LSP Details & API Specifications

[**Brief**](#_8k1jnkjjdkgd) **6**

[Problems in current lending systems](#_9eheyhprq5du) 7

[The need](#_6isx6nbfenyg) 7

[Available infrastructure](#_obs2oaogpb4j) 7

[The solution](#_1zzfvi20xmh) 7

[Steps for MSME to avail invoice financing](#_n5cnabb8o6eg) 8

[How Perfios helps lenders leverage B4SME](#_1fob9te) 9

[Brief Architecture of LSP](#_3znysh7) 9

[**API Details**](#_ljkel8lbxbar) **10**

[**Prerequisites**](#_iyevvao9rryp) **10**

[**API Header**](#_cl45pllw4u70) **10**

[**Create Loan Application API**](#_y8yqlcecw8ka) **10**

[Request Parameters](#_bbxb6clwt815) 11

[Response Parameters:](#_1zw9iqu64h4s) 11

[**Sample Invoices**](#_y9r05gc4h77m) **12**

[**Sample Borrower Data**](#_gcea8wcb8pua) **13**

[Request Details:](#_z8orld4nllfn) 14

[Sample Request:](#_vlpp3hayp2vq) 14

[Sample Response:](#_qe3jspm6cu3f) 17

[**Error Codes**](#_qsfnmu13jvvn) **18**

[Sample Error Request](#_i8fqhk9pe57g) 18

[**Trigger OTP API**](#_t5fttd2njyzw) **18**

[Request Parameters:](#_qe9j44lsqlm3) 19

[Response Parameters](#_stbg4vud6n3v) 19

[Request Details:](#_ax7zjavndsz4) 20

[Sample Request](#_n2jmd8ttzseq) 20

[Sample Response](#_65x59wjrk315) 20

[Error Codes](#_wwh8sdj8396p) 21

[Sample Error Request](#_5ieeimt9t3ss) 21

[**Verify OTP API**](#_ag12tqsxqp5d) **21**

[Request Parameters](#_gca87mo2ecmw) 21

[Response Parameters](#_qtv782kk50vp) 22

[Request Details:](#_dvc0g0so3t4b) 22

[Sample Response](#_1t3uxuv4c1nc) 22

[Error Codes](#_srlshveppigj) 22

[Sample Error Request](#_tahk94jg8ytv) 23

[**Grant Loan Request API**](#_bk2wkqpeiz6m) **23**

[Request Parameters](#_q9ht3yacfcaz) 24

[Response Parameters](#_eohtq3e4dodo) 24

[Request Details](#_fzoqxfmuqmed) 24

[Sample Request](#_br1hijpbhur9) 24

[Sample Response](#_lwb6ym5hqjz6) 25

[Error Codes](#_bg8b9sxkqr7) 25

[Sample Error Request](#_7bfejcu6evbs) 25

[**Loan Summary API**](#_glkt3g1hfe7x) **26**

[Request Parameters](#_hg7142alr7x5) 26

[Response Parameters](#_pqwf7twcjvbd) 26

[Request Details](#_cknmw3dbg9y6) 27

[Sample Request](#_3orz5dbsckta) 28

[Sample Response](#_wgvqo6iu3u34) 28

[Error Codes](#_12bqwzc1ywow) 29

[Sample Error Request](#_c9me2fsun2g9) 29

[**Loan Statement API**](#_aeb37c319wq) **30**

[Request Parameters](#_87btf9y6c0hh) 30

[Response Parameters](#_4369624psljv) 30

[Request Details](#_kj9ir7qbvm7g) 31

[Sample Request](#_t9syuxqk0kfg) 31

[Sample Response](#_ah8mqb1lsfpt) 32

[Error Codes](#_8o4fxvejvjbm) 32

[Sample Error Request](#_l4zg4wc2p7aq) 32

[**Set Disbursement Plan API**](#_xi2vjzlz95h9) **33**

[Request Parameters](#_gtvt3tgwgzu9) 33

[Response Parameters](#_nq430bn14juz) 33

[Request Details](#_px0pwedgkhsm) 34

[Sample Request](#_d7i77deeucqu) 34

[Sample Response](#_27bvj1gd2j3r) 34

[Error Codes](#_gn5luzm13m2n) 34

[Sample Error Request](#_l385an48hr3a) 34

[**Set Disbursement Account**](#_j221lv8e6h27) **35**

[Request Parameters](#_wlce2j40inzr) 35

[Response Parameters](#_deoexufiyn17) 36

[Request Details](#_5j6h32jarmwj) 36

[Sample Request](#_4igjx4gjb3m5) 36

[Sample Response](#_xu1wxfltt2rf) 36

[Error Codes](#_k33ooq7vaork) 36

[Sample Error Request](#_meokaffgelc5) 36

[**Trigger Disbursement**](#_wuszra4fk8o8) **37**

[Request Parameters](#_jay0i442h8t) 37

[Response Parameters](#_svq2fq4597cr) 38

[Payment Details](#_rn9q322kua30) 38

[Request Details](#_ywnqws3m138z) 39

[Sample Request:](#_3q4bgosvq7pt) 39

[Sample response](#_flkio8bttn42) 39

[**Get Offers API**](#_2jnx2qhtcx8e) **40**

[Request Parameters](#_d5251pw9bs3t) 40

[Response Parameters](#_hej2rzg2wqr7) 40

[Request Details](#_l8prrr9kh2fz) 41

[Sample Request](#_ssdriuard4) 41

[Sample response](#_v88irtyrjvt7) 41

[Error Codes](#_bs4ex7l7upe1) 44

[Sample Error Request](#_t7xvog6f6jlg) 45

[**Set Offer API**](#_hjeq87gg5cn) **45**

[Request Parameters](#_mojuoc57gxfo) 45

[Response Parameters](#_xq4esq0hsbc) 46

[Request Details](#_p1hif87h7e9l) 46

[Sample Request](#_ovugzdim3zjy) 46

[Sample Response](#_n5wgwq2g0tdl) 47

[Error Codes](#_9y1o2cgta9jq) 47

[Sample Error Request](#_xdgg345u6smb) 47

[**Trigger Loan Acceptance API**](#_lnczpz7pybh4) **47**

[Request Parameters](#_w4emx5618go) 48

[Response Parameters](#_q8jntm75n8o3) 48

[Request Details](#_k26hj1vl06wz) 49

[Sample Request](#_ffvduc7s9mhg) 49

[Sample Response](#_t2i5urns4ykl) 49

[Error Codes](#_fombapv8ffup) 49

[Sample Error Request](#_m6i5ypnql8vs) 49

[**Verify Loan Acceptance API**](#_etudwf1y2ut4) **50**

[Request Parameters](#_yhldx9d14mv9) 50

[Response Parameters](#_ht7pajn45owc) 51

[Request Details](#_frr0dxs8vxkr) 51

[Sample Request](#_uj77mcfvmesb) 51

[Sample Response](#_r07tyehuhex3) 51

[Error Codes](#_ld60gkpimcsg) 52

[Sample Error Request](#_mxjcc6kvwtft) 52

[**Repayment APIs**](#_1e32w5nlcv0f) **53**

[**Set Repayment Plan API**](#_g86u97hvn9tl) **53**

[Request Parameters](#_ya9bzbfhde5q) 53

[Response Parameters](#_8qep85egzs5j) 54

[Request Details](#_hr8rnavkwa69) 54

[Sample Request](#_d58qau4nnfea) 54

[Sample Response](#_l4smdbau1l3a) 54

[Error Codes](#_48jc0tralmhh) 55

[Sample Error Request](#_56z60xoav5z0) 55

[**Trigger Repayment**](#_e4zgp9365h54) **55**

[Request Parameters](#_yvyjeh46rh2h) 56

[Response Parameters](#_afjt1zq1rxiq) 56

[Request Details](#_wic10epu22i3) 56

[Sample Response](#_3f3zw6vmsj74) 56

[Error Codes](#_vysuqs5jisql) 57

[Sample Error Request](#_c9n21zf27gpr) 57

[**Repayment Status API**](#_m8bcsem95bt8) **57**

[Request Parameters](#_k9xot07c8tgd) 58

[Response Parameters](#_58k7to9ja55q) 58

[Request Details](#_tvop43ofr0t2) 58

[Sample Request](#_sftasbe6bbpb) 58

[Sample Response](#_4xgbhxj7yrdv) 59

[Error Codes](#_p68veflv2vq0) 59

[Sample Error Request](#_k03sp1ijn0h5) 59

[**Download Analysed Report API**](#_51y5o64iv9o9) **60**

[Request Parameters](#_v0zxlivrxxr) 60

[Request Details:](#_834vbxujpbcw) 60

