flicket

Release 0.2.3

evereux@gmail.com

Jun 21, 2020

CONTENTS:

1 Why Flicket? 3

1.1 Requirements . . . 3
1.2 Installation . . . 3
1.3 Administration . . . 5
1.4 Exporting / Importing Flicket Users . . . 6
1.5 Installing A Webserver . . . 7
1.6 Adding Additional Languages . . . 9
1.7 Flicket - FAQ . . . 10
1.8 Screenshots . . . 12
1.9 API . . . 18

Python Module Index 35

HTTP Routing Table 37

Index 39

Flicket is a simple web based ticketing system written in Python using the flask web framework which supports English and French locales.

CHAPTER

ONE

WHY FLICKET?

I could not find a simple open source ticketing system that I really liked. So, decided to have a crack at creating something written in Python.

1.1 Requirements

1.1.1 Operating System

This will run on either Linux or Windows. Mac is untested.

1.1.2 Python

Python =>3.6 - I have not tested earlier versions of Python 3.

1.1.3 SQL Database Server

Out of the box Flicket is configured to work with MySQL. But there should be no reason other SQLAlchemy supported databases won't work just as well.

Note: When I last tried SQLite I had problems configuring the email settings within the administration settings. You may have to change them manually within SQLite.

1.1.4 Web Server

For a production environment a webserver such as Apache or nginx should be used to serve the application.

1.2 Installation

First read *Requirements*.

It is good practise to create a virtual environment before installing the python package requirements. Virtual environments can be considered a sand boxed python installation for a specific application. They are used since one application may require a different version of a python module than another.

1.2.1 Getting Flicket

The source code for Flicket is hosted at GitHub. You can either get the latest frozen zip file or use the latest master branch.

> **Warning:** If you are upgrading from a previous version please read the CHANGELOG.

Zip Package

Download Flicket Dist.zip and unzip.

Master Branch

Get the latest master branch from github using git:

```
git clone https://github.com/evereux/flicket.git
```

Alternatively, download and unzip the master branch zip file.

1.2.2 Installing Python Requirements

Install the requirements using pip::

```
(env) C:\<folder_path>\flicket> pip install -r requirements.txt
```

1.2.3 Set Up

1. Create your database and a database user that will access the flicket database.
2. If you are using a database server other than MySQL you should change the db_type value within *config.py*. See SQLAlchemy_documentation for options.
3. Create the configuration json file:

   ```
   python -m scripts.create_json
   ```

4. Upgrade the database using manage.py from the command line:

   ```
   python manage.py db upgrade
   ```

6. Run the set-up script:. This is required to create the Admin user and site url defaults. These can be changed again via the admin panel once you log in:

   ```
   python manage.py run_set_up
   ```

7. Running development server for testing:

   ```
   python manage.py runserver
   ```

Log into the server using the username *admin* and the password defined during the setup process.

1.3 Administration

1.3.1 Command Line Options

From the command line the following options are available.

```
python manage.py

usage: manage.py [-?]
                 {db,run_set_up,export_users,import_users,update_user_posts,update_
↪user_assigned,email_outstanding_tickets,runserver,shell}
                 ...

positional arguments:
  {db,run_set_up,export_users,import_users,update_user_posts,update_user_assigned,
↪email_outstanding_tickets,runserver,shell}
    db                   Perform database migrations
    run_set_up
    export_users         Command used by manage.py to export all the users from
                         the database to a json file. Useful if we need a list
                         of users to import into other applications.
    import_users         Command used by manage.py to import users from a json
                         file formatted such: [ { username, name, email,
                         password. ]
    update_user_posts    Command used by manage.py to update the users total
                         post count. Use when upgrading from 0.1.4.
    update_user_assigned
                         Command used by manage.py to update the users total
                         post count. Use if upgrading to 0.1.7.
    email_outstanding_tickets
                         Script to be run independently of the webserver.
                         Script emails users a list of outstanding tickets that
                         they have created or been assigned. To be run on a
                         regular basis using a cron job or similar. Email
                         functionality has to be enabled.
    runserver            Runs the Flask development server i.e. app.run()
    shell                Runs a Python shell inside Flask application context.

optional arguments:
  -?, --help             show this help message and exit
```

1.3.2 Administration Config Panel

Options

For email configuration the following options are available. At a minimum you should configure *mail_server*, *mail_port*, *mail_username* and *mail_password*.

For more information regarding these settings see the documentation for Flask-Mail.

class flicket_admin.models.flicket_config.**FlicketConfig**(**kwargs)

Server configuration settings editable by administrators only via the administration page */flicket_admin/config/*.

For email configuration settings see https://flask-mail.readthedocs.io/en/latest/ for more information.

Parameters

- **mail_server** (*str*) – example: *smtp.yourcompany.com*.
- **mail_port** (*int*) – example: *567*
- **mail_use_tls** (*bool*) – example: *true*
- **mail_use_ssl** (*bool*) – example: *false*
- **mail_debug** (*bool*) – example: *false*
- **mail_username** (*str*) – example: *flicket.admin*
- **mail_password** (*str*) –
- **mail_default_sender** (*str*) – example: *flicket.admin@yourcompany.com*
- **mail_max_emails** (*int*) –
- **mail_suppress_send** (*bool*) –
- **mail_ascii_attachments** (*bool*) –
- **application_title** (*str*) – Changes the default banner text from *Flicket*. Can typically be your company name.
- **posts_per_page** (*str*) – Maximum number of posts / topics displayed per page.
- **allowed_extensions** (*str*) – A comma delimited list of file extensions users are allowed to upload. DO NOT include the . before the extension letter.
- **ticket_upload_folder** (*str*) – The folder used for file uploads.
- **base_url** (*str*) – The sites base url. This is used to resolve urls for emails and links. Broken links are probably a result of not setting this value.
- **csv_dump_limit** (*str*) – The maximum number of rows exported to csv.
- **change_category** (*bool*) – Enable/disable change category.
- **change_category_only_admin_or_super_user** (*bool*) – Only admins or super users can change category.

static extension_allowed(*filename*)
Validates extension of a given filename and returns True if valid. Otherwise False. :param filename: :return:

static valid_extensions()
Returns a list of valid extensions. :return: list()

1.4 Exporting / Importing Flicket Users

1.4.1 Exporting

If you need to export the users from the Flicket database you can run the following command:

python manage.py export_users

This will output a json file formatted thus:

```
[
    {
        'username': 'jblogs',
        'name': 'Joe Blogs',
```

(continues on next page)

(continued from previous page)

```
        'email':'jblogs@email.com',
        'password': 'bcrypt_encoded_string'
    }
]
```

1.4.2 Importing

If you need to import users run the following command:

    python manage.py import_users

The file has to formatted as shown in the Exporting example.

1.5 Installing A Webserver

Currently the documentation will only describe how to install and configure the Apache webserver on Windows since this can be a bit trickier than on Linux. However, some of the steps here can also be used in Linux.

The instructions provided are for use with Python and Apache. You must ensure both Python and Apache have been compiled with the same version of Visual Studio. Also, Python and Apache must both be compiled for the same CPU architecture (x86 x64).

Also, the paths defined in this guide can be changed. You can by all means use different paths but you should try and and get the webserver running with the settings defined herein first.

1.5.1 Apache - Windows

Prior to installing a webserver you should confirm that flicket is working correctly by running the developement webserver as described in the *Installation* instructions.

Install mod_wsgi

Download the applicable *mod_wsgi* whl for your flavour of Apache and Python from the Unofficial Windows Binaries for Python Extension Packages page. For example, if you have *Python 3.6 x64* and *Apache 2.4 x64* you would get the whl *mod_wsgi-4.6.5+ap24vc14-cp36-cp36m-win_amd64.whl*.

Whilst **active in your flicket virtual environment** install *mod_wsgi*:

```
pip install <path_to_download>mod_wsgi-4.6.5+ap24vc14-cp36-cp36m-win_amd64.whl
```

Installing Apache

Download Apache compiled with VC14 from the apache lounge.

Unzip the apache folder to your *c:* directory. You should end up with a folder structure like this:

```
C:\Apache24
    C:\Apache24\bin
    C:\Apache24\cgi-bin
    ...
```

Open the file *C:\Apache24\conf\httpd.conf* in a text editor like notepad++.

Ensure SRVROOT is pointing to the correct folder:

```
SRVROOT "C:\Apache24"
```

Uncomment mod_version line:

```
LoadModule version_module modules/mod_version.so
```

Add the following lines (put these after the other LoadModule declarations):

```
LoadModule wsgi_module "<path_to_your_virtualenv>/lib/site-packages/mod_wsgi/server/
↪mod_wsgi.cp36-win_amd64.pyd"
WSGIPythonHome "<path_to_your_virtualenv>"
```

Uncomment the vhosts line:

```
Include conf/extra/httpd-vhosts.conf
```

Edit the file *C:\Apache24\conf\extra\httpd-vhosts.conf*.

Comment out the existing configurations lines by prefixing with a # (good reference for future troubleshooting).

