Rest API Services User Guide

Oracle FLEXCUBE Universal Banking

Release 14.4.0.0.0

May 2020

ORACLE

Rest API Services User Guide
Oracle Financial Services Software Limited

Oracle Park

Off Western Express Highway
Goregaon (East)
Mumbai, Maharashtra 400 063
India
Worldwide Inquiries:
Phone: +91 22 6718 3000
Fax: +91 22 6718 3001
www.oracle.com/financialservices/

Copyright © 2007, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are “commercial computer software” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate failsafe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

ORACLE

Table of Contents

1. PREFACE 1-1

1.1 INTRODUCTION 1-1
1.2 AUDIENCE 1-1
1.3 DOCUMENTATION ACCESSIBILITY 1-1
1.4 ORGANIZATION 1-1

2. REST API SERVICES - OVERVIEW 2-1

2.1 INTRODUCTION 2-1
2.1.1 Rest Services Security / Authentication Scheme 2-2
2.1.2 Rest HTTP Headers 2-3

3. REST API SERVICES - DETAILS 3-1

3.1 RETAIL CUSTOMER REST API SERVICES 3-1
3.1.1 Retail Customer Search Services 3-1
3.1.2 Retail Customer Image/Signature Search 3-3
3.2 RETAIL CUSTOMER ACCOUNT REST API SERVICES 3-4
3.2.1 Retail Customer Accounts 3-4
3.2.2 CASA Account Balance 3-6
3.2.3 TD Account Balance/Renewal Date 3-6
3.3 CORE REST API SERVICES 3-7
3.3.1 Standing Instructions for Accounts 3-8
3.3.2 Account Transaction Detail Services 3-10
3.3.3 Retail Customer Loan Services 3-10
3.3.4 Foreign Exchange Currency Rate Service 3-12
3.3.5 Branch Query Services 3-13
3.3.6 Amount Block Service 3-13
*3.3.7 Funds Transfer Service ..... **Error! Bookmark not defined.***
3.3.8 Single Journal Dr/Cr Service 3-14
3.3.9 Stop Cheque Service 3-16
3.3.10 Rest Services Additional Changes 3-17

1. Preface

1.1 Introduction

This User Guide is intended to familiarize you with the light-weight REST based services of Oracle FLEXCUBE. The manual gives you an overview of the services developed, input\request parameters and the response of the each REST service.

1.2 Audience

The manual is intended for integrating external systems to Oracle FLEXCUBE using RESTFUL framework.

1.3 Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

1.4 Organization

The manual is organized into the following chapters

Chapter	Description
Chapter 1	*Preface* - Gives information on the intended audience. It also lists the various chapters covered in this User Manual.
Chapter 2	*Rest API* - An Overview provides a snapshot of the features of the entire module.
Chapter 3	*Rest API Services* –Provides in-depth details of each of the rest APIs.

2. Rest API Services - Overview

2.1 Introduction

Oracle FLEXCUBE Universal Banking has adopted the industry standard of providing light-weight services through Representational State Transfer (popularly known as ReST) services. These lightweight services support the current trend of mobile/tablet based applications. These services help to improve the response time for tablet/mobile applications, as these applications are very chatty, involves numerous hits to the server and the expected response time is minimal for each of the requests.

The REST service requests can either use GET or POST methods.

The REST services using GET methods are listed below:

1. Fetch Customer details by Id, Email, national Id, phone number and also services to fetch the customer signature and image.
2. Fetch Customer Accounts, CASA accounts.
3. Fetch the TD account balance and the TD renewal date.
4. Fetch Standing instructions for an account.
5. Fetch the last 10 credit/debit/all transactions for accounts.
6. Fetch the loan accounts for a customer.
7. Fetch the loan account balance.
8. Fetch the loan instalment details.
9. Fetch the foreign currency exchange rate.
10. Fetch the list of Branches and branch address.

The following are the services using POST methods:

1. Create amount block.
2. Create a fund transfer.
3. Create a stop cheque.
4. Create a single journal debit/credit service.

More information on the above listed services can be found in the respective sections of the User manual.

