Demo web App

Installation Instructions

To run the demo application you need:

Python 3.7 pip install the included requirements.txt SQLite3, SQLite 3, effectively any version should work.

To run the application start it after installing the requirements with:

```
 On Windows:
   set FLASK_APP=demo_app

On Linux:
   export FLASK_APP=demo_app

flask init-db
flask run --host=0.0.0.0 --port=8080
```

Acceptance Criteria

As a UI user I can:

1. Register through web portal
2. Review my own user information from the main view

As an API Consumer I can:

1. Register new users
2. Review users registered in system
3. If authenticated I can get personal information of users
4. If authenticated I can update personal information of users

API Brief

The Application exposes a simple API with the following routes:

Route	Methods	Authentication
/api/auth/token	GET	Basic
/api/users	GET	Token
/api/users	POST	None
/api/users/{username}	GET, PUT	Token

Headers

Your request headers should set at minimum:

```
'Content-Type': 'application/json'
```

Authentication

Access to users information requires a Token based authentication.

To receive a token perform basic authentication against `/api/auth/token` using the username/password you registered with in the Web interface.

For example:

```
>>> curl -u username:1234 http://localhost:8080/api/auth/token
{
  "status": "SUCCESS",
  "token": "MzMyNjQyMzAzODMwNjk1Mzg1MDU4OTA3MTEyMDM3MTQ2NDg5Mzg2"
}
```

For subsequent accesses to endpoints requiring a token update your request headers to include it:

```
'Content-Type': 'application/json'
'Token': 'MzMyNjQyMzAzODMwNjk1Mzg1MDU4OTA3MTEyMDM3MTQ2NDg5Mzg2'
```

General API responses

All API calls respond in the following scheme:

```
{'status': 'SUCCESS/FAILURE',
           'message': 'human readable message',
           'payload': {...}}
```

Updating user information

Subject information can be updated by sending PUT requests with a simple payload like:

```

{'datapoint1': 'value',
 'datapoint2': 'value',
  ...}
```