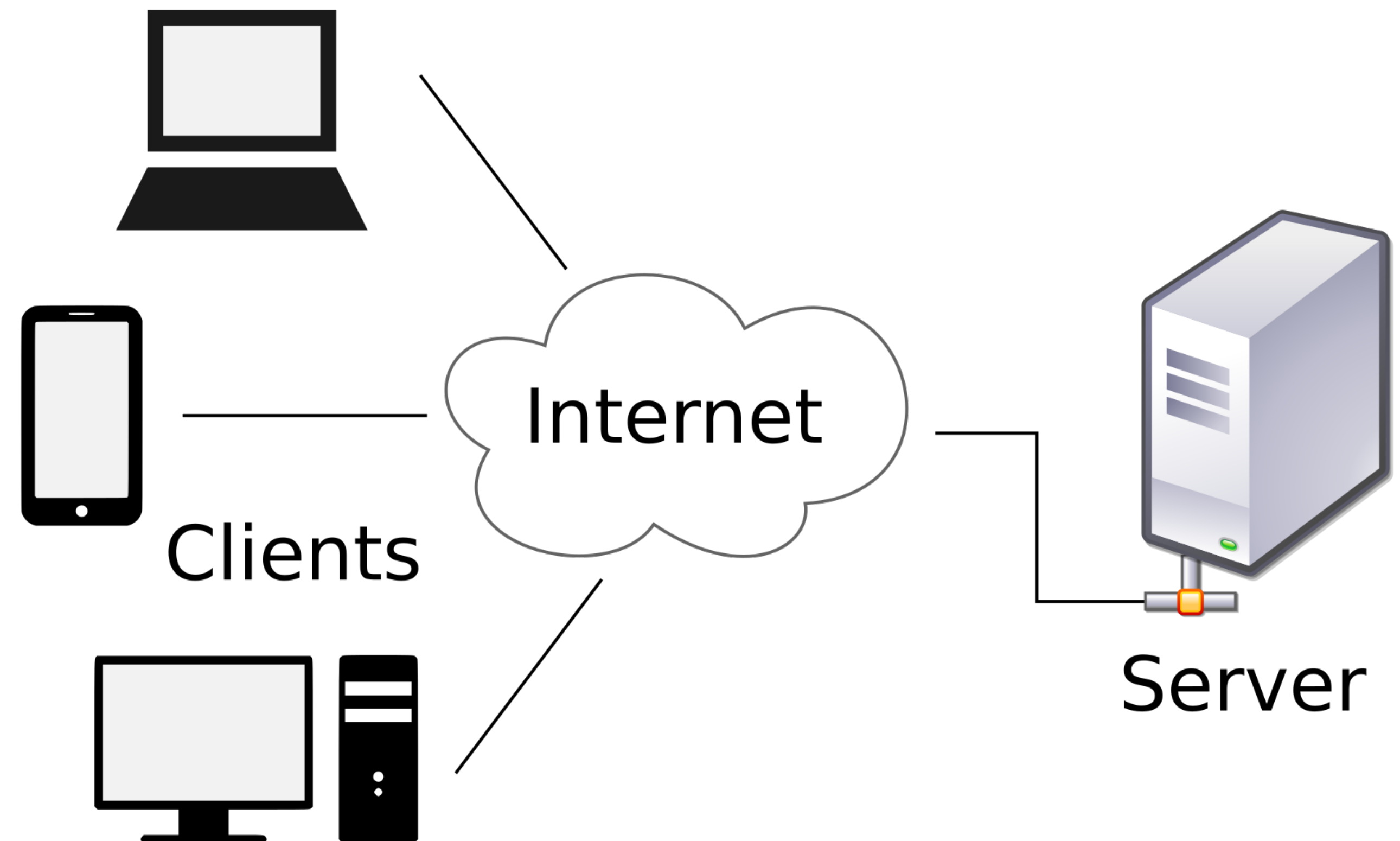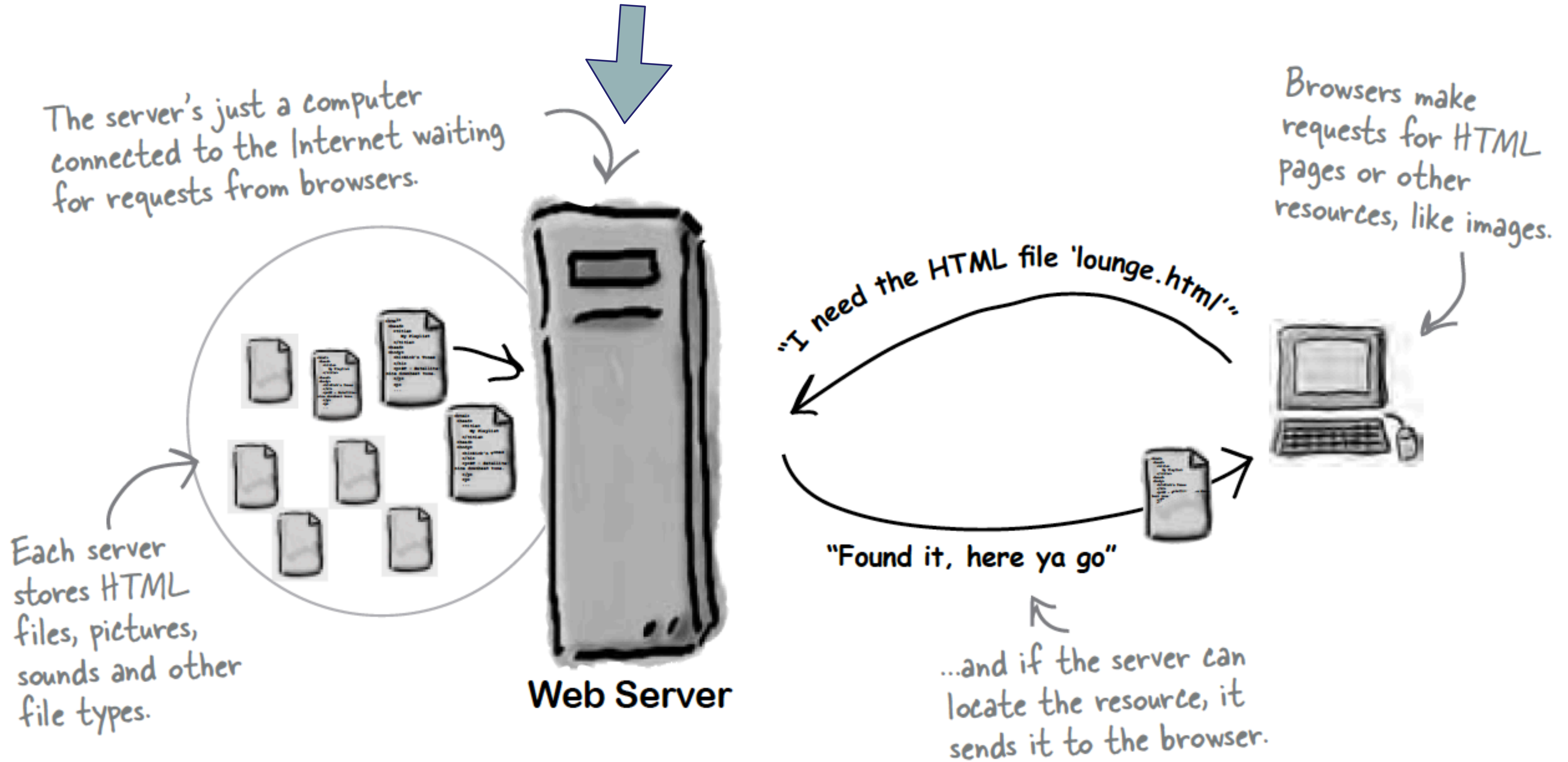# Web App Structure