Each of the services listed above will be a URL. These URLs can be tested using any of the available browser plug-ins (E.g. Postman, RestEasy etc.) Or these services can also be integrated with an API manager depending on its use. Below is the format of a Get customer service example which will take the customer number as the input

Http ://(IP) :(Port)/CustomerService/v12.3/customers/{customerNumber}

The input for the GET services is passed as a part of the URL and if more than one input needs to be passed they should be separated by a forward slash (/). Apart from passing the customer number as a part of the URL, the services also expect a few header parameters to be passed to the service. The following are the header parameters passed:

1. userId : Flexcube UBS login user Id
2. password : Encrypted Flexcube UBS password of the user
3. branch : The bank branch to which the user belongs
4. msgId : A alphanumeric input from the consumer
5. source : Source of the request whether it is FCUBS or an external system
6. Content-Type: This mentions the format of the input /output. Since we are using the JSON format we will give the content-type as application/json.

UserId, password and msgId are for authenticating the user accessing the service. More information on authentication can be found under Section 2.

For the POST services we need to provide data input in JSON format. An example of JSON input is shown below. The output of REST services will also be in JSON format. Sample input/output for each service is given under the respective sections of the User manual.

```
{
   "account" : "207000124025",
   "amount" : "1",
   "branch" : "207",
   "effectiveDate" : " 2015-08-02"
}
```

The list of services will be discussed in detail in the following chapters.

2.1.1 Rest Services Security / Authentication Scheme

Security in Restful Services is implemented using the Oracle FLEXCUBE SMS Authentication system. All requests to the Restful services needs to contain the headers as below.

1. userId - Oracle FLEXCUBE UBS login user name.
2. password – Encrypted Oracle FLEXCUBE UBS login password.
3. msgId –This message id will be a unique id to identify the request.

The userId and password headers should contain the userId and password that is created in the Oracle FLEXCUBE application. The password field should be encrypted with the msgId as the key. The Rest API application will decrypt the password and validate the same against the Oracle FLEXCUBE SMS maintenance, and will service the request only if the validation succeeds.

To enable the authentication scheme in the FCLiteAPI.properties file of the Restful API application, the property AUTHENTICATION_SCHEME should be configured as FLEXCUBE. This can be enabled only if the Oracle FLEXCUBE UBS application is configured to use the native SMS and not any third-party authentication.

2.1.2 Rest HTTP Headers

The following are the headers that are expected in all the requests that are sent to the Rest API application.

1. userId - Oracle FLEXCUBE UBS login user name.
2. password – Encrypted Oracle FLEXCUBE UBS login password.
3. msgId –This message id will be a unique id to identify the request.
4. branch – The branch code from where the request is sent.
5. source – The source code of the external system.

3. Rest API Services - Details

3.1 Retail Customer Rest API Services

Following are the list of Retail Customer Rest API Services that are available.

SL No.	Service	Method	Sample URL
1	Retail Customer Search By Customer ID	GET	http://(IP):(Port)/CustomerService/v12.3/customers/{customerNumber}
2	Retail Customer Search By Email	GET	http://(IP):(Port)/CustomerService/v12.3/customers/email/{email}
3	Retail Customer Search By National ID	GET	http://(IP):(Port)/CustomerService/v12.3/customers/nationalId/{nationalId}
4	Retail Customer Search By Mobile Number	GET	http://(IP):(Port)/CustomerService/v12.3/customers/mobileNo/{mobileNo}
5	Retail Customer Image Search	GET	http://(IP):(Port)/CustomerService/v12.3/customers/image/{customerNo}
6	Retail Customer Signature Search	GET	http://(IP):(Port)/CustomerService/v12.3/customers/signature/{customerNo}
7	Retail Customer Search By First Name	GET	http://(IP):(Port)/CustomerService/v12.3/customers/customerName/{customerName}
8	Retail Customer Search By RM User Id	Get	http://(IP):(Port)/CustomerService/v12.3/customers/userId/{rmUserId}

3.1.1 Retail Customer Search Services

The customer GET services (ie search by customer id, email, national id, mobile number and first name) help us in retrieving the customer information by passing the required parameters according to the service used as input. The following are the expected functionality of the services.

- These services will query for the customers that are present in Oracle FLEXCUBE.

