Python SDK Version 2

-

Zoho CRM

-zoho.com/crm-

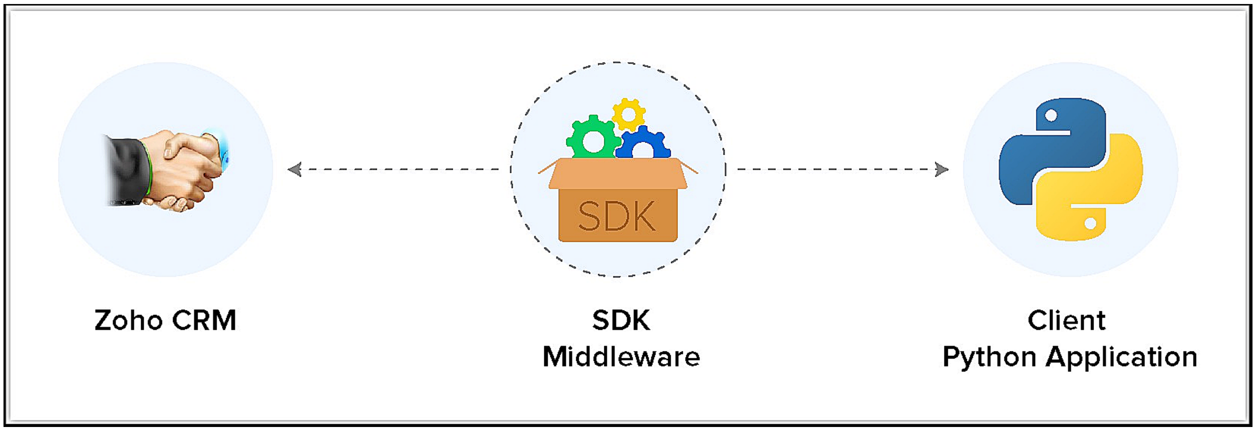
Table of Contents

Overview

Python SDK offers a way to create client python applications that can be integrated with Zoho CRM. This SDK makes the access and use of necessary CRM APIs with ease. In other words, it serves as a wrapper for the REST APIs, making it easier to use the services of Zoho CRM.

A point to note would be that the developer of the client application should create programming code elements along with configuration-related properties files, interface implementations, instances or objects. Authentication to access Zoho CRM APIs is through OAuth authentication mechanism. Invariably, HTTP requests and responses are taken care by SDK.

A sample of how an SDK acts a middle ware or interface between Zoho CRM and a client Python application.



Python SDK allows you to:

1. Exchange data between Zoho CRM and the client application where the CRM entities are modelled as classes.
2. CRM API equivalents are declared and defined as simple member methods in your Python application.
3. Push data into Zoho CRM, by accessing appropriate APIs of the CRM Service.

Environmental Setup

Python SDK is installable through "pip". Pip is a tool for dependency management in Python. The SDK requires the following from the client app:

- Client app must have Python 2.7 or above.
    - https://www.python.org/downloads/

- Client app must have Python requests being installed
    - https://2.python-requests.org/en/master/user/install/

- Python SDK must be installed through pip.
- The method ZCRMRestClient.initialize() must be called on starting up of your application

Note

- The access and refresh tokens are environment-specific and domain-specific. When you handle various environments and domains such as Production, Sandbox, or Developer and IN, CN, US, EU, or AU, respectively, you must use the access token and refresh token generated only in those respective environments and domains. The SDK throws an error, otherwise.
- For example, if you generate the tokens for your Sandbox environment in the CN domain, you must use only those tokens for that domain and environment. You cannot use the tokens generated for a different environment or a domain.

Installation of SDK

The SDK can be installed for your client application through Pip. Pip is a tool for dependency management in Python.

Installing Pip

Please refer the document below to install pip

```
https://pip.pypa.io/en/stable/installing/
```

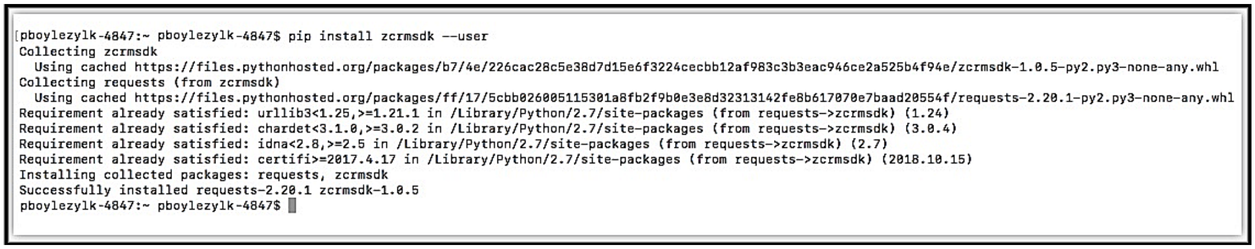
Installing the SDK

Here's how you install the Python SDK

- Run the command below:

```
pip install zcrmsdk
```

Note:
You also need to have Python requests installed. Add the code "pip install requests", to install it.



```
pboylezylk-4847:~ pboylezylk-4847$ pip install zcrmsdk --user
Collecting zcrmsdk
  Using cached https://files.pythonhosted.org/packages/b7/4e/226cac28c5e38d7d15e6f3224cecbb12af983c3b3eac946ce2a525b4f94e/zcrmsdk-1.0.5-py2.py3-none-any.whl
Collecting requests (from zcrmsdk)
  Using cached https://files.pythonhosted.org/packages/ff/17/5cbb026005115301a8fb2f9b0e3e8d32313142fe8b617070e7baad20554f/requests-2.20.1-py2.py3-none-any.whl
Requirement already satisfied: urllib3<1.25,>=1.21.1 in /Library/Python/2.7/site-packages (from requests->zcrmsdk) (1.24)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in /Library/Python/2.7/site-packages (from requests->zcrmsdk) (3.0.4)
Requirement already satisfied: idna<2.8,>=2.5 in /Library/Python/2.7/site-packages (from requests->zcrmsdk) (2.7)
Requirement already satisfied: certifi>=2017.4.17 in /Library/Python/2.7/site-packages (from requests->zcrmsdk) (2018.10.15)
Installing collected packages: requests, zcrmsdk
Successfully installed requests-2.20.1 zcrmsdk-1.0.5
pboylezylk-4847:~ pboylezylk-4847$
```

Python SDK will be installed and a package named 'zcrmsdk' will be created in the installation directory of python (ex. '/Library/Python/2.7/site-packages').

Upgrade the SDK

- Run this command to upgrade the Python SDK to the latest version.

```
pip install --upgrade zcrmsdk
```

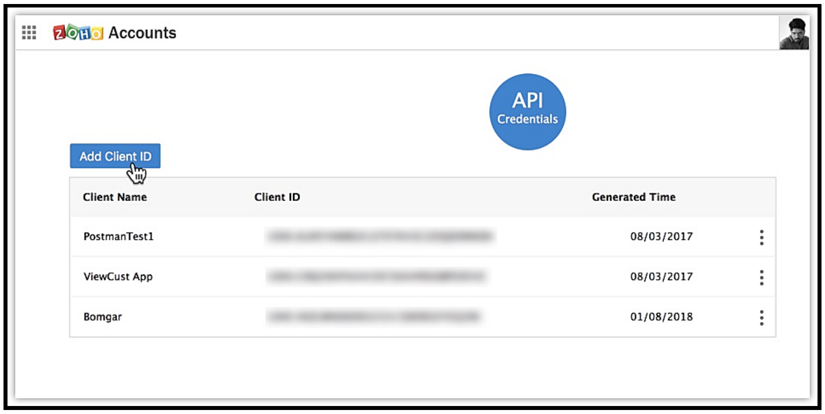
Register your application

All the Zoho CRM APIs are authenticated with OAuth2 standards, so it is mandatory to register and authenticate your client app with Zoho.

To register:

1. Go to the site: **https://api-console.zoho.com**

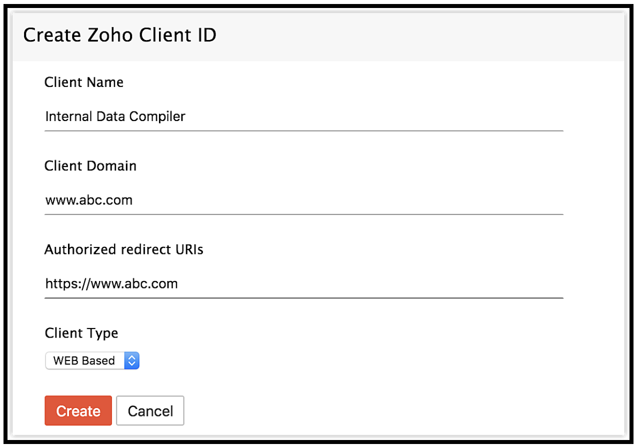
2. Click **Add Client ID**.



Client Name	Client ID	Generated Time
PostmanTest1	[redacted]	08/03/2017
ViewCust App	[redacted]	08/03/2017
Bomgar	[redacted]	01/08/2018

3. Enter the **Client Name**, **Client Domain** and **Authorized Redirect URL**.

4. Select the **Client Type** as **Web based**


Create Zoho Client ID

Client Name: Internal Data Compiler

Client Domain: www.abc.com

Authorized redirect URIs: https://www.abc.com

Client Type: WEB Based

5. Click **Create**.

6. Your Client app would have been created and displayed by now.

7. The newly registered app's Client ID and Client Secret can be found by clicking **Options → Edit.**

> **Note**:
> Options is the three dot icon at the right corner.

Registered applications will receive the following credentials:

- Client id – The consumer key generated from the connected app.
- Client Secret – The consumer secret generated from the connected app.
- Redirect URI – The Callback URL that you registered during the app registration.

Import SDK in your Project

- To import the SDK in your project, use the following statement.

```
import zcrmsdk
```

- To import the SDK and rename it with a custom name, use the following statement.

```
import zcrmsdk as your_custom_name
```

> **Note**
> You must use this SDK name while referencing any class in the SDK.

A few examples:

- To initialize the configuration properties file, use

```
zcrmsdk.ZCRMRestClient.initialize(config)
```

- To get the instance of the Contacts module, use

```
module_ins = crmsdk.ZCRMModule.get_instance('Contacts') #
```

```
    crmsdk is the custom name
```

Configuration

Before creating your python application, you need to authenticate the app with Zoho by

- Passing a configuration dictionary, and then calling *ZCRMRestClient.initialize(config);*

> **Note**:
> Please note that the configuration files (.properties files) under resources folder are not supported from ZCRMSDK - version 2.0.0 onwards.

You must pass the Configuration dictionary during initialization as *ZCRMRestClient.initialize(configuration)*. System throws an error, otherwise. Refer to the Configuration Dictionary table below for the key-value pairs.

Configuration Dictionary
You must pass the configuration values as python dictionary (key-value pair) as argument when you call the ZCRMRestClient.initialize(config) function. Below is the list of keys that should be present in the dictionary.

Keys	File Persistence	MySQL Persistence	Custom DB Persistence
Mandatory	client_id	client_id	client_id
	client_secret	client_secret	client_secret
	redirect_uri	redirect_uri	redirect_uri
	token_persistence_	mysql_username	persistence_handler

	path		_class
	-	mysql_password	persistence_handler_path
	-	mysql_port	-
Optional	applicationLogFilePath	applicationLogFilePath	applicationLogFilePath
	sandbox	sandbox	sandbox
	access_type	access_type	access_type
	accounts_url	accounts_url	accounts_url
	apiBaseUrl	apiBaseUrl	apiBaseUrl
	apiVersion	apiVersion	apiVersion
	currentUserEmail	currentUserEmail	currentUserEmail

Keys

- **client_id, client_secret, and redirect_uri**: Your OAuth client's configurations that you get after registering your Zoho client.
- **access_type**: Must be set to offline as the SDK does not support online OAuth client, as of now.
- **token_persistence_path**: The path to store the OAuth related tokens in file. If this is set, then no need of database for persistence; it happens through file only.
- **applicationLogFilePath**: The absolute path to log the exceptions during the usage

of the SDK.

- **sandbox**: Boolean key to make API calls to the sandbox environment. The value true makes API calls to the sandbox. Default value is false.
- **accounts_url**: The domain-specific accounts URL from which you generated the tokens.
    - For US: https://accounts.zoho.com
    - For EU: https://accounts.zoho.eu
    - For CN: https://accounts.zoho.com.cn
    - For IN: https://accounts.zoho.in
- apiBaseUrl: The domain-specific API URL from which you make API calls. For users from domains other than the US, this key is mandatory.
    - For US: https://www.zohoapis.com
    - For EU: https://www.zohoapis.eu
    - For CN: https://www.zohoapis.com.cn
    - For IN: https://www.zohoapis.in
- currentUserEmail: The email ID of the current user. When this is not set, the SDK throws an exception.

Note:

- If the **Optional keys** are not specified, their default values will be assigned automatically.
- In case the "token_persistence_path" key is not provided with any value, the system automatically sets the persistence in a database, instead of a file. At that point you would have to provide additional MySQL or custom DB parameters.
- You must use the same domain (.com, .eu, .com.cn, or .in) to generate tokens

Below is an example of a Python dictionary containing the mandatory keys.

```
config = {
"client_id":"1000.3xxxxxxxxxxxxxxxxxxxxxxxxX0YW",
"client_secret":"29xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx7e32",
"redirect_uri":"https://www.abc.com",
}
```

Below is an example of a Python dictionary containing all the keys.

```
config = {
"apiBaseUrl":"https://www.zohoapis.com",
"apiVersion":"v2",
"currentUserEmail":"email@gmail.com"
"sandbox":"False"
"applicationLogFilePath":"",
"client_id":"1000.3xxxxxxxxxxxxxxxxxxxxxxxxX0YW",
"client_secret":"29xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx7e32",
"redirect_uri":"https://www.abc.com",
"accounts_url":"https://accounts.zoho.com",
"token_persistence_path":"",
"access_type":"online",
//Use the below keys for MySQL DB persistence
"mysql_username":"",
"mysql_password":"",
"mysql_port":"3306",
//Use the below keys for custom DB persistence
"persistence_handler_class" : "Custom",
"persistence_handler_path": "/Users/Zoho/Desktop/PythonSDK/CustomPersistance.py"
}
```

Note:
In case "token_persistence_path" or "persistence_handler_path" is empty and the user fails to give the mysql parameters, the default values will be assumed. The default value for "mysql_username" will be "root", password will be left empty, and the "port" will be "3306".

SQL Database Persistence

In case you want to use the DB persistence and you have a MySQL DB ready, you need to install the following package through "pip".

```
pip install mysql-connector
```

Once these packages are installed, you can use the mysql fields (keys) in the configuration dictionary to achieve persistence. If **token_persistence_path** is provided, then persistence happens only in the file. In this case, there is no need of MySQL.

Create an empty file with the name **zcrm_oauthtokens.pkl** in the mentioned token_persistence_path.

Custom DB Persistence

Python ZCRMSDK Version 2.0.0 supports token persistence to store your tokens in your custom DB.

To use custom DB persistence, you must

- Implement AbstractBaseClass (**AbstractZohoOAuthPersistence** from OAuthClient.py).
- Write custom implementation of the following functions:
    - **get_oauthtokens(user_email)**
    - **save_oauthtokens(oauthtokens)**
    - **delete_oauthtokens(user_email)**
- Include the following new keys in configuration:
    - **persistence_handler_class** - ClassName of the custom implementation.
    - **persistence_handler_path** - Absolute path of the module(.py file).

> Note
> The access and refresh tokens are environment-specific and domain-specific. When you handle various environments and domains such as Production, Sandbox, or Developer and IN, CN, US, EU, or AU, respectively, you must use the access token and refresh token generated only in those respective environments and domains. The SDK throws an error, otherwise.
> For example, if you generate the tokens for your Sandbox environment in the CN domain, you must use only those tokens for that domain and environment. You cannot use the tokens generated for a different environment or a domain.

Initialize the Application

Now the app is ready to be initialized after defining configuration file/dictionary for your app.

Generating grant and refresh tokens

For a Single User(self-authorized)

For self client apps, the self authorized grant token should be generated from the Zoho Developer Console (https://accounts.zoho.com/developerconsole). The developer console has an option to generate grant token for a user directly. This option may be handy when your app is going to use only one CRM user's credentials for all its operations or for your development testing.

Login to the User's account.

Visit https://accounts.zoho.com/developerconsole

Click on the **Options** → **Self Client** option of the client for which you wish to authorize.

Enter one or more (comma separated) valid Zoho CRM scopes that you wish to authorize in the “Scope” field and choose the time of expiry. Provide “aaaserver.profile.READ” scope along with Zoho CRM scopes.

Copy the **grant token** that is displayed on the screen.

Generate refresh_token from grant token by making a POST request with the URL below:

```
https://accounts.zoho.com/oauth/v2/token?code={grant_token}
&redirect_uri={redirect_uri}&client_id={client_id}&client_s
ecret={client_secret}&grant_type=authorization_code
```

7. Copy the refresh token for backup.

Note:

- The generated grant token is valid only for the stipulated time you chose while generating it. Hence, the access and refresh tokens should be generated within that time.
- The OAuth client registration and grant token generation must be done in the same Zoho account's (meaning - login) developer console.

For Multiple Users

For multiple users, it is the responsibility of your client app to generate the grant token from the users trying to login.

- Your Application's UI must have a "**Login with Zoho**" option to open the grant token URL of Zoho, which would prompt for the user's Zoho login credentials.
- Upon successful login of the user, the grant token will be sent as a param to your registered redirect URL.

Generating access tokens

Access token can be generated by grant token or refresh token. Following any one of the two methods given below is sufficient.

The following code snippet should be executed from your main class to get access token.

```
ZCRMRestClient.initialize()
oauth_client = ZohoOAuth.get_client_instance()
grant_token="paste_grant_token_here"
oauth_tokens =
oauth_client.generate_access_token(grant_token)
```

Please paste the generated grant token in the string literal mentioned. This is one time process only.

Generating access tokens from refresh token

The following code snippet should be executed from your main class to get access token.

```
ZCRMRestClient.initialize()
oauth_client = ZohoOAuth.get_client_instance()
refresh_token="paste_refresh_token_here"
user_identifier="provide_user_identifier_like_email_here"
oauth_tokens =
oauth_client.generate_access_token_from_refresh_token(refre
sh_token,user_identifier)
```

Please paste the generated refresh token in the string literal mentioned. This is one time process only.

Note:

- The above code snippet is **valid only once per grant token**. Upon its successful execution, the generated access and refresh tokens would have been persisted through your persistence handler class.
- Once the OAuth tokens have been persisted, subsequent API calls would use the persisted access and refresh tokens. **The SDK will take care of refreshing the access token using refresh token, as and when required.**
- **The access and refresh tokens are environment-specific and domain-specific**. When you handle various environments and domains such as Production, Sandbox, or Developer and IN, CN, US, EU, or AU, respectively, you must use the access token and refresh token generated only in those respective environments and domains. The SDK throws an error, otherwise.
- For example, if you generate the tokens for your Sandbox environment in the CN domain, you must use only those tokens for that domain and environment. You cannot use the tokens generated for a different environment or a domain.