structure · front-end · backend · router · model · view · controller
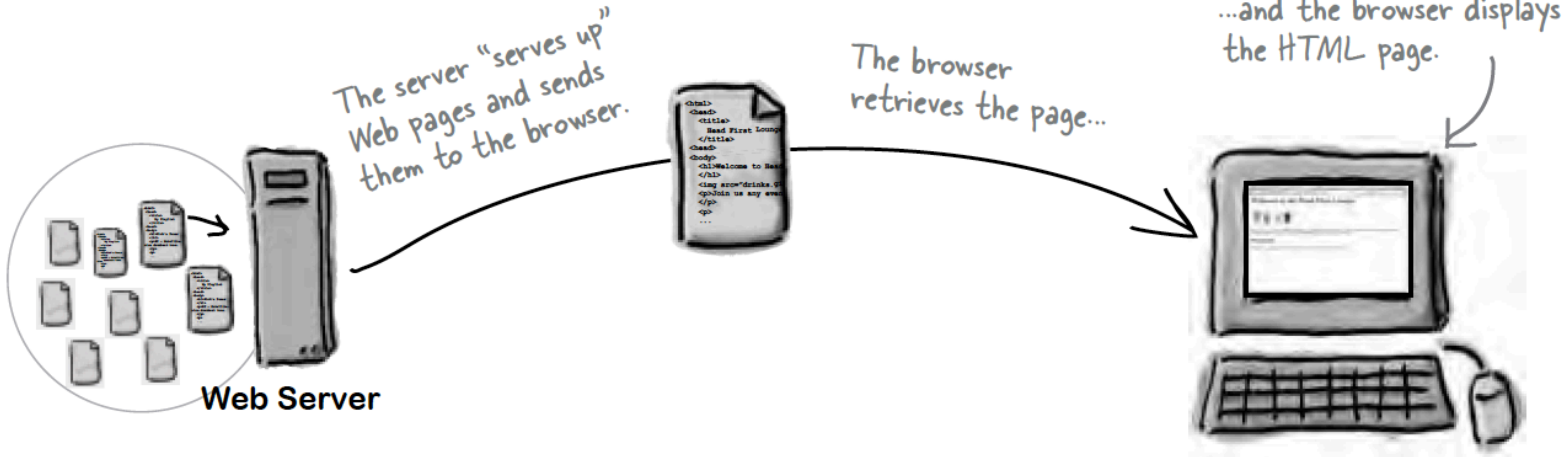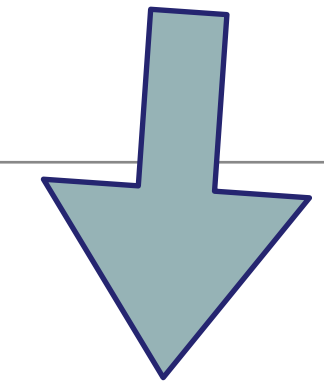
Clients and Servers

- Client/Server Computing:

  - The interaction between two programs when they communicate across a network.

  - A program at one site sends a request to a program at another site and awaits a response.

