

# CAB230 Assignment

SERVER SIDE REPORT

AUTHOR: ANYE ZHU (RENNY)

STUDENT NUMBER: N10322434

DATE OF SUBMISSION: 2<sup>ND</sup> JUNE

## Table of Contents

<b>1. Introduction.....</b>	<b>1</b>
1.1. Authentication routes .....	1
1.2. Search and Helpers routes .....	1
1.3. Security .....	1
1.4. Data connectivity .....	1
1.5. Swagger docs .....	1
1.6. Middleware.....	1
<b>2. Technical Description of the Application .....</b>	<b>1</b>
2.1. Authentication routes .....	1
2.2. Information routes.....	1
2.3. Application Middleware .....	1
2.4. Data connectivity .....	2
2.5. Swagger docs .....	2
<b>3. Security .....</b>	<b>2</b>
<b>4. Testing and limitations .....</b>	<b>3</b>
4.1. Register.....	3
4.2. Login .....	4
4.3. Offences.....	4
4.4. Search.....	5
4.5. Search (Filters) .....	5
<b>5. References.....</b>	<b>5</b>
<b>6. Appendix.....</b>	<b>6</b>
6.1. Installation guide .....	6
6.2. Swagger UI.....	7

## 1. Introduction

### 1.1. Authentication routes

**Register:** Register is implemented.

**Login:** Login is implemented.

### 1.2. Search and Helpers routes

**Search:** After authenticated with token, the search function could be used.

**Helpers:** Offences, areas, ages, genders and years could always be used.

### 1.3. Security

**Middleware security:** helmet is implemented.

**HTTPS:** HTTPS is used, which is much more secure than HTTP.

**JWT:** JWT is used for security.

### 1.4. Data connectivity

Use MySQL Workbench 8.0 CE to provide and save data.

### 1.5. Swagger docs

Swagger docs is implemented.

### 1.6. Middleware

Knex (<https://knexjs.org/>) is used, which is a well-established middleware.

## 2. Technical Description of the Application

### 2.1. Authentication routes

**Register:** The URL is <https://localhost/register>. When users register successfully, email and password will be saved into database.

**Login:** The URL is <https://localhost/login>. When users input the email and password, if the email and password have already existed in database, then users could get the token, and could use the following function, search.

### 2.2. Information routes

**Search:** The URL is <https://localhost/search>. User needs to login firstly and then gets the token. Then user can use search function in the client side. When the user enters some keywords and click the search button, the data would be selected from database. Next, the client-side responses the results.

**Helpers:** The URLs are <https://localhost/offences>, <https://localhost/areas>, <https://localhost/ages>, <https://localhost/genders>, <https://localhost/years>. These information routes are to select data from database and return the results.

### 2.3. Application Middleware

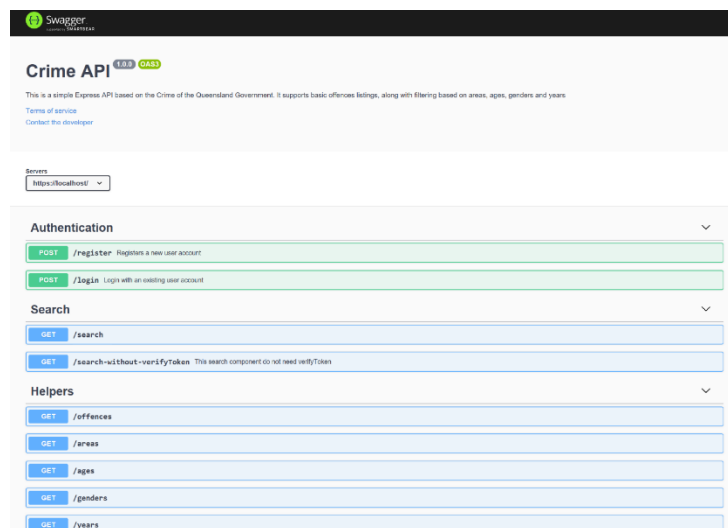
Knex and Helmet. Actually, I already used 'helmet' before demoing on Friday, but I forgot to show the tutor the version with 'helmet', which made me lose the score for this part. I hope I can get the score of this part.

