

# Guía rápida del servidor.

---

## !oneTouchEasy

Alberto Junquera Ramírez

Camila López

Enrique Poyato Ortiz

Eric Olsson

Marta Maquedano

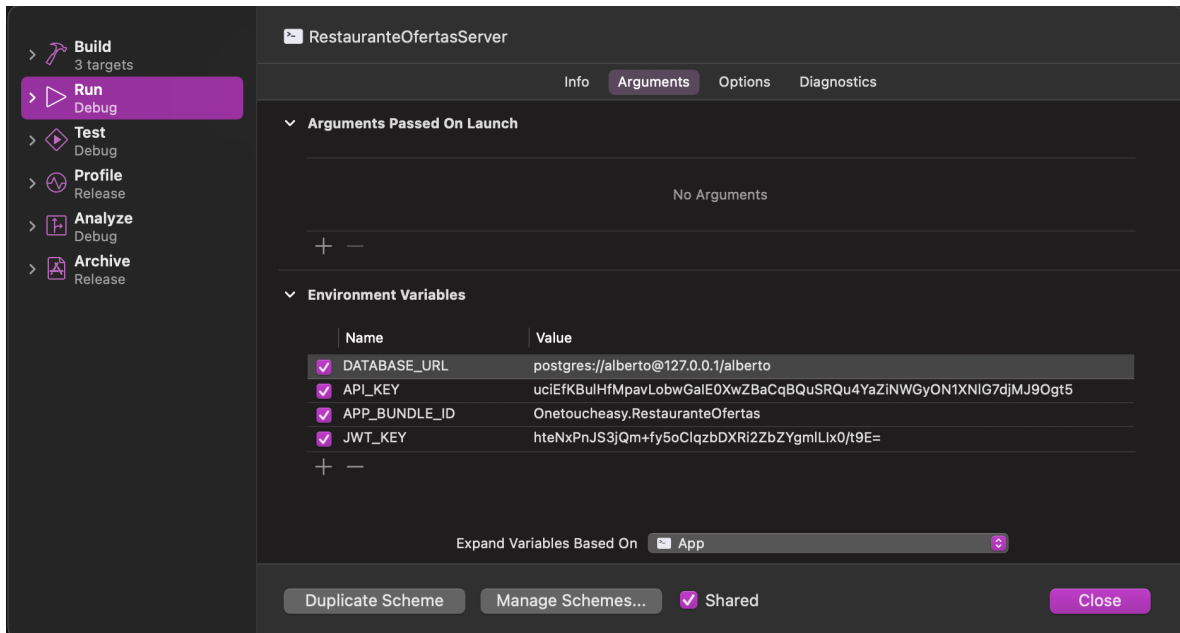
# Índice:

<b>Configuración previa:</b> .....	<b>3</b>
<b>xCode</b> .....	<b>3</b>
<b>Android:</b> .....	<b>3</b>
<b>JWT</b> .....	<b>4</b>
<b>Nota sobre las migraciones</b> .....	<b>5</b>
<b>EndPoints:</b> .....	<b>6</b>
Version.....	6
SignUp.....	7
SignIn.....	8
Refresh Token.....	9
Create Restaurant.....	10
Update Restaurant.....	12
AllRestaurants.....	14
GetRestaurantByID.....	15
GetRestaurantsByCompany.....	16
GetRestaurantsByType.....	17
getRestaurantWithOffers.....	18
getRestaurantsWithOffers.....	19
getOfferByID.....	20
getOffers.....	21

# Configuración previa:

## xCode

Agregar las variables de entorno. Principalmente la base de datos local. El resto deberían estar por defecto.



DATABASE_URL	
APP_BUNDLE_ID	Onetoucheasy.RestauranteOfertas
API_KEY (orig)	uciEfKBuIHfMpavLobwGaIE0XwZBaCqBQuSRQu4YaZiNWGyON1XNIg7dJMJ9Ogt5
JWT_KEY	hteNxPnJS3jQm+fy5oClqzbDXRi2ZbZYgmLLIx0/t9E=

## Android:

Como detalle a tener en cuenta, decir que para acceder a la dirección de localhost (127.0.0.1) debe de utilizarse la dirección especial 10.0.2.2. Aunque ya está preconfigurado en el proyecto en el archivo "RemoteModule.kt":

```
fun provideRetrofit(okHttpClient: OkHttpClient, moshi: Moshi): Retrofit {  
    return Retrofit  
        .builder()  
        .baseUrl("http://10.0.2.2:8080/api/")  
        .client(okHttpClient)  
        .addConverterFactory(MoshiConverterFactory.create(moshi))  
        .build()  
}
```

# JWT

Para el acceso a los EndPoints de la base de datos, se necesita un JWT de acceso. Este token tiene información variada. En el payload, se puede ver que tiene la fecha de expiración, el bundle de la app, el tipo de token de acceso que es y el tipo de usuario al que corresponde dicho token entre otros.

## Encoded

PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE2OTQ1MzM2MjIuNDgyNzcsImIzcyI6I  
k9uZXRvdWNoZWZzeS5SZXN0YXVyYW50ZU9mZXJ0  
YXMiLCJ0eXB1IjoieYWNjZXNub2t1biIsInN1YiI  
6IjE2ODNGOEE3LUQyODYtNDU2Ni04QjRFLTlCMj  
BDNTE0MEJDRSIsInVzZXJ0eXB1IjoieY29tcGFue  
SJ9.A2y4T-  
S_aWS0MEF3ON7svxZvrUrFELlVfDTkRvV515c
```

## Decoded

EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{  
  "alg": "HS256",  
  "typ": "JWT"  
}
```

PAYLOAD: DATA

```
{  
  "exp": 1694533622.48277,  
  "iss": "Onetoucheasy.RestauranteOfertas",  
  "type": "accessToken",  
  "sub": "1683F8A7-D286-4566-8B4E-9B20C5140BCE",  
  "userType": "company"  
}
```

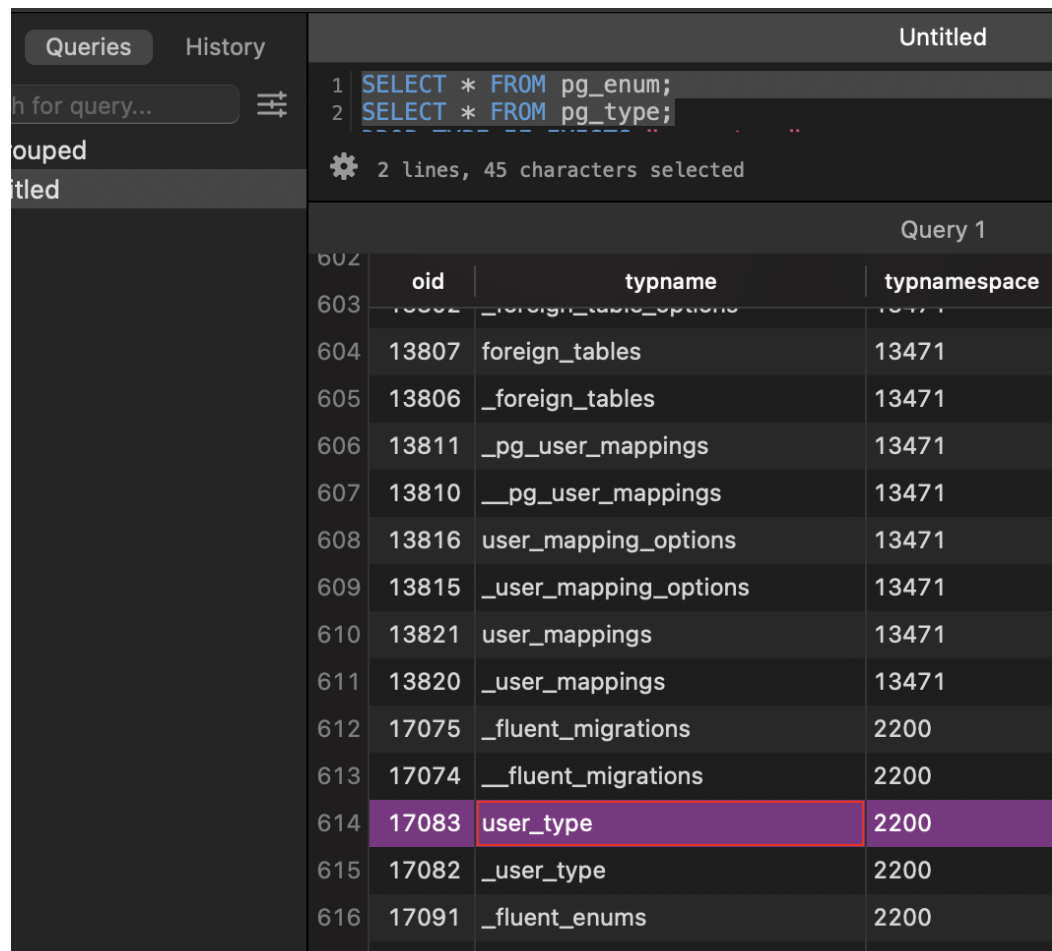
VERIFY SIGNATURE

```
HMACSHA256(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),  
    
) ☐ secret base64 encoded
```

