<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 List

Todo

Make tea

Delete

Go for snooze

Delete

Make more tea

Delete

Title

Title

Add Todo

Dashboard Routes + Controller

GET	/dashboard	Dashboard.index
POST	/dashboard/addtodo	Dashboard.addTodo
GET	/dashboard/deletetodo/{id}	Dashboard.deleteTodo

```
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");
    }

    public static void deleteTodo(Long id)
    {
        Todo todo = Todo.findById(id);
        todo.delete();
        Logger.info("Deleting " + todo.title);
        redirect("/dashboard");
    }
}
```

Dashboard Index Action

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

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

```
...  
<div class="ui bulleted list">  
  #{list items: todolist, as: 'todo'}  
  <tr>  
    <td> ${todo.title} </td>  
    <td> <a href="/dashboard/deletetodo/${todo.id}" class="ui tiny red button"> Delete </a> </td>  
  #{/list}  
</div>  
</tbody>  
</table>  
...
```

Make tea	Delete
Go for snooze	Delete
Make more tea	Delete

Dashboard addTodo Action

POST /dashboard/addtodo

Dashboard.addTodo

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

```
<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>
```

Title

Add Todo

Dashboard deleteTodo Action

Make more tea

Delete

GET /dashboard/deletetodo/{id}

Dashboard.deleteTodo

```
public static void deleteTodo(Long id)
{
    Todo todo = Todo.findById(id);
    todo.delete();
    Logger.info("Deleting " + todo.title);
    redirect("/dashboard");
}
```

```
...
<div class="ui bulleted list">
  #{list items:todoList, as:'todo'}
  <tr>
    <td> ${todo.title} </td>
    <td> <a href="/dashboard/deletetodo/${todo.id}" class="ui tiny red button"> Delete </a> </td>
  #{/list}
</div>
</tbody>
</table>
...
```


<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 of the database structure and a central SQL editor. The SQL editor contains the query `SELECT * FROM TODO`. Below the editor, the results of the query are displayed as a table with three rows. The table has columns `ID` and `TITLE`. The rows contain the following data: (1, Make tea), (2, Go for snooze), and (3, Make more tea). Below the table, the text `(3 rows, 5 ms)` indicates the number of rows and execution time. An `Edit` button is located below the table.

jdbc:h2:mem:play

- + todo
- + information_schema
- + Sequences
- + Users
- i H2 1.4.193 (2016-10-31)

Run Run Selected Auto complete Clear SQL statement:

```
SELECT * FROM TODO
```

```
SELECT * FROM TODO;
```

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

(3 rows, 5 ms)

Edit