[Sample Response:](#_ejs0vjajan7r) 60

[**Transaction Callback API**](#_ng3tngjovetk) **61**

[Request Parameters](#_fgttd21y63) 61

[Response Parameters](#_qz2rc0cvciuv) 61

[Request Details](#_pq77odjha457) 62

[Sample request](#_15febp598448) 62

[Sample Response](#_xfhx798tg0em) 62

[Error Codes](#_9editi50c70) 62

[Sample Error Request](#_lhci91b63aj7) 62

[**Offer Status Callback API**](#_jqdadvb8c0z4) **63**

[Request Parameters](#_jw83t5dm4nkq) 63

[Response Parameters](#_11s8tekyphvn) 64

[Request Details](#_dmcum2w9yrru) 64

[Sample Request](#_z98ahsczhn4m) 64

[Sample Request](#_k00d3454djek) 64

[Error Codes](#_fy608an3rxlt) 65

[Sample Error Request](#_ntprr64evevi) 65

[**Loan Status API**](#_yrlj01h3cw8a) **65**

[Request Parameters](#_nf0l308tg6re) 65

[Response Parameters](#_qzhb83ah1b2) 66

[Request Details:](#_b0js5tbqxrke) 66

[Sample Request](#_ffu0jnnk18vl) 66

[Sample Response](#_r1xvfmq54po) 66

[Error Codes](#_zg3luuezbbrm) 67

[Sample Error Request](#_dfq8ndpzoc91) 67

[**Raise Dispute**](#_qal2kpqvkgvw) **68**

[Request Parameters](#_dhq8prokeidu) 68

[Response Parameters](#_ugsseudbdo8o) 68

[Request Details](#_o9wz7ncxwoc7) 69

[Sample Request](#_cqnyc2fds9j3) 69

[Sample Response](#_tjuuxyhlqjkd) 69

[Error Codes](#_lyd8p5353nmj) 70

[Sample Error Request](#_45azhkluto50) 70

[**Dispute Status**](#_dmoe1frmhsos) **70**

[Request Parameters](#_gb28sur5ye1p) 70

[Response Parameters](#_fbtqg0tfz17v) 71

[Request Details](#_trrv0p1azu91) 71

[Sample Request](#_u07merlrv3wq) 71

[Sample Response](#_u7mtj5ieg1re) 72

[Error Codes](#_wvlwnds9nqdc) 72

[Sample Error Request](#_vcnd6j1nkk3g) 72

[**Flow Diagram**](#_yqy9vcy0vh8x) **73**

[**Mapping of flow**](#_jgqed6po7a8j) **75**

[**Mapping of API**](#_h56q894imf2r) **78**

[**User Journey & API Sequence**](#_pao2uwlucevs) **80**

LSP

# Brief

BHIM for MSME Lending (“B4L”) is a scalable, safe, quick Infusion of credit through Invoice Financing for MSMEs.

As per U K Sinha led RBI Panel on MSME, Micro, Small and Medium Enterprises form a vital component of the Indian Economy. The Government and Reserve Bank of India have taken a number of measures from time to time to support this sector. A number of committees appointed by the Government/RBI have identified issues and made recommendations in the past. Many of these recommendations have guided different policy initiatives. However, MSMEs continue to face challenges of formalization, access to knowledge services, access to timely and adequate finance, improving competitiveness, availability of skilled manpower, access to latest technology and marketing. The MSME sector is yet to benefit from the advances in digitization, which can substantially reduce the cost and time for this sector. SHGs and rural entrepreneurship have made very good progress in the last few years, but creating the environment for the next stage of their growth is a task yet to be accomplished. The sector was also affected in the recent past due to structural changes in the economy.

## Problems in current lending systems

1. Most bank lending to SMEs is securitized and lumpy. SMEs need to provide scarce collateral, and receive loans whose amounts, terms and tenures are out of sync with their most pressing need – working capital finance
2. SME’s customers hold the balance of power, sometimes delaying vendor payments due to their own cashflow cycles. While the SME is starved of working capital, they are still expected to meet production and supply targets
3. Invoice discounting platforms such as TReDS attempt to solve the problem, though they still underwrite the buyer – who continues to hold the balance of power re: accepting invoices
4. While some new-age fintech lenders meet SME’s needs of hi-frequency, low-value, timely credit, these lenders lack the reach and balance sheet strength to offer a scalable solution
5. Lenders lack a systemically enforceable lien, monitoring and collection mechanism

## The need

1. A way to combine the low-cost capital of traditional lenders with the low cost of acquisition of marketplaces that aggregate SMEs
2. A new approach that is completely digital in nature, uses public digital infrastructure to improve timeliness, velocity and safety of credit
3. A legally tenable and systemic framework to protect lenders by providing visibility to borrowers’ financial data for timely intervention

## Available infrastructure

1. Lendable invoice via GST
2. Complete digital application & continuous monitoring of bank account via Account Aggregator (AA)
3. Public Credit Registry (PCR) to prevent multiple financing.
4. Disbursement through UPI
5. E-lien and mandates through NPCI

## The solution

The Ministry of Finance is reviewing the Sahay App - an app-based solution for invoice financing to help MSME for better accessibility to credit. The solution brings together traditional lenders with Loan Service Providers (LSPs) that aggregate SMEs and can in future augment the solution with their transactional data to enrich risk-based pricing

## Steps for MSME to avail invoice financing

* Create Sahay Account
* Share GST Invoices for loan
* Register with Account Aggregator and give consent to share past financial information.
* Request loan offers
* Select loan offers
* Select bank account for loan deposit
* Give consent to Account Aggregator to access financial information (Bank account & GST) till loan tenure for credit monitoring,
* Agree to loan terms
* Loan amount transferred.

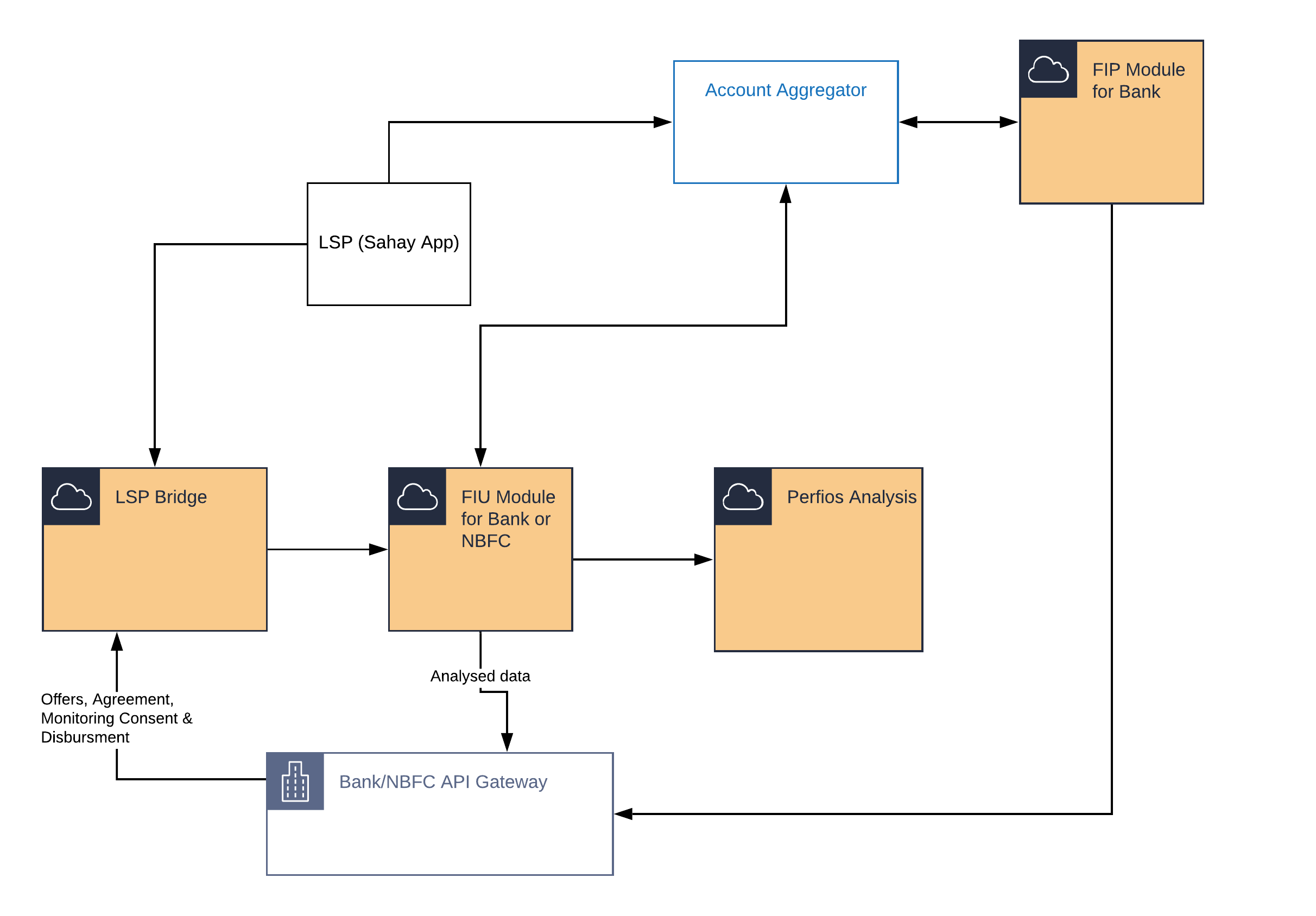
The Sahay App is a reference architecture that LSPs – such as Amazon, Flipkart, Ola, Swiggy, Zomato - can build on top of, much like various UPI payment apps built on top of the BHIM UPI reference architecture

## How Perfios helps lenders leverage B4SME

Perfios provides the following solutions to enable any Bank/NBFC to quickly plug into the Sahay system.

