Contents

[Model: 1](#_Toc42961806)

[View: 1](#_Toc42961807)

[Controller: 1](#_Toc42961808)

[Example: 2](#_Toc42961809)

[Explanation of the MVC data route: 2](#_Toc42961810)

[Pseudocode: 2](#_Toc42961811)

## Model:

Responsible for managing data

The brain of the application

Could be, noSQL, or MariaDB, or Mongo

Just the DB driver that needs to change

Could be a simple file (for example ,pull data from Json)

Communicate with the controller

The controller could updates de the view in some cases

## View:

View of the app

User interface, what the user sees when interact with the application (usually html with css and also dynamic values)

The controller communicates with the view

The template engine allows dynamic data (depends on what template engine), some example: haml, jinja or jinja2

**(template engine is really nice and interesting, there are many many)**

## Controller:

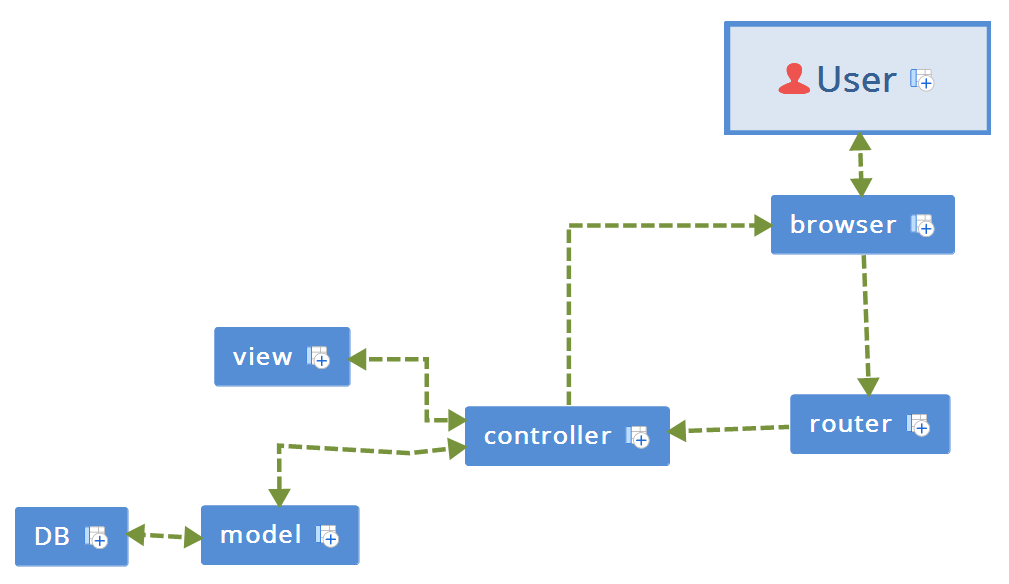
Middle between view and model

Takes user input, click in a link (get request), or submitting a form (post request), updating, from browser you can do get or post

Kind of a middle man, take data from db, the controller witll take that data and load the view and pass the data into it. From there, the template engine take control from it and do some logic some putput for the variables.

Can also Load view wo passing the data

## Example:



## Explanation of the MVC data route:

The user ssees the eweb on the browser, make a request, the route will call a controller method besed on the route (the requested could be some kind of link) if data is needed, the controller will interact with the model that interact with the database once the controller send the data to the view it will be handled by the template engine and then the controller with send the data to the browser for the user to see.

## Pseudocode:

<http://some-site-example.com/users/profile/1>

/**routes**  
 users/profile/:id = users.getProfile(id)

/**controllers**  
 class users{  
 function getProfile(id){  
 profile = this.UserModel.getProfile(id)  
  
 renderView(‘users/profile, profile’)

}

/**models**

Class usermodel{  
 function getProfile(id){  
 data = this.db.get(‘SELECT \* FROM users WHERE id = id’)  
 return data;

}

/**views**

/users

/profile

<h1>{{ profile.name }}</h1>  
 <ul>  
 <li>Email: {{ profile.email }}</li>  
 <li>Phone: {{ profile.phone }}</li>  
 </ul>