Java SDK

Version 2

Zoho CRM

--zoho.com/crm--

-

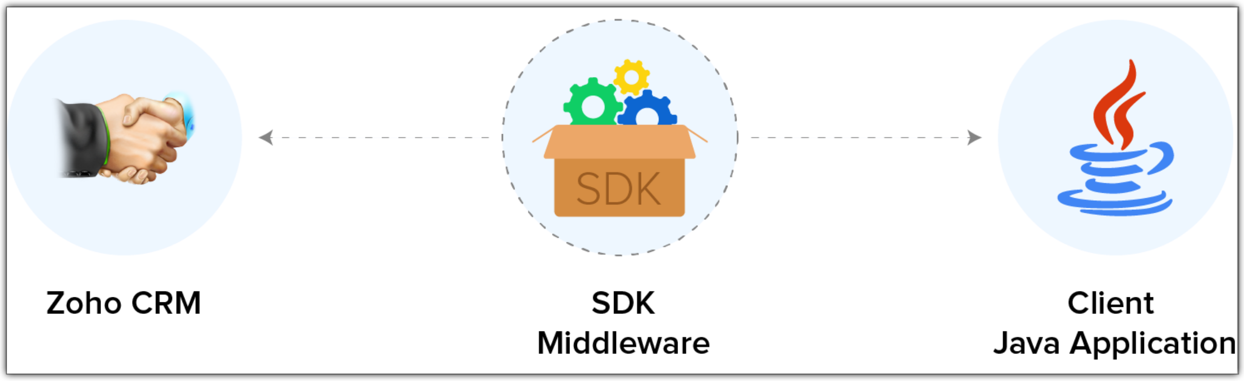
Table of Contents

Overview

Java SDK offers a way to create client java applications that can be integrated with Zoho CRM. This SDK makes the access and use of necessary CRM APIs easy. 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 of by the SDK.

A sample of how an SDK acts a middleware or interface between Zoho CRM and a client Java application.



Java SDK allows you to

1. Exchange data between Zoho CRM and the client application where the CRM entities are modeled as classes.
2. Declare and define CRM API equivalents as simple functions in your Java application.
3. Push data into Zoho CRM by accessing appropriate APIs of the CRM Service.

> **Note**:
> - For the sake of better explanation, we have used Eclipse to describe how to get started on using the SDK.

Environmental Setup

Java SDK requires Java (version 7 and above) to be setup in your development environment.

Including the SDK in your project

Java SDK is available through Maven distribution. You can include the SDK to your project using:

- Maven
- Gradle
- Downloadable JARs (by Zoho)

Maven Distribution

Maven is a build automation tool used primarily for Java projects. Maven addresses two aspects of building software: first, it describes how software is built, and second, it describes its dependencies.

If you are using Maven to build your project, we already have the dependencies set up.

You just need to include the following in your **pom.xml** file, which will get created once your Java project is created using Maven.

```
<repositories>
<repository>
<id>java-sdk</id>
<url>https://maven.zohodl.com</url>
</repository>
</repositories>

<dependencies>
<dependency>
<groupId>com.zoho.crm</groupId>
<artifactId>java-sdk</artifactId>
<version>2.1.3</version>
</dependency>
```

```
</dependencies>
```

Gradle

```
repositories{
    maven { url "https://maven.zohodl.com" }
}
dependencies{
    implementation 'com.zoho.crm:java-sdk:2.1.3'
}
```

Downloadable JAR File
Download SDK

This version downloads simply the SDK without dependent jars. In this case, the following jars are to be made available by adding them in the referenced libraries of your java application. The jars can be downloaded from here.

The list of dependency JARs that you need are:

- hamcrest-all-1.3
- httpclient-4.4.1
- httpcore-4.4.4
- httpmime-4.5.3
- json-20170516
- servlet-api-2.5
- commons-logging-1.1.3
- mysql-connector-java-5.1.44-bin.jar

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.

Using Java SDK through Maven

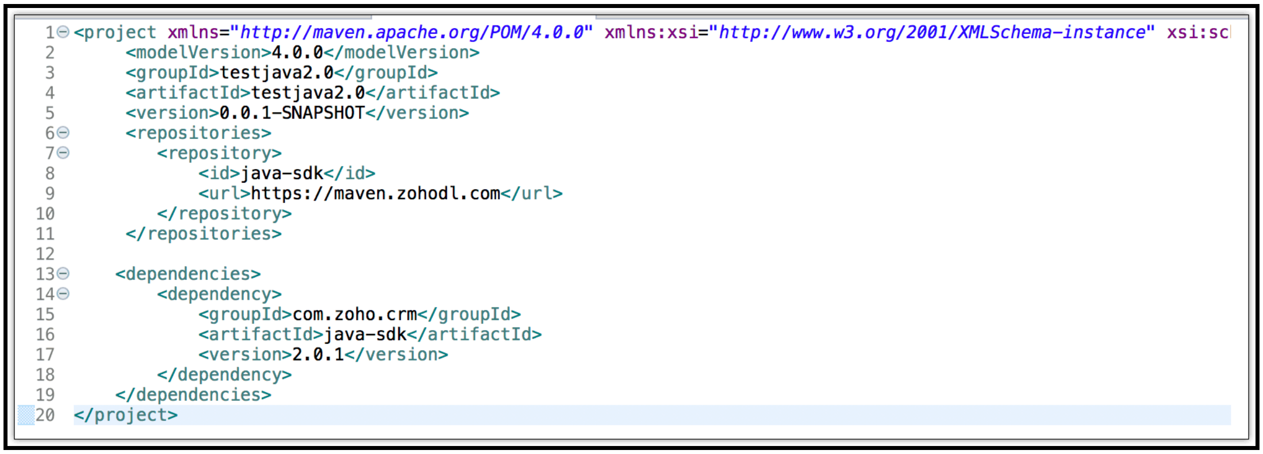
Follow the below steps to use the Java ZCRSMDK through Maven.

1. Create a Maven project.

2. Place the below code in your **pom.xml** file of your Maven project.

```
<repositories>
<repository>
<id>java-sdk</id>
<url>https://maven.zohodl.com</url>
</repository>
</repositories>

<dependencies>
<dependency>
<groupId>com.zoho.crm</groupId>
<artifactId>java-sdk</artifactId>
<version>2.0.1</version>
</dependency>
</dependencies>
```

The below image shows how your pom.xml should look like.



```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:sc
     <modelVersion>4.0.0</modelVersion>
     <groupId>testjava2.0</groupId>
     <artifactId>testjava2.0</artifactId>
     <version>0.0.1-SNAPSHOT</version>
     <repositories>
        <repository>
            <id>java-sdk</id>
            <url>https://maven.zohodl.com</url>
        </repository>
     </repositories>

    <dependencies>
        <dependency>
            <groupId>com.zoho.crm</groupId>
            <artifactId>java-sdk</artifactId>
            <version>2.0.1</version>
        </dependency>
    </dependencies>
</project>
```

3. Update the maven project. Under project explorer, right click the project name, select **Maven > update** project. The jar will be downloaded in the maven dependencies.

4. Inside your source code, import the appropriate files from the SDK.

5. Initialize the ZCRMRestClient class using the following code.

```
HashMap <String, String> zcrmConfigurations = new HashMap <String, String >();
zcrmConfigurations.put("minLogLevel", "ALL");
zcrmConfigurations.put("currentUserEmail","{user_email_id}");
zcrmConfigurations.put("client_id","1000. xxxxxxxxx");
zcrmConfigurations.put("client_secret","xxxxxxxxxxxx");
zcrmConfigurations.put("redirect_uri","https: //crm.zoho.com"/);
zcrmConfigurations.put("persistence_handler_class","com.zoho.oauth.clientapp.Z

"com.zoho.oauth.clientapp.ZohoOAuthFilePersistence" for file, user can
implement his own persistence and provide the path here
zcrmConfigurations.put("oauth_tokens_file_path","relativepath/to/oauthtokens.pr

zcrmConfigurations.put("domainSuffix","com");//optional. Default is com. "cn" and
"eu" supported
zcrmConfigurations.put("accessType","Production");//Production->www(default),
Development->developer, Sandbox->sandbox(optional)
zcrmConfigurations.put("apiBaseUrl","https://www.zohoapis.com");//optional
zcrmConfigurations.put("iamURL","https://accounts.zoho.com");//optional
zcrmConfigurations.put("logFilePath","/path/to/file.log");//optional
zcrmConfigurations.put("mysql_passsword","pass@123");//optional
zcrmConfigurations.put("mysql_username",username");//optional
ZCRMRestClient.initialize(zcrmConfigurations);//for initializing
```

6. Generate the grant token when you initialize your application.

7. Use one of the below methods to generate the access token

- from the **grant token**.

```
ZohoOAuthClient client = ZohoOAuthClient.getInstance();
client.generateAccessToken("generated_grant_token");
```

- from the **refresh token**.

```
ZohoOAuthClient client = ZohoOAuthClient.getInstance();
client.generateAccessTokenFromRefreshToken("refresh_token");
```

8. You can now access the functionalities of the SDK. Refer to sample codes to make various API calls through the SDK.

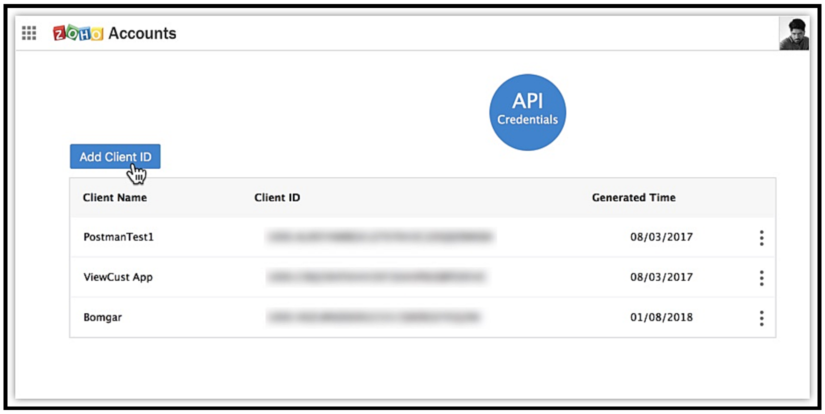
Register your application

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**

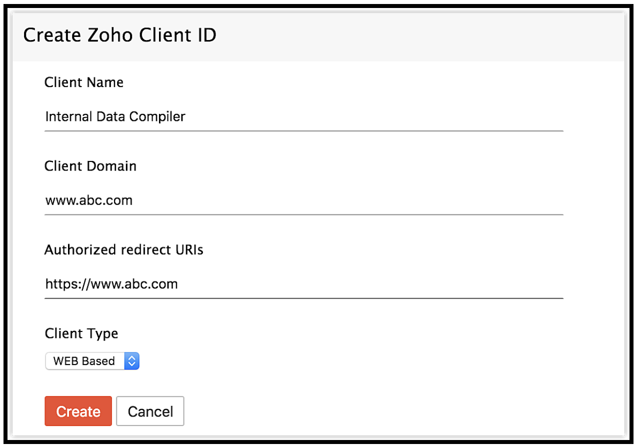
2. Click **Add Client ID**.



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.

Configuration

To access the CRM services through the SDK, the client application must be first authenticated. This can be done by passing a key-value configuration pair to the initialization process.

The configuration map with the required and optional keys is given below. You must include this map with the required keys in every API call you make through the SDK.

> **Note**
> From Java SDK version 2.0.0, only configuration map is supported.

```
HashMap <String, String> zcrmConfigurations = new HashMap <String, String >();
zcrmConfigurations.put("minLogLevel", "ALL");
zcrmConfigurations.put("currentUserEmail","{user_email_id}");
zcrmConfigurations.put("client_id","1000. xxxxxxxxx");
zcrmConfigurations.put("client_secret","xxxxxxxxxxxx");
zcrmConfigurations.put("redirect_uri","https://crm.zoho.com"/);
zcrmConfigurations.put("persistence_handler_class","com.zoho.oauth.clientapp.Zo

"com.zoho.oauth.clientapp.ZohoOAuthFilePersistence" for file, user can implement
his own persistence and provide the path here
zcrmConfigurations.put("oauth_tokens_file_path","relativepath/to/oauthtokens.prop

zcrmConfigurations.put("domainSuffix","com");//optional. Default is com. "cn" and
"eu" supported
zcrmConfigurations.put("accessType","Production");//Production->www(default),
Development->developer, Sandbox->sandbox(optional)
zcrmConfigurations.put("access_type","offline");//optional
zcrmConfigurations.put("apiBaseUrl","https://www.zohoapis.com");//optional
zcrmConfigurations.put("iamURL","https://accounts.zoho.com");//optional
zcrmConfigurations.put("logFilePath","/path/to/file.log");//optional
zcrmConfigurations.put("mysql_password","pass@123");//optional
zcrmConfigurations.put("mysql_username",username");//optional
ZCRMRestClient.initialize(zcrmConfigurations);//for initializing
```

Mandatory Keys

- **client_id, client_secret, and redirect_uri** are your OAuth client’s configurations that you get after registering your Zoho client.

- **currentUserEmail** is the email ID of the current user.

- **persistence_handler_class** is the implementation of ZohoOAuthPersistenceInterface. Refer to Java Persistence for more details.

Optional Keys

- **logFilePath** is the absolute file path of the file that stores the logs.

- **domainSuffix** represents the domain from which API calls are made. For US, use "com"(default). For EU, use "eu", and use "cn" for the CN domain.

- **accessType** represents the account from which the API calls are made. Possible values are "Production", "Development", and "Sandbox".

- **access_type** must be set only to "offline" as online OAuth client is not supported by the Java SDK as of now.

- **apiBaseUrl** - URL to be used when calling an API. It denotes the domain of the user. This URL may be:
  - https://www.zohoapis.com (default)
  - https://www.zohoapis.eu
  - https://www.zohoapis.com.cn
  - https://www.zohoapis.jp

- **iamURL** - Default value is set as US domain. This value can be changed based on your domain (EU, CN).

    - https://accounts.zoho.com
    - https://accounts.zoho.eu
    - https://accounts.zoho.com.cn
    - https://accounts.zoho.jp

- **mysql_passsword** and **mysql_username** represent the **MySQL** username and password, respectively. Please note that these keys are mandatory when you use **MySQL DB** persistence.

> **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.

Persistence

Persistent classes in an application implement the entities of the business problem. In Java SDK, two default persistence classes with their implementations are provided. On the other hand, if a developer wants his specific implementation, he can define his custom implementation by connecting to their respective database system.