## 2.4. Data connectivity

Use MySQL Workbench 8.0 CE to provide and save data.

## 2.5. Swagger docs

The URL is <https://localhost/docs/>. The user can see the APIs in this website. This is swagger UI (<https://swagger.io/tools/swagger-ui/>), it is for visualizing and interacting with the API's resources.



# 3. Security

**Middleware security:** Knex, Helmet.

**Https:** HTTPS is used for secure communication. It is the standard security technology that establishes an encrypted connection between a web server and a browser.

**JWT:** JWT is used for security, because only the server should know the "secret" that is used to generate the JWT

## 4. Testing and limitations

Test/results example in the table

Action Test/results	4.1. Register	4.2. Login	4.3. offences	4.4. Search	4.5. Search (Filters)
Test example	email: demo01- user@gmail .com password: userpwd12 3	email: demo01- user@gmail .com password: userpwd12 3	Click on the offences button	offence: Assault	<div>offence: Assault</div> <div>area: Brisbane City Council</div> <div>age: Adult</div> <div>gender: Female</div> <div>year: 2001</div>
Results - Client side	Register successfully	Login successfully	Show the offences table	Show the table with the search result	Show the table with the search result
Results - Server side	Email and password are created (201)	Login is OK (200)	Offence is OK (304)	Search is OK (200)	Search with these filters is OK (200)
Results - Database	Email and password are saved to database	Check the same data in the database as the entered email and password	Get offence data from database	Get the search result from database	Get the search results from database

### 4.1. Register

Client side:



## Server side:

```
POST /register 201 15.364 ms - {"x-dns-prefetch-control":"off","x-frame-options":"SAMEORIGIN","strict-transport-security":{"max-age=15552000; includeSubDomains"},"x-download-options":"noopen","x-content-type-options":"nosniff","x-xss-protection":"1; mode=block","access-control-allow-origin":"*","content-type":"application/json; charset=utf-8","content-length":"70","etag":"W/\"46-2kzoy+9xUSuF8K/71GGISxEdhFk\""}
```

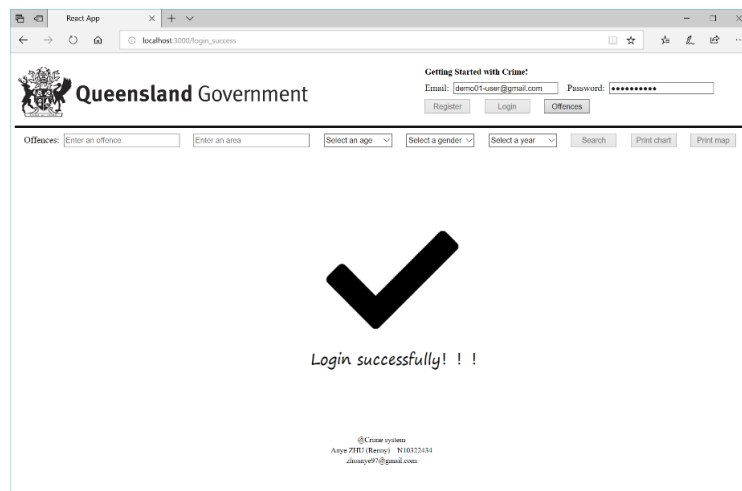
## Database:

demo01-user@gmail.com	userpwd123	2019-06-01 01:21:07	2019-06-01 01:21:07
-----------------------	------------	---------------------	---------------------

So, the /register part can run successfully on both the client side and the server side.