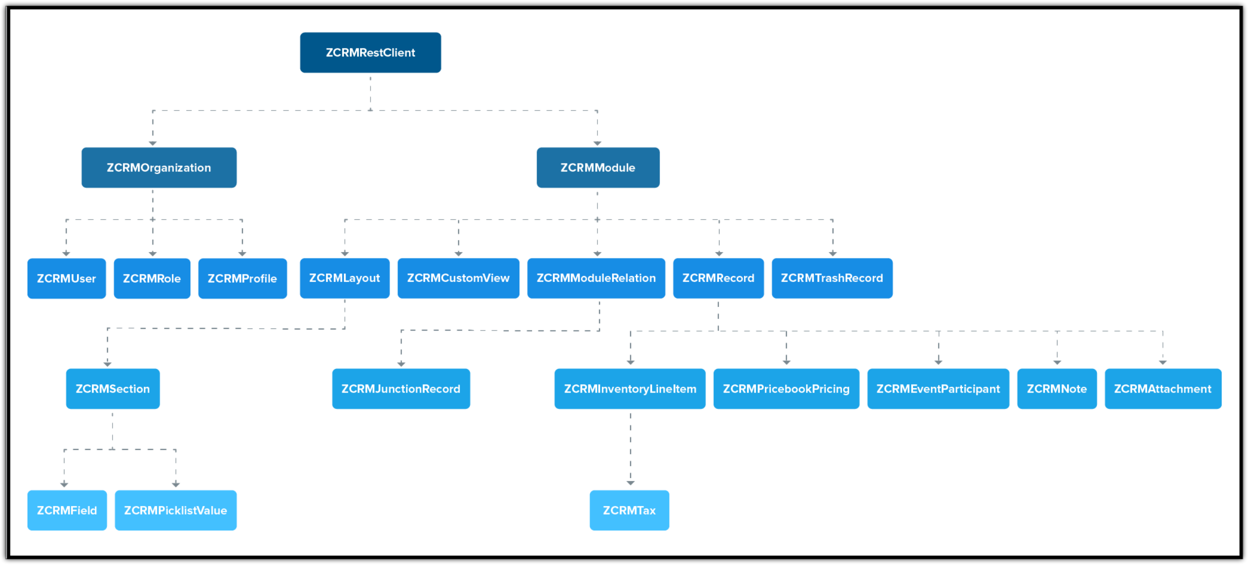
Class Hierarchy

All Zoho CRM entities are modelled as modules having classes, methods and instance variables applicable to that particular entity. ZCRMRestClient is the base class of the Python SDK. ZCRMRestClient has methods to get instances of various other Zoho CRM entities. It is in RestClient module.

- ZCRMRestClient is the base class of the SDK.
- This class has, methods to get instances of various other Zoho CRM entities.
- The class relations and hierarchy of the SDK follows the entity hierarchy inside Zoho CRM.

- Each class entity has functions to fetch its own properties and to fetch data of its immediate child entities through an API call.For example: a Zoho CRM module (ZCRMModule) object will have member functions to get a module's properties like display name, module Id, etc, and will also have functions to fetch all its child objects (like ZCRMLayout).

The class relations and hierarchy of the SDK follows the entity hierarchy inside Zoho CRM. The class hierarchy of various Zoho CRM entities are given below:



As appearing in the hierarchy, every entity class will have instance variables to fetch its own properties and to fetch data of its immediate child entities through an API call.

Instance Objects

It is not always effective to follow the complete class hierarchy from the top to fetch the data of an entity at some lower level, since this would involve API calls at each level. In order to handle this, every entity class will have a get_instance() method to get its own dummy object and instance variables to get dummy objects of its child entities.

Note:
getInstance() methods would not have any of its properties filled, since it would not fire an API call. This would just return a dummy object that shall be only used to access the non-static methods of the class.

Summing it up,

- **ZCRMRestClient.get_module("Contacts")** would return the actual Contacts module, that has all the properties of the Contacts module filled through an API call.

- **ZCRMRestClient.get_module_instance("Contacts")** would return a dummy ZCRMModule object that would refer to the Contacts module, with no properties filled, since this doesn't make an API call.

Hence, to get records from a module, you need not to start all the way from **ZCRMRestClient**. Instead, you could get a ZCRMModule instance with **ZCRMModule.get_instance()** and then invoke its nonstatic get_records() method from the created instance. This would avoid the API call which would have been triggered to populate the ZCRMModule object.

Accessing record properties

Since record properties are dynamic across modules, we have only given the common fields like createdTime, createdBy, owner etc, as ZCRMRecord's default members. All other record properties are available as a map in ZCRMRecord object.

To access the individual field values of a record, use the getter and setter methods available. The keys of the record properties map are the API names of the module's fields. API names of all fields of all modules are available under,

Setup → Marketplace → APIs → CRM API → API Names.

- To get a field value, use record.get_field_value(field_api_name);
- To set a field value, use record.set_field_value(field_api_name, new_value);
- While setting a field value, please make sure of that the set value is of the data type of the field to which you are going to set it.

Responses & Exceptions

APIResponse, **BulkAPIResponse** and **FileAPIResponse** are the wrapper objects for Zoho CRM APIs' responses. All methods that invoke the APIs return one of these three objects.

- A method-seeking entity would return **APIResponse** object, whereas a method-seeking list of entities would return **BulkAPIResponse** object.
- Use the instance variable "**data**" to get the entity data alone from the response wrapper objects. **APIResponse.data** returns a single Zoho CRM entity object, while

BulkAPIResponse.data returns a list of Zoho CRM entity objects.

- **FileAPIResponse** will be returned for file download APIs to download a photo or an attachment from a record or note such as **record.download_photo()**, **record.download_attachment(attachment_id)** etc.
  -

Other than data, these response wrappers have the following properties.

- **ResponseInfo** - any other information, if provided by the API, besides the actual data. It is available through the member **info.response.info**.
- **EntityResponse list** - status of the individual entities in a bulk API. For example, an insert records API may partially fail because of a few records. This array gives the individual records' creation status. It is available through member **bulk_entity_response:response.bulk_entity_response**

Start the App

The SDK requires the following line of code being invoked every time your app gets started.

```
ZCRMRestClient.initialize(configuration)
```

This method should be called from the main class of your python application to start the application. It needs to be invoked without any exception.

Check Exceptions

All unexpected behaviors like faulty API responses, SDK anomalies are handled by the SDK and are thrown only as a single exception — **ZCRMException**. Hence, it's enough to catch this exception alone in the client app code.

Sample Codes

All of Zoho CRM's APIs can be used through the Python SDK, to enable your custom application to perform data sync to the best degree. Here are the sample codes for all the API methods available in our SDK.

Rest Client Operations

These methods involve authentications procedures that are to be included in your application, to provide access to Zoho CRM's data.

Methods	Description
get_org_details	To fetch information about your CRM account organization.
get_current_user	To fetch information about the user who is currently accessing Zoho CRM's data through your application.
get_all_modules	To fetch information about all modules in your CRM account.
get_module	To fetch information about a particular module in your CRM account.

Organization Operations

These methods involve actions that can be performed in your application, to modify the data that pertains to your Zoho CRM's organization. For instance, you can get the list of all the users (employees) that are present in your organization at any point of time.

Methods	Description
get_user	To fetch information about a specific user in your CRM account.
get_all_users	To fetch the list of all the users from your CRM account.
	To fetch the list of all the active users in

get_all_active_users	your CRM account.
get_all_deactive_users	To fetch the list of all the non-active users in your CRM account.
get_all_active_confirmed_users	To fetch the list of all the active and confirmed users in your CRM account.
get_all_confirmed_users	To fetch the list of all the confirmed users in your CRM account.
get_all_not_confirmed_users	To fetch the list of all the users who are not confirmed in your CRM account.
get_all_deleted_users	To fetch the list of all the users who were deleted from your CRM account.
get_all_admin_users	To fetch the list of all the users who have admin level permissions in your CRM account.
get_all_active_confirmed_admin_users	To fetch the list of all the active and confirmed users who have admin level permissions in your CRM account.
create_user	To create a new user in your CRM account.
update_user	To update details of an existing user in your CRM account.

delete_user	To delete a user from your CRM account.
get_current_user	To fetch information on the user who is currently accessing Zoho CRM's data through your app.
get_profiles	To fetch the list of all the profiles that were created in your CRM account.
get_profile	To fetch information about a particular profile in your CRM account.
get_all_roles	To fetch information about a particular role in your CRM account.
get_role	To fetch information about a particular role in your CRM account.
get_variable_groups	To get the details of any variable group.
get_variables	To retrieve all the available variables for your organization.
create_variables	To create new variables in CRM.
update_variables	To update the details of variables in CRM.
get_organization_taxes	To fetch the list of all organization taxes.
get_organization_tax	To fetch information about a particular organization tax.

create_organization_taxes	To create a new organization tax.
update_organization_taxes	To update the details of a particular organization tax.
delete_organization_taxes	To delete all the taxes associated to the organization.
delete_organization_tax	To delete a particular tax associated to the organization.

Module Operations

These methods involve actions that can be performed in your application, to modify the data in your CRM at the module level. For instance, you can get all the records from a module, search for specific ones, delete them, and do more.

Methods	Description
get_fields	To fetch the list of all the fields that are available in a module.
get_field	To fetch information about a particular field available in a module.
get_layout	To fetch information about a particular layout of a module.
get_all_layouts	To fetch the list of all the layouts that are available for a module.

get_customview	To fetch information about a particular custom view of a module.
get_all_customviews	To fetch the list of all the custom views that are available for a module.
get_all_relatedlists	To fetch the list of all the related lists that are available for a module.
get_relatedlist	To fetch the list of all the related lists that are available for a module.
update_module_settings	To update the settings of a module. Ex: territory, custom view, etc.
get_records	To fetch the list of all the records that are available in a module.
search_records	To search for records in a module based on a Word(text).
search_records_by_phone	To search for records in a module based on the Phone number.
search_record_email	To search for records in a module based on Email address.
search_records_by_criteria	To search for records in a module based on a criteria specified by the user.
update_records	To update details of multiple records in a

	module.
create_records	To create a new record in a module.
delete_records	To delete existing records from a module.
get_deleted_records	To fetch the list of all the records that were deleted from a module.
get_recyclebin_records	To fetch the list of all the records that were deleted from a module and stored in the recycle bin.
get_permanently_deleted_records	To fetch the list of all the records that were permanently deleted from a module.
upsert_records	To insert/update records in a module.
mass_update_records	To update single field value for multiple records. Same value will get updated for all the mentioned records.
get_tags	To fetch the list of all the tags that were created for a module.
get_tag_count	To fetch the total count of tags that were created for a module.
create_tags	To create new tags for a module.
update_tags	To update the details of existing tags for a module.

add_tags_to_multiple_records	To associate tags to records in a module.
remove_tags_from_multiple_records	To disassociate tags from records in a module.

Record Operations

These methods involve actions that can be performed in your application, to access or modify data that are stored in a particular record. You could fetch the details of a record, create new ones, update existing ones, upload notes, attachments, photos, etc.

Methods	Description
create_record	To create a new record.
update_record	To update an existing record.
delete_record	To delete an existing record.
convert_record	To convert a record (Leads to Contacts/Deals).
get_notes	To fetch the notes that were attached to a record.
add_note	To add a note to a record.
update_note	To update a note that was previously added to a record.

delete_note	To delete a note from a record.
get_attachments	To fetch the list of attachments of a record.
upload_attachment	To upload an attachment to a record.
upload_link_as_attachment	To upload a link as an attachment to a record.
download_attachment	To download an attachment that was uploaded to a record.
delete_attachment	To delete an attachment that was added to a record.
upload_photo	To upload a photo to a record.
download_photo	To download a photo that was added to a record.
delete_photo	To delete a photo that was added to a record.
add_relation	To make a relation between two records.
remove_relation	To remove a relation between two records.
get_related_records	To fetch the list of records in a related list.

get_blueprint	To get the next available transitions for that record, fields available for each transition, current value of each field, and validation(if any).
update_blueprint	To update a single transition at a time.
get_variable_group	To get the details of a specific variable group.
get_variable	To get the details of a specific variable.
update_variable	To update the details of a specific variable.
delete_variable	To delete a specific variable.
add_tags	To add tags to a record.
remove_tags	To remove tags from a record.
delete_tag	To remove tags from a record.
merge_tag	To merge two tags.

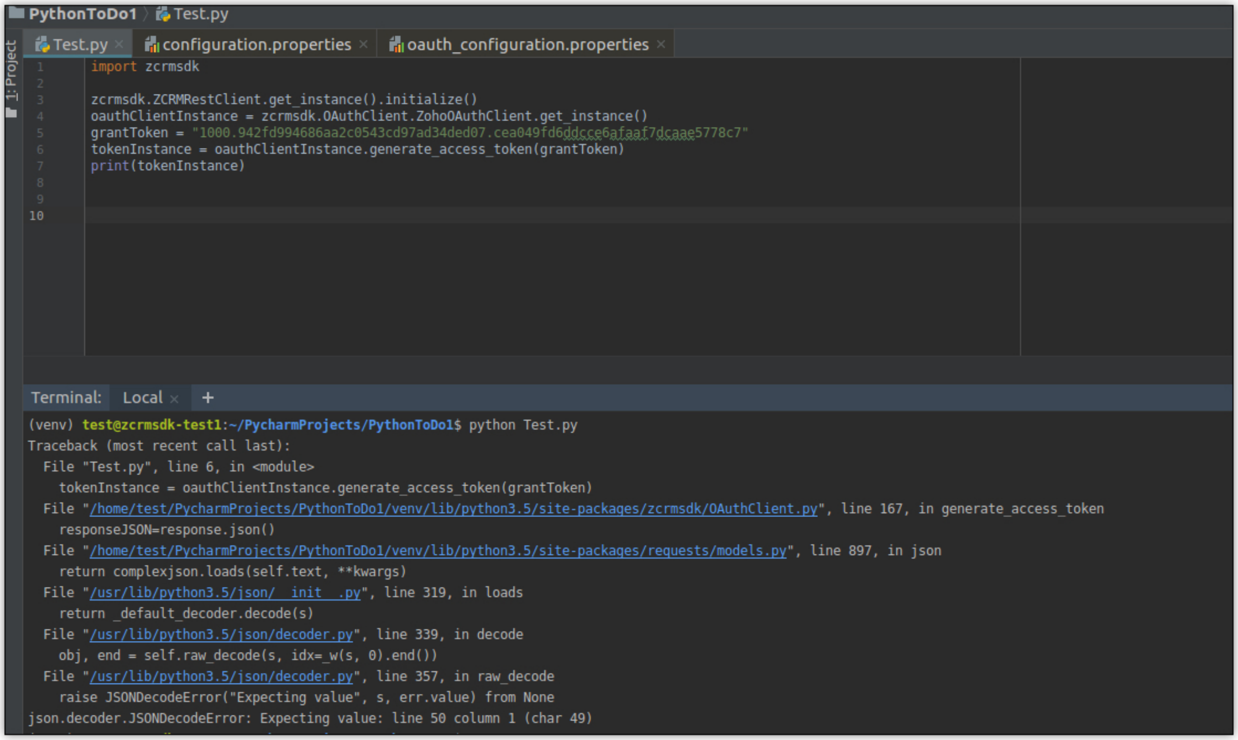
Note Operations

These methods involve actions that can be performed in your application, to access or modify notes of data in your CRM.

Methods	Description
note_get_attachments	To fetch all attachments of a note.
note_upload_attachment	To upload a note attachment to a record.
note_download_attachment	To download an attachment that was attached to a record.
note_delete_attachment	To delete an attachment that was attached to a record.

Errors and Solutions

No OAuth and Configuration Properties

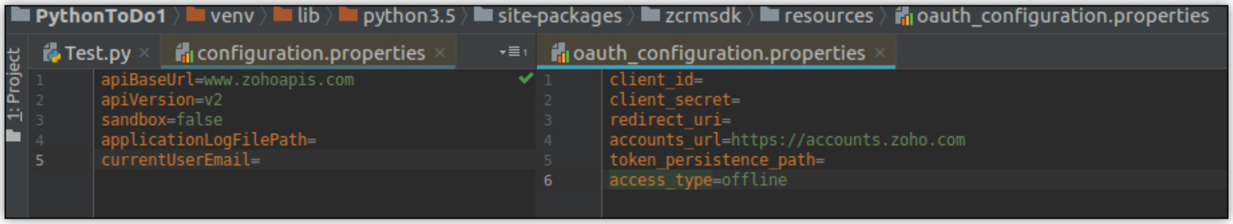


```python
import zcrmsdk

zcrmsdk.ZCRMRestClient.get_instance().initialize()
oauthClientInstance = zcrmsdk.OAuthClient.ZohoOAuthClient.get_instance()
grantToken = "1000.942fd994686aa2c0543cd97ad34ded07.cea049fd6ddcce6afaaf7dcaae5778c7"
tokenInstance = oauthClientInstance.generate_access_token(grantToken)
print(tokenInstance)
```

```
(venv) test@zcrmsdk-test1:~/PycharmProjects/PythonToDo1$ python Test.py
Traceback (most recent call last):
  File "Test.py", line 6, in <module>
    tokenInstance = oauthClientInstance.generate_access_token(grantToken)
  File "/home/test/PycharmProjects/PythonToDo1/venv/lib/python3.5/site-packages/zcrmsdk/OAuthClient.py", line 167, in generate_access_token
    responseJSON=response.json()
  File "/home/test/PycharmProjects/PythonToDo1/venv/lib/python3.5/site-packages/requests/models.py", line 897, in json
    return complexjson.loads(self.text, **kwargs)
  File "/usr/lib/python3.5/json/__init__.py", line 319, in loads
    return _default_decoder.decode(s)
  File "/usr/lib/python3.5/json/decoder.py", line 339, in decode
    obj, end = self.raw_decode(s, idx=_w(s, 0).end())
  File "/usr/lib/python3.5/json/decoder.py", line 357, in raw_decode
    raise JSONDecodeError("Expecting value", s, err.value) from None
json.decoder.JSONDecodeError: Expecting value: line 50 column 1 (char 49)
```

Reason

The user has not set the configuration properties or the OAuth properties.

