

API Provided:

- **Update Token (Post)**

Method Description:

User can use update token function to add new tokens or replace the old one. Server uses hackathonName as the primary key, when token pair are uploaded, server will check whether this hackathonName exist in the system, if it is a new name, just add it to the token set. If the hackathonName exist in the system, server will replace the old pair.

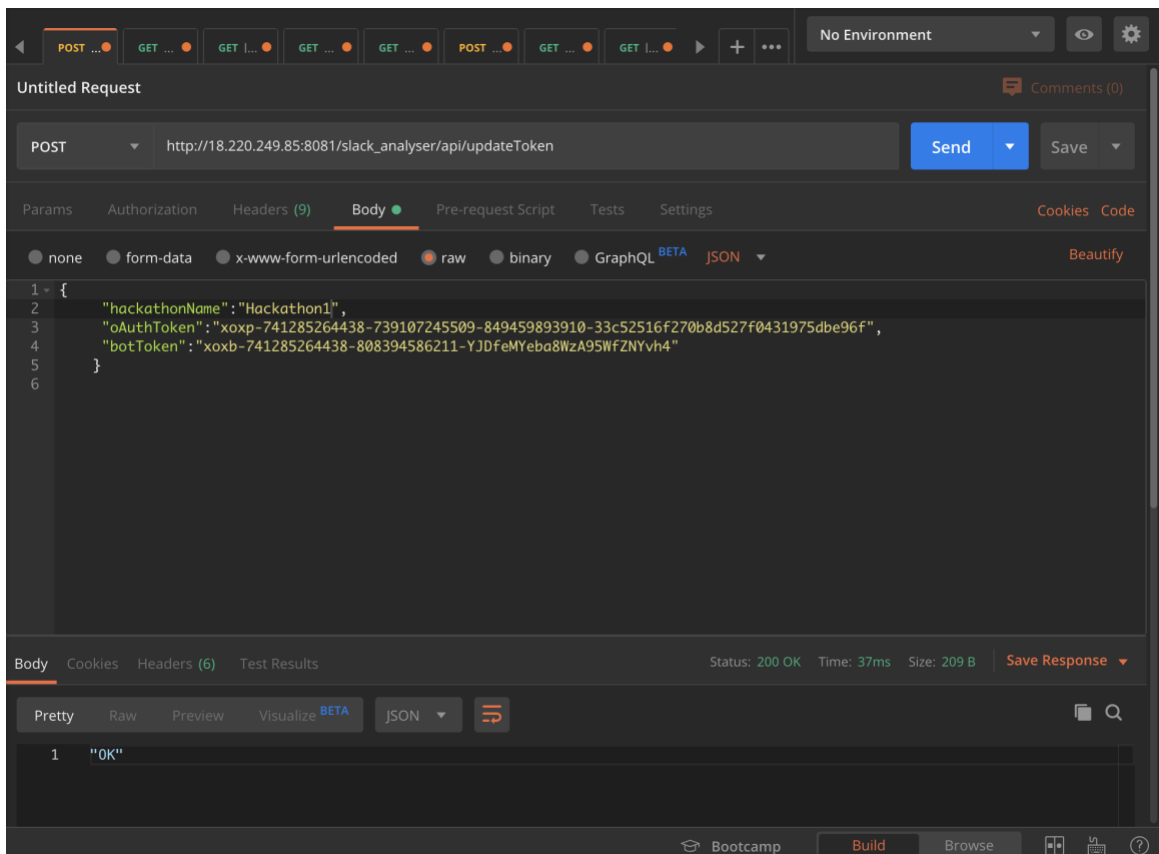
URL: http://<host-ip>/slack_analyser/api/updateToken

Input format:

- Headers: "Content-Type": "application/json"
- Body:

```
{
  "hackathonName":String,
  "oAuthToken":String,
  "botToken":String
}
```

Response: "OK"