## 4.2. Login

### Client side:



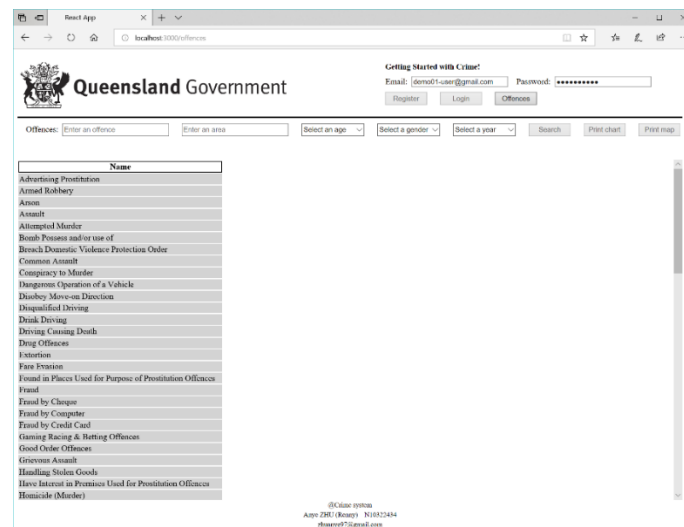
### Server side:

```
POST /login 200 2.512 ms - {"x-dns-prefetch-control":"off","x-frame-options":"SAMEORIGIN","strict-transport-security":{"max-age=15552000; includeSubDomains"},"x-download-options":"noopen","x-content-type-options":"nosniff","x-xss-protection":"1; mode=block","access-control-allow-origin":"*","content-type":"application/json; charset=utf-8","content-length":"215","etag":"W/\"d7-d022aPgQxR3nIAEP02jMrHc03kY\""}
```

As can be seen from the pictures above, the /login part can work successfully as well.

## 4.3. Offences

### Client side:



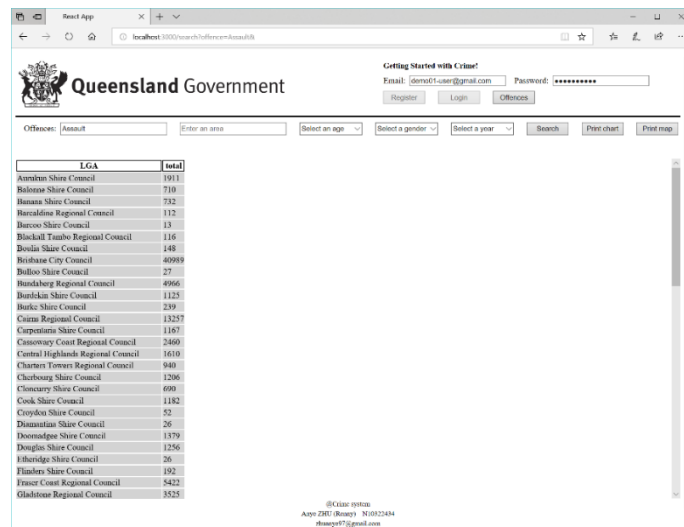
### Server sides:

```
GET /offences 304 5.107 ms - {"x-dns-prefetch-control":"off","x-frame-options":"SAMEORIGIN","strict-transport-security":{"max-age=15552000; includeSubDomains"},"x-download-options":"noopen","x-content-type-options":"nosniff","x-xss-protection":"1; mode=block","access-control-allow-origin":"*","etag":"W/\"960-04zLdTMUyLVNb4Bvz7na0H7ubZs\""}
```

The code returned by the server is 304, which is different from the code that needs to be returned (returns 200). So, although this part work successfully, it is not perfect.

