GetLoanPaymentSchedule API GATEWAY DOC

Table of Contents

1. INTRODUCTION

1.1 Design Document Purpose

The purpose of this document is to provide a detailed specification of the **GetLoanPaymentSchedule** in sufficient depth to:

➢ Enable the component to be built and tested.
➢ Ensure that it can be enhanced, supported and maintained by other areas of the organization after initial implementation.

1.2 Design Reviews

The service design will be reviewed within Middleware team and security testing team as needed. Once finalized, design resource will provide an overview to other teams such as front-end application development team and various project resources.

2. API CONNECT COMPONENTS

➢ API Connect is used to expose the service to front end applications.
➢ Within API Connect, there are multiple Products. The GetLoanPaymentSchedule API is exposed within **GetLoanPaymentSchedule Product.**
➢ Service Consumers must subscribe to the API. A unique application Identification (client-id) and a secret will be generated.
➢ The assigned Client-id must be supplied in the header for each API.
➢ URLs for invoking the services can be found in API Connect Developer portal and mentioned in the below section.

3. STEPS TO ACCESS & SUBSCRIBE IN PORTAL

Refer Subscription User manual shared during initial on board.

4. API AUTHENTICATION

JWT Access token to be passed in JSON wrapper as string value in "JWTokenValue" field. The Token can be generated by subscribing to **TOKEN API**.

4.1 TOKEN API

➢**Overview:** To retrieve access token.

➢**Request Type**: POST

➢**Mandatory HTTP Headers**:
■ X-IBM-Client-Id: xxxxxxxxxxxxxxx
■ X-IBM-Client-Secret: xxxxxxxxxxxxxxxxx

➢**SIT URL**: https://apiuat.ujjivansfb.in/ujjivan/development/v1/tokens
➢**PROD URL:** https://api.ujjivansfb.in/ujjivan/production/v1/tokens

5. GetLoanPaymentSchedule API DETAILS
➢ **Overview:**

Service is designed to retrieve loan EMI payment schedule for a given account. The API Gateway makes a backend ESB call which in turn connects to provider system **CRM** of service name **FindAmortSchedule & GetLoanPaymentSchedule (BR.Net).**

➢ **Request Type**: POST

➢ **API URL**:

UAT :

https://apiuat.ujjivansfb.in/ujjivan/development/v1/getFullAccStatWithPagination/getAccountStmtPagination

PROD : To be done

➢ **Mandatory HTTP Headers**:
   ■ X-IBM-Client-Id: xxxxxxxxxxxxxxx

➢ **Accept**: application/json

➢ **Content-type**: application/json

➢ **REQUEST PAYLOAD:**

At API Gateway the request has to be passed as JSON wrapper. The sample payload is as below,

```
 {
"RequestEncryptedValue": "",
 "TransactionId": "",
 "JWTokenValue": ""
 }
```

The individual fields of the Standard JSON Request Payload are described below-

- **RequestEncryptedValue** : This will contain the encrypted value of original REST-JSON request sample.
- **TransactionId** : External partners need to set a transaction ID to uniquely identify every request, in order to retrieve it from an audit trail at a later date.
- **JWTokenValue** : This value can be obtained by invoking Bank's token generator service.

(For more details on request encryption Refer UjjivanSFB_API_Integration_TechnicalProcess_document)

➢ **Actual REST-JSON Input Parameter:**

Element Name	M/O/C	Data Type	Size	Description	Validation Rules
getLoanPaymentScheduleReq / reqHdr	Complex				
reqHdr/consumerContext/applicationId	M	String	3	Application id from which request originated. Example IB, MB,HHD, BRN	
reqHdr/ serviceContext/ uniqueMsgId	M	Stirng		Unique request message id for each message generated from consumer for tracking purpose.	
reqHdr/ serviceContext/ reqMsgDateTime	M	DateTime		Request time stamp in the format CCYY-MM-DDThh:mm:ss.sss	

reqHdr/ serviceContext/ serviceName	M	String		ServiceName to be provided.	
reqHdr/ serviceContext/ serviceVersion	O	String		Service version to be provided. It's value is 1.	
reqHdr/ providerContext/ providerId	O/F	String	3	Provide application Id from where data expecting	
reqHdr/ userContext/ appUserID	M	String	20	AppUserId will be provided which is created unique to user. Client need to pass this value each and every request.	
reqHdr/ userContext/ appPassword	M	String	64	AppPassword will be provided which is specific to user. Client need to pass this value each and every request.	
getLoanPaymentSched uleReq / body	M	Complex			
getLoanPaymentSched uleReq /body/accountNumber	M	String	16	Loan account number of the account for amendment or rescheduling of loans	

➢ **RESPONSE PAYLOAD STRUCTURE**

A standard JSON wrapper containing encrypted response will be obtained. The encrypted response is of AES/CBC/256 mode with Initiaization vector concatenated at the start of original JSON response.