* LSP Bridge
* FIU (Financial Information User) Module
* Account Aggregator (Perfios AA)
* FIP (Financial Information Provider) Module
* Existing Analysis of Bank & GST will continue to feed to Bank/NBFC BRE

## Brief Architecture of LSP



# API Details

This API document provides API’s that have to be implemented by Lenders to participate in any LSP program.

# Prerequisites

Lender should be FIU ready & FIP ready (if applicable)

# API Header

All API should have following headers mandatorily

|  |  |  |
| --- | --- | --- |
| **Header key** | **Sample Header Value** | **Data Type** |
| lsp\_id | abcdSME | String (6, 20) |
| api\_key | 87fd4fae-46f9-11ea-af29-47db1bc3dd6b | String (36) |

# Create Loan Application API

This API is triggered to create a loan request at the lender API Gateway.

Method: POST

Content-Type: Application/JSON

URI: **/lsp/v1/createLoanApplication**

## Request Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| txnId | Unique Transaction Id | 4aafa5b6-46f9-11ea-9dc8-ff5a54aeda44 | String (36) | Mandatory |
| type | Loan Type | CASHFLOW | String (20) | Mandatory & hardcoded |
| borrower | Borrower Details |  | JSON | Mandatory |
| name | Name of Borrower | Raman Singh | String | Mandatory |
| contactDetails | Contact Details of borrower from GSTN Profile |  | JSON Array | Mandatory |
| documents | GSTN Profile data |  | JSON Array | Mandatory |
| collaterals | Eligible Invoices |  | JSON Array | Mandatory |
| terms | Loan request terms |  | JSON | Mandatory |
| requestedAmount | Requested loan amount |  | Integer | Mandatory |
| currency | Currency | INR | String(3,3) | Mandatory and hardcoded. |
| applicants | Applicant details |  | JSON Array | Mandatory |

## Response Parameters:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| status | Loan Status | PROCESSING | OFFERED | GRANTED | REJECTED| ACTION\_REQUIRED | ENUM | Mandatory |
| lender | Lender Details |  | JSON | Mandatory |

## Sample Invoices

The invoice data would be base64 encoded in creating a loan application.

The schema for GSTN invoice can be found in the ANX1-GetB2B API published in [GSTN](https://developer.gst.gov.in/apiportal/taxpayer/newreturns.) portal

However, for the first phase, we might actually be using the old returns api - GSTR1 - Get B2B Invoices. The documents.data field on decoding gives the B2B invoices return.

**{**

**"cflag": "N",**

**"updby": "S",**

**"pos": "29",**

**"rchrg": "N",**

**"inum": "JP/201910/114",**

**"chksum": "e21278dd6a91ec980373df84045379ff46fc558b300c486d5043d1483fdf1159",**

**"itms": [{**

**"itm\_det": {**

**"camt": 13664.61,**

**"samt": 13664.61,**

**"iamt": null,**

**"txval": 151829,**

**"csamt": 0,**

**"rt": 18**

**},**

**"num": 1801**

**}],**

**"val": 179158.22,**

**"inv\_typ": "R",**

**"idt": "04-10-2019"**

**}**

## Sample Borrower Data

JSON schema of borrower data as received in creating loan applications.  
  
**{**

**"state\_cd": "RJ506",**

**"state": "Circle-B, Bharatpur, AC / CTO Ward",**

**"pradr": {**

**"ntr": "Retail Business, Wholesale Business",**

**"stcd": "Rajasthan",**

**"bno": "P.No.-24, Basement",**

**"lt": "",**

**"st": "Aasha Vihar, Rani Sati Nagar, Nirman Nagar, Ajmer Road",**

**"lg": "",**

**"flno": "",**

**"pncd": "302019",**

**"bnm": "",**

**"loc": "Jaipur"**

**},**

**"status": "Active",**

**"lastUpdateDate": "2019-05-15",**

**"center\_cd": "WO0602",**

**"tradename": "M/S SUPRIYA TRADING COMPANY",**

**"name": "MANISH KUMAR",**

**"type": "Regular",**

**"center": "GST RANGE-XXVI BHARATPUR",**

**"adadr": [{**

**"addr": {**

**"dst": "Jaipur",**

**"city": "",**

**"loc": "JAIPUR",**

**"bno": "6/12",**

**"lt": "",**

**"st": "VAISHALI NAGAR",**

**"lg": "",**

**"flno": "",**

**"pncd": "302021",**

**"bnm": "CHITRA KUT",**

**"stcd": "Rajasthan"**

**},**

**"ntr": "Retail Business, Wholesale Business"**

**}],**

**"gstin": "08ACWPK7904K1ZV",**

**"registrationDate": "2017-07-01",**

**"fetchTime": "2019-12-25T18:43:44.052",**

**"constitution": "Proprietorship",**

**"nature": ["Retail Business", "Wholesale Business"],**

**"cancellationDate": null,**

**"status\_code": 1**

**}**

## Request Details:

### Sample Request:

{

"transactionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"type": "CASHFLOW",

"borrower": {

"name": "John doe",

"contactDetails": [

{

"type": "PRIMARY",

"phone": "7777777777",

"email": "abc@xyz.com",

"address": {

"co": "<care of>",

"hba": "<House/Building/Apartment>",

"srl": "<Street/Road/Lane>",

"landmark": "",

"als": "<Area/Locality/Sector>",

"vtc": "<Village/Town/City>",

"pincode": "",

"po": "",

"district": "",

"state": "",

"country": "",

"uri": "<digital address>",

"latitude": "",

"longitude": ""

}

}

],

"guarantors": [],

"documents": [

{

"format": "JSON",

"reference": "b8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"source": "GSTN",

"sourceIdentifier": "GSTN",

"type": "GSTN\_PROFILE",

"isDataInline": true,

"data": ""

}

]

},

"collaterals": [

{

"collateralPrimaryId": "<GSTIN>\_<INVOICENUM>",

"collateralPrimaryIdType": "GST\_INVOICE",

"type": "GST\_INVOICE",

"additionalIdentifiers": [],

"valuation": {

"value": "",

"currency": "",

"date": "",

"source": "GSTN"

},

"parties": [],

"documents": [

{

"format": "JSON",

"reference": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"source": "GSTN",

"sourceIdentifier": "GSTN",

"type": "GSTN B2B INVOICE",

"isDataInline": true,

"data": "eyJjaGtzdW0iOiJCQlVJQlVJVUlKS0tCSktHVVlGVEZHVVkiLCJ1cGRieSI6IlMiLCJpbnVtIjoiUzAwODQwMCIsImlkdCI6IjI0LTExLTIwMTYiLCJ2YWwiOjcyOTI0OC4xNiwicG9zIjoiMDYiLCJyY2hyZyI6Ik4iLCJldGluIjoiMDFBQUJDRTU1MDdSMVo0IiwiaW52X3R5cCI6IlIiLCJjZmxhZyI6Ik4iLCJkaWZmX3BlcmNlbnQiOjAuNjUsIm9wZCI6IjIwMTYtMTIiLCJpdG1zIjpbeyJudW0iOjEsIml0bV9kZXQiOnsicnQiOjUsInR4dmFsIjoxMDAwMCwiaWFtdCI6MzI1LCJjYW10IjowLCJzYW10IjowLCJjc2FtdCI6MTB9fV19"

}

]

}

],

"terms" : {

"requestedAmount" : "50000.00",

"currency" : "INR"

}

}

### Sample Response:

{

"status" : "PROCESSING",

"lender" : {

"primaryId" : "AWW PA7645M" ,

"primaryIdType" : "PAN" ,

"additionalIdentifiers" : [],

"name" : "ABC Bank" ,

"category" : "ORGANIZATION" ,

"contactDetails" : [],

"documents" : []

}

}

## Error Codes

### Sample Error Request

{

"code" : "ERROR\_LOAN\_TYPE\_NOT\_SUPORTED",

"message" : "Loan type not supported.",

"success" : false

}

|  |  |
| --- | --- |
| ERROR CODE | Message |
| ERROR\_LOAN\_TYPE\_NOT\_SUPORTED | Loan type not supported |
| ERROR\_LOAN\_AMOUNT\_EXCEEDED | Requested loan amount exceeds limit. |
| ERROR\_INVALID\_COLLATERAL\_DATA | Invalid collateral data |
| ERROR\_INVALID\_BORROWER\_DATA | Invalid borrower data |

# 

# Trigger OTP API

SMS Format for the OTP

[#] {6 Digit OTP} is the One Time Password (OTP) to complete the loan process with {Lender Name}. {appToken}

Eg- 467866 is the One Time Password (OTP) to complete the loan process with ABCD Bank.