#### 4.4. Search

Client side:



Server side:

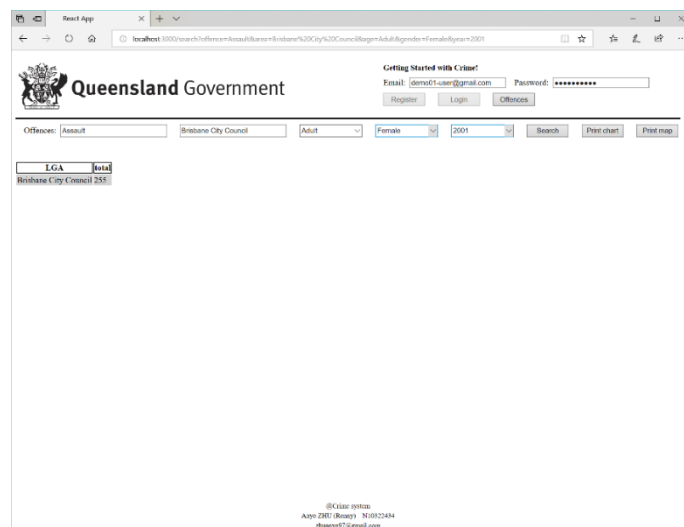
```
GET /search?offence=Assault& 200 254.030 ms - {"x-dns-prefetch-control":"off","x-frame-options":"SAMEORIGIN","strict-transport-security":"max-age=15552000; includeSubDomains","x-download-options":"noopen","x-content-type-options":"nosniff","x-xss-protection":"1; mode=block","access-control-allow-origin":"*","content-type":"application/json; charset=utf-8","content-length":"3745","etag":"W/\"ea1-mQ95sH9m5WyJt5doJGom0jSGg5M\\\""}

```

The returned code in the search section meets the requirement (return 200). And this part could run successfully.

#### 4.5. Search (Filters)

Client side:



Server side:

```
GET /search?offence=Assault&area=Brisbane%20City%20Council&age=Adult&gender=Female&year=2001 200 198.416 ms - {"x-dns-prefetch-control":"off","x-frame-options":"SAMEORIGIN","strict-transport-security":"max-age=15552000; includeSubDomains","x-download-options":"noopen","x-content-type-options":"nosniff","x-xss-protection":"1; mode=block","access-control-allow-origin":"*","content-type":"application/json; charset=utf-8","content-length":"56","etag":"W/\"38-ZGRxZFeHJ8kyVEhx3PFwLwDtLE\\\""}

```

The returned code meets the requirement (return 200) when the search section runs with filters. This part can work successfully.

## 5. References

Swagger UI, 2019. Retrieved from <https://swagger.io/tools/swagger-ui/>

KNEX.JS, 2019. Retrieved from <https://knexjs.org/>

Helmet, 2019. Retrieved from <https://helmetjs.github.io/>

Jsonwebtoken, 2019. Retrieved from <https://www.npmjs.com/package/jsonwebtoken>

Download MySQL Installer, 2019. Retrieved from <https://dev.mysql.com/downloads/windows/installer/8.0.html>

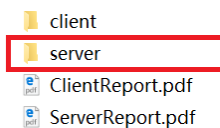
What's the Difference between HTTP and HTTPS?, 2019. Retrieved from <https://www.globalsign.com/en/blog/the-difference-between-http-and-https/>

How safe is JWT?, 2019. Retrieved from <https://stackoverflow.com/questions/35817325/how-safe-is-jwt>

## 6. Appendix

### 6.1. Installation guide

Step1: Open **server** file with VS Code.