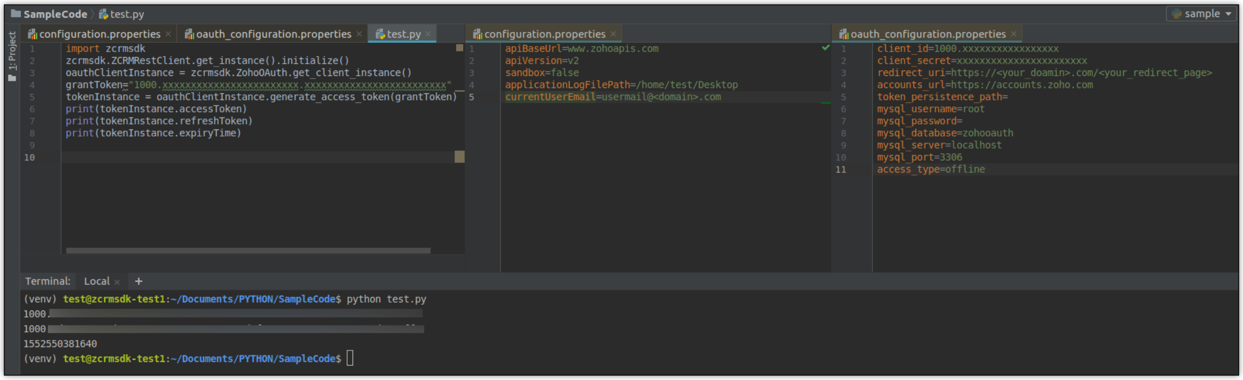

PythonToDo1 〉 venv 〉 lib 〉 python3.5 〉 site-packages 〉 zcrmsdk 〉 resources 〉 oauth_configuration.properties
1: Project
Test.py
configuration.properties
oauth_configuration.properties

```
apiBaseUrl=www.zohoapis.com
apiVersion=v2
sandbox=false
applicationLogFilePath=
currentUserEmail=
```

```
client_id=
client_secret=
redirect_uri=
accounts_url=https://accounts.zoho.com
token_persistence_path=
access_type=offline
```

Solution

Set configuration and OAuth configuration properties as displayed in the below image.



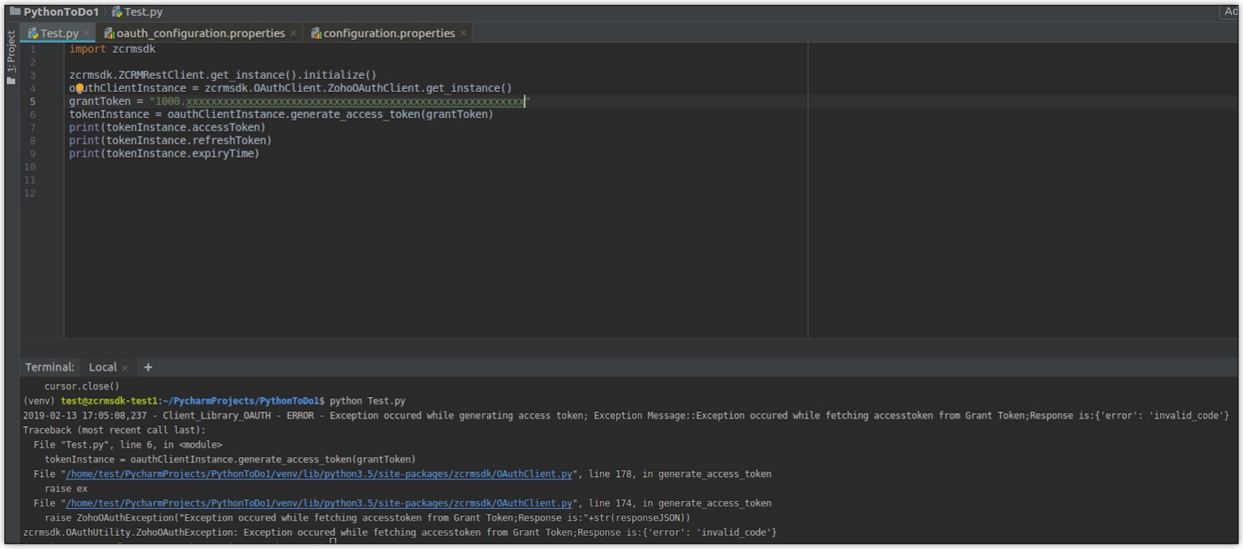
```
import zcrmsdk
zcrmsdk.ZCRMRestClient.get_instance().initialize()
oauthClientInstance = zcrmsdk.ZohoOAuth.get_client_instance()
grantToken="1000.xxxxxxxxxxxxxxxxxxxxxxxx.xxxxxxxxxxxxxxxxxxxxxxxxx"
tokenInstance = oauthClientInstance.generate_access_token(grantToken)
print(tokenInstance.accessToken)
print(tokenInstance.refreshToken)
print(tokenInstance.expiryTime)
```

```
apiBaseUrl=www.zohoapis.com
apiVersion=v2
sandbox=false
applicationLogFilePath=/home/test/Desktop
currentUserEmail=usermail@<domain>.com
```

```
client_id=1000.xxxxxxxxxxxxxxxxx
client_secret=xxxxxxxxxxxxxxxxxxxxxxx
redirect_uri=https://<your_doamin>.com/<your_redirect_page>
accounts_url=https://accounts.zoho.com
token_persistence_path=
mysql_username=root
mysql_password=
mysql_database=zohooauth
mysql_server=localhost
mysql_port=3306
access_type=offline
```

```
(venv) test@zcrmsdk-test1:~/Documents/PYTHON/SampleCode$ python test.py
1000.[redacted]
1000 [redacted]
1552550381640
(venv) test@zcrmsdk-test1:~/Documents/PYTHON/SampleCode$ 
```

Invalid Code



PythonToDo1 〉 Test.py

Test.py | oauth_configuration.properties | configuration.properties

```python
import zcrmsdk

zcrmsdk.ZCRMRestClient.get_instance().initialize()
oauthClientInstance = zcrmsdk.OAuthClient.ZohoOAuthClient.get_instance()
grantToken = "1000.xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
tokenInstance = oauthClientInstance.generate_access_token(grantToken)
print(tokenInstance.accessToken)
print(tokenInstance.refreshToken)
print(tokenInstance.expiryTime)
```

Terminal: Local

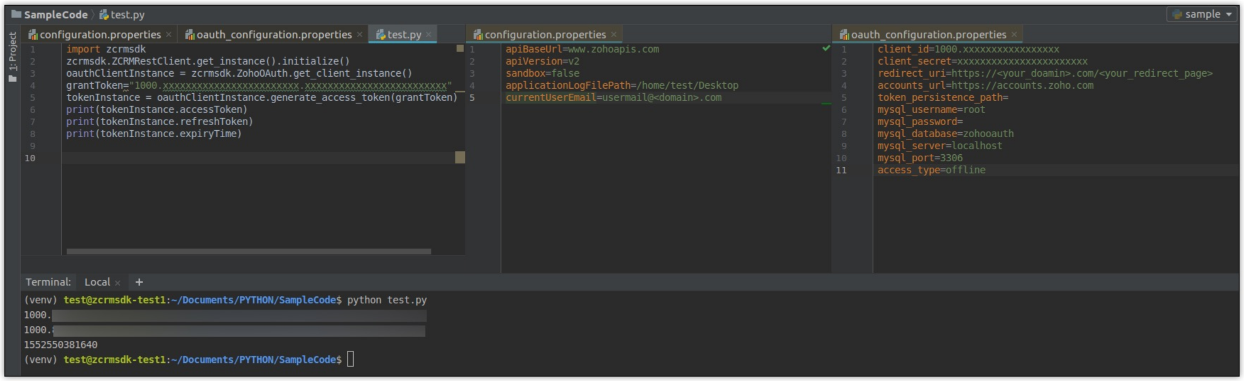
```
    cursor.close()
(venv) test@zcrmsdk-test1:~/PycharmProjects/PythonToDo1$ python Test.py
2019-02-13 17:05:08,237 - Client_Library_OAUTH - ERROR - Exception occured while generating access token; Exception Message::Exception occured while fetching accesstoken from Grant Token;Response is:{'error': 'invalid_code'}
Traceback (most recent call last):
  File "Test.py", line 6, in <module>
    tokenInstance = oauthClientInstance.generate_access_token(grantToken)
  File "/home/test/PycharmProjects/PythonToDo1/venv/lib/python3.5/site-packages/zcrmsdk/OAuthClient.py", line 178, in generate_access_token
    raise ex
  File "/home/test/PycharmProjects/PythonToDo1/venv/lib/python3.5/site-packages/zcrmsdk/OAuthClient.py", line 174, in generate_access_token
    raise ZohoOAuthException("Exception occured while fetching accesstoken from Grant Token;Response is:"+str(responseJSON))
zcrmsdk.OAuthUtility.ZohoOAuthException: Exception occured while fetching accesstoken from Grant Token;Response is:{'error': 'invalid_code'}
```

Reason

The user's grant token(authorization code) has expired.

Solution

Regenerate the grant token and use within the stipulated time.



SampleCode 〉 test.py

configuration.properties | oauth_configuration.properties | test.py

```python
import zcrmsdk
zcrmsdk.ZCRMRestClient.get_instance().initialize()
oauthClientInstance = zcrmsdk.ZohoOAuth.get_client_instance()
grantToken="1000.xxxxxxxxxxxxxxxxxxxxxxxxxx.xxxxxxxxxxxxxxxxxxxxxxxxxx"
tokenInstance = oauthClientInstance.generate_access_token(grantToken)
print(tokenInstance.accessToken)
print(tokenInstance.refreshToken)
print(tokenInstance.expiryTime)
```

configuration.properties

```
apiBaseUrl=www.zohoapis.com
apiVersion=v2
sandbox=false
applicationLogFilePath=/home/test/Desktop
currentUserEmail=usermail@<domain>.com
```

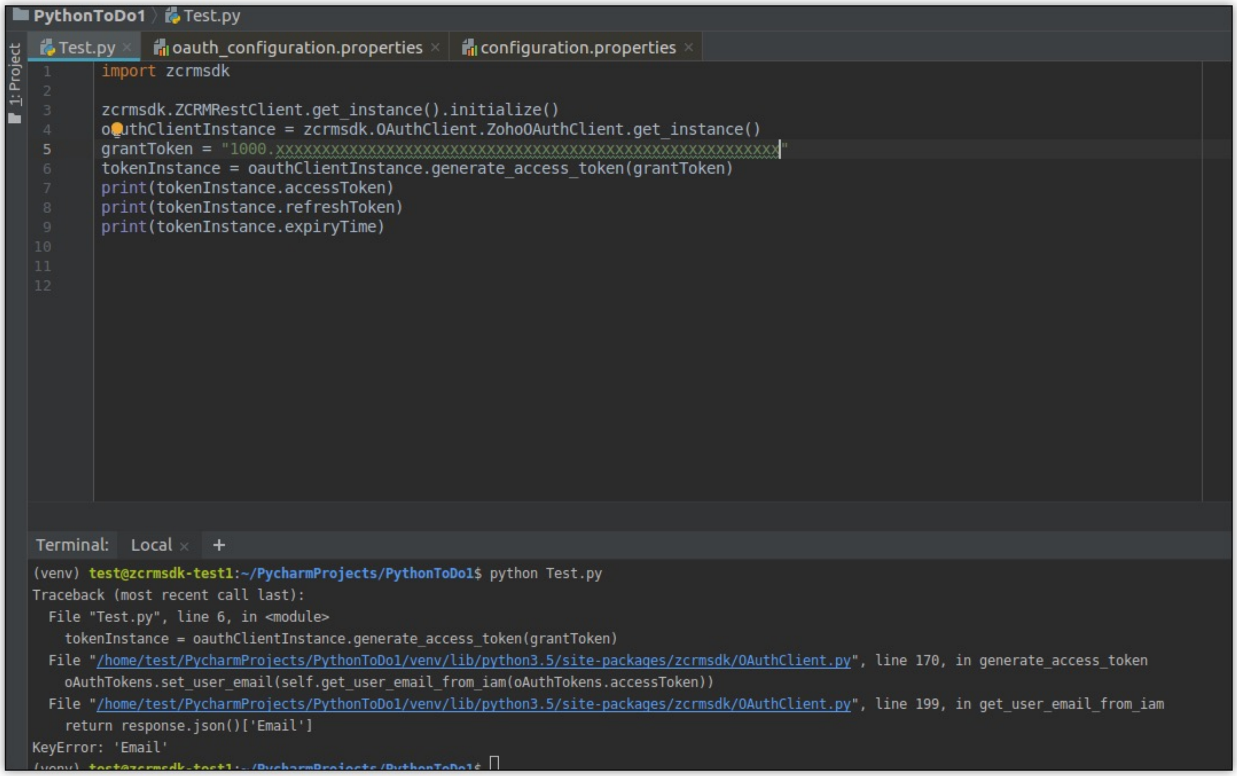
oauth_configuration.properties

```
client_id=1000.xxxxxxxxxxxxxxxxx
client_secret=xxxxxxxxxxxxxxxxxxxxxxxx
redirect_uri=https://<your_doamin>.com/<your_redirect_page>
accounts_url=https://accounts.zoho.com
token_persistence_path=
mysql_username=root
mysql_password=
mysql_database=zohooauth
mysql_server=localhost
mysql_port=3306
access_type=offline
```

Terminal: Local

```
(venv) test@zcrmsdk-test1:~/Documents/PYTHON/SampleCode$ python test.py
1000.[redacted]
1000.[redacted]
1552550381640
(venv) test@zcrmsdk-test1:~/Documents/PYTHON/SampleCode$ 
```

KeyError: 'Email'

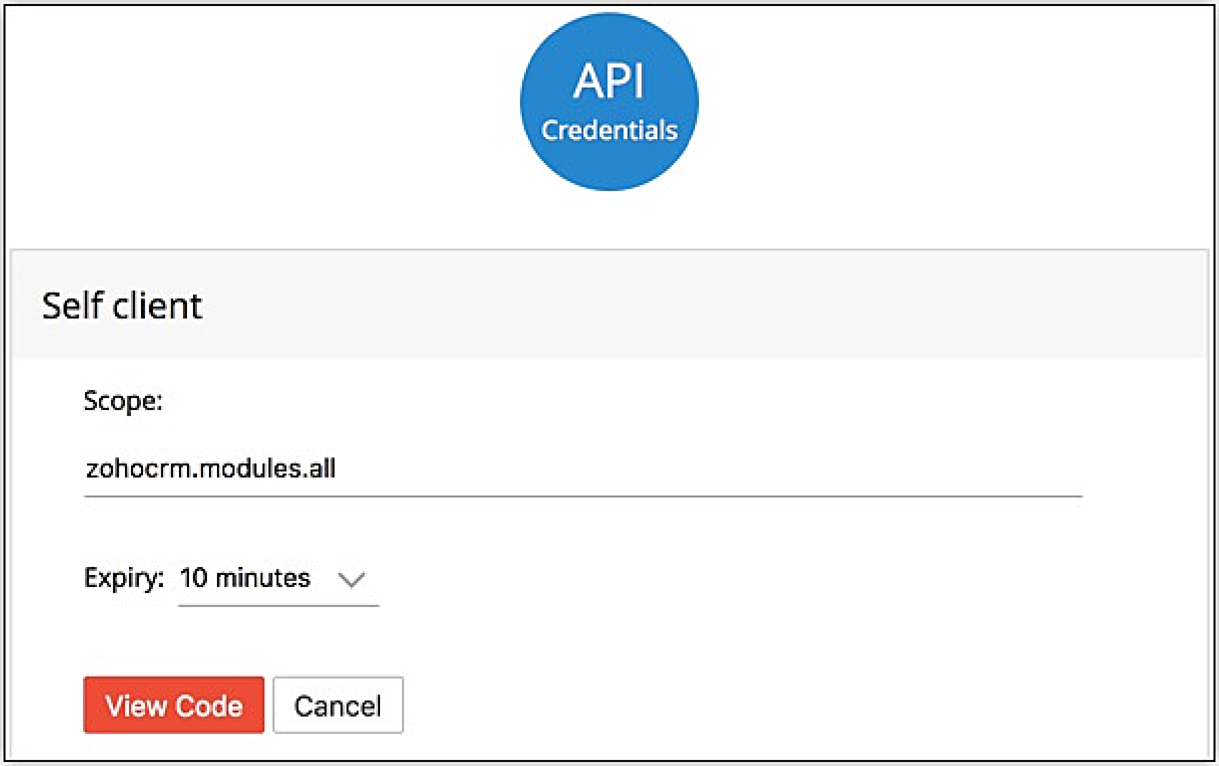


```python
import zcrmsdk

zcrmsdk.ZCRMRestClient.get_instance().initialize()
oauthClientInstance = zcrmsdk.OAuthClient.ZohoOAuthClient.get_instance()
grantToken = "1000.xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
tokenInstance = oauthClientInstance.generate_access_token(grantToken)
print(tokenInstance.accessToken)
print(tokenInstance.refreshToken)
print(tokenInstance.expiryTime)
```

```
(venv) test@zcrmsdk-test1:~/PycharmProjects/PythonToDo1$ python Test.py
Traceback (most recent call last):
  File "Test.py", line 6, in <module>
    tokenInstance = oauthClientInstance.generate_access_token(grantToken)
  File "/home/test/PycharmProjects/PythonToDo1/venv/lib/python3.5/site-packages/zcrmsdk/OAuthClient.py", line 170, in generate_access_token
    oAuthTokens.set_user_email(self.get_user_email_from_iam(oAuthTokens.accessToken))
  File "/home/test/PycharmProjects/PythonToDo1/venv/lib/python3.5/site-packages/zcrmsdk/OAuthClient.py", line 199, in get_user_email_from_iam
    return response.json()['Email']
KeyError: 'Email'
(venv) test@zcrmsdk-test1:~/PycharmProjects/PythonToDo1$
```

Reason

The user has not included the scope “Aaaserver.profile.Read” while generating the grant token(authorization code).