Implementing OAuth Persistence

Once the application is authorized, OAuth access and refresh tokens can be used for subsequent user data requests to Zoho CRM. Hence, they need to be persisted by the client app.

The persistence is achieved by writing an implementation of the inbuilt **ZohoPersistenceHandler** interface, which has the following callback methods.

- **saveOAuthData(ZohoOAuthTokens tokens)** — invoked while fetching access and refresh tokens from Zoho.

- **deleteOAuthTokens()**— invoked before saving the newly received tokens.

- **getOAuthTokens()** — invoked before firing a request to fetch the saved tokens. This method should return ZohoOAuthTokens object for the library to process it.

Our Java SDK provides two sample implementations of **ZohoPersistenceHandler** interface within the client library, and you can use any one of the two interface implementations.

ZohoOAuthFilePersistence

ZohoOAuthFilePersistence uses a file to write and read the OAuth tokens data. If you are using this persistence, then you need to provide the following attributes and their values in the **configuration map**.

```
persistence_handler_class=com.zoho.oauth.clientapp.ZohoOAuthFilePersistence
oauth_tokens_file_path= relative path of the "oauthtokens.properties" file in the workspace.
```

ZohoOAuthDBPersistence

ZohoOAuthDBPersistence uses **MySQL** persistence. To use this, you should make sure of the following.

- **MySQL** should be running in the same machine serving at the default port 3306.
- The database name should be **"zohooauth"**.
- There must be a table "**oauthtokens**" with the columns "**useridentifier**" (varchar), "**accesstoken**" (varchar), "**refreshtoken**" (varchar) and "**expirytime**" (bigint).

If you are using **ZohoOAuthDBPersistence**, then you need to provide the following attributes and values in the **configuration map**.

```
persistence_handler_class=com.zoho.oauth.clientapp.ZohoOAuthDBPersistence
mysql_username=root
mysql_password=****
```

Note:

- **ZohoOAuthFilePersistence** implementation only supports to store and refresh only a single user's token. Hence this shall be used if your app accesses Zoho APIs on behalf of a single user only.
- In case if your app has to support for multiple users, please use **ZohoOAuthDBPersistence** or write your own implementation of **ZohoPersistenceHandler**.

Initialization

Now the app is ready to be initialized after defining OAuth configuration file and OAuth persistence handler class for your app.

Generating the grant token

For a Single User

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.

1. Login to the User's account.
2. Visit https://api-console.zoho.com/
3. Click on the **Options** → **Self Client** option of the client for which you wish to authorize.
4. 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.

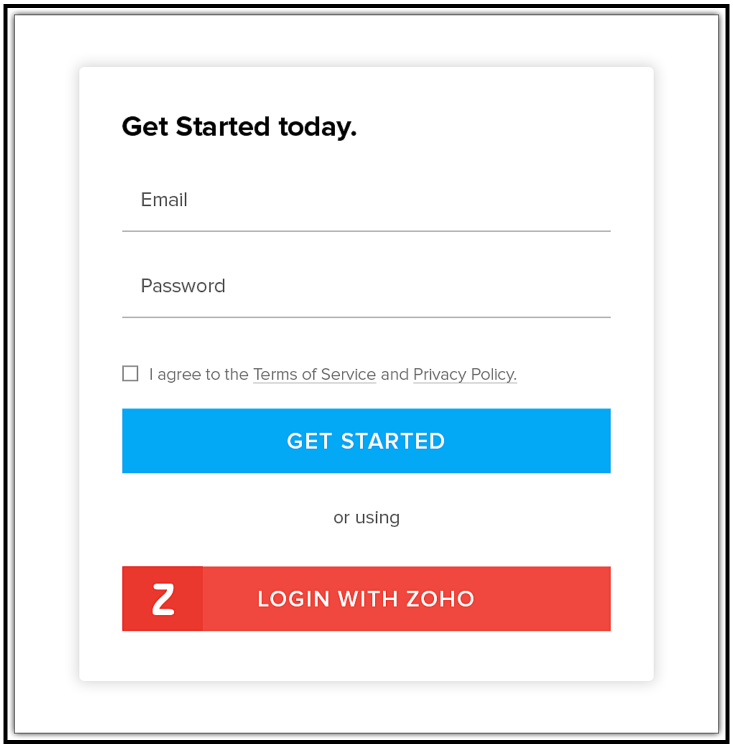
5. Copy the grant token that is displayed on the screen.

> **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.



□ I agree to the Terms of Service and Privacy Policy.

- Upon successful login of the user, the grant token will be sent as a param to your registered redirect URL.

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.

Generating Access and Refresh tokens from Grant Token

After obtaining the grant token, the following code snippet should be put in a java file and executed from your main class to get access and refresh tokens.

```
ZCRMRestClient.initialize(configurations_map);
ZohoOAuthClient cli = ZohoOAuthClient.getInstance();
String grantToken = “paste_the_self_authorized_grant_token_here”;
ZohoOAuthTokens tokens = cli.generateAccessToken(grantToken);
String accessToken = tokens.getAccessToken();
String refreshToken = tokens.getRefreshToken();
System.out.println("access token = " + accessToken + " & refresh token = " +
 refreshToken);
```

Access and refresh tokens are generated and stored in the **oauth_tokens.properties** file if you are using **ZohoOAuthFilePersistence** and the same will be stored in your **Mysql** database table "**oauthtokens**" if you are using **ZohoOAuthDBPersistence**, on executing the above java code.

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**.

Generating Access token from Refresh Token

You can now generate access token from the refresh token. Use the below code snippet.

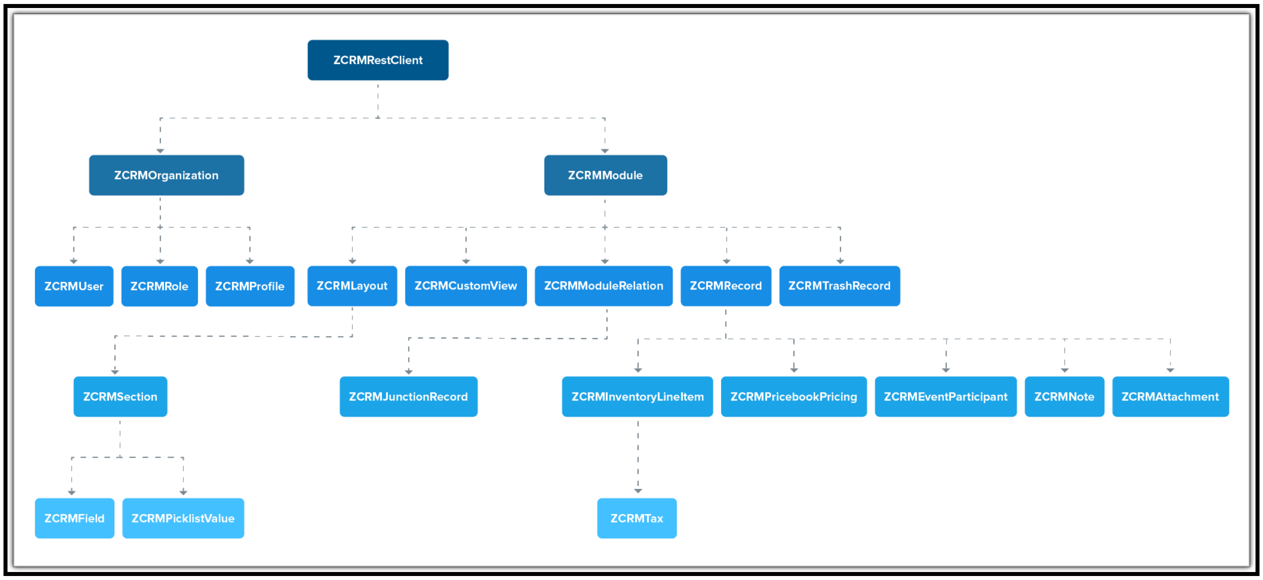
```
ZohoOAuthClient client = ZohoOAuthClient.getInstance();
client.generateAccessTokenFromRefreshToken(refreshToken, userMailId);
```

Class Hierarchy

All Zoho CRM entities are modeled as classes having members and methods applicable to that particular entity.

- **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 hierarchy of various Zoho CRM entities is depicted as:



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 every level. In order to handle this, every entity class will have a **getInstance()** method to get its own dummy object and methods 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.getModule("Contacts")** would return the actual Contacts module, that has all the properties of the Contacts module filled through an API call.

- **ZCRMRestClient.getModuleInstance("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, there's no need to start from **ZCRMRestClient**. Instead, you could get a **ZCRMModule** instance with **ZCRMModule.getInstance()** and then invoke its non-static **getRecords()** method from the created instance. This would avoid the API call that would otherwise 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.getFieldValue(fieldAPIName);
```

- To set a field value, use

```
record.setFieldValue(fieldAPIName, newValue);
```

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 API calling methods would 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.
- **FileAPIResponse** will be returned for file download APIs to download a photo or an attachment from a record or note such as **record.downloadPhoto**, **record.downloadAttachment** etc.
- Use the **getData()** method to get the entity data alone from the response wrapper objects. **APIResponse.getData()** would return a single Zoho CRM entity object, while **BulkAPIResponse.getData()** would return a list of Zoho CRM entity objects.
- **FileAPIResponse** has two defined methods namely **FileAPIResponse.getFileName()** which returns the name of the file that is downloaded and **FileAPIResponse.getFileAsStream()** that gives the file content as InputStream.

Other than data, these response wrapper objects have the following properties:
1. **ResponseHeaders** - remaining API counts for the present day/window and time elapsed for the present window reset.

2. **ResponseInfo** - any other information, if provided by the API, in addition to the actual data.

3. **List<EntityResponse>** - status of 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.

Start the App

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

```
ZCRMRestClient.initialize(configuration_map);
```

This method should be called from the main class of your java 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 Java 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
getOrganizationDetails	To fetch all the details regarding your organization in your CRM account.
getCurrentUser	To fetch information about the user who is currently accessing Zoho CRM's data through your application.
getCurrentUserEmailID	To fetch the email ID of the user, who's is currently accessing CRM's data.
getAllModules	To fetch the list of all the modules available in your CRM account.

getModule	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
getUser	To fetch information about a specific user in your CRM account.
getAllUsers	To fetch the list of all the users from your CRM account.
getAllActiveUsers	To fetch the list of all the active users in your CRM account.
getAllDeactiveUsers	To fetch the list of all the non-active users in your CRM account.
getAllActiveConfirmedUsers	To fetch the list of all the active and confirmed users in your CRM account.
getAllAdminUsers	To fetch the list of all the users who have admin level permissions in your CRM account.
	To create a new user in your CRM

createUser	account.
updateUser	To update details of an existing user in your CRM account.
deleteUser	To delete a user from your CRM account.
getAllProfiles	To fetch the list of all the profiles that were created in your CRM account.
getUser	To fetch information about a specific user in your CRM account.
getAllUsers	To fetch the list of all the users from your CRM account.
getAllActiveUsers	To fetch the list of all the active users in your CRM account.
getAllDeactiveUsers	To fetch the list of all the non-active users in your CRM account.
getAllActiveConfirmedUsers	To fetch the list of all the active and confirmed users in your CRM account.
getAllAdminUsers	To fetch the list of all the users who have admin level permissions in your CRM account.
createUser	To create a new user in your CRM

	account.
updateUser	To update details of an existing user in your CRM account.
deleteUser	To delete a user from your CRM account.

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.

> **Note**
> Please note that the updateRecords() method of SDK 1.x.x has changed to massUpdateRecords() from version 2.

Methods	Description
getAllFields	To fetch the list of all the fields that are available in a module.
getLayoutDetails	To fetch information about a particular layout of a module.
getAllLayouts	To fetch the list of all the layouts that are available for a module.
getCustomView	To fetch information about a particular custom view of a module.

getAllCustomViews	To fetch the list of all the custom views that are available for a module.
getAllRelatedLists	To fetch the list of all the related lists that are available for a module.
getRecords	To fetch the list of all the records that are available in a module.
searchRecordsByWord	To search for records in a module based on a Word(text).
searchRecordsByPhone	To search for records in a module based on the Phone number.
searchRecordsByEmail	To search for records in a module based on Email address.
searchRecordsByCriteria	To search for records in a module based on a criteria specified by the user.
updateRecords	To update details of multiple records in a module.
massUpdateRecords	To update details of a single field across the given record IDs in a module.
createRecords	To create a new record in a module.
deleteRecords	To delete existing records from a module.

getAllDeletedRecords	To fetch the list of all the records that were deleted from a module.
getRecycleBinRecords	To fetch the list of all the records that were deleted from a module and stored in the recycle bin.
getPermanentlyDeletedRecords	To fetch the list of all the records that were permanently deleted from a module.
upsertRecords	To insert/update records in a module.
getTags	To fetch the list of all the tags that were created for a module.
getTagCount	To fetch total count of the tags that were created for a module.
createTags	To create new tags for a module.
updateTags	To update details of existing tags for a module.
addTagMultipleRecord	To associate tags to records in a module.
removeTagMultipleRecord	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	To create new records.
update	To update existing records.
delete	To delete existing records.
convert	To convert records(Leads to Contacts/Deals).
getNotes	To fetch the notes that were attached to a record.
addNote	To add a note to a record.
updateNote	To update a note that was previously added to a record.
deleteNote	To delete a note from a record.
getAttachments	To fetch the list of attachments of a record.
uploadAttachment	To upload an attachment to a record.

uploadLinkAsAttachment	To upload a link as an attachment to a record.
downloadAttachment	To download an attachment that was uploaded to a record.
deleteAttachment	To delete an attachment that was added to a record.
uploadPhoto	To upload a photo to a record.
downloadPhoto	To download a photo that was added to a record.
deletePhoto	To delete a photo that was added to a record.
addRelation	To make a relation between two records.
removeRelation	To remove a relation between two records.
addTagSpecificRecord	To add tags to a specific record.
removeTagSpecificRecord	To remove tags from a specific record.
getRelatedListRecords	To get the list of related list records.

Tag Operations

These methods allow you to add or remove tags in your CRM

Methods	Description
updateTag	To update details of an existing tag.
deleteTag	To delete a tag.
mergeTag	To merge two tags.

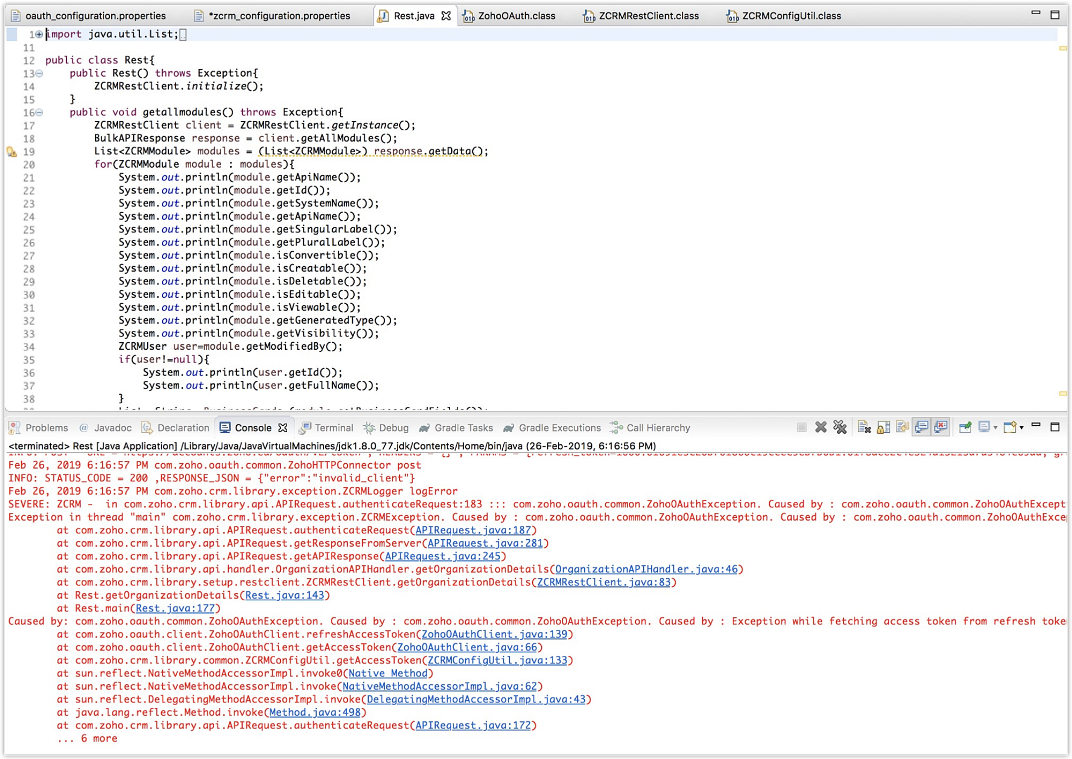
Variable Operations

These methods allow you to add, update, and delete variables for your organization.

Methods	Description
getVariable	To fetch information about a specific variable created for your organization.
updateVariable	To update the details of a specific variable.
deleteVariable	To delete a specific variable.
getVariableGroup	To fetch information about a specific variable group created for your organization.

Errors and solutions

1. Exception while fetching access tokens from refresh tokens

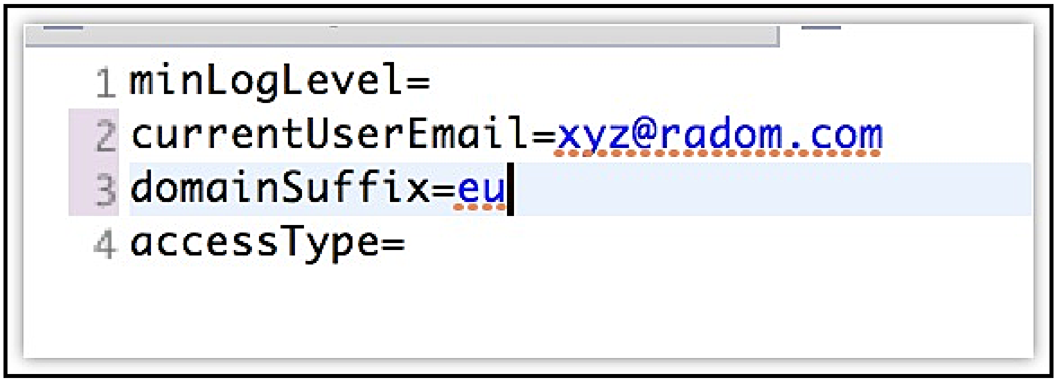