Step2: Execute **npm install** in the terminal.

```
PS C:\Users\Zhuan\Desktop\CAB230 Assignment\Server_Side Code> npm i

> core-js@2.6.9 postinstall C:\Users\Zhuan\Desktop\CAB230 Assignment\Server_Side Code\node_modules\core-js
> node scripts/postinstall || echo "ignore"

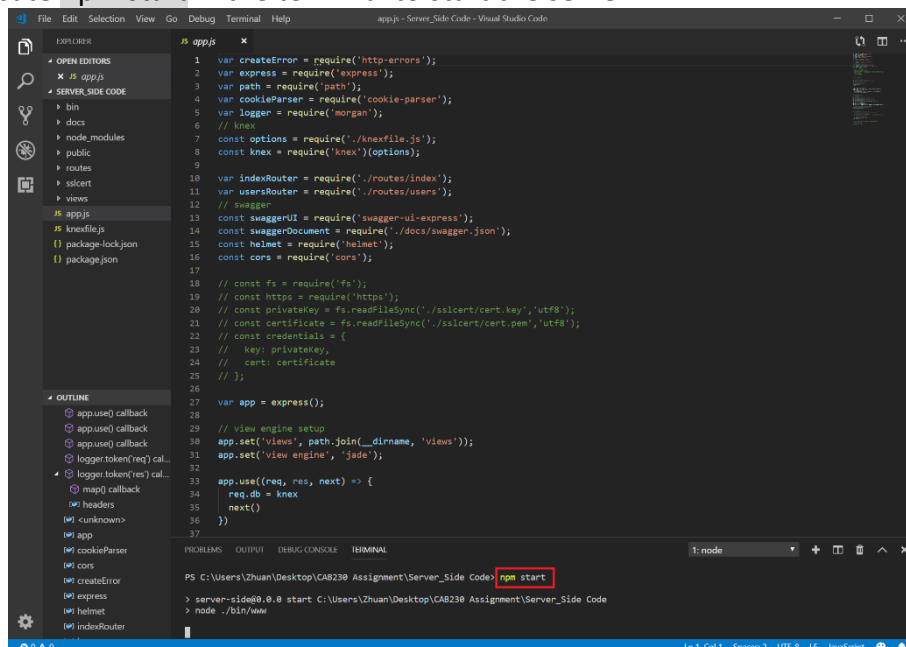
Thank you for using core-js ( https://github.com/zloirock/core-js ) for polyfilling JavaScript standard library!

The project needs your help! Please consider supporting of core-js on Open Collective or Patreon:
> https://opencollective.com/core-js
> https://www.patreon.com/zloirock

Also, the author of core-js ( https://github.com/zloirock ) is looking for a good job -)

added 592 packages from 425 contributors and audited 2691 packages in 12.45s
found 5 vulnerabilities (4 low, 1 moderate)
run 'npm audit fix' to fix them, or 'npm audit' for details
PS C:\Users\Zhuan\Desktop\CAB230 Assignment\Server_Side Code>
```

Step3: Execute **npm start** in the terminal to start the server.



Then, the server starts up.



## 6.2. Swagger UI (<https://localhost/docs/>)

Register: <https://localhost/register>

Login: <https://localhost/login>

POST

register

Register a new user

For register

Parameters

No parameters

Request body

application/json

Raw

JSON

XML

YAML

GraphQL

Form

JSON Schema

OpenAPI

```
{
  "email": "test-user@demo.com",
  "password": "1234567890123456"
}
```

Control

Reset

Example

Clear

Responses

201

Raw

JSON

XML

YAML

GraphQL

Form

JSON Schema

OpenAPI

```
{
  "email": "test-user@demo.com",
  "password": "1234567890123456"
}
```

Control

Reset

Example

Clear

Raw

JSON

XML

YAML

GraphQL

Form

JSON Schema

OpenAPI

```
{
  "email": "test-user@demo.com",
  "password": "1234567890123456"
}
```

Control

Reset

Example

Clear

[illegible]

Search: <https://localhost/search>. This part needs the token, so I would test this function with the other route <https://localhost/search-without-verifyToken>

[illegible]

Offences: <https://localhost/offences>

[illegible]

Areas: <https://localhost/areas>

GET /forms

List of Actions to filter search results by

Parameters

No parameters

Clear

Example

Clear

Responses

200

Curl

curl -X GET "https://localhost:3000/" -H "accept: \*/\*"

