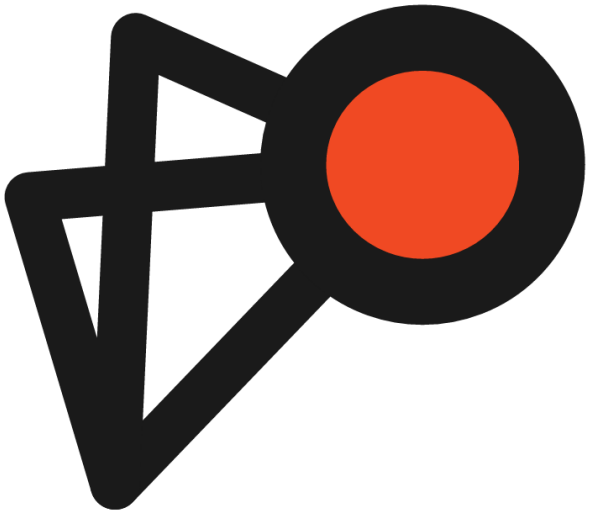
Foundit



Ken Ford, Ryan Davis, Eugene Ho, Luke Nguyen

Version 1.0

Thu May 4 2017

Table of Contents

Table of contents

Namespace Index

Namespace List

Hierarchical Index

Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

- Model
  - foundit.models.Queuery .....
- AppConfig
  - foundit.apps.FounditConfig.....

Class Index

Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

File Index

File List

Here is a list of all files with brief descriptions:

Namespace Documentation

foundit Namespace Reference

Namespaces

- **admin**
- **apps**
- **foundit**
- **foundit_old**
- **graph**
- **graphtest**
- **models**
- **tests**
- **urls**
- **utils**
- **views**

foundit.admin Namespace Reference

foundit.apps Namespace Reference

Classes

- class **FounditConfig**

foundit.foundit Namespace Reference

Functions

- def **schedule** (subreddit, postLimit, topComLimit, topReplyLimit, topWordLimit, topUserLimit, oldestPostLimit, activePostLimit, wc)
- def **getSubmissionAge** (submission)
  *temp=q.fetch_job(jobq[qindex]).get_id.result temp=q.fetch_job(jobq[qindex]).id.result if(temp): results.append(temp) q.remove(q.fetch_job(jobq[qindex]).id) check+=1 gindex+=1 print("WORKER SEARCH #"+str(qindex)+(" DONE!!!")+"vTOTAL COMPLETE: "+str(check)) time.sleep(workercount+3) qindex+=1 time.sleep(workercount*2) print("WAITING...") if(check!=workercount): qindex=0 COMBINE ALL DATA ONCE CHECK PASSES ORDER OF RETURN FOR WORKERS 0titleWords, 1nounDict, 2userDict, 3topCom, 4topReply, 5oldestPost, 6activePost, 7postsAnalyzed, 8totalLengthAll, 9commentsAnalyzed)*
- def **adjust** (l, limit, indexToCompare, thingToAdd)
- def **search** (subreddit, postLimit, topComLimit, topReplyLimit, topWordLimit, topUserLimit, oldestPostLimit, activePostLimit, startpos, endpos, qindex)

Variables

- **q** = Queue(connection=conn)

Function Documentation

def foundit.foundit.adjust (*l*, *limit*, *indexToCompare*, *thingToAdd*)

```
The adjust function is responsible for the handling of the word graph. It first fills a list of
words to compare which the lowest value is popped off in exchange for another word to add.
```

def foundit.foundit.getSubmissionAge (*submission*)

temp=q.fetch_job(jobq[qindex]).get_id.result temp=q.fetch_job(jobq[qindex]).id.result if(temp): results.append(temp) q.remove(q.fetch_job(jobq[qindex]).id) check+=1 gindex+=1 print("WORKER SEARCH #"+str(qindex)+(" DONE!!!")+"vTOTAL COMPLETE: "+str(check)) time.sleep(workercount+3) qindex+=1 time.sleep(workercount*2) print("WAITING...") if(check!=workercount): qindex=0 COMBINE ALL DATA ONCE CHECK PASSES ORDER OF RETURN FOR WORKERS 0titleWords, 1nounDict, 2userDict, 3topCom, 4topReply, 5oldestPost, 6activePost, 7postsAnalyzed, 8totalLengthAll, 9commentsAnalyzed)

```
This function gets the age of the current submission based on the comparision between the submission
time and the current time.
```

def foundit.foundit.schedule (*subreddit*, *postLimit*, *topComLimit*, *topReplyLimit*, *topWordLimit*, *topUserLimit*, *oldestPostLimit*, *activePostLimit*, *wc*)

```
The schedule function is responsible for scheduling worker jobs. These workers will divide the
total work time amongst themselves to decrease the time of the search and compiling of Foundit.
```

def foundit.foundit.search (*subreddit*, *postLimit*, *topComLimit*, *topReplyLimit*, *topWordLimit*, *topUserLimit*, *oldestPostLimit*, *activePostLimit*, *startpos*, *endpos*, *qindex*)

```
The search function is responsible for gathering and parsing the reddit data for further use. It
also uses the language tool NLTK to create a working dictionary for the found words to be passed
through. This also alleviates the need to worry about common words such as "the" or "a".  The
function then returns parsed reddit data ready for further use.
```