```java
import java.util.List;

public class Rest{
    public Rest() throws Exception{
        ZCRMRestClient.initialize();
    }
    public void getallmodules() throws Exception{
        ZCRMRestClient client = ZCRMRestClient.getInstance();
        BulkAPIResponse response = client.getAllModules();
        List<ZCRMModule> modules = (List<ZCRMModule>) response.getData();
        for(ZCRMModule module : modules){
            System.out.println(module.getApiName());
            System.out.println(module.getId());
            System.out.println(module.getSystemName());
            System.out.println(module.getApiName());
            System.out.println(module.getSingularLabel());
            System.out.println(module.getPluralLabel());
            System.out.println(module.isConvertible());
            System.out.println(module.isCreatable());
            System.out.println(module.isDeletable());
            System.out.println(module.isEditable());
            System.out.println(module.isViewable());
            System.out.println(module.getGeneratedType());
            System.out.println(module.getVisibility());
            ZCRMUser user=module.getModifiedBy();
            if(user!=null){
                System.out.println(user.getId());
                System.out.println(user.getFullName());
            }
```

```
<terminated> Rest [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_77.jdk/Contents/Home/bin/java (26-Feb-2019, 6:16:56 PM)
Feb 26, 2019 6:16:57 PM com.zoho.oauth.common.ZohoHTTPConnector post
INFO: STATUS_CODE = 200 ,RESPONSE_JSON = {"error":"invalid_client"}
Feb 26, 2019 6:16:57 PM com.zoho.crm.library.exception.ZCRMLogger logError
SEVERE: ZCRM -  in com.zoho.crm.library.api.APIRequest.authenticateRequest:183 ::: com.zoho.oauth.common.ZohoOAuthException. Caused by : com.zoho.oauth.common.ZohoOAuthExcept
Exception in thread "main" com.zoho.crm.library.exception.ZCRMException. Caused by : com.zoho.oauth.common.ZohoOAuthException. Caused by : com.zoho.oauth.common.ZohoOAuthExce
	at com.zoho.crm.library.api.APIRequest.authenticateRequest(APIRequest.java:187)
	at com.zoho.crm.library.api.APIRequest.getResponseFromServer(APIRequest.java:281)
	at com.zoho.crm.library.api.APIRequest.getAPIResponse(APIRequest.java:245)
	at com.zoho.crm.library.api.handler.OrganizationAPIHandler.getOrganizationDetails(OrganizationAPIHandler.java:46)
	at com.zoho.crm.library.setup.restclient.ZCRMRestClient.getOrganizationDetails(ZCRMRestClient.java:83)
	at Rest.getOrganizationDetails(Rest.java:143)
	at Rest.main(Rest.java:177)
Caused by: com.zoho.oauth.common.ZohoOAuthException. Caused by : com.zoho.oauth.common.ZohoOAuthException. Caused by : Exception while fetching access token from refresh toke
	at com.zoho.oauth.client.ZohoOAuthClient.refreshAccessToken(ZohoOAuthClient.java:139)
	at com.zoho.oauth.client.ZohoOAuthClient.getAccessToken(ZohoOAuthClient.java:66)
	at com.zoho.crm.library.common.ZCRMConfigUtil.getAccessToken(ZCRMConfigUtil.java:133)
	at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
	at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
	at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
	at java.lang.reflect.Method.invoke(Method.java:498)
	at com.zoho.crm.library.api.APIRequest.authenticateRequest(APIRequest.java:172)
	... 6 more
```

Reason

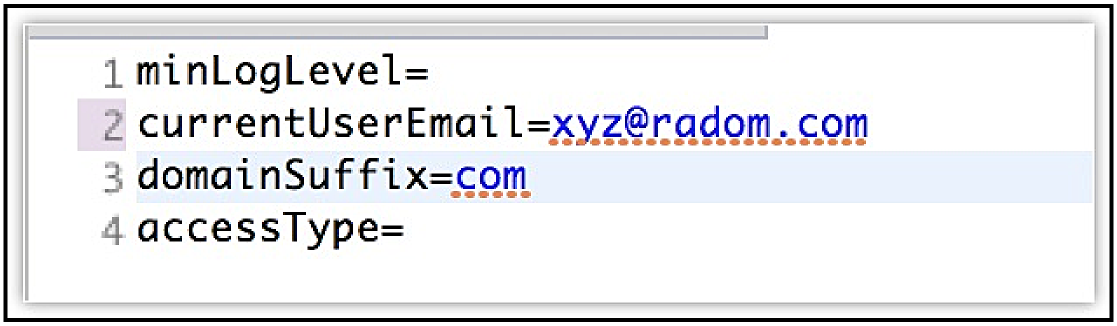
The grant token was generated from the "accounts.zoho.com" domain, but domain suffix was given as "eu" domain.



```
minLogLevel=
currentUserEmail=xyz@radom.com
domainSuffix=eu
accessType=
```

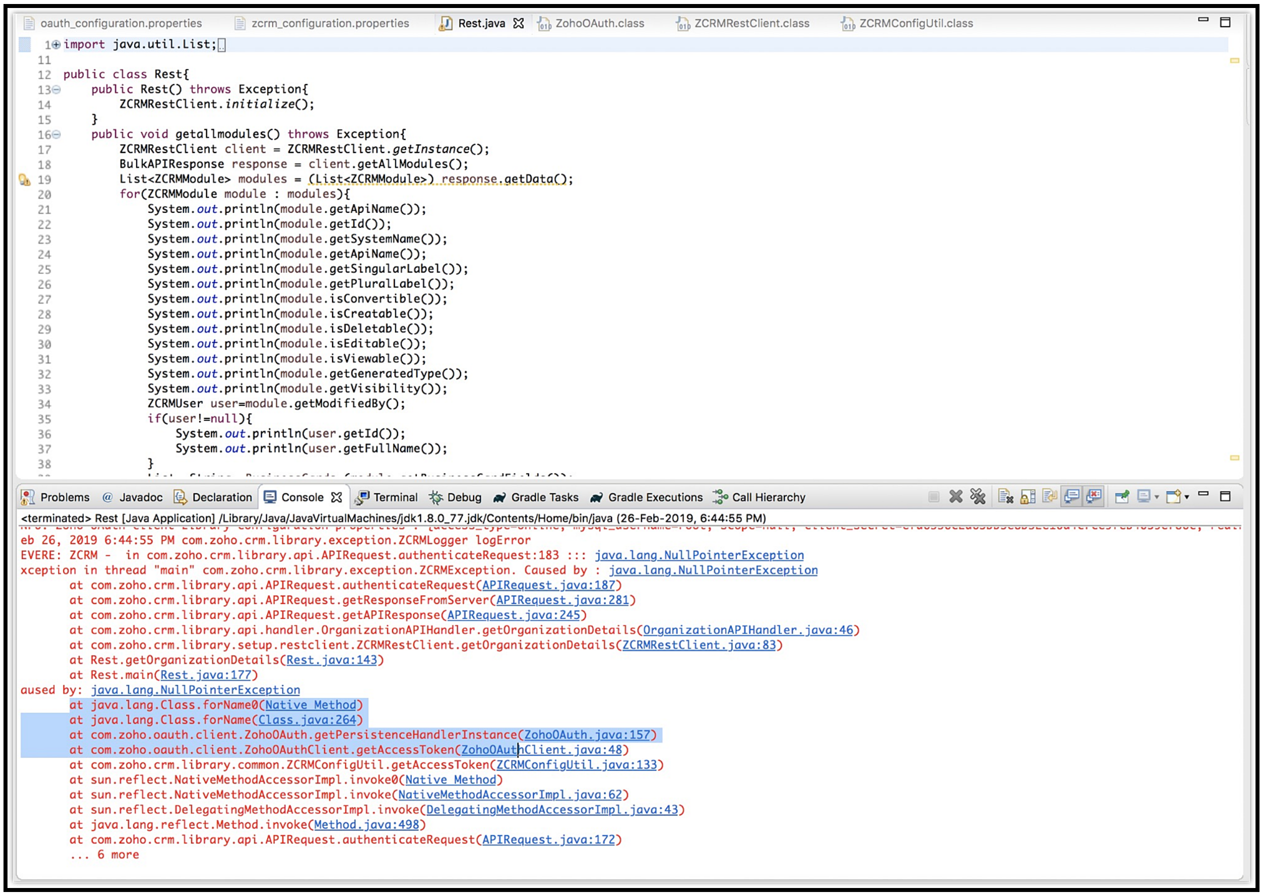
Solution

Provide the same domain suffix in the "zcrm_configuration.properties" file.



```
minLogLevel=
currentUserEmail=xyz@radom.com
domainSuffix=com
accessType=
```

2. java.lang.NullPointerException

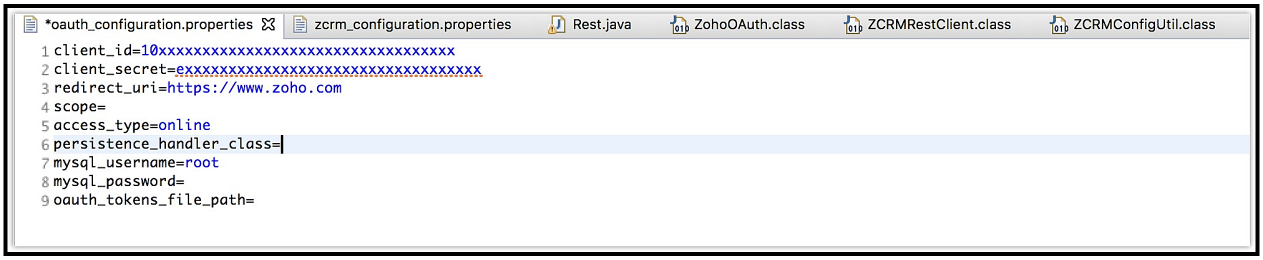


```
import java.util.List;

public class Rest{
    public Rest() throws Exception{
        ZCRMRestClient.initialize();
    }
    public void getallmodules() throws Exception{
        ZCRMRestClient client = ZCRMRestClient.getInstance();
        BulkAPIResponse response = client.getAllModules();
        List<ZCRMModule> modules = (List<ZCRMModule>) response.getData();
        for(ZCRMModule module : modules){
            System.out.println(module.getApiName());
            System.out.println(module.getId());
            System.out.println(module.getSystemName());
            System.out.println(module.getApiName());
            System.out.println(module.getSingularLabel());
            System.out.println(module.getPluralLabel());
            System.out.println(module.isConvertible());
            System.out.println(module.isCreatable());
            System.out.println(module.isDeletable());
            System.out.println(module.isEditable());
            System.out.println(module.isViewable());
            System.out.println(module.getGeneratedType());
            System.out.println(module.getVisibility());
            ZCRMUser user=module.getModifiedBy();
            if(user!=null){
                System.out.println(user.getId());
                System.out.println(user.getFullName());
            }
```