Response (200)

https://localhost:3000/

Server response

Code

Details

200

Response body

```

{
  "actions": [
    {
      "name": "New User",
      "url": "/new-user",
      "method": "POST",
      "description": "Create a new user",
      "status": "Active",
      "created_at": "2023-01-01T00:00:00Z",
      "updated_at": "2023-01-01T00:00:00Z",
      "deleted_at": null,
      "is_public": true,
      "is_featured": false,
      "is_archived": false,
      "is_deleted": false,
      "is_hidden": false,
      "is_locked": false,
      "is_readonly": false,
      "is_test": false,
      "is_demo": false,
      "is_preview": false,
      "is_simulation": false,
      "is_sandbox": false,
      "is_staging": false,
      "is_production": false,
      "is_development": false,
      "is_testing": false,
      "is_validation": false,
      "is_verification": false,
      "is_authentication": false,
      "is_authorization": false,
      "is_encryption": false,
      "is_decryption": false,
      "is_compression": false,
      "is_decompression": false,
      "is_caching": false,
      "is_uncaching": false,
      "is_session": false,
      "is_cookie": false,
      "is_token": false,
      "is_jwt": false,
      "is_oauth": false,
      "is_openid": false,
      "is_saml": false,
      "is_mfa": false,
      "is_totp": false,
      "is_otp": false,
      "is_2fa": false,
      "is_3fa": false,
      "is_4fa": false,
      "is_5fa": false,
      "is_6fa": false,
      "is_7fa": false,
      "is_8fa": false,
      "is_9fa": false,
      "is_10fa": false,
      "is_11fa": false,
      "is_12fa": false,
      "is_13fa": false,
      "is_14fa": false,
      "is_15fa": false,
      "is_16fa": false,
      "is_17fa": false,
      "is_18fa": false,
      "is_19fa": false,
      "is_20fa": false,
      "is_21fa": false,
      "is_22fa": false,
      "is_23fa": false,
      "is_24fa": false,
      "is_25fa": false,
      "is_26fa": false,
      "is_27fa": false,
      "is_28fa": false,
      "is_29fa": false,
      "is_30fa": false,
      "is_31fa": false,
      "is_32fa": false,
      "is_33fa": false,
      "is_34fa": false,
      "is_35fa": false,
      "is_36fa": false,
      "is_37fa": false,
      "is_38fa": false,
      "is_39fa": false,
      "is_40fa": false,
      "is_41fa": false,
      "is_42fa": false,
      "is_43fa": false,
      "is_44fa": false,
      "is_45fa": false,
      "is_46fa": false,
      "is_47fa": false,
      "is_48fa": false,
      "is_49fa": false,
      "is_50fa": false,
      "is_51fa": false,
      "is_52fa": false,
      "is_53fa": false,
      "is_54fa": false,
      "is_55fa": false,
      "is_56fa": false,
      "is_57fa": false,
      "is_58fa": false,
      "is_59fa": false,
      "is_60fa": false,
      "is_61fa": false,
      "is_62fa": false,
      "is_63fa": false,
      "is_64fa": false,
      "is_65fa": false,
      "is_66fa": false,
      "is_67fa": false,
      "is_68fa": false,
      "is_69fa": false,
      "is_70fa": false,
      "is_71fa": false,
      "is_72fa": false,
      "is_73fa": false,
      "is_74fa": false,
      "is_75fa": false,
      "is_76fa": false,
      "is_77fa": false,
      "is_78fa": false,
      "is_79fa": false,
      "is_80fa": false,
      "is_81fa": false,
      "is_82fa": false,
      "is_83fa": false,
      "is_84fa": false,
      "is_85fa": false,
      "is_86fa": false,
      "is_87fa": false,
      "is_88fa": false,
      "is_89fa": false,
      "is_90fa": false,
      "is_91fa": false,
      "is_92fa": false,
      "is_93fa": false,
      "is_94fa": false,
      "is_95fa": false,
      "is_96fa": false,
      "is_97fa": false,
      "is_98fa": false,
      "is_99fa": false,
      "is_100fa": false,
      "is_101fa": false,
      "is_102fa": false,
      "is_103fa": false,
      "is_104fa": false,
      "is_105fa": false,
      "is_106fa": false,
      "is_107fa": false,
      "is_108fa": false,
      "is_109fa": false,
      "is_110fa": false,
      "is_111fa": false,
      "is_112fa": false,
      "is_113fa": false,
      "is_114fa": false,
      "is_115fa": false,
      "is_116fa": false,
      "is_117fa": false,
      "is_118fa": false,
      "is_119fa": false,
      "is_120fa": false,
      "is_121fa": false,
      "is_122fa": false,
      "is_123fa": false,
