

# Web App Introduction

---

Web App Introduction   




PLAY VIDEO

Structure of a web app:  
Front-end Vs Backend.  
Routers, Models, Views,  
Controllers

# Starter App

- We will always start a new project by importing a template project from github.
- glitch-template-1 is a 'boilerplate' project providing foundation components for a starter application.
- Extend/repurpose this project to build interesting applications.

The screenshot shows the GitHub repository page for 'wit-hdip-comp-sci-2019 / glitch-template-1'. The repository has 1 star, 0 forks, and 0 issues. The main content area shows a list of files and folders with their commit history. The latest commit is by 'edelestar' adding 'eslint', made 1 hour ago. The repository includes folders for 'controllers', 'utils', and 'views', and files for '.eslintrc.json', '.gitignore', 'README.md', 'package.json', 'routes.js', and 'server.js'. The README.md file is expanded, showing a welcome message and project description.

| File/Folder    | Commit Message   | Commit Time |
|----------------|------------------|-------------|
| controllers    | baseline version | yesterday   |
| utils          | baseline version | yesterday   |
| views          | baseline version | yesterday   |
| .eslintrc.json | eslint added     | 1 hour ago  |
| .gitignore     | baseline version | yesterday   |
| README.md      | baseline version | yesterday   |
| package.json   | baseline version | yesterday   |
| routes.js      | baseline version | yesterday   |
| server.js      | baseline version | yesterday   |

**README.md**

## Welcome to the Glitch Template 1

A starter project for learning Glitch.

This is an Express.js project, designed to work well with the Glitch development environment. It includes basic express setup, templating, routing and session support.

# Import from Github...

glitch.com says

Paste the full URL of your repository

`https://github.com/wit-hdip-comp-sci-2019/glitch-template-1`

Cancel OK

**Your Projects** →

**wit-hdip-comp-sci-2019-glitch-template-1-1**  
Base project for the git cloning

**hello-express**  
A simple Node app built on Express, instantly up and running.

**hello-sqlite**  
A simple Node app with a SQLite database to hold app data.

**Clone from Git Repo**

**related-mechanic-1**  
e project that does c things

**Help Others, Get Thanks** →

Looks like nobody is asking for help right now. [Learn about helping](#)

dashboard.js - wit-hdip-comp- x +

https://glitch.com/edit/#!/wit-hdip-comp-sci-2019-github-template-1-2?path=controllers/dashboard...

wit-hdip-comp-sci-2019... Show controllers/dashboard.js

Share New File

- assets
- controllers/about.js
- controllers/dashboard.js
- utils/logger.js
- views/layouts/main.hbs
- views/partials/mainpanel.hbs
- views/partials/menu.hbs
- views/about.hbs
- views/dashboard.hbs
- .eslintrc.json
- .gitignore
- README.md
- package.json
- routes.js
- server.js

```
1 'use strict';
2
3 const logger = require("../utils/logger");
4
5 const dashboard = {
6   index(request, response) {
7     logger.info("dashboard rendering");
8     const viewData = {
9       title: "Template 1 Dashboard"
10    };
11    response.render("dashboard", viewData);
12  }
13 };
14
15 module.exports = dashboard;
16
```