```
<terminated> Rest [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_77.jdk/Contents/Home/bin/java (26-Feb-2019, 6:44:55 PM)
eb 26, 2019 6:44:55 PM com.zoho.crm.library.exception.ZCRMLogger logError
EVERE: ZCRM -  in com.zoho.crm.library.api.APIRequest.authenticateRequest:183 ::: java.lang.NullPointerException
xception in thread "main" com.zoho.crm.library.exception.ZCRMException. Caused by : java.lang.NullPointerException
	at com.zoho.crm.library.api.APIRequest.authenticateRequest(APIRequest.java:187)
	at com.zoho.crm.library.api.APIRequest.getResponseFromServer(APIRequest.java:281)
	at com.zoho.crm.library.api.APIRequest.getAPIResponse(APIRequest.java:245)
	at com.zoho.crm.library.api.handler.OrganizationAPIHandler.getOrganizationDetails(OrganizationAPIHandler.java:46)
	at com.zoho.crm.library.setup.restclient.ZCRMRestClient.getOrganizationDetails(ZCRMRestClient.java:83)
	at Rest.getOrganizationDetails(Rest.java:143)
	at Rest.main(Rest.java:177)
aused by: java.lang.NullPointerException
	at java.lang.Class.forName0(Native Method)
	at java.lang.Class.forName(Class.java:264)
	at com.zoho.oauth.client.ZohoOAuth.getPersistenceHandlerInstance(ZohoOAuth.java:157)
	at com.zoho.oauth.client.ZohoOAuthClient.getAccessToken(ZohoOAuthClient.java:48)
	at com.zoho.crm.library.common.ZCRMConfigUtil.getAccessToken(ZCRMConfigUtil.java:133)
	at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
	at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
	at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
	at java.lang.reflect.Method.invoke(Method.java:498)
	at com.zoho.crm.library.api.APIRequest.authenticateRequest(APIRequest.java:172)
	... 6 more
```

Reason

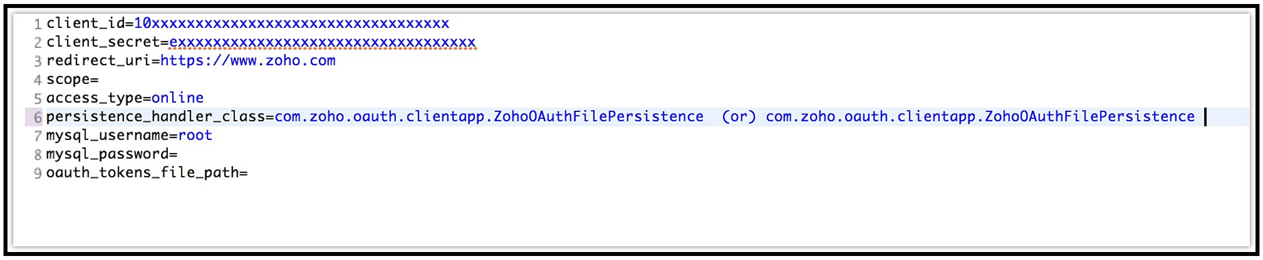
The user has not specified the persistence handler class.



```
client_id=10xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
client_secret=exxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
redirect_uri=https://www.zoho.com
scope=
access_type=online
persistence_handler_class=
mysql_username=root
mysql_password=
oauth_tokens_file_path=
```

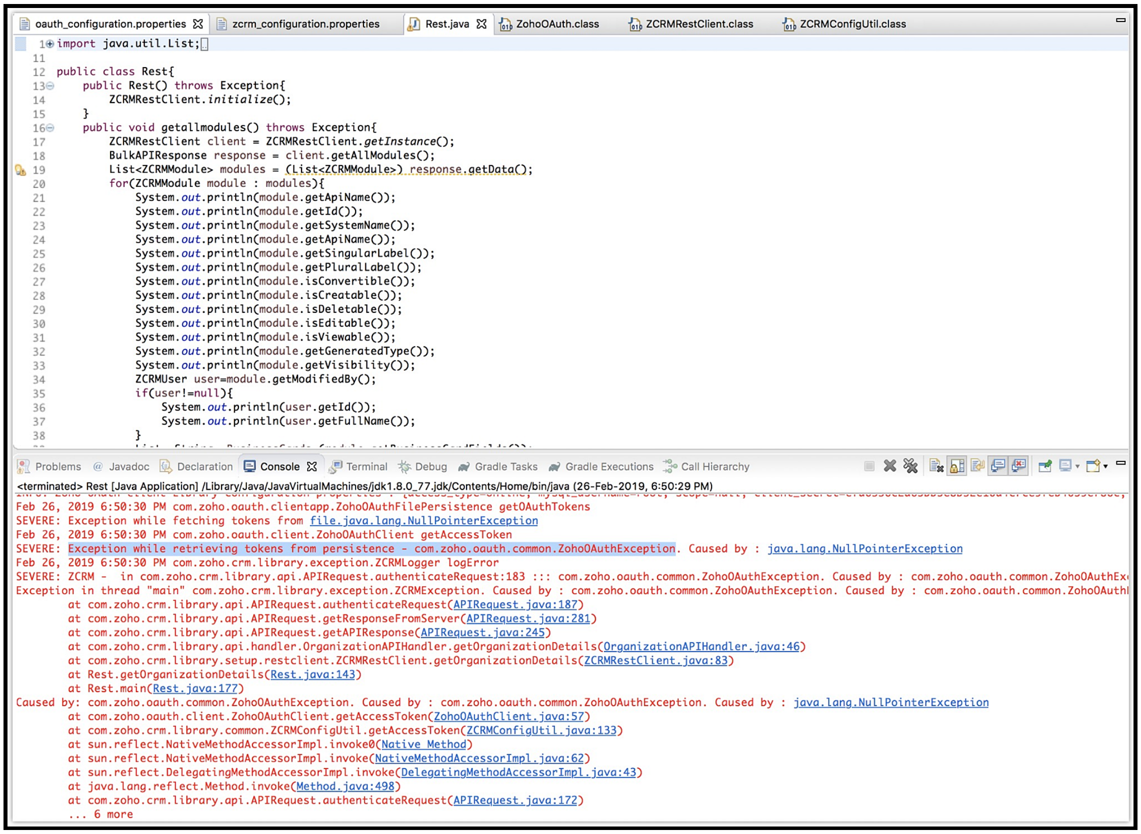
Solution

Specify the persistence handler class in the "oauth_configuration.properties" file.



```
client_id=10xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
client_secret=exxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
redirect_uri=https://www.zoho.com
scope=
access_type=online
persistence_handler_class=com.zoho.oauth.clientapp.ZohoOAuthFilePersistence  (or) com.zoho.oauth.clientapp.ZohoOAuthFilePersistence
mysql_username=root
mysql_password=
oauth_tokens_file_path=
```

3. Exception while retrieving tokens from persistence

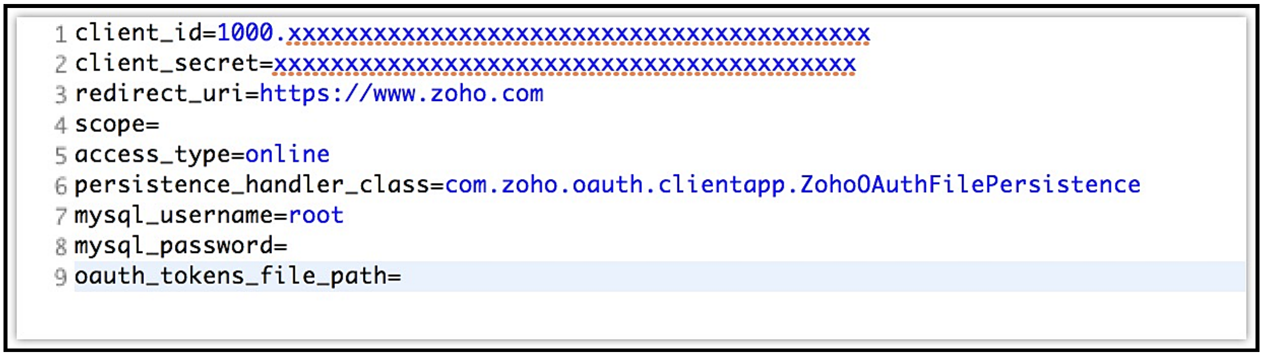


```java
import java.util.List;

public class Rest{
    public Rest() throws Exception{
        ZCRMRestClient.initialize();
    }
    public void getallmodules() throws Exception{
        ZCRMRestClient client = ZCRMRestClient.getInstance();
        BulkAPIResponse response = client.getAllModules();
        List<ZCRMModule> modules = (List<ZCRMModule>) response.getData();
        for(ZCRMModule module : modules){
            System.out.println(module.getApiName());
            System.out.println(module.getId());
            System.out.println(module.getSystemName());
            System.out.println(module.getApiName());
            System.out.println(module.getSingularLabel());
            System.out.println(module.getPluralLabel());
            System.out.println(module.isConvertible());
            System.out.println(module.isCreatable());
            System.out.println(module.isDeletable());
            System.out.println(module.isEditable());
            System.out.println(module.isViewable());
            System.out.println(module.getGeneratedType());
            System.out.println(module.getVisibility());
            ZCRMUser user=module.getModifiedBy();
            if(user!=null){
                System.out.println(user.getId());
                System.out.println(user.getFullName());
            }
```

```
<terminated> Rest [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_77.jdk/Contents/Home/bin/java (26-Feb-2019, 6:50:29 PM)
Feb 26, 2019 6:50:30 PM com.zoho.oauth.clientapp.ZohoOAuthFilePersistence getOAuthTokens
SEVERE: Exception while fetching tokens from file.java.lang.NullPointerException
Feb 26, 2019 6:50:30 PM com.zoho.oauth.client.ZohoOAuthClient getAccessToken
SEVERE: Exception while retrieving tokens from persistence - com.zoho.oauth.common.ZohoOAuthException. Caused by : java.lang.NullPointerException
Feb 26, 2019 6:50:30 PM com.zoho.crm.library.exception.ZCRMLogger logError
SEVERE: ZCRM -  in com.zoho.crm.library.api.APIRequest.authenticateRequest:183 ::: com.zoho.oauth.common.ZohoOAuthException. Caused by : com.zoho.oauth.common.ZohoOAuthEx
Exception in thread "main" com.zoho.crm.library.exception.ZCRMException. Caused by : com.zoho.oauth.common.ZohoOAuthException. Caused by : com.zoho.oauth.common.ZohoOAuth
        at com.zoho.crm.library.api.APIRequest.authenticateRequest(APIRequest.java:187)
        at com.zoho.crm.library.api.APIRequest.getResponseFromServer(APIRequest.java:281)
        at com.zoho.crm.library.api.APIRequest.getAPIResponse(APIRequest.java:245)
        at com.zoho.crm.library.api.handler.OrganizationAPIHandler.getOrganizationDetails(OrganizationAPIHandler.java:46)
        at com.zoho.crm.library.setup.restclient.ZCRMRestClient.getOrganizationDetails(ZCRMRestClient.java:83)
        at Rest.getOrganizationDetails(Rest.java:143)
        at Rest.main(Rest.java:177)
Caused by: com.zoho.oauth.common.ZohoOAuthException. Caused by : com.zoho.oauth.common.ZohoOAuthException. Caused by : java.lang.NullPointerException
        at com.zoho.oauth.client.ZohoOAuthClient.getAccessToken(ZohoOAuthClient.java:57)
        at com.zoho.crm.library.common.ZCRMConfigUtil.getAccessToken(ZCRMConfigUtil.java:133)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at com.zoho.crm.library.api.APIRequest.authenticateRequest(APIRequest.java:172)
        ... 6 more
```

Reason

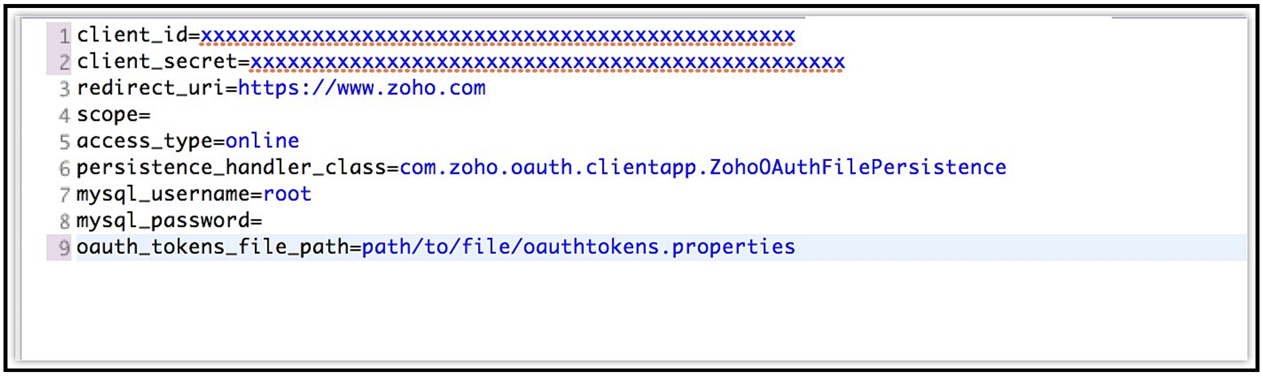
The "oauth_token_file_path" key was not specified in the "oauth_configuration.properties" file.