Add the following:

```
<VirtualHost *:8000>

    ServerName <ip_address or hostname>
    ServerAlias <ip_address or hostname>
    ServerAdmin <your_email@there.com>

    DocumentRoot C:\Apache24\htdocs

    <Directory C:\Apache24\htdocs>
    <IfVersion < 2.4>
        Order allow,deny
        Allow from all
    </IfVersion>
    <IfVersion >= 2.4>
        Require all granted
    </IfVersion>
    </Directory>

    WSGIScriptAlias / <path_to_flicket>run.wsgi
    # Make sure to enable so that the API requests will work.
    WSGIPassAuthorization On

    <Directory <path_to_flicket>>
    <IfVersion < 2.4>
        Order allow,deny
        Allow from all
    </IfVersion>
    <IfVersion >= 2.4>
        Require all granted
    </IfVersion>
    </Directory>

</VirtualHost>
```

Edit the file *run.wsgi* so that the path points to your Flicket virtual environment.

Register Apache As A Service

Navigate to the Apache folder and register the service with name *Apache HTTP Server*:

```
cd "C:\Apache24\bin"
httpd.exe -k install -n "Apache HTTP Server"
```

Start Apache

To start the service use either Windows Serivce Manage and start the service *Apache HTTP Server* or from the command prompt whilst in the folder *c:\Apache24\bin*:

```
httpd -k start -n "Apache HTTP Server"
```

Flicket should now be available in your browser by accessing http:\<ip_address or hostname>:8000

Troubleshooting

To troubleshoot problems starting the apache service or accessing the webpage you should start by reading your Apache installations log files normally located in *c:Apache24logs*.

1.6 Adding Additional Languages

Flicket now supports additional languages through the use of Flask Babel. To add an additional local:

- Edit *SUPPORTED_LANGUAGES* in *config.py* and add an additional entry to the dictionary. For example: *{'en': 'English', 'fr': 'Francais', 'de': 'German'}*
- Whilst in the project root directory you now need to initialise the new language to generate a template file for it.

```
pybabel init -i messages.pot -d application/translations -l de
```

- In the folder *application/translations* there should now be a new folder *de*.
- Edit the file *messages.po* in that folder. For example:

```
msgid "403 Error - Forbidden"
msgstr "403 Error - Verboten"
```

- Compile the translations for use:

```
pybabel compile -d application/translations
```

- If any python or html text strings have been newly tagged for translation run:

```
pybabel extract -F babel.cfg -o messages.pot .
```

- To get the new translations added to the .po files:

```
pybabel update -i messages.pot -d application/translations
```

1.7 Flicket - FAQ

1.7.1 What is Flicket?

Flicket is a simple open source ticketing system driven by the python flask web micro framework.

Flicket also uses the following python packages:

> alembic, bcrypt, flask-admin, flask-babel, flask-login, flask-migrate, flask-principal, flask-sqlalchemy, flask-script, flask-wtf, jinja2, Markdown, WTForms

See *README.rst* for full requirements.

Licensing

For licensing see *LICENSE.md*

1.7.2 Tickets

General

1. How do I create a ticket?

   Select 'create ticket' from the Flicket pull down menu.

2. How do I assign a ticket?

   Scenario: You have raised a ticket and you know to whom the ticket should be assigned.

   Navigate to [flicket home page](/flicket/) and select the ticket you wish to assign. Within the ticket page is a button to *assign* ticket.

3. How do I release a ticket?

   Scenario: You have been assigned a ticket but the ticket isn't your responsibility to complete or you are unable to for another reason.

   Navigate to [flicket home page](/flicket/) and select the ticket to which you have been assigned. Within the ticket page is a button to *release* the ticket from your ticket list.

4. How do I close a ticket?

   Scenario: The ticket has been resolved to your satisfaction and you want to close the ticket.

   Navigate to [flicket home page](/flicket/) and select the ticket which you would like to close. Within the ticket page is a button to *replay and close* the ticket.

   Only the following persons can close a ticket: * Administrators. * The user which has been assigned the ticket. * The original creator of the ticket.

   > You may *claim* the ticket so that you may close it.

5. What is markdown?

   > Markdown is a lightweight markup language with plain text formatting syntax.
   >
   > The text contents of a ticket can be made easier to read by employing markdown syntax.

Searching

The ticket main page can be filtered to show only results of a specific interest to you. Tickets can be filtered by department, category, user and a text string.

Departments

Note: Only administrators or super users can add / edit or delete departments.

1. How do I add new departments?

   Navigate to Departments via the menu bar and use the add departments form.

2. How do I edit departments?

   Navigate to [departments](/flicket/departments/) and select the edit link against the department name.

3. How do I delete departments?

   Navigate to [departments](/flicket/departments/) and select the remove link against the department name. This is represented with a cross.

Categories

Note: Only administrators or super users can add / edit or delete categories.

1. How do I add categories?

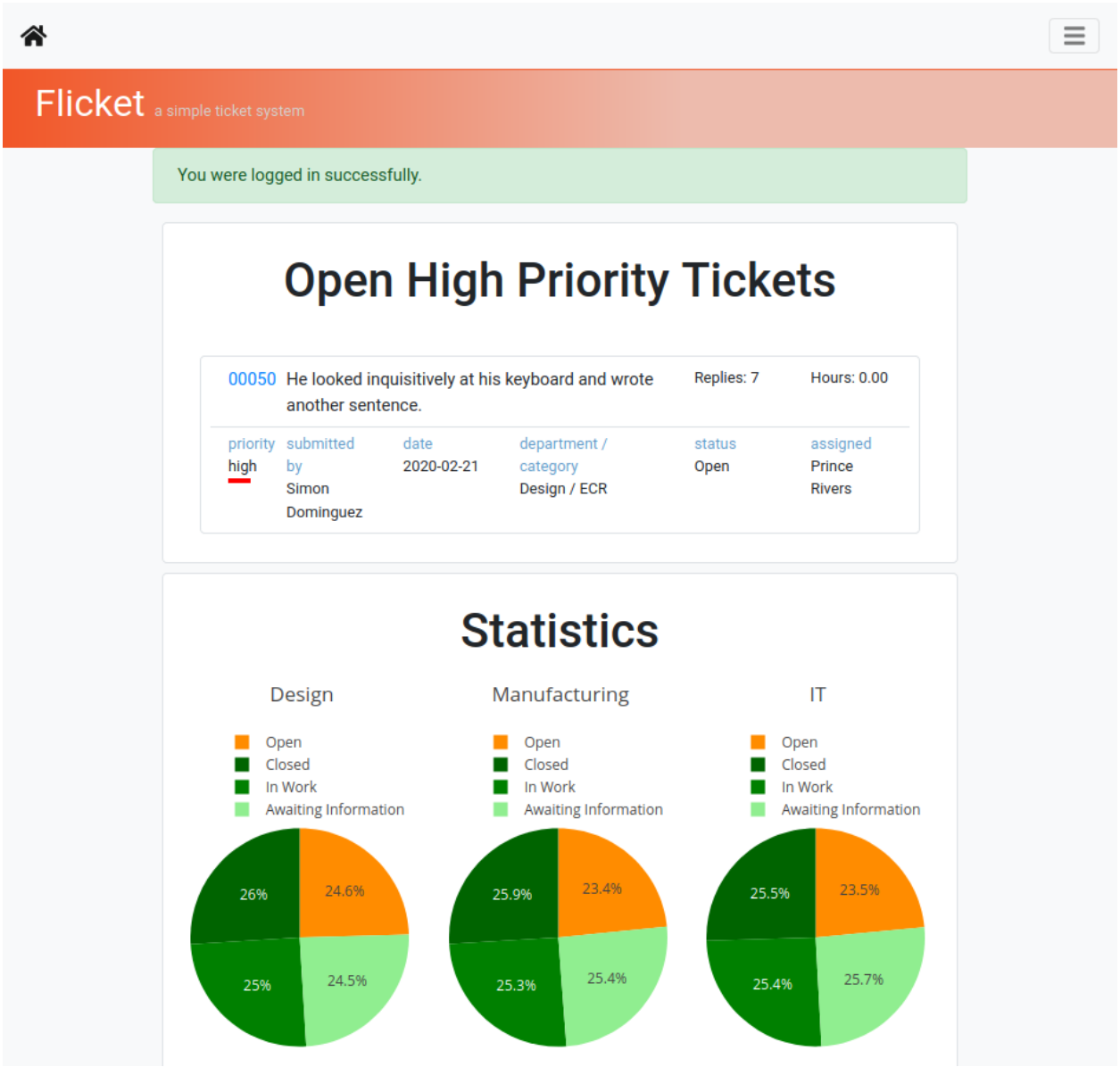
   Navigate to [departments](/flicket/departments/) and select the link to add categories against the appropriate department name.

1. How do I edit categories?

   Navigate to [departments](/flicket/departments/) and select the link to add categories against the appropriate department name.