Tools

New

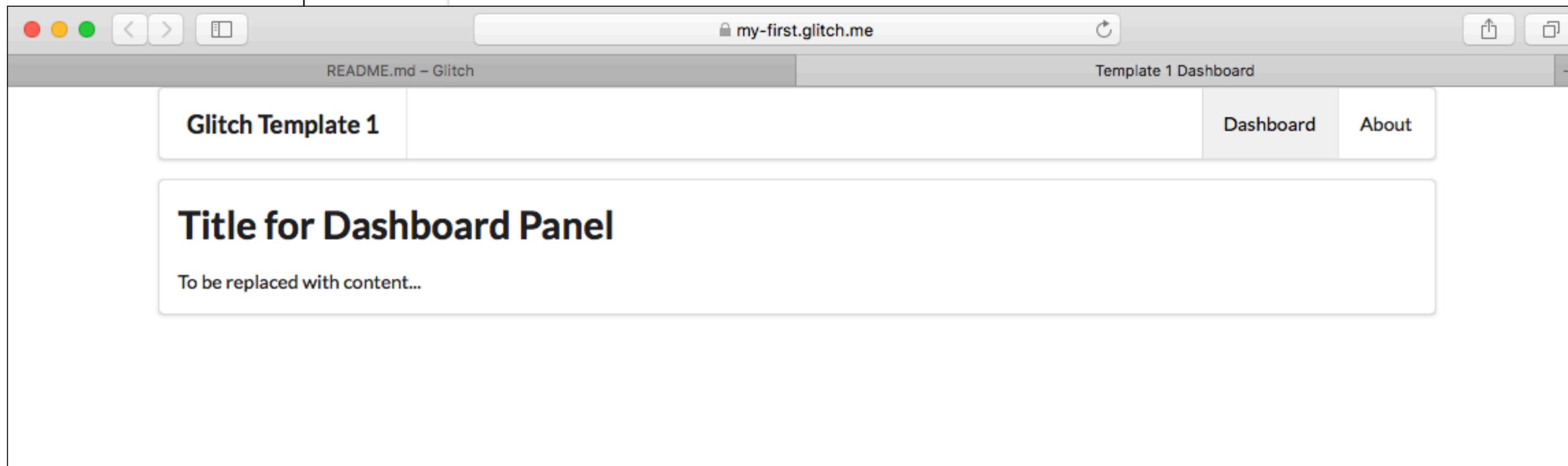
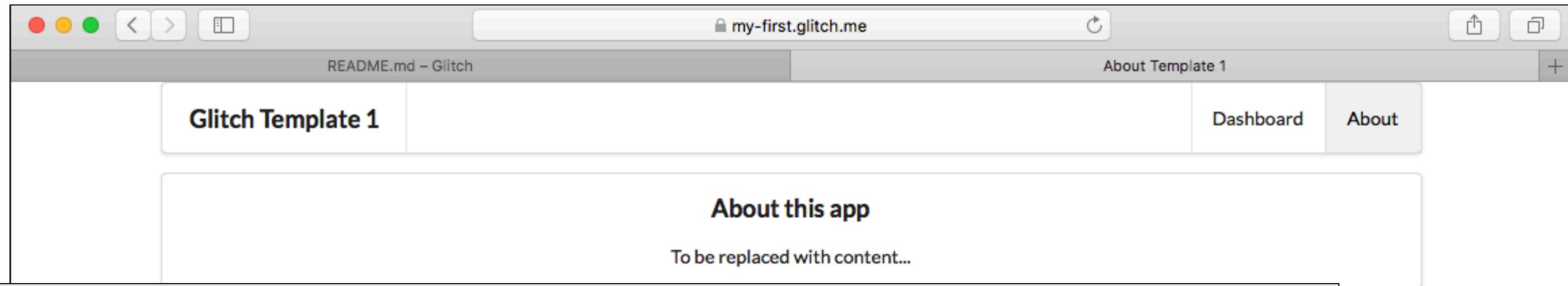
Logs Clear Debugger Console

```
+ lodash 4.17.11 8:25 AM
+ lowdb 1.0.0 8:25 AM
+ uuid 3.3.2 8:25 AM
+ winston 3.2.1 8:25 AM

Total install time: 6499ms 8:25 AM

a few seconds ago
{ message: 'glitch-template-1 started on port 3000',
  level: 'info',
  label: 'Playlist' } 8:25 AM
```

- The starter app has a menu + 2 'Views':
  - Dashboard
  - About



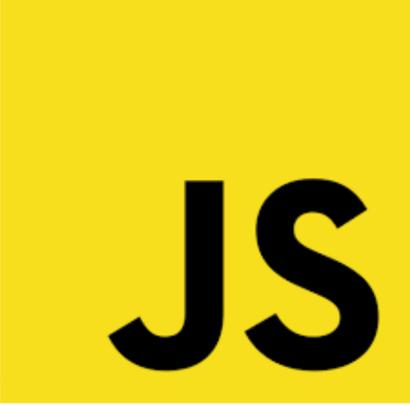
# Application Structure

---

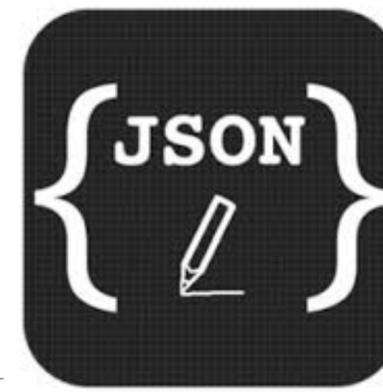
```
📁 assets
controllers/about.js
controllers/dashboard.js
utils/logger.js
views/layouts/main.hbs
views/partials/mainpanel.hbs
views/partials/menu.hbs
views/about.hbs
views/dashboard.hbs
.eslintrc.json
.gitignore
README.md
package.json
routes.js
server.js
```

- App implements Routes + Model/View/Controller Architecture
- These objects collaborate to support structured, predictable application workflow

## Back-end



JS



- All written in Javascript + JSON
- Consists of:
  - **Server** - main entry point
  - **Routes** - supported urls
  - **Controllers** - objects to handle the routes
  - **Config** - .gitignore, .eslintrc.js, package.json, readme.md
- Will include **Models** later...

```
controllers/about.js
controllers/dashboard.js
utils/logger.js
.eslintrc.json
.gitignore
README.md
package.json
routes.js
server.js
```

## Front-end

---



```
📁 assets  
views/layouts/main.hbs  
views/partials/mainpanel.hbs  
views/partials/menu.hbs  
views/about.hbs  
views/dashboard.hbs
```

- All written in HTML + handlebars
- Handlebars: Templating language
- Similar to EJS, it supports:
  - **Layouts**
  - **Partials**
  - **Views**
- These are very similar to EJS equivalents

# Back-end + Front-end

---

```
📁 assets
controllers/about.js
controllers/dashboard.js
utils/logger.js
views/layouts/main.hbs
views/partials/mainpanel.hbs
views/partials/menu.hbs
views/about.hbs
views/dashboard.hbs
.eslintrc.json
.gitignore
README.md
package.json
routes.js
server.js
```

- Building a new app:
- Extend, modify, replace and create new routes + controllers
- Rewrite handlebars templates to support new features, data and user experience.