```
client_id=1000.xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
client_secret=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
redirect_uri=https://www.zoho.com
scope=
access_type=online
persistence_handler_class=com.zoho.oauth.clientapp.ZohoOAuthFilePersistence
mysql_username=root
mysql_password=
oauth_tokens_file_path=
```

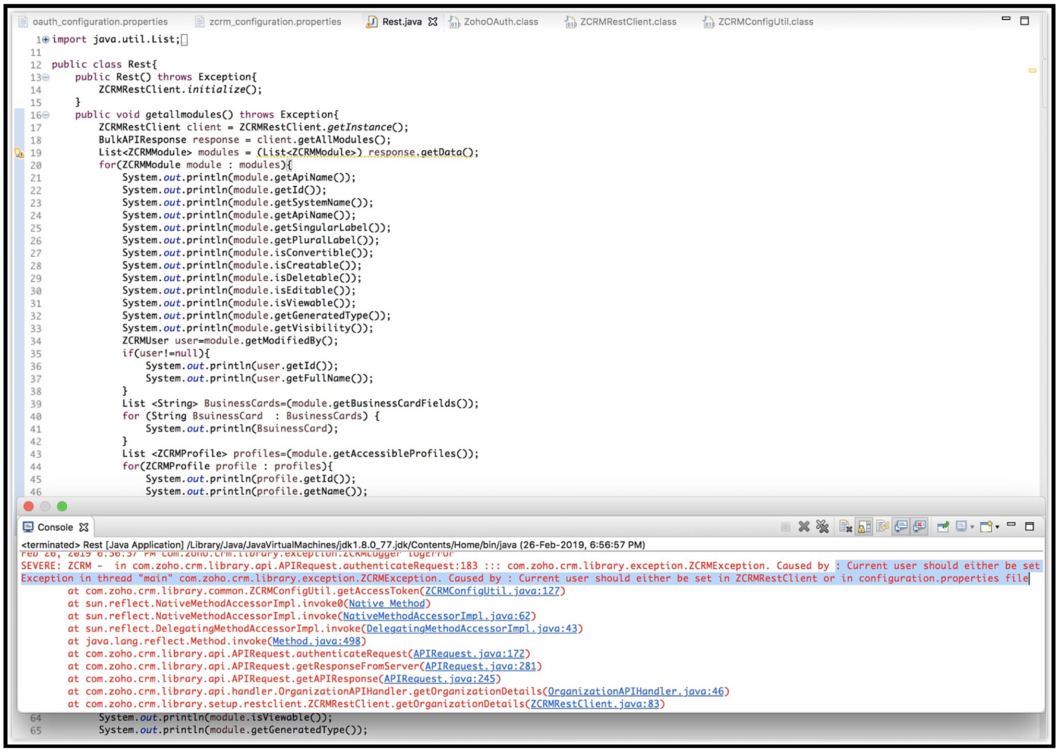
Solution

Provide the oauth_token_file_path in the respective file.



```
client_id=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
client_secret=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
redirect_uri=https://www.zoho.com
scope=
access_type=online
persistence_handler_class=com.zoho.oauth.clientapp.ZohoOAuthFilePersistence
mysql_username=root
mysql_password=
oauth_tokens_file_path=path/to/file/oauthtokens.properties
```

4. Current user must be either set in ZCRMRestClient or configuration.properties



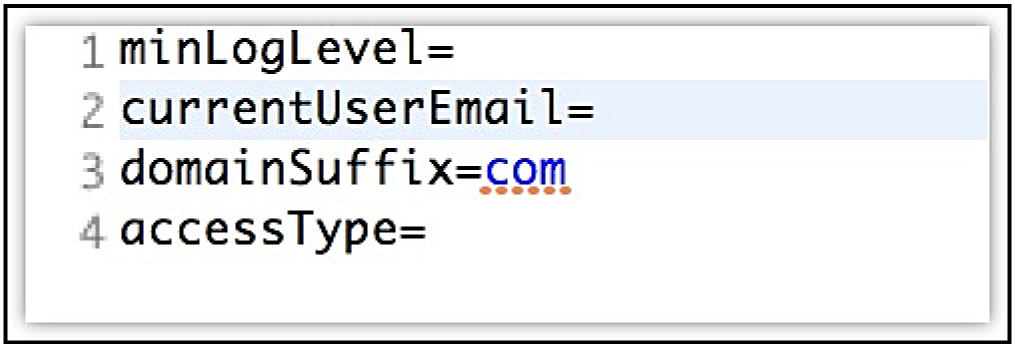
```java
import java.util.List;

public class Rest{
    public Rest() throws Exception{
        ZCRMRestClient.initialize();
    }
    public void getallmodules() throws Exception{
        ZCRMRestClient client = ZCRMRestClient.getInstance();
        BulkAPIResponse response = client.getAllModules();
        List<ZCRMModule> modules = (List<ZCRMModule>) response.getData();
        for(ZCRMModule module : modules){
            System.out.println(module.getApiName());
            System.out.println(module.getId());
            System.out.println(module.getSystemName());
            System.out.println(module.getApiName());
            System.out.println(module.getSingularLabel());
            System.out.println(module.getPluralLabel());
            System.out.println(module.isConvertible());
            System.out.println(module.isCreatable());
            System.out.println(module.isDeletable());
            System.out.println(module.isEditable());
            System.out.println(module.isViewable());
            System.out.println(module.getGeneratedType());
            System.out.println(module.getVisibility());
            ZCRMUser user=module.getModifiedBy();
            if(user!=null){
                System.out.println(user.getId());
                System.out.println(user.getFullName());
            }
            List <String> BusinessCards=(module.getBusinessCardFields());
            for (String BsuinessCard  : BusinessCards) {
                System.out.println(BsuinessCard);
            }
            List <ZCRMProfile> profiles=(module.getAccessibleProfiles());
            for(ZCRMProfile profile : profiles){
                System.out.println(profile.getId());
                System.out.println(profile.getName());
```

```
<terminated> Rest [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_77.jdk/Contents/Home/bin/java (26-Feb-2019, 6:56:57 PM)
Feb 26, 2019 6:56:57 PM com.zoho.crm.library.exception.ZCRMLogger logError
SEVERE: ZCRM -  in com.zoho.crm.library.api.APIRequest.authenticateRequest:183 ::: com.zoho.crm.library.exception.ZCRMException. Caused by : Current user should either be set
Exception in thread "main" com.zoho.crm.library.exception.ZCRMException. Caused by : Current user should either be set in ZCRMRestClient or in configuration.properties file
	at com.zoho.crm.library.common.ZCRMConfigUtil.getAccessToken(ZCRMConfigUtil.java:127)
	at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
	at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
	at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
	at java.lang.reflect.Method.invoke(Method.java:498)
	at com.zoho.crm.library.api.APIRequest.authenticateRequest(APIRequest.java:172)
	at com.zoho.crm.library.api.APIRequest.getResponseFromServer(APIRequest.java:281)
	at com.zoho.crm.library.api.APIRequest.getAPIResponse(APIRequest.java:245)
	at com.zoho.crm.library.api.handler.OrganizationAPIHandler.getOrganizationDetails(OrganizationAPIHandler.java:46)
	at com.zoho.crm.library.setup.restclient.ZCRMRestClient.getOrganizationDetails(ZCRMRestClient.java:83)
```

```java
        System.out.println(module.isViewable());
        System.out.println(module.getGeneratedType());
```

Reason

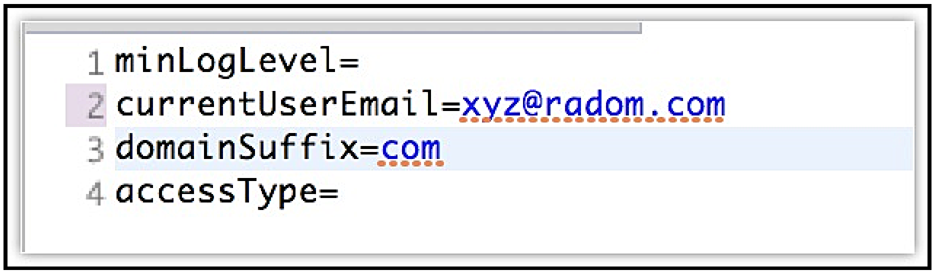
The currentUserEmail must have been left empty.



```
minLogLevel=
currentUserEmail=
domainSuffix=com
accessType=
```

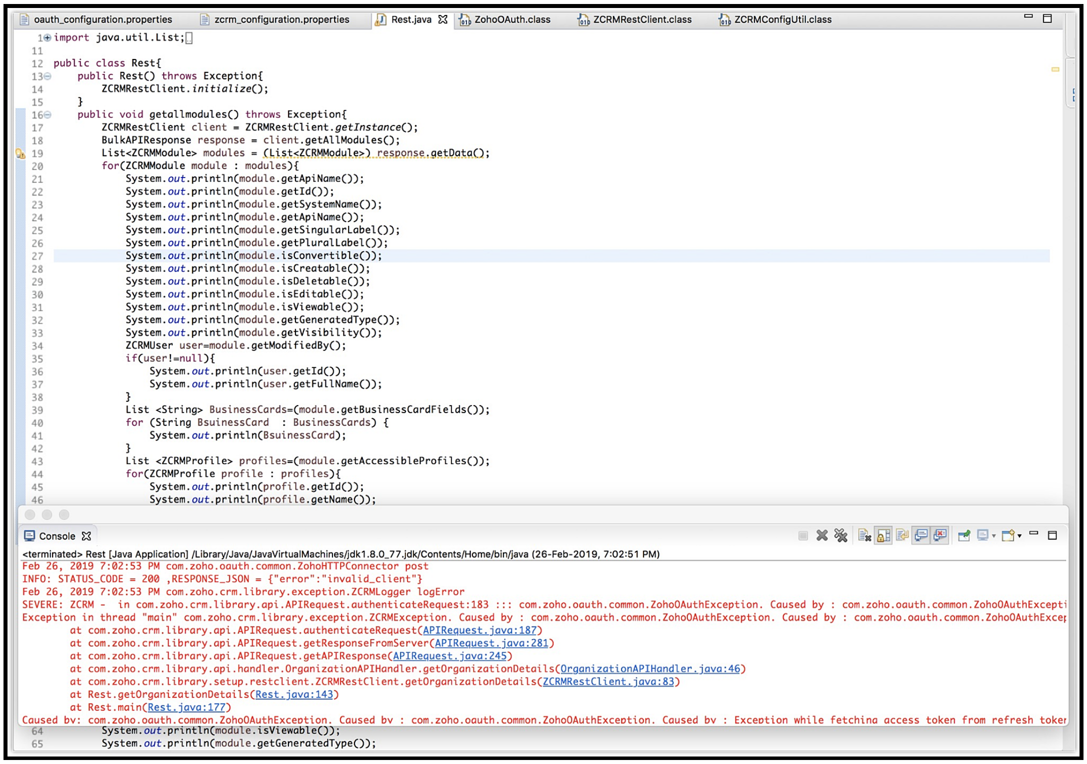
Solution

Specify the currentUserEmail.



```
minLogLevel=
currentUserEmail=xyz@radom.com
domainSuffix=com
accessType=
```

5. Invalid Client



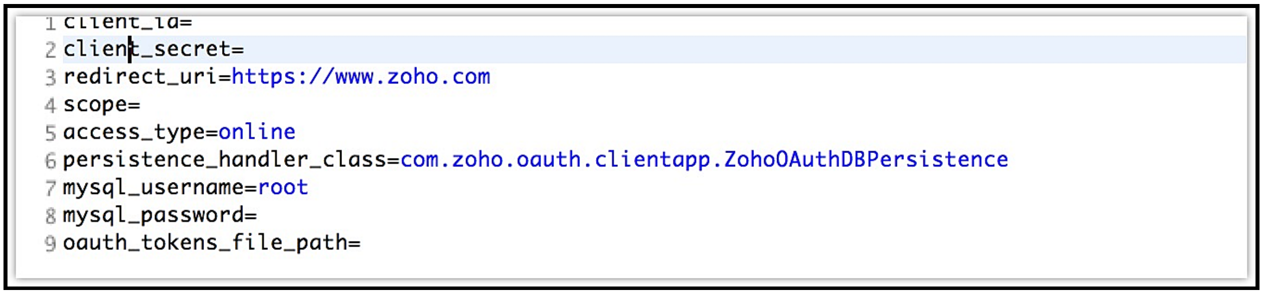
```java
import java.util.List;

public class Rest{
    public Rest() throws Exception{
        ZCRMRestClient.initialize();
    }
    public void getallmodules() throws Exception{
        ZCRMRestClient client = ZCRMRestClient.getInstance();
        BulkAPIResponse response = client.getAllModules();
        List<ZCRMModule> modules = (List<ZCRMModule>) response.getData();
        for(ZCRMModule module : modules){
            System.out.println(module.getApiName());
            System.out.println(module.getId());
            System.out.println(module.getSystemName());
            System.out.println(module.getApiName());
            System.out.println(module.getSingularLabel());
            System.out.println(module.getPluralLabel());
            System.out.println(module.isConvertible());
            System.out.println(module.isCreatable());
            System.out.println(module.isDeletable());
            System.out.println(module.isEditable());
            System.out.println(module.isViewable());
            System.out.println(module.getGeneratedType());
            System.out.println(module.getVisibility());
            ZCRMUser user=module.getModifiedBy();
            if(user!=null){
                System.out.println(user.getId());
                System.out.println(user.getFullName());
            }
            List <String> BusinessCards=(module.getBusinessCardFields());
            for (String BsuinessCard  : BusinessCards) {
                System.out.println(BsuinessCard);
            }
            List <ZCRMProfile> profiles=(module.getAccessibleProfiles());
            for(ZCRMProfile profile : profiles){
                System.out.println(profile.getId());
                System.out.println(profile.getName());
```

```
<terminated> Rest [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_77.jdk/Contents/Home/bin/java (26-Feb-2019, 7:02:51 PM)
Feb 26, 2019 7:02:53 PM com.zoho.oauth.common.ZohoHTTPConnector post
INFO: STATUS_CODE = 200 ,RESPONSE_JSON = {"error":"invalid_client"}
Feb 26, 2019 7:02:53 PM com.zoho.crm.library.exception.ZCRMLogger logError
SEVERE: ZCRM -  in com.zoho.crm.library.api.APIRequest.authenticateRequest:183 ::: com.zoho.oauth.common.ZohoOAuthException. Caused by : com.zoho.oauth.common.ZohoOAuthExcepti
Exception in thread "main" com.zoho.crm.library.exception.ZCRMException. Caused by : com.zoho.oauth.common.ZohoOAuthException. Caused by : com.zoho.oauth.common.ZohoOAuthExcep
	at com.zoho.crm.library.api.APIRequest.authenticateRequest(APIRequest.java:187)
	at com.zoho.crm.library.api.APIRequest.getResponseFromServer(APIRequest.java:281)
	at com.zoho.crm.library.api.APIRequest.getAPIResponse(APIRequest.java:245)
	at com.zoho.crm.library.api.handler.OrganizationAPIHandler.getOrganizationDetails(OrganizationAPIHandler.java:46)
	at com.zoho.crm.library.setup.restclient.ZCRMRestClient.getOrganizationDetails(ZCRMRestClient.java:83)
	at Rest.getOrganizationDetails(Rest.java:143)
	at Rest.main(Rest.java:177)
Caused by: com.zoho.oauth.common.ZohoOAuthException. Caused by : com.zoho.oauth.common.ZohoOAuthException. Caused by : Exception while fetching access token from refresh toker
```

```java
        System.out.println(module.isViewable());
        System.out.println(module.getGeneratedType());
```

Reason

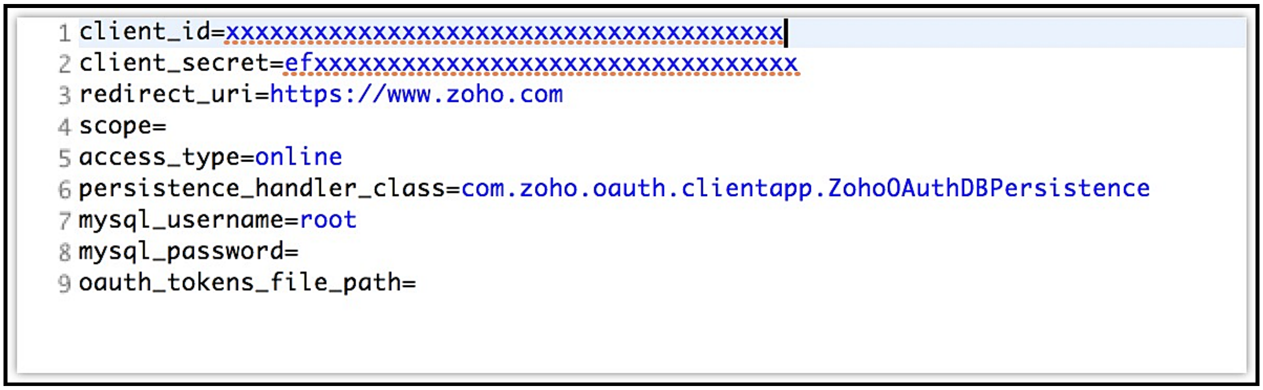
The client_id and the client_secret keys must have been left empty.