# Nota sobre las migraciones

El tipo de usuario se ha gestionado creando un tipo enum en la base. Esto crea una referencia en la propia base de datos y en la tabla de tipos de PostgreSQL. Si se borra la base de datos y se ejecuta de nuevo la migración, saltará el error de que el tipo del enum ya existe. Esto se debe a que se tiene que borrar también de la tabla interna de tipos de datos de PostgreSQL. Para llevarlo a cabo:

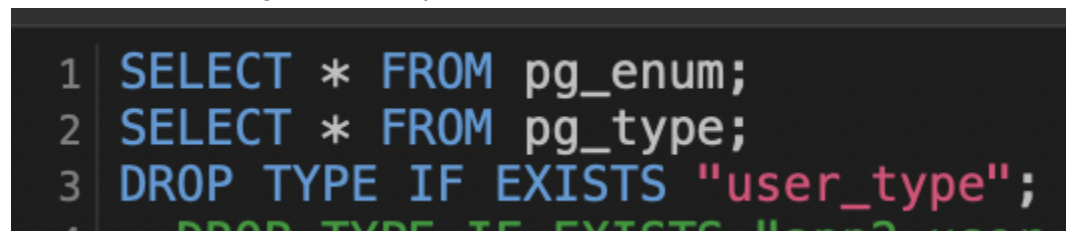
- 1) Se localiza el enum en la tabla de PostgreSQL con los comandos que se muestran.



The screenshot shows a database client interface with a 'Queries' tab. The query editor contains two lines of SQL: `SELECT * FROM pg_enum;` and `SELECT * FROM pg_type;`. Below the editor, the results of 'Query 1' are displayed in a table. The table has four columns: 'oid', 'typname', and 'typnamespace'. The row for 'user\_type' (oid 17083, namespace 2200) is highlighted in purple.

oid	typname	typnamespace
13807	foreign_tables	13471
13806	_foreign_tables	13471
13811	_pg_user_mappings	13471
13810	__pg_user_mappings	13471
13816	user_mapping_options	13471
13815	_user_mapping_options	13471
13821	user_mappings	13471
13820	_user_mappings	13471
17075	_fluent_migrations	2200
17074	__fluent_migrations	2200
17083	user_type	2200
17082	_user_type	2200
17091	_fluent_enums	2200

- 2) Se elimina con la siguiente query.

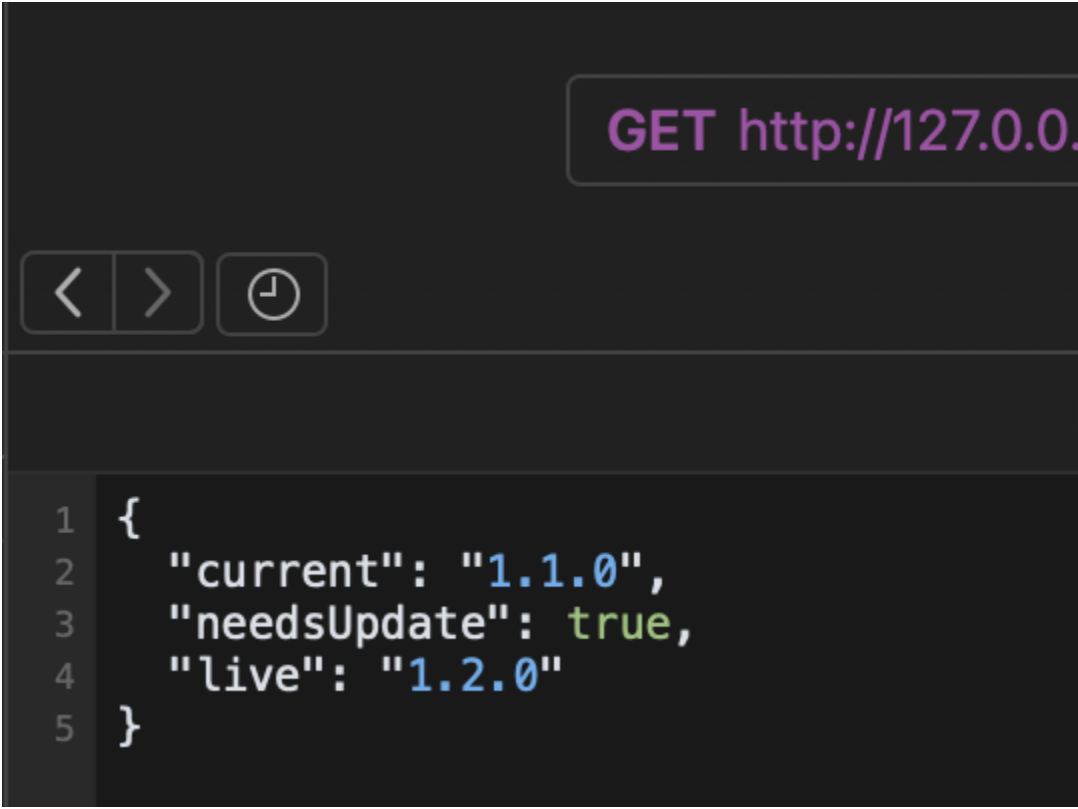


The screenshot shows a SQL query editor with the following commands: `SELECT * FROM pg_enum;`, `SELECT * FROM pg_type;`, `DROP TYPE IF EXISTS "user_type";`, and `DROP TYPE IF EXISTS "user_type";`.

```
1 SELECT * FROM pg_enum;
2 SELECT * FROM pg_type;
3 DROP TYPE IF EXISTS "user_type";
4 DROP TYPE IF EXISTS "user_type";
```

