University of Sheffield

COM1001 SPRING SEMESTER

Professor Phil McMinn

p.mcminn@sheffield.ac.uk

An Introduction to Sinatra

Live Demonstration:

Coding `hello_world`

Featuring:

- Review of setting up a Codio project
- Running the example with the `sinatra` command, which starts a **web server** that serves the pages of our web application
- Introducing a Sinatra route
- The base URL
- Viewing the HTTP Request and Response

Sinatra

A **Domain Specific Language (DSL)** for writing web applications in Ruby

It is relatively easy to learn

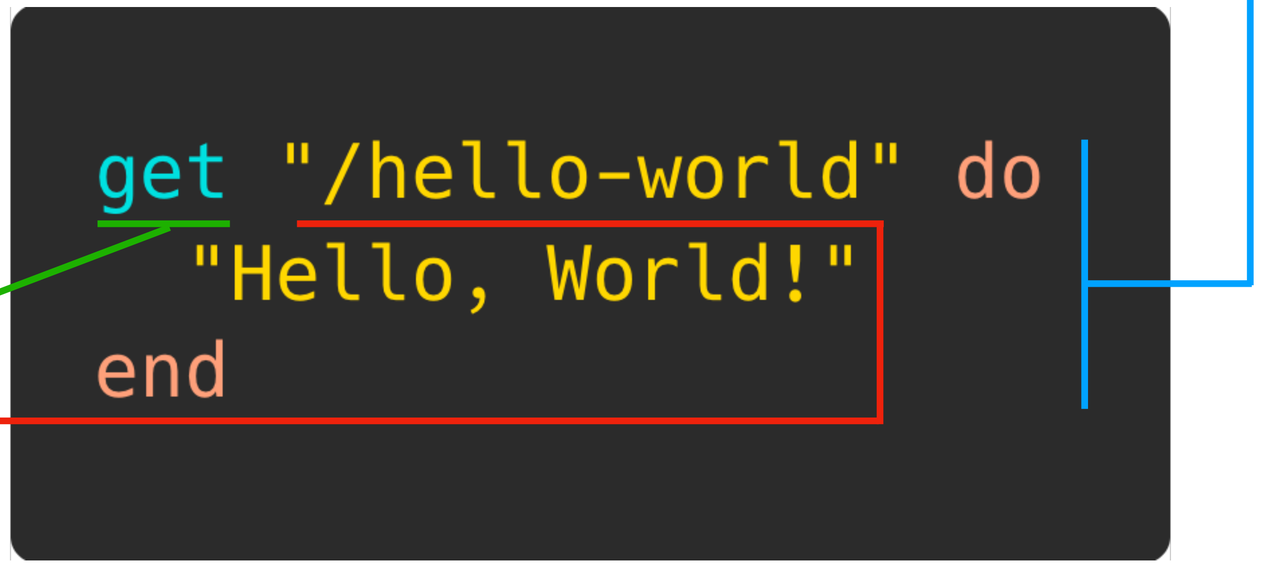
It is designed so that developers can write web applications quickly

It has been used commercially by several organisations from Apple, to LinkedIn, to GitHub, to the NSA

Website and documentation:
`http://sinatrarb.com`

Hello, World!

Sinatra involves writing **blocks** of Ruby code that determine which **HTTP Requests** the web application will respond to. These involve specifying a **HTTP method** and a **resource identifier**, which together are referred to as a **route**.



Ruby uses underscores as filename separators, but **resource identifiers** **should always use hyphens**, which might be confusing at first, but means that the resulting URLs have hyphens rather than underscores too. This is important because it's hard to distinguish underscores from spaces when a URL is underlined. Google recommends hyphens as separators in URLs for this reason. This is known as "**kebab-case**".