- **Get All Token (Get)**

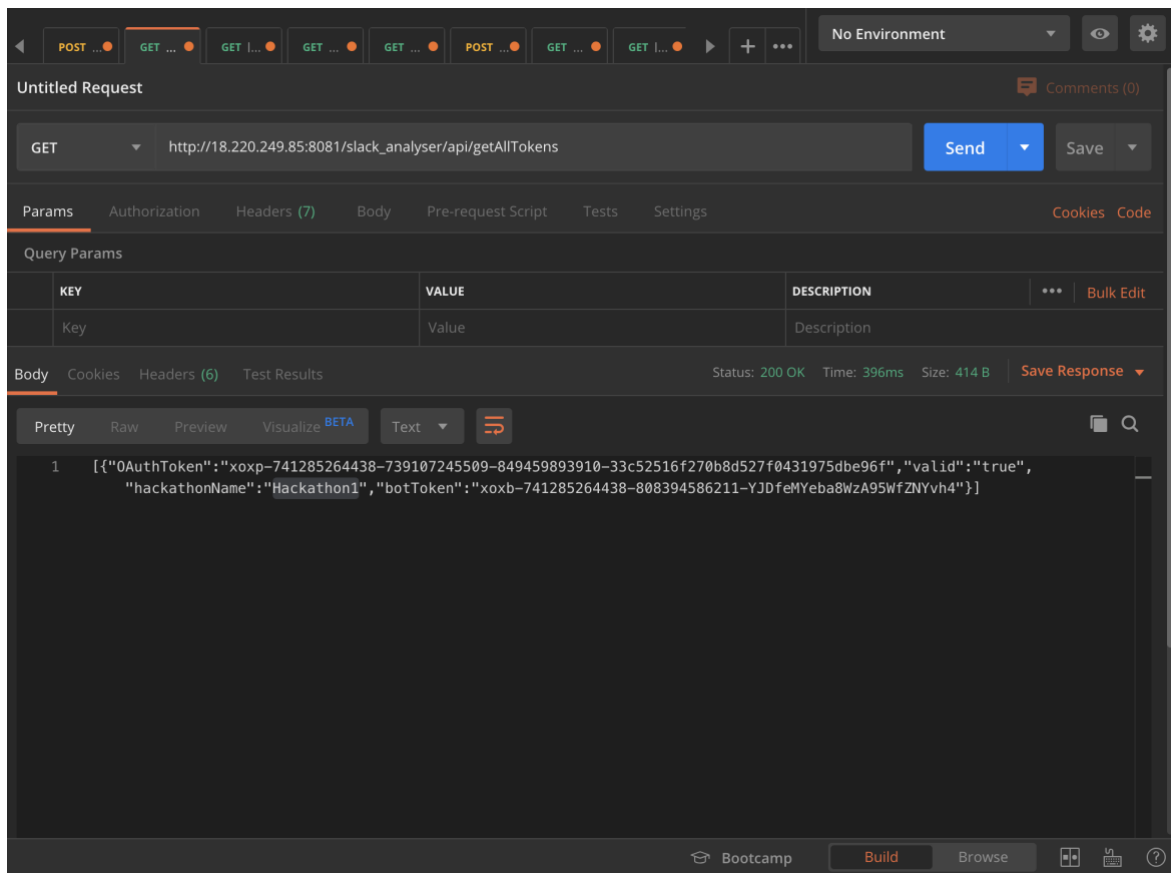
Method Description:

User can use get all tokens function to view all tokens exist in the system. Server will reply a json array to show all the token pairs and also the validity of them. If the token is not valid, it may be expired or wrong, user can use update token method to update new token pair.