This API is triggered to initiate an otp to authenticate the number linked with the disbursement account.

Method: POST

Content Type: Application/JSON

URL: **/lsp/v1/triggerOtp**

## Request Parameters:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier. Same throughout the loan journey |  |  |  |
| selectedOfferIds | Id’s of the offers selected by the user. Would be the same as the id sent in response to generating offers API. |  | Array | Yes |
| disbursementAccountDetails |  |  |  |  |

## Response Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for the loan application |  | String(36) | Yes |
| sessionId | Unique sessionId related to the otp request. Should expiry after 15 mins. |  | String(36) | Yes |
| otp | Otp sent to the borrower(This field will be present if the lender doesn’t want to do the verification of otp. Verification will be done by us.) |  | int(6) | No |

## Request Details:

### Sample Request

{

"transactionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"selectedOfferIds": [

"BKCCMSME10"

],

"disbursementAccountDetails": {

"accountType": "CURRENT",

"accountIFSC": "KHS6761232",

"accountNumber": "123456781234"

}

}

### Sample Response

{

"transactionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"sessionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"otp": "123455"

}

## Error Codes

### Sample Error Request

{

"code" : "ERROR\_OTP\_PROVIDER\_DOWN",

"message" : "Otp provider is down. Unable to send message",

"success" : false

}

|  |  |
| --- | --- |
| **ERROR CODE** | **Message** |
| ERROR\_OTP\_PROVIDER\_DOWN | Otp provider is down. Unable to send message |

# 

# Verify OTP API

API to verify OTP sent to the borrower for a given loan application

Method: GET

URL: **/lsp/v1/verifyOtp/{sessionId}/{otp}**

## Request Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| sessionId | SessionId related to the otp request |  | String | Yes |
| otp | Otp Sent to the user |  | int | yes |

## Response Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| success | Status of otp validation | true|false | boolean | yes |
| message | Relevant output message |  | String | yes |

## Request Details:

### Sample Response

{

"success": true,

"message": "Otp Authenticated"

}

## Error Codes

### Sample Error Request

{

"code" : "ERROR\_OTP\_PROVIDER\_DOWN",

"message" : "Otp provider is down. Unable to send message",

"success" : false

}

|  |  |
| --- | --- |
| **ERROR CODE** | **Message** |
| ERROR\_INVALID\_OTP | Otp provided is invalid. |
| ERROR\_INVALID\_OTP\_SESSION\_KEY | Otp session key is invalid or expired. |
| ERROR\_MULTIPLE\_\_INCORRECT\_OTP\_ATTEMPTS | Multiple incorrect otp attempts. |

# 

# Grant Loan Request API

This API is invoked by the LSP once the borrower has agreed to the terms of the agreement, given

consent for monitoring and set up a repayment option. This API creates the actual loan in the lender’s

system

Method: **POST**

URL: **/lsp/v1/grantLoan**

Content-Type: **Application/JSON**

## Request Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String | Yes |

## Response Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String | Yes |
| status | Status of loan Application | PROCESSING | OFFERED | GRANTED | REJECTED| ACTION\_REQUIRED | Enum | Yes |

## Request Details

### Sample Request

{

"transactionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb"

}

### Sample Response

{

"transactionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"status": "GRANTED"

}

## Error Codes

### Sample Error Request

{

"code" : "ERROR\_INTERNAL",

"message" : "Internal error.",

"success" : false

}

|  |  |
| --- | --- |
| **ERROR CODE** | **Message** |
| ERROR\_INVALID\_VUA | Invalid vua entered. |
| ERROR\_INTERNAL | Internal error. |

# 

# Loan Summary API

This API is invoked to get the summary of the loan as of the current date.

Method: **POST**

URL: **/lsp/v1/getLoanSummary**

Content-Type: **Application/JSON**

## Request Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String | Yes |

## Response Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String | Yes |
| status | Status of loan application | GRANTED | DEFAULTED| COMPLETED | Enum | Yes |
| createdDate | Date when loan was created |  | Date | Yes |
| startDate | Date from when loan is active |  | Date | Yes |
| endDate | Date till when loan is active |  | Date | Yes |
| summary | Summary details of loan |  | Object | Yes |

**Summary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Type** | **Mandatory** |
| nextRepayment | Details of next payment |  | JSON | Yes |
| principalPaid | Principal paid in the past. Upto 2 digits precision |  | Float | Yes |
| interestPaid | Interest Paid. Upto 2 digits precision. |  | Float | Yes |
| penaltyPaid | Penalty Paid upto 2 digits precise. |  | Float | Yes |
| principalPending | Principal pending in the past. Upto 2 digits precision |  | Float | Yes |
| interestPending |  |  |  |  |
| amountDisbursed |  |  | Float | Yes |
| amountRepaid |  |  | Float | Yes |
| tenure |  |  | Object | Yes |
| description |  |  | String | Yes |

## Request Details

### Sample Request

{

"transactionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb"

}

### Sample Response

{

"transactionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"loanStatus": "GRANTED",

"createdDate": "2019-12-30T13:55:00Z",

"startDate": "2019-12-30T00:00:00Z",

"endDate": "2020-03-01T00:00:00Z",

"summary": {

"nextRepayment": {

"totalAmount": "5000.00",

"principal": "4000.00",

"interestAmount": "1000.00",

"penalty": "0.00",

"fee": "0.00",

"date": "2020-10-10",

"installmentNumber": "1"

},

"principalPaid": "10000.00",

"interestPaid": "100.00",

"penaltyPaid": "10.00",

"principalPending": "10000.00",

"interestPending": "100.00",

"amountDisbursed": "1000.00",

"amountRepaid": "1000.00",

"tenure": {

"duration": "3",

"unit": "MONTH"

},

"description": "short description for the loan"

}

}

## Error Codes

### Sample Error Request

{

"code" : "ERROR\_OTP\_PROVIDER\_DOWN",

"message" : "Otp provider is down. Unable to send message",

"success" : false

}

|  |  |
| --- | --- |
| **ERROR CODE** | **Message** |
| ERROR\_INVALID\_TRANSACTION | Invalid transaction id provided. |
| ERROR\_INTERNAL | Internal Error |

# 

# Loan Statement API

This method is called by the LSP to get the account statement for the loan for a specified period. If the

start and end dates are blank, it will return the entire Loan Statement .

Method**: POST**

URL**: /lsp/v1/getLoanStatement**

Content-Type**: Application/JSON**

## Request Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String | Yes |
| startDate | Start Date from which statement is requested |  | Date | No |
| endDate | End Date till which statement is requested. |  | Date | No |

## Response Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String | Yes |
| accountStatement | Array of transaction details |  | Array | Yes |

**Transaction**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| date | Date when the transaction is carried out |  | Date | Yes |
| narration | Narration as appears in the bank statement |  | String | Yes |
| txnRefNo | Reference number of transaction |  | String | Yes |
| amount | Transaction Amount |  | Float | Yes |
| type | Deposit for repayment and withdrawal for disbursement transactions | DEPOSIT | WITHDRAWAL | ENUM | Yes |
| closingBalance | Closing balance of account after transaction |  | Float | Yes |
| url | Any additional information that has to passed as link |  | String | No |
| extensibleData | For any new data to be added |  | JSON | No |

## Request Details

### Sample Request

{

"transactionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"startDate": "2020-01-01",

"endDate": "2020-12-12",

}

### Sample Response

{

"transactionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"accountStatement" : [

{

"date" : "25-01-2020" ,

"narration" : "Repayment of Loan no : xxxxxxx9090" ,

"txnRefNo" : "004618602124" ,

"amount" : "50000.00" ,

"type" : "DEPOSIT" ,

"closingBalance" : "50000.00"

}

]

}

## Error Codes

### Sample Error Request

{

"code" : "ERROR\_OTP\_PROVIDER\_DOWN",

"message" : "Otp provider is down. Unable to send message",

"success" : false

}

|  |  |
| --- | --- |
| **ERROR CODE** | **Message** |
| ERROR\_INVALID\_TRANSACTION | Invalid transaction id provided. |
| ERROR\_INTERNAL | Internal Error |

# 

# Set Disbursement Plan API

This api tells the lender about the disbursement plan chosen by the borrower.