The Base URL

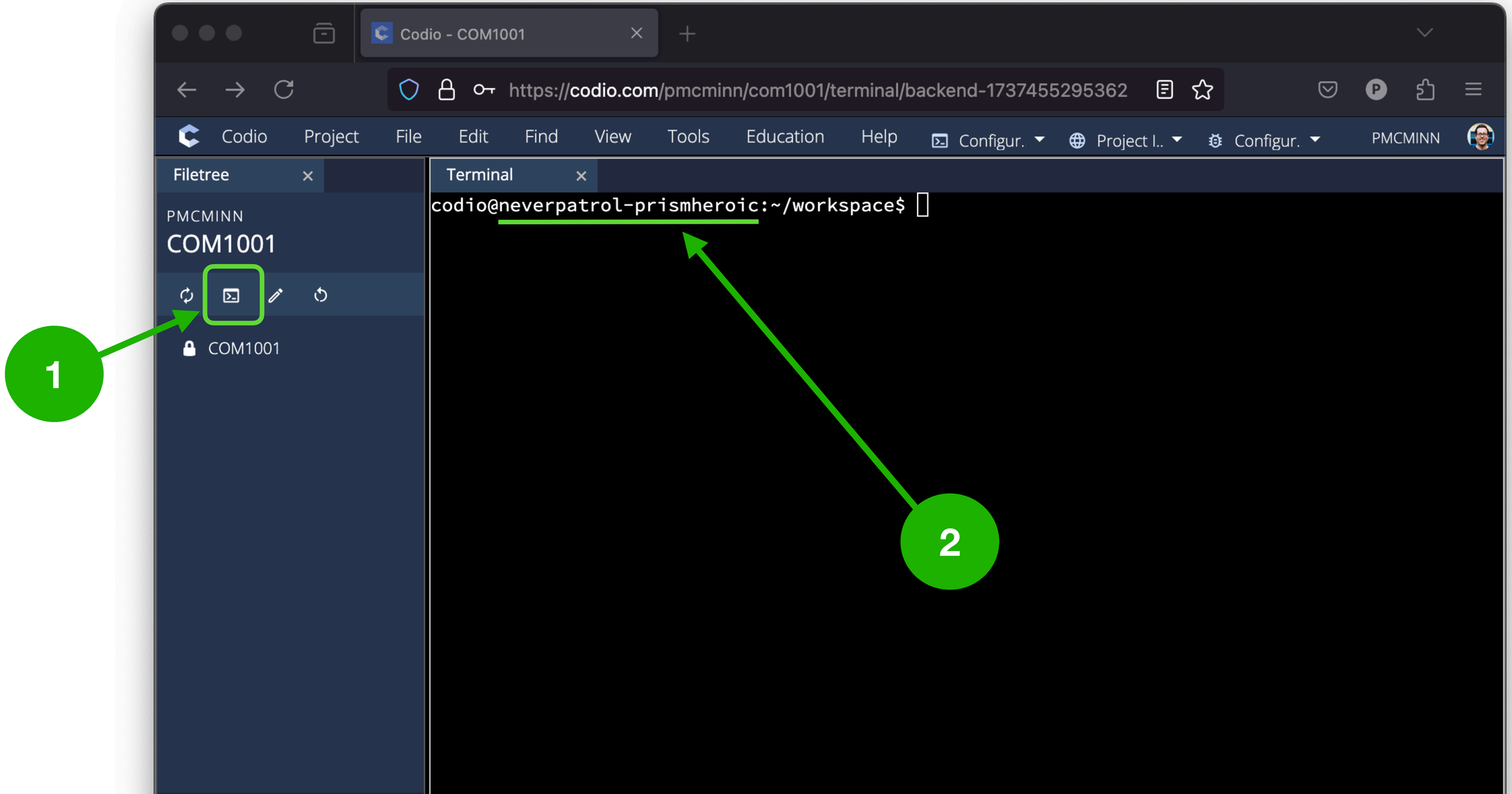
The **Base URL** forms the initial part of every URL of your web application.

It is made up of the **HTTP protocol** being used, the **domain name** of your web server, and the **port number** it is using for your web application.

For development on Codio, the port number is 4567, and the domain name is derived from the box name of your project, as follows:

```
https://box-name-4567.codio.io
```