Self client

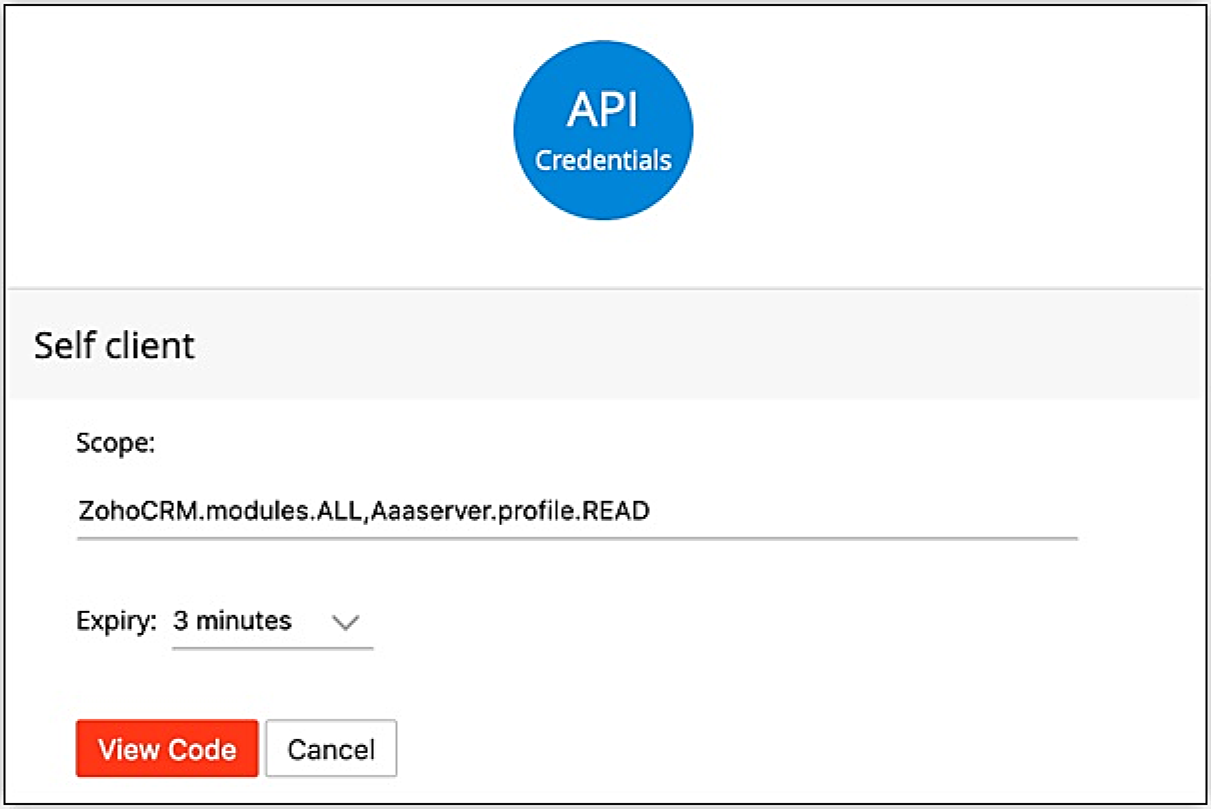
Scope:

zohocrm.modules.all

Expiry: 10 minutes

Solution

Include the scope “Aaaserver.profile.Read” while generating the grant token(authorization code).



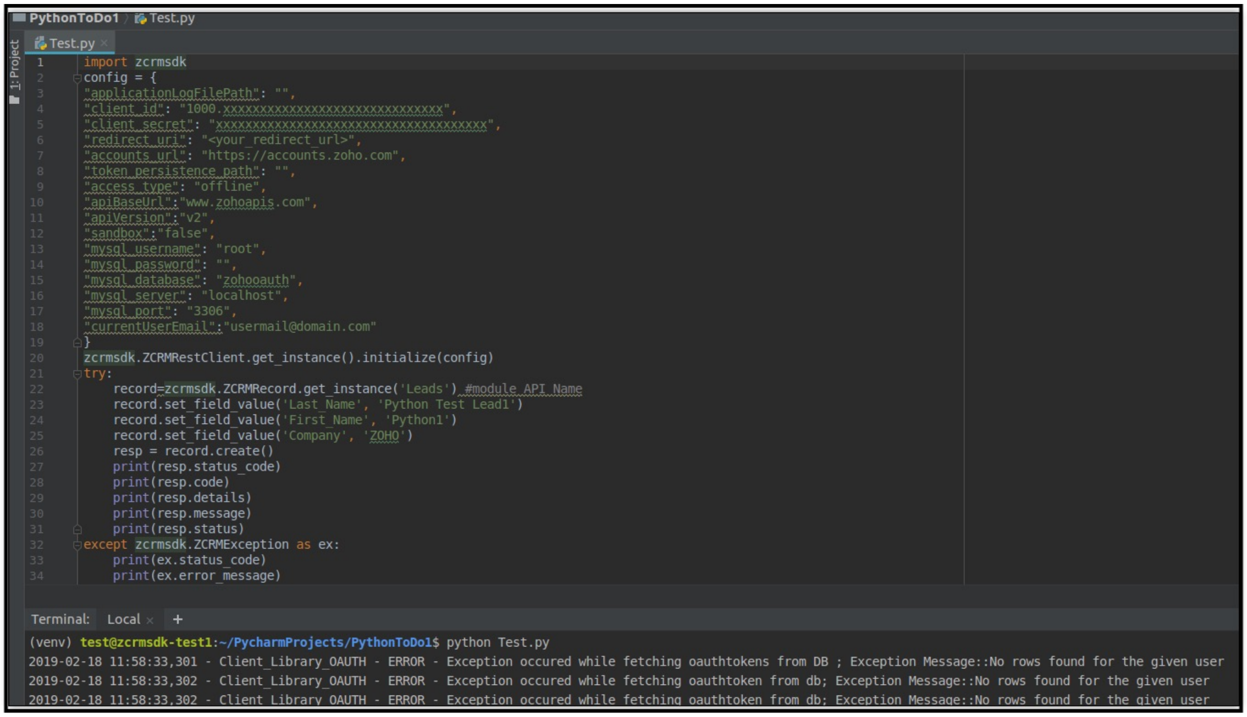
Self client

Scope:

ZohoCRM.modules.ALL,Aaaserver.profile.READ

Expiry: 3 minutes

Exception occurred while fetching OAuthtoken from DB

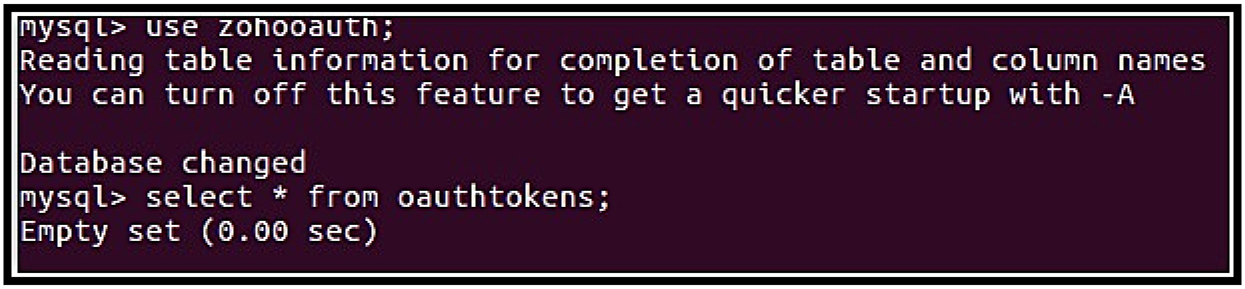


```
import zcrmsdk
config = {
"applicationLogFilePath": "",
"client_id": "1000.xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
"client_secret": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
"redirect_uri": "<your_redirect_url>",
"accounts_url": "https://accounts.zoho.com",
"token_persistence_path": "",
"access_type": "offline",
"apiBaseUrl":"www.zohoapis.com",
"apiVersion":"v2",
"sandbox":"false",
"mysql_username": "root",
"mysql_password": "",
"mysql_database": "zohooauth",
"mysql_server": "localhost",
"mysql_port": "3306",
"currentUserEmail":"usermail@domain.com"
}
zcrmsdk.ZCRMRestClient.get_instance().initialize(config)
try:
    record=zcrmsdk.ZCRMRecord.get_instance('Leads') #module API Name
    record.set_field_value('Last_Name', 'Python Test Lead1')
    record.set_field_value('First_Name', 'Python1')
    record.set_field_value('Company', 'ZOHO')
    resp = record.create()
    print(resp.status_code)
    print(resp.code)
    print(resp.details)
    print(resp.message)
    print(resp.status)
except zcrmsdk.ZCRMException as ex:
    print(ex.status_code)
    print(ex.error_message)
```

```
(venv) test@zcrmsdk-test1:~/PycharmProjects/PythonToDo1$ python Test.py
2019-02-18 11:58:33,301 - Client_Library_OAUTH - ERROR - Exception occured while fetching oauthtokens from DB ; Exception Message::No rows found for the given user
2019-02-18 11:58:33,302 - Client_Library_OAUTH - ERROR - Exception occured while fetching oauthtoken from db; Exception Message::No rows found for the given user
2019-02-18 11:58:33,302 - Client_Library_OAUTH - ERROR - Exception occured while fetching oauthtoken from db; Exception Message::No rows found for the given user
```

Reason

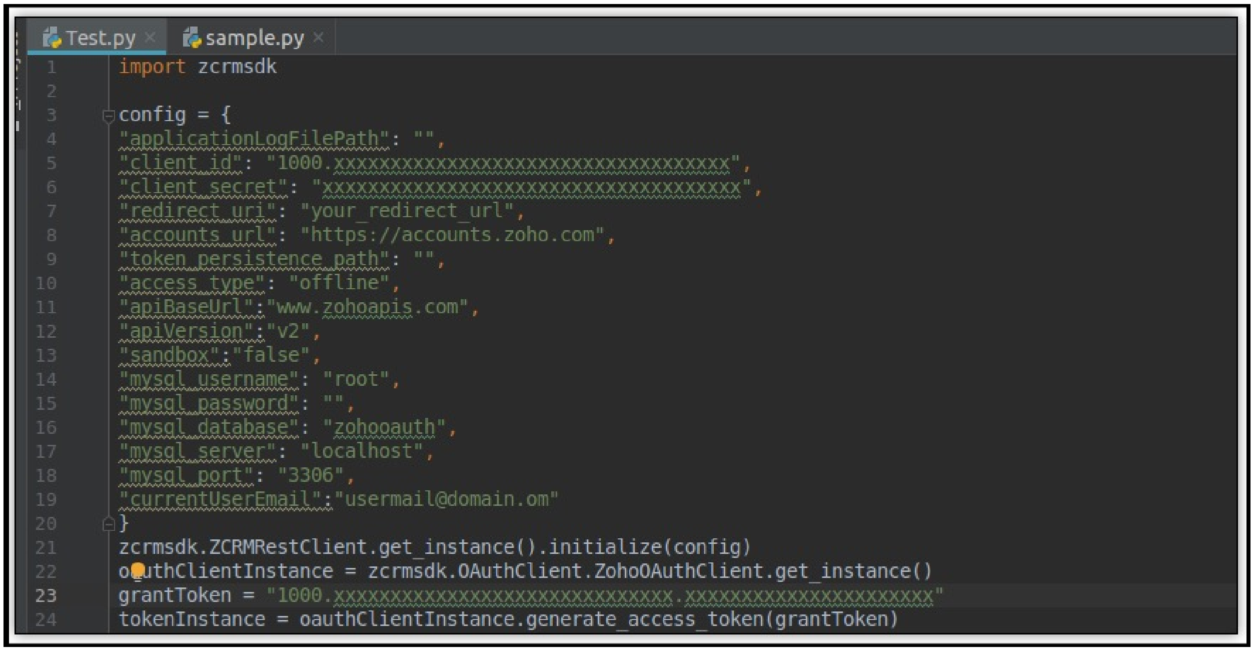
The email ID of the current user(currentUserEmail) is not found in the database because the access token is not generated for that user.



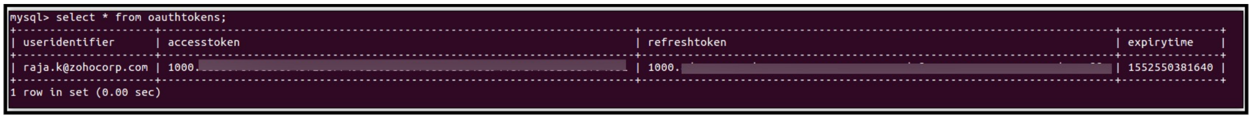
```
mysql> use zohooauth;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from oauthtokens;
Empty set (0.00 sec)
```

Solution

Generate access token for that user.

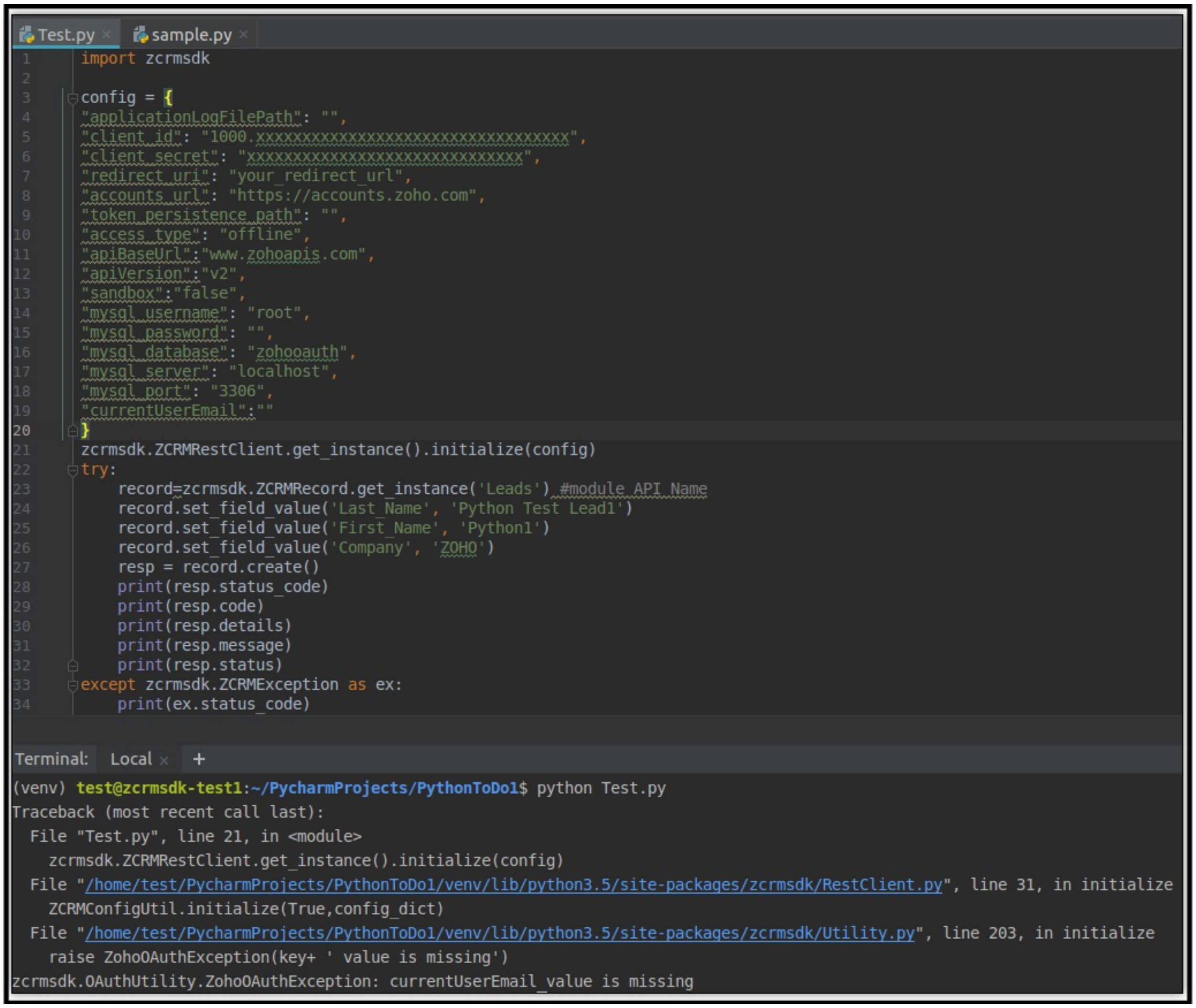


```python
import zcrmsdk

config = {
"applicationLogFilePath": "",
"client_id": "1000.xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
"client_secret": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
"redirect_uri": "your_redirect_url",
"accounts_url": "https://accounts.zoho.com",
"token_persistence_path": "",
"access_type": "offline",
"apiBaseUrl":"www.zohoapis.com",
"apiVersion":"v2",
"sandbox":"false",
"mysql_username": "root",
"mysql_password": "",
"mysql_database": "zohooauth",
"mysql_server": "localhost",
"mysql_port": "3306",
"currentUserEmail":"usermail@domain.om"
}
zcrmsdk.ZCRMRestClient.get_instance().initialize(config)
oauthClientInstance = zcrmsdk.OAuthClient.ZohoOAuthClient.get_instance()
grantToken = "1000.xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.xxxxxxxxxxxxxxxxxxxxxx"
tokenInstance = oauthClientInstance.generate_access_token(grantToken)
```



```
mysql> select * from oauthtokens;
+--------------------+--------------------------------------+--------------------------------------+---------------+
| useridentifier     | accesstoken                          | refreshtoken                         | expirytime    |
+--------------------+--------------------------------------+--------------------------------------+---------------+
| raja.k@zohocorp.com | 1000.[redacted]                     | 1000.[redacted]                      | 1552550381640 |
+--------------------+--------------------------------------+--------------------------------------+---------------+
1 row in set (0.00 sec)
```

currentUserEmail value is missing



```
import zcrmsdk

config = {
"applicationLogFilePath": "",
"client_id": "1000.xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
"client_secret": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
"redirect_uri": "your_redirect_url",
"accounts_url": "https://accounts.zoho.com",
"token_persistence_path": "",
"access_type": "offline",
"apiBaseUrl":"www.zohoapis.com",
"apiVersion":"v2",
"sandbox":"false",
"mysql_username": "root",
"mysql_password": "",
"mysql_database": "zohooauth",
"mysql_server": "localhost",
"mysql_port": "3306",
"currentUserEmail":""
}
zcrmsdk.ZCRMRestClient.get_instance().initialize(config)
try:
    record=zcrmsdk.ZCRMRecord.get_instance('Leads') #module API Name
    record.set_field_value('Last_Name', 'Python Test Lead1')
    record.set_field_value('First_Name', 'Python1')
    record.set_field_value('Company', 'ZOHO')
    resp = record.create()
    print(resp.status_code)
    print(resp.code)
    print(resp.details)
    print(resp.message)
    print(resp.status)
except zcrmsdk.ZCRMException as ex:
    print(ex.status_code)
```