URL: **/lsp/v1/setDisbursement**

Method **: POST**

Content-type: Application/JSON

## Request Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String(36) | Yes |
| planId | Disbursement Plan Id as sent in offers request |  | String | Yes |

## Response Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String(36) | Yes |
| status | Status of Disbursement plan | Success | boolean | Yes |

## Request Details

### Sample Request

{

"transactionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"planId": "asdfdsrdcdc4bbb4eb1b9e1b4996fbff8acb"

}

### Sample Response

{

"transactionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"status": true

}

## Error Codes

### Sample Error Request

{

"code" : "ERROR\_DISBURSEMENT\_NOT\_VALID",

"message" : "Selected disbursement is invalid",

"success" : false

}

|  |  |
| --- | --- |
| **ERROR CODE** | **Message** |
| ERROR\_DISBURSEMENT\_NOT\_VALID | Selected disbursement not valid |
| ERROR\_INVALID\_DISBURSEMENT\_ID | Invalid disbursement id. |

# Set Disbursement Account

This API is invoked to inform the lender about the disbursement account that the borrower

has chosen.

Method **:POST**

URL: **/lsp/v1/setDisbursementAccount**

Content-type: Application/JSON

## Request Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String(36) | Yes |
| accountId | Disbursement account Id as sent in offers request |  | String | Yes |

## Response Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String(36) | Yes |
| status |  | Success | String | Yes |

## Request Details

### Sample Request

{

"transactionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"accountId": "asdfdsrdcdc4bbb4eb1b9e1b4996fbff8acb"

}

### Sample Response

{

"transactionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"status": "Success"

}

## Error Codes

### Sample Error Request

{

"code" : "ERROR\_INVALID\_TRANSACTION",

"message" : "Invalid transaction id provided.",

"success" : false

}

|  |  |
| --- | --- |
| **ERROR CODE** | **Message** |
| ERROR\_INVALID\_TRANSACTION | Invalid transaction id provided. |
| ERROR\_INTERNAL | Internal Error |
| ERROR\_INVALID\_ACCOUNT\_ID | Invalid account id provided. |

# Trigger Disbursement

API to disburse loan amounts to borrowers.

Method: **POST**

Content-Type: **Application/JSON**

URL: **/lsp/v1/triggerDisbursement**

## Request Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String | Yes |

## Response Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String | Yes |
| payment | Details of payment |  | Object | Yes |

## Payment Details

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| date | Date when the amount was paid by the borrower. |  | Date | Yes |
| principal | Principal amount paid by borrower. |  | float | No |
| interest | Interest paid by borrower |  | folat | No |
| fee | Fee paid by borrower |  | Float | No |
| penalty | Penalty paid by borrower |  | Float | No |
| amount | Amount to be paid by borrower |  | Float | No |
| installmentNumber | The installment which is being currently paid |  | String | Yes |
| status | Status of repayment | SUCCESS | FAILURE | PENDING\_AUTH | Enum | Yes |
| txnRefNo | Receipt number of the transaction |  | String | Yes |

## Request Details

### Sample Request:

{

"transactionId" : "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

}

### Sample response

{

"transactionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"payment" : {

"totalAmount" : "5000.00" ,

"date" : "2020-10-10" ,

"status" : "SUCCESS" ,

"installmentNumber" : "1" ,

"txnRefNo" : "004618602124"

}

}

# Get Offers API

Once the consents needed for fetching data are ready, a request is sent to the Lenders to

generate the offer for a set of loan applications. All these loan applications belong to the same

borrower.

URL: **/lsp/v1/getOffers**

Request: **POST**

Content-Type: **Application/JSON**

## Request Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String(36) | Yes |

## Response Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String(36) | Yes |
| rejectionDetails | Rejection details if any given by lender.l |  | String | No |
| actionRequired | Details of actions requested from the borrower by the lender |  | String | No |
| offers | Offers for the loan application |  | JSON Array | No |
| status | Loan Application Status | PROCESSING | OFFERED | GRANTED | REJECTED| ACTION\_REQUIRED | ENUM | Yes |

## Request Details

### Sample Request

{

"transactionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb"

}

### Sample response

{

"transactionId": "x8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"offers": [{

"id": "y8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"validTill": "2018-12-06T00:00:00.000Z",

"terms": {

"requestedAmount": "50000.00",

"currency": "INR",

"sanctionedAmount": "50000.00",

"interestType": "FIXED",

"interestRate": "12.00",

"interestAmount": "6000.00",

"totalAmount": "56000.00",

"tenure": {

"duration": "3",

"unit": "MONTH"

},

"legalAgreement": {

"type": "TEXT",

"data": "<Base64 Encoded Data>"

},

"charges": {

"prepayment": {

"chargeType": "RATE\_BASED",

"description": "Description of charge calculation”,

"data": {

"rate": "6",

"applicableParameter": "TOTAL\_LOAN\_AMOUNT"

}

},

"bounce": {

"chargeType": "FIXED\_AMOUNT",

"description": "Description of charge calculation",

"data": {

"amount": "3000.00"

}

},

"latePayment": {

"chargeType": "FIXED\_AMOUNT",

"description": "Description of charge calculation",

"data": {

"amount": "3000.00"

}

},

"processing": {

"chargeType": "FIXED\_AMOUNT",

"description": "Description of charge calculation",

"data": {

"amount": "3000.00"

}

}

}

},

"disbursement": {

"plans": [{

"id": "hb8c6822bd4bbb4eb1b9e1b4996fbff8acb",

"automatic": false,

"scheduleType": "ONE\_TIME",

"noOfInstallments": "1",

"totalAmount": "10000.00",

"status": "INACTIVE"

}],

"accountDetails": [{

"id": "hb8c6822bd4bbb4eb1b9e1b4996fbff8acb",

"accountDataType": "ACCOUNT",

"data": {

"accountType": "CURRENT",

"maskedAccountNumber": "XXXXXXXXX9090"

}

}]

},

"repayment": {

"plans": [{

"id": "a8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"automatic": false,

"scheduleType": "ONE\_TIME",

"payNowAllowed": false,

"changeMethodAllowed": false,

"noOfInstallments": "1",

"tenure": {

"duration": "1",

"unit": "MONTH"

},

"title": "Title of the repayment plan",

"shortDescription": "short description of repayment plan ",

"url": "lender url showing repayment details ",

"penalty": "0.00",

"principal": "9000.00",

"interestAmount": "1000.00",

"totalAmount": "10000.00",

"status": "INACTIVE"

}]

}

}],

"loanApplicationStatus": "OFFERED"

}

## Error Codes

### Sample Error Request

{

"code" : "ERROR\_BRE\_PROCESSING\_FAILED",

"message" : "Processing of analysed report failed",

"success" : false

}

|  |  |
| --- | --- |
| **ERROR CODE** | **Message** |
| ERROR\_BRE\_PROCESSING\_FAILED | Processing of analysed report failed |
| ERROR\_OTHER | Internal error. |

# Set Offer API

API which tells the lender about the offer selected by the borrower for that loan application.

URL : **/lsp/v1/setOffer**

Method: **POST**

Content-type: **Application/JSON**

## Request Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String(36) | Yes |
| offerId | Offer Id as passed in get offer response |  | String | yes |

## Response Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String(36) | Yes |
| status | Status of the offer selected | PROCESSING |  Status OFFERED |    OFFER\_ACCEPTED |  GRANTED | REJECTED|  ACTION\_REQUIRED | Enum | yes |

## Request Details

### Sample Request

{

"transactionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"offerId": "y8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

}

### Sample Response

{

"transactionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"status": "OFFERED",

}

## Error Codes

### Sample Error Request

{

"code" : "ERROR\_OFFER\_NOT\_VALID",

"message" : "Selected offer is invalid",

"success" : false

}

|  |  |
| --- | --- |
| **ERROR CODE** | **Message** |
| ERROR\_OFFER\_NOT\_VALID | Selected offer not valid |
| ERROR\_INVALID\_OFFER\_ID | Invalid offer id. |

# Trigger Loan Acceptance API

This API is invoked by the LSP when the user agrees to the terms of the agreement. The Asynchronous response for this API should include the lender sending an OTP to the phone number associated with disbursement account.

SMS Format for the OTP

[#] {6Digit OTP} is the One Time Password (OTP) to complete the loan process with {Lender Name}.

{appToken}

Eg- 467866 is the One Time Password (OTP) to complete the loan process with ABC Bank.

