Playlist: <https://www.youtube.com/watch?v=XlvsJLer_No&list=PLZlA0Gpn_vH8jbFkBjOuFjhxANC63OmXM>

# Video 1: How the backend works

Front End: What we see on a web page

Back End: The parts of a website we don’t interact with

The server will receive a request from the client in the form of a URL.

The **protocol** is either HTTP or HTTPS, where HTTP is an unecrypted request and HTTPS is an encrypted one.

The **host** is the domain name for the server (such as gaminggalaxy.gg)

The **path** is where on the server the request is going to.

* For example, a request that goes to youtube.com/watch will go to the watch section of youtube’s server.

The query string is broken down to a set of keys and values and comes after the path, this is used to tell the server exactly what you want.

https://youtube.com/watch?v=[video id] contains the query string v=video id

In order for the server to know what exactly to do, you need to specify an action using REST.

# Video 2: What is rest?

REST stands for

RE – Representation

S – State

T – Transfer

It’s just a fancy way of saying that a server has the ability to respond to create, read, update and delete requests.

GET http://example.com/users Get the resource from the /users path

POST http://example.com/users Create a resource on the /users path

GET http://example.com/users/1 Get a resource from the /users/1 path

PUT http://example.com/users/1 Update a resource on the /users/1 path

DELETE http://example.com/users/1 Delete a resource on the /users/1 path

# Video 3: MVC Explained in 4 Minutes

MVC is ***Model, View, Controller***

Let’s say a user is requesting a particular page from a server.

The ***controller’s*** responsibility is to take that request and make sure it’s valid and that it goes to the right place.

The ***model’s*** responsibility is to take a request from the controller and getting the appropriate data from the database. This appropriate data is then sent back to the controller

The ***view’s*** responsibility is to take that appropriate data and display it to the user.

The model and the view ***never*** interact with one another. All traffic goes through the controller

# Video 4: Project Setup

# Video 5: Author’s Index/Create/New Routes

# Video 6: Books Index/Create/New Routes

# Video 7: File Upload Setup

# Video 8: Authors, Show/Edit/Update/Delete

# Video 9: Finish Backend

# Video 10: General CSS Styles

# Video 11: Form CSS Styles

# Video 12: Finishing Touches