# EndPoints:

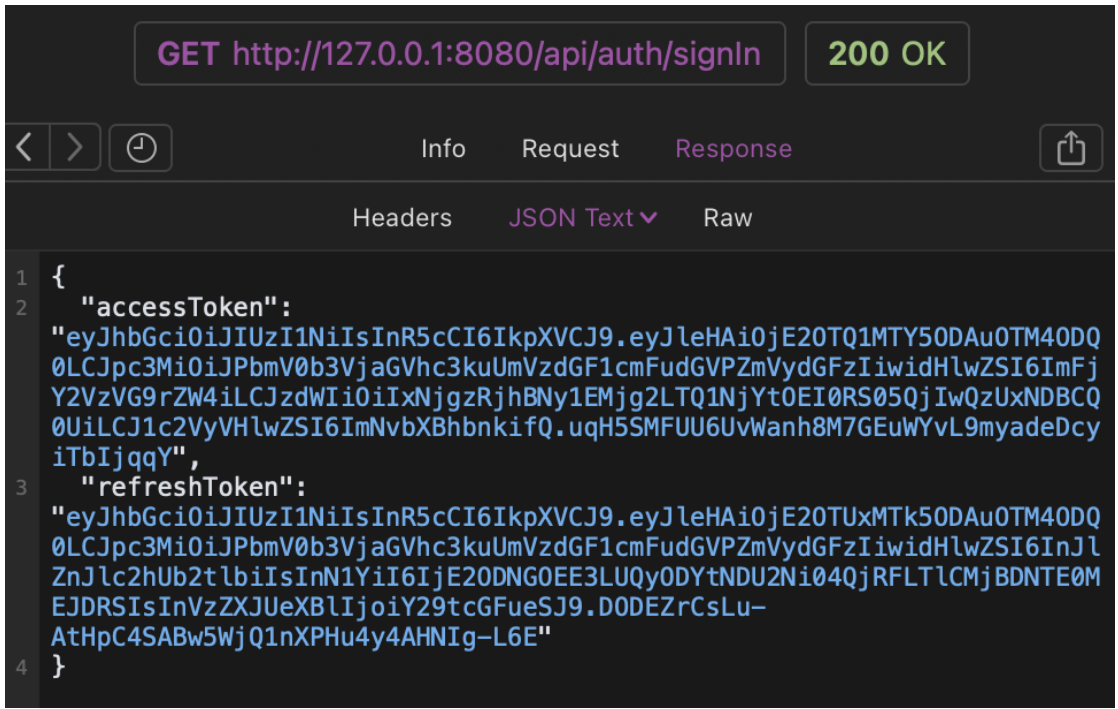
## Version

<u>EndPoint</u>	
<a href="http://127.0.0.1:8080/api/version">http://127.0.0.1:8080/api/version</a>	
<u>Método</u>	
Get	
<u>Headers</u>	
CDS-ApiKey	El valor de la ApiKey.
<u>URL Params</u>	
current	La versión actual de la aplicación. Ej. 1.1.0
<u>Respuesta</u>	
 <p>The screenshot shows a REST client interface. At the top, a purple box displays the request: <b>GET http://127.0.0.1:8080/api/version</b>. Below this, there are navigation buttons (back, forward, and a clock icon). The response is shown in a dark-themed code editor with line numbers 1 through 5 on the left. The JSON response is: <pre>{   "current": "1.1.0",   "needsUpdate": true,   "live": "1.2.0" }</pre></p>	

# SignUp

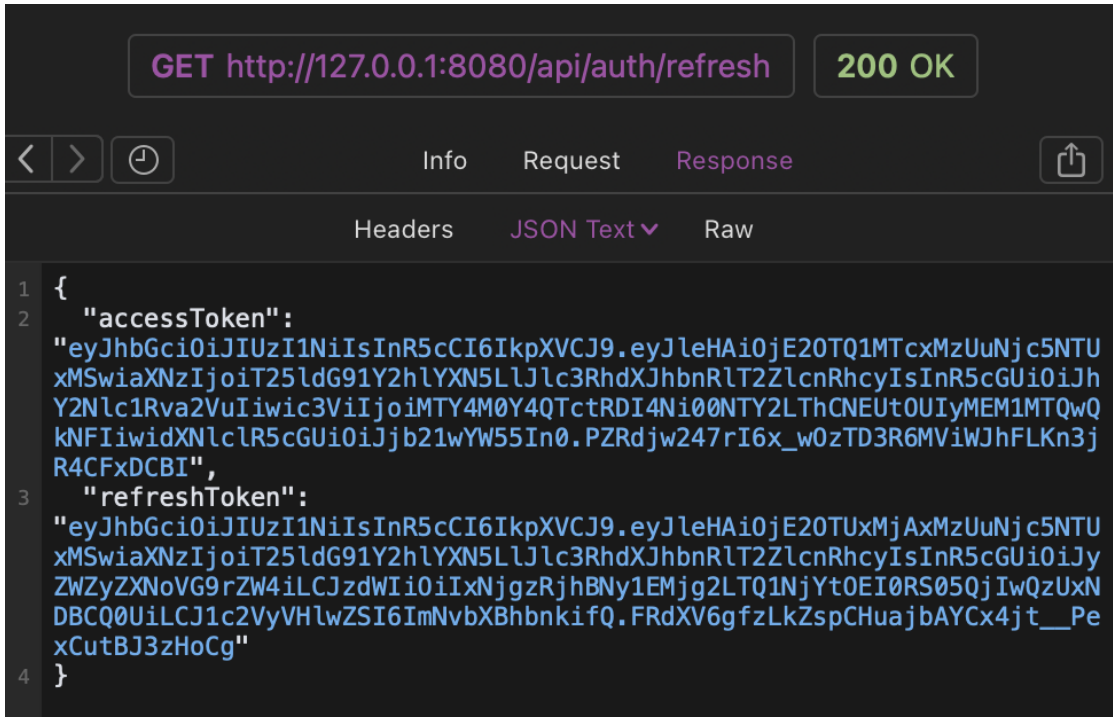
EndPoint	
<a href="http://127.0.0.1:8080/api/auth/signUp">http://127.0.0.1:8080/api/auth/signUp</a>	
Método	
Post	
Headers	
CDS-ApiKey	El valor de la ApiKey.
Body	
name	El nombre del usuario / empresa.
email	Un correo electrónico.
password	La contraseña. Debe tener como mínimo 6 caracteres.
type	El tipo de usuario. A través de esta petición sólo se pueden crear "customer" o "company".
Respuesta	
<div><div>POST http://127.0.0.1:8080/api/auth/signUp200 OK</div><div><div>&lt;&gt;🕒InfoRequestResponse🔗</div><div>HeadersJSON Text ▾Raw</div><div><pre>1 { 2   "accessToken":   "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE2OTQ1MTQwNjQuNjM4NTg   xLCJpc3MiOiJPbmV0b3VjaGVhc3kuUmVzdGF1cmFudGVPZmVydGFzIiwidHlwZSI6ImFj   Y2VzVG9rZW4iLCJzdWIiOiJDNTZFQUU1NC0wODIxLTRFNDR0E0yQzgzOEI4MjQ3M   0UiLCJ1c2VyVHlwZSI6ImNvbXBhbnkifQ.hSR85jqLBG0xR0Z0PIjX_6_9PEUuUGF0xqD   iT7tygp8",   "refreshToken":   "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE2OTUxMTQwNjQuNjM4NTg   xLCJpc3MiOiJPbmV0b3VjaGVhc3kuUmVzdGF1cmFudGVPZmVydGFzIiwidHlwZSI6ImFj   ZnJlc2hUb2t1b2IiLCJ1c2VyVHlwZSI6ImNvbXBhbnkifQ.hSR85jqLBG0xR0Z0PIjX_6_9PEUuUGF0xqD   iT7tygp8" 3 4 }</pre></div></div></div>	

## SignIn

EndPoint	
<a href="http://127.0.0.1:8080/api/auth/signIn">http://127.0.0.1:8080/api/auth/signIn</a>	
Método	
Get	
Headers	
CDS-ApiKey	El valor de la ApiKey.
Authorization	"Basic {email:contraseña}" {email:contraseña} es el String resultante al codificar email:contraseña en Base64.
Auth	
Username	Correo electrónico.
Password	Contraseña.
Respuesta	
	