URL: [.<host-ip>/slack_analyser/api/getAllTokens](http://<host-ip>/slack_analyser/api/getAllTokens)

Response:

```
[
  {
    "hackathonName":String,
    "Valid":"true",
    "OAuthToken":String,
    "botToken":String
  },
  {
    "hackathonName":String,
    "Valid":"false",
    "OAuthToken":String,
    "botToken":String
  },
  ...
]
```



- **Update new messages to MongoDB (Get)**

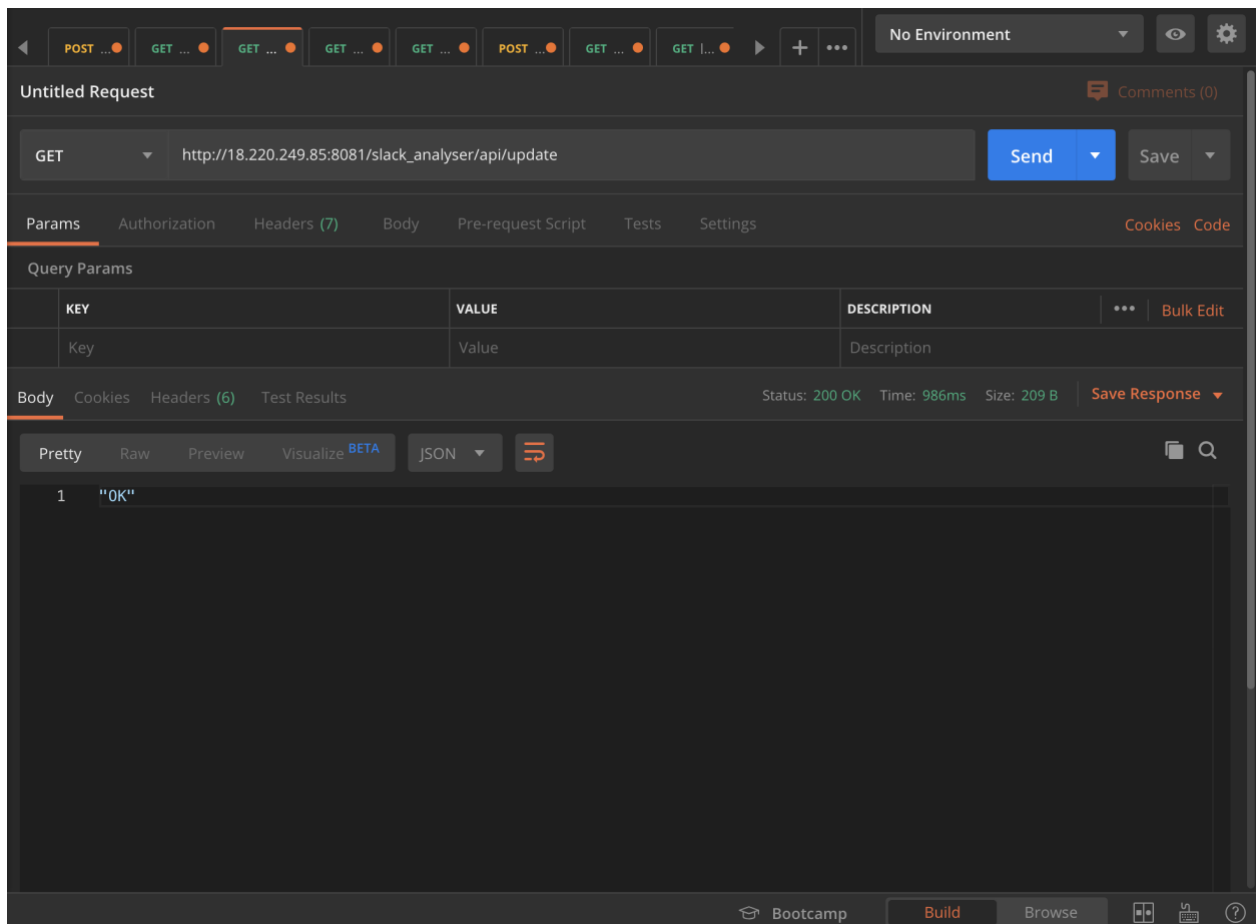
Method Description:

When user send a request to update new messages, server will first use tokens in the system to update all slack workspace one by one. For each workspace, first it gets the newest timestamp in mongoDB collection and use it to call Slack API and get the new messages. Server will also call another Slack API to get user information and combined it with the message, then write it into one document in the mongo collection.

