Bankend: python

REST/HTTP/RPC: node.js / Flask

[GET]

construct a API Response object
- get_response()

load address formats from a file
- load_formats()

verify that the provided address matches the format of the country provided
- verify_address(address, country_code)

allow the user to read address formats, and provide accessibility for frontend
- GetFormats(Resource)

allow the user to retrieve all stored addresses:
- responses={200: "Success", 400: "Invalid request format"})

allow user get a specific address based upon the identifier
- AddressById(Resource):
  responses(200: "Success", 400: “Error: No request body")

allow the user to search based on a country specific format
- GetByCountry()
  responses(200: "Success", 400: “Error: Country is not currently handled by this API.")

allow Normalize

[PUT]

allow user to update the specified address
- response(400, {"error": "Address has not yet been created."})

Database: MySQL/mongoDB,

Resource of Data:

`http://s3.amazonaws.com/data.openaddresses.io/openaddr-collected-global.zip`

[445 folders! All countries, all styles, each folder has flat files for each]

Example of .CSV file

```
: 1 fn1 = './data/us/wa/city_of_tacoma.csv'

: 1 data = pd.read_csv(fn1)

: 1 data.head()
```

:

	LON	LAT	NUMBER	STREET	UNIT	CITY	DISTRICT	REGION	POSTCODE	ID	HASH
0	-122.511366	47.302188	5402	N SHIRLEY ST	NaN	NaN	NaN	NaN	NaN	NaN	fe186786aba9e32c
1	-122.512642	47.303057	5501	N PEARL ST	NaN	NaN	NaN	NaN	NaN	NaN	649f166e3abcfe1e
2	-122.510176	47.303138	5401	YACHT CLUB RD	NaN	NaN	NaN	NaN	NaN	NaN	71354744591d3065
3	-122.509588	47.304110	5401	YACHT CLUB RD	NaN	NaN	NaN	NaN	NaN	NaN	3dfafa85de34c62c
4	-122.508763	47.305405	5603	N WATERFRONT DR	NaN	NaN	NaN	NaN	NaN	NaN	142e2d0f53db3c18

```
CREATE DATABASE worldAddress;
USE worldAddressn;
CREATE TABLE `worldAddress_db`(
         `id` INT(10) UNSIGNED,
         `country_id` INT(10) UNSIGNED,
         `country_Hash` INT(16) UNSIGNED
         `country_ISO` CHAR(2),
         `country_name` VARCHAR(64),
         `city_name` VARCHAR(128),
         `street_name` VARCHAR(128),
         `region_number` VARCHAR(128),
         `district` VARCHAR(128),
         `number` INT(10),
         `latitude` DOUBLE,
         `longitude` DOUBLE,
         `post_code` VARCHAR(30),
         `time_zone` VARCHAR(8),
         INDEX `idx_id` (`id`),
         INDEX `idx_country_id` (`country_id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
```

Metadata: JSON

Structure JSON for Country:

Cached and Stored user input.

Address.json:

```
{
  “AM”:{
             "name": "Armenia",
             "format": {
             "Address1": "",
             "Address2": "",
             "City": "",
             "State":"",
             "ZipCode": ""
   }
},
"US": {
   "name": "United States",
   "format": {
    "Address1": "",
    "Address2": "",
    "City": "^[a-zA-Z]+(?:[\\s-][a-zA-Z]+)*$",
    "State": {
     "AL": "Alabama",
     "AK": "Alaska",
     "AZ": "Arizona",
     "AR": "Arkansas",
     "CA": "California",
     "CO": "Colorado",
     "CT": "Connecticut",
```

```
      "DE": "Delaware",
      "FL": "Florida",
      "GA": "Georgia",
      "HI": "Hawaii",
      "ID": "Idaho",
      "IL": "Illinois",
      "IN": "Indiana",
      "IA": "Iowa",
      "KS": "Kansas",
      "KY": "Kentucky",
      "LA": "Louisiana",
      "ME": "Maine",
      "MD": "Maryland",
      "MA": "Massachusetts",
      "MI": "Michigan",
      "MN": "Minnesota",
      "MS": "Mississippi",
      "MO": "Missouri",
      "MT": "Montana",
      "NE": "Nebraska",
      "NV": "Nevada",
      "NH": "New Hampshire",
      "NJ": "New Jersey",
      "NM": "New Mexico",
      "NY": "New York",
      "NC": "North Carolina",
      "ND": "North Dakota",
      "OH": "Ohio",
      "OK": "Oklahoma",
      "OR": "Oregon",
      "PA": "Pennsylvania",
      "RI": "Rhode Island",
      "SC": "South Carolina",
      "SD": "South Dakota",
      "TN": "Tennessee",
      "TX": "Texas",
      "UT": "Utah",
      "VT": "Vermont",
      "VA": "Virginia",
      "WA": "Washington",
      "WV": "West Virginia",
      "WI": "Wisconsin",
      "WY": "Wyoming"
    },
    "ZipCode": "^\\d{5}$"
  }
 },
}
```

Structure JSON for address search

Address2.json

```
{
  "label": "Country",
  "options": [
    {
      "label": "Afghanistan",
      "iso": "AF",
      "fields": [
        {
          "thoroughfare": {
            "label": "Address 1"
          }
        },
        {
          "premise": {
            "label": "Address 2"
          }
        },
        {
          "locality": [
            {
              "localityname": {
                "label": "City"
              }
            },
            {
              "postalcode": {
                "label": "Postal code",
                "format": "^\\d{4}$",
                "eg": "1001"
              }
            }
          ]
        }
      ]
    },
{
      "label": "Aland Islands",
      "iso": "AX",
      "fields": [
        {
          "thoroughfare": {
            "label": "Address 1"
          }
        },
        {
          "premise": {
            "label": "Address 2"
          }
        },
        {
          "locality": [
            {
              "localityname": {
                "label": "City"
              }
            },
            {
              "postalcode": {
                "label": "Postal code",
                "format": "^22\\d{3}$",
                "eg": "22150"
              }
            }
          ]
        }
      ]
```

```
  }
}
```

FrontEnd: php for webserver pages, html, CSS, Javascript.

Dynamic Form