My First Backend

Subject | 1 Solution | Additional Resources (1)

My First Backend

Technical details	
Submit file	app*
Languages	It needs to be completed in the language you are working on right now. If you are doing Bootcamp Javascript, then javascript (file extension will be .js). If you are doing Bootcamp Ruby, then Ruby (file extension will be .rb). It goes the same for Python, Java, C++, Rust, ...

Description

Control Center

roup formation

Progress

ubmitted

Peer eview

Go To DoCode

Access:
READ WRITE
Go To Gitea

Review PR

Give Up

inished

Create a backend app with light web framework (javascript you will use `express`, ruby you will use `sinatra`, python you will use `flask`)

You don't need to create a database, just store the information `hard coded` inside your file.

In order to find all the information needed, you will have to search online. Wikipedia, Google and Fan's website will have all the information you need for this project! :)

Part I

It will have a route `GET` on `/`. This action will give randomly (in a pool of at least 20) a name of a song from `Frank Sinatra`.
Wikipedia Page

Example00

```
$>curl -i http://web-
XXXXXXXXX.docode.YYYY.qwasar.io
HTTP/1.1 200 OK
Content-Type: text/html;charset=utf-8
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Content-Length: 19

New York, New York
$>
```

Example01

```
$>curl http://web-
XXXXXXXXX.docode.YYYY.qwasar.io
My Way
$>
```

Part II

Continue to build your backend `app.js`.
This time we will add multiple `pages` or `routes`:

Looking for a group

lucas_v

Also working on the project


abdullay_b


rasulov_d


suxrobov_s


risaliev_e


xujamura_d


to-xtasi_a


rahmono_az


isakov_a


asomov_u


axmadxo-_j


sodiqov_o


alpeisso_n


sultonov_x


nigmatu_mu


begimqul_g


xudaboye_s


norxojay_o


stupakov_k


sodikov_ar


sultanba_y

Just finished


parpiev_s


diyarova_s


muhiddin_s


uktamov_s


zharas_


orifjon


g-


isakov_

`GET` on `/`. This action will give randomly (in a pool of at least 20) a name of a song from `Frank Sinatra`. Wikipedia Page

`GET` on `/birth_date`. This action will give `Frank Sinatra` birth date.

`GET` on `/birth_city`. This action will give `Frank Sinatra` birth city.

`GET` on `/wives`. This action will give all the name of `Frank Sinatra` wife.

Format:
wife1, wife2, wife3, wife4

`GET` on `/picture`. This action will give `Frank Sinatra`'s picture.

(https://upload.wikimedia.org/wikipedia/commons/a/af/Frank_Sinatra_%2757.jpg)

Example00

```
$>curl -i http://web-
XXXXXXXXX.docode.YYYY.qwasar.io
HTTP/1.1 200 OK
Content-Type: text/html;charset=utf-8
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Content-Length: 19

New York, New York
$>
```

Example01

```
$>curl http://web-
XXXXXXXXX.docode.YYYY.qwasar.io
My Way
$>
```

Part III

g o_a ofurov_s a


abdurahi_a

yusupov_f

abduhaki_b

qayumov_s

radjapov_i

xudaboye_s

musayev_s

tsay_d

Type: Project

Group Size: 1 Participant

Review system: Peer Review

Difficulty: Initiation

Average duration: 1 Week

Project's Metadata

Project

id: 101

name: my-first-backend

visible: True

Continue to build your backend `app.js`, in this last part we will add two more routes:

> `GET` on `/public`. This action will print `"Everybody can see this page"`
>
> `GET` on `/protected`. This action will be protected by a HTTP Basic access authentication and print `"Welcome, authenticated client"` if you are authorized with the login `admin` and password `admin` otherwise it will provide a `401 Not authorized`.

Example00

```
$>curl -i http://web-
XXXXXXXXX.docode.YYYY.qwasar.io/protected
HTTP/1.1 401 Unauthorized
Content-Type: text/html;charset=utf-8
WWW-Authenticate: Basic realm="Restricted
Area"
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Content-Length: 15

Not authorized
$>
```

Example01

```
$>curl -i http://admin:admin@web-
XXXXXXXXX.docode.YYYY.qwasar.io/protected
HTTP/1.1 200 OK
Content-Type: text/html;charset=utf-8
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Content-Length: 29

Welcome, authenticated client
$>
```

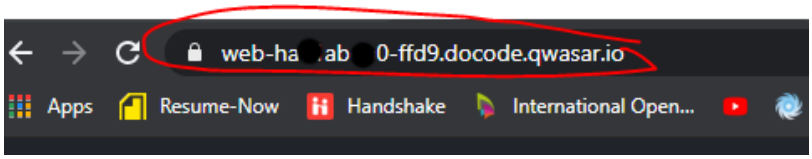
Example02

```
$>curl -i http://web-
XXXXXXXXX.docode.YYYY.qwasar.io/public
HTTP/1.1 200 OK
Content-Type: text/html;charset=utf-8
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Content-Length: 27

Everybody can see this page
$>
```

How can I access my docode server from the browser?

Your server will be accessible at this URL:
XXXXXXXXX is your docode ID
http://web-XXXXXXXXX.docode.YYYY.qwasar.io



Technical specifications

Your repository must be clean of any libraries. (by library we are refering to gems / node_modules).

How to prevent yourself to commit and push those extra files?

You have two ways:

> Easy, don't git add them.
>
> Even better, use a gitignore

You can also write a section inside your README.md on `how-to install your app`

Google Sinatra
Google HTTP Code (200, 204, 400, 401, 500)
Google HTTP Basic access authentication
Google curl (-i, -I, -X GET, etc)