- The customers should be valid Individual customer that has record stat as Open and also authorized.
- On failure case i.e., if the customer searched doesn't present in the system then it will throw an error stating appropriate error message.
- On success case, one or many record(s) containing the customer information will be sent back in the response.
- In case the tanking functionality in Customer Screen is enabled, only the authorized modifications will be made available through the GET services. Any unauthorized modifications will not be available for View through the GET Services.
- The parameters of the services and their fetch details are given below:
  - Search By Customer ID - Parameter to be passed is the customer id and the services will respond with query results on exact fetch of the parameter that is passed.
  - Search By Email - Parameter to be passed is the email and the services will respond with query results on starts with the the parameter that is passed.
  - Search by National Id – Parameters to be passed is the National Id and the services will respond with query results on starts with the the parameter that is passed.
  - Search by Mobile Number – Parameters to be passed is the mobile number without the ISD number and the services will respond with query results on starts with the the parameter that is passed.
  - Search by First Name– Parameters to be passed is the First Name and the services will respond with query results on starts with the parameter that is passed.
  - Search By RM User ID - Parameter to be passed is the RM id and the services will respond with query results on exact fetch of the parameter that is passed.

Example: Given below is a sample response of a customer. In case of services which return more than one customer, a list of such JSON objects would be returned.


```
{
 "pinCode": "string",
 "nationality": "string",
 "kyc": "string",
 "customerNo": "string",
 "shortName": "string",
 "add1": "string",
 "add2": "string",
 "add3": "string",
 "lastName": "string",
 "sex": "string",
```


ORACLE

```
"fax": "string",
"mobileNO": "string",
"homeTelNo": "string",
"email": "string",
"custPref": "string",
"dob": "string",
"firstName": "string",
"telephone": "string",
"maritalStatus": "string",
"customerType": "string",
"uniqueIdName": "string",
"uniqueIdValue": "string",
"updateAt": "string",
"createdAt": "string",
"customerCategory": "string",
"branchCode": "string",
"add4": "string"
}
```


3.1.2 Retail Customer Image/Signature Search

These services (is customer image and signature) helps us in retrieving the images/signatures of customer by passing customer number as input. The following are the expected functionality of the service.

- This service will query for the images/signatures of customer that are present in Oracle FLEXCUBE.
- The customers should be valid Individual customer that has record stat as Open and also authorized.
- On failure case i.e., if the customer searched doesn't present in the system then it will throw an error stating "Invalid Customer Number"
- On success case, list of images/signatures of the customer will be sent back in the response.

ORACLE

- In case the tanking functionality in Customer Screen is enabled, only the authorized modifications will be made available through the GET services. Any unauthorized modifications will not be available for View through the GET Services.

```
{
  "branchCode": "string",
  "customerId": "string",
  "imageList": [
    "string"
  ]
}
```

3.2 Retail Customer Account Rest API Services

Following are the list of Retail Customer Rest API Services that are available.

SL No.	Service	Method	Sample URL
1	Retail Customer Accounts	GET	http://(IP):(Port)/AccountService/v12.3/accounts/casa/{customerId}
2	CASA Account Balance	GET	http://(IP):(Port)/AccountService/v12.3/accounts/casaBalance/{accountNumber}
3	TD Account Balance	GET	http://(IP):(Port)/AccountService/v12.3/accounts/tdBalance/{accountNumber}
4	TD Renewal Date	GET	http://(IP):(Port)/AccountService/v12.3/accounts/tdRenewelDate/{accountNumber}

3.2.1 Retail Customer Accounts

This service helps us in retrieving the retail customer accounts information by passing customer id as input. The following are the expected functionality of the service.

- This service will query for the customer accounts that are present in Oracle FLEXCUBE.
- The customers should be valid Individual customer that has record stat as Open and authorized.
- On failure case i.e., if the customer searched doesn't present in the system then it will throw an error stating "Invalid Customer"
- On success case, list of accounts of that customer will be sent back in the response.

- In case the tanking functionality in Account Screen is enabled, only the authorized modifications will be made available through the GET services. Any unauthorized modifications will not be available for View through the GET Services.

Example: Sample JSON response


```
{
 "accounts": {
  "custNo": "string",
  "custAcNo": "string",
  "availableBalance": "string",
  "ccy": "string",
  "acStatCrOvd": "string",
  "acStatDrOvd": "string",
  "minReqdBal": "string",
  "address1": "string",
  "address2": "string",
  "address3": "string",
  "address4": "string",
  "ibanAcNo": "string",
  "updatedAt": "string",
  "accStatus": "string",
  "branchCode": "string",
  "acOpenDate": "string",
  "accountType": "string",
  "acStmtCycle": "string",
  "acDesc": "string",
  "cardDetails": {
```

```
            "cardNo": "string"
          }
        }
}
```


3.2.2 CASA Account Balance

This service helps us in retrieving the retail customer account balance by passing account number as input. The following are the expected functionality of the service.