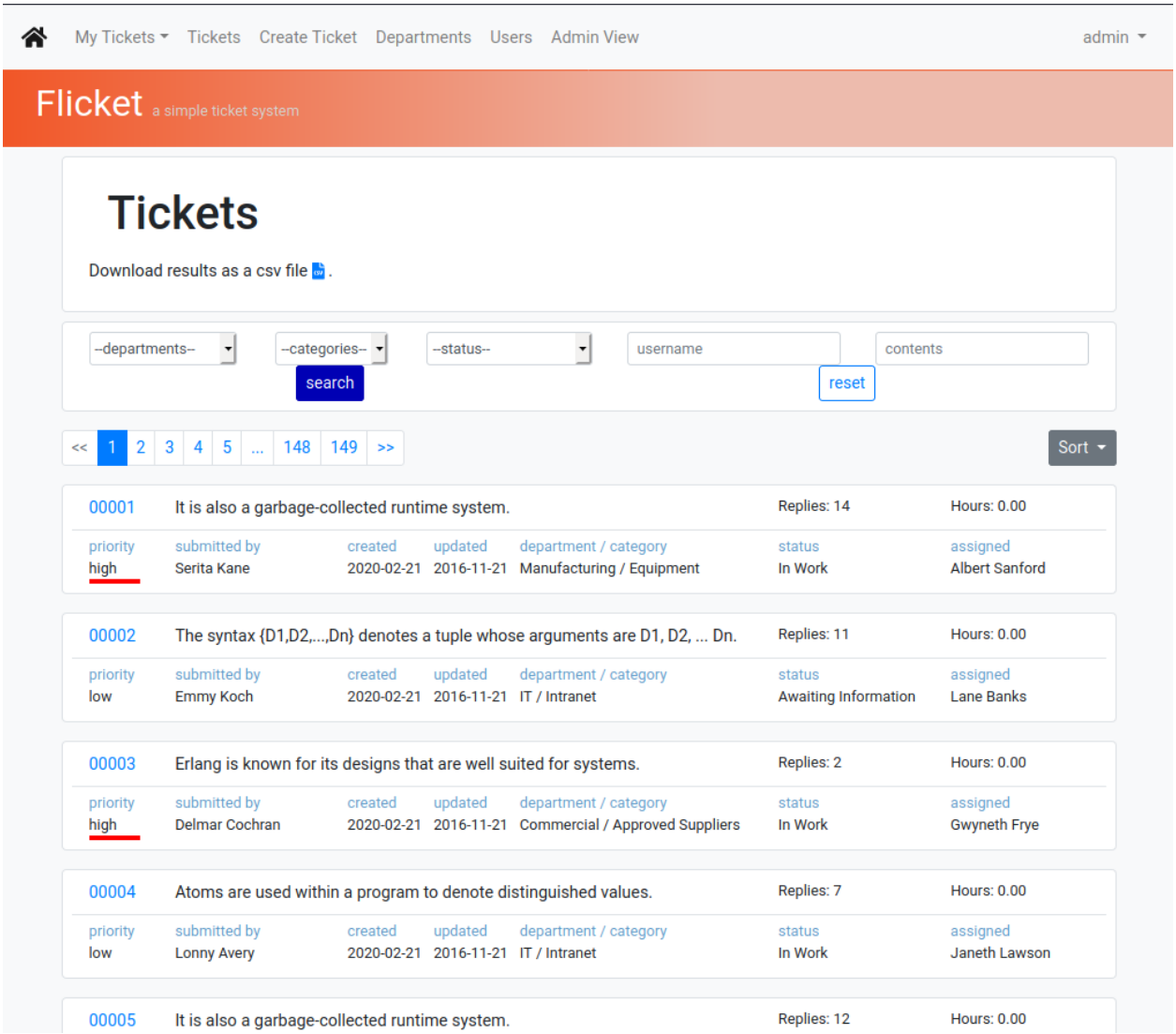
1.8 Screenshots

1.8.1 Home Page

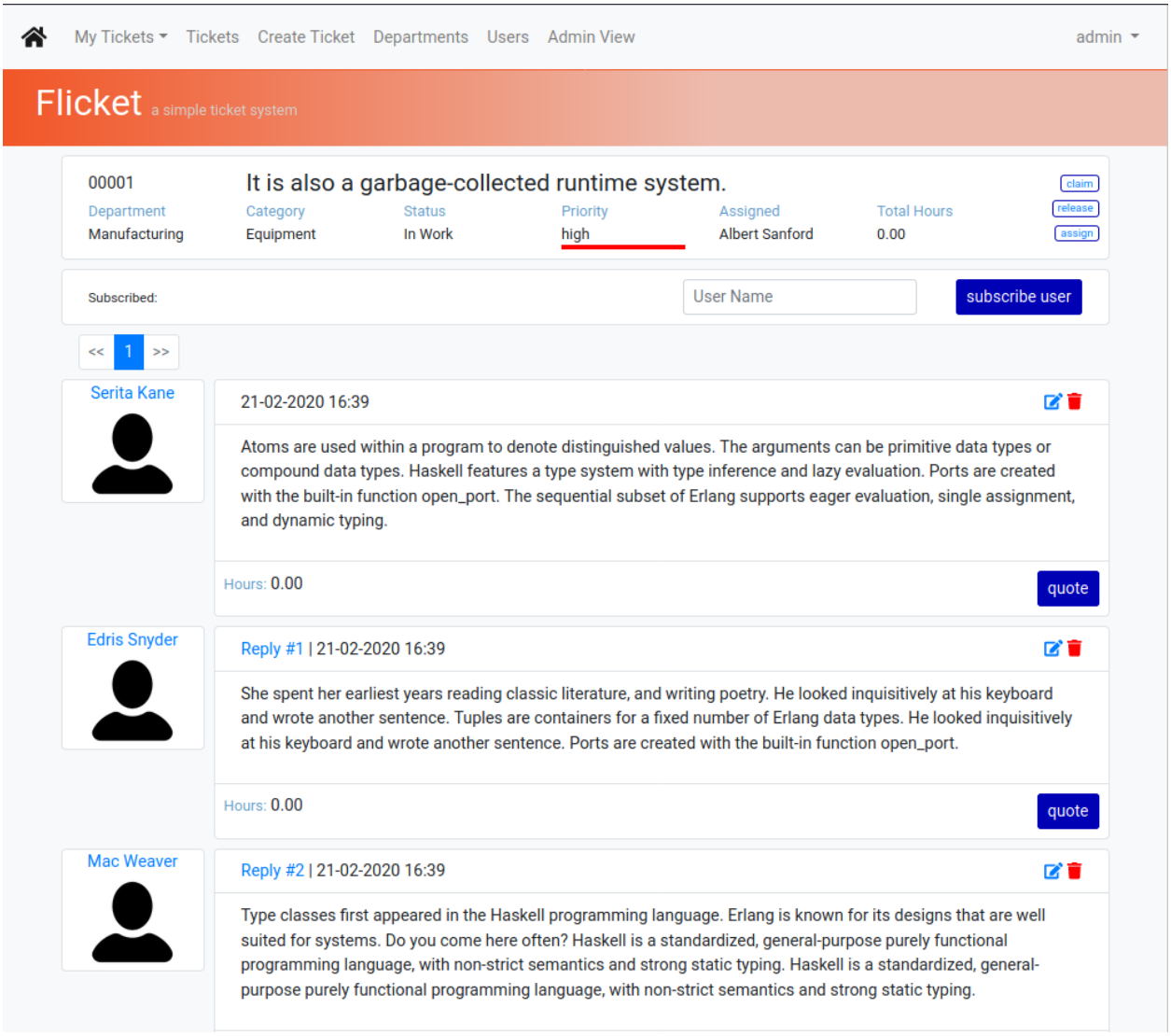


1.8.2 Tickets

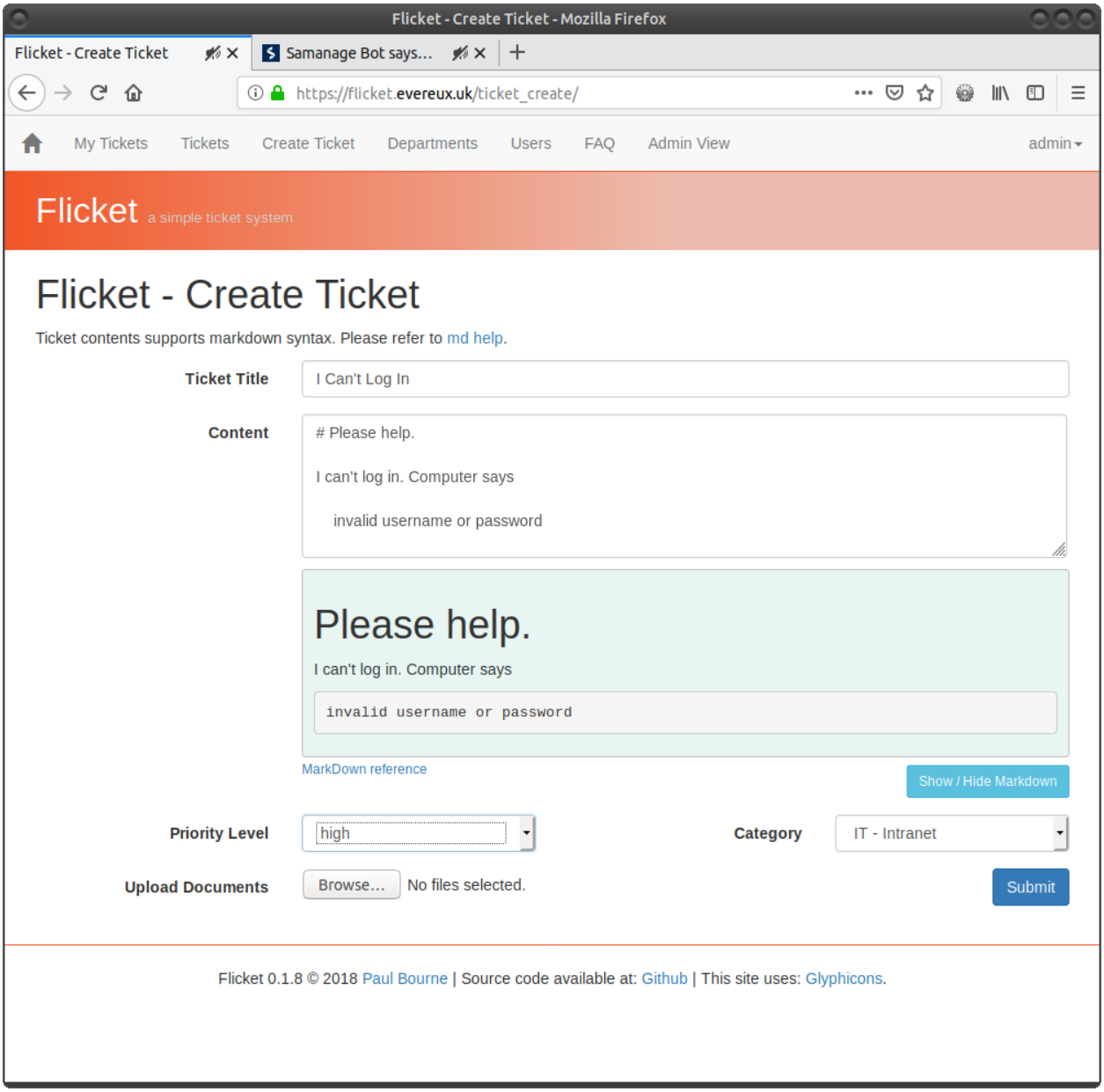
All tickets.