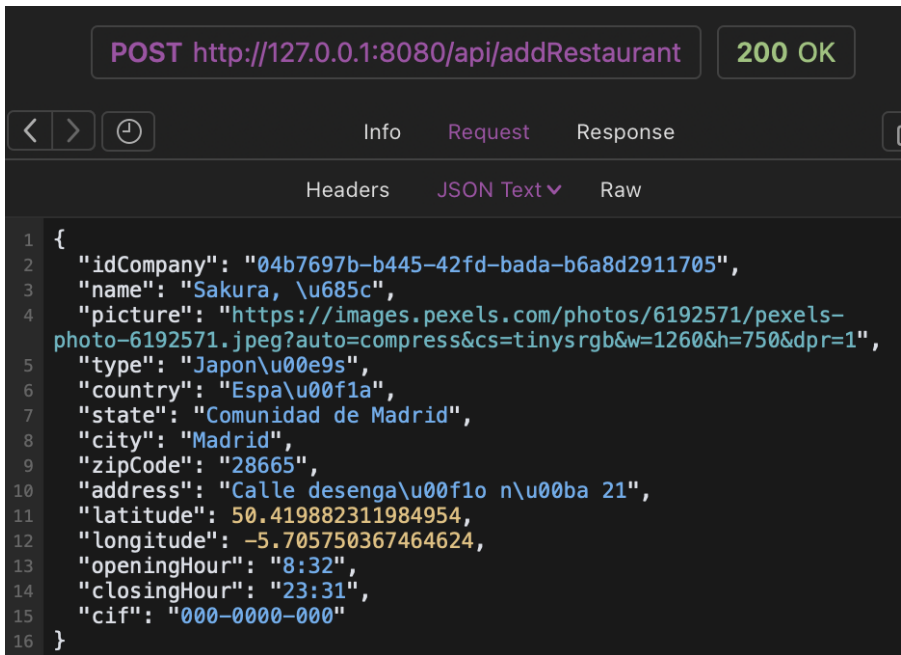


## Refresh Token

EndPoint	
<a href="http://127.0.0.1:8080/api/auth/refresh">http://127.0.0.1:8080/api/auth/refresh</a>	
Método	
Get	
Headers	
CDS-ApiKey	El valor de la ApiKey.
Authorization	Bearer refreshToken. refreshToken es un JWT que permite regenerar las claves JWT.
Respuesta	
	

## Create Restaurant

<b><u>EndPoint</u></b>	
<a href="http://127.0.0.1:8080/api/addRestaurant">http://127.0.0.1:8080/api/addRestaurant</a>	
<b><u>Método</u></b>	
Post	
<b><u>Headers</u></b>	
CDS-APIKey	El valor de la ApiKey.
Authorization	Bearer accessToken.
<b><u>Body</u></b>	
idCompany	idUsuario de tipo "company"
name	Nombre del restaurante.
picture	Url de una imagen.
type	Tipo de restaurante. Ej. Tailandés.
country	País
state	Estado, comunidad autónoma o prefectura.
city	Ciudad.
zipCode	Código postal. String (el código postal es un número pero no utiliza las propiedades de los números)
address	Dirección.
latitude	Latitud
longitude	Longitud
openingHour	Hora de apertura.
closingHour	Hora de cierre.
cif	Código de identificación fiscal.
<b><u>Respuesta</u></b>	



Tupla en la base de datos

id	uuid
06dd3bd3-5c4e-46b4-bff5-536ba2e1f17e	▼
id_company	uuid
04b7697b-b445-42fd-bada-b6a8d2911705	▼
created_at	text
2023-09-12T11:03:10Z	▼
name	text
Sakura, 桜	▼
picture	text
https://images.pexels.com/photos/6192571/pexels-photo-6192571.jpeg?auto=compress&cs=tinysrgb&w=1260&h=750&dpr=1	▼
cif	text
NULL	▼
type	text
Japonés	▼
id_address	uuid
a260cb96-cd27-4004-9df4-58800622cf95	▼
id_coordinates	uuid
053adc4b-69d6-4871-9b97-4c3f8e7757d5	▼
opening_hour	text
8:32	▼
closing_hour	text
23:31	▼

## Update Restaurant

<b><u>EndPoint</u></b>	
<a href="http://127.0.0.1:8080/api/addRestaurant">http://127.0.0.1:8080/api/addRestaurant</a>	
<b><u>Método</u></b>	
Post	
<b><u>Headers</u></b>	
CDS-ApiKey	El valor de la ApiKey.
Authorization	Bearer accessToken.
<b><u>Body</u></b>	
idCompany	idUsuario de tipo "company"
name	Nombre del restaurante.
picture	Url de una imagen.
type	Tipo de restaurante. Ej. Tailandés.
country	País
state	Estado, comunidad autónoma o prefectura.
city	Ciudad.
zipCode	Código postal. String (el código postal es un número pero no utiliza las propiedades de los números)
address	Dirección.
latitude	Latitud
longitude	Longitud
openingHour	Hora de apertura.
closingHour	Hora de cierre.
cif	Código de identificación fiscal.
<b><u>Respuesta (cambiando el nombre)</u></b>	

PUT ...:8080/api/restaurants/06dd3bd3-5c4e-46b4-bff5-536ba2e1f17e

200 OK

< > ⌚

Info Request Response

⬆

Headers JSON Text ⌵ Raw

1 {

2 "idCompany": "04b7697b-b445-42fd-bada-b6a8d2911705",

3 "name": "Sakura, \u685c 2.0",

4 "picture": "https://images.pexels.com/photos/6192571/pexels-photo-6192571.jpeg?auto=compress&cs=tinysrgb&w=1260&h=750&dpr=1",

5 "type": "Japon\u00e9s",

6 "country": "Espa\u00f1a",

7 "state": "Comunidad de Madrid",

8 "city": "Madrid",

9 "zipCode": "28665",

10 "address": "Calle desenga\u00f1o n\u00ba 21",

11 "latitude": 50.419882311984954,

12 "longitude": -5.705750367464624,



13 "openingHour": "8:32",

14 "closingHour": "23:31",

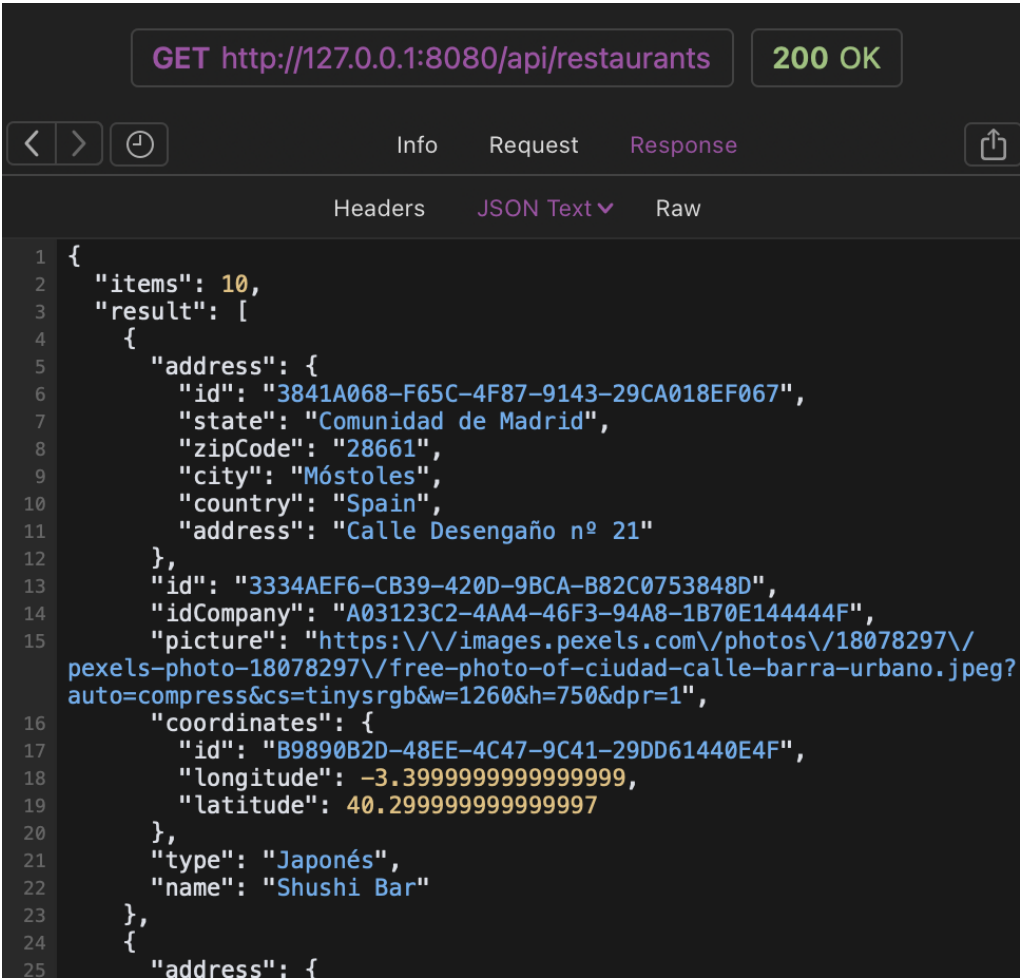
15 "cif": "000-0000-000"