      "is_124fa": false,
      "is_125fa": false,
      "is_126fa": false,
      "is_127fa": false,
      "is_128fa": false,
      "is_129fa": false,
      "is_130fa": false,
      "is_131fa": false,
      "is_132fa": false,
      "is_133fa": false,
      "is_134fa": false,
      "is_135fa": false,
      "is_136fa": false,
      "is_137fa": false,
      "is_138fa": false,
      "is_139fa": false,
      "is_140fa": false,
      "is_141fa": false,
      "is_142fa": false,
      "is_143fa": false,
      "is_144fa": false,
      "is_145fa": false,
      "is_146fa": false,
      "is_147fa": false,
      "is_148fa": false,
      "is_149fa": false,
      "is_150fa": false,
      "is_151fa": false,
      "is_152fa": false,
      "is_153fa": false,
      "is_154fa": false,
      "is_155fa": false,
      "is_156fa": false,
      "is_157fa": false,
      "is_158fa": false,
      "is_159fa": false,
      "is_160fa": false,
      "is_161fa": false,
      "is_162fa": false,
      "is_163fa": false,
      "is_164fa": false,
      "is_165fa": false,
      "is_166fa": false,
      "is_167fa": false,
      "is_168fa": false,
      "is_169fa": false,
      "is_170fa": false,
      "is_171fa": false,
      "is_172fa": false,
      "is_173fa": false,
      "is_174fa": false,
      "is_175fa": false,
      "is_176fa": false,
      "is_177fa": false,
      "is_178fa": false,
      "is_179fa": false,
      "is_180fa": false,
      "is_181fa": false,
      "is_182fa": false,
      "is_183fa": false,
      "is_184fa": false,
      "is_185fa": false,
      "is_186fa": false,
      "is_187fa": false,
      "is_188fa": false,
      "is_189fa": false,
      "is_190fa": false,
      "is_191fa": false,
      "is_192fa": false,
      "is_193fa": false,
      "is_194fa": false,
      "is_195fa": false,
      "is_196fa": false,
      "is_197fa": false,
      "is_198fa": false,
      "is_199fa": false,
      "is_200fa": false,
      "is_201fa": false,
      "is_202fa": false,
      "is_203fa": false,
      "is_204fa": false,
      "is_205fa": false,
      "is_206fa": false,
      "is_207fa": false,
      "is_208fa": false,
      "is_209fa": false,
      "is_210fa": false,
      "is_211fa": false,
      "is_212fa": false,
      "is_213fa": false,
      "is_214fa": false,
      "is_215fa": false,
      "is_216fa": false,
      "is_217fa": false,
      "is_218fa": false,
      "is_219fa": false,
      "is_220fa": false,
      "is_221fa": false,
      "is_222fa": false,
      "is_223fa": false,
      "is_224fa": false,
      "is_225fa": false,
      "is_226fa": false,
      "is_227fa": false,
      "is_228fa": false,
      "is_229fa": false,
      "is_230fa": false,
      "is_231fa": false,
      "is_232fa": false,
      "is_233fa": false,
      "is_234fa": false,
      "is_235fa": false,
      "is_236fa": false,
      "is_237fa": false,
      "is_238fa": false,
      "is_239fa": false,
      "is_240fa": false,
      "is_241fa": false,
      "is_242fa": false,
      "is_243fa": false,
      "is_244fa": false,
      "is_245fa": false,
      "is_246fa": false,
      "is_247fa": false,
      "is_248fa": false,
      "is_249fa": false,
      "is_250fa": false,
      "is_251fa": false,
      "is_252fa": false,
      "is_253fa": false,
      "is_254fa": false,
      "is_255fa": false,
      "is_256fa": false,
      "is_257fa": false,
      "is_258fa": false,
      "is_259fa": false,
      "is_260fa": false,
      "is_261fa": false,
      "is_262fa": false,
      "is_263fa": false,
      "is_264fa": false,
      "is_265fa": false,
      "is_266fa": false,
      "is_267fa": false,
      "is_268fa": false,
      "is_269fa": false,
      "is_270fa": false,
      "is_271fa": false,
      "is_272fa": false,
      "is_273fa": false,
      "is_274fa": false,
      "is_275fa": false,
      "is_276fa": false,
      "is_277fa": false,
      "is_278fa": false,
      "is_279fa": false,
      "is_280fa": false,
      "is_281fa": false,
      "is_282fa": false,
      "is_283fa": false,
      "is_284fa": false,
      "is_285fa": false,
      "is_286fa": false,
      "is_287fa": false,
      "is_288fa": false,
      "is_289fa": false,
      "is_290fa": false,
      "is_291fa": false,
      "is_292fa": false,
      "is_293fa": false,
      "is_294fa": false,
      "is_295fa": false,
      "
```

