

Geocoder

A utility for determining the latitude and longitude for locations both inside and outside of the US.

Installation

You will need the geocoder-mysql.zip file, which contains the code for geocoder-mysql. You will also need to create a mysql database and an associated user which the tool can use. The default database to which the tool will attempt to connect is mysql://localhost/geocoder, using the user "geocoder" with password "geocoder".

Dependencies

This version of the geocoder uses mysql as its relational database engine and requires the Sequel and FasterCSV gems to function properly.

Usage

Syntax:

```
ruby geocoder-mysql.rb [user pass host db_name table_name] <city || location> <state || country>
```

Provide the city/location and the 2-letter abbreviation of the state (if in the US) or country (if outside the US) in which it is located as arguments to the tool. If you are not using the default settings for user, host, etc., you will need to supply these on the command line as well. Alternatively, you could simply edit the default constants near the top of the geocode-mysql.rb file to the settings that you wish to use. Note that cities with multiple words must be quoted. Due to the normal convention of placing a comma between the city and state/country, this is also allowed. If the location is known, the tool will print the latitude and longitude associated with the given location to the terminal. It may take several seconds for the geocoder to complete a query.

Valid Examples:

```
>ruby geocode-mysql.rb geocoder geocoder localhost geocoder geocodes Cullowhee NC  
(Cullowhee, NC) : (35.3137111, -83.1765333)
```

```
>ruby geocode.rb "Salt Lake City" UT  
(Salt Lake City, UT) : (40.7607794, -111.8910474)
```

```
>ruby geocode.rb "Washington D.C.", US  
(Washington D.C., US) : (38.8951118, -77.0363658)
```

```
>ruby geocode.rb Moscow, RU  
(Moscow, RU) : (55.7522222, 37.6155556)
```

```
>ruby geocode.rb Tokyo JP  
(Tokyo, JP) : (35.6895265930799, 139.691677093506)
```