16 }

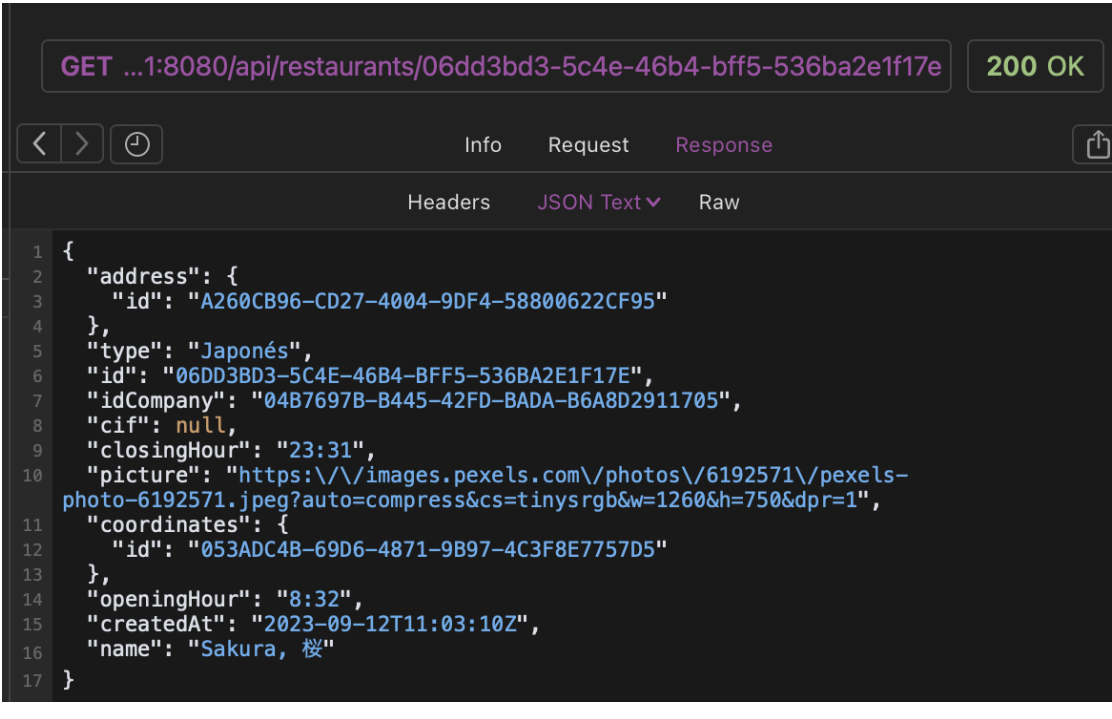
## Tupla en la base de datos

 Search for field...	
id	uuid
06dd3bd3-5c4e-46b4-bff5-536ba2e1f17e	⌵
id_company	uuid
04b7697b-b445-42fd-bada-b6a8d2911705	⌵
created_at	text
2023-09-12T11:03:10Z	⌵
name	text
Sakura, 桜 2.0	⌵
picture	text
https://images.pexels.com/photos/6192571/pexels-photo-6192571.jpeg?auto=compress&cs=tinysrgb&w=1260&h=750&dpr=1	⌵
cif	text
NULL	⌵
type	text
Japonés	⌵
id_address	uuid
a260cb96-cd27-4004-9df4-58800622cf95	⌵
id_coordinates	uuid
053adc4b-69d6-4871-9b97-4c3f8e7757d5	⌵
opening_hour	text
8:32	⌵
closing_hour	text
23:31	⌵

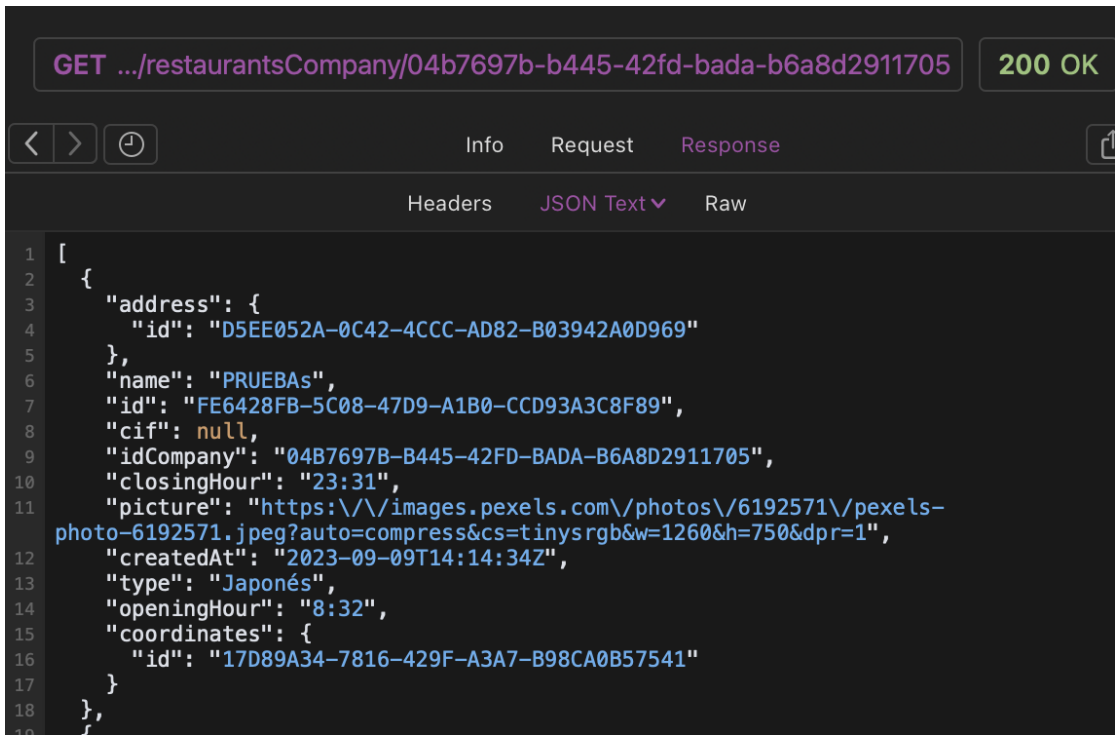
## AllRestaurants

EndPoint	
<a href="http://127.0.0.1:8080/api/restaurants">http://127.0.0.1:8080/api/restaurants</a>	
Método	
Get	
Headers	
CDS-ApiKey	El valor de la ApiKey.
Authorization	Bearer accessToken.
Respuesta	
 <pre>1 { 2   "items": 10, 3   "result": [ 4     { 5       "address": { 6         "id": "3841A068-F65C-4F87-9143-29CA018EF067", 7         "state": "Comunidad de Madrid", 8         "zipCode": "28661", 9         "city": "Móstoles", 10        "country": "Spain", 11        "address": "Calle Desengaño nº 21" 12      }, 13      "id": "3334AEF6-CB39-420D-9BCA-B82C0753848D", 14      "idCompany": "A03123C2-4AA4-46F3-94A8-1B70E144444F", 15      "picture": "https://images.pexels.com/photos/18078297/ pexels-photo-18078297/free-photo-of-ciudad-calle-barra-urbano.jpeg? auto=compress&amp;cs=tinysrgb&amp;w=1260&amp;h=750&amp;dpr=1", 16      "coordinates": { 17        "id": "B9890B2D-48EE-4C47-9C41-29DD61440E4F", 18        "longitude": -3.3999999999999999, 19        "latitude": 40.299999999999997 20      }, 21      "type": "Japonés", 22      "name": "Shushi Bar" 23    }, 24    { 25      "address": {</pre>	