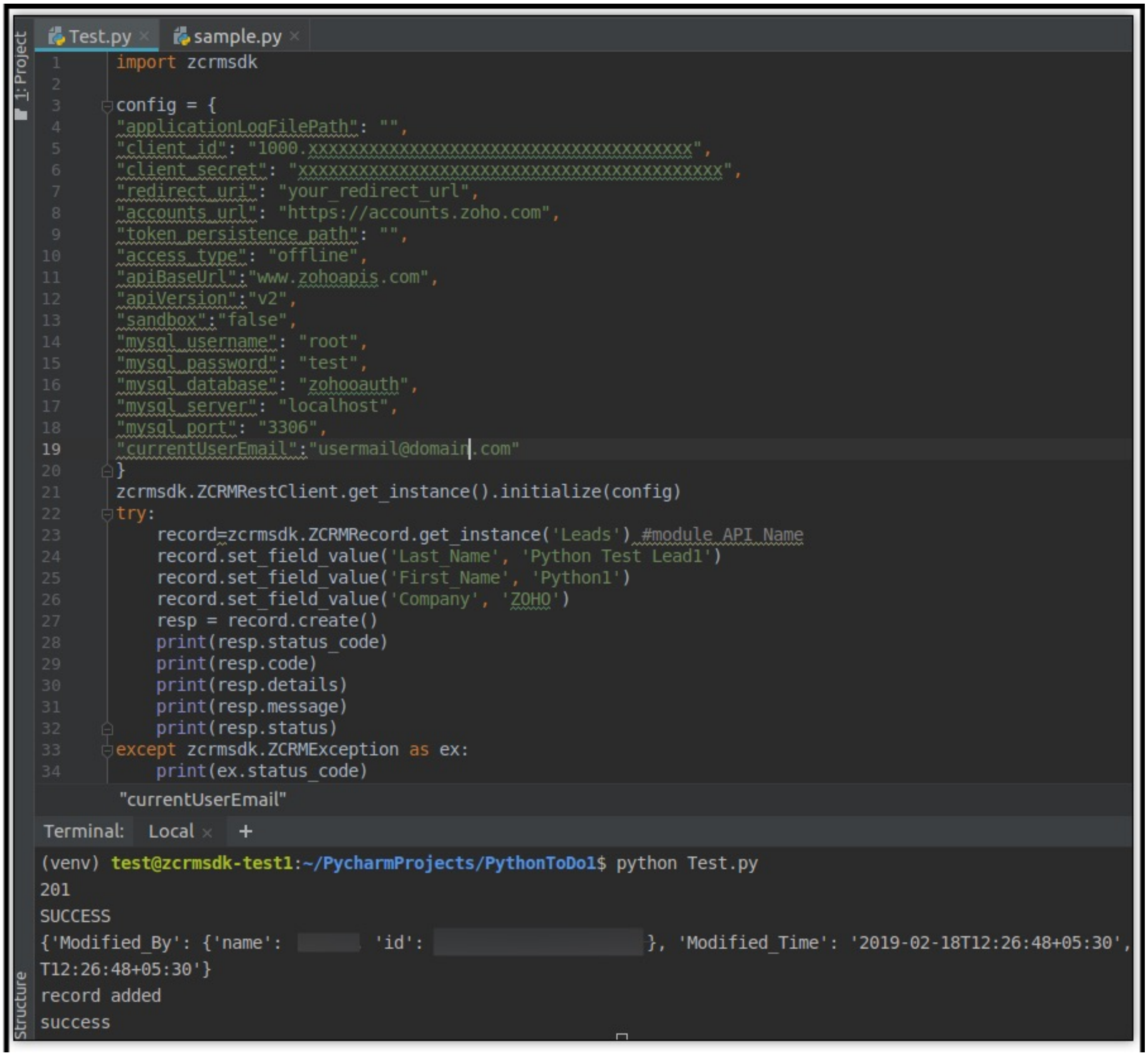
```
(venv) test@zcrmsdk-test1:~/PycharmProjects/PythonToDo1$ python Test.py
Traceback (most recent call last):
  File "Test.py", line 21, in <module>
    zcrmsdk.ZCRMRestClient.get_instance().initialize(config)
  File "/home/test/PycharmProjects/PythonToDo1/venv/lib/python3.5/site-packages/zcrmsdk/RestClient.py", line 31, in initialize
    ZCRMConfigUtil.initialize(True,config_dict)
  File "/home/test/PycharmProjects/PythonToDo1/venv/lib/python3.5/site-packages/zcrmsdk/Utility.py", line 203, in initialize
    raise ZohoOAuthException(key+ ' value is missing')
zcrmsdk.OAuthUtility.ZohoOAuthException: currentUserEmail_value is missing
```

Reason

The email ID of the current user is not specified in the field "currentUserEmail".

Solution

Enter the email ID of the current user in the "currentUserEmail" field.

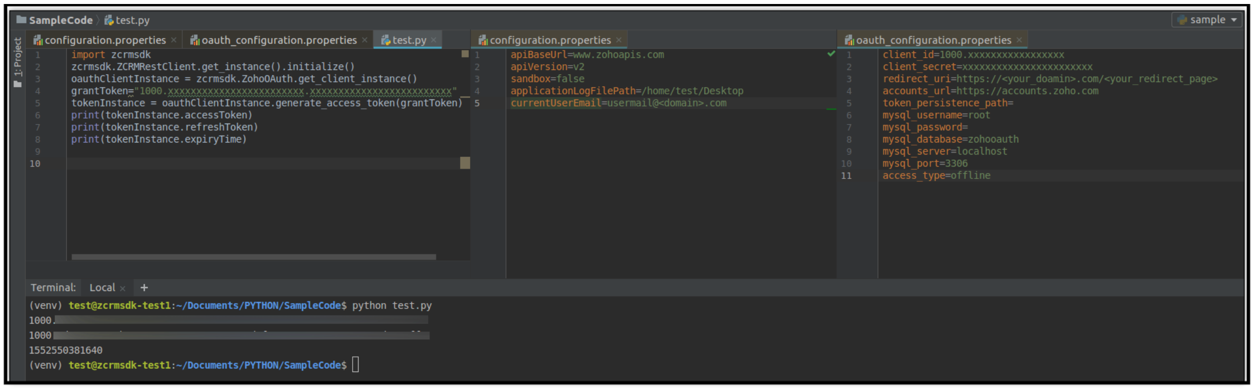


```python
import zcrmsdk

config = {
"applicationLogFilePath": "",
"client_id": "1000.xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
"client_secret": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
"redirect_uri": "your_redirect_url",
"accounts_url": "https://accounts.zoho.com",
"token_persistence_path": "",
"access_type": "offline",
"apiBaseUrl":"www.zohoapis.com",
"apiVersion":"v2",
"sandbox":"false",
"mysql_username": "root",
"mysql_password": "test",
"mysql_database": "zohooauth",
"mysql_server": "localhost",
"mysql_port": "3306",
"currentUserEmail":"usermail@domain.com"
}
zcrmsdk.ZCRMRestClient.get_instance().initialize(config)
try:
    record=zcrmsdk.ZCRMRecord.get_instance('Leads') #module API Name
    record.set_field_value('Last_Name', 'Python Test Lead1')
    record.set_field_value('First_Name', 'Python1')
    record.set_field_value('Company', 'ZOHO')
    resp = record.create()
    print(resp.status_code)
    print(resp.code)
    print(resp.details)
    print(resp.message)
    print(resp.status)
except zcrmsdk.ZCRMException as ex:
    print(ex.status_code)
```

"currentUserEmail"

```
(venv) test@zcrmsdk-test1:~/PycharmProjects/PythonToDo1$ python Test.py
201
SUCCESS
{'Modified_By': {'name': [redacted] 'id': [redacted]}, 'Modified_Time': '2019-02-18T12:26:48+05:30',
T12:26:48+05:30'}
record added
success
```

SDK not Initialized



```python
import zcrmsdk
zcrmsdk.ZCRMRestClient.get_instance().initialize()
oauthClientInstance = zcrmsdk.ZohoOAuth.get_client_instance()
grantToken="1000.xxxxxxxxxxxxxxxxxxxxxxxx.xxxxxxxxxxxxxxxxxxxxxxxxx"
tokenInstance = oauthClientInstance.generate_access_token(grantToken)
print(tokenInstance.accessToken)
print(tokenInstance.refreshToken)
print(tokenInstance.expiryTime)
```

configuration.properties

```
apiBaseUrl=www.zohoapis.com
apiVersion=v2
sandbox=false
applicationLogFilePath=/home/test/Desktop
currentUserEmail=usermail@<domain>.com
```

oauth_configuration.properties

```
client_id=1000.xxxxxxxxxxxxxxxx
client_secret=xxxxxxxxxxxxxxxxxxxxxxx
redirect_uri=https://<your_doamin>.com/<your_redirect_page>
accounts_url=https://accounts.zoho.com
token_persistence_path=
mysql_username=root
mysql_password=
mysql_database=zohooauth
mysql_server=localhost
mysql_port=3306
access_type=offline
```

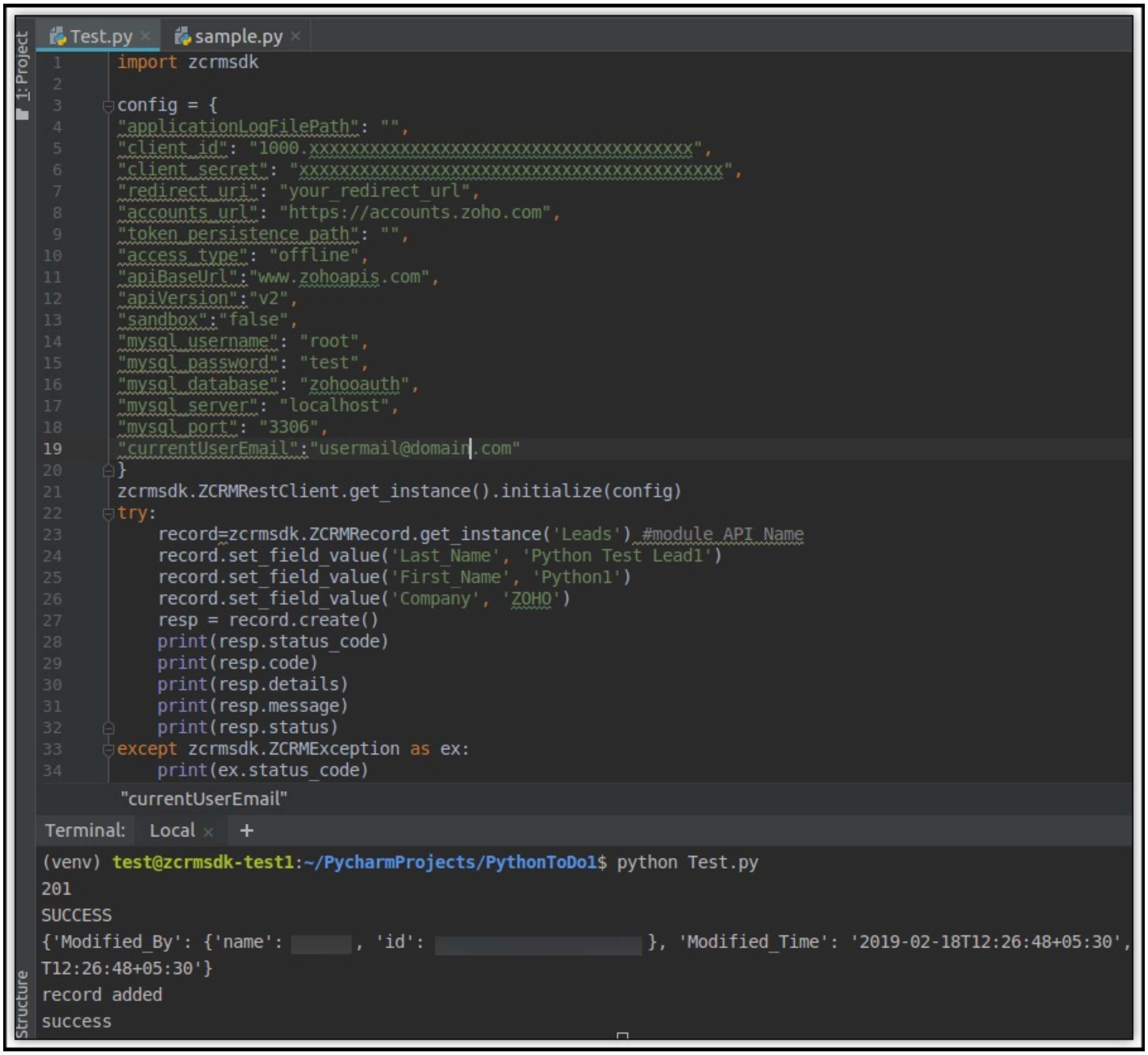
```
(venv) test@zcrmsdk-test1:~/Documents/PYTHON/SampleCode$ python test.py
1000.[redacted]
1000 [redacted]
1552550381640
(venv) test@zcrmsdk-test1:~/Documents/PYTHON/SampleCode$ 
```

Reason

The SDK is not initialized.

Solution

Follow the steps mentioned in Initialization - Python SDK to initialize the SDK.



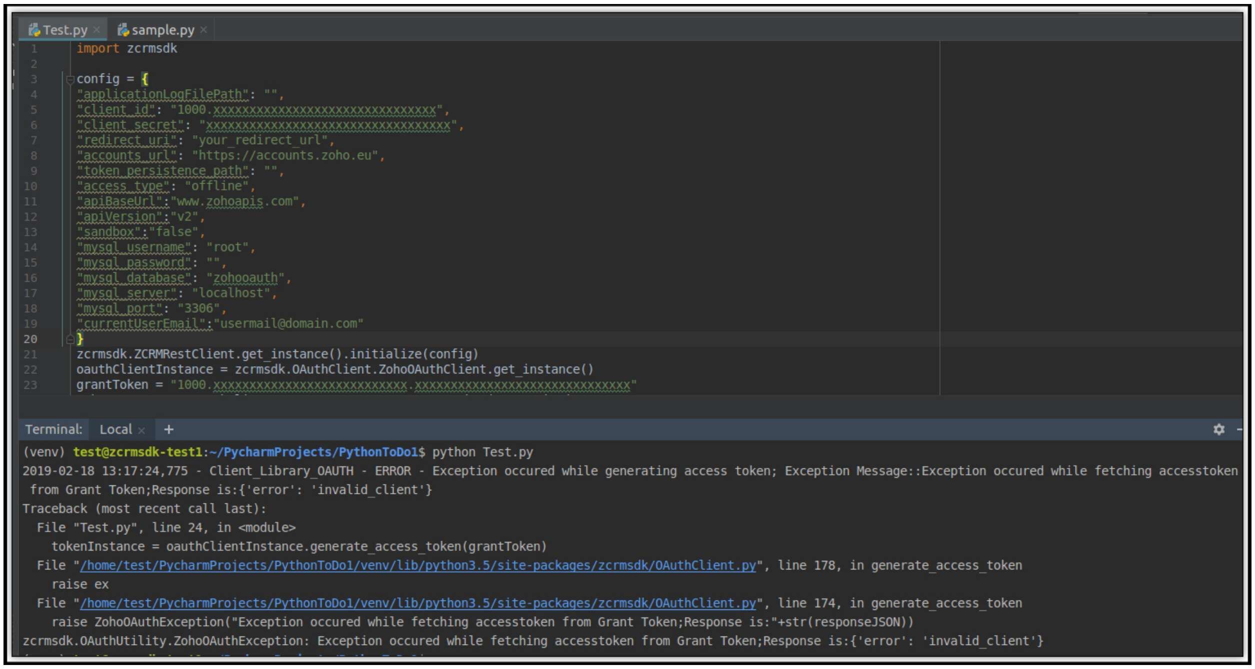
```
import zcrmsdk

config = {
"applicationLogFilePath": "",
"client_id": "1000.xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
"client_secret": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
"redirect_uri": "your_redirect_url",
"accounts_url": "https://accounts.zoho.com",
"token_persistence_path": "",
"access_type": "offline",
"apiBaseUrl":"www.zohoapis.com",
"apiVersion":"v2",
"sandbox":"false",
"mysql_username": "root",
"mysql_password": "test",
"mysql_database": "zohooauth",
"mysql_server": "localhost",
"mysql_port": "3306",
"currentUserEmail":"usermail@domain.com"
}
zcrmsdk.ZCRMRestClient.get_instance().initialize(config)
try:
    record=zcrmsdk.ZCRMRecord.get_instance('Leads') #module API Name
    record.set_field_value('Last_Name', 'Python Test Lead1')
    record.set_field_value('First_Name', 'Python1')
    record.set_field_value('Company', 'ZOHO')
    resp = record.create()
    print(resp.status_code)
    print(resp.code)
    print(resp.details)
    print(resp.message)
    print(resp.status)
except zcrmsdk.ZCRMException as ex:
    print(ex.status_code)
```

"currentUserEmail"

Terminal: Local +

```
(venv) test@zcrmsdk-test1:~/PycharmProjects/PythonToDo1$ python Test.py
201
SUCCESS
{'Modified_By': {'name': [redacted], 'id': [redacted]}, 'Modified_Time': '2019-02-18T12:26:48+05:30',
T12:26:48+05:30'}
record added
success
```

Invalid Client



```
import zcrmsdk

config = {
"applicationLogFilePath": "",
"client_id": "1000.xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
"client_secret": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
"redirect_uri": "your_redirect_url",
"accounts_url": "https://accounts.zoho.eu",
"token_persistence_path": "",
"access_type": "offline",
"apiBaseUrl":"www.zohoapis.com",
"apiVersion":"v2",
"sandbox":"false",
"mysql_username": "root",
"mysql_password": "",
"mysql_database": "zohooauth",
"mysql_server": "localhost",
"mysql_port": "3306",
"currentUserEmail":"usermail@domain.com"
}
zcrmsdk.ZCRMRestClient.get_instance().initialize(config)
oauthClientInstance = zcrmsdk.OAuthClient.ZohoOAuthClient.get_instance()
grantToken = "1000.xxxxxxxxxxxxxxxxxxxxxxxxxxxx.xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
```

