Yum4FIT Documentation

Release 0.2

Simon Štefunko

Feb 12, 2018

Contents:

CHAPTER 1

Intro

Yum4FIT is the simple application to improve your cooking skills. The application provides its funcionality via CLI part and also user is able to open graphic interface. The main functionality is to provides recipes for the user. User can cook the food according to recipe and share the picture of the food on Instagram. After that, the application offers an option to gain XP points for likes on Yum4FIT foods on Instagram. The user's level is calculated from his XP points. The generated recipes are respecting the level of the user. So let's cook beter and better and become a master!

1.1 Configuration

This part briefly talks about the configuration of Yum4FIT application. If the use wants to properly use the functionality of this application, there is need to be some configuration done. The recommendation is to create **config.cfg** file, which is called configuration file or config file.

Example of the config file:

```
[instagram]
username = RandomUser
password = MyTotalySecretPassword
hashtag = #Yum4FIT

[yummly]
api-id = Th1s1sTh31D
api-key = 4ppl1c4t10nk3yfr0mYuMmlY

[server]
url = http://someserver:5000/
```

1.1.1 Section [instagram]

This section includes parameters for InstagramAPI.

- username - Instagram username

- password - Instagram password
- hashtag - Instagram hashtag for #Yum4FIT food - not recommended to change

1.1.2 Section [yummly]

- api-id - Application identification code on Yummly
- api-key - Application key code on Yummly

1.1.3 Section [server]

- url - URL of the deployed Flask server of the host user

1.2 Command line interface

The Yum4FIT provides command line interface. The commands are of this application are accessible via entry point **yum**. It offers an option to set a few parameters via command line

1.2.1 Options

- –config/-c - set the path and filename to configuration file
- **–username/-u - Instagram username in CLI part. May be set as environment variable** *IG_USERNAME*
- **–password/p Instagram password used in CLI part. May be set as environment variable** *IG_PASSWORD*

1.2.2 Commands

- **recipe**

  Generates the level-respected recipe. This command returns the name of the food, needed ingredients and hyperlink to the step-by-step guide with details, how to cook the food. There are some parameters, that may be used with this command:

  - –diet/-d - filter foods according to diet
  - –alergy/-a - filter foods according to allergies
  - –cuisine/-c - filter foods according to national cuisines
  - –exclude_cuisine/-ec - filter foods according to excluded national cuisines
  - –ingredient/-i - filter foods according to ingredients
  - –exclude_ingredient/-ei - filter foods according to excluded ingredients
  - –holiday/-h - filter foods according to the holiday
  - –exclude_holiday/-eh - filter foods according to the excluded holiday
  - –phrase/-p - filter foods according to the phrase
  - **–file/f - if this flag is set, CLI parameters are ignored and parameters and loaded** from the configuration file in section *[parameters]*

- **share**

  Uploads photo of the food on the Instagram. It needs the filepath as an argument. This command offers one option:

  - –caption/-c - caption for the Instagram post

- **food**

  Provides the list of all user's Yum4FIT food *(with #Yum4FIT hashtag)* from Instagram. The food is represented with the *id* of the Instagram post and *url* of the Instagram photo

- **gain**

  Collects the XP points calculated from the all sumarized likes of Yum4FIT posts. This command save the state and returns the simple text representation of the state. Using this command, user is able to achieve higher level. There is one option:

  - –id/-i - identification code of food post for retrieve XP gain from one post

- **add_friend**

  Add friend to friend list with id of the food post. This commnad create relationship between the Instagram user and food, so it should represent cooking and sharing the food with friend. After this command, the friend is able to confirm sharing of the food via web part of application. The friend's like on the food has value of **5** regular likes. It needs two arguments - *username* of the friend and *id* of the food post.

- **run_server**

  Runs the flask server, which is the web part of the application. There are some options:

  - –host/-h - server address
  - –port/-p - server port
  - –debug/-d - debug mode

