

Github_Scanner Backend API

1. POST: Add

Add solutions' github repos for every launched hackthon:

URL: http://18.217.114.101:8080/github_scanner/api/add

Input format:

- Headers:
"Content-Type": "application/json"
- Body:
A hackthon_name, with an array of String which contains all solutions' github repo links. In JSON format.

Example:

```
{
  "hackthon_name": "Hackthon1",
  "gitRepos": ["https://github.com/chenwang0227/Hackthon1"]
}
```

Key	Data type	Value	Data type
"hackthon_name"	String	Hackthon's name	String
"gitRepos"	String	All solutions' github repo links	An array of String

Response : 200"OK"

The screenshot displays a REST client interface with the following details:

- Method:** POST
- URL:** http://18.217.114.101:8080/github_scanner/api/add
- Body:** A JSON object:

```
{
  "hackthon_name": "Hackthon1",
  "gitRepos": ["https://github.com/chenwang0227/Hackthon1"]
}
```
- Status:** 200 OK
- Time:** 560 ms
- Size:** 134 B
- Response Body:** "OK"

2. POST: Search

Search specific hackthon projects based on certain criteria.

URL: http://18.217.114.101:8080/github_scanner/api/search

Input format:

- Headers:
[{"key": "Content-Type", "name": "Content-Type", "value": "application/json", "description": "", "type": "text"}]
- Body:

Example: these parameters can be null

```
{
  "hackthon_name": List<String>,
  "project_name": String,
  "problem": String,
  "industry": String,
  "technology": String,
  "email": String,
  "header": String,
  "data": String
}
```

Key	Data type	Value	Data type
"hackthon_name"	String	Specify the search range	List<String>
"project_name"	String	A specific project name you want to search	String
"problem"	String	A specific problem name you want to search	String
"industry"	String	A specific industry name you want to search	String
"technology"	String	A specific technology name you want to search	String
"email"	String	A specific user you want to search	String
"header"	String	Any other header you want to search	String
"data"	String	Any other keyword under that header you want to search	String

Example:

Input:

```
{
  "hackthon_name": ["Hackthon1", "Hackthon2"],
  "project_name": null,
  "problem": "Provenance",
  "industry": null,
  "technology": null,
  "email": null,
  "header": "Descriptions",
  "data": "Algo"
}
```

Output:

[

```

{
  "_id": {
    "timestamp": 1573434205,
    "machineIdentifier": 5113427,
    "processIdentifier": 5502,
    "counter": 3491831,
    "time": 1573434205000,
    "date": "2019-11-11T01:03:25.000+0000",
    "timeSecond": 1573434205
  },
  "hackthon_name": "Hackthon1",
  "project_name": "AlgoMed",
  "problem": [
    "Provenance"
  ],
  "industry": [
    "Health Care"
  ],
  "technology": [
    "Progressive Web App",
    "Algorand SDK"
  ],
  "email": [
    "jhons@andrew.cmu.edu",
    "ellenw@andrew.cmu.edu"
  ],
  "Descriptions": "The AlgoMed - Computerized Pressure Algometer is the first software-based computerized Algometer to offer real-time visual & auditory feedback to control & monitor applied pressure rates.<br/>Algometers are designed to quantify and document levels of tenderness via pressure threshold measurement and pain sensitivity via pain tolerance measurement. Pressure algometry is a reliable measure of pain in muscle, joints, tendons, and ligaments.<br/>The challenge in performing algometry tests is to apply continuous pressure at a constant rate on the patient selected body site without having the possibility to monitor it in real time.",
  "Achievements": "Finally build a solution to solve.."
}
]

```

▶ http://localhost:8080/github_scanner/api/search

Examples (0) ▼

POST

http://18.217.114.101:8080/github_scanner/api/search

Send

Save

Params

Authorization

Headers (1)

Body

Pre-request Script

Tests

Cookies

Code

Comments (0)

● none

● form-data

● x-www-form-urlencoded

● raw

● binary

JSON (application/json) ▼

Beautify

```
1 {
2   "hackthon_name": ["Hackthon1", "Hackthon2"],
3   "project_name": null,
4   "problem": "Provenance",
5   "industry": null,
6   "technology": null,
7   "email": "ellenw@andrew.cmu.edu",
8   "header": "Descriptions",
9   "data": "Algo"
10 }
```

Body

Cookies

Headers (3)

Test Results

Status: 200 OK

Time: 372 ms

Size: 1.21 KB

Save

Download

Pretty

Raw

Preview

JSON ▼

```
1 {
2   "_id": {
3     "timestamp": 1573685268,
4     "machineIdentifier": 5113427,
5     "processIdentifier": 9138,
6     "counter": 14613903,
7     "time": 1573685268000,
8     "date": "2019-11-13T22:47:48.000+0000",
9     "timeSecond": 1573685268
10  },
11   "hackthon_name": "Hackthon1",
12   "project_name": "AlgoMed",
13   "problem": [
14     "Provenance"
15  ],
16   "industry": [
17     "Health Care"
18  ],
19   "technology": [
20     "Progressive Web App",
21     "Algorand SDK"
22  ],
23   "email": [
24     "jhons@andrew.cmu.edu",
25     "ellenw@andrew.cmu.edu"
26  ],
27   "Descriptions": "The AlgoMed – Computerized Pressure Algometer is the first software-based computerized Algometer to offer real-time visual & auditory feedback to control & monitor applied pressure rates.<br/>Algometers are designed to quantify and document levels of tenderness via pressure threshold measurement and pain sensitivity via pain tolerance measurement. Pressure algometry is a reliable measure of pain in muscle, joints, tendons, and ligaments.<br/>The challenge in performing algometry tests is to apply continuous pressure at a constant rate on the patient selected body site without having the possibility to monitor it in real time."
28   ,
29   "Achievements": "Finally build a solution to solve.."
30 }
31 }
```

3. GET: All

Get all launched hackthons' names that are already stored in database.

URL: http://18.217.114.101:8080/github_scanner/api/all

Response:

An array of all hackthons' names in JSON format.

```
[  
  "Hackthon1",  
  "Hackthon2",  
  "Hackthon3",  
  "Hackthon4"  
]
```

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** `http://18.217.114.101:8080/github_scanner/api/all`
- Buttons:** Send, Save
- Tabs:** Params, Authorization, Headers, Body, Pre-request Script, Tests
- Params Table:**

KEY	VALUE	DESCRIPTION
Key	Value	Description
- Status:** 200 OK, Time: 307 ms, Size: 179 B
- Body Tab:** Pretty, Raw, Preview, JSON (selected)
- Response Body (JSON):**

```
1 [
2   "Hackthon1",
3   "Hackthon2",
4   "Hackthon3",
5   "Hackthon4"
6 ]
```