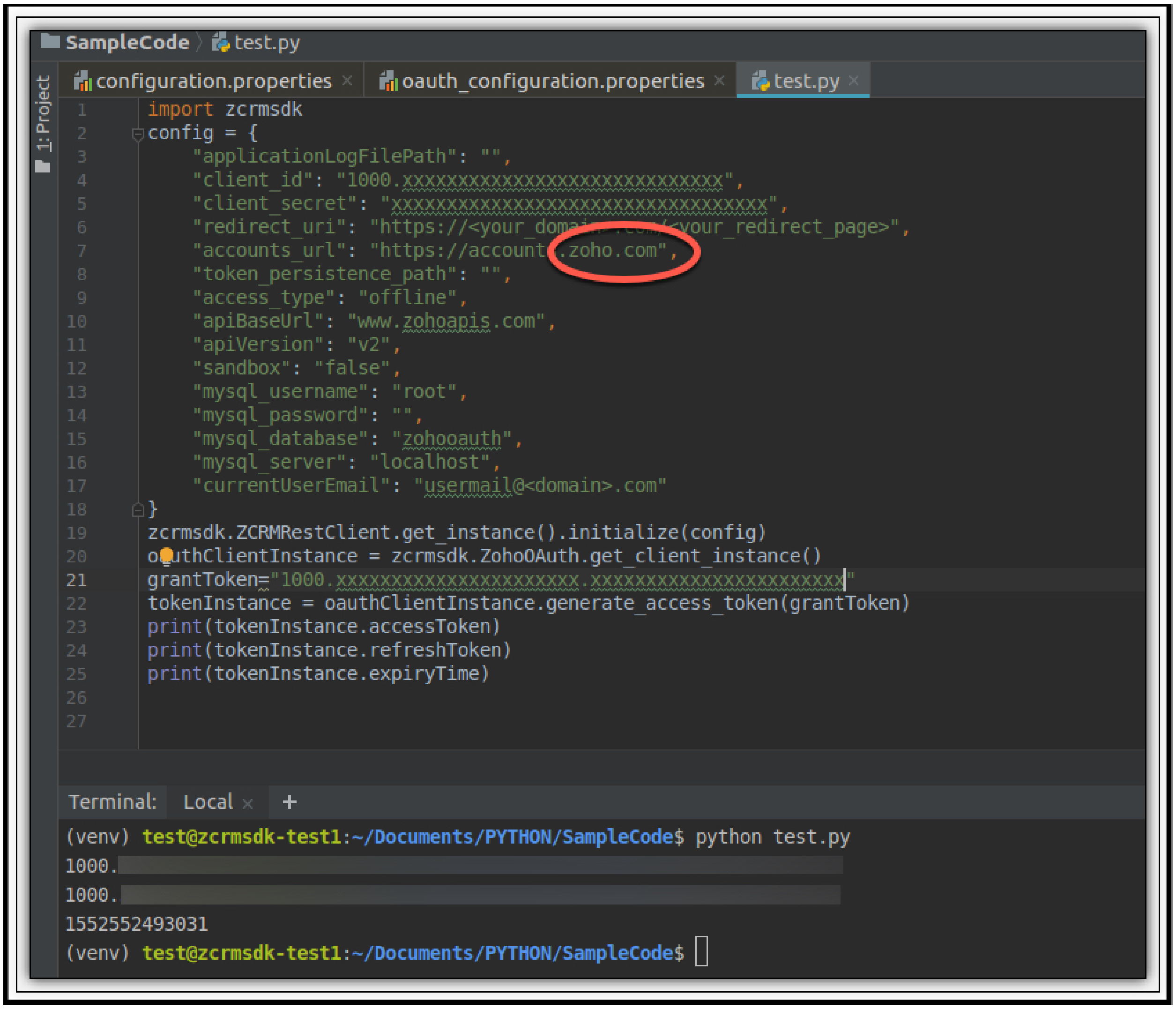
```
(venv) test@zcrmsdk-test1:~/PycharmProjects/PythonToDo1$ python Test.py
2019-02-18 13:17:24,775 - Client_Library_OAUTH - ERROR - Exception occured while generating access token; Exception Message::Exception occured while fetching accesstoken
 from Grant Token;Response is:{'error': 'invalid_client'}
Traceback (most recent call last):
  File "Test.py", line 24, in <module>
    tokenInstance = oauthClientInstance.generate_access_token(grantToken)
  File "/home/test/PycharmProjects/PythonToDo1/venv/lib/python3.5/site-packages/zcrmsdk/OAuthClient.py", line 178, in generate_access_token
    raise ex
  File "/home/test/PycharmProjects/PythonToDo1/venv/lib/python3.5/site-packages/zcrmsdk/OAuthClient.py", line 174, in generate_access_token
    raise ZohoOAuthException("Exception occured while fetching accesstoken from Grant Token;Response is:"+str(responseJSON))
zcrmsdk.OAuthUtility.ZohoOAuthException: Exception occured while fetching accesstoken from Grant Token;Response is:{'error': 'invalid_client'}
```

Reason

The user tries to generate grant token and access token from different domains. For example, the user generates the grant token from 'accounts.zoho.com' and tries to generate the access token from accounts.zoho.eu'.

Solution

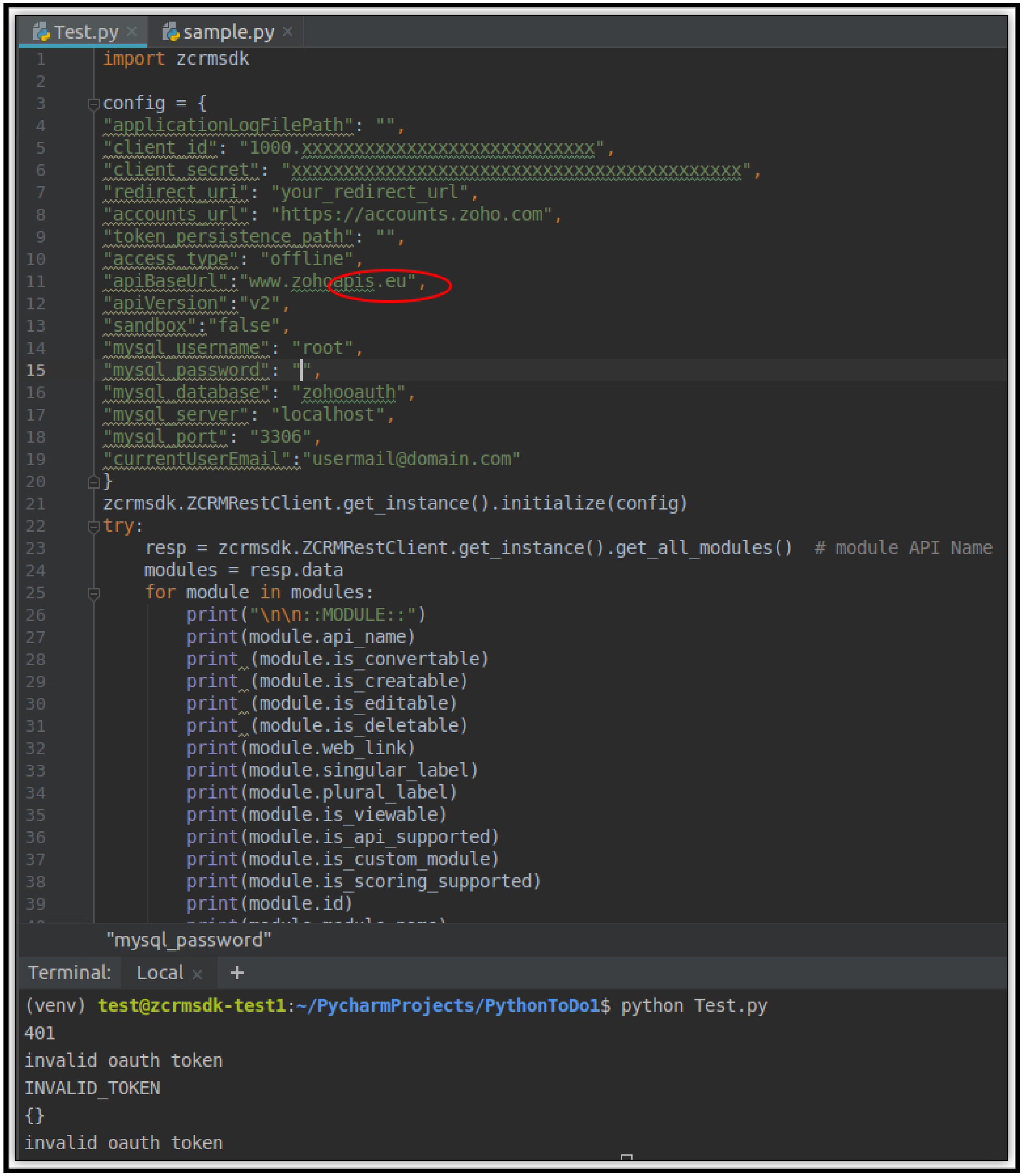
Generate both grant token and access token from the same domain.



```
import zcrmsdk
config = {
    "applicationLogFilePath": "",
    "client_id": "1000.xxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "client_secret": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "redirect_uri": "https://<your_domain>.com/<your_redirect_page>",
    "accounts_url": "https://accounts.zoho.com",
    "token_persistence_path": "",
    "access_type": "offline",
    "apiBaseUrl": "www.zohoapis.com",
    "apiVersion": "v2",
    "sandbox": "false",
    "mysql_username": "root",
    "mysql_password": "",
    "mysql_database": "zohooauth",
    "mysql_server": "localhost",
    "currentUserEmail": "usermail@<domain>.com"
}
zcrmsdk.ZCRMRestClient.get_instance().initialize(config)
oauthClientInstance = zcrmsdk.ZohoOAuth.get_client_instance()
grantToken="1000.xxxxxxxxxxxxxxxxxxxxxx.xxxxxxxxxxxxxxxxxxxxxxxx"
tokenInstance = oauthClientInstance.generate_access_token(grantToken)
print(tokenInstance.accessToken)
print(tokenInstance.refreshToken)
print(tokenInstance.expiryTime)
```

```
(venv) test@zcrmsdk-test1:~/Documents/PYTHON/SampleCode$ python test.py
1000.[redacted]
1000.[redacted]
1552552493031
(venv) test@zcrmsdk-test1:~/Documents/PYTHON/SampleCode$
```

Invalid OAuth Token



```
import zcrmsdk

config = {
"applicationLogFilePath": "",
"client_id": "1000.xxxxxxxxxxxxxxxxxxxxxxxxxxxx",
"client_secret": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
"redirect_uri": "your_redirect_url",
"accounts_url": "https://accounts.zoho.com",
"token_persistence_path": "",
"access_type": "offline",
"apiBaseUrl":"www.zohoapis.eu",
"apiVersion":"v2",
"sandbox":"false",
"mysql_username": "root",
"mysql_password": "",
"mysql_database": "zohooauth",
"mysql_server": "localhost",
"mysql_port": "3306",
"currentUserEmail":"usermail@domain.com"
}
zcrmsdk.ZCRMRestClient.get_instance().initialize(config)
try:
    resp = zcrmsdk.ZCRMRestClient.get_instance().get_all_modules()  # module API Name
    modules = resp.data
    for module in modules:
        print("\n\n::MODULE::")
        print(module.api_name)
        print (module.is_convertable)
        print (module.is_creatable)
        print (module.is_editable)
        print (module.is_deletable)
        print(module.web_link)
        print(module.singular_label)
        print(module.plural_label)
        print(module.is_viewable)
        print(module.is_api_supported)
        print(module.is_custom_module)
        print(module.is_scoring_supported)
        print(module.id)
```

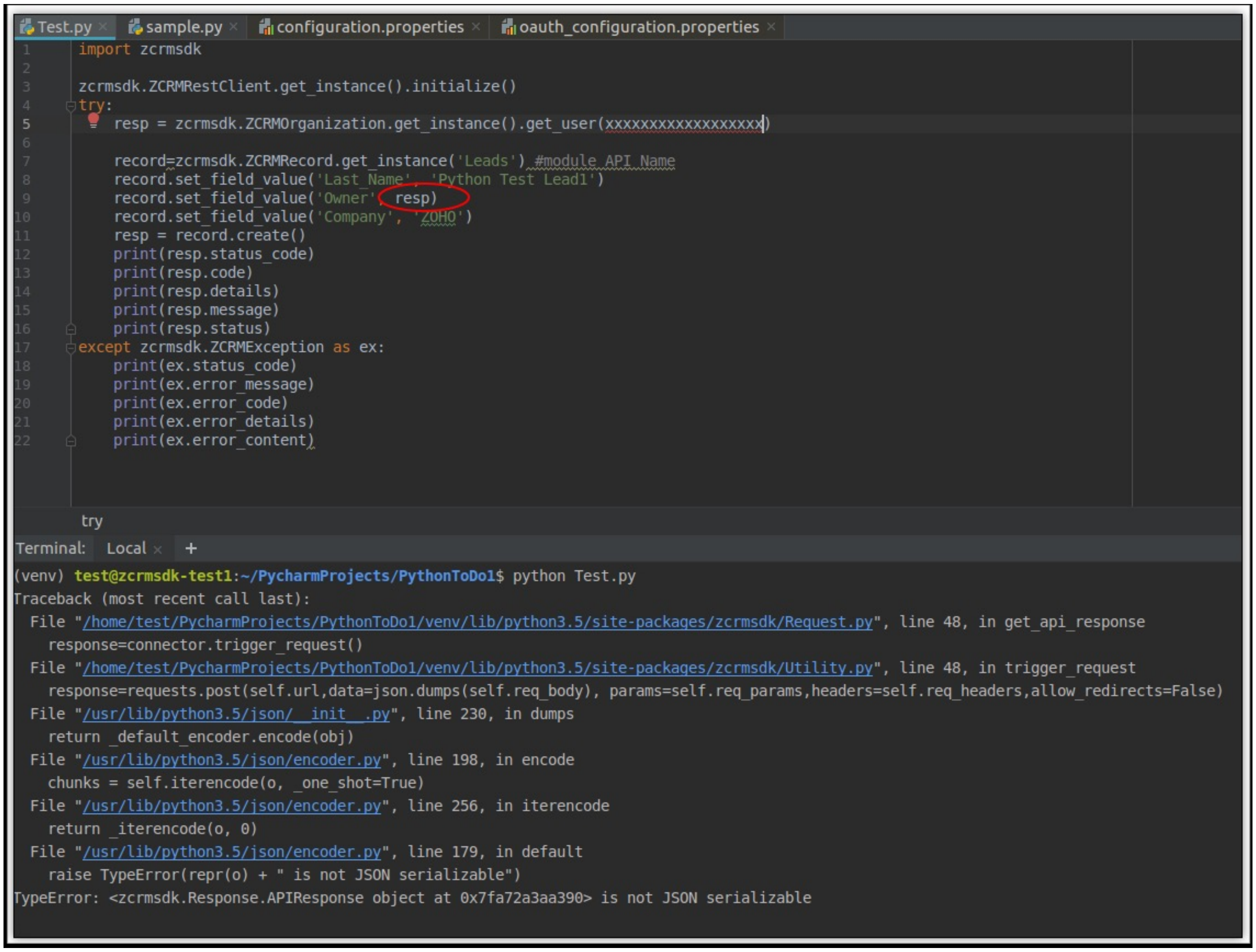
```
(venv) test@zcrmsdk-test1:~/PycharmProjects/PythonToDo1$ python Test.py
401
invalid oauth token
INVALID_TOKEN
{}
invalid oauth token
```

Reason

The user generates access token in a domain but tries to access a record in a different domain. For example, the user generates the access token from www.zohoapis.com and tries to access a module in www.zohoapis.eu

Solution
Access the records in the same domain as you generated the access token in.

The Object is not JSON Serializable

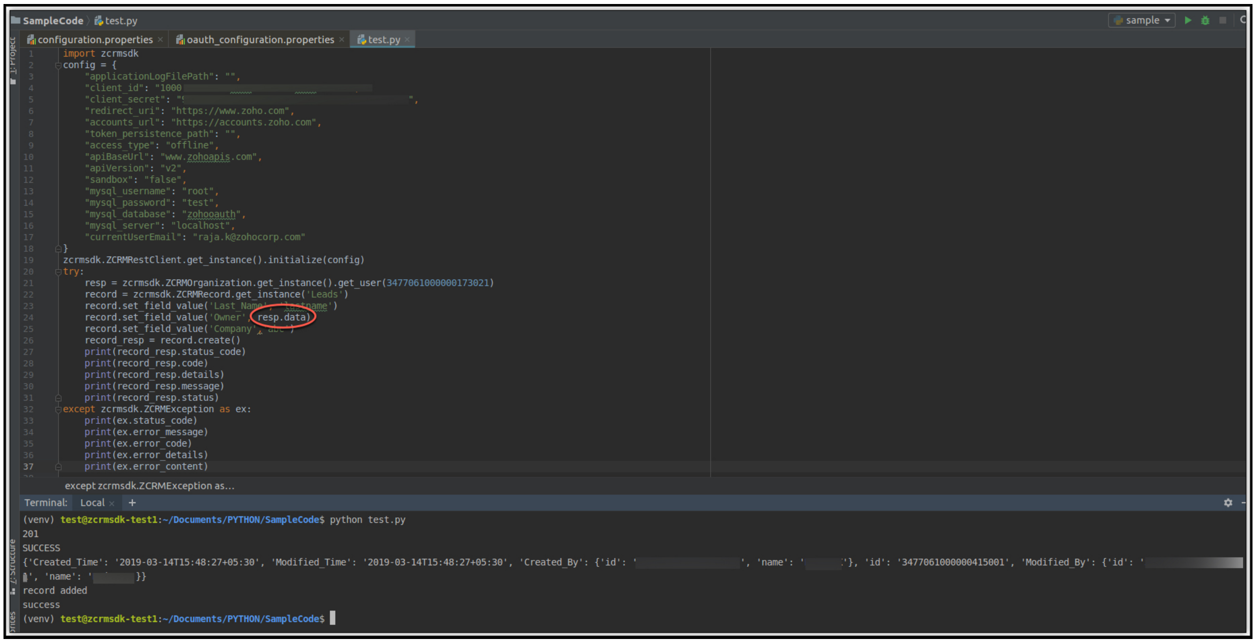


```python
import zcrmsdk

zcrmsdk.ZCRMRestClient.get_instance().initialize()
try:
    resp = zcrmsdk.ZCRMOrganization.get_instance().get_user(xxxxxxxxxxxxxxxxxx)

    record=zcrmsdk.ZCRMRecord.get_instance('Leads') #module API Name
    record.set_field_value('Last_Name', 'Python Test Lead1')
    record.set_field_value('Owner', resp)
    record.set_field_value('Company', 'ZOHO')
    resp = record.create()
    print(resp.status_code)
    print(resp.code)
    print(resp.details)
    print(resp.message)
    print(resp.status)
except zcrmsdk.ZCRMException as ex:
    print(ex.status_code)
    print(ex.error_message)
    print(ex.error_code)
    print(ex.error_details)
    print(ex.error_content)
```

```
(venv) test@zcrmsdk-test1:~/PycharmProjects/PythonToDo1$ python Test.py
Traceback (most recent call last):
  File "/home/test/PycharmProjects/PythonToDo1/venv/lib/python3.5/site-packages/zcrmsdk/Request.py", line 48, in get_api_response
    response=connector.trigger_request()
  File "/home/test/PycharmProjects/PythonToDo1/venv/lib/python3.5/site-packages/zcrmsdk/Utility.py", line 48, in trigger_request
    response=requests.post(self.url,data=json.dumps(self.req_body), params=self.req_params,headers=self.req_headers,allow_redirects=False)
  File "/usr/lib/python3.5/json/__init__.py", line 230, in dumps
    return _default_encoder.encode(obj)
  File "/usr/lib/python3.5/json/encoder.py", line 198, in encode
    chunks = self.iterencode(o, _one_shot=True)
  File "/usr/lib/python3.5/json/encoder.py", line 256, in iterencode
    return _iterencode(o, 0)
  File "/usr/lib/python3.5/json/encoder.py", line 179, in default
    raise TypeError(repr(o) + " is not JSON serializable")
TypeError: <zcrmsdk.Response.APIResponse object at 0x7fa72a3aa390> is not JSON serializable
```

Reason
An incorrect object is passed that cannot be serialized into JSON.