My Tickets
Tickets
Create Ticket
Departments
Users
Admin View
admin
Flicket
a simple ticket system
Tickets
Download results as a csv file .
--departments--
--categories--
--status--
username
contents
search
reset
<< 1 2 3 4 5 ... 148 149 >>
Sort
00001 It is also a garbage-collected runtime system. Replies: 14 Hours: 0.00
priority high
submitted by Serita Kane
created 2020-02-21
updated 2016-11-21
department / category Manufacturing / Equipment
status In Work
assigned Albert Sanford
00002 The syntax {D1,D2,...,Dn} denotes a tuple whose arguments are D1, D2, ... Dn. Replies: 11 Hours: 0.00
priority low
submitted by Emmy Koch
created 2020-02-21
updated 2016-11-21
department / category IT / Intranet
status Awaiting Information
assigned Lane Banks
00003 Erlang is known for its designs that are well suited for systems. Replies: 2 Hours: 0.00
priority high
submitted by Delmar Cochran
created 2020-02-21
updated 2016-11-21
department / category Commercial / Approved Suppliers
status In Work
assigned Gwyneth Frye
00004 Atoms are used within a program to denote distinguished values. Replies: 7 Hours: 0.00
priority low
submitted by Lonny Avery
created 2020-02-21
updated 2016-11-21
department / category IT / Intranet
status In Work
assigned Janeth Lawson
00005 It is also a garbage-collected runtime system. Replies: 12 Hours: 0.00