URL: **/lsp/v1/triggerLoanAcceptance**

Method: **POST**

Content-type: **Application/JSON**

## Request Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String(36) | Yes |
| credBlock | Credentials block which contains details for triggering and accepting loans |  | Object | Yes |

## Response Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String(36) | Yes |
| otpSessionKey | Otp Session token for the given request |  | String | Yes |
| maskedPhoneNumber | Masked Phone number to which the otp was sent. |  | String | Yes |

## Request Details

### Sample Request

{

"transactionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"credBlock": {

"type": "OTP",

"data": {

"appToken": "0aCFGTK",

}

}

}

### Sample Response

{

"transactionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"otpSessionKey": "asdfdsrdcdc4bbb4eb1b9e1b4996fbff8acb",

"maskedPhoneNumber": "xxxxxx6622"

}

## Error Codes

### Sample Error Request

{

"code" : "ERROR\_INVALID\_TRANSACTION",

"message" : "Invalid transaction id provided.",

"success" : false

}

|  |  |
| --- | --- |
| **ERROR CODE** | **Message** |
| ERROR\_INVALID\_TRANSACTION | Invalid transaction id provided. |
| ERROR\_INTERNAL | Internal Error |

# 

# Verify Loan Acceptance API

This API is invoked by the LSP when the user enters the OTP received on the mobile number

associated with the disbursement account and agrees to the terms of the agreement. The response for this API should include the lender acknowledging the receipt of OTP and verifying if the user has entered the correct OTP that was shared by the lender.

Method: **POST**

URL: **/lsp/v1/verifyLoanAcceptance**

Content-type: **Application/JSON**

## Request Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String(36) | Yes |
| otpSessionKey | Otp Session token for the given request |  | String | Yes |
| otp | Otp as entered by the borrower |  | String | Yes |

## Response Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String(36) | Yes |
| otpSessionKey | Otp Session token for the given request |  | String | Yes |
| status | Status of authentication | SUCCESS|FAILURE | Enum | Yes |

## Request Details

### Sample Request

{

"transactionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"otpSessionKey": "asdfdsrdcdc4bbb4eb1b9e1b4996fbff8acb",

"otp": "xxxxxx6622"

}

### Sample Response

{

"transactionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"otpSessionKey": "asdfdsrdcdc4bbb4eb1b9e1b4996fbff8acb",

"status": "SUCCESS"

}

## Error Codes

### Sample Error Request

{

"code" : "ERROR\_INVALID\_TRANSACTION",

"message" : "Invalid transaction id provided.",

"success" : false

}

|  |  |
| --- | --- |
| **ERROR CODE** | **Message** |
| ERROR\_INVALID\_TRANSACTION | Invalid transaction id provided. |
| ERROR\_INTERNAL | Internal Error |
| ERROR\_INVALID\_OTP | Invalid otp provided. |
| ERROR\_INVALID\_OTP\_SESSION | Invalid otp session. |

# 

# Repayment APIs

The basic expectations from the repayment APIs are the below

1. When the lender returns an offer, a list of repayment plans should be provided which

include the details about the schedule/when the repayments can be made.

1. The Borrower chooses a repayment plan upon which setRepaymentPlan is

called. The lender returns a payment page url as part of the response to

setRepaymentPlan. This url will be loaded by the LSP inside the app which would show

the kinds of payment methods that the lender supports for the chosen repayment plan.

1. The Borrower can choose a method on the payment page provided by the lender. In

case of BHIM for lending, this will be a one-time e-mandate which the borrower can

initiate by entering his vpa details. Post this, the lender will trigger a request to the PSP

associated with the vpa for getting the authorization from the borrower. Post successful

authorization, e-mandate becomes active and the repayment plan is also considered to

be ACTIVE.

1. When the repayment is to be initiated, by an option other than e-mandate(not for BHIM

for Lending), the user will be taken to the payment page from where he can choose a

payment method and make the payment.

# Set Repayment Plan API

Url: **/lsp/v1/setRepayment**

Method **:POST**

Content-type: **Application/JSON**

## Request Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String(36) | Yes |
| planId | Repayment Plan Id as sent in offers request |  | String | Yes |

## Response Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String(36) | Yes |
| status | Status of repayment plan | Success | String | Yes |

## Request Details

### Sample Request

{

"transactionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"planId": "asdfdsrdcdc4bbb4eb1b9e1b4996fbff8acb"

}

### Sample Response

{

"transactionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"status": "Success"

}

# 

## Error Codes

### Sample Error Request

{

"code" : "ERROR\_REPAYMENT\_PLAN\_NOT\_VALID",

"message" : "Selected repayment plan not valid",

"success" : false

}

|  |  |
| --- | --- |
| **ERROR CODE** | **Message** |
| ERROR\_REPAYMENT\_PLAN\_NOT\_VALID | Selected repayment plan not valid |
| ERROR\_EMANDATE\_AUTHORIZATION | E-Mandate Authorization failed.. |

# 

# Trigger Repayment

This api is used for triggering repayment

URL: **/lsp/v1/triggerRepayment/{transactionId}**

Method: **POST**

Content-type: **Application/JSON**

## Request Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String(36) | Yes |

## Response Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String(36) | Yes |
| payment | Details of repayment |  | Object | Yes |

## Request Details

### Sample Response

{

"transactionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"Payment": {

"id": "hhcc6822bd4bbb4eb1b9e1b4996fbff8acb",

"status": "PENDING\_AUTH",

"paymentUrl": "https://acmbank.in/swgjdeikdk"

}

}

## Error Codes

### Sample Error Request

{

"code" : "ERROR\_INVALID\_TRANSACTION",

"message" : "Invalid transaction id provided.",

"success" : false

}

|  |  |
| --- | --- |
| **ERROR CODE** | **Message** |
| ERROR\_INVALID\_TRANSACTION | Invalid transaction id provided. |
| ERROR\_INTERNAL | Internal Error |

# 

# Repayment Status API

This api provides LSP with information on the status of the repayment done by borrower to lender.

Url: **/lsp/v1/repaymentStatus**

Method: **POST**

Content-Type: **Application/JSON**

## Request Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String(36) | Yes |
| paymentId | Unique Id as passed in trigger repayment |  | String | Yes |

## Response Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String(36) | Yes |
| payment | Details of repayment |  | Object | Yes |

## 

## Request Details

### Sample Request

{

"transactionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"paymentId": "hhcc6822bd4bbb4eb1b9e1b4996fbff8acb"

}

### Sample Response

{

"transactionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"Payment": {

"id": "hhcc6822bd4bbb4eb1b9e1b4996fbff8acb"

"totalAmount": "500000"

"date": "2020-01-01"

"status": "Success"

"installmentNumber": "12"

"txnRefNo": "17334222"

}

}

## Error Codes

### Sample Error Request

{

"code" : "ERROR\_INVALID\_TRANSACTION",

"message" : "Invalid transaction id provided.",

"success" : false

}

|  |  |
| --- | --- |
| **ERROR CODE** | **Message** |
| ERROR\_INVALID\_TRANSACTION | Invalid transaction id provided. |
| ERROR\_INTERNAL | Internal Error |
| ERROR\_INVALID\_PAYMENT\_ID | Payment Id provided is invalid or expired. |

# 

# Download Analysed Report API

API exposed by **Perfios** for lenders to download **Perfios** analysed report of both Bank & GST

## Request Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String(36) | Yes |

## Request Details:

URL: **/lsp/v1/downloadReport/{transactionId}**

METHOD: **GET**

### Sample Response:

Zip file containing the analysed report. (Bank,GST,Invoices)

Report type for Bank and gst will be as per customized requirements.

Report name: **{transactionId}.zip**

Bank analysed report name: **{transactionId\_Bank}.xml/json**

Gst Data: **{transactionId}.xml/json**

Invoices: **{transactionId}.xml/json**.

# Transaction Callback API

This API will be used by perfios to tell the lender on the status of the analysed report.

URL: **/lsp/v1/transactionCallback**

Method: **POST**

Content-type: **Application/JSON**

## Request Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String(36) | Yes |
| status | Status of Analysed Report |  | String | Yes |
| message | Reason for failure if any |  | String | Yes |

## Response Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String(36) | Yes |
| success |  | true|false | boolean | Yes |

## 

## Request Details

### Sample request

{

"transactionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"status": "COMPLETED",

"message": "Analysed report is ready to be downloaded."

}

### Sample Response

{

"transactionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"success": "true"

}

## Error Codes

### Sample Error Request

{

"code" : "ERROR\_INVALID\_TRANSACTION",

"message" : "Invalid transaction id provided.",

"success" : false

}

|  |  |
| --- | --- |
| **ERROR CODE** | **Message** |
| ERROR\_INVALID\_TRANSACTION | Invalid transaction id provided. |
| ERROR\_INTERNAL | Internal Error |

# 

# Offer Status Callback API

This callback will be sent by lender to perfios once they have consumed perfios analysed report and generated offers for the loan application.