- This service will query for the retail customer account balance that are present in Oracle FLEXCUBE.
- The account should be valid ,open and also authorized
- On failure case i.e., if the account searched doesn't present in the system then it will throw an error stating "Invalid Account"
- On success case, balance of customer account will be sent back in the response.

Example: : Sample JSON response.


```
{
        "availableBalance": "string",
        "custNo": "string",
        "ccy": "string",
        "openingBalance": "string",
        "currentBalance": "string",
        "blockedAmount": "string",
        "custAcNo": "string"
}
```


3.2.3 TD Account Balance/Renewal Date

This service helps us in retrieving the TD account balance/ Renewal Date by passing account number as input. The following are the expected functionality of the service.

- This service will query for the TD account balance / renewal date that are present in Oracle FLEXCUBE.
- The account should be valid , open and also authorized

- On failure case i.e., if the account searched doesn't present in the system then it will throw an error stating "Invalid TD Account"
- On success case, balance of TD account will be sent back in the response.

Example : Sample JSON response.


```
{
        "custNo": "string",
        "acc": "string",
        "ccy": "string",
        "maturityDate": "string",
        "acOpenDate": "string",
        "maturityAmount": "string"
}
{
        "custNo": "string",
        "acc": "string",
        "ccy": "string",
        "maturityDate": "string",
        "acOpenDate": "string",
        "renewelDate": "string"
}
```


3.3 Core Rest API Services

Following are the list of Core API Rest Services that are available.

Service	Method	Sample URL
Standing Instructions for Accounts	GET	http://(IP):(Port)/AccountService/v12.3/accounts/standingInstructions/{drAccount}
Last 10 Debit Transactions	GET	http://(IP):(Port)/AccountService/v12.3/accounts

Service	Method	Sample URL
for Accounts		/debitTransactions/{accountNumber}
Last 10 Credit Transactions for Accounts	GET	http:///(IP):(Port)/AccountService/v12.3/accounts /creditTransactions/{accountNumber}
Last 10 Debit / Credit Transactions for Accounts	GET	http:///(IP):(Port)/AccountService/v12.3/accounts /allTransactions/{accountNumber}
Retail Customer Loan Accounts	GET	http:///(IP):(Port)/AccountService/v12.3/accounts/loan/{custNo}
Loan Account Balance (Outstanding Principal)	GET	http:///(IP):(Port)/AccountService/v12.3/accounts /loanBalance/{accountNumber}
Loan Instalment Details	GET	http:///(IP):(Port)/AccountService/v12.3/accounts /loanInstallment/{accountNumber}
Foreign Currency Exchange Rate	GET	http:///(IP):(Port)/ExchangeRateService/v2.3/ExchangeRates Query Params: currency1, currency2, branch
Branches List	GET	http:///(IP):(Port)/Branchservice/v12.3/branches/{branchName}
Branch Address	GET	http:///(IP):(Port)/Branchservice/v12.3/branches/address/{branchCode}
Block Amount	POST	http:///(IP):(Port)/AmountBlockService/v12.3/amountBlock/create
Debit Request	POST	http:///(IP):(Port)/SingleJournalService/v12.3/singleJrnl/createContract
Credit Request	POST	http:///(IP):(Port)/SingleJournalService/v12.3/singleJrnl/createContract
Stop Cheque	POST	http:///(IP):(Port)/StopPaymentService/v12.3/stopPayment/createContract

3.3.1 Standing Instructions for Accounts

This service helps us in retrieving the Standing Instructions by passing account number as input. The following are the expected functionality of the service.

- This service will query for the Standing Instructions of the account that are present in Oracle FLEXCUBE.
- The contract status and standing instruction must be authorized.

- On failure case i.e., if the account searched doesn't present in the system then it will throw an error stating "Standing Instructions Not Found"
- On success case, Standing Instructions of the account will be sent back in the response.