## GetRestaurantById

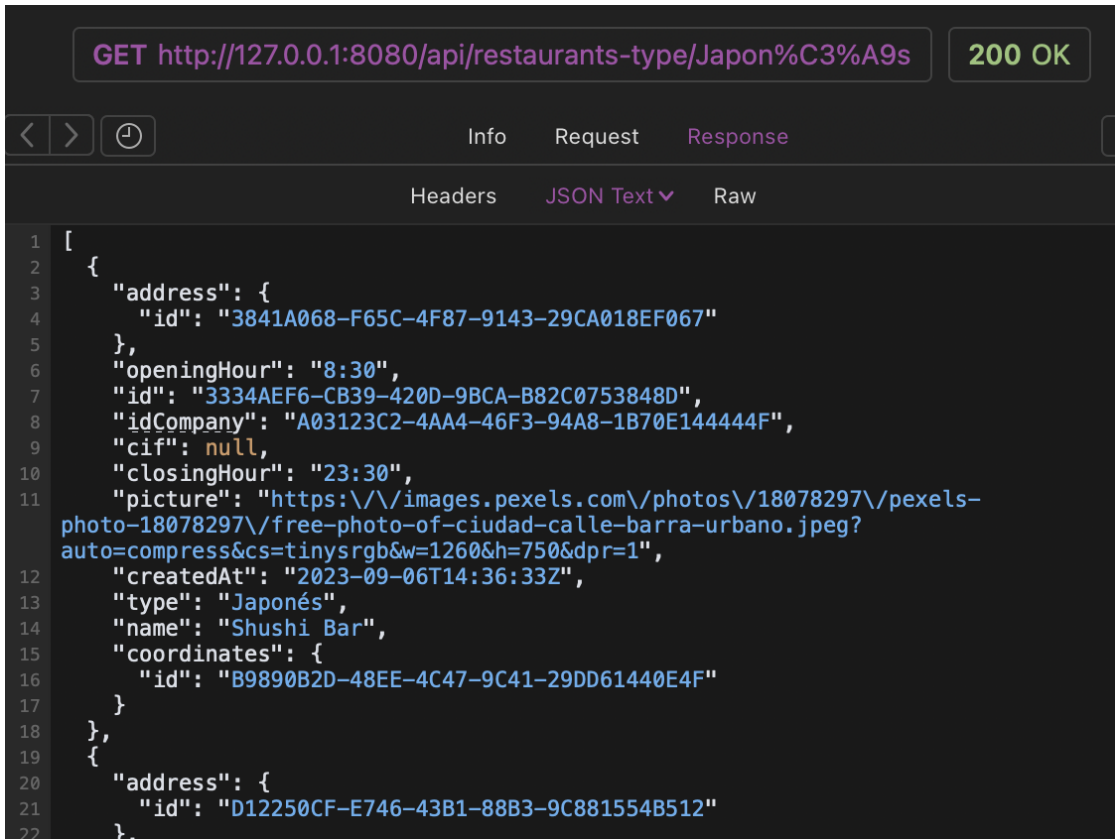
EndPoint	
<a href="http://127.0.0.1:8080/api/restaurants/idRestaurante">http://127.0.0.1:8080/api/restaurants/idRestaurante</a>	
Método	
Get	
Headers	
CDS-ApiKey	El valor de la ApiKey.
Authorization	Bearer accessToken.
Respuesta	
 <pre>1 { 2   "address": { 3     "id": "A260CB96-CD27-4004-9DF4-58800622CF95" 4   }, 5   "type": "Japonés", 6   "id": "06DD3BD3-5C4E-46B4-BFF5-536BA2E1F17E", 7   "idCompany": "04B7697B-B445-42FD-BADA-B6A8D2911705", 8   "cif": null, 9   "closingHour": "23:31", 10  "picture": "https:\\\\images.pexels.com\\photos\\6192571\\pexels- 11 photo-6192571.jpeg?auto=compress&amp;cs=tinysrgb&amp;w=1260&amp;h=750&amp;dpr=1", 12  "coordinates": { 13    "id": "053ADC4B-69D6-4871-9B97-4C3F8E7757D5" 14  }, 15  "openingHour": "8:32", 16  "createdAt": "2023-09-12T11:03:10Z", 17  "name": "Sakura, 桜" 18 }</pre>	

## GetRestaurantsByCompany

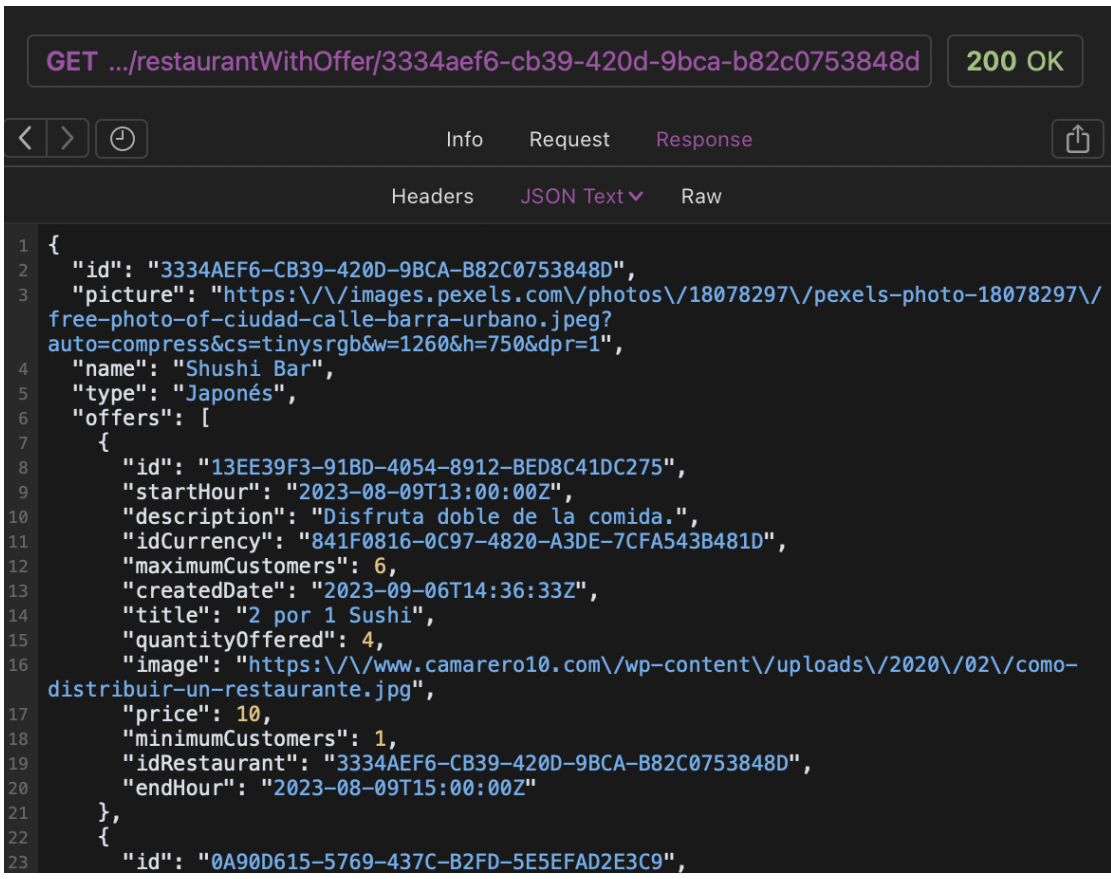
EndPoint	
<a href="http://127.0.0.1:8080/api/restaurantsCompany/idCompany">http://127.0.0.1:8080/api/restaurantsCompany/idCompany</a>	
Método	
Get	
Headers	
CDS-ApiKey	El valor de la ApiKey.
Authorization	Bearer accessToken.
Respuesta	
	