Finding the Box Name of Your Project

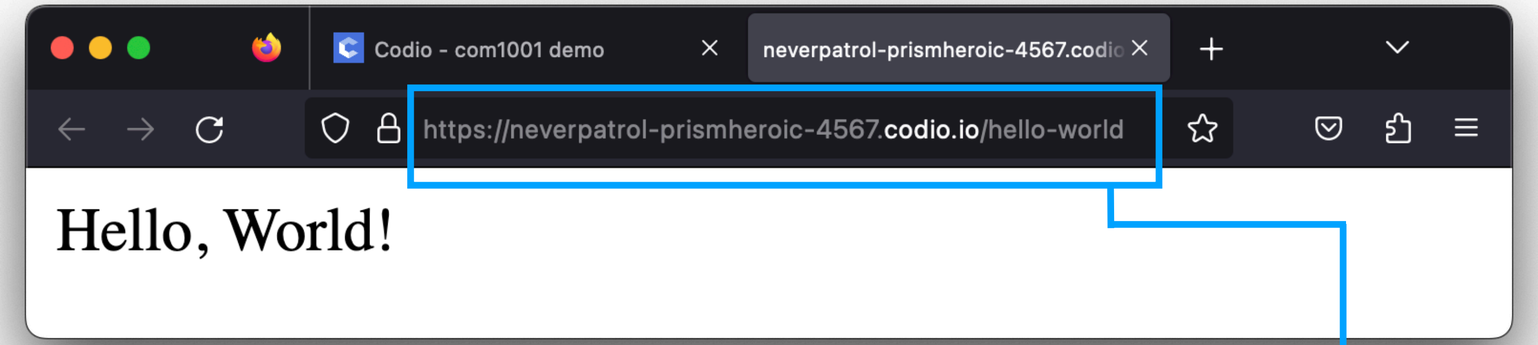


Viewing the Hello World Example

To invoke a Sinatra `get` route, we need to suffix the route's **resource identifier** to the **Base URL**, then enter this URL into the browser to display the resulting page.

So for the Hello World example, we need suffix `/hello-world`

```
get "/hello-world" do
  "Hello, World!"
end
```



The box name here is "`neverpatrol-prismheroic`", so my **base URL** is `https://neverpatrol-prismheroic-4567.codio.io`

and the complete URL I need is `https://neverpatrol-prismheroic-4567.codio.io/hello-world`

The HTTP Request from My Browser

```
GET /hello-world HTTP/2
Host: bordersmall-almondscarlet-4567.codio.io
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:134.0) Gecko/20100101 Firefox/134.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate, br, zstd
Referer: https://codio.com/
…
```

The HTTP Response from the Web Server

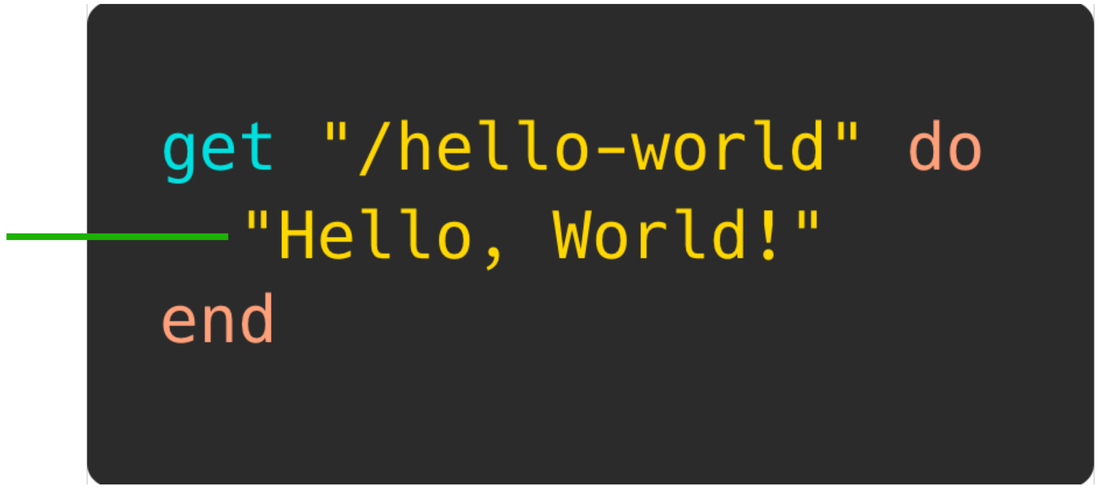
```
HTTP/2 200
content-type: text/html;charset=utf-8
content-length: 13
…
```

The HTTP response from a Sinatra Route

The body of the HTTP response is the return value of the block of the route that matches the HTTP request.

In Ruby, **we don't need to use the return keyword if it's the last line of a block**, it's just the value.

Hence in the Hello World example, the response of the HTTP body is "`Hello, World`".



Sinatra – Summary

- Sinatra is a domain specific language that helps map **HTTP requests** to **blocks of Ruby code** through so-called “**routes**”

- Sinatra generates a **HTTP response** according to instructions in the block of code corresponding to the route.

- To see Sinatra web pages, we need to use the correct **base URL** for our web server and use the **resource identifier** corresponding to the route.