```
client_id=
client_secret=
redirect_uri=https://www.zoho.com
scope=
access_type=online
persistence_handler_class=com.zoho.oauth.clientapp.ZohoOAuthDBPersistence
mysql_username=root
mysql_password=
oauth_tokens_file_path=
```

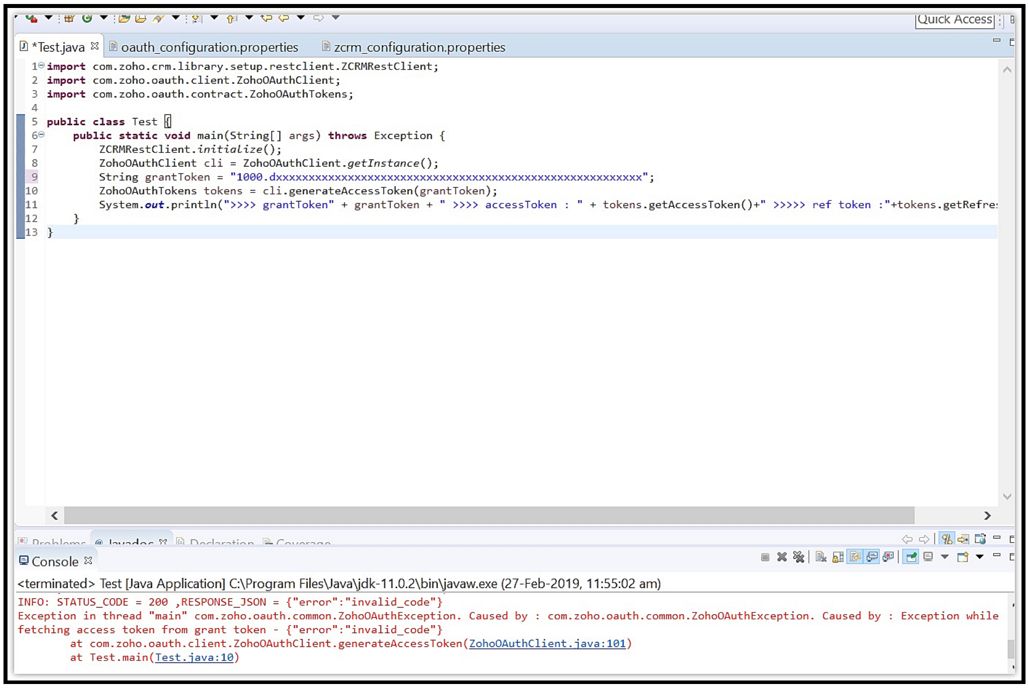
Solution

Provide the two keys in the oauth_configuration.properties file.



```
client_id=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
client_secret=efxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
redirect_uri=https://www.zoho.com
scope=
access_type=online
persistence_handler_class=com.zoho.oauth.clientapp.ZohoOAuthDBPersistence
mysql_username=root
mysql_password=
oauth_tokens_file_path=
```

6. Exception while fetching access tokens from grant token



```java
import com.zoho.crm.library.setup.restclient.ZCRMRestClient;
import com.zoho.oauth.client.ZohoOAuthClient;
import com.zoho.oauth.contract.ZohoOAuthTokens;

public class Test {
    public static void main(String[] args) throws Exception {
        ZCRMRestClient.initialize();
        ZohoOAuthClient cli = ZohoOAuthClient.getInstance();
        String grantToken = "1000.dxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx";
        ZohoOAuthTokens tokens = cli.generateAccessToken(grantToken);
        System.out.println(">>>> grantToken" + grantToken + " >>>> accessToken : " + tokens.getAccessToken()+" >>>>> ref token :"+tokens.getRefre
    }
}
```

```
<terminated> Test [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (27-Feb-2019, 11:55:02 am)
INFO: STATUS_CODE = 200 ,RESPONSE_JSON = {"error":"invalid_code"}
Exception in thread "main" com.zoho.oauth.common.ZohoOAuthException. Caused by : com.zoho.oauth.common.ZohoOAuthException. Caused by : Exception while
fetching access token from grant token - {"error":"invalid_code"}
        at com.zoho.oauth.client.ZohoOAuthClient.generateAccessToken(ZohoOAuthClient.java:101)
        at Test.main(Test.java:10)
```

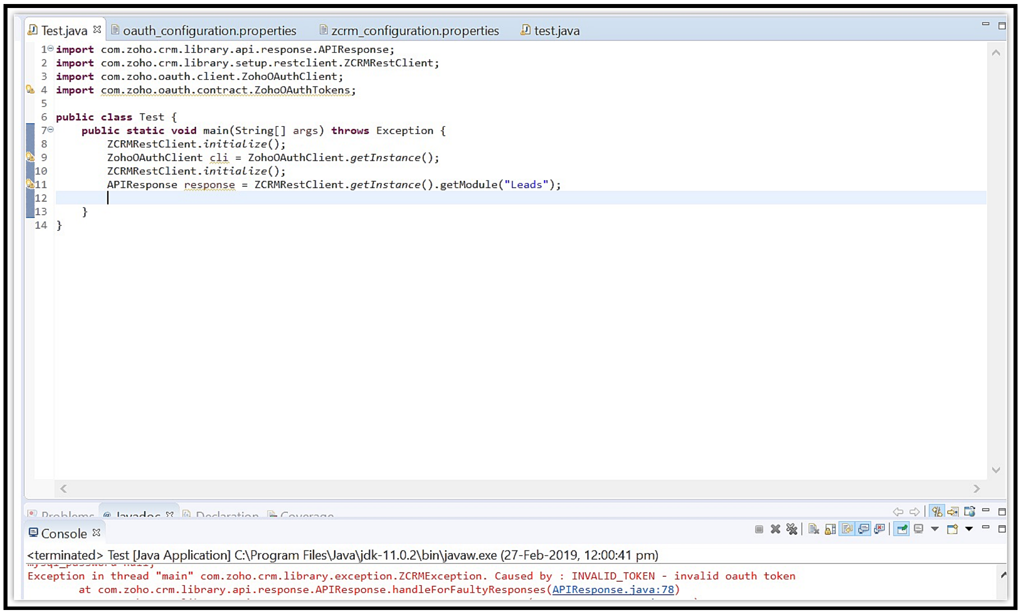
Reason

The grant token has expired.

Solution

Regenerate the grant token use it to generate new access token within the stipulated time.

7. Invalid OAuth token

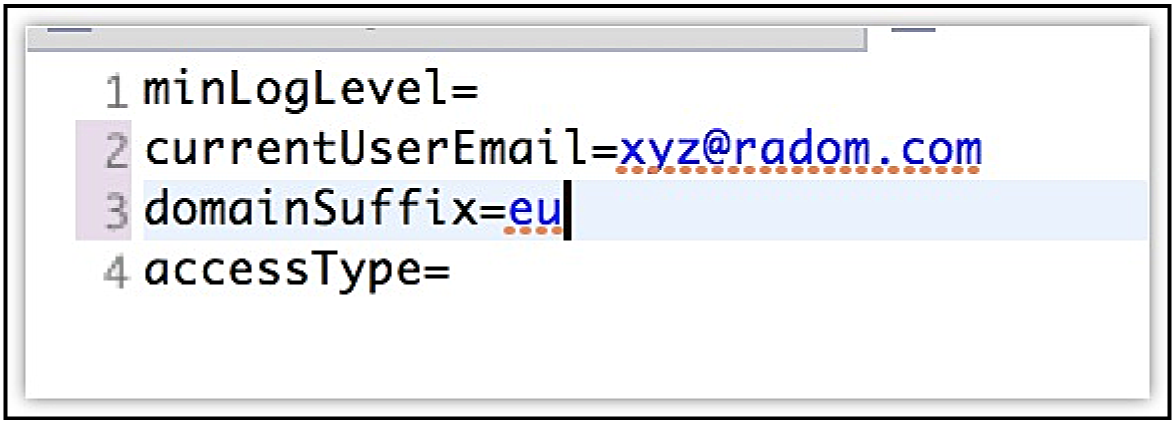


```java
import com.zoho.crm.library.api.response.APIResponse;
import com.zoho.crm.library.setup.restclient.ZCRMRestClient;
import com.zoho.oauth.client.ZohoOAuthClient;
import com.zoho.oauth.contract.ZohoOAuthTokens;

public class Test {
    public static void main(String[] args) throws Exception {
        ZCRMRestClient.initialize();
        ZohoOAuthClient cli = ZohoOAuthClient.getInstance();
        ZCRMRestClient.initialize();
        APIResponse response = ZCRMRestClient.getInstance().getModule("Leads");
        
    }
}
```

```
<terminated> Test [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (27-Feb-2019, 12:00:41 pm)
Exception in thread "main" com.zoho.crm.library.exception.ZCRMException. Caused by : INVALID_TOKEN - invalid oauth token
	at com.zoho.crm.library.api.response.APIResponse.handleForFaultyResponses(APIResponse.java:78)
```

Reason

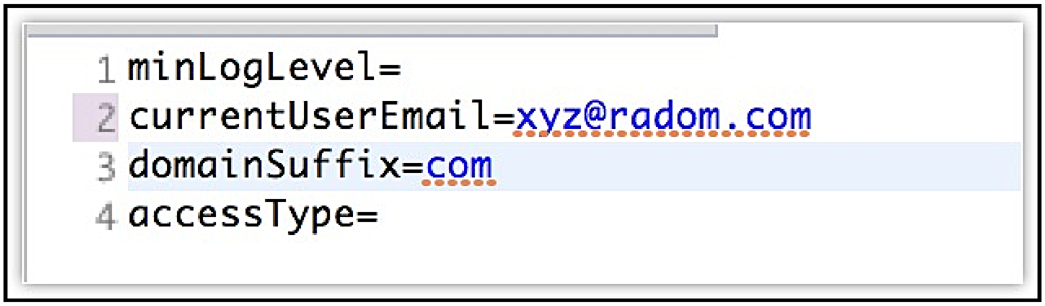
The user has generated access token and is trying to access module record from different domains. For example, the user generates the access token from accounts.zoho.com and tries to fetch module record data from accounts.zoho.eu.



```
minLogLevel=
currentUserEmail=xyz@radom.com
domainSuffix=eu
accessType=
```

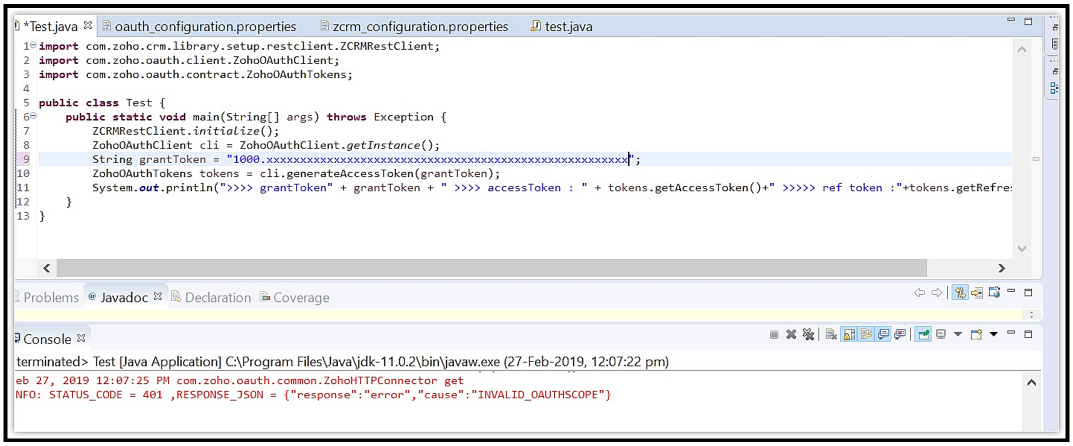
Solution

Use the same domain



```
minLogLevel=
currentUserEmail=xyz@radom.com
domainSuffix=com
accessType=
```

8. Invalid OAuth scope