URL: **/lsp/v1/offerStatusCallback**

Request: **POST**

Content-Type: **Application/JSON**

## Request Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String(36) | Yes |
| success | Flag to suggest if offers are generated | true|false | boolean | Yes |
| message | Relevant message |  | String | Yes |

## Response Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional/perfios-aa/logs/perfios-aa-logs/perfios-aa.2020-02-28.log** |
| success | Flag to suggest if offers are generated | true|false | boolean | Yes |
| message | Relevant message |  | String | Yes |

## Request Details

### Sample Request

{

"transactionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"success": "true",

"message": "Offers have been generated."

}

### Sample Request

{

"success": "true",

"message": "Offers have been generated."

}

## Error Codes

### Sample Error Request

{

"code" : "ERROR\_INVALID\_TRANSACTION",

"message" : "Invalid transaction id provided.",

"success" : false

}

|  |  |
| --- | --- |
| **ERROR CODE** | **Message** |
| ERROR\_INVALID\_TRANSACTION | Invalid transaction id provided. |
| ERROR\_INTERNAL | Internal Error |

# 

# Loan Status API

This API is invoked to check the status of loan application.

URL: **/lsp/v1/getLoanStatus**

Request: **POST**

Content-Type: **Application/JSON**

## Request Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String(36) | Yes |

## Response Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String(36) | Yes |
| status | GRANTED |DEFAULTED COMPLETED |  | ENUM | Yes |
| startDate | Date from when loan was active |  | DATE | Yes |
| endDate | Date till when loan is active |  | DATE | Yes |

## Request Details:

### Sample Request

{

"transactionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb"

}

### Sample Response

{

"transactionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"status": "GRANTED",

"startDate": "2019-12-30T00:00:00Z",

"endDate": "2020-12-30T00:00:00Z"

}

## Error Codes

### Sample Error Request

{

"code" : "ERROR\_INVALID\_TRANSACTION",

"message" : "Invalid transaction id provided.",

"success" : false

}

|  |  |
| --- | --- |
| **ERROR CODE** | **Message** |
| ERROR\_INVALID\_TRANSACTION | Invalid transaction id provided. |
| ERROR\_INTERNAL | Internal Error |

# 

# Raise Dispute

This API is invoked by the LSP to create a dispute in the lender system for an issue that is faced by the

borrower.

Method: **POST**

URL: **/lsp/v1/raiseDispute**

Content-type**: Application/JSON**

## Request Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String | Yes |
| complaintCode | Complaint Code | AMOUNT\_NOT\_DISBURSED | REPAYMENT\_FAIURES | OTHERS | Enum | Yes |
| message | Relevant message for dispute |  | String | Yes |

## Response Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String | Yes |
| status | Status of dispute raised | NEW | PROCESSING | NEED\_MORE\_ INFO | RESOLVED | ENUM | Yes |
| dispute | Dispute Details |  | Object | Yes |

## Request Details

### Sample Request

{

"transactionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"complaintCode": "AMOUNT\_NOT\_DISBURSED",

"message": "Amount not disbursed to the account yet"

}

### Sample Response

{

"transactionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"status": "RESOLVED",

"dispute": {

“id”: “asdasdasdagvtrdvre”,

“status”: “NEED\_MORE\_INFO”,

“url”: ”URL where dispute can be tracked”

}

}

## Error Codes

### Sample Error Request

{

"code" : "ERROR\_INVALID\_TRANSACTION",

"message" : "Invalid transaction id provided.",

"success" : false

}

|  |  |
| --- | --- |
| **ERROR CODE** | **Message** |
| ERROR\_INVALID\_TRANSACTION | Invalid transaction id provided. |
| ERROR\_INTERNAL | Internal Error |

# 

# Dispute Status

This API is invoked to check the status of the dispute.

Method: **GET**

URL: **/lsp/v1/disputeStatus**

Content-type:**Application/Json**

## Request Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String | Yes |
| disputeId | Dispute Id as sent in raise request |  | String | Yes |

## Response Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Sample Value** | **Data Type** | **Mandatory/Optional** |
| transactionId | Unique identifier for loan application |  | String | Yes |
| status | Status of dispute raised | NEW | PROCESSING | NEED\_MORE\_ INFO | RESOLVED | ENUM | Yes |
| message |  |  | String | Yes |

## Request Details

Method: **GET**

URL: **/lsp/v1/disputeStatus**

### Sample Request

{

"transactionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"dispute": {

“id”: “asdasdasdagvtrdvre”

}

}

### Sample Response

{

"transactionId": "e8cc6822bd4bbb4eb1b9e1b4996fbff8acb",

"status": "AMOUNT\_NOT\_DISBURSED",

"message": "Amount not disbursed to the account yet"

}

## Error Codes

All error codes must have a status code of **400**

### Sample Error Request

{

"code" : "ERROR\_INVALID\_TRANSACTION",

"message" : "Invalid transaction id provided.",

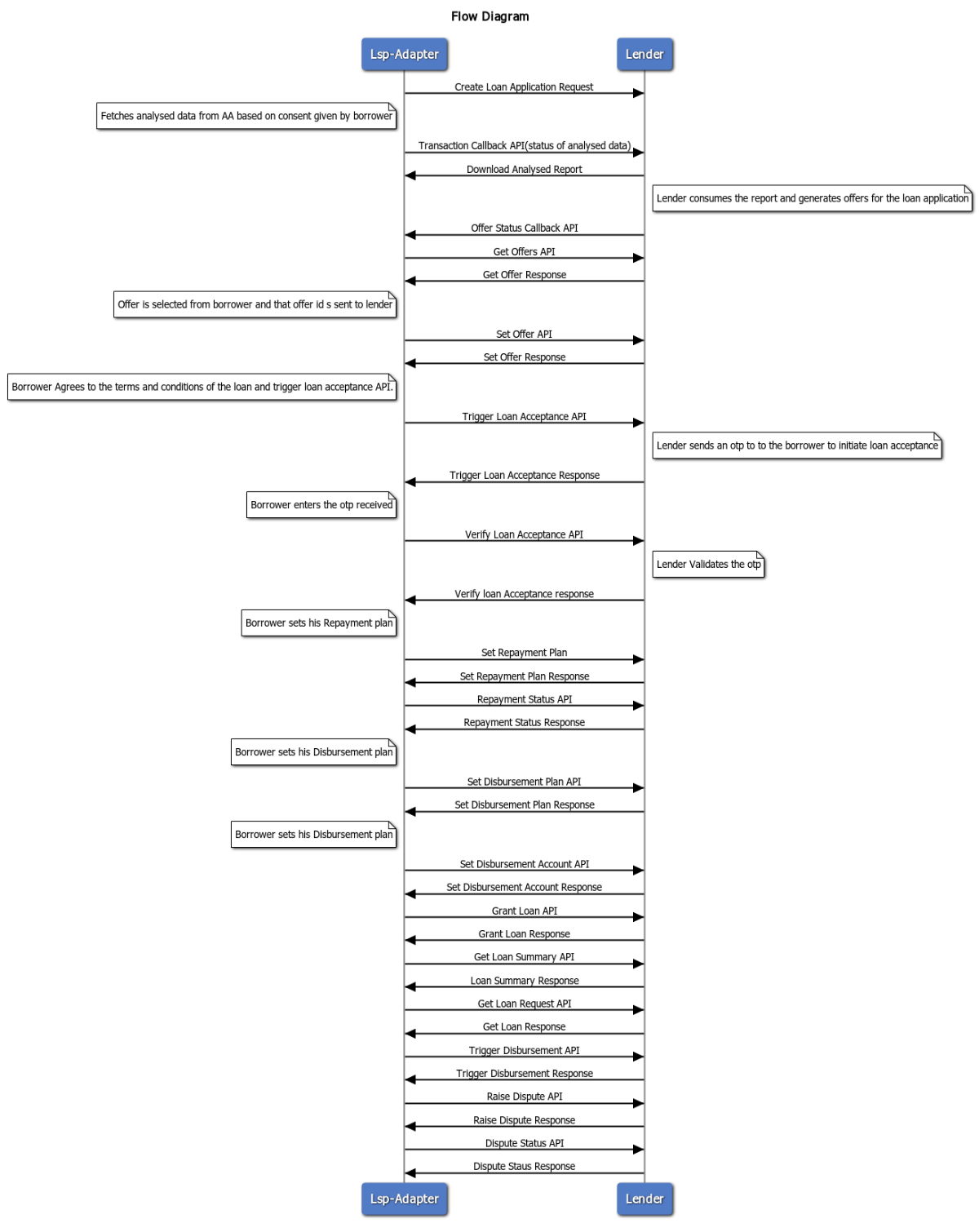
"success" : false

}

|  |  |
| --- | --- |
| **ERROR CODE** | **Message** |
| ERROR\_INVALID\_TRANSACTION | Invalid transaction id provided. |
| ERROR\_INTERNAL | Internal Error |
| ERROR\_INVALID\_DISPUTE\_ID | Invalid dispute id provided. |