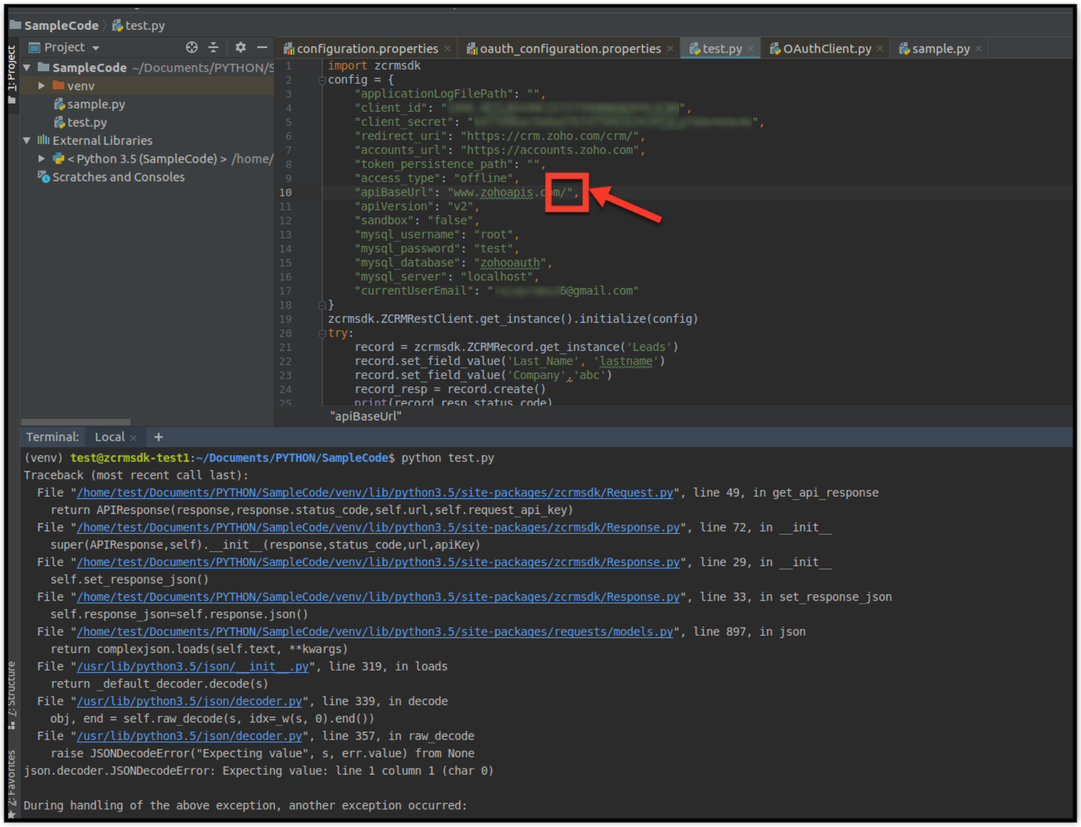
Solution
Use the instance variable "data" to get the entity data alone from the response wrapper objects. In the example, use "resp.data"



```
import zcrmsdk
config = {
    "applicationLogFilePath": "",
    "client_id": "1000[redacted]
    "client_secret": "[redacted]",
    "redirect_uri": "https://www.zoho.com",
    "accounts_url": "https://accounts.zoho.com",
    "token_persistence_path": "",
    "access_type": "offline",
    "apiBaseUrl": "www.zohoapis.com",
    "apiVersion": "v2",
    "sandbox": "false",
    "mysql_username": "root",
    "mysql_password": "test",
    "mysql_database": "zohooauth",
    "mysql_server": "localhost",
    "currentUserEmail": "raja.k@zohocorp.com"
}
zcrmsdk.ZCRMRestClient.get_instance().initialize(config)
try:
    resp = zcrmsdk.ZCRMOrganization.get_instance().get_user(3477061000000173021)
    record = zcrmsdk.ZCRMRecord.get_instance('Leads')
    record.set_field_value('Last_Name', 'lastname')
    record.set_field_value('Owner', resp.data)
    record.set_field_value('Company', 'abc')
    record_resp = record.create()
    print(record_resp.status_code)
    print(record_resp.code)
    print(record_resp.details)
    print(record_resp.message)
    print(record_resp.status)
except zcrmsdk.ZCRMException as ex:
    print(ex.status_code)
    print(ex.error_message)
    print(ex.error_code)
    print(ex.error_details)
    print(ex.error_content)
```

```
(venv) test@zcrmsdk-test1:~/Documents/PYTHON/SampleCode$ python test.py
201
SUCCESS
{'Created_Time': '2019-03-14T15:48:27+05:30', 'Modified_Time': '2019-03-14T15:48:27+05:30', 'Created_By': {'id': '[redacted]', 'name': '[redacted]'}, 'id': '3477061000000415001', 'Modified_By': {'id': '[redacted]', 'name': '[redacted]'}}
record added
success
(venv) test@zcrmsdk-test1:~/Documents/PYTHON/SampleCode$
```

ZCRMException Caused by:'Unknown ZCRMException'



```
import zcrmsdk
config = {
    "applicationLogFilePath": "",
    "client_id": "[redacted]",
    "client_secret": "[redacted]",
    "redirect_uri": "https://crm.zoho.com/crm/",
    "accounts_url": "https://accounts.zoho.com",
    "token_persistence_path": "",
    "access_type": "offline",
    "apiBaseUrl": "www.zohoapis.com/",
    "apiVersion": "v2",
    "sandbox": "false",
    "mysql_username": "root",
    "mysql_password": "test",
    "mysql_database": "zohooauth",
    "mysql_server": "localhost",
    "currentUserEmail": "[redacted]6@gmail.com"
}
zcrmsdk.ZCRMRestClient.get_instance().initialize(config)
try:
    record = zcrmsdk.ZCRMRecord.get_instance('Leads')
    record.set_field_value('Last_Name', 'lastname')
    record.set_field_value('Company','abc')
    record_resp = record.create()
    print(record_resp.status_code)
```

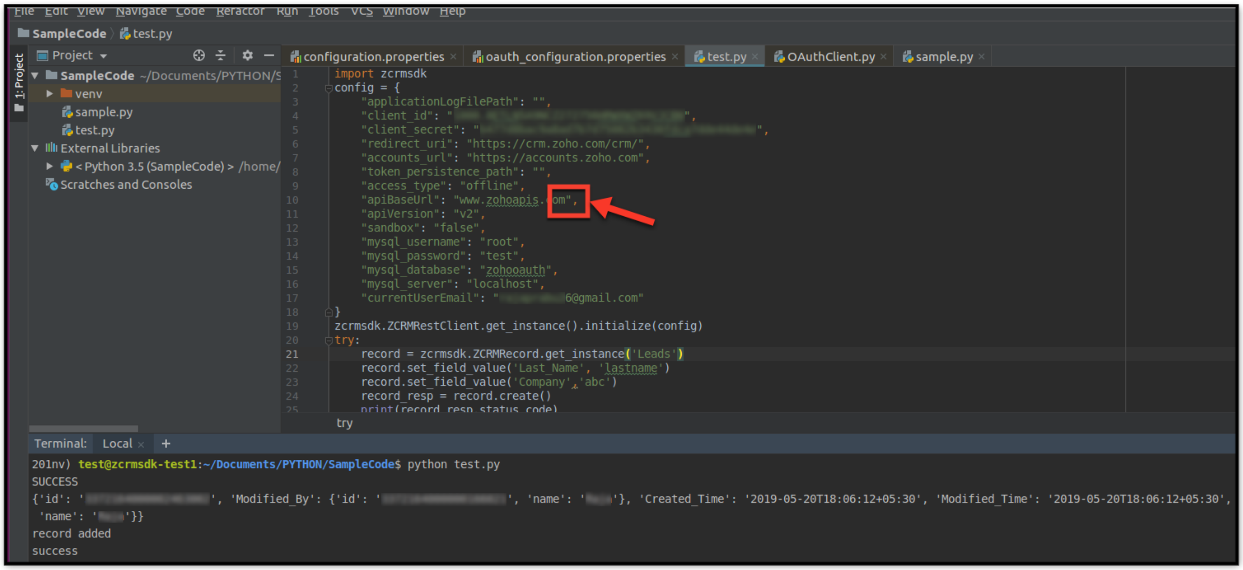
```
(venv) test@zcrmsdk-test1:~/Documents/PYTHON/SampleCode$ python test.py
Traceback (most recent call last):
  File "/home/test/Documents/PYTHON/SampleCode/venv/lib/python3.5/site-packages/zcrmsdk/Request.py", line 49, in get_api_response
    return APIResponse(response,response.status_code,self.url,self.request_api_key)
  File "/home/test/Documents/PYTHON/SampleCode/venv/lib/python3.5/site-packages/zcrmsdk/Response.py", line 72, in __init__
    super(APIResponse,self).__init__(response,status_code,url,apiKey)
  File "/home/test/Documents/PYTHON/SampleCode/venv/lib/python3.5/site-packages/zcrmsdk/Response.py", line 29, in __init__
    self.set_response_json()
  File "/home/test/Documents/PYTHON/SampleCode/venv/lib/python3.5/site-packages/zcrmsdk/Response.py", line 33, in set_response_json
    self.response_json=self.response.json()
  File "/home/test/Documents/PYTHON/SampleCode/venv/lib/python3.5/site-packages/requests/models.py", line 897, in json
    return complexjson.loads(self.text, **kwargs)
  File "/usr/lib/python3.5/json/__init__.py", line 319, in loads
    return _default_decoder.decode(s)
  File "/usr/lib/python3.5/json/decoder.py", line 339, in decode
    obj, end = self.raw_decode(s, idx=_w(s, 0).end())
  File "/usr/lib/python3.5/json/decoder.py", line 357, in raw_decode
    raise JSONDecodeError("Expecting value", s, err.value) from None
json.decoder.JSONDecodeError: Expecting value: line 1 column 1 (char 0)

During handling of the above exception, another exception occurred:
```

Reason

The apiBaseUrl key in the configuration dictionary must have had the value as "https://www.zohoapis.com/". The slash at the end of the line is automatically added by the SDK. Hence if the user adds a slash as well, the final input to the SDK will have two slashes. Ex: "https://www.zohoapis.com//".

Solution

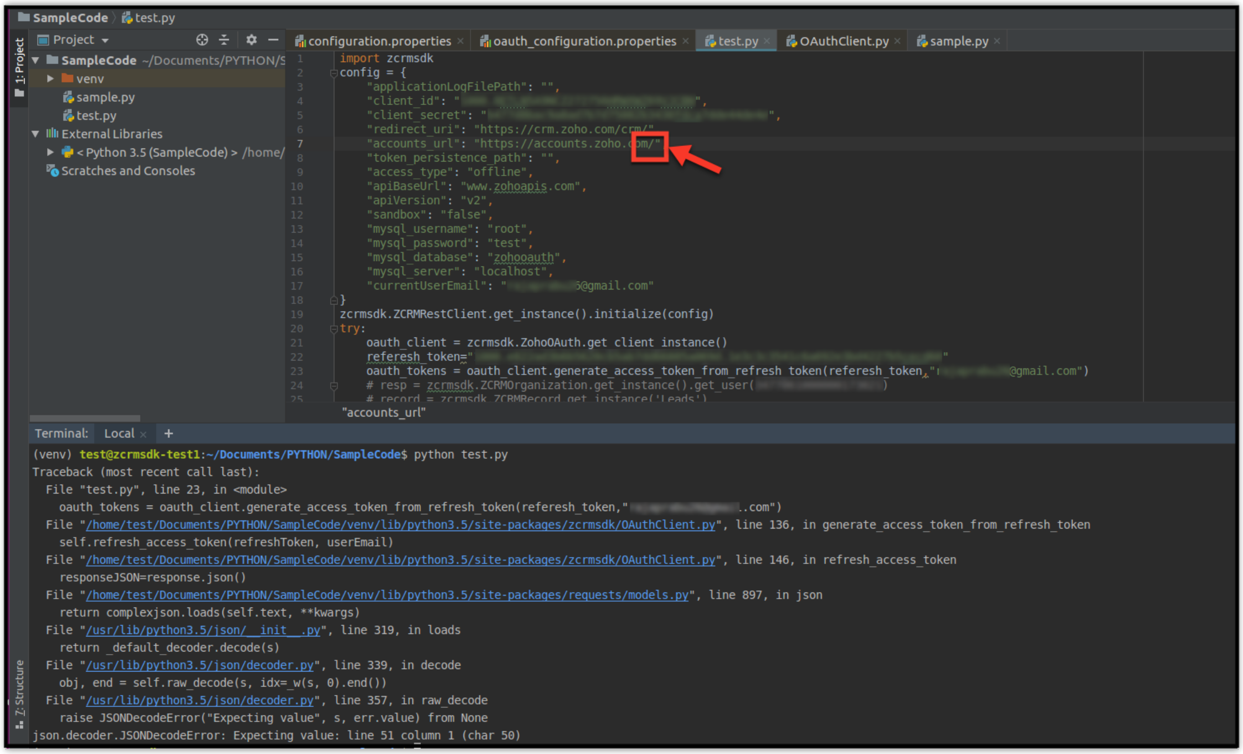
Remove the (/) slash after https://www.zohoapis.com in the "apiBaseUrl" key of the configuration dictionary.



```
import zcrmsdk
config = {
    "applicationLogFilePath": "",
    "client_id": "[redacted]",
    "client_secret": "[redacted]",
    "redirect_uri": "https://crm.zoho.com/crm/",
    "accounts_url": "https://accounts.zoho.com",
    "token_persistence_path": "",
    "access_type": "offline",
    "apiBaseUrl": "www.zohoapis.com",
    "apiVersion": "v2",
    "sandbox": "false",
    "mysql_username": "root",
    "mysql_password": "test",
    "mysql_database": "zohooauth",
    "mysql_server": "localhost",
    "currentUserEmail": "[redacted]6@gmail.com"
}
zcrmsdk.ZCRMRestClient.get_instance().initialize(config)
try:
    record = zcrmsdk.ZCRMRecord.get_instance('Leads')
    record.set_field_value('Last_Name', 'lastname')
    record.set_field_value('Company','abc')
    record_resp = record.create()
    print(record_resp.status_code)
```

```
201nv) test@zcrmsdk-test1:~/Documents/PYTHON/SampleCode$ python test.py
SUCCESS
{'id': '[redacted]', 'Modified_By': {'id': '[redacted]', 'name': '[redacted]'}, 'Created_Time': '2019-05-20T18:06:12+05:30', 'Modified_Time': '2019-05-20T18:06:12+05:30',
 'name': '[redacted]'}}
record added
success
```

ZohoOAuthException Caused by:'ZohoOAuthException Caused by:'Exception while fetching access token from refresh token - HTTP/1.1 404



```python
import zcrmsdk
config = {
    "applicationLogFilePath": "",
    "client_id": "[redacted]",
    "client_secret": "[redacted]",
    "redirect_uri": "https://crm.zoho.com/crm/",
    "accounts_url": "https://accounts.zoho.com/",
    "token_persistence_path": "",
    "access_type": "offline",
    "apiBaseUrl": "www.zohoapis.com",
    "apiVersion": "v2",
    "sandbox": "false",
    "mysql_username": "root",
    "mysql_password": "test",
    "mysql_database": "zohooauth",
    "mysql_server": "localhost",
    "currentUserEmail": "[redacted]@gmail.com"
}
zcrmsdk.ZCRMRestClient.get_instance().initialize(config)
try:
    oauth_client = zcrmsdk.ZohoOAuth.get_client_instance()
    referesh_token="[redacted]"
    oauth_tokens = oauth_client.generate_access_token_from_refresh_token(referesh_token,"[redacted]@gmail.com")
    # resp = zcrmsdk.ZCRMOrganization.get_instance().get_user([redacted])
    # record = zcrmsdk.ZCRMRecord.get_instance('Leads')
```

"accounts_url"

Terminal: Local

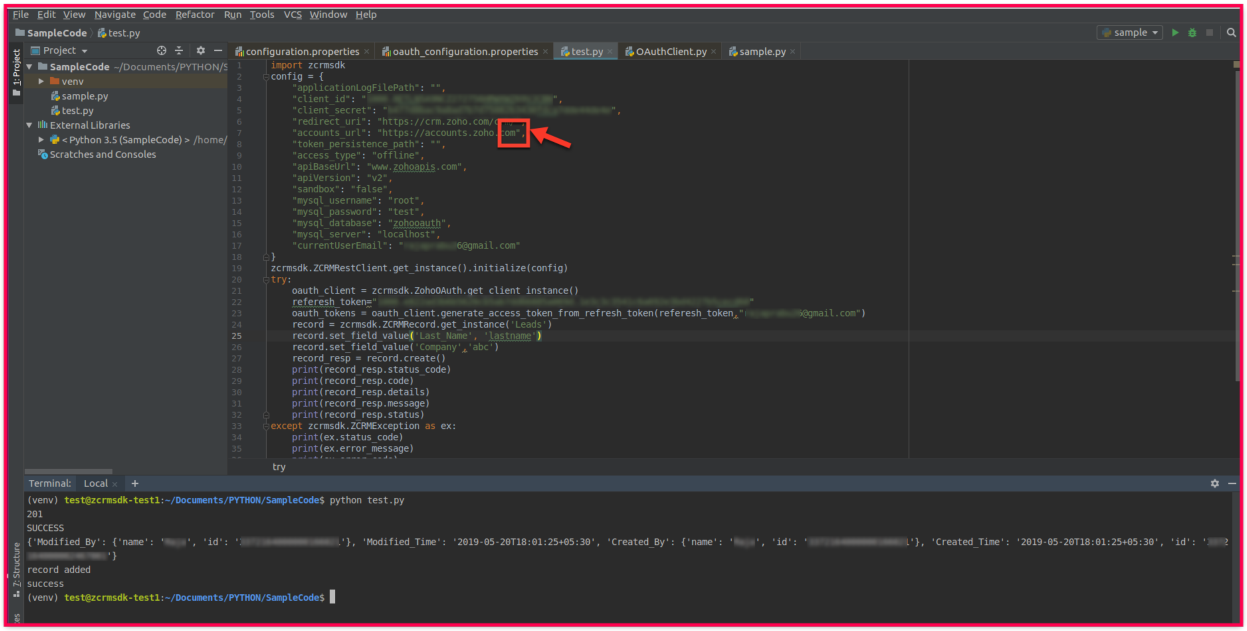
```
(venv) test@zcrmsdk-test1:~/Documents/PYTHON/SampleCode$ python test.py
Traceback (most recent call last):
  File "test.py", line 23, in <module>
    oauth_tokens = oauth_client.generate_access_token_from_refresh_token(referesh_token,"[redacted].com")
  File "/home/test/Documents/PYTHON/SampleCode/venv/lib/python3.5/site-packages/zcrmsdk/OAuthClient.py", line 136, in generate_access_token_from_refresh_token
    self.refresh_access_token(refreshToken, userEmail)
  File "/home/test/Documents/PYTHON/SampleCode/venv/lib/python3.5/site-packages/zcrmsdk/OAuthClient.py", line 146, in refresh_access_token
    responseJSON=response.json()
  File "/home/test/Documents/PYTHON/SampleCode/venv/lib/python3.5/site-packages/requests/models.py", line 897, in json
    return complexjson.loads(self.text, **kwargs)
  File "/usr/lib/python3.5/json/__init__.py", line 319, in loads
    return _default_decoder.decode(s)
  File "/usr/lib/python3.5/json/decoder.py", line 339, in decode
    obj, end = self.raw_decode(s, idx=_w(s, 0).end())
  File "/usr/lib/python3.5/json/decoder.py", line 357, in raw_decode
    raise JSONDecodeError("Expecting value", s, err.value) from None
json.decoder.JSONDecodeError: Expecting value: line 51 column 1 (char 50)
```

Reason

The accounts_url key in the configuration dictionary must have had the value as "https://accounts.zoho.com/". The slash at the end of the line is automatically added by the SDK. Hence if the user adds a slash as well, the final input to the SDK will have two slashes. Ex: "https://accounts.zoho.com//".