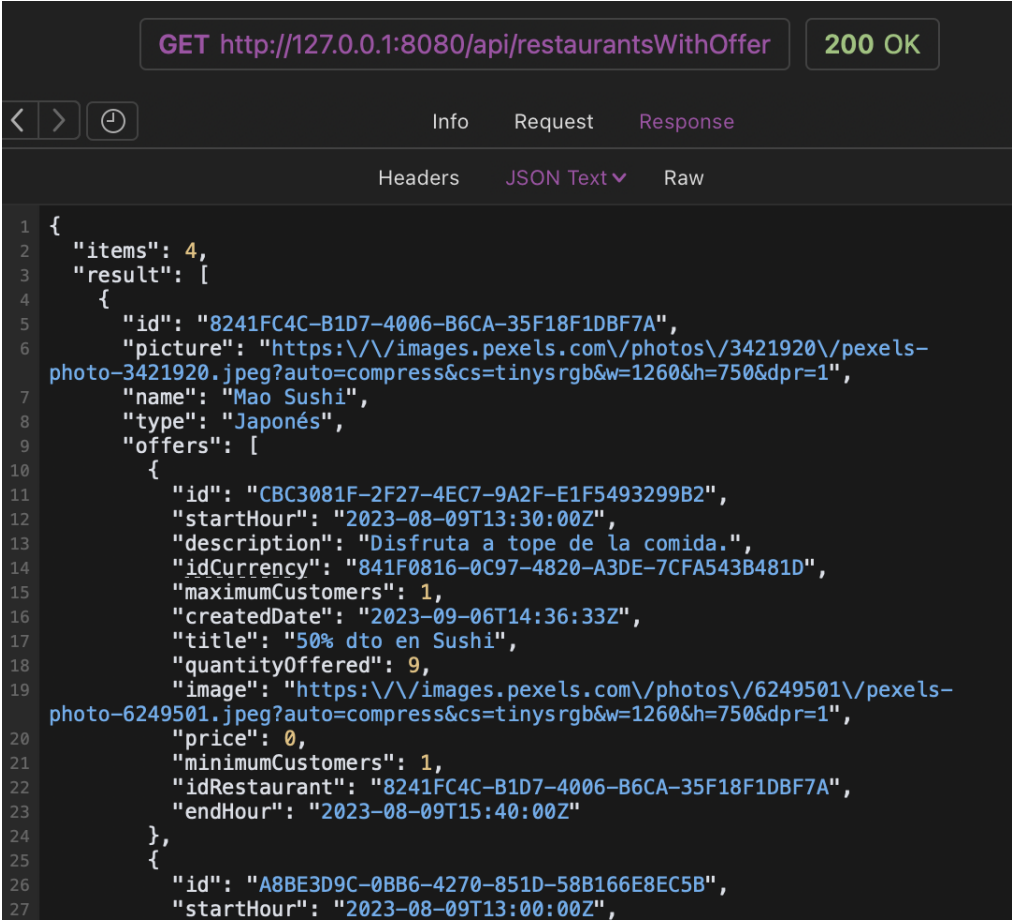
## GetRestaurantsByType

EndPoint	
<a href="http://127.0.0.1:8080/api/restaurants-type/restaurantType">http://127.0.0.1:8080/api/restaurants-type/restaurantType</a> Ej.: restaurantType = Japonés	
Método	
Get	
Headers	
CDS-APIKey	El valor de la ApiKey.
Authorization	Bearer accessToken.
Respuesta	
 <p>The screenshot shows a REST client interface with a GET request to <code>http://127.0.0.1:8080/api/restaurants-type/Japon%C3%A9s</code> returning a <code>200 OK</code> status. The response is displayed in JSON format, showing an array of restaurant objects. The first object in the array contains details for a restaurant named 'Shushi Bar', including its address, opening and closing hours, ID, company ID, CIF, picture URL, creation time, type, and coordinates.</p> <pre>1 [ 2   { 3     "address": { 4       "id": "3841A068-F65C-4F87-9143-29CA018EF067" 5     }, 6     "openingHour": "8:30", 7     "id": "3334AEF6-CB39-420D-9BCA-B82C0753848D", 8     "idCompany": "A03123C2-4AA4-46F3-94A8-1B70E144444F", 9     "cif": null, 10    "closingHour": "23:30", 11    "picture": "https:\\\\images.pexels.com\\photos\\18078297\\pexels- 12    photo-18078297\\free-photo-of-ciudad-calle-barra-urbano.jpeg? 13    auto=compress&amp;cs=tinysrgb&amp;w=1260&amp;h=750&amp;dpr=1", 14    "createdAt": "2023-09-06T14:36:33Z", 15    "type": "Japonés", 16    "name": "Shushi Bar", 17    "coordinates": { 18      "id": "B9890B2D-48EE-4C47-9C41-29DD61440E4F" 19    } 20  }, 21  { 22    "address": { 23      "id": "D12250CF-E746-43B1-88B3-9C881554B512" 24    } 25  } 26 ]</pre>	