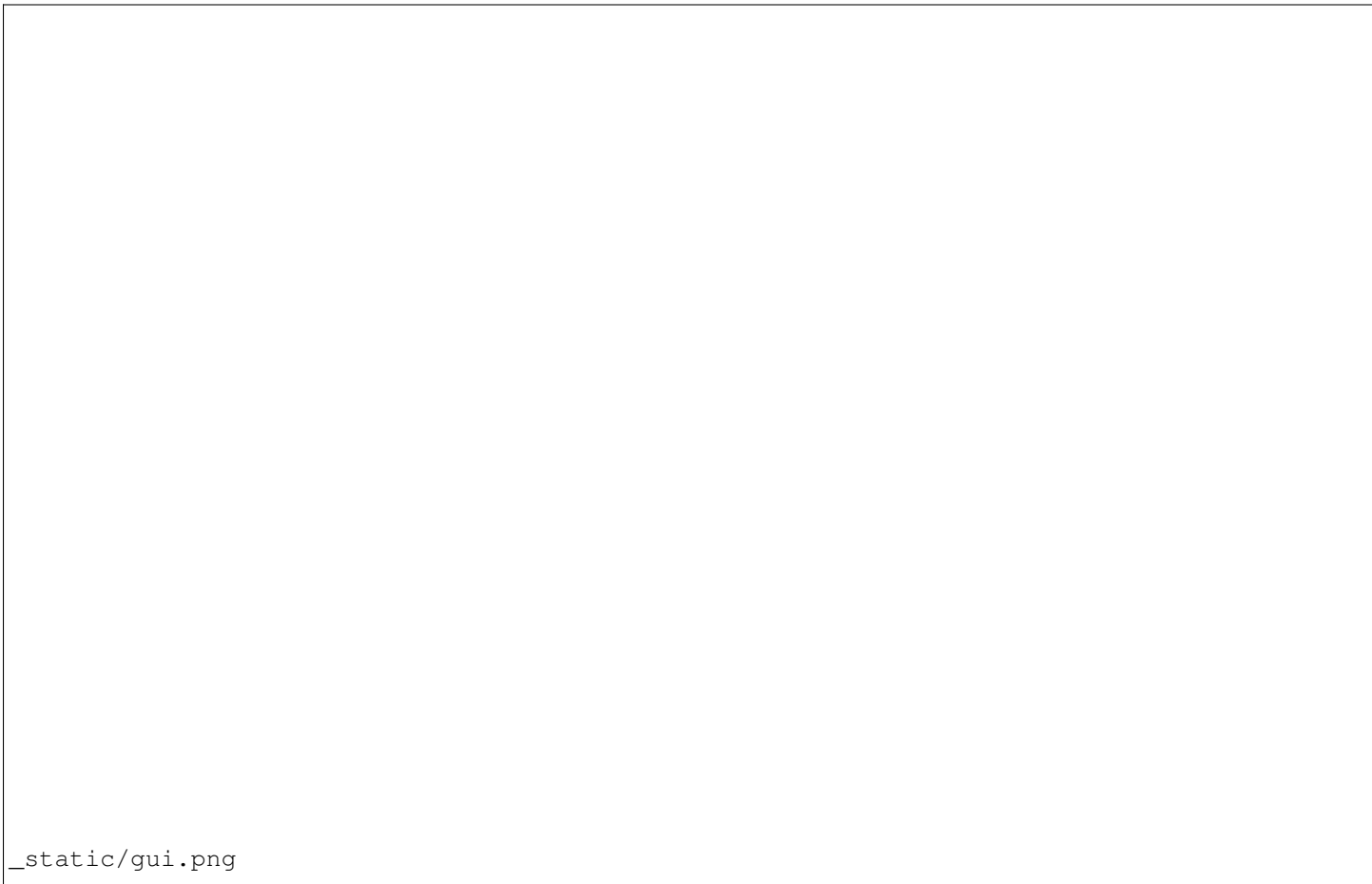
- **run**

  Opens the GUI part of the application

1.3 Graphic user interface

The main purpose of graphic user interface of this applications is to provide the functionality for users, who prefer GUI over CLI. Interface is very simple and minimalistic. Here you can see example:

1.3.1 Example of user's GUI



1.3.2 GUI parts

- **Food list**

  As you can see on the example, *All your food* block shows the brief list of all user's food posts. Every item of this list consits of the picture of image (picture from Instagram) and the ID of that Instagram post. The main purpose of this was to extend the functionality of Instagram for this use. It's scrollable and its images are loaded from the URLs.

- **Add friend button**

  Button *Add friend* under the list adds friend to friend list. The username of the friend must be written to input line next to button and one item(food) have to be chosen in the list.

- **Share button**

  Button *Share* open filedialog, in which user may choose the picture. After that, the caption may be set. Click on this button includes uploading chosen picture to Instagram.

- **Profile box**

  Profile block sumarizes brief information about the host user - username, fullname, email, profile picture and level. Profile box includes also *Gain button*

- **Gain button**

  Click on this button runs collecting yums from the users - it means the likes on Instagram posts will be collected and the new state will be set. The level of the user may change after this action.

- **Actual/last recipe box**

  The box under the profile displays the name and ingredients of the actual/last recipe. Mostly, there is also the picture of that food loaded from the URL. There are two special buttons inside this box.

- **Guide button**

  Guide button opens the browser and redirects to the page, where should be step-by-step guide how to cood actual/last food and see more detail information about the food.

- **Generate button**

  Generates the new recipe to the actual recipe box. If the user wants to set some searching parameters, all he needs to do is to write them to section *[parameters]* in configuration file. To load parameters from file, *Checkbox "Parameters from config file"* must be checked.

1.4 Web part

The web part of the application is builded using Flask. Flask server renders simple web page index that display brief information about the host user. There is the profile box including information about the user on the top of the page. The actual/last food is displayed right below the profile box. It consits of the name of the food, ingredients, step-by-step guide link and the picture of the food.

The third main part of the web is food list of the host user. Guest user is able to choose foods and give them *Yum*. Yum means like on the post on the Instagram and also the confirmation of sharing that food with the host. So when host cooks the food, uploads it on Instagram and set the guest as a *friend*, it should represents sharing of the food. After that, if the guests send *Yum* on the shared food, he confirms that he shared the food and liked that! This is the main reason, why are for theese special likes more important that regular. Someone, who really enjoyed your food appreciate your work.

Synchronisation of the data is achieved using POST method from GUI and CLI part to web. The web part uses mechanisms for authenticating and processing theese requests.

Server may be started using CLI command *yum run_server*.

1.5 yum

1.5.1 yum package

Submodules

yum.CONSTANTS module

yum.Guest module

yum.Parser module

yum.account module

yum.cli module

yum.connector module

yum.food module

yum.gui module

yum.profile module

yum.state module

yum.unity module

yum.web module

yum.yum module

Module contents

CHAPTER 2

Indices and tables

- genindex
- modindex
- search