Solution

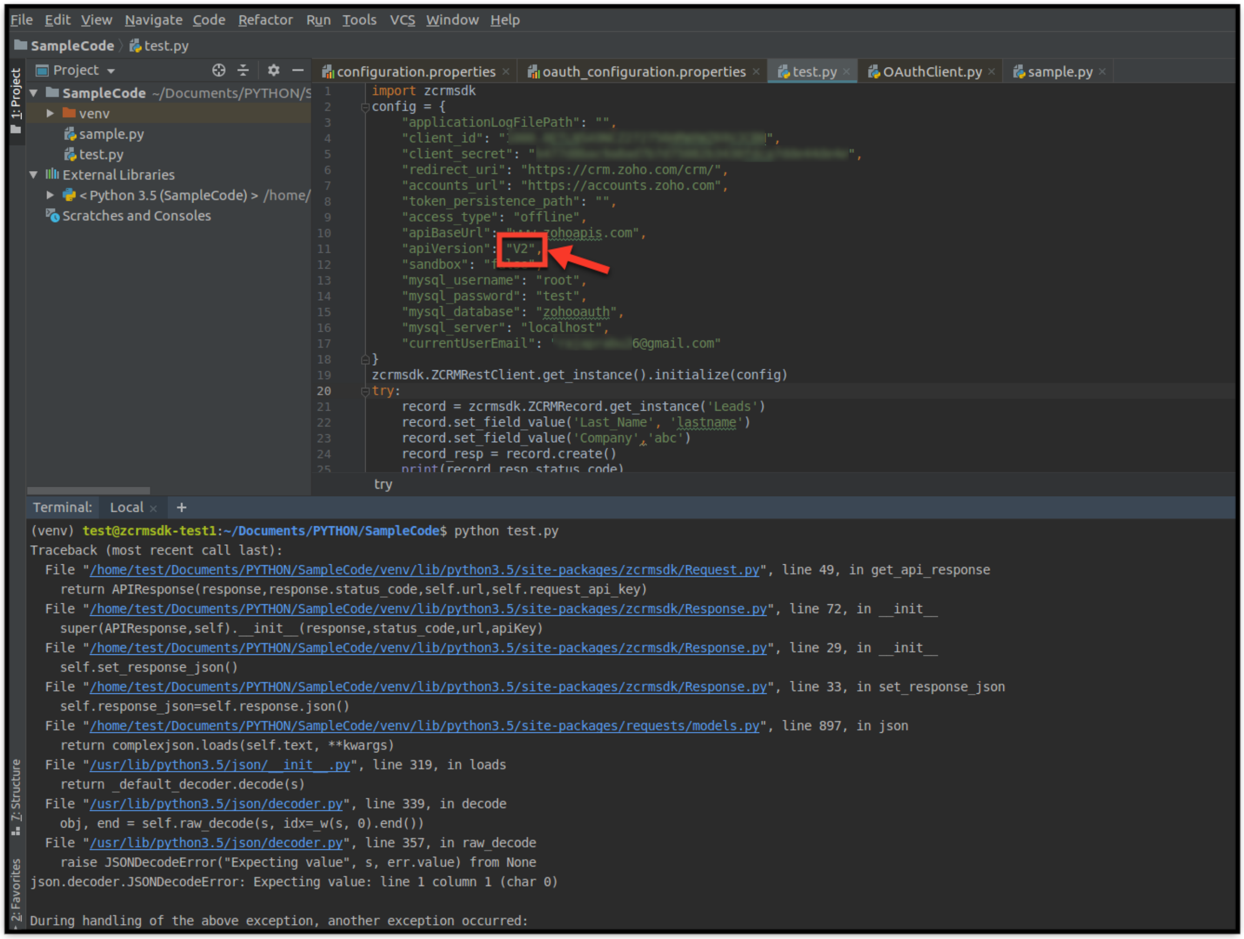
Remove the (/) slash after https://accounts.zoho.com in the "accounts_url" key of the configuration dictionary.



```
import zcrmsdk
config = {
    "applicationLogFilePath": "",
    "client_id": "[redacted]",
    "client_secret": "[redacted]",
    "redirect_uri": "https://crm.zoho.com/c
    "accounts_url": "https://accounts.zoho.com",
    "token_persistence_path": "",
    "access_type": "offline",
    "apiBaseUrl": "www.zohoapis.com",
    "apiVersion": "v2",
    "sandbox": "false",
    "mysql_username": "root",
    "mysql_password": "test",
    "mysql_database": "zohooauth",
    "mysql_server": "localhost",
    "currentUserEmail": "[redacted]6@gmail.com"
}
zcrmsdk.ZCRMRestClient.get_instance().initialize(config)
try:
    oauth_client = zcrmsdk.ZohoOAuth.get_client_instance()
    referesh_token="[redacted]"
    oauth_tokens = oauth_client.generate_access_token_from_refresh_token(referesh_token,"[redacted]@gmail.com")
    record = zcrmsdk.ZCRMRecord.get_instance('Leads')
    record.set_field_value('Last_Name', 'lastname')
    record.set_field_value('Company','abc')
    record_resp = record.create()
    print(record_resp.status_code)
    print(record_resp.code)
    print(record_resp.details)
    print(record_resp.message)
    print(record_resp.status)
except zcrmsdk.ZCRMException as ex:
    print(ex.status_code)
    print(ex.error_message)
```

```
(venv) test@zcrmsdk-test1:~/Documents/PYTHON/SampleCode$ python test.py
201
SUCCESS
{'Modified_By': {'name': '[redacted]', 'id': '[redacted]'}, 'Modified_Time': '2019-05-20T18:01:25+05:30', 'Created_By': {'name': '[redacted]', 'id': '[redacted]'}, 'Created_Time': '2019-05-20T18:01:25+05:30', 'id': '[redacted]
[redacted]'}
record added
success
(venv) test@zcrmsdk-test1:~/Documents/PYTHON/SampleCode$
```

ZCRMException Caused by:'Unknown ZCRMException'



```
import zcrmsdk
config = {
    "applicationLogFilePath": "",
    "client_id": "[redacted]",
    "client_secret": "[redacted]",
    "redirect_uri": "https://crm.zoho.com/crm/",
    "accounts_url": "https://accounts.zoho.com",
    "token_persistence_path": "",
    "access_type": "offline",
    "apiBaseUrl": "www.zohoapis.com",
    "apiVersion": "V2",
    "sandbox": "false",
    "mysql_username": "root",
    "mysql_password": "test",
    "mysql_database": "zohooauth",
    "mysql_server": "localhost",
    "currentUserEmail": "[redacted]6@gmail.com"
}
zcrmsdk.ZCRMRestClient.get_instance().initialize(config)
try:
    record = zcrmsdk.ZCRMRecord.get_instance('Leads')
    record.set_field_value('Last_Name', 'lastname')
    record.set_field_value('Company','abc')
    record_resp = record.create()
    print(record_resp.status_code)
```

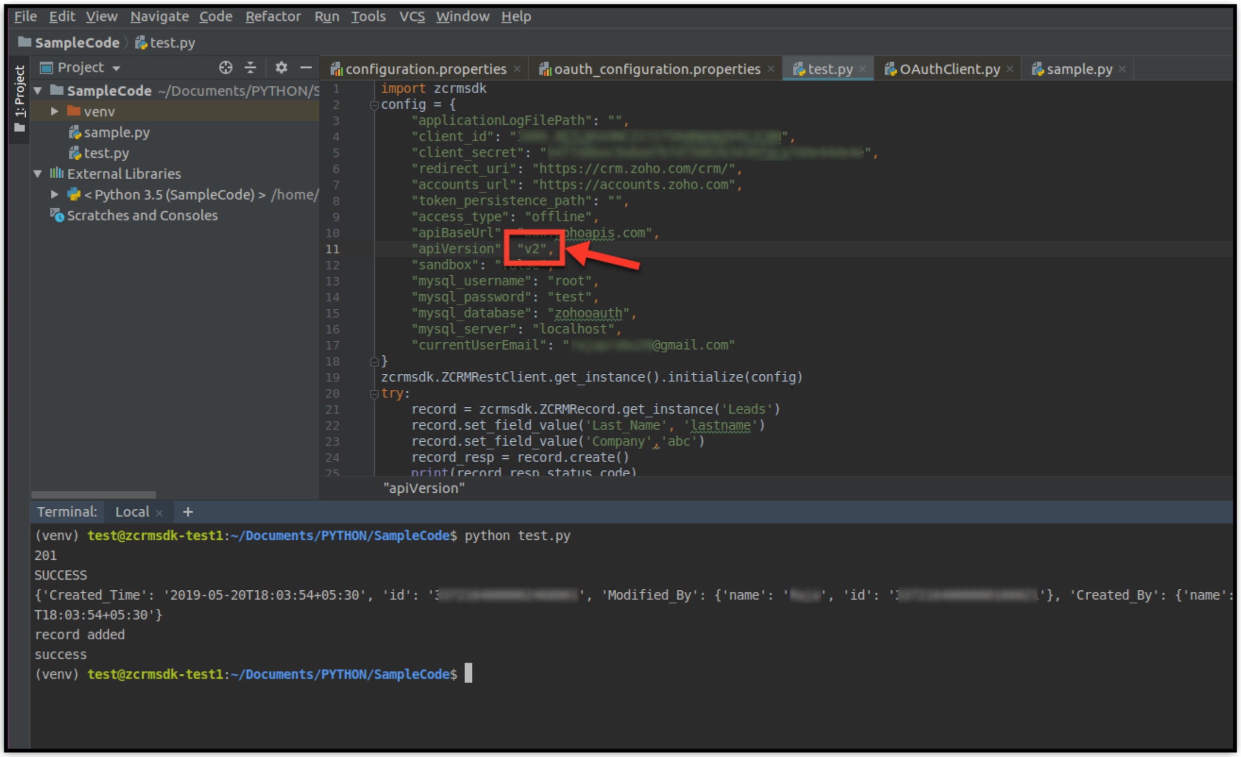
```
(venv) test@zcrmsdk-test1:~/Documents/PYTHON/SampleCode$ python test.py
Traceback (most recent call last):
  File "/home/test/Documents/PYTHON/SampleCode/venv/lib/python3.5/site-packages/zcrmsdk/Request.py", line 49, in get_api_response
    return APIResponse(response,response.status_code,self.url,self.request_api_key)
  File "/home/test/Documents/PYTHON/SampleCode/venv/lib/python3.5/site-packages/zcrmsdk/Response.py", line 72, in __init__
    super(APIResponse,self).__init__(response,status_code,url,apiKey)
  File "/home/test/Documents/PYTHON/SampleCode/venv/lib/python3.5/site-packages/zcrmsdk/Response.py", line 29, in __init__
    self.set_response_json()
  File "/home/test/Documents/PYTHON/SampleCode/venv/lib/python3.5/site-packages/zcrmsdk/Response.py", line 33, in set_response_json
    self.response_json=self.response.json()
  File "/home/test/Documents/PYTHON/SampleCode/venv/lib/python3.5/site-packages/requests/models.py", line 897, in json
    return complexjson.loads(self.text, **kwargs)
  File "/usr/lib/python3.5/json/__init__.py", line 319, in loads
    return _default_decoder.decode(s)
  File "/usr/lib/python3.5/json/decoder.py", line 339, in decode
    obj, end = self.raw_decode(s, idx=_w(s, 0).end())
  File "/usr/lib/python3.5/json/decoder.py", line 357, in raw_decode
    raise JSONDecodeError("Expecting value", s, err.value) from None
json.decoder.JSONDecodeError: Expecting value: line 1 column 1 (char 0)

During handling of the above exception, another exception occurred:
```

Reason

The "apiVersion" key in the configuration dictionary is "V2", with a capital "V".

Solution

Change the value of apiVersion to "v2" in the configuration dictionary.



```
import zcrmsdk
config = {
    "applicationLogFilePath": "",
    "client_id": "[redacted]",
    "client_secret": "[redacted]",
    "redirect_uri": "https://crm.zoho.com/crm/",
    "accounts_url": "https://accounts.zoho.com",
    "token_persistence_path": "",
    "access_type": "offline",
    "apiBaseUrl" [illegible]ohoapis.com",
    "apiVersion" "v2",
    "sandbox": "false",
    "mysql_username": "root",
    "mysql_password": "test",
    "mysql_database": "zohooauth",
    "mysql_server": "localhost",
    "currentUserEmail": "[redacted]@gmail.com"
}
zcrmsdk.ZCRMRestClient.get_instance().initialize(config)
try:
    record = zcrmsdk.ZCRMRecord.get_instance('Leads')
    record.set_field_value('Last_Name', 'lastname')
    record.set_field_value('Company','abc')
    record_resp = record.create()
    print(record_resp.status_code)
```

```
(venv) test@zcrmsdk-test1:~/Documents/PYTHON/SampleCode$ python test.py
201
SUCCESS
{'Created_Time': '2019-05-20T18:03:54+05:30', 'id': '3[redacted]', 'Modified_By': {'name': '[redacted]', 'id': '3[redacted]'}, 'Created_By': {'name':
T18:03:54+05:30'}
record added
success
(venv) test@zcrmsdk-test1:~/Documents/PYTHON/SampleCode$
```

Release Notes

Current Version:

1. ZCRMSDK - VERSION 2.0.13

- Install command

```
pip install zcrmsdk==2.0.13
```

Notes

- This version supports lar_id, trigger, and process keys in the insert and update records operations.

Previous Versions

1. ZCRMSDK - VERSION 2.0.12

- Install command

```
pip install zcrmsdk==2.0.12
```

Notes

- Fixed the "KeyError" in the Get Users operation.

2. ZCRMSDK - VERSION 2.0.11

- Install command

```
pip install zcrmsdk==2.0.11
```

Notes

This version supports additional parameters in the Get Records operation.

3. ZCRMSDK - VERSION 2.0.10

- Install command

```
pip install zcrmsdk==2.0.10
```

Notes

- The SDK throws the right exception when the token is not present in the persisted source.
- Resolved the AttributeError wherever invalid attribute references were made.

4. ZCRMSDK - VERSION 2.0.9

- Install command

```
pip install zcrmsdk==2.0.9
```

Notes

- This version supports pagination and headers in Get Deleted Records, Get Deleted Records from the Recycle Bin, and Get Permanently Deleted Records operations.

5. ZCRMSDK - VERSION 2.0.8

- Install command

```
pip install zcrmsdk==2.0.8
```

Notes

- Fixed the keyError when you retrieve Events using get_records method and the participant(s) was invited only through an email.

6. ZCRMSDK - VERSION 2.0.7

- Install command

```
pip install zcrmsdk==2.0.7
```

Notes

- Added module name check for Product_Details, Pricing_Details and Participants keys in Get Records.
- SDK throws exception if the accounts scope (aaaserver.profile.READ) is not included in the generated grant token or when the user's email cannot be fetched with the generated access token.

7. ZCRMSDK - VERSION 2.0.6

- Install command

```
pip install zcrmsdk==2.0.6
```

Notes

- Prevented directory switch in File Persistence.

8. ZCRMSDK - VERSION 2.0.5

- Install command

```
pip install zcrmsdk==2.0.5
```

Notes

- Handled the change in OAuth token response.

9. ZCRMSDK - VERSION 2.0.4

- Install command

```
pip install zcrmsdk==2.0.4
```

Notes

- "If-Modified-Since" header is supported in get_records().
- Fixed importError in Custom DB Implementation in Python version 3 and below.

10. ZCRMSDK - VERSION 2.0.3

- Install command

```
pip install zcrmsdk==2.0.3
```

Notes

- Blueprint is supported.

11. ZCRMSDK - VERSION 2.0.2

- Install command

```
pip install zcrmsdk==2.0.2
```

Notes

- Issue in BulkAPIResponse is fixed.

12. ZCRMSDK - VERSION 2.0.1

- Install command

```
pip install zcrmsdk==2.0.1
```

Notes

- Organization Taxes are supported.
- Tags are supported.
- Variables and Variable Groups are supported.
- Attachments are supported in the Notes module.

13. ZCRMSDK - VERSION 2.0.0

- Install command

```
pip install zcrmsdk==2.0.0
```

Notes

- This version supports Custom DB Persistence.
- The configuration files (.properties files) are not supported from this version onwards.

14. ZCRMSDK - VERSION 1.0.9

- Install command

```
pip install zcrmsdk==1.0.9
```

Notes

- "page" and "per_page" parameters are now supported in Get Users methods.
- Expiry time window increased to five seconds.

15. ZCRMSDK - VERSION 1.0.8

- Install command

```
pip install zcrmsdk==1.0.8
```

Notes

- Duplicate check fields can be ignored if not necessary in upsert record method.

16. ZCRMSDK - VERSION 1.0.7

- Install command

```
pip install zcrmsdk==1.0.7
```

Notes

- Duplicate check fields can be added in upsert record method.

17. ZCRMSDK - VERSION 1.0.6

- Install command

```
pip install zcrmsdk==1.0.6
```

Notes

- Search records API is now supported based on phone, email, word and criteria.

18. ZCRMSDK - VERSION 1.0.5

- Install command

```
pip install zcrmsdk==1.0.5
```

Notes

- In get_user function, Isonline keyError fixed.

19. ZCRMSDK - VERSION 1.0.4

- Install command

```
pip install zcrmsdk==1.0.4
```

Notes

- Support given to take SDK configuration details as dictionary

20. ZCRMSDK - VERSION 1.0.3

- Install command

```
pip install zcrmsdk==1.0.3
```

Notes

- In use of file for storing tokens.

21. ZCRMSDK - VERSION 1.0.2

- Install command

```
pip install zcrmsdk==1.0.2
```

Notes

- Module Import issue fix in Python3.x.
- Conversion of Record IDs (from string to long) is removed. Now, IDs will be displayed as string only.