



CheckForDuplicate API GATEWAY DOC

Ujjivan Small Finance Bank Limited
Grape Garden, No.27, 3rd A Cross, 18th Main, 6th Block, Koramangala, Bengaluru – 560 095

Table of Contents

1. INTRODUCTION	2
2. API CONNECT COMPONENTS.....	3
3. STEPS TO ACCESS & SUBSCRIBE IN PORTAL	3
4. AUTHENTICATION	3
5. CheckForDuplicate API DETAILS.....	3

1.1 Design Document Purpose

The purpose of this document is to provide a detailed specification of the **CheckForDuplicate** 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 CheckForDuplicate API is exposed within **CheckForDuplicate Products.**
- 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 also mentioned in the below section.

3. STEPS TO ACCESS & SUBSCRIBE IN PORTAL

Refer Subscription User manual shared during initial on board.

4. AUTHENTICATION

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

3.1 TOKEN API

- **Overview:** To retrieve access token.
- **Request Type:** GET
- **Mandatory HTTP Headers:**
 - X-IBM-Client-Id: xxxxxxxxxxxxxxxx
 - X-IBM-Client-Secret: xxxxxxxxxxxxxxxx
- **SIT URL:** <https://apiuat.ujjivansfb.in/ujjivan/development/v1/tokens>

5. CheckForDuplicate API DETAILS

- **Overview:**

Service is designed to dedupe check based on the parameters. The API Gateway makes a backend ESB call which in turn connects to provider system **PDX** of service name **CheckDuplicate**.

- **Request Type:** POST
- **API URL:**
 - UAT : <https://apiuat.ujjivansfb.in/ujjivan/development/v2/findCustomer/checkForDuplicate>
 - PROD : <https://api.ujjivansfb.in/ujjivan/development/v1/findCustomer/checkForDuplicate>
- **Mandatory HTTP Headers:**
 - X-IBM-Client-Id: xxxxxxxxxxxxxxxx
- **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": "",
  "JWTTokenValue": ""
}
```

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.
- **JWTTokenValue** : This value can be obtained by invoking Bank's token generator service.

(For more details on request encryption Refer UjivanSFB_API_Integration_TechnicalProcess_document)

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

Element Name	M/O /C	Data Type	Size	Description	Validation Rules
checkForDuplicateReq/reqHdr		Complex			
reqHdr/consumerContext/applicationId	M	String	3	Application id from which request originated. Example IB, MB,HHD, BRN	
reqHdr/ serviceContext/uniqueMsgId	M	String		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.	
checkForDuplicateReq/body	Complex				
body/ requestid	M	String	30	Need to send unique number for every dedupe request	Need not be exceed by 30.
body/ clientid	M	String	30	It can be same value as requestid and it should be unique for every request.	
body/ sourceId	O	String		SourceId should be there. LIKE HHD, BRN	
body /name	M	String			
body /dateOfBirth	O	Date		Dd/MM/YYYY	
body /gender	O	String			
body /motherName	O	String			
body /fatherName	O	String			
body /maritalStatus	O	String		Need to pass string value like MARRIED or SINGLE	
body /spouse	O	String			
body /aadhar	O	String			
body /pan	O	String			
body / passport	O	String			
body /voterId	O	String			
body /drivingLicenceNo	O	String			
body /rationCard	O	String			
body /nREGA	O	String	20	National Rural Employment Guarantee	
body / nPR	O	String	20	National Population Register	
body / cKYC	O	String	20	CKYC number	
body / status	O	String	30		
body /contactDtls	M	Complex/ repeating			
contactDtls /mobileNumber	M	Number	10		

body /email	O	Complex/ repeating			
email /emailId	O	Number	125		
body / addressDtls	M	Complex/ Repeating			
addressDtls /addresstype	O	String			
addressDtls / address	M	String			
addressDtls /city	O	String			
addressDtls /state	O	String			
addressDtls /pincode	O	String			
addressDtls /landmark	O	String			

➤ **RESPONSE PAYLOAD STRUCTURE**

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

The sample response structure is as below,

```
{
  "ResponseOfEncryptedValue": "<encrypted 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					
checkForDuplicateRes /resHdr	Complex				
resHdr/consumerContext /applicationId	M	String	3	Value will be echoed from request message	
resHdr/consumerContext / terminalId	O	String		Value will be echoed from request message	
resHdr/ serviceContext/ uniqueMsgId	M	String		Value will be echoed from request message	
resHdr/ serviceContext/ reqMsgDateTime	O	DateTime		Value will be echoed from request message	
resHdr/ serviceContext/ timeZone	O	String	3	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/ serviceContext/ conversationId	O	String		Value will be echoed from request message	
resHdr/ serviceContext/ parentMsgId	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	
resHdr/responseStatus/ errorSource	O	String		If Status = 1, value will be populated with error system source	Will be populated in case of exceptions thrown by ESB
resHdr/responseStatus/ errorCode	C	String		If Status = 1, value will be populated with error code	
resHdr/responseStatus/ errorDescription	C	String		If Status = 1, value will be populated with error description	
resHdr/responseStatus/ 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	
checkForDuplicateRes/ body	M	Complex			
body/requestId	M	String	16		
body/status	M	String	1		
body/matchCount	M	String	3		
body/message	M	String	25		
Body/matchType	M	String			
body/ sourceId	O	String			
Body / customerDtIs	M	Complex/ Repeating			
customerDtIs / custId	M	String	16		
customerDtIs/name	M	String			
customerDtIs/dateOfBirth	O	Date			
customerDtIs/gender	O	String			

customerDtls/motherName	O	String			
customerDtls/fatherName	O	String			
customerDtls/martialStatus	O	String			
customerDtls/spouse	O	String			
customerDtls/aadhar	O	String	12		
customerDtls/pan	O	String	10		
customerDtls/ passport	O	String			
customerDtls/voterId	M	String			
customerDtls/drivingLicenceNo	O	String			
customerDtls/rationCard	O	String			
body/customerDtls / contactDtls	M	Complex/ Repeating			
ContactDtls /mobileNumber	M	Number	10		
body/customerDtls / email	O	Complex/ Repeating			
email /emailId	O	Number	125		
body/customerDtls/addressDtls	M	Complex/ Repeating			
addressDtls /addressstype	O	String			
addressDtls / address	M	String			
addressDtls /city	O	String			
addressDtls /state	O	String			
addressDtls /pincode	O	String			
addressDtls /landmark	O	String			
checkForDuplicateRes/body/ errorInfo	O, Repeating	Complex			Will be populated in case of any business exception from back end
errorInfo/errorCode	M	String		Error code	
errorInfo/errorDescription	M	String		Error Desc	
errorInfo/errorType	O	String		Error Type	
errorInfo/errorSource	O	String			Application code where error generated

➤ **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.

-----END OF DOCUMENT-----