```
import com.zoho.crm.library.setup.restclient.ZCRMRestClient;
import com.zoho.oauth.client.ZohoOAuthClient;
import com.zoho.oauth.contract.ZohoOAuthTokens;

public class Test {
    public static void main(String[] args) throws Exception {
        ZCRMRestClient.initialize();
        ZohoOAuthClient cli = ZohoOAuthClient.getInstance();
        String grantToken = "1000.xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx";
        ZohoOAuthTokens tokens = cli.generateAccessToken(grantToken);
        System.out.println(">>>> grantToken" + grantToken + " >>>> accessToken : " + tokens.getAccessToken()+" >>>>> ref token :"+tokens.getRefre
    }
}
```

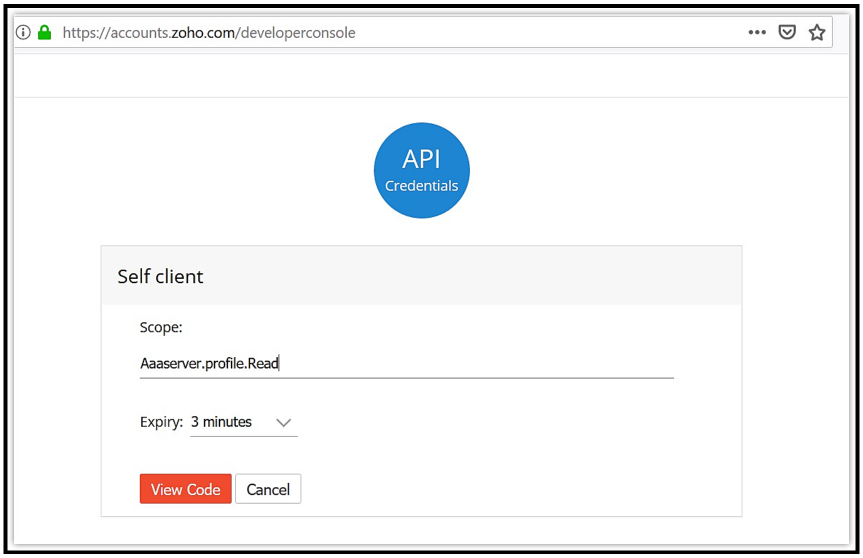
```
terminated> Test [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (27-Feb-2019, 12:07:22 pm)
eb 27, 2019 12:07:25 PM com.zoho.oauth.common.ZohoHTTPConnector get
NFO: STATUS_CODE = 401 ,RESPONSE_JSON = {"response":"error","cause":"INVALID_OAUTHSCOPE"}
```

Reason

The user must have not specified the Aaaserver.profile.read scope while generating the grant token.

Solution

Include Aaaserver.profile.read in the scope and generate a new grant token.



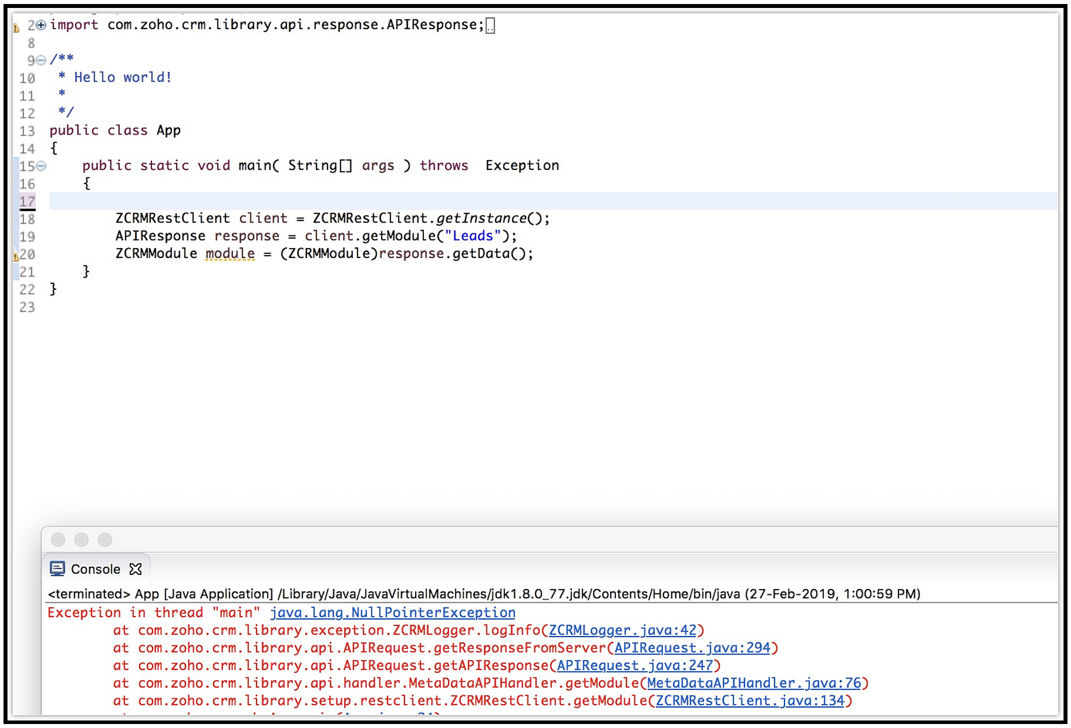
Self client

Scope:

Aaaserver.profile.Read

Expiry: 3 minutes

9. Exception in main:Java.Lang.NullPointerException



```
import com.zoho.crm.library.api.response.APIResponse;

/**
 * Hello world!
 *
 */
public class App
{
    public static void main( String[] args ) throws  Exception
    {

        ZCRMRestClient client = ZCRMRestClient.getInstance();
        APIResponse response = client.getModule("Leads");
        ZCRMModule module = (ZCRMModule)response.getData();
    }
}
```

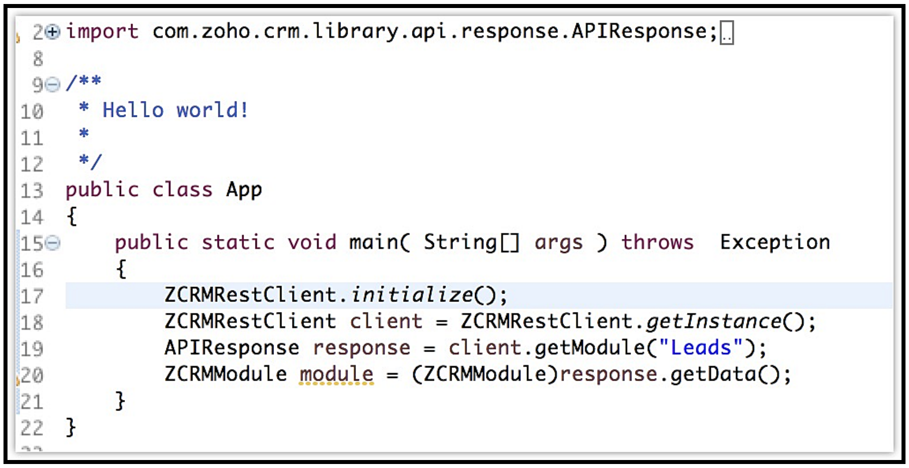
```
<terminated> App [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_77.jdk/Contents/Home/bin/java (27-Feb-2019, 1:00:59 PM)
Exception in thread "main" java.lang.NullPointerException
	at com.zoho.crm.library.exception.ZCRMLogger.logInfo(ZCRMLogger.java:42)
	at com.zoho.crm.library.api.APIRequest.getResponseFromServer(APIRequest.java:294)
	at com.zoho.crm.library.api.APIRequest.getAPIResponse(APIRequest.java:247)
	at com.zoho.crm.library.api.handler.MetaDataAPIHandler.getModule(MetaDataAPIHandler.java:76)
	at com.zoho.crm.library.setup.restclient.ZCRMRestClient.getModule(ZCRMRestClient.java:134)
```

Reason

The ZCRMRestClient must have not been initialized in the app.

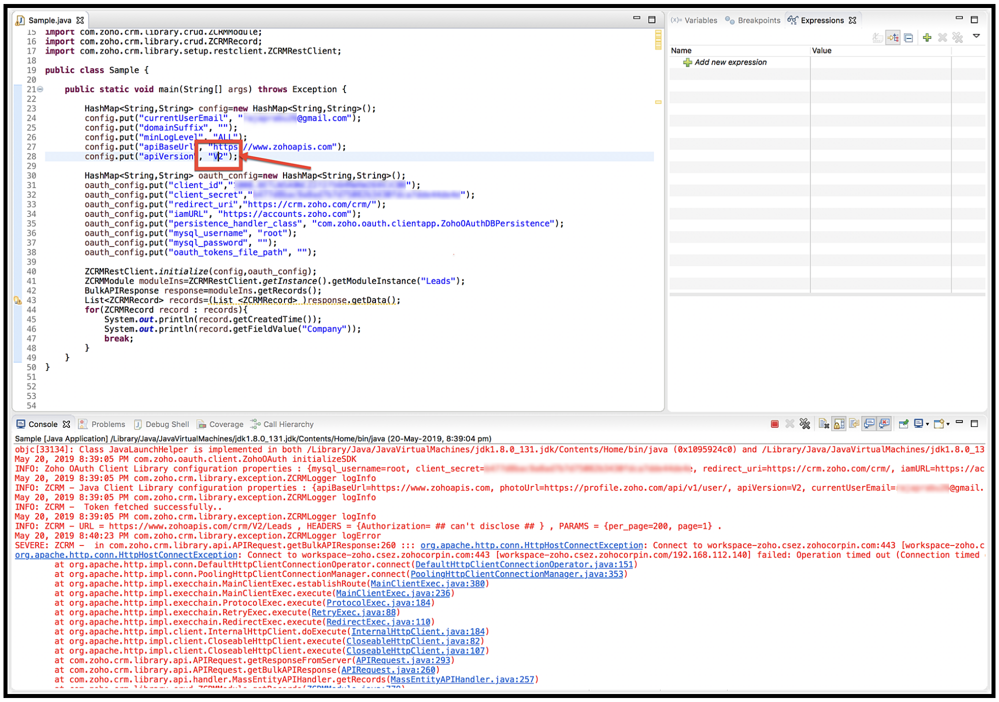
Solution

Include ZCRMRestClient.initialize() in the code.



```
import com.zoho.crm.library.api.response.APIResponse;

/**
 * Hello world!
 *
 */
public class App
{
    public static void main( String[] args ) throws  Exception
    {
        ZCRMRestClient.initialize();
        ZCRMRestClient client = ZCRMRestClient.getInstance();
        APIResponse response = client.getModule("Leads");
        ZCRMModule module = (ZCRMModule)response.getData();
    }
}
```

10. Error: Connect to workspace-zoho.csez.zohocorpin.com:443



```
import com.zoho.crm.library.crud.ZCRMModule;
import com.zoho.crm.library.crud.ZCRMRecord;
import com.zoho.crm.library.setup.restclient.ZCRMRestClient;

public class Sample {

    public static void main(String[] args) throws Exception {

        HashMap<String,String> config=new HashMap<String,String>();
        config.put("currentUserEmail", "[redacted]@gmail.com");
        config.put("domainSuffix", "");
        config.put("minLogLevel", "ALL");
        config.put("apiBaseUrl", "https://www.zohoapis.com");
        config.put("apiVersion", "V2");

        HashMap<String,String> oauth_config=new HashMap<String,String>();
        oauth_config.put("client_id","[redacted]");
        oauth_config.put("client_secret","[redacted]");
        oauth_config.put("redirect_uri","https://crm.zoho.com/crm/");
        oauth_config.put("iamURL", "https://accounts.zoho.com");
        oauth_config.put("persistence_handler_class", "com.zoho.oauth.clientapp.ZohoOAuthDBPersistence");
        oauth_config.put("mysql_username", "root");
        oauth_config.put("mysql_password", "");
        oauth_config.put("oauth_tokens_file_path", "");

        ZCRMRestClient.initialize(config,oauth_config);
        ZCRMModule moduleIns=ZCRMRestClient.getInstance().getModuleInstance("Leads");
        BulkAPIResponse response=moduleIns.getRecords();
        List<ZCRMRecord> records=(List <ZCRMRecord> )response.getData();
        for(ZCRMRecord record : records){
            System.out.println(record.getCreatedTime());
            System.out.println(record.getFieldValue("Company"));
            break;
        }
    }
}
```

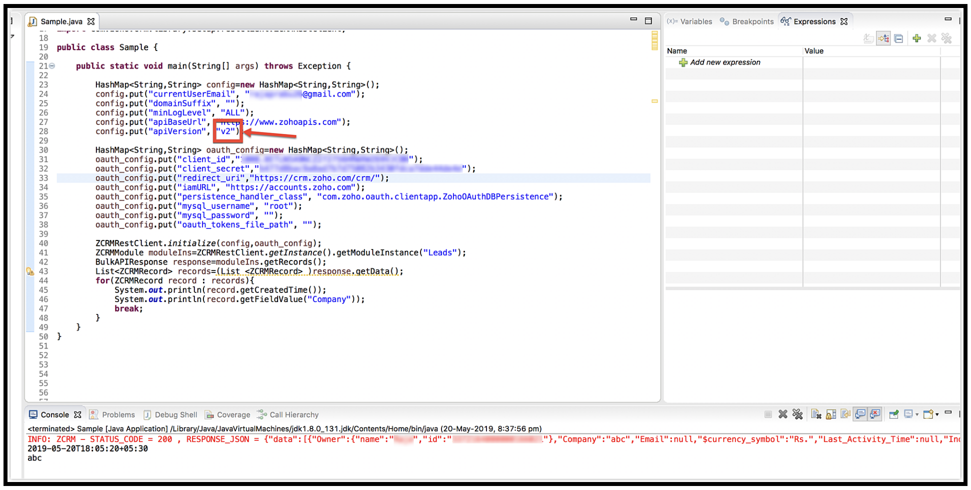
```
Sample [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_131.jdk/Contents/Home/bin/java (20-May-2019, 8:39:04 pm)
objc[33134]: Class JavaLaunchHelper is implemented in both /Library/Java/JavaVirtualMachines/jdk1.8.0_131.jdk/Contents/Home/bin/java (0x1095924c0) and /Library/Java/JavaVirtualMachines/jdk1.8.0_13
May 20, 2019 8:39:05 PM com.zoho.oauth.client.ZohoOAuth initializeSDK
INFO: Zoho OAuth Client Library configuration properties : {mysql_username=root, client_secret=[redacted], redirect_uri=https://crm.zoho.com/crm/, iamURL=https://ac
May 20, 2019 8:39:05 PM com.zoho.crm.library.exception.ZCRMLogger logInfo
INFO: ZCRM - Java Client Library configuration properties : {apiBaseUrl=https://www.zohoapis.com, photoUrl=https://profile.zoho.com/api/v1/user/, apiVersion=V2, currentUserEmail=[redacted]@gmail.
May 20, 2019 8:39:05 PM com.zoho.crm.library.exception.ZCRMLogger logInfo
INFO: ZCRM -  Token fetched successfully..
May 20, 2019 8:39:05 PM com.zoho.crm.library.exception.ZCRMLogger logInfo
INFO: ZCRM - URL = https://www.zohoapis.com/crm/V2/Leads , HEADERS = {Authorization= ## can't disclose ## } , PARAMS = {per_page=200, page=1} .
May 20, 2019 8:40:23 PM com.zoho.crm.library.exception.ZCRMLogger logError
SEVERE: ZCRM -  in com.zoho.crm.library.api.APIRequest.getBulkAPIResponse:260 ::: org.apache.http.conn.HttpHostConnectException: Connect to workspace-zoho.csez.zohocorpin.com:443 [workspace-zoho.c
org.apache.http.conn.HttpHostConnectException: Connect to workspace-zoho.csez.zohocorpin.com:443 [workspace-zoho.csez.zohocorpin.com/192.168.112.140] failed: Operation timed out (Connection timed
        at org.apache.http.impl.conn.DefaultHttpClientConnectionOperator.connect(DefaultHttpClientConnectionOperator.java:151)
        at org.apache.http.impl.conn.PoolingHttpClientConnectionManager.connect(PoolingHttpClientConnectionManager.java:353)
        at org.apache.http.impl.execchain.MainClientExec.establishRoute(MainClientExec.java:380)
        at org.apache.http.impl.execchain.MainClientExec.execute(MainClientExec.java:236)
        at org.apache.http.impl.execchain.ProtocolExec.execute(ProtocolExec.java:184)
        at org.apache.http.impl.execchain.RetryExec.execute(RetryExec.java:88)
        at org.apache.http.impl.execchain.RedirectExec.execute(RedirectExec.java:110)
        at org.apache.http.impl.client.InternalHttpClient.doExecute(InternalHttpClient.java:184)
        at org.apache.http.impl.client.CloseableHttpClient.execute(CloseableHttpClient.java:82)
        at org.apache.http.impl.client.CloseableHttpClient.execute(CloseableHttpClient.java:107)
        at com.zoho.crm.library.api.APIRequest.getResponseFromServer(APIRequest.java:293)
        at com.zoho.crm.library.api.APIRequest.getBulkAPIResponse(APIRequest.java:260)
        at com.zoho.crm.library.api.handler.MassEntityAPIHandler.getRecords(MassEntityAPIHandler.java:257)
```

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.