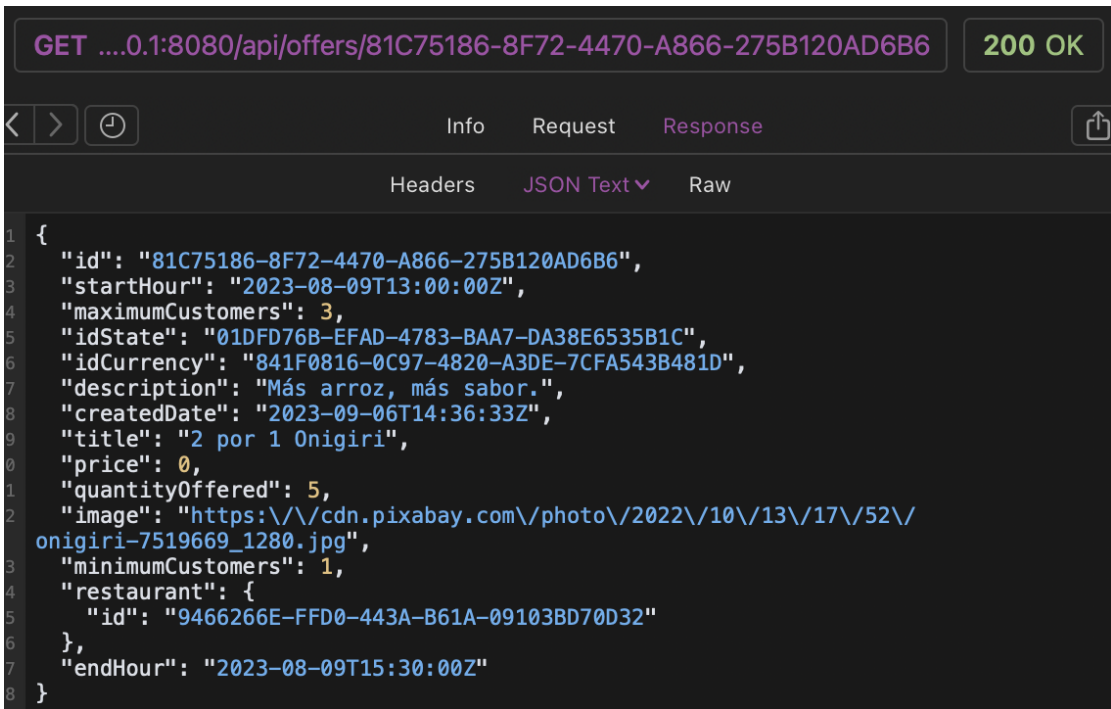
## getRestaurantWithOffers

EndPoint	
<a href="http://127.0.0.1:8080/api/restaurantWithOffer/idRestaurant">http://127.0.0.1:8080/api/restaurantWithOffer/idRestaurant</a>	
Método	
Get	
Headers	
CDS-APIKey	El valor de la ApiKey.
Authorization	Bearer accessToken.
Respuesta	
 <pre>1 { 2   "id": "3334AEF6-CB39-420D-9BCA-B82C0753848D", 3   "picture": "https://images.pexels.com/photos/18078297/pexels-photo-18078297/free-photo-of-ciudad-calle-barra-urbano.jpeg?auto=compress&amp;cs=tinysrgb&amp;w=1260&amp;h=750&amp;dpr=1", 4   "name": "Shushi Bar", 5   "type": "Japonés", 6   "offers": [ 7     { 8       "id": "13EE39F3-91BD-4054-8912-BED8C41DC275", 9       "startHour": "2023-08-09T13:00:00Z", 10      "description": "Disfruta doble de la comida.", 11      "idCurrency": "841F0816-0C97-4820-A3DE-7CFA543B481D", 12      "maximumCustomers": 6, 13      "createdDate": "2023-09-06T14:36:33Z", 14      "title": "2 por 1 Sushi", 15      "quantityOffered": 4, 16      "image": "https://www.camarero10.com/wp-content/uploads/2020/02/como-distribuir-un-restaurante.jpg", 17      "price": 10, 18      "minimumCustomers": 1, 19      "idRestaurant": "3334AEF6-CB39-420D-9BCA-B82C0753848D", 20      "endHour": "2023-08-09T15:00:00Z" 21    }, 22    { 23      "id": "0A90D615-5769-437C-B2FD-5E5EFAD2E3C9",</pre>	

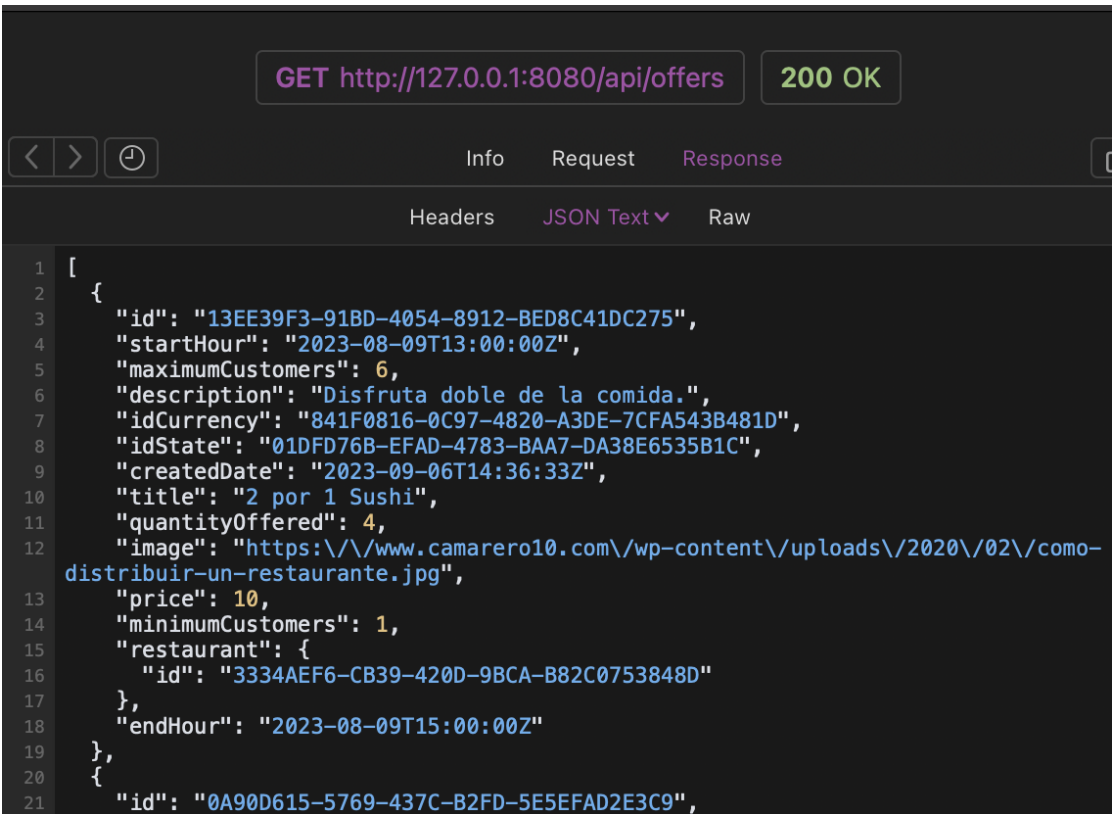
## getRestaurantsWithOffers

EndPoint	
<a href="http://127.0.0.1:8080/api/restaurantsWithOffer">http://127.0.0.1:8080/api/restaurantsWithOffer</a>	
Método	
Get	
Headers	
CDS-APIKey	El valor de la ApiKey.
Authorization	Bearer accessToken.
Respuesta	
 <pre>1 { 2   "items": 4, 3   "result": [ 4     { 5       "id": "8241FC4C-B1D7-4006-B6CA-35F18F1DBF7A", 6       "picture": "https://images.pexels.com/photos/3421920/pexels- 7 photo-3421920.jpeg?auto=compress&amp;cs=tinysrgb&amp;w=1260&amp;h=750&amp;dpr=1", 8       "name": "Mao Sushi", 9       "type": "Japonés", 10      "offers": [ 11        { 12          "id": "CBC3081F-2F27-4EC7-9A2F-E1F5493299B2", 13          "startHour": "2023-08-09T13:30:00Z", 14          "description": "Disfruta a tope de la comida.", 15          "idCurrency": "841F0816-0C97-4820-A3DE-7CFA543B481D", 16          "maximumCustomers": 1, 17          "createdDate": "2023-09-06T14:36:33Z", 18          "title": "50% dto en Sushi", 19          "quantityOffered": 9, 20          "image": "https://images.pexels.com/photos/6249501/pexels- 21 photo-6249501.jpeg?auto=compress&amp;cs=tinysrgb&amp;w=1260&amp;h=750&amp;dpr=1", 22          "price": 0, 23          "minimumCustomers": 1, 24          "idRestaurant": "8241FC4C-B1D7-4006-B6CA-35F18F1DBF7A", 25          "endHour": "2023-08-09T15:40:00Z" 26        }, 27        { 28          "id": "A8BE3D9C-0BB6-4270-851D-58B166E8EC5B", 29          "startHour": "2023-08-09T13:00:00Z",</pre>	

## getOfferById

EndPoint	
<a href="http://127.0.0.1:8080/api/offers/idOffer">http://127.0.0.1:8080/api/offers/idOffer</a>	
Método	
Get	
Headers	
CDS-APIKey	El valor de la ApiKey.
Authorization	Bearer accessToken.
Respuesta	
	

## getOffers

EndPoint	
<a href="http://127.0.0.1:8080/api/offers">http://127.0.0.1:8080/api/offers</a>	
Método	
Get	
Headers	
CDS-APIKey	El valor de la ApiKey.
Authorization	Bearer accessToken.
Respuesta	
 <pre>1 [ 2   { 3     "id": "13EE39F3-91BD-4054-8912-BED8C41DC275", 4     "startHour": "2023-08-09T13:00:00Z", 5     "maximumCustomers": 6, 6     "description": "Disfruta doble de la comida.", 7     "idCurrency": "841F0816-0C97-4820-A3DE-7CFA543B481D", 8     "idState": "01DFD76B-EFAD-4783-BAA7-DA38E6535B1C", 9     "createdDate": "2023-09-06T14:36:33Z", 10    "title": "2 por 1 Sushi", 11    "quantityOffered": 4, 12    "image": "https://www.camarero10.com/wp-content/uploads/2020/02/como- distribuir-un-restaurante.jpg", 13    "price": 10, 14    "minimumCustomers": 1, 15    "restaurant": { 16      "id": "3334AEF6-CB39-420D-9BCA-B82C0753848D" 17    }, 18    "endHour": "2023-08-09T15:00:00Z" 19  }, 20  { 21    "id": "0A90D615-5769-437C-B2FD-5E5EFAD2E3C9",</pre>	