Project IPGEO for pure 360

Project description

Introduction

This task is intended for prospective PHP Developers to demonstrate capability in the following skills:

- Retrieving and working with remote resources
- Building and populating databases
- Structuring PHP applications
- Designing and serving REST APIs
- Use of Github (public or private repo) or other public code repository service
- Documenting web services and applications

Output

The output of the task should include the following:

1. A process for constructing a database to hold the GeoLite Country reference data
2. A process that checks if the GeoLite Country database is populated and if not, downloads the
   GeoLite Country database file from http://geolite.maxmind.com/download/geoip/database/GeoIPCountryCSV.zip and populates the database from the data file
3. An API that supports a restful GET endpoint that returns the country when supplied with an IP address (GET /locationByIP?IP=127.0.0.1)
4. An accessible online source code repository containing the output of this task and appropriate documentation

Support

Please don't hesitate to ask if any of the instructions are unclear to you. Questions can be sent to the following address and will be answered as quickly as possible:

michael.pinnell@pure360.com

Database design

mysql server details

server: dty.mine.nu:3306
user: pure360
password: Pure360
database: pure360
table: ipgeo

The table ipgeo has the data from the cvs. This table we will be used to query data for our REST API. To do this faster two indexes got created for columns a and b (that have the ip information in numeric form).

The information that we store on the database is.

- ip1 , as variable , tell us from what ip the subnet starts
- ip2 , as variable , tell us from what ip the subnet stops
- a , as big integer , has the ip1 in numeric form
- b , as big integer , has the ip2 in numeric form
- country code , 3 char var
- country , var

Name	Type	Length	Decimals	Not Null	Key
id	int	11	0	☑	🔑1
ip1	varchar	15	0	☐	
ip2	varchar	15	0	☐	
a	bigint	10	0	☐	
b	bigint	10	0	☐	
countryCode	varchar	4	0	☐	
country	varchar	50	0	☐	

database Indexes

To make the database query faster we use two indexes for column a and b.

Its a unique type, btree method, as we are dealing with unique numbers.

APPLICATION DESIGN

Data import

This php software imports the data on the online zipped cvs to a mysql database.

Located:

`/import.php`

How to run:

From the terminal type

`php -f import.php`

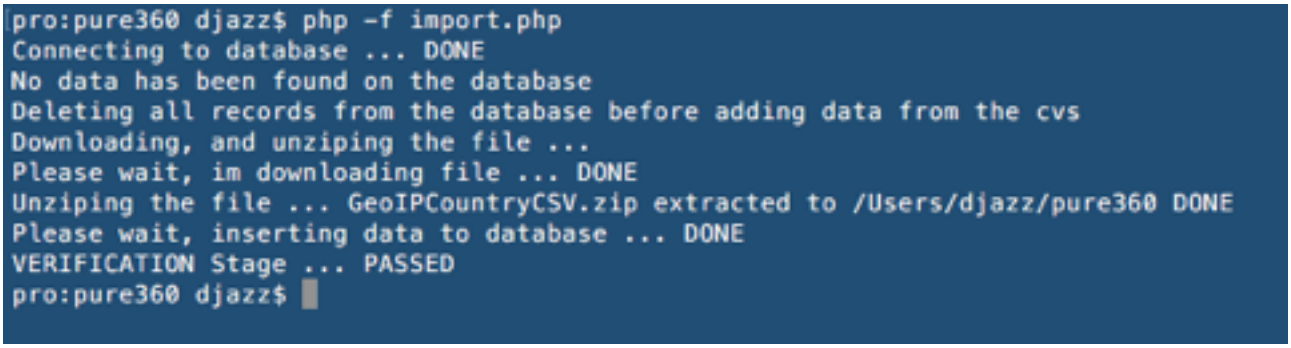
This application will check theres data on the database, then if the cvs exsit, then if we need to download the file and then do the appropriate steps to do all these tasks.

If a step fail you will see the step that failed.

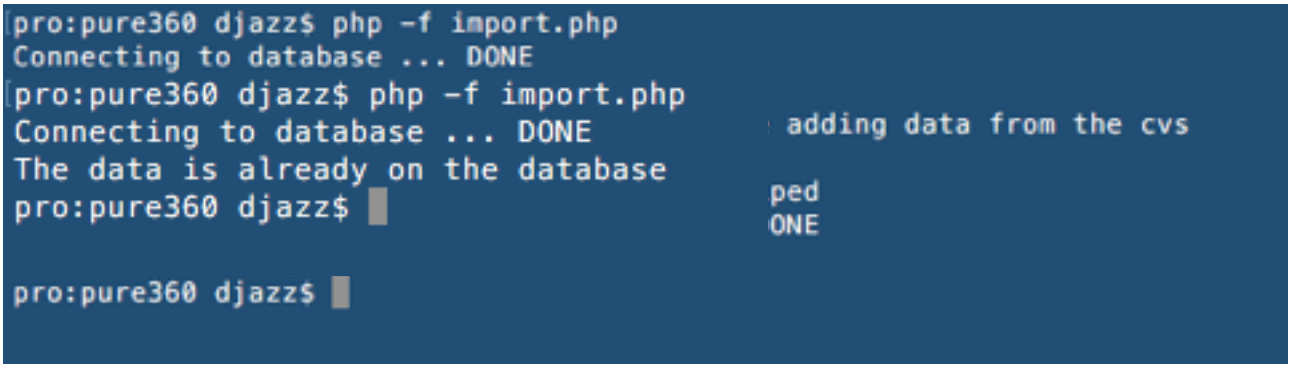
We use 1 query to insert all data and we filter the input of the country name.

Once the import process get completed then there a verification phase that will check if the records on the CVS is equal to the number of records on the database.

First run example



Running after truncating the ipgeo table but we still have the cvs file.



Running in case that its all done.

The REST API

How to run:

from the testing server run

/rest.php?ip=64.35.63.67

It uses $_GET[ip] as input to get the ip to check, then it runs this mysql query and we have the answer in a json output.

The query the brings the data from ipgeo and its using the a and b indexes

```
$result = mysqli_query($connection,"SELECT *  FROM ipgeo USE INDEX (a,b)
WHERE `a` <= $iplong AND `b` >= $iplong LIMIT 0,1");
```

Name	Type	Length	Decimals	Not Null	Key
id	int	11	0	☑	🔑
ip1	varchar	15	0	☐	
ip2	varchar	15	0	☐	
a	bigint	10	0	☐	
b	bigint	10	0	☐	
countryCode	varchar	4	0	☐	
country	varchar	50	0	☐	