Create a webservice which can shorten urls like TinyURL and bit.ly

Deliverables

- Source code
- Simple way to to setup a virtual environment with the necessary python libraries
- Instructions to start the application
- Instructions to run unit test(s)

Application requirements

- Written in Python
- The webservice (for example Flask) is exposing the following endpoints:
    - POST **/shorten**
      The request body will have the following content:
      {
      "url": "https://www.energyworx.com/",
      "shortcode": "ewx123"
      }
      When no shortcode provided it should create a random shortcode for the provided url. The shortcode has a length of 6 characters and will contain only alphanumeric characters or underscores.

      Returns http status 201 with the following body:
      {
      "shortcode": "ewx123"
      }

      Errors:

400	Url not present
409	Shortcode already in use
412	The provided shortcode is invalid

    - GET **/<shortcode>**
      Returns http status 302 with the Location header containing the url

      Errors:

404	Shortcode not found

    - GET **/<shortcode>/stats**
      Returns http status 200 with the following body:
      {
      "created": "2017-05-10T20:45:00.000Z",
      "lastRedirect": "2018-05-16T10:16:24.666Z",
      "redirectCount": 6

}

<created> contains the creation datetime of the shortcode (in ISO8601)
<lastRedirect> contains the datetime of the last usage of the shortcode (in ISO8601)
<redirectCount> indicates the number of times the shortcode has been used

Errors:

404	Shortcode not found

- A datastore isn't mandatory although we would be more impressed if you use one
- Everything must be executable on Mac and Linux

Tips:

- Keep the code clean and simple.
- Make sure your unit test(s) is/are only testing your own code.