Variable Documentation

foundit.foundit.q = Queue(connection=conn)

foundit.foundit_old Namespace Reference

Functions

- def **getSubmissionAge** (submission)
- def **search** (subreddit, postLimit, topComLimit, topWordLimit, topUserLimit, ohSnapLimit, oldestPostLimit)

Function Documentation

def foundit.foundit_old.getSubmissionAge (*submission*)

def foundit.foundit_old.search (*subreddit*, *postLimit*, *topComLimit*, *topWordLimit*, *topUserLimit*, *ohSnapLimit*, *oldestPostLimit*)

foundit.graph Namespace Reference

Functions

- def **uautolabel** (rects, ax)
- def **urenderGraph** (dataSet)
- def **renderGraph** (dataSet)

Function Documentation

def foundit.graph.renderGraph (*dataSet*)

```
The renderGraph function converts data into a bar graph object. It handles the measuring, drawing,
and defining of the various aspects of the graph. From this point, it converts the graph code from
Python to HTML and returns this value. This HTML code is then transferred to the website for the
viewer.
```

def foundit.graph.uautolabel (*rects*, *ax*)

```
Function attaches a text label above each bar displaying its height. This function provides the
reader easy context on each value in the finished graph so that an understanding of the data can
be made faster.
```

def foundit.graph.urenderGraph (*dataSet*)

```
This function renders a graph for the use in the specialized word graphs. This portion of the code
was not completed at the time of submission, and so this function is currently not being used.
```

foundit.graphtest Namespace Reference

Variables

- list **a** = [1, 2]
- list **b** = [2, 5]
- **bins**
- **weights**

Variable Documentation

foundit.graphtest.a = [1, 2]

list foundit.graphtest.b = [2, 5]

foundit.graphtest.bins

foundit.graphtest.weights

foundit.models Namespace Reference

Classes

- class **Queuery**

foundit.tests Namespace Reference

foundit.urls Namespace Reference

Variables

- list **urlpatterns**

Variable Documentation

list foundit.urls.urlpatterns

```
Initial value:    1 = [
    2     url(r'^$', views.index, name='index'),
    3     url(r'^results/$', views.results, name='results'),
    4     url(r'^loading/$', views.loading, name='loading'),
    5     url(r'^loading/checkJob', views.checkJob, name='checkJob'),
    6     url(r'^testResults/$', views.testResults, name='testResults'),
    7 ]
```

foundit.utils Namespace Reference

Functions

- def **returnURL** (str1, str2)

Function Documentation

def foundit.utils.returnURL (*str1*, *str2*)

```
combine strings into full URL
```

foundit.views Namespace Reference

Functions

- def **index** (request)
- def **loading** (request)
- def **checkJob** (request)
- def **testResults** (request)
- def **results** (request)

Variables

- **q** = Queue(connection=conn)
- string **title** = ""
- int **workercount** = 5

Function Documentation

def foundit.views.checkJob (*request*)

```
This function requests the jobid of a reddit instance that was established in loading. From this
point, it returns an html object of either the jobid or the existing job with that id.
```

def foundit.views.index (*request*)

```
The index function establishes connection to the index of website.
```

def foundit.views.loading (*request*)

```
The loading function is responsible for requesting a reddit instance and gathering the various
data from it. From that point, it adds these values to a list and saves this as a job. Then the
function establishes a context for the instance which is made from the current job id and subreddit
values. Finally, this function returns the rendering of the context.
```

def foundit.views.results (*request*)

```
This function is responsible for parsing the data from reddit into appropriate variables.
Additionally, it passes this data into graphing functions for its later use in providing
result-based graphs on the website.
Results also returns the context variable which contains the necessary data for later construction
of the website.
This function provides the heavy lifting of Foundit, as it provides the data used for the later
analtyics.
```

def foundit.views.testResults (*request*)

```
This function's purpose is to return an html response of results as a test of the HTMLResponse
function.
```

Variable Documentation

foundit.views.q = Queue(connection=conn)

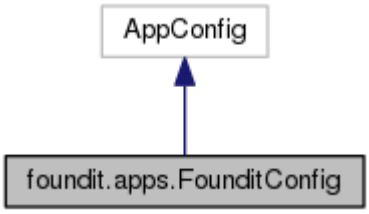
string foundit.views.title = ""

int foundit.views.workercount = 5

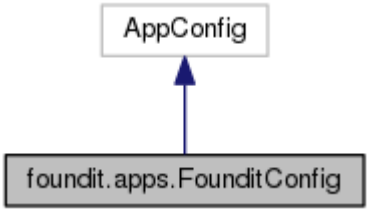
Class Documentation

foundit.apps.FounditConfig Class Reference

Inheritance diagram for foundit.apps.FounditConfig:



Collaboration diagram for foundit.apps.FounditConfig:



Static Public Attributes

- string **name** = 'foundit'

Member Data Documentation

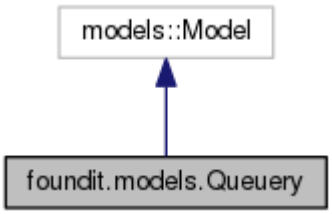
string foundit.apps.FounditConfig.name = 'foundit' `[static]`

The documentation for this class was generated from the following file:

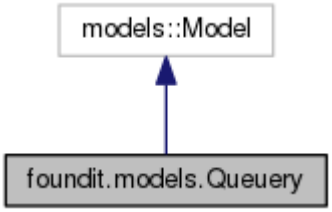
- /home/user/Dropbox/School/CSCI Stuff/CSCI 3308- Software Development/Foundit Project/ufoundit/foundit/**apps.py**

foundit.models.Queuery Class Reference

Inheritance diagram for foundit.models.Queuery:



Collaboration diagram for foundit.models.Queuery:



Public Member Functions

- def **__str__** (self)

Static Public Attributes

- **subreddit** = models.CharField(max_length=200)

Member Function Documentation

def foundit.models.Queuery.__str__ (***self*****)**

Member Data Documentation

foundit.models.Queuery.subreddit = models.CharField(max_length=200) `[static]`

The documentation for this class was generated from the following file:

- /home/user/Dropbox/School/CSCI Stuff/CSCI 3308- Software Development/Foundit Project/ufoundit/foundit/**models.py**

File Documentation

/foundit/__init__.py File Reference

Namespaces

- **foundit**

//foundit/admin.py File Reference

Namespaces

- **foundit.admin**

/foundit/apps.py File Reference

Classes

- class **foundit.apps.FounditConfig**

Namespaces

- **foundit.apps**

/foundit/foundit.py File Reference

Namespaces

- **foundit.foundit**

Functions

- def **foundit.foundit.schedule** (subreddit, postLimit, topComLimit, topReplyLimit, topWordLimit, topUserLimit, oldestPostLimit, activePostLimit, wc)
- def **foundit.foundit.getSubmissionAge** (submission)
  *temp=q.fetch_job(jobq[qindex]).get_id.result temp=q.fetch_job(jobq[qindex]).id.result if(temp): results.append(temp) q.remove(q.fetch_job(jobq[qindex]).id) check+=1 gindex+=1 print("WORKER SEARCH #"+str(qindex)+(" DONE!!!")+"vTOTAL COMPLETE: "+str(check)) time.sleep(workercount+3) qindex+=1 time.sleep(workercount*2) print("WAITING...") if(check!=workercount): qindex=0 COMBINE ALL DATA ONCE CHECK PASSES ORDER OF RETURN FOR WORKERS 0titleWords, 1nounDict, 2userDict, 3topCom, 4topReply, 5oldestPost, 6activePost, 7postsAnalyzed, 8totalLengthAll, 9commentsAnalyzed)*
- def **foundit.foundit.adjust** (l, limit, indexToCompare, thingToAdd)
- def **foundit.foundit.search** (subreddit, postLimit, topComLimit, topReplyLimit, topWordLimit, topUserLimit, oldestPostLimit, activePostLimit, startpos, endpos, qindex)

Variables

- **foundit.foundit.q** = Queue(connection=conn)

/foundit/foundit_old.py File Reference

Namespaces

- **foundit.foundit_old**

Functions

- def **foundit.foundit_old.getSubmissionAge** (submission)
- def **foundit.foundit_old.search** (subreddit, postLimit, topComLimit, topWordLimit, topUserLimit, ohSnapLimit, oldestPostLimit)

/foundit/graph.py File Reference

Namespaces

- **foundit.graph**

Functions

- def **foundit.graph.uautolabel** (rects, ax)
- def **foundit.graph.urenderGraph** (dataSet)
- def **foundit.graph.renderGraph** (dataSet)

/foundit/graphtest.py File Reference

Namespaces

- **foundit.graphtest**

Variables

- list **foundit.graphtest.a** = [1, 2]
- list **foundit.graphtest.b** = [2, 5]
- **foundit.graphtest.bins**
- **foundit.graphtest.weights**

/foundit/models.py File Reference

Classes

- class **foundit.models.Queuery**

Namespaces

- **foundit.models**

/foundit/tests.py File Reference

Namespaces

- **foundit.tests**

/foundit/urls.py File Reference

Namespaces

- **foundit.urls**

Variables

- list **foundit.urls.urlpatterns**

/foundit/utils.py File Reference

Namespaces

- **foundit.utils**

Functions

- def **foundit.utils.returnURL** (str1, str2)

/foundit/views.py File Reference

Namespaces

- **foundit.views**

Functions

- def **foundit.views.index** (request)
- def **foundit.views.loading** (request)
- def **foundit.views.checkJob** (request)
- def **foundit.views.testResults** (request)
- def **foundit.views.results** (request)

Variables

- **foundit.views.q** = Queue(connection=conn)
- string **foundit.views.title** = ""
- int **foundit.views.workercount** = 5

Index

INDEX