# 

# Flow Diagram



# Mapping of flow

|  |  |  |  |
| --- | --- | --- | --- |
| **Request** | **Sahay API** | **Perfios LSP Bridge** | **Lender API** |
| Request sent to Perfios LSP Bridge with GSTN Profile of the applicant and eligible invoices. | Create loan Request | Perfios passes the loan request to the lender in order to initiate a loan request. | The lender accepts the request and sends an acknowledgement. |
| Create loan Response | Perfios Sends the acknowledgement back to Sahay. |
| Sahay sends a consent request to adapter to fetch data | Consent Handle Request | Sends consent request to Perfios FIU. FIU fetches data from all relevant FIP's | No action expected from Lender. |
| Consent Handle Response | Perfios sends the consent handle ids back to Sahay. |
| Sahay triggers the status of consent. | Consent Handle Status Request | Perfios LSP Bridge sends consent status acknowledgement to Sahay. | No action expected from Lender. |
| Consent Handle Status Response | Perfios LSP Bridge sends a response of consent status |
|  | Perfios FIU initiates data fetch from user accepted bank accounts (FIP) and GSTN. |
| Perfios FIU downloads data and passes it to the analysis engine. |
| Analysed data is downloaded from Perfios FIU. |
| Adapter sends Lender a notification that data is ready for download. | Lender Downloads the analysed data of Bank & GSTN data for the loan applications to generate offers. The lender sends a notification to Perfios LSP Bridge indicating offers are ready. |
| Sahay send a request asking for offers for a given loan application | Get Offer Request | Perfios LSP Bridge downloads the offers for the loan application | The lender provides the offers to Perfios LSP Bridge. |
| Get Offer Response | Perfios LSP Bridge sends the offers to Sahay |  |
| Sahay sends the selected offer to Adapter | Set Offer Request | Perfios LSP Bridge validates the offer and sends it to Lender for tracking. | Lender Acknowledges the offer selected. |
| Set Offer Response | Perfios LSP Bridge sends an acknowledgement to Sahay for the offer selected. |
| Applicant Agrees to the terms and conditions of the loan and trigger loan acceptance | Trigger Loan Acceptance Request | Perfios LSP Bridge sends a loan acceptance request to the lender. | The lender sends an OTP to the applicant who accepted the loan and sends an acknowledgment to Perfios |
| Trigger Loan Acceptance Response | Perfios LSP Bridge sends the acknowledgement back to Sahay |
| Sahay sends the OTP entered by the applicant to period for validation | Verify Loan Acceptance Request | Perfios LSP Bridge sends the OTP to the lender for validation. | The lender verifies the OTP. |
| Verify Loan Acceptance Response | Perfios LSP Bridge sends verification status to Sahay. |
| Sahay sends a get repayment request for a loan application | Get Repayment Plan Request | Perfios LSP Bridge validates the request and sends the list of repayment plans for the loan application | No action expected from Lender. |
| Get Repayment Plan Response | Perfios LSP Bridge sends the repayment plan to Sahay. |
| Sahay sends the repayment plan id chosen by the applicant | Set Repayment Plan | Perfios LSP Bridge forwards the plan id to the lender for tracking | Lender Acknowledges the plan id sent by Perfios LSP Bridge. |
| Set Repayment Plan Response | Perfios LSP Bridge sends a lender an acknowledgement for the plan id selected |
| Credit Monitoring Consent |  | Perfios LSP bridge initiates Credit monitoring consent to Perfios FIU, which would be accepted by the applicant. | No action expected from Lender |
| Upon applicant agreeing to the terms of the loan, given  consent for monitoring and set up a repayment option. This API creates the actual loan in the lender’s  system. | Grant Loan Request | Perfios LSP Bridge validates the request and forwards a grant loan request to the lender. | The lender accepts the loan and acknowledges it. |
| Grant Loan Response | Perfios LSP Bridge sends a response to Sahay on status of grant loan acceptance |
| Sahay Sends a request to get disbursement options for a loan application | Get Disbursement Request | Perfios LSP Bridge acknowledges the request | No action expected from Lender. |
| Get Disbursement Response | Perfios LSP Bridge sends all disbursement plans for that loan application |
| Sahay sends the disbursement plan set by the applicant | Set Disbursement Plan Request | Perfios LSP Bridge forwards the plan id for disbursement to the lender for tracking | The lender receives the plan id and sends an acknowledge to Perfios LSP Bridge |
| Set Disbursement Plan Response | Perfios LSP Bridge sends an acknowledgement back to Sahay |
| Sahay sends request to get all disbursement accounts for a loan application | Set Disbursement Account Request | Perfios LSP Bridge forwards the disbursement account set by applicant to lender for tracking | The lender set the account and sends an acknowledgement to Perfios LSP Bridge |
| Set Disbursement Account Response | Perfios LSP Bridge sends an acknowledgement back to Sahay |
| Sahay trigger a disbursement request | Trigger disbursement Request | Perfios LSP Bridge sends a trigger request to the lender | The lender accepts the request and disburses the money to the said account. |
| Trigger Disbursement Response | Perfios LSP Bridge sends a notification to Sahay on disbursement status | Lender acknowledges the amount has been disbursed |
| Sahay trigger a repayment request for loan application | Trigger Repayment Request | Perfios LSP Bridge informs lender about the repayment request | The lender checks and sends the status of repayment to Perfios LSP Bridge |
| Trigger Repayment Response | Perfios LSP Bridge sends repayment info to Sahay |
| Sahay sends a repayment status check | Trigger Repayment Status Request | Perfios LSP Bridge notifies lender for a status request | NA |
| Trigger Status Repayment Response | Perfios LSP Bridge sends repayment status to Sahay |
| Applicant raises a complaint about dispute management. | Raise dispute | Perfios LSP Bridge forwards the complaint to the lender | Lender records the dispute by applicant. |
| Raise Dispute Response | Perfios LSP Bridge sends an acknowledgement back to Sahay |
| Applicant updates the dispute | Update Dispute Request | Perfios LSP Bridge sends an updated complaint to the lender | Lender updates the dispute by applicant. |
| Update Dispute Response | Perfios LSP Bridge sends an acknowledgement back to Sahay |

# Mapping of API

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Url** | **Async/Sync** | **Invoked By** | **Alternative mapping** |
| Create Loan Application | **/lsp/v1/createLoanApplication** | Sync | Perfios |  |
| Transaction Callback | **/lsp/v1/transactionCallback** | Sync | Perfios |  |
| Download Analysed Report | **/lsp/v1/downloadReport/{transactionId}** | Sync | Client |  |
| Offer Status Callback API | **/lsp/v1/offerStatusCallback** | Sync | Client |  |
| Get Offer API | **/lsp/v1/getOffers** | Sync | Perfios. Will be triggered only after the offer status callback is sent by the lender. |  |
| Set Offer API | **/lsp/v1/setOffer** | Sync | Perfios |  |
| Trigger Loan Acceptance | **/lsp/v1/triggerLoanAcceptance** | Sync | Perfios |  |
| Verify Loan Acceptance | **/lsp/v1/verifyLoanAcceptance** | Sync | Perfios |  |
| Set Repayment | **/lsp/v1/setRepayment** | Sync | Perfios |  |
| Repayment Status | **/lsp/v1/repaymentStatus** | Sync | Perfios |  |
| Grant Loan | **/lsp/v1/grantLoan** | Sync | Perfios |  |
| Set Disbursement Plan | **/lsp/v1/setDisbursement** | Sync | Perfios |  |
| Set Disbursement Account | **/lsp/v1/setDisbursementAccount** | Sync | Perfios |  |
| Trigger Disbursement API | **/lsp/v1/triggerDisbursement** | Sync | Perfios |  |
| Trigger Repayment | **/lsp/v1/triggerRepayment** | Sync | Perfios |  |
| Raise Dispute | **/lsp/v1/raiseDispute** | Sync | Perfios |  |
| Dispute Status | **/lsp/v1/disputeStatus** | Sync | Perfios |  |
| Loan Status | **/lsp/v1/getLoanStatus** | Sync | Perfios |  |
| Loan Summary API | **/lsp/v1/getLoanSummary** | Sync | Perfios |  |
| Loan Statement API | **/lsp/v1/getLoanStatement** | Sync | Perfios |  |
| Trigger OTP | **/lsp/v1/triggerOtp** | Sync | Perfios |  |
| Verify OTP | **/lsp/v1/verifyOtp** | Sync | Perfios |  |

# User Journey & API Sequence