```java
public class Sample {

    public static void main(String[] args) throws Exception {

        HashMap<String,String> config=new HashMap<String,String>();
        config.put("currentUserEmail", "[redacted]@gmail.com");
        config.put("domainSuffix", "");
        config.put("minLogLevel", "ALL");
        config.put("apiBaseUrl", "https://www.zohoapis.com");
        config.put("apiVersion", "v2")

        HashMap<String,String> oauth_config=new HashMap<String,String>();
        oauth_config.put("client_id","[redacted]");
        oauth_config.put("client_secret","[redacted]");
        oauth_config.put("redirect_uri","https://crm.zoho.com/crm/");
        oauth_config.put("iamURL", "https://accounts.zoho.com");
        oauth_config.put("persistence_handler_class", "com.zoho.oauth.clientapp.ZohoOAuthDBPersistence");
        oauth_config.put("mysql_username", "root");
        oauth_config.put("mysql_password", "");
        oauth_config.put("oauth_tokens_file_path", "");

        ZCRMRestClient.initialize(config,oauth_config);
        ZCRMModule moduleIns=ZCRMRestClient.getInstance().getModuleInstance("Leads");
        BulkAPIResponse response=moduleIns.getRecords();
        List<ZCRMRecord> records=(List <ZCRMRecord> )response.getData();
        for(ZCRMRecord record : records){
            System.out.println(record.getCreatedTime());
            System.out.println(record.getFieldValue("Company"));
            break;
        }
    }
}
```

```
<terminated> Sample [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_131.jdk/Contents/Home/bin/java (20-May-2019, 8:37:56 pm)
INFO: ZCRM - STATUS_CODE = 200 , RESPONSE_JSON = {"data":[{"Owner":{"name":"[redacted]","id":"[redacted]"},"Company":"abc","Email":null,"$currency_symbol":"Rs.","Last_Activity_Time":null,"Ind
2019-05-20T18:05:20+05:30
abc
```

Release Notes

Java SDK for Zoho CRM was introduced on February 28, 2018. Since then, there have been many updates, bug fixes, patches, on a regular basis, to provide better solutions to people.

```
oho.crm:java-sdk:2.1.3'
}
```

- Notes:
  – From this version, the response will contain the value "null" instead of an empty string when you do not input values for the optional fields in the Events API.

Previous Versions

1. ZCRMSDK - VERSION 2.1.2

- Install command - Maven:(in pom.xml)

```
<repositories>
<repository>
<url>https://maven.zohodl.com</url>
</repository>
</repositories>
<dependencies>

<dependency>
<groupId>com.zoho.crm</groupId>
<artifactId>java-sdk</artifactId>
<version>2.1.2</version>
</dependency>
</dependencies>
```

- Install command - Gradle

```
repositories{
maven { url "https://maven.zohodl.com" }
}
dependencies{
implementation 'com.zoho.crm:java-sdk:2.1.2'
}
```

- Notes:

-- Fixed the issue of rendering improper response while fetching the related records of a module.

2. ZCRMSDK - VERSION 2.1.1

- Install command - Maven:(in pom.xml)

```
<repositories>
<repository>
<url>https://maven.zohodl.com</url>
</repository>
```

```
</repositories>

<dependencies>
<dependency>
<groupId>com.zoho.crm</groupId>
<artifactId>java-sdk</artifactId>
<version>2.1.1</version>
</dependency>
</dependencies>
```

- Install command - Gradle

```
repositories{
maven { url "https://maven.zohodl.com" }
}
dependencies{
implementation 'com.zoho.crm:java-sdk:2.1.1'
}
```

3. ZCRMSDK - VERSION 2.1.0

- Install command - Maven:(in pom.xml)

```
<repositories>
<repository>
<url>https://maven.zohodl.com</url>
</repository>
</repositories>

<dependencies>
<dependency>
<groupId>com.zoho.crm</groupId>
<artifactId>java-sdk</artifactId>
<version>2.1.0</version>
</dependency>
</dependencies>
```

- Install command - Gradle

```
repositories{
maven { url "https://maven.zohodl.com" }
}
dependencies{
implementation 'com.zoho.crm:java-sdk:2.1.0'
}
```

- Notes:

  -- This version supports the "process" key in the request body for record operations.

4. ZCRMSDK - VERSION 2.0.6

- Install command - Maven:(in pom.xml)

```
<repositories>
<repository>
<url>https://maven.zohodl.com</url>
</repository>
</repositories>

<dependencies>
<dependency>
<groupId>com.zoho.crm</groupId>
<artifactId>java-sdk</artifactId>
<version>2.0.6</version>
</dependency>
</dependencies>
```

- Install command - Gradle

```
repositories{
maven { url "https://maven.zohodl.com" }
}
dependencies{
implementation 'com.zoho.crm:java-sdk:2.0.6'
}
```

- Notes:

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

5. ZCRMSDK - VERSION 2.0.5

- Install command - Maven:(in pom.xml)

```
<repositories>
<repository>
<url>https://maven.zohodl.com</url>
</repository>
</repositories>

<dependencies>
<dependency>
<groupId>com.zoho.crm</groupId>
<artifactId>java-sdk</artifactId>
<version>2.0.5</version>
</dependency>
</dependencies>
```

- Install command - Gradle

```
repositories{
maven { url "https://maven.zohodl.com" }
}
dependencies{
implementation 'com.zoho.crm:java-sdk:2.0.5'
}
```

- Notes:

  -- 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.

6. ZCRMSDK - VERSION 2.0.4

- Install command - Maven:(in pom.xml)

```
<repositories>
<repository>
<url>https://maven.zohodl.com</url>
</repository>
</repositories>

<dependencies>
<dependency>
<groupId>com.zoho.crm</groupId>
<artifactId>java-sdk</artifactId>
<version>2.0.4</version>
</dependency>
</dependencies>
```

- Install command - Gradle

```
repositories{
maven { url "https://maven.zohodl.com" }
}
dependencies{
implementation 'com.zoho.crm:java-sdk:2.0.4'
```

}

- Notes:

  -- Handled the module name check for the Product_Details, Pricing_Details, and

Participants keys in GET records.

7. ZCRMSDK - VERSION 2.0.3

- Install command - Maven:(in pom.xml)

```
<repositories>
<repository>
<url>https://maven.zohodl.com</url>
</repository>
</repositories>

<dependencies>
<dependency>
<groupId>com.zoho.crm</groupId>
<artifactId>java-sdk</artifactId>
<version>2.0.3</version>
</dependency>
</dependencies>
```

- Install command - Gradle

```
repositories{
maven { url "https://maven.zohodl.com" }
}
dependencies{
implementation 'com.zoho.crm:java-sdk:2.0.3'
}
```

- Notes:

  -- Handled null module in related list.

8. ZCRMSDK - VERSION 2.0.2

- Install command - Maven:(in pom.xml)

```
<repositories>
```

```
<repository>
<url>https://maven.zohodl.com</url>
</repository>
</repositories>

<dependencies>
<dependency>
<groupId>com.zoho.crm</groupId>
<artifactId>java-sdk</artifactId>
<version>2.0.2</version>
</dependency>
</dependencies>
```

- Install command - Gradle

```
repositories{
maven { url "https://maven.zohodl.com" }
}
dependencies{
implementation 'com.zoho.crm:java-sdk:2.0.2'
}
```

- Notes:

  -- Handled OAuth response change.

9. ZCRMSDK - VERSION 2.0.1

- Install command - Maven:(in pom.xml)

```
<repositories>
<repository>
<url>https://maven.zohodl.com</url>
</repository>
</repositories>

<dependencies>
```

```
<dependency>
<groupId>com.zoho.crm</groupId>
<artifactId>java-sdk</artifactId>
<version>2.0.1</version>
</dependency>
</dependencies>
```

- Install command - Gradle

```
repositories{
maven { url "https://maven.zohodl.com" }
}
dependencies{
implementation 'com.zoho.crm:java-sdk:2.0.1'
}
```

- Notes:
  – Variables and Variable group APIs are supported.
  – Fix in generate token via grant token.

10. ZCRMSDK - VERSION 2.0.0

- Install command - Maven:(in pom.xml)

```
<repositories>
<repository>
<url>https://maven.zohodl.com</url>
</repository>
</repositories>

<dependencies>
<dependency>
<groupId>com.zoho.crm</groupId>
<artifactId>java-sdk</artifactId>
<version>2.0.0</version>
</dependency>
```

```
</dependencies>
```

- Install command - Gradle

```
repositories{
maven { url "https://maven.zohodl.com" }
}
dependencies{
implementation 'com.zoho.crm:java-sdk:2.0.0'
}
```

- Notes:
  – Configuration details should be passed ONLY as a map (array). Configuration properties files are not supported.
  – Update records method provided.
  – Multiple clients are supported through this jar.
  – Access token can be generated from refresh token.
  – Expiry time increased from 10 to 15000 milliseconds(15 seconds).
  – "deleted_by" key value as NULL handled in getPermanentlyDeletedRecords() response.

11. ZCRMSDK - VERSION 1.0.6

- Install command - Maven:(in pom.xml)

```
<repositories>
<repository>
<url>https://maven.zohodl.com</url>
</repository>
</repositories>

<dependencies>
<dependency>
<groupId>com.zoho.crm</groupId>
<artifactId>java-sdk</artifactId>
<version>1.0.6</version>
```

```
</dependency>
</dependencies>
```

- Install command - Gradle

```
repositories{
maven { url "https://maven.zohodl.com" }
}
dependencies{
implementation 'com.zoho.crm:java-sdk:1.0.6'
}
```

- Notes:
  -- Tag APIs are supported in this version.
  -- JSONException (related to subforms) is fixed in the getRecord function.

12. ZCRMSDK - VERSION 1.0.5

- Install command - Maven:(in pom.xml)

```
<repositories>
<repository>
<url>https://maven.zohodl.com</url>
</repository>
</repositories>

<dependencies>
<dependency>
<groupId>com.zoho.crm</groupId>
<artifactId>java-sdk</artifactId>
<version>1.0.5</version>
</dependency>
</dependencies>
```

- Install command - Gradle

```
repositories{
maven { url "https://maven.zohodl.com" }
}
dependencies{
implementation 'com.zoho.crm:java-sdk:1.0.5'
}
```

- Notes:

-- Logs can be written in the file by specifying the full path name in configuration.properties file (logFilePath=path/to/file.log).

13. ZCRMSDK - VERSION 1.0.4

- Install command - Maven:(in pom.xml)

```
<repositories>
<repository>
<url>https://maven.zohodl.com</url>
</repository>
</repositories>

<dependencies>
<dependency>
<groupId>com.zoho.crm</groupId>
<artifactId>java-sdk</artifactId>
<version>1.0.4</version>
</dependency>
</dependencies>
```

- Install command - Gradle

```
repositories{
maven { url "https://maven.zohodl.com" }
}
dependencies{
implementation 'com.zoho.crm:java-sdk:1.0.4'
}
```

- Notes:

  -- Added lead assignment rule support.

  -- Duplicate check fields support while upserting the record.

14. ZCRMSDK - VERSION 1.0.3

- Install command - Maven:(in pom.xml)

```
<repositories>
<repository>
<url>https://maven.zohodl.com</url>
</repository>
</repositories>

<dependencies>
<dependency>
<groupId>com.zoho.crm</groupId>
<artifactId>java-sdk</artifactId>
<version>1.0.3</version>
</dependency>
</dependencies>
```

- Install command - Gradle

```
repositories{
maven { url "https://maven.zohodl.com" }
}
dependencies{
implementation 'com.zoho.crm:java-sdk:1.0.3'
}
```

- Notes:

-- Issue Fix.

15. ZCRMSDK - VERSION 1.0.2

- Install command - Maven:(in pom.xml)

```
<repositories>
<repository>
<url>https://maven.zohodl.com</url>
</repository>
</repositories>

<dependencies>
<dependency>
<groupId>com.zoho.crm</groupId>
<artifactId>java-sdk</artifactId>
<version>1.0.2</version>
</dependency>
</dependencies>
```

- Install command - Gradle

```
repositories{
maven { url "https://maven.zohodl.com" }
}
dependencies{
implementation 'com.zoho.crm:java-sdk:1.0.2'
}
```

- Notes:

  -- Issue Fix

16. ZCRMSDK - VERSION 1.0.0

- Install command - Maven:(in pom.xml)

```
<repositories>
<repository>
```

```xml
<url>https://maven.zohodl.com</url>
</repository>
</repositories>

<dependencies>
<dependency>
<groupId>com.zoho.crm</groupId>
<artifactId>java-sdk</artifactId>
<version>1.0.0</version>
</dependency>
</dependencies>
```

- Install command - Gradle

```groovy
repositories{
maven { url "https://maven.zohodl.com" }
}
dependencies{
implementation 'com.zoho.crm:java-sdk:1.0.0'
}
```

- Notes:

  -- Supported Subform APIs.

  -- Updated JSON jar.

  -- Supported page and perPage options for Get Users.

  -- Supported User CRUD APIs.