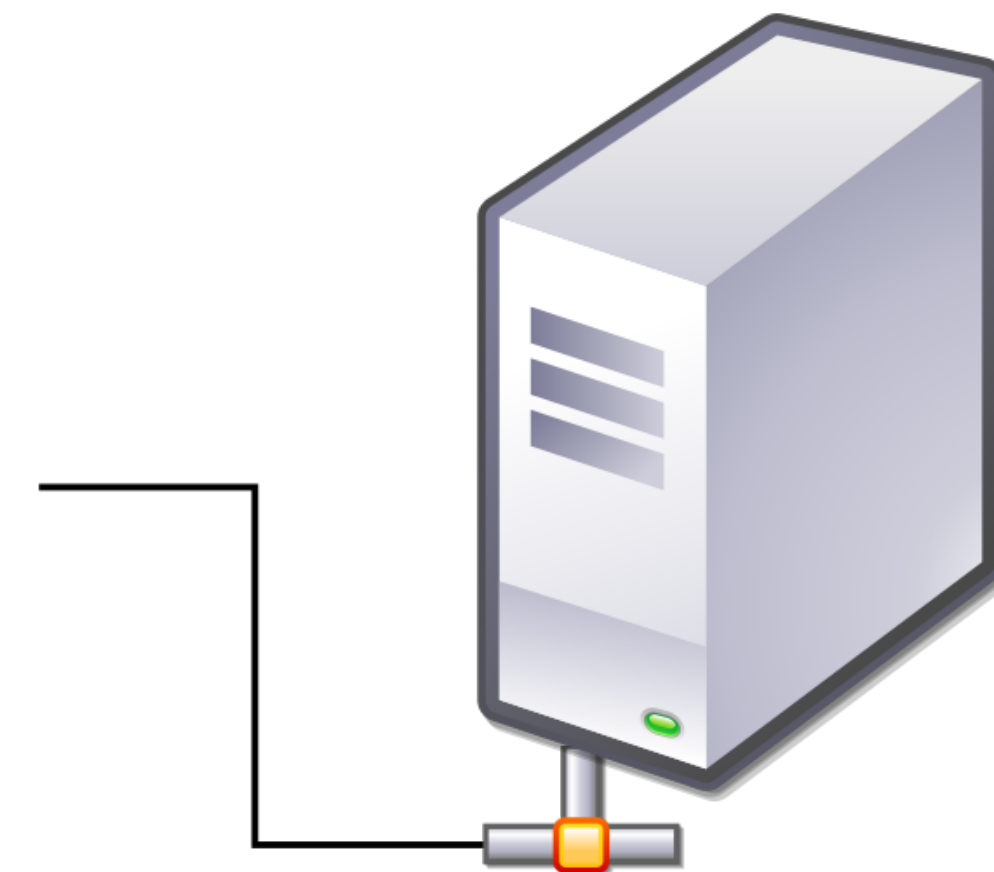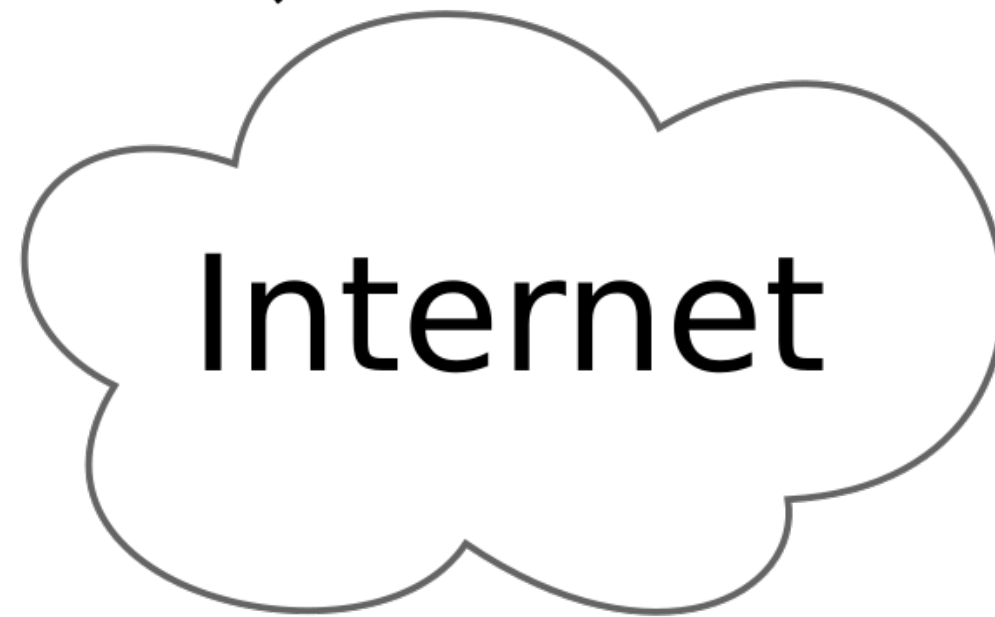 - The requesting program is called a client; the program satisfying the request is called the server.

Clients

Internet

Server

Role of Server

The server's just a computer connected to the Internet waiting for requests from browsers.

Browsers make requests for HTML pages or other resources, like images.

Each server stores HTML files, pictures, sounds and other file types.

Web Server

"I need the HTML file 'lounge.html'"

"Found it, here ya go"

...and if the server can locate the resource, it sends it to the browser.

Role of Client

...and the browser displays the HTML page.

The server "serves up" Web pages and sends them to the browser.