URL: http://<host-ip>/slack_analyser/api/update

Input: null

Response : 200"OK"



- **Search messages (Get)**

Method Description:

User can search messages by using this API call, and we provide the search by hackathon name, channel name, user's email, keyword, time period. When this api is called, the server will connect to the mongoDB and use mongoDB's ad-hoc query to filter the data and send back a json array contains all the messages which fulfil the requirement to the user.

URL: http://<host-ip>/slack_analyser/api/search

Input format:

Parameters in the URL (these parameters can be null):

```
{
  "keyword": String,    // keyword the message contains
  "email": String,      // email for the user
  "start": String,      // a long number of the oldest message's timestamp
  "end": String,        // a long number of the newest message's timestamp
  "channel": String,    // channel name
  "hackathonName": String, // hackathon name
}
```

```
}
```

Example:

URL: ip>/slack_analyser/api/search?keyword=today&email=xiangyum@andrew.cmu.edu [<host-](mailto:xiangyum@andrew.cmu.edu)

Output:

```
[
  {
    "_id": "d76d0998-dbdf-4c56-9f22-aba374ed0249",
    "timeStamp": "1570024872.000100",
    "hackathonName": "ISM Project",
    "channelName": "capstone",
    "userEmail": "xiangyum@andrew.cmu.edu",
    "text": "Room for today's meeting",
    "userID": "UMR3577EZ"
  },
  {
    "_id": "0d9aa9a3-df4f-4c2e-bb7a-6ef98bb8196d",
    "timeStamp": "1568120700.000700",
    "hackathonName": "ISM Project",
    "channelName": "capstone",
    "userEmail": "xiangyum@andrew.cmu.edu",
    "text": "Booked interview room A007C for today's meeting",
    "userID": "UMR3577EZ"
  }
]
```

POST ...

GET ...

GET ...

GET ...

GET ...

POST ...

GET ...

GET ...

+

...

No Environment

GET

http://18.220.249.85:8081/slack_analyser/api/search?keyword=today&email=xiangyum@andrew.cmu.edu

Send

Save

ParamsAuthorizationHeaders (7)BodyPre-request ScriptTestsSettingsCookiesCode

Query Params

| | KEY | VALUE | DESCRIPTION | ... | Bulk Edit |
|-------------------------------------|---------|-------------------------|-------------|-----|-----------|
| <input checked="" type="checkbox"/> | keyword | today | | | |
| <input checked="" type="checkbox"/> | email | xiangyum@andrew.cmu.edu | | | |
| | Key | Value | Description | | |

BodyCookiesHeaders (6)Test Results

Status: 200 OKTime: 39msSize: 681 BSave Response

PrettyRawPreviewVisualize BETAJSON

```
1  {
2    {
3      "_id": "d76d0998-dbdf-4c56-9f22-aba374ed0249",
4      "timeStamp": "1570024872.000100",
5      "hackathonName": "Hackathon1",
6      "channelName": "capstone",
7      "userEmail": "xiangyum@andrew.cmu.edu",
8      "text": "Room for today's meeting",
9      "userID": "UMR3577EZ"
10   },
11   {
12     "_id": "0d9aa9a3-df4f-4c2e-bb7a-6ef98bb8196d",
13     "timeStamp": "1568120700.000700",
14     "hackathonName": "Hackathon1",
15     "channelName": "capstone",
16     "userEmail": "xiangyum@andrew.cmu.edu",
17     "text": "Booked interview room A007C for today's meeting",
18     "userID": "UMR3577EZ"
19   }
20 }
```

BootcampBuildBrowse