

Documentation

/

Methods: HEAD, OPTIONS, GET

Arguments: *None*

Description: Gets basic splash screen (click log in or sign up with facebook)

Returns:

render_template: basic splash

/api/allgraphs/<uuid>

Methods: HEAD, OPTIONS, GET

Arguments: *uuid*

Description: Returns every set of tiles (only called when app is first loaded) with default parameters

Returns:

JSON: parameters for all of the homepage graphs

Args:

uuid (str): user id generated by facebook javascript SDK

/api/chatbot/<uuid>

Methods: POST, OPTIONS

Arguments: *uuid*

Description: Chat with the chatbot

Args:

uuid (str): user id generated by facebook javascript SDK

Returns:

JSON: reply to user message (sent in the request body) as:

```
{
  'uuid': users uuid (str)
  'message': message content (str)
}
```

/api/compute-user-stats/<uuid>

Methods: PUT, OPTIONS

Arguments: *uuid*

Description: Computes all message statistics for the given user

Args:

uuid (str): user id generated by facebook javascript SDK

Returns:

status code only. 2XX if signup success, else 5XX.

/api/histogram/<uuid>/<histogram_type>

Methods: HEAD, OPTIONS, GET

Arguments: *histogram_type* , *uuid*

Description: Requests histogram given parameter of person (me, friend 1, friend 2, etc...)

and the selected time frame (1 week, 1 month, 3 month, etc...)

Args:

uuid (str): user id generated by facebook javascript SDK

histogram_type (str): type of histogram tile to route to

Returns:

JSON: parameters for histogram construction as:

```
{
  'graphType': 'histogram'
  'someMetaData': information (str),
  'numMessagesOverTime': [
    {Date (str) : value (int)},
    {'2017-10-23T19:00:00.511Z' : 55},
    ...
    {Date (str) : value (int)}
  ]
}
```

/api/login/<uuid>

Methods: PUT, OPTIONS

Arguments: *uuid*

Description: Attempt to login to home page

Returns:

String: UUID of logged in user

Args:

uuid (str): user id generated by facebook javascript SDK

/api/signup/<uuid>/<name>

Methods: POST, OPTIONS

Arguments: *uuid* , *name*

Description: Attempts to create user and then return the UUID of the newly signed up user

Args:

uuid (str): user id generated by facebook javascript SDK

Returns:

status code only. 2XX if signup success, else 4XX.

/api/upload/<uuid>

Methods: POST, OPTIONS

Arguments: *uuid*

Description: Uploads user data to server

Args:

uuid (str): user id generated by facebook javascript SDK

Returns:

status code only. 2XX if signup success, else 4XX.

/api/user-freq-stats/<uuid>

Methods: HEAD, OPTIONS, GET

Arguments: *uuid*

Description: Requests line graph given parameter of person (me, friend 1, friend 2, etc...)

and the selected time frame (1 week, 1 month, 3 month, etc...)

Args:

uuid (str): user id generated by facebook javascript SDK

sender (str): name of the sender that the user messages

date (str): the start date

Returns:

JSON: parameters for graph construction

```
{
  'graphType': 'linegraph'
  'someMetaData': information (str),
  'numMessagesOverTime': [
    {Date (str) : value (int)},
    {'2017-10-23T19:00:00.511Z' : 55},
    ...
    {Date (str) : value (int)}
  ]
}
```

/api/wordle/<uuid>

Methods: HEAD, OPTIONS, GET

Arguments: *uuid*

Description: Requests wordle given parameter of person (me, friend 1, friend 2, etc...)

and the selected time frame (1 week, 1 month, 3 month, etc...)

Args:

uuid (str): user id generated by facebook javascript SDK

Returns:

JSON: parameters for wordle construction structured as:

```
{
```

```
'Thread name': the name of the thread (str)
'wordList': [
  {given word (str):number of occurrences (int)},
  ...
  {given word (str):number of occurrences (int)}
]
}
```