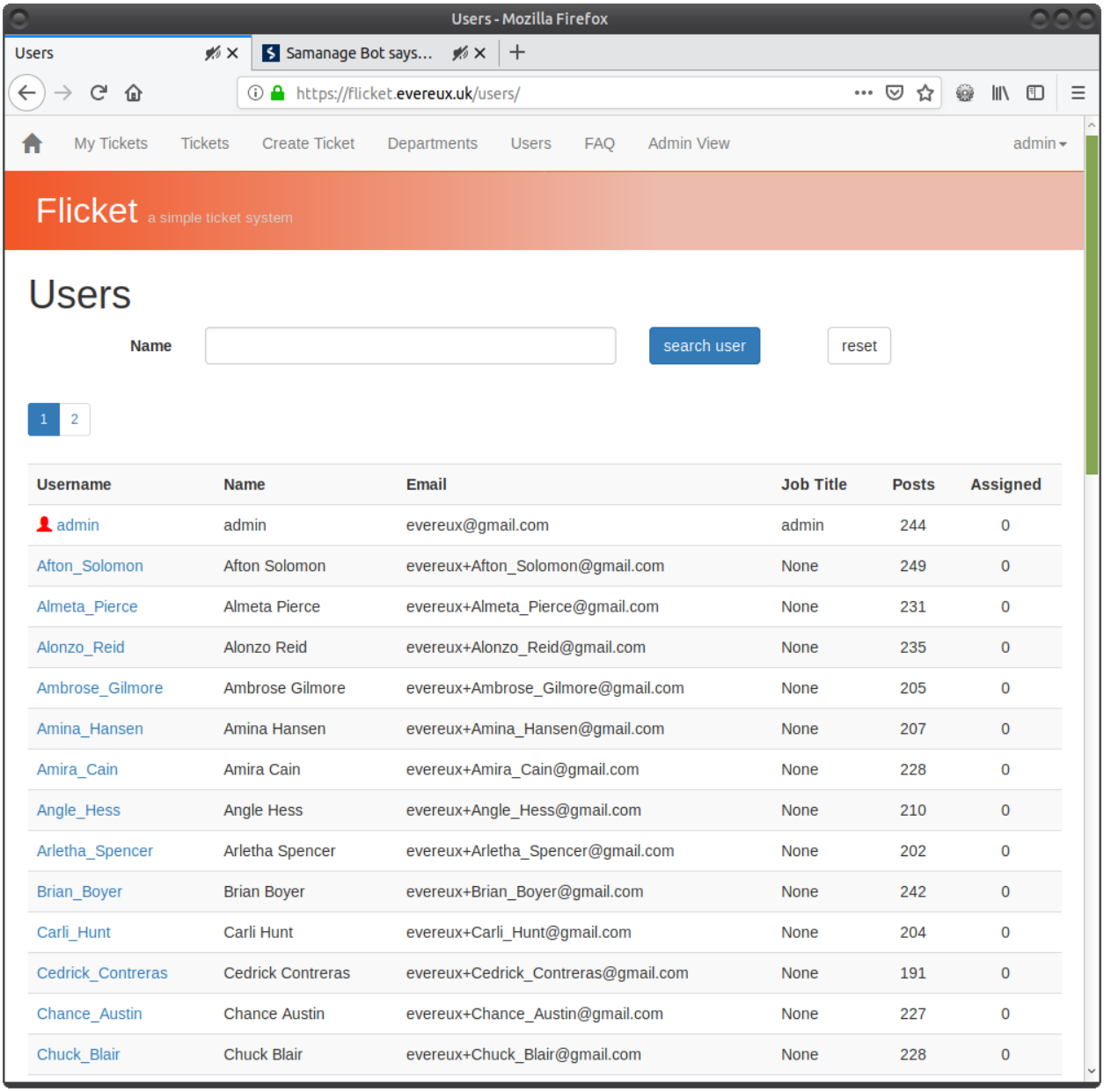
1.8.3 View Ticket



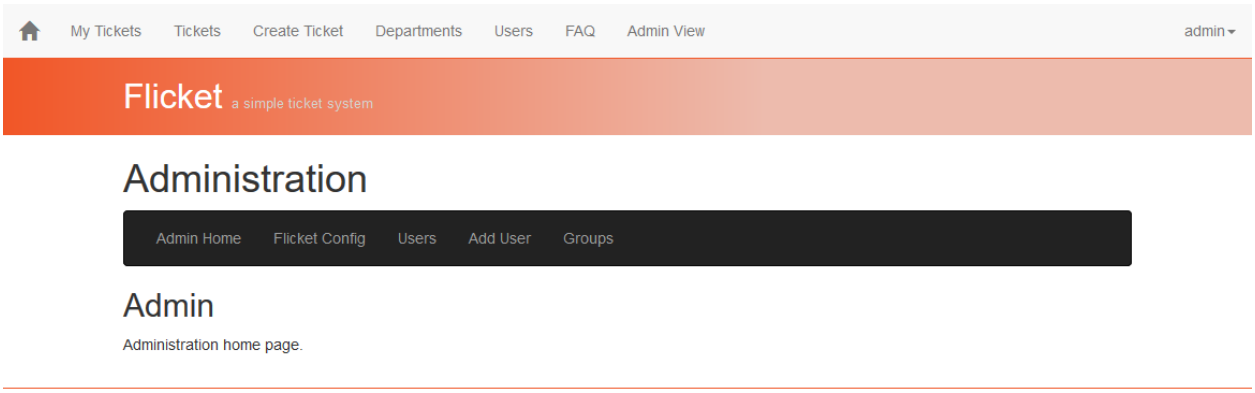
1.8.4 Create Ticket



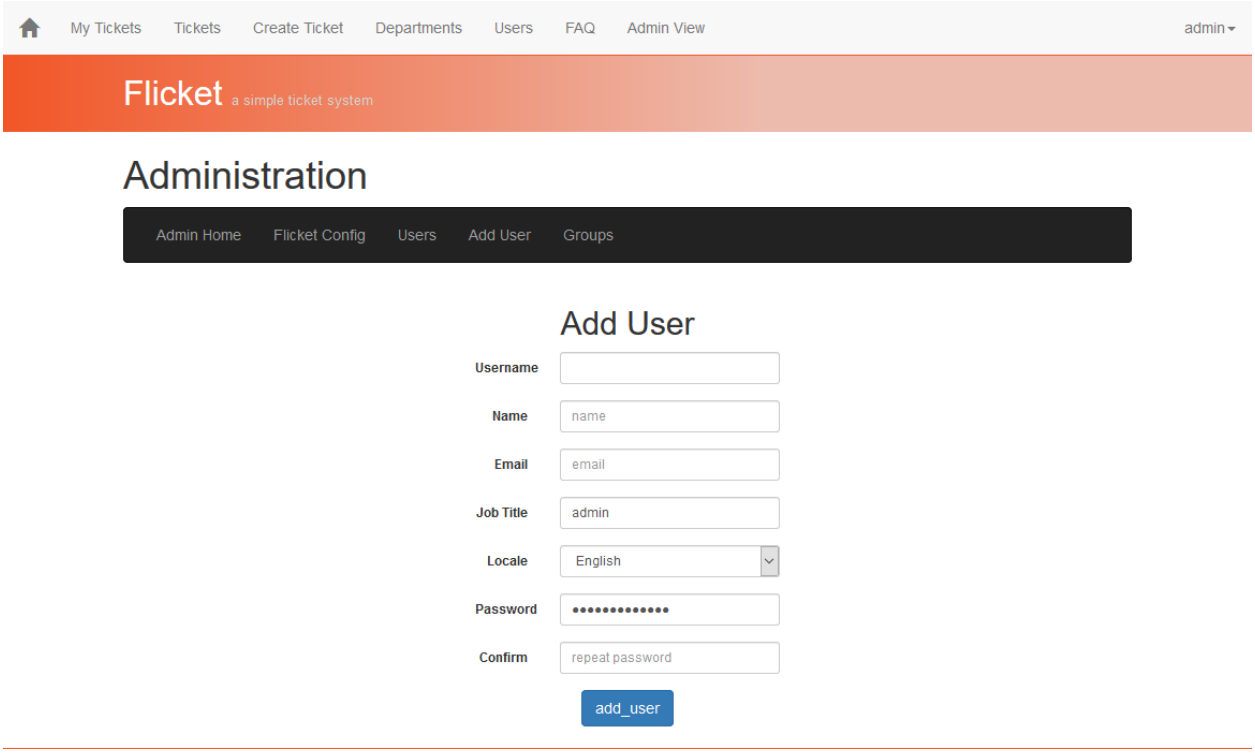
1.8.5 Users


Users - Mozilla Firefox
Users
Samanage Bot says...
https://flicket.evereux.uk/users/
My Tickets
Tickets
Create Ticket
Departments
Users
FAQ
Admin View
admin
Flicket
a simple ticket system
Users
Name
search user
reset
1
2
Username
Name
Email
Job Title
Posts
Assigned
admin
admin
evereux@gmail.com
admin
244
0
Afton_Solomon
Afton Solomon
evereux+Afton_Solomon@gmail.com
None
249
0
Almeta_Pierce
Almeta Pierce
evereux+Almeta_Pierce@gmail.com
None
231
0
Alonzo_Reid
Alonzo Reid
evereux+Alonzo_Reid@gmail.com
None
235
0
Ambrose_Gilmore
Ambrose Gilmore
evereux+Ambrose_Gilmore@gmail.com
None
205
0
Amina_Hansen
Amina Hansen
evereux+Amina_Hansen@gmail.com
None
207
0
Amira_Cain
Amira Cain
evereux+Amira_Cain@gmail.com
None
228
0
Angle_Hess
Angle Hess
evereux+Angle_Hess@gmail.com
None
210
0
Arletha_Spencer
Arletha Spencer
evereux+Arletha_Spencer@gmail.com
None
202
0
Brian_Boyer
Brian Boyer
evereux+Brian_Boyer@gmail.com
None
242
0
Carli_Hunt
Carli Hunt
evereux+Carli_Hunt@gmail.com
None
204
0
Cedrick_Contreras
Cedrick Contreras
evereux+Cedrick_Contreras@gmail.com
None
191
0
Chance_Austin
Chance Austin
evereux+Chance_Austin@gmail.com
None
227
0
Chuck_Blair
Chuck Blair
evereux+Chuck_Blair@gmail.com
None
228
0

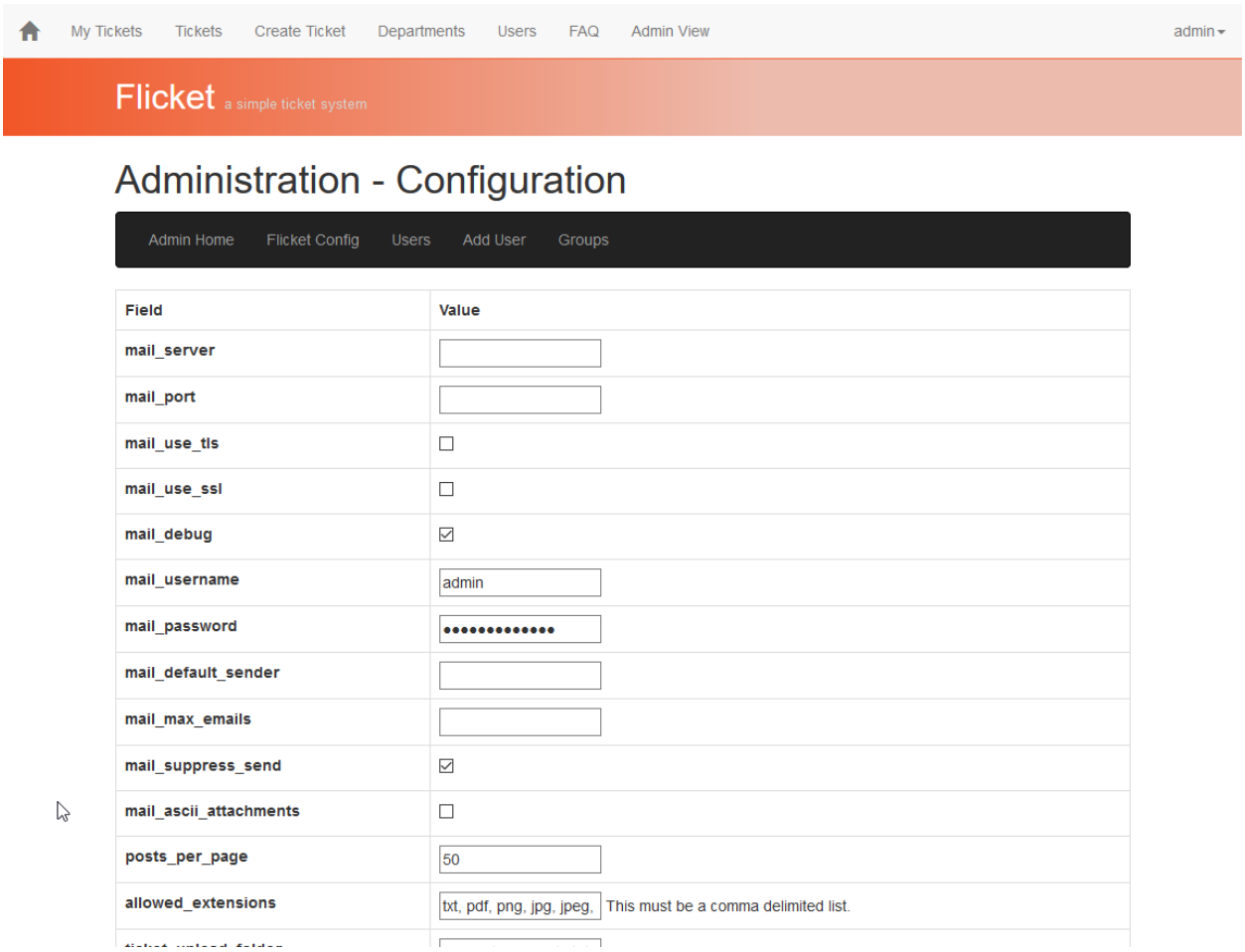
1.8.6 Admin Panel



1.8.7 Admin Panel - Add User



1.8.8 Admin Panel - Configuration



1.9 API

1.9.1 Authentication / Tokens

Get Token

The user will need to provide their username and password to retrieve an authentication token. The authentication token is required to access all other parts of the API.

```
# example using httpie
http --auth <username>:<password> POST http://localhost:5000/flicket-api/tokens
```

Response

```
HTTP/1.0 200 OK
Content-Length: 50
Content-Type: application/json
Date: Sat, 29 Sep 2018 14:01:00 GMT
Server: Werkzeug/0.14.1 Python/3.6.5
```

(continues on next page)

(continued from previous page)

```
{
    "token": "<token>"
}
```

Delete Token

```
# example using httpie
http DELETE http://localhost:5000/flicket-api/tokens "Authorization: Bearer <token>"
```