Example: Sample JSON response.


```
{
 "contractRefNo": "string",
 "crAccBr": "string",
 "crAccCcy": "string",
 "crAccount": "string",
 "drAccBr": "string",
 "drAccCcy": "string",
 "drAccount": "string",
 "instructionNo": "string",
 "standingInstruction": {
  "firstExecDate": "string",
  "firstValueDate": "string",
  "instVersionNo": "string",
  "instructionNo": "string",
  "nextExecDate": "string",
  "nextValueDate": "string",
  "productCode": "string",
  "productType": "string"
 }
}
```

3.3.2 Account Transaction Detail Services

These services (ie get last 10 debit/credit and both transactions) helps us in retrieving the last 10 debit/credit or both transactions by passing account number as input. The following are the expected functionality of the service.

- This service will query for the debit transactions of the account that are present in Oracle FLEXCUBE.
- The account should be valid, open and authorized.
- On failure case i.e., if the account searched is not present in the system then it will throw an appropriate error message.
- On success case, last 10 debit transactions of the account will be sent back in the response.

Example: Sample JSON response.


```
{
 "custNo": "string",
 "accountNo": "string",
 "debitCreditTransactions": {
  "acEntrySrNo": "string",
  "trnRefNo": "string",
  "acNo": "string",
  "lcyAmount": "string",
  "acCcy": "string",
  "valueDt": "string",
  "txnInitDate": "string"
 }
}
```


3.3.3 Retail Customer Loan Services

These services (ie retrieve loan details, loan balance details and instalment details) help us in retrieving the Retail Customer Loan Accounts and their details by passing customer number or loan account number (according to the parameter defined in the service) as input. The following are the expected functionality of the service:

- These services will query for the Retail Customer Loan Accounts / Loan Outstanding Balance or the Loan Instalment Details that are available in Oracle FLEXCUBE.

- The account should be valid and account status should be authorized
- On failure case i.e., if the account searched doesn't present in the system then it will throw appropriate error message.

Example: Sample JSON response.


```
{
  "custNo": "string",
  "loanAccount": {
    "accountNo": "string",
    "branchCode": "string",
    "bookDate": "string",
    "maturityDate": "string",
    "ccy": "string",
    "originalStDate": "string",
    "productId": "string",
    "productCtgry": "string",
    "custNo": "string"
  }
}
```

```
{
  "accountNumber": "string",
  "outstandingPrincipal": "string",
  "outstandingInterest": "string",
  "ccy": "string",
  "custNo": "string"
```

```
}
```

```
{
  "scheduleDueDate": "string",
  "custNo": "string",
  "emiAmount": "string",
  "accountNumber": "string"
}
```

3.3.4 Foreign Exchange Currency Rate Service

This service helps us in retrieving the Currency Exchange Rates by passing currencies CCY1, CCY2 and branch (Branch code will be picked up from the request header) as input. The following are the expected functionality of the service.

- This service will query for the Exchange Rates of the given currencies that are present in Oracle FLEXCUBE.
- The given currencies must be valid and authorized.
- On failure case i.e., if the currencies searched doesn't present in the system then it will throw an error stating "Exchange rates not found for given currencies"

```
{
  "branchCode": "string",
  "buyRate": "string",
  "midRate": "string",
  "ccy1": "string",
  "ccy2": "string",
  "saleRate": "string",
  "asOn": "string",
  "rateType": "string"
}
```

3.3.5 Branch Query Services

These services (get Branch List and get Branch Address) help us in retrieving the Branches list/ Branch Address by passing either branch name or the branch code as input. The following are the expected functionality of the service.

- This service will query for the Branch address for the given branch code that are present in Oracle FLEXCUBE.
- The branches should be valid, open and authorized.
- On failure case i.e., if the branch code searched doesn't present in the system then it will throw an error stating "Invalid Branch"

Example: Sample JSON response.


```
{
 "branchCode": "string",
 "branchName": "string",
 "bankCode": "string",
 "bankName": "string"
}
```

```
{
 "branchCode": "string",
 "branchAddr1": "string",
 "branchAddr2": "string",
 "branchAddr3": "string",
 "bankName": "string",
 "countryCode": "string"
}
```


3.3.6 Amount Block Service

This service helps us in blocking the amount for an account by passing account, amount, branch and effective date as input. The following are the expected functionality of the service.