Years: <https://localhost/years>

GET /jmeter

List of items to filter search results by

Parameters

No parameters

Execute

Clear

Responses

curl -k -H "https://localhost/jmeter" -H "https://1"

Response info

https://localhost/jmeter

Server response

Code

200

Response body

```

{
  "name": "jmeter",
  "url": "https://localhost/jmeter",
  "description": "jmeter",
  "author": "jmeter",
  "version": "1.0",
  "license": "MIT",
  "keywords": "jmeter",
  "dependencies": {
    "express": "4.18.2",
    "body-parser": "1.20.1",
    "morgan": "1.10.0",
    "helmet": "5.1.1",
    "cors": "2.8.5",
    "dotenv": "16.0.3",
    "compression": "1.7.4",
    "cookie-parser": "1.4.6",
    "socket.io": "4.7.4"
  },
  "scripts": {
    "start": "node index.js"
  },
  "main": "index.js",
  "engines": {
    "node": "18.18.0"
  }
}
```

Response headers

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 312
Date: Wed, 14 Nov 2024 14:13:10 GMT
Access-Control-Allow-Origin: https://localhost
Access-Control-Allow-Headers: Content-Type, X-Requested-With, Accept
Access-Control-Allow-Methods: GET, POST, PUT, DELETE, OPTIONS
Server: Express/4.18.2
X-Content-Type-Options: nosniff
X-Frame-Options: DENY
X-XSS-Protection: 1; mode=block
X-WebKit-CSP: default-src: 'self'; script-src: 'self'; style-src: 'self'; font-src: 'self'; image-src: 'self'; connect-src: 'self';

```

Responses

Code

200

Description

No info

Ages: <https://localhost/ages>

[illegible]

Genders: <https://localhost/genders>

List of Domains for Your search results by

Parameters

No parameters

Cancel

Exclude

Clear

Responses

Get
 

```
GET / 407 "https://localhost/gendres" - "image,txt"
```

Request URL
 

```
https://localhost/gendres
```

Status response
 Details

Code

200

Request body

```

{
  "image": " ",
  "image": " ",
  "image": " ",
  "image": " "
}

```

Download

Request headers

```

Host: localhost:8080
Connection: keep-alive
Content-Type: application/json
Accept: */*
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/42.0.2311.90 Safari/537.36

```

Responses

Code

200

Description

Links