Time	1 hour
Learning Goals	• Learn the differences between Rails and Sinatra API interaction

We have learnt what is HTTP, and how to build an API endpoint for it in Sinatra. But when we build the production app, we probably want to do it in Rails. So how do we go about doing that?

The Differences - Rails VS Sinatra

Routes

In Sinatra creating a route is as simple as

```
# creates a route for GET localhost:4567/users/1
get "/users/:id" do
end
```

In Rails, its a bit more complicated

```
# config/routes.rb

# creates a route for GET localhost:3000/users/1
resources :users, only: [:show]
# or the manual way
get '/users/:id', :to => 'users#show'

# In Rails, you need to specify the format for the route and which 'controller#action' to direct the request to
# e.g. get 'listings', :to => 'listings#index' will handle "localhost:3000" and direct it to 'ListingController', method "index"
```

Handling Header/Body

There's no difference! Both Rails and Sinatra are built on the same middleware - Rack, so both use the same way to handle headers and body. Heres an example:

Sinatra

```
post "/users" do
  # check request headers for "password" header
  p request.env["HTTP_PASSWORD"]
  # check request body for "password" params
  p params["password"]
  "Success"
end
```

Rails

```
class UserController < ActiveController:Base
  def index
    # check request headers for "password" header
    p request.env["HTTP_PASSWORD"]
    # check request body for "password" params
    p params["password"]
    render text: "Success"
  end
end
```

Rendering Responses

You might have noticed that the last line in each Sinatra/Rails controller is slightly different.

Both servers render the same response, the `code` is 200, the `body` is `"Success"`.

Up to now, you have been building websites, so the server usually renders an entire HTML document, containing layout, HTML, links to CSS and JS scripts.

In computer-computer communications, communications can be very concise since computers don't really care about graphics/alignment etc.

The standard format for API communications is JSON. This is because JSON is a limited format that is capable of expressing most of the basic programming data structures.

Example:

```
{
 "x": 123, # integers
 "y": "abc", # strings
 "z": [1,2,3], # arrays
 "a": {
   "b": true # booleans
 } # hashes
}
```

So how do we render JSON in Sinatra and Rails?

Sinatra

```
get "/root_path" do
 {a: 123, b: "xyz"}.to_json
end
```

Rails

```
class RootController do
 def index
```

```
  json_content = {a: 123, b: "xyz"}
  render json: json_content
 end
end
```

As you can see, Rails automatically converts the Ruby data to JSON format for you automatically!