Responds

```
HTTP/1.0 204 NO CONTENT
Content-Length: 0
Content-Type: text/html; charset=utf-8
Date: Sat, 29 Sep 2018 14:13:19 GMT
Server: Werkzeug/0.14.1 Python/3.6.5
```

1.9.2 Users

Get User By ID

GET /flicket-api/user/(**int:** *user_id*)

Request

```
GET /flicket-api/user/1 HTTP/1.1
HOST: localhost:5000
Accept: application/json
Authorization: Bearer <token>
```

Response

```
HTTP/1.0 200 OK
Content-Length: 355
Content-Type: application/json
Date: Sun, 30 Jun 2019 14:15:37 GMT
Server: Werkzeug/0.14.1 Python/3.7.3

{
    "avatar": "http://127.0.0.1:5000/flicket/static/flicket_avatars/5bxk0qxt.jpg",
    "email": "evereux@gmail.com",
    "id": 1,
    "job_title": "admin",
    "links": {
        "self": "http://127.0.0.1:5000/flicket-api/user/1",
        "users": "http://127.0.0.1:5000/flicket-api/users/"
    },
    "name": "admin",
    "total_posts": 12505,
    "username": "admin"
}
```

Get Users

GET /flicket-api/users/

Request

```
GET /flicket-api/users/ HTTP/1.1
HOST: localhost:5000
Accept: application/json
Authorization: Bearer <token>
```

Response

```
HTTP/1.0 200 OK
Content-Length: 355
Content-Type: application/json
Date: Sun, 30 Jun 2019 14:15:37 GMT
Server: Werkzeug/0.14.1 Python/3.7.3

{
    "_links": {
        "next": null,
        "prev": null,
        "self": "http://localhost:5000/flicket-api/users/?page=1&per_page=50"
    },
    "_meta": {
        "page": 1,
        "per_page": 50,
        "total_items": 48,
        "total_pages": 1
    },
    "items": [
        {
            "avatar": "http://localhost:5000/flicket/static/flicket_avatars/__
↪default_profile.png",
            "email": "evereux@gmail.com",
            "id": 1,
            "job_title": "admin",
            "links": {
                "self": "http://localhost:5000/flicket-api/user/1",
                "users": "http://localhost:5000/flicket-api/users/"
            },
            "name": "admin",
            "total_posts": 6381,
            "username": "admin"
        },
        {
            "avatar": "http://localhost:5000/flicket/static/flicket_avatars/__
↪default_profile.png",
            "email": "admin@localhost",
            "id": 2,
            "job_title": "unknown",
            "links": {
                "self": "http://localhost:5000/flicket-api/user/2",
                "users": "http://localhost:5000/flicket-api/users/"
            },
            "name": "notification",
            "total_posts": 6445,
            "username": "notification"
```

(continues on next page)

(continued from previous page)

```
        },
    ]
}
```

1.9.3 Tickets

Get Ticket By ID

GET /flicket-api/ticket/(int: *ticket_id*)

Request

```
GET /flicket-api/ticket/1 HTTP/1.1
HOST: localhost:5000
Accept: application/json
Authorization: Bearer <token>
```

Response

```
HTTP/1.0 200 OK
Content-Length: 1835
Content-Type: application/json
Date: Sun, 30 Jun 2019 14:15:37 GMT
Server: Werkzeug/0.14.1 Python/3.7.3

{
    "assigned_id": 7,
    "category_id": 1,
    "content": "She spent her earliest years reading classic literature, and␣
↪writing poetry. Haskell
    features a type system with type inference and lazy evaluation. They are␣
↪written as strings of
    consecutive alphanumeric characters, the first character being lowercase.␣
↪Tuples are containers for
    a fixed number of Erlang data types. Erlang is a general-purpose, concurrent,␣
↪functional programming
    language. Where are my pants? He looked inquisitively at his keyboard and␣
↪wrote another sentence. The
    arguments can be primitive data types or compound data types. It is also a␣
↪garbage-collected runtime
    system. He looked inquisitively at his keyboard and wrote another sentence.␣
↪Do you come here often?
    Ports are created with the built-in function open_port. He looked␣
↪inquisitively at his keyboard and
    wrote another sentence. Haskell features a type system with type inference␣
↪and lazy evaluation.",
    "date_added": "Sun, 23 Jun 2019 18:25:36 GMT",
    "date_modified": null,
    "id": 1,
    "links": {
        "assigned": "http://localhost:5000/flicket-api/user/7",
        "category": "http://localhost:5000/flicket-api/category/1",
        "histories": "http://localhost:5000/flicket-api/histories/?topic_id=1",
        "modified_by": null,
        "priority": "http://localhost:5000/flicket-api/priority/3",
```

(continues on next page)

(continued from previous page)


```
        "self": "http://localhost:5000/flicket-api/ticket/1",
        "started_ny": "http://localhost:5000/flicket-api/user/12",
        "status": "http://localhost:5000/flicket-api/status/2",
        "subscribers": "http://localhost:5000/flicket-api/subscriptions/1/",
        "tickets": "http://localhost:5000/flicket-api/tickets/"
    },
    "modified_id": null,
    "started_id": 12,
    "status_id": 2,
    "ticket_priority_id": 3,
    "title": "He looked inquisitively at his keyboard and wrote another sentence."
}
```


Get Tickets

GET /flicket-api/tickets/

Request

```
GET /flicket-api/tickets/ HTTP/1.1
HOST: localhost:5000
Accept: application/json
Authorization: Bearer <token>
```

Response


```
HTTP/1.0 200 OK
Content-Length: 2244
Content-Type: application/json
Date: Sun, 30 Jun 2019 14:15:37 GMT
Server: Werkzeug/0.14.1 Python/3.7.3

{
    "_links": {
        "next": "http://localhost:5000/flicket-api/tickets/2/?per_page=1",
        "prev": null,
        "self": "http://localhost:5000/flicket-api/tickets/1/?per_page=1"
    },
    "_meta": {
        "page": 1,
        "per_page": 1,
        "total_items": 10000,
        "total_pages": 10000
    },
    "items": [
        {
            "assigned_id": 7,
            "category_id": 1,
            "content": "She spent her earliest years reading classic␣
↪literature, and writing poetry. Haskell
            features a type system with type inference and lazy evaluation.␣
↪They are written as strings of
            consecutive alphanumeric characters, the first character being␣
↪lowercase. Tuples are containers
            for a fixed number of Erlang data types. Erlang is a general-
↪purpose, concurrent, functional
```


(continues on next page)

(continued from previous page)


```
            programming language. Where are my pants? He looked␣
↪inquisitively at his keyboard and wrote
            another sentence. The arguments can be primitive data types or␣
↪compound data types. It is also a
            garbage-collected runtime system. He looked inquisitively at his␣
↪keyboard and wrote another
            sentence. Do you come here often? Ports are created with the␣
↪built-in function open_port. He
            looked inquisitively at his keyboard and wrote another sentence.␣
↪Haskell features a type system
            with type inference and lazy evaluation.",
            "date_added": "Sun, 23 Jun 2019 18:25:36 GMT",
            "date_modified": null,
            "id": 1,
            "links": {
                "assigned": "http://localhost:5000/flicket-api/user/7",
                "category": "http://localhost:5000/flicket-api/category/1",
                "histories": "http://localhost:5000/flicket-api/histories/?
↪topic_id=1",
                "modified_by": null,
                "priority": "http://localhost:5000/flicket-api/priority/3",
                "self": "http://localhost:5000/flicket-api/ticket/1",
                "started_ny": "http://localhost:5000/flicket-api/user/12",
                "status": "http://localhost:5000/flicket-api/status/2",
                "subscribers": "http://localhost:5000/flicket-api/
↪subscriptions/1/",
                "tickets": "http://localhost:5000/flicket-api/tickets/"
            },
            "modified_id": null,
            "started_id": 12,
            "status_id": 2,
            "ticket_priority_id": 3,
            "title": "He looked inquisitively at his keyboard and wrote␣
↪another sentence."
        }
    ]
}
```


Create Ticket

POST /flicket-api/tickets(str:title,str:content,int:category_id,int:ticket_priority_id)

Request

```
POST /flicket-api/tickets HTTP/1.1
HOST: localhost:5000
Accept: application/json
Authorization: Bearer <token>

{
    "title": "this is my ticket",
    "content": "this is my content",
    "category_id": 1,
    "ticket_priority_id": 1
}
```

Response

```
HTTP/1.0 201 CREATED
Content-Length: 903
Content-Type: application/json
Date: Fri, 28 Jun 2019 12:04:59 GMT
Location: http://localhost:5000/flicket-api/ticket/10001
Server: Werkzeug/0.14.1 Python/3.7.3

{
    "assigned_id": null,
    "category_id": 1,
    "content": "this is my content",
    "date_added": "Fri, 28 Jun 2019 13:04:59 GMT",
    "date_modified": null,
    "id": 10001,
    "links": {
        "assigned": null,
        "category": "http://localhost:5000/flicket-api/category/1",
        "histories": "http://localhost:5000/flicket-api/histories/?topic_id=10001
↪",
        "modified_by": null,
        "priority": "http://localhost:5000/flicket-api/priority/1",
        "self": "http://localhost:5000/flicket-api/ticket/10001",
        "started_ny": "http://localhost:5000/flicket-api/user/1",
        "status": "http://localhost:5000/flicket-api/status/1",
        "subscribers": "http://localhost:5000/flicket-api/subscriptions/10001/",
        "tickets": "http://localhost:5000/flicket-api/tickets/"
    },
    "modified_id": null,
    "started_id": 1,
    "status_id": 1,
    "ticket_priority_id": 1,
    "title": "this is my ticket"
}
```