The sample response structure is as below,

```
{
  "ResponseOfEncrytptedValue": "<encrytped value>",
  "TransactionId": "162193467244544"
}
```

Using the static key shared by Ujjivan Bank, decryption of the value in tag ResponseEncryptedValue should be performed with AES/CBC/256/IV
TransactionId: This field will contain the transaction ID which was passed in request.

(For more details on response encryption Refer UjjivanSFB_API_Integration_TechnicalProcess_document)

➢ **ACTUAL OUTPUT PARAMETERS:**

Response					
getLoanPaymentSche duleRes /resHdr	Complex				
resHdr/consumerCon text/applicationId	M	String	3	Value will be echoed from request message	
resHdr/ serviceContext/ uniqueMsgId	M	Stirng		Value will be echoed from request message	
resHdr/ serviceContext/ reqMsgDateTime	O	DateTime		Value will be echoed from request message	
resHdr/ serviceContext/ serviceName	M	String		Value will be echoed from request message	

reqHdr/ serviceContext/ serviceVersion	O	String		Value will be echoed from request message	
resHdr/ providerContext/ providerId	M	String	3	Provider System Id from where data is sending	
resHdr/ providerContext/ responseMsgDateTime	O	DateTime		Response message date and time stamp.	
resHdr/responseStatus/status	M	String		Response status from ESB Status 0 = Successful, 1 = Failure	
resHdr/responseStatus/ esbResDateTime	M	DateTime		Response date time stamp from ESB	
getLoanPaymentScheduleRes / body	M	Complex			
getLoanPaymentScheduleRes /body/amortizationDtls/ amortizationScheduleDtls	M	Complex/ Repeating		Returns the loan schedule details	
amortizationScheduleDtls/ combInstlmentAmt	M	Decimal		Holds Sum of installement amount and fixed charge amount	
amortizationScheduleDtls/ cummInterestAmt	M	Decimal		Holds The total interest amount due for repayment for the particular installment	
amortizationScheduleDtls/ cummPrincipalAmt	M	Decimal		holds The total principal amount due for repayment for the particular installment	
amortizationScheduleDtls/ flowDate	M	Date		Holds date on which flow occurs	
amortizationScheduleDtls /instalmentlAmt	M	Decimal		Holds the installment amount to be amortized	
amortizationScheduleDtls /interestAmt	M	Decimal		Holds the interest amount that has to be amortized	
amortizationScheduleDtls /principalAmt	M	Decimal		holds the principal amount that has to be amortized	
amortizationScheduleDtls /principalOutStandingAmt	M	Decimal		Holds The amount pending for amortization after the principal amount is amortized after each installment	
amortizationScheduleDtls /serialNumber	M	String	6	Holds the primary key	

getLoanPaymentScheduleRes /body/errorInfo	O, Repeating	Complex			Will be populated in case of any business exception from back end
getLoanPaymentScheduleRes /body/errorInfo/ errorSource	O	String		If Status = 1, value will be populated with error system source	Will be populated in case of exceptions thrown by ESB
getLoanPaymentScheduleRes /body/errorInfo/ errorCode	C	String		If Status = 1, value will be populated with error code	
getLoanPaymentScheduleRes /body/errorInfo/ errorDescription	C	String		If Status = 1, value will be populated with error description	
getLoanPaymentScheduleRes /body/errorInfo/ errorType	O	String	1	If Status = 1, value will be populated with error Type	
resHdr/ additionalDetails/ details1	O/F	String		Future Use	
resHdr/ additionalDetails/ details1	O/F	String		Future Use	
resHdr/ additionalDetails/ details1	O/F	String		Future Use	

➢ **Error Handling**

HTTP Status Code	HTTP Message	More Information
401	Unauthorized	Access token missing or validation
400	Bad Request	The parameters are invalid or missing.
503	Service Unavailable	The parameters were valid but the request failed.
200	Refer section 5 for more details	

--**END OF DOCUMENT**--