

Developer manual

FD Advice

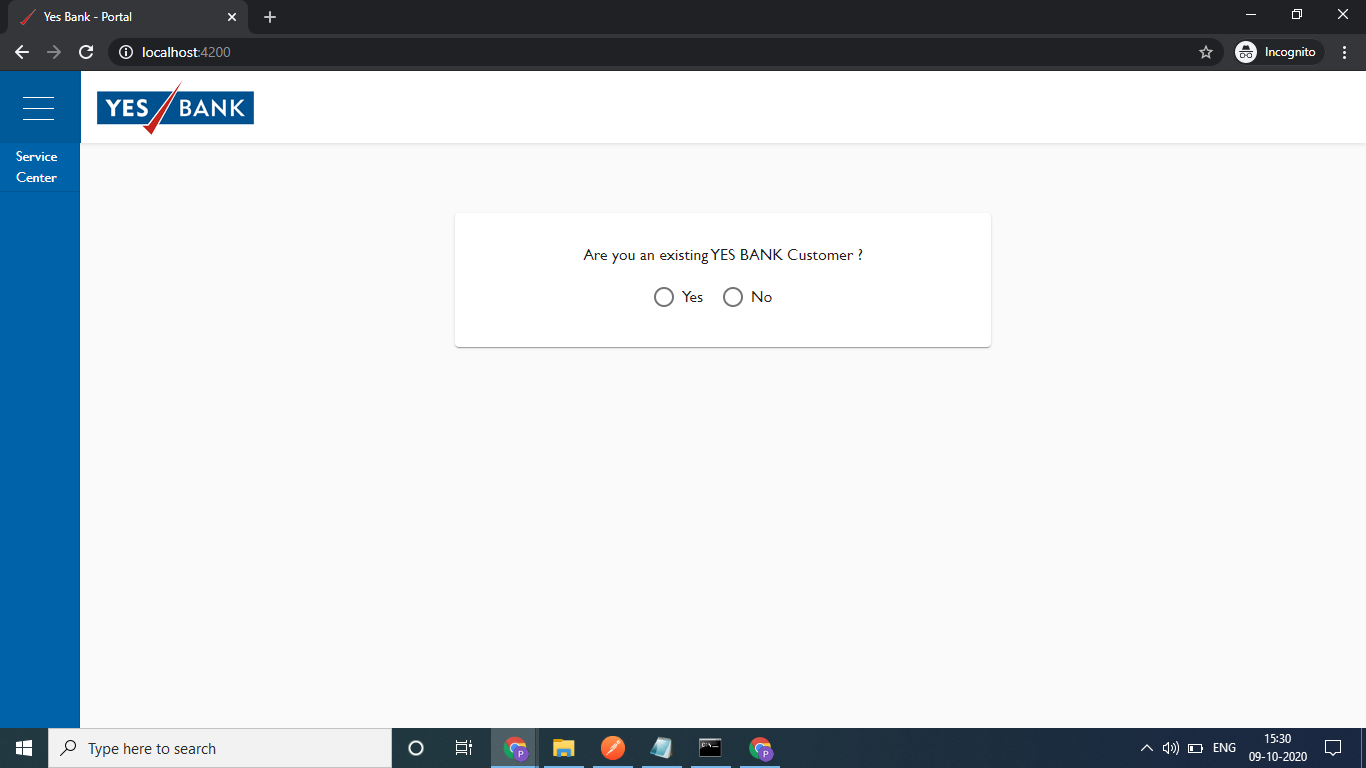


Octomber 9, 2020

Yes bank

[Company address]

1) First Screen : Create Session



**Api called :** Create Session

Service Name :- Create Session

Project :- YBLIGNITE

Module :- Session

Workflow :- Create Session

Process Name :- Wrapper :: Create Or Renew Session

Nodes: -

1. Generate Session ID: - In this call Activity we call the Create Or Renew Session Id process to generate the session Id.
2. Create Or Renew Session Id: - In this script node we generate the session Id and return it to Generate Sesseion Id call Activity.

Audit Log SR :



Create Session:

Response:

{

"payload": {

"auth": {