1.9.4 Posts

Get Post By ID

GET /flicket-api/post/(int: *post_id*)

Request

```
GET /flicket-api/priority/1 HTTP/1.1
HOST: localhost:5000
Accept: application/json
Authorization: Bearer <token>
```

Response

```
HTTP/1.0 200 OK
Content-Length: 1231
Content-Type: application/json
Date: Sun, 30 Jun 2019 13:00:13 GMT
Server: Werkzeug/0.14.1 Python/3.7.3

{
```

(continues on next page)

(continued from previous page)


```
    "content": "The Galactic Empire is nearing completion of the Death Star, a␣
↪space station with the power
     to destroy entire planets. Initially ...",
    "data_added": "Sun, 05 May 2019 14:19:42 GMT",
    "date_modified": null,
    "id": 1,
    "links": {
        "created_by": "http://127.0.0.1:5000/flicket-api/user/15",
        "posts": "http://127.0.0.1:5000/flicket-api/posts/1/",
        "self": "http://127.0.0.1:5000/flicket-api/post/1"
    },
    "ticket_id": 1,
    "user_id": 15
}
```


Get Posts

Retrieve all posts associated to a ticket by ticket_id.

GET /flicket-api/posts/(int: *ticket_id*)/(int: *page*)/ Request

```
GET /flicket-api/posts/1/ HTTP/1.1
HOST: localhost:5000
Accept: application/json
Authorization: Bearer <token>
```

Response


```
HTTP/1.0 200 OK
Content-Length: 27640
Content-Type: application/json
Date: Sun, 30 Jun 2019 15:41:09 GMT
Server: Werkzeug/0.14.1 Python/3.7.3

{
    "_links": {
        "next": null,
        "prev": null,
        "self": "http://127.0.0.1:5000/flicket-api/posts/1/1/?per_page=50"
    },
    "_meta": {
        "page": 1,
        "per_page": 50,
        "total_items": 25,
        "total_pages": 1
    },
    "items": [
        {
            "content": "The Galactic Empire is nearing completion of the Death␣
↪Star, a space station with
            the power to destroy entire planets. Initially composing light-
↪hearted and irreverent works,
            he also wrote serious, sombre and religious pieces beginning in the␣
↪1930s. Erlang is known for
```


(continues on next page)

(continued from previous page)


```
            its designs that are well suited for systems. It is also a garbage-
↪collected runtime system.
            They are written as strings of consecutive alphanumeric characters,␣
↪the first character being
            lowercase. The sequential subset of Erlang supports eager evaluation,␣
↪single assignment, and
            dynamic typing. Tuples are containers for a fixed number of Erlang␣
↪data types. Tuples are
            containers for a fixed number of Erlang data types. The arguments can␣
↪be primitive data types
            or compound data types. Type classes first appeared in the Haskell␣
↪programming language. The
            arguments can be primitive data types or compound data types.",
            "data_added": "Sun, 05 May 2019 14:19:42 GMT",
            "date_modified": null,
            "id": 1,
            "links": {
                "created_by": "http://127.0.0.1:5000/flicket-api/user/15",
                "posts": "http://127.0.0.1:5000/flicket-api/posts/1/",
                "self": "http://127.0.0.1:5000/flicket-api/post/1"
            },
            "ticket_id": 1,
            "user_id": 15
        }
    ]
}
```


1.9.5 Departments

Get Department by ID

GET /flicket-api/department/(int: *department_id*)

Request

```
GET /flicket-api/department/1 HTTP/1.1
HOST: localhost:5000
Accept: application/json
Authorization: Bearer <token>
```

Response


```
HTTP/1.0 200 OK
Content-Length: 191
Content-Type: application/json
Date: Sun, 30 Jun 2019 12:37:21 GMT
Server: Werkzeug/0.14.1 Python/3.7.3

{
    "department": "Design",
    "id": 1,
    "links": {
        "departments": "http://127.0.0.1:5000/flicket-api/departments/",
        "self": "http://127.0.0.1:5000/flicket-api/department/1"
    }
}
```

Get Departments

GET /flicket-api/departments/

Request

```
GET /flicket-api/departments/ HTTP/1.1
HOST: localhost:5000
Accept: application/json
Authorization: Bearer <token>
```

Response

```
HTTP/1.0 200 OK
Content-Length: 2307
Content-Type: application/json
Date: Sun, 30 Jun 2019 12:40:21 GMT
Server: Werkzeug/0.14.1 Python/3.7.3

{
    "_links": {
        "next": null,
        "prev": null,
        "self": "http://127.0.0.1:5000/flicket-api/departments/?page=1&per_page=50
↪"
    },
    "_meta": {
        "page": 1,
        "per_page": 50,
        "total_items": 9,
        "total_pages": 1
    },
    "items": [
        {
            "department": "Commercial",
            "id": 6,
            "links": {
                "departments": "http://127.0.0.1:5000/flicket-api/departments/",
                "self": "http://127.0.0.1:5000/flicket-api/department/6"
            }
        },
        {
            "department": "Design",
            "id": 1,
            "links": {
                "departments": "http://127.0.0.1:5000/flicket-api/departments/",
                "self": "http://127.0.0.1:5000/flicket-api/department/1"
            }
        },
        {
            "department": "Human Resources",
            "id": 5,
            "links": {
                "departments": "http://127.0.0.1:5000/flicket-api/departments/",
                "self": "http://127.0.0.1:5000/flicket-api/department/5"
            }
        },
        {
            "department": "IT",
```

(continues on next page)

(continued from previous page)

```
            "id": 3,
            "links": {
                "departments": "http://127.0.0.1:5000/flicket-api/departments/",
                "self": "http://127.0.0.1:5000/flicket-api/department/3"
            }
        }
    ]
}
```

Create Department

POST http://localhost:5000/flicket-api/departments(**str:** *department*)

Request

```
POST /flicket-api/departments HTTP/1.1
HOST: localhost:5000
Accept: application/json
Authorization: Bearer <token>

{
    "department": "new department"
}
```

Response

1.9.6 Priorities

Get Priority By ID

GET /flicket-api/priority/(**int:** *priority_id*)

Request

```
GET /flicket-api/priority/1 HTTP/1.1
HOST: localhost:5000
Accept: application/json
Authorization: Bearer <token>
```

Response

```
HTTP/1.0 200 OK
Content-Length: 182
Content-Type: application/json
Date: Sun, 30 Jun 2019 14:15:37 GMT
Server: Werkzeug/0.14.1 Python/3.7.3

{
    "id": 1,
    "links": {
        "priorities": "http://127.0.0.1:5000/flicket-api/priorities/",
        "self": "http://127.0.0.1:5000/flicket-api/priority/1"
    },
    "priority": "low"
}
```

Get Priorities

GET /flicket-api/priorities/

Request

```
GET /flicket-api/priorities/ HTTP/1.1
HOST: localhost:5000
Accept: application/json
Authorization: Bearer <token>
```

Response

```
HTTP/1.0 200 OK
Content-Length: 903
Content-Type: application/json
Date: Sun, 30 Jun 2019 12:34:06 GMT
Server: Werkzeug/0.14.1 Python/3.7.3

{
    "_links": {
        "next": null,
        "prev": null,
        "self": "http://127.0.0.1:5000/flicket-api/priorities/1/?per_page=50"
    },
    "_meta": {
        "page": 1,
        "per_page": 50,
        "total_items": 3,
        "total_pages": 1
    },
    "items": [
        {
            "id": 1,
            "links": {
                "priorities": "http://127.0.0.1:5000/flicket-api/priorities/",
                "self": "http://127.0.0.1:5000/flicket-api/priority/1"
            },
            "priority": "low"
        },
        {
            "id": 2,
            "links": {
                "priorities": "http://127.0.0.1:5000/flicket-api/priorities/",
                "self": "http://127.0.0.1:5000/flicket-api/priority/2"
            },
            "priority": "medium"
        },
        {
            "id": 3,
            "links": {
                "priorities": "http://127.0.0.1:5000/flicket-api/priorities/",
                "self": "http://127.0.0.1:5000/flicket-api/priority/3"
            },
            "priority": "high"
        }
    ]
}
```

1.9.7 Status

Get Status By ID

GET /flicket-api/status/(**int:** *status_id*)

Request

```
GET /flicket-api/status/1 HTTP/1.1
HOST: localhost:5000
Accept: application/json
Authorization: Bearer <token>
```

Response

```
HTTP/1.0 200 OK
Content-Length: 175
Content-Type: application/json
Date: Mon, 01 Jul 2019 11:17:00 GMT
Server: Werkzeug/0.14.1 Python/3.7.3

{
    "id": 1,
    "links": {
        "self": "http://127.0.0.1:5000/flicket-api/status/1",
        "statuses": "http://127.0.0.1:5000/flicket-api/statuses/"
    },
    "status": "Open"
}
```

