

Manual de deployment

Puntos de recarga Bip! en estaciones Metro

10/06/2017

felipe castro valenzuela

Índice

Introducción	2
Análisis y requerimientos del sistema.....	3
Arquitectura software	4
Requisitos software.....	5
Configuración	6
Anexos	8
Glosario	9

Introducción

El siguiente documento comprende de la explicación técnica de las tecnologías aplicadas en la creación del sistema, así como los pasos a seguir para su despliegue y configuración.

Análisis y requerimientos del sistema

Desarrollo en Python "Puntos de recarga Bip! en estaciones Metro"

1. Tomar como entrada la información de puntos de carga Bip! que está disponible

En: http://datos.gob.cl/api/action/datastore_search?resource_id=ba0cd493-8bec-4806-91b5-4c2b5261f65e

Más información sobre el recurso lo puedes encontrar en

<http://datos.gob.cl/dataset/33355/resource/ba0cd493-8bec-4806-91b5-4c2b5261f65e>

2. Almacenar estos datos en MongoDB.
3. Exponer esta información como servicio de búsqueda en el puerto 9000 solo con Método GET. Se le podrá pasar como parámetro un punto geográfico P(latitud, longitud) y el radio R de búsqueda.
4. El resultado de la búsqueda será un objeto JSON con los puntos cercanos a P(latitud,longitud) dentro del radio R.

Desarrollo Java o PHP:

1. Desarrollar una pequeña aplicación web que lea el servicio "Puntos de recarga BIP! en estaciones Metro"
2. Mostrar los puntos de recarga cercanos en un mapa, utilizando geolocalización (obtener a través del browser tu ubicación actual).

Arquitectura software

El sistema encomendado requiere de múltiples tecnologías para su despliegue, comprende de una capa de datos desarrollada con python y es un servicio de tipo RESTful, luego hay una capa de comunicación desarrollada en php, que se encarga de enrutar la petición y dirigirla a la aplicación configurada y la capa front end desarrollada en angularJS que se comunica con la capa de comunicación entregándole los datos requeridos para su despliegue todo en formato Json

Requisitos software

1. Tener instalado y configurado el motor de base de datos mongoDB
Como referencia tomar las siguientes páginas
<https://docs.mongodb.com/manual/tutorial/install-mongodb-on-windows/>
<https://docs.mongodb.com/manual/tutorial/install-mongodb-enterprise-on-ubuntu/>
2. Tener instalado Python 3.6.1
Página de referencia <https://www.python.org/downloads/>
3. Tener instalado un ambiente local con apache y php 5
Página de referencia <https://www.apachefriends.org/es/index.html>
4. Para este ejemplo se ocupó el sistema operativo Windows 10 pro

Configuración

Para su optimo requerimiento se recomienda cargar todos los archivos dentro del localhost del servidor local tomar como referencia esta ruta “C:\xampp\htdocs\” y sobre Windows

1. Clonar el repositorio <https://github.com/pipec80/ugdfullstack.git>

```
git clone https://github.com/pipec80/ugdfullstack.git
```

2. Configurar las path de petición al servicio de la capa de comunicación, para que el front end pueda acceder a los datos del servicio , la ruta es:

“**ugdfullstack/app/ config.js**” y editar el campo

```
constant('URL_API', 'http://mi-ruta-localhost/ugdfullstack/layer_communication/rest.php')
```

3. Para cargar la base de datos del sistema hay que ingresar al siguiente archivo:

“**ugdfullstack/layer_data/ load_datos_bip.py**”

Se necesitan las siguientes dependencias para que funcione el script:

```
python -m pip install pymongo
pip3 install requests
```

Luego estando en la misma path “**ugdfullstack/layer_data/**” ejecutar la instrucción:

```
python .\load_datos_bip.py
```

El resultado exitoso muestra en consola el siguiente mensaje:

```
puntos cargados 100
```

4. Para la instalación de la capa de servicio restful hay que ingresar al siguiente archivo:
“**ugdfullstack/layer_data/ restapi.py**”

Se necesitan las siguientes dependencias para que funcione el script:

Instalamos virtualenv, el cual nos permitirá crear entornos virtuales

```
pip install virtualenv
```

Dentro de la carpeta “**ugdfullstack/layer_data/**” creamos nuestro entorno virtual.

```
virtualenv flask
```

Luego dentro de la misma carpeta Instalar flask

```
pip install Flask
```

Con estos pasos tenemos instalado el servicio RESTful para probar el servicio hay que ejecutar el siguiente comando:

```
python .\restapi.py
```

URL base de los servicios de prueba es: <http://127.0.0.1:9000/>

5. Ejecutar la interfaz web <http://localhost/ugdfullstack/app/>

Anexos

Glosario

Término	Descripción