- This service will block the amount for the given account which is present in Oracle FLEXCUBE.
- The account should be valid, open and authorized.

- On failure case i.e amount block creation will not happen and throws the relevant error with failure message
- On success case, Amount block number with the success message will be sent back in the response.
- The external system used for the Amount Block Service, should have Auto Auth rights on the below service and operation in FLEXCUBE.

Service	FCUBSCustomerService
Operation	CreateAmtBlk
FC Function Id	CADAMBLK

Example:

Request in JSON	{
	"account": "string",
	"amount": "string",
	"effectiveDate": "string",
	"txnBranch": "string"
	}

Response in JSON format	{
	"account": "string",
	"amount": "string",
	"effectiveDate": "string",
	"txnBranch": "string",
	"amountBlockNum": "string",
	"transactionStatus": "string"
	}

3.3.7 Single Journal Dr/Cr Service

This service helps us in debiting/crediting the amount from an account given by passing account, branch, currency, additional text, amount, batch number, debit/credit, transaction code and ultimate value date as input. The following are the expected functionality of the service:

- This service will debit/credit the amount from the account which is present in Oracle FLEXCUBE.

- The account should be valid, open and authorized.
- On failure case i.e., debit/credit will not happen and throws the relevant error with failure message
- On success case, amount will be debited/credited from the account.
- The external system used for the Single Journal Service, should have Auto Auth rights on the below service and operation in FLEXCUBE.

Service	FCUBSDEService
Operation	CreateSingleJrnl
FC Function ID	DEDTRONL

Example :

Request in JSON format	{
	"accountNumber": "string",
	"valDate": "string",
	"batchNo": "string",
	"currency": "string",
	"amount": "string",
	"accountBranch": "string",
	"transactionCode": "string",
	"addlText": "string",
	"debitOrCredit": "string"
	}

Response in JSON format	{ "accountNumber": "string", "valDate": "string", "batchNo": "string", "currency": "string", "amount": "string",

	"accountBranch": "string", "transactionCode": "string", "addlText": "string", "debitOrCredit": "string", "transactionId": "string", "status": "string" }

3.3.8 Stop Cheque Service

This service helps us in stopping the cheque by passing account, branch, reason, effective date, amount, start check number, end check number and stop payment type as input. The following are the expected functionality of the service.

- This service will stop the cheque payment which is present in Oracle FLEXCUBE.
- The account should be valid, open and authorized.
- The cheque number should be valid and it should belong to the customer account.
- On failure case i.e., stop cheque will not happen and throws the relevant error with failure message
- On success case, given cheque will be stopped for payment.
- The external system used for the Single Journal Service, should have Auto Auth rights on the below service and operation in FLEXCUBE.

Service	FCUBSAccService
Operation	CreateStopPayments
FC Function ID	CADSPMNT

Request in JSON format	{ "accountNumber": "string", "accountBranch": "string", "stopPaymentType": "string", "startCheckNo": "string",

	"endCheckNo": "string", "effectiveDate": "string", "reason": "string" }

Response in JSON format	{
	"accountNumber": "string",
	"accountBranch": "string",
	"stopPaymentType": "string",
	"startCheckNo": "string",
	"endCheckNo": "string",
	"effectiveDate": "string",
	"reason": "string",
	"response": "string"
	}

3.3.9 Rest Services Additional Changes

Support for Rest Services are enhanced to all modules through ODT. In ODT at RAD level user can opt for which Rest services are required, ODT automatically generate rest artefacts for Function IDs opted. These Rest Services can be deployed into application.

List of services enabled



Rest_Services.xlsx

Rest Documentations

Please refer to *Rest Documentaion.zip* to access swagger.json files for the ODT generated services. Replace "host":"10.10.10.10:1010" with the I.P. address and port of the server where the REST service is deployed. This has to be done for all the swagger.json files.

If any service is getting updated or a new service is getting generated, then follow the below steps to generate swagger.json for those services:

a. Generate service EAR file and deploy the service.

b. Once the service is running, to obtain swagger.json, type service URL followed by swagger.json in the browser. E.g. https://10.10.10.10:1010/CustomerService/swagger.json.

The generated swagger.json can be loaded in a swagger UI to obtain the swagger documentation.



Rest Documentation.zip

Please refer the attachment panel for Rest_Services.xlsx and Rest Documentaion.zip.