Get Statuses

GET /flicket-api/statuses/

Request

```
GET /flicket-api/statuses/ HTTP/1.1
HOST: localhost:5000
Accept: application/json
Authorization: Bearer <token>
```

Response

```
HTTP/1.0 200 OK
Content-Length: 1114
Content-Type: application/json
Date: Mon, 01 Jul 2019 11:18:23 GMT
Server: Werkzeug/0.14.1 Python/3.7.3

{
    "_links": {
        "next": null,
        "prev": null,
        "self": "http://127.0.0.1:5000/flicket-api/departments/?page=1&per_page=50
↪"
    },
    "_meta": {
        "page": 1,
```

(continues on next page)

(continued from previous page)


```
        "per_page": 50,
        "total_items": 4,
        "total_pages": 1
    },
    "items": [
        {
            "id": 1,
            "links": {
                "self": "http://127.0.0.1:5000/flicket-api/status/1",
                "statuses": "http://127.0.0.1:5000/flicket-api/statuses/"
            },
            "status": "Open"
        },
        {
            "id": 2,
            "links": {
                "self": "http://127.0.0.1:5000/flicket-api/status/2",
                "statuses": "http://127.0.0.1:5000/flicket-api/statuses/"
            },
            "status": "Closed"
        },
        {
            "id": 3,
            "links": {
                "self": "http://127.0.0.1:5000/flicket-api/status/3",
                "statuses": "http://127.0.0.1:5000/flicket-api/statuses/"
            },
            "status": "In Work"
        },
        {
            "id": 4,
            "links": {
                "self": "http://127.0.0.1:5000/flicket-api/status/4",
                "statuses": "http://127.0.0.1:5000/flicket-api/statuses/"
            },
            "status": "Awaiting Information"
        }
    ]
}
```


1.9.8 Subscriptions

Get Subscription By ID

GET /flicket-api/subscription/(int: *subscription_id*)

Request

```
GET /flicket-api/subscription/1 HTTP/1.1
HOST: localhost:5000
Accept: application/json
Authorization: Bearer <token>
```

Response

```
HTTP/1.0 200 OK
Content-Length: 356
Content-Type: application/json
Date: Mon, 01 Jul 2019 11:21:57 GMT
Server: Werkzeug/0.14.1 Python/3.7.3

{
    "id": 1,
    "links": {
        "self": "http://127.0.0.1:5000/flicket-api/subscription/1",
        "subscriptions": "http://127.0.0.1:5000/flicket-api/subscriptions/",
        "ticket": "http://127.0.0.1:5000/flicket-api/ticket/10001",
        "user": "http://127.0.0.1:5000/flicket-api/user/1"
    },
    "ticket_id": 10001,
    "user_def": "admin",
    "user_id": 1
}
```

Get Subscriptions

Get all subscribers to ticket.

GET /flicket-api/subscriptions/(int: *ticket_id*)/

Request

```
GET /flicket-api/users/ HTTP/1.1
HOST: localhost:5000
Accept: application/json
Authorization: Bearer <token>
```

Response

```
HTTP/1.0 200 OK
Content-Length: 666
Content-Type: application/json
Date: Mon, 01 Jul 2019 11:27:12 GMT
Server: Werkzeug/0.14.1 Python/3.7.3

{
    "_links": {
        "next": null,
        "prev": null,
        "self": "http://127.0.0.1:5000/flicket-api/subscriptions/10001/1/?per_
↪page=50"
    },
    "_meta": {
        "page": 1,
        "per_page": 50,
        "total_items": 1,
        "total_pages": 1
    },
    "items": [
        {
            "id": 1,
            "links": {
```

(continues on next page)

(continued from previous page)

```
                "self": "http://127.0.0.1:5000/flicket-api/subscription/1",
                "subscriptions": "http://127.0.0.1:5000/flicket-api/subscriptions/
↪",
                "ticket": "http://127.0.0.1:5000/flicket-api/ticket/10001",
                "user": "http://127.0.0.1:5000/flicket-api/user/1"
            },
            "ticket_id": 10001,
            "user_def": "admin",
            "user_id": 1
        }
    ]
}
```

1.9.9 Uploads

Get Upload By ID

GET /flicket-api/upload/(**int:** *upload_id*)

Request

```
GET /flicket-api/upload/1 HTTP/1.1
HOST: localhost:5000
Accept: application/json
Authorization: Bearer <token>
```

Response

```
HTTP/1.0 200 OK
Content-Length: 415
Content-Type: application/json
Date: Mon, 01 Jul 2019 11:46:54 GMT
Server: Werkzeug/0.14.1 Python/3.7.3

{
    "filename": "ccv4ufb6.jpg",
    "id": 1,
    "image": "http://127.0.0.1:5000/flicket_uploads/ccv4ufb6.jpg",
    "links": {
        "post": "http://127.0.0.1:5000/flicket-api/post/276646",
        "self": "http://127.0.0.1:5000/flicket-api/upload/1",
        "ticket": null,
        "uploads": "http://127.0.0.1:5000/flicket-api/uploads/"
    },
    "original_filename": "photos-1.jpg",
    "post_id": 276646,
    "topic_id": null
}
```

Get Uploads

GET /flicket-api/uploads/

Request

```
GET /flicket-api/uploads/ HTTP/1.1
HOST: localhost:5000
Accept: application/json
Authorization: Bearer <token>
```

Response

```
HTTP/1.0 200 OK
Content-Length: 1231
Content-Type: application/json
Date: Mon, 01 Jul 2019 11:49:33 GMT
Server: Werkzeug/0.14.1 Python/3.7.3

{
    "_links": {
        "next": null,
        "prev": null,
        "self": "http://127.0.0.1:5000/flicket-api/uploads/1/?per_page=50"
    },
    "_meta": {
        "page": 1,
        "per_page": 50,
        "total_items": 2,
        "total_pages": 1
    },
    "items": [
        {
            "filename": "ccv4ufb6.jpg",
            "id": 1,
            "image": "http://127.0.0.1:5000/flicket_uploads/ccv4ufb6.jpg",
            "links": {
                "post": "http://127.0.0.1:5000/flicket-api/post/276646",
                "self": "http://127.0.0.1:5000/flicket-api/upload/1",
                "ticket": null,
                "uploads": "http://127.0.0.1:5000/flicket-api/uploads/"
            },
            "original_filename": "photos-1.jpg",
            "post_id": 276646,
            "topic_id": null
        },
        {
            "filename": "5w0hdo10.jpg",
            "id": 2,
            "image": "http://127.0.0.1:5000/flicket_uploads/5w0hdo10.jpg",
            "links": {
                "post": "http://127.0.0.1:5000/flicket-api/post/276677",
                "self": "http://127.0.0.1:5000/flicket-api/upload/2",
                "ticket": null,
                "uploads": "http://127.0.0.1:5000/flicket-api/uploads/"
            },
            "original_filename": "the_basta_rock_sunrise_4k-wallpaper-3554x1999.
↪jpg",
            "post_id": 276677,
            "topic_id": null
        }
    ]
}
```

PYTHON MODULE INDEX

f

flicket_admin, 5
flicket_api.views.departments, 26
flicket_api.views.posts, 24
flicket_api.views.priorities, 28
flicket_api.views.status, 29
flicket_api.views.subscriptions, 31
flicket_api.views.tickets, 21
flicket_api.views.tokens, 18
flicket_api.views.uploads, 33
flicket_api.views.users, 19

HTTP ROUTING TABLE

/flicket-api

GET /flicket-api/department/(int:department_id), 26
GET /flicket-api/departments/, 27
GET /flicket-api/post/(int:post_id), 24
GET /flicket-api/posts/(int:ticket_id)/(int:page)/, 25
GET /flicket-api/priorities/, 29
GET /flicket-api/priority/(int:priority_id), 28
GET /flicket-api/status/(int:status_id), 30
GET /flicket-api/statuses/, 30
GET /flicket-api/subscription/(int:subscription_id), 31
GET /flicket-api/subscriptions/(int:ticket_id)/, 32
GET /flicket-api/ticket/(int:ticket_id), 21
GET /flicket-api/tickets/, 22
GET /flicket-api/upload/(int:upload_id), 33
GET /flicket-api/uploads/, 33
GET /flicket-api/user/(int:user_id), 19
GET /flicket-api/users/, 20
POST /flicket-api/tickets(str:title,str:content,int:category_id,int:ticket_priority_id), 23

/http:

POST http://localhost:5000/flicket-api/departments(str:department), 28

INDEX

E

extension_allowed() (*flicket_admin.models.flicket_config.FlicketConfig static method*), 6

F

flicket_admin (*module*), 5
flicket_api.views.departments (*module*), 26
flicket_api.views.posts (*module*), 24
flicket_api.views.priorities (*module*), 28
flicket_api.views.status (*module*), 29
flicket_api.views.subscriptions (*module*), 31
flicket_api.views.tickets (*module*), 21
flicket_api.views.tokens (*module*), 18
flicket_api.views.uploads (*module*), 33
flicket_api.views.users (*module*), 19
FlicketConfig (*class in flicket_admin.models.flicket_config*), 5

V

valid_extensions() (*flicket_admin.models.flicket_config.FlicketConfig static method*), 6