**Extracción de datos de Idealista a través del servicio API**

La web Idealista.com ofrece un servicio gratuito para la descarga de los datos públicos que tiene de todas las propiedades para venta y alquiler registradas en sus servidores, y que son la base de nuestro proyecto para la puesta en marcha de Scope.

Una vez registrados en el servicio ( <http://developers.idealista.com/access-request> ), nos proporcionaron unas claves de acceso, y lo que hemos realizado son numerosas llamadas a la API para obtener las viviendas ubicadas principalmente en toda Comunidad de Madrid para nuestro propósito. Este servicio tiene básicamente dos restricciones:

* no más de 100 llamadas por usuario y mes
* cada llamada devuelve un máximo de 50 viviendas
* la búsqueda hemos de realizarla por coordenadas geoespaciales

El primer paso necesario es la autentificación de los usuarios para poder realizar la petición a la API. Para ello hemos hecho uso de la utilidad oauth2 que proporcionan y en la que hay que configurar estos tres puntos:

1. Codificar las claves secret y key proporcionadas
2. Obtener el token de acceso
3. Solicitar la autentificación del Api con el token

Para realizar la llamada generamos una URL con base <https://api.idealista.com/3.5/es/search>? y a la que vamos añadiendo los parámetros de entrada necesarios.

Como parámetros fijos requeridos utilizamos:

* *Country* : ‘es’
* *Operation* : ‘sale’ o ‘rent’
* *propertyType* : 'homes'
* *locationId* : '0-EU-ES-28'
* *maxItems* : 50

Como parámetros variables utilizamos:

* *center* : latitud, longitud
* *distance* : metros de distancia desde center

La respuesta a la llamada lo devuelve en formato JSON con la siguiente estructura:

{

"actualPage":1,

"elementList":[

{ elemento1},

{elemento2},

..,

{elemento50}

],

"itemsPerPage":50,

"lowerRangePosition":0,

"numPaginations":0,

"paginable":false,

"resultSummary": [

"comprar",

"garajes",

"madrid provincia",

"entre 3.000 y 500.000 euros",

"con video profesional"

],

"total":1,

"totalPages":1,

"upperRangePosition":1

}

Con este formato lo que realmente necesitamos es lo que contiene “elementList”, que es una lista de 50 elementos máximos, cada uno con toda la información de la vivienda que devuelve, y que tiene la siguiente estructura:

{

“showAddress": true,

"detailedType": {

"typology": "flat" },

"country": "es",

"price" 500,

"rooms": 2,

"propertyType": "flat",

"size": 76,

"floor": "bj",

"url": "http://www.idealista.com/34257073",

"address": "avenida de la pedriza, 13",

"hasLift": true,

"suggestedTexts": {

"subtitle": "Manzanares el Real",

"title": "Piso en avenida de la pedriza, 13" },

"operation": "rent",

"priceByArea": 7,

"province": "Madrid",

"thumbnail": https://img3.idealista.com/thumbs?wi=140&he=105&en=%2BtSLyO%2

VWesisrLJUHqccKZOukuYN5VRYLz1RLlDEtglEcKFetA5LMd2SHdJFqqai0nA%3D%3D&ch=-1072186729",

"numPhotos": 19,

"latitude": 40.7282093,

"status": "good",

"propertyCode": "34257073",

"longitude": -3.8716999,

"exterior": false,

"newDevelopment": false,

"municipality": "Manzanares el Real",

"bathrooms": 2,

"hasVideo": false,

"parkingSpace": {

"hasParkingSpace": true,

"isParkingSpaceIncludedInPrice": true }

}

Nuestra idea para poder almacenar todos estos datos, ha sido almacenarlos en la base de datos NoSql MongoDB. Para ello nos creamos la colección *Madrid* en la que hemos ido recogiendo todos los registros devueltos por Idealista.

Debido a la gran cantidad de datos que hemos necesitado y a las restricciones antes indicadas que la web establece, hemos realizado un programa en Python (Peticion\_idealista.py) que realice las llamadas a partir de unas coordenadas concretas y vaya almacenando en Mongo todos los registros, incluyendo antes la fecha de carga en cada uno de ellos, ya que este dato no nos lo proporcionan.

A través de este método hemos obtenido más de 65.000 registros.