wowa Documentation

Release 0.1

Felipe Arruda Pontes

December 01, 2014

Contents

Contents:

CHAPTER 1

wowa

1.1 About this:

WOWA is a simple aplication that uses the WOW API to get info about the auction house price for some items.

1.2 How Does It Works:

The idea is to register you World of Warcraft character, then will search for an Item, and mark it as tracked.

After this the *wowa* will start to getter data about the avarage price on that item, and the price that you sold them in the auction house.

It will generate a simple chart showing the comparison on the avarage and your prices, each day, for up to the last 30 days.

For more information read more on the documentation: http://wowa.readthedocs.org/en/latest/usage.html

1.3 Install:

Checkout the documentation page about this: http://wowa.readthedocs.org/en/latest/install.html

1.3.1 LICENSE

This software is distributed using MIT license, see LICENSE file for more details.

CHAPTER 2

Install

Very simple, just follow this steps:

1. Clone *wowa* locally:

   ```
   $ git clone https://github.com/arruda/wowa.git
   ```

2. Since *wowa* uses PostgreSQL in local development and in production, you'll need to install at least *libpq* and *Python* header files. On ubuntu this can be done by:

   ```
   $ sudo apt-get install libpq-dev python-dev
   ```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

   ```
   $ mkvirtualenv wowa
   $ cd wowa/
   $ pip install -r requirements/local.txt
   ```

4. Run the migrations:

   ```
   $ python wowa/manage.py migrate
   ```

5. Create a super-user for you:

   ```
   $ python wowa/manage.py createsuperuser
   ```

6. Run the server:

   ```
   $ python wowa/manage.py runserver
   ```

That's all.

CHAPTER 3

Usage

CHAPTER 4

Data Analysis

I'll describe here how I image the data analysis process will look like:

4.1 Get the Data:

- Every hour the system will try to get a more recent data from WOW API on the auctions of all registered Realms;
- This auction's data will be saved with the timestamp of the *last_modified* field from WOW API;

4.2 Analysis:

When a user, ask for the tracked item (ex: *#12345*), it will:

- get the realm it's been asked for;
- get the datetime for the latest 30 days, counting on today;
- For each day, it will first try to check if the information is already saved in a Django Model (PostgreSQL), this will happen if the analysis for that day/realm/item was already done;
- **if the information is not present in PostgreSQL, then search the MongoDB's data and get all the auctions.json of that day;**
  - after that, it will go through each *auctions.json*, and get the *Avarage Price* and the *User Price* for the tracked item (ex: #12345);
  - then it will make the *avarage* for the hole day with the *Avarage Price* of each *auction.json*, and the same thing with the *User Price*.
  - will save this information in a Django Model (PostgreSQL), to make it easy to retrieve, and don't re-do work.

CHAPTER 5

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

5.1 Types of Contributions

5.1.1 Report Bugs

Report bugs at https://github.com/arruda/wowa/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

5.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” is open to whoever wants to implement it.

5.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “feature” is open to whoever wants to implement it.

5.1.4 Write Documentation

wowa could always use more documentation, whether as part of the official wowa docs, in docstrings, or even on the web in blog posts, articles, and such.

5.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/arruda/wowa/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

5.2 Get Started!

Ready to contribute? Here's how to set up *wowa* for local development.

1. Fork the *wowa* repo on GitHub.
2. Clone your fork locally:

   ```
   $ git clone git@github.com:your_name_here/wowa.git
   ```

3. Since *wowa* uses PostgreSQL in local development and in production, you'll need to install at least *libpq* and *Python* header files. On ubuntu this can be done by:

   ```
   $ sudo apt-get install libpq-dev python-dev
   ```

4. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

   ```
   $ mkvirtualenv wowa
   $ cd wowa/
   $ pip install -r requirements/local.txt
   ```

5. Create a branch for local development:

   ```
   $ git checkout -b name-of-your-bugfix-or-feature
   ```

   Now you can make your changes locally.

6. When you're done making changes, check that your changes pass flake8 and the tests. But first you'll need to create a new virtualenv:

   ```
   $ mkvirtualenv wowa_test
   $ pip install -r requirements/test.txt
   $ make lint
   $ make coverage
   ```

7. Commit your changes and push your branch to GitHub:

   ```
   $ git add .
   $ git commit -m "Your detailed description of your changes."
   $ git push origin name-of-your-bugfix-or-feature
   ```

8. Submit a pull request through the GitHub website.

5.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring.

3. Check https://travis-ci.org/arruda/wowa/pull_requests and make sure that the tests pass.

CHAPTER 6

Credits

6.1 Development Lead

- Felipe Arruda Pontes <contato@arruda.blog.br>

6.2 Contributors

None yet. Why not be the first?

CHAPTER 7

Deploy

This is where you describe how the project is deployed in production.

CHAPTER 8

Indices and tables

- *genindex*
- *modindex*
- *search*