The browser retrieves the page...

```
<html>
<head>
<title>
Head First Lounge
</title>
<head>
<body>
<h1>Welcome to Head
</h1>
<img src="drinks.gi
<p>Join us any even
</p>
<p>
...
```
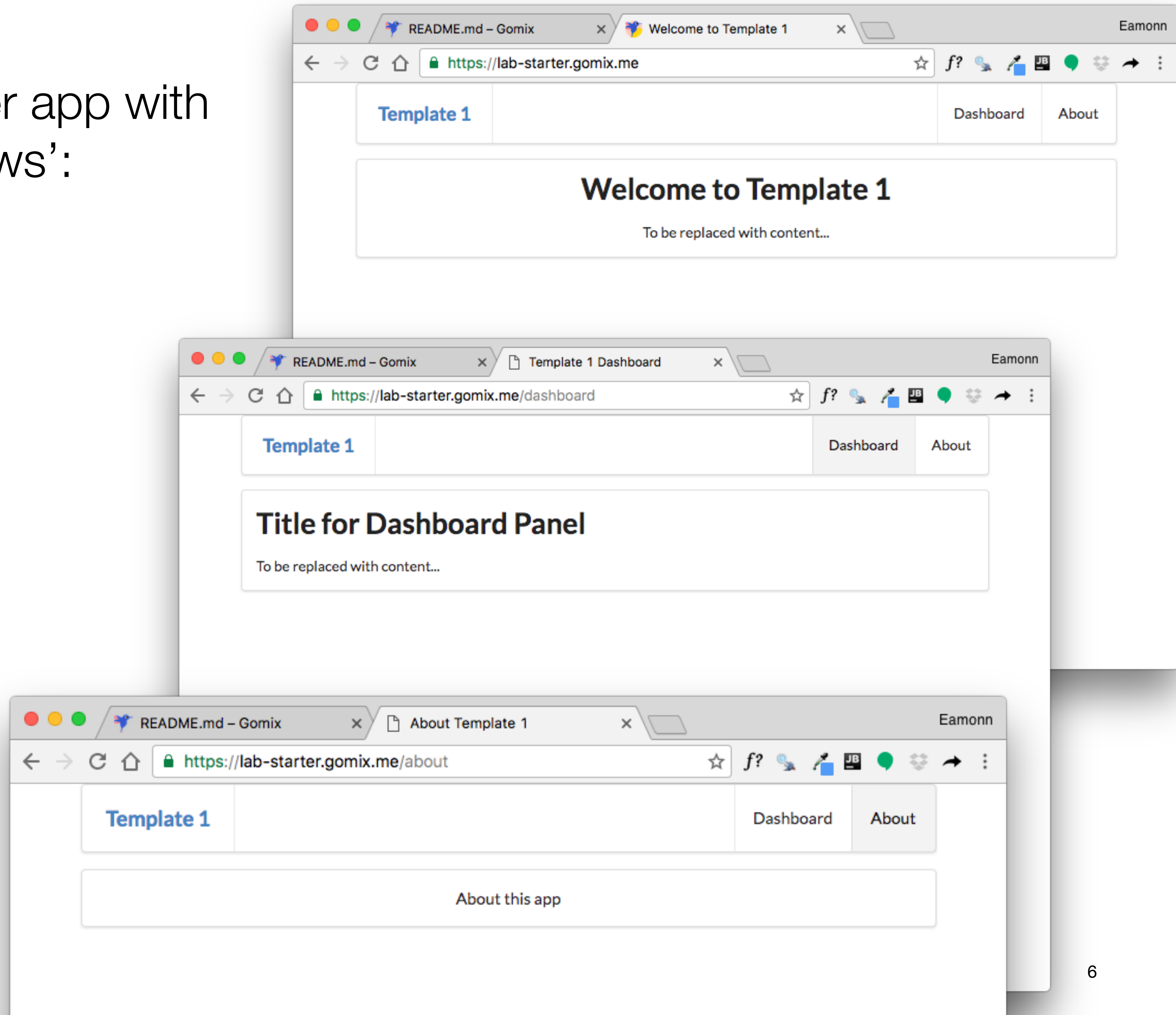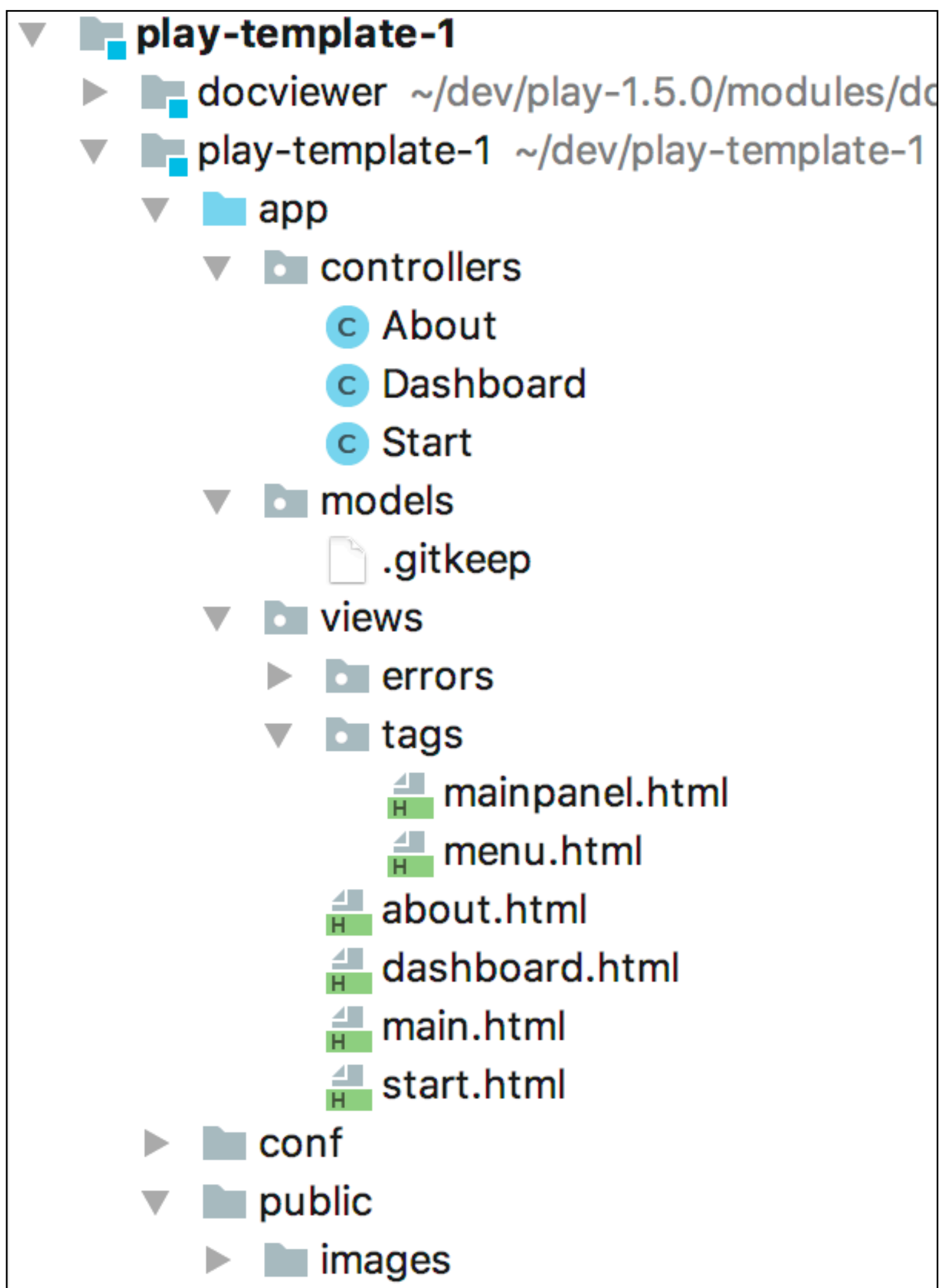
Web Server

Front End

Clients

Internet

Back End

Server

- Consider a starter app with a menu + 3 'Views':

  - Start

  - Dashboard

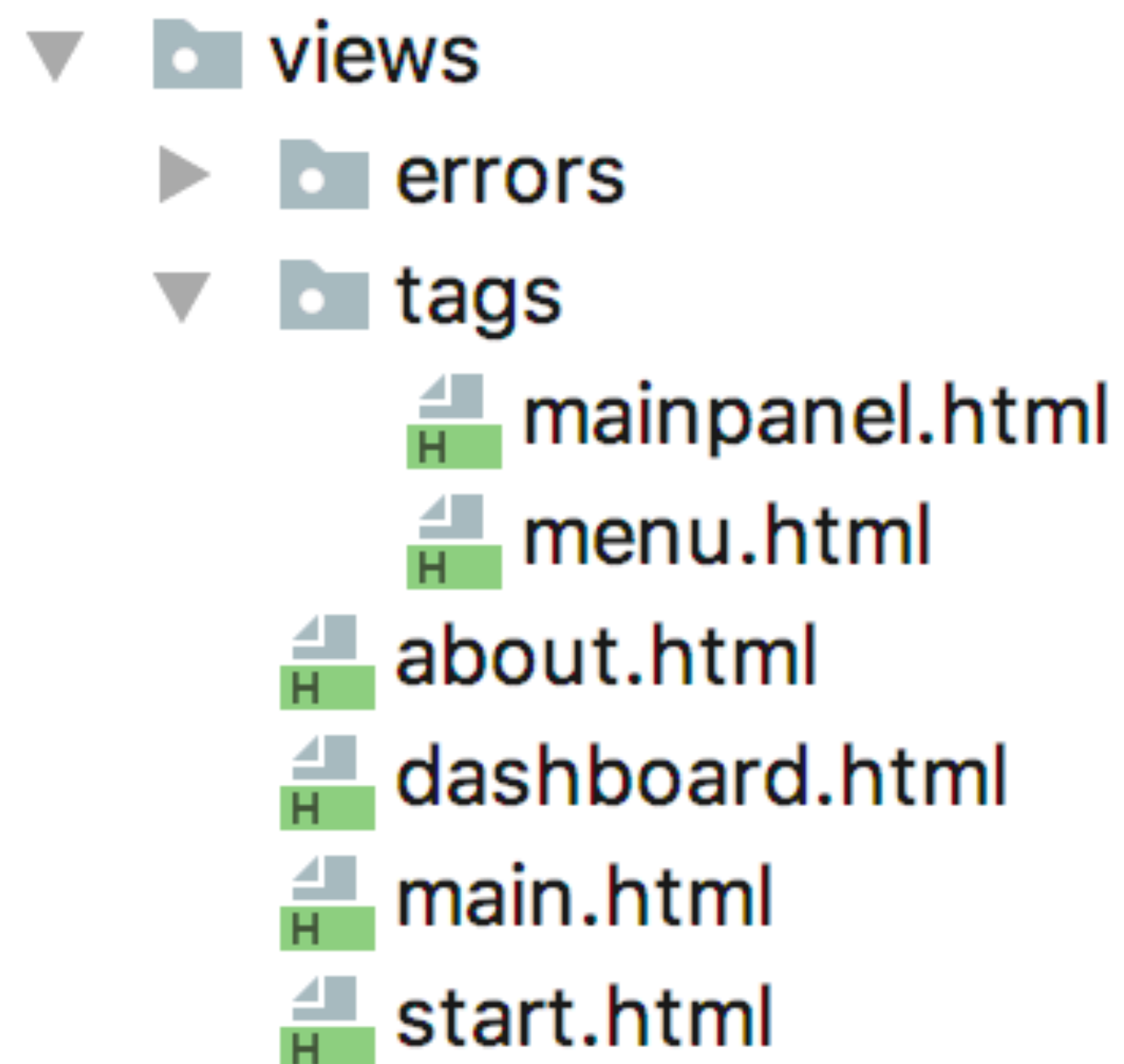 - About

Application Structure - Intellij Idea Java Project

```
▼ 🖿 play-template-1
  ▶ 🖿 docviewer  ~/dev/play-1.5.0/modules/dc
  ▼ 🖿 play-template-1  ~/dev/play-template-1
    ▼ 📁 app
      ▼ 📁 controllers
        🅒 About
        🅒 Dashboard
        🅒 Start
      ▼ 📁 models
        📄 .gitkeep
      ▼ 📁 views
        ▶ 📁 errors
        ▼ 📁 tags
          📄 mainpanel.html
          📄 menu.html
        📄 about.html
        📄 dashboard.html
        📄 main.html
        📄 start.html
    ▶ 📁 conf
    ▼ 📁 public
      ▶ 📁 images
```

- App implements Routes + Model/View/ Controller Architecture

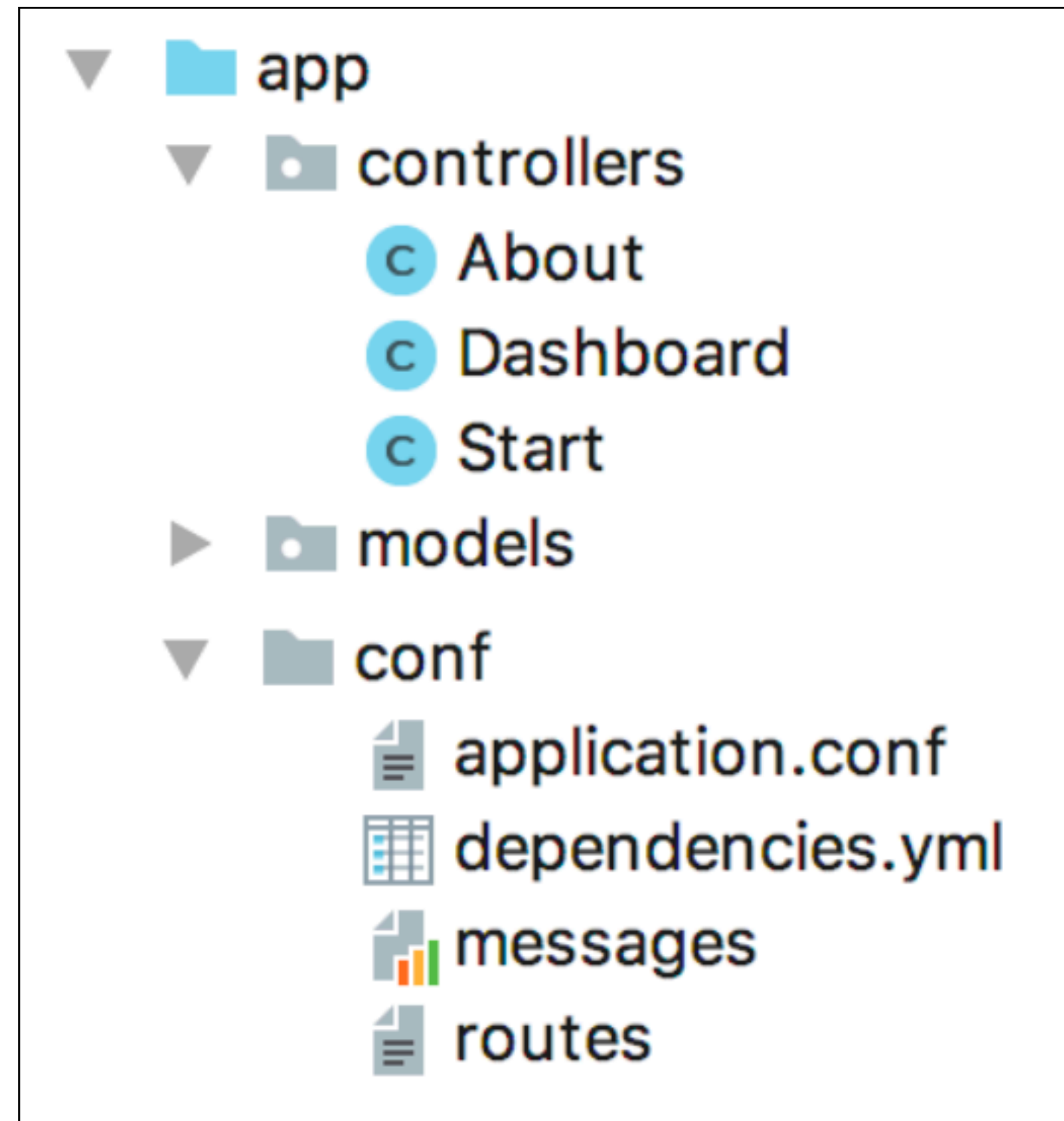- Back-end + Front-end collaborate to support structured, predictable application workflow

Front-end


```
▼  📁 views
   ▶  📁 errors
   ▼  📁 tags
         📄 mainpanel.html
         📄 menu.html
      📄 about.html
      📄 dashboard.html
      📄 main.html
      📄 start.html
```

- All written in HTML + templating

- Templating language supports:

  - **Layouts**

  - **Partials**
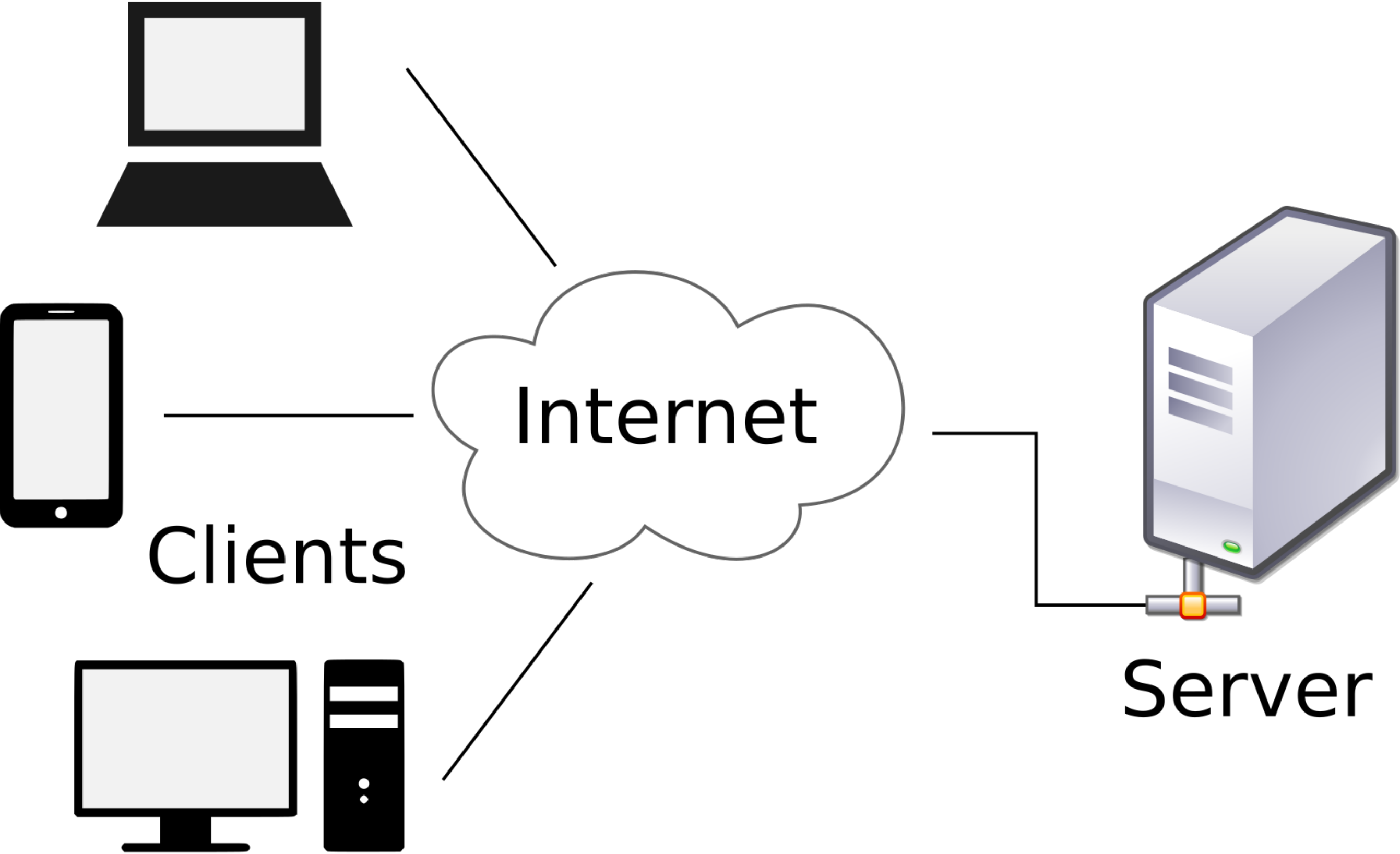
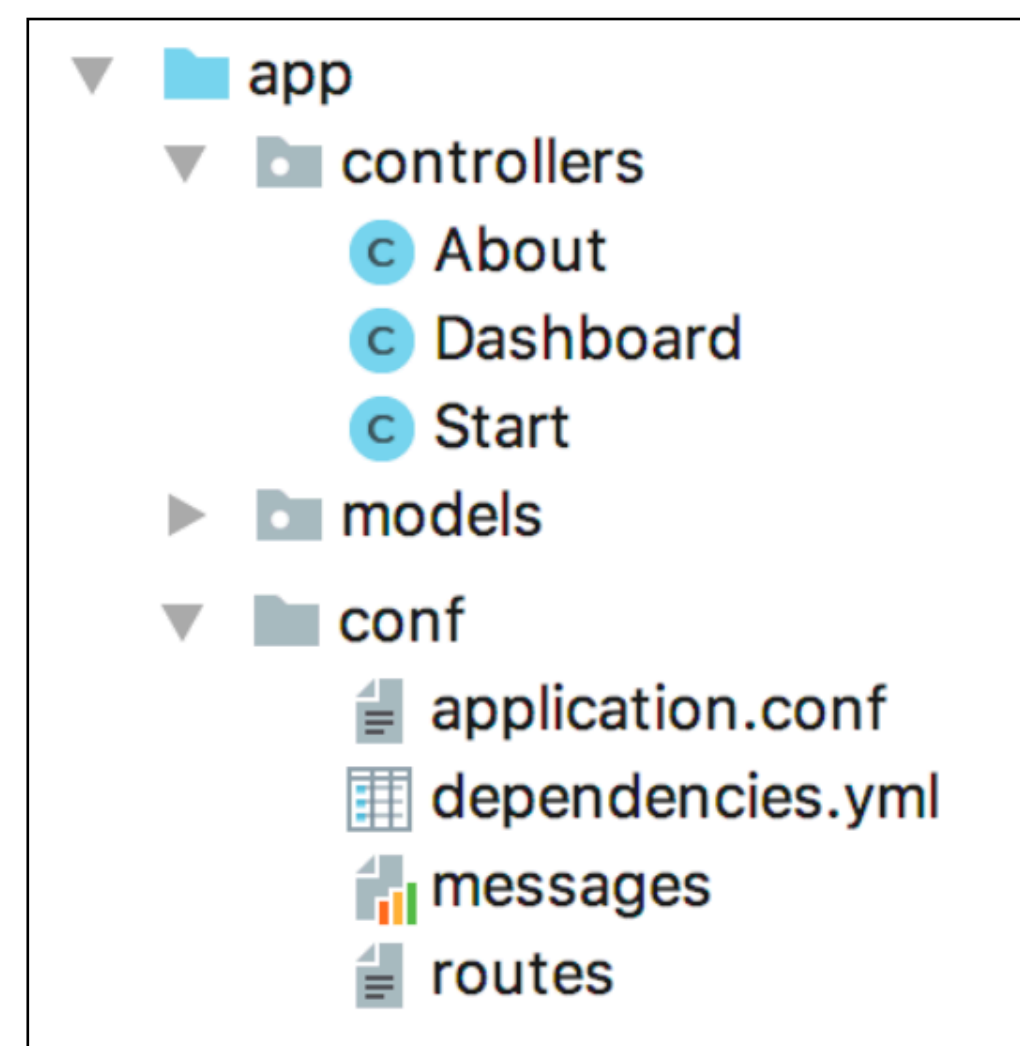 - **Views**

- These are similar to EJS equivalents

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…

views
- errors
- tags
  - mainpanel.html
  - menu.html
- about.html
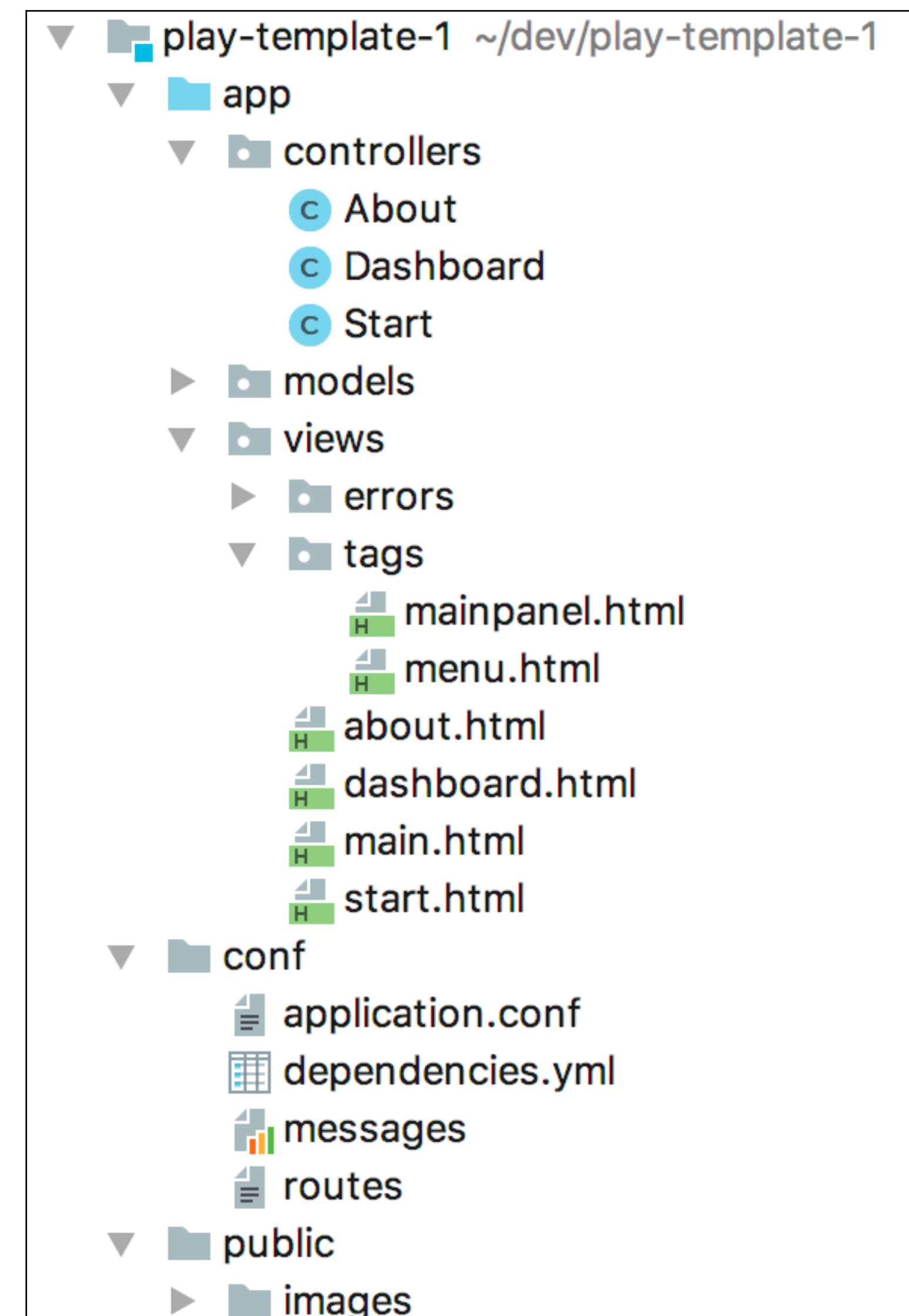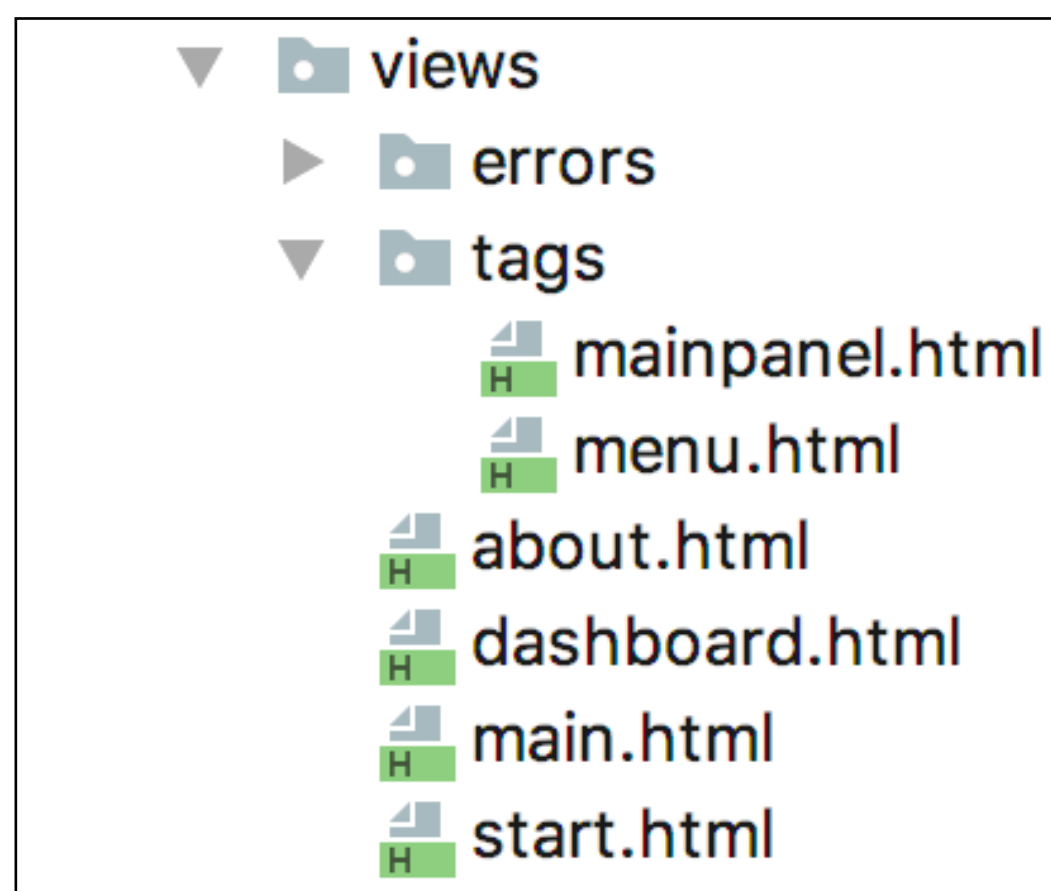- dashboard.html
- main.html
- start.html

Clients

Internet

Server

app
- controllers
  - About
  - Dashboard
  - Start
- models
- conf
  - application.conf
  - dependencies.yml
  - messages
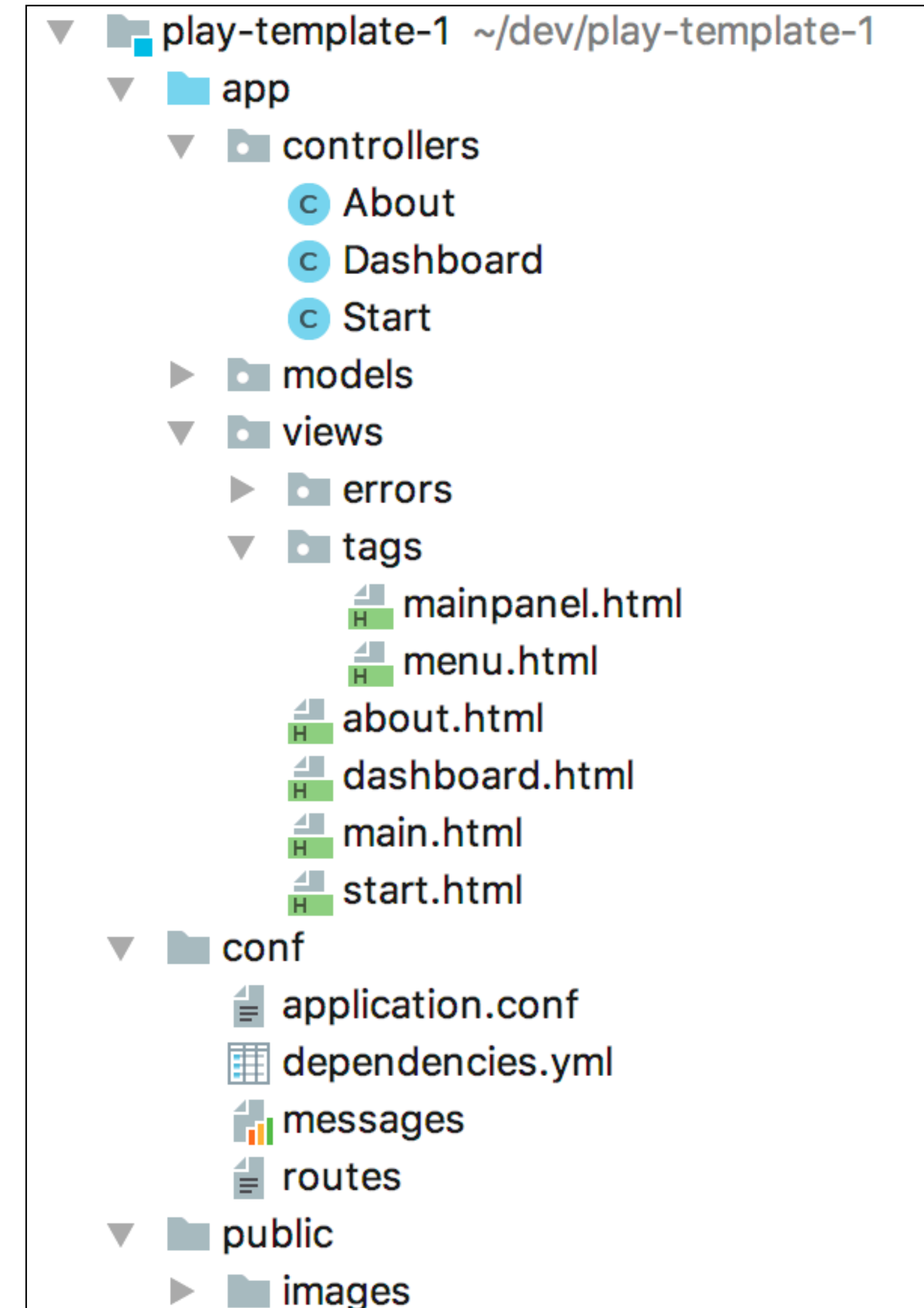 - routes

Back-end + Front-end

+

=

Back-end + Front-end

- To build a new app:

  - Extend, modify, replace and create new routes
    + controllers

  - Rewrite templates to support new features,
    data and user experience.

```
▼ 🗂 play-template-1 ~/dev/play-template-1
  ▼ 📁 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
  ▼ 📁 public
    ▶ 🗂 images
```

Starter App

- We will always start a new project by importing a template project from github.

- play-template-1 is a 'boilerplate' project providing foundation components for a starter application.

- Extend/repurpose this project to build interesting applications.

Web App Structure

structure · front-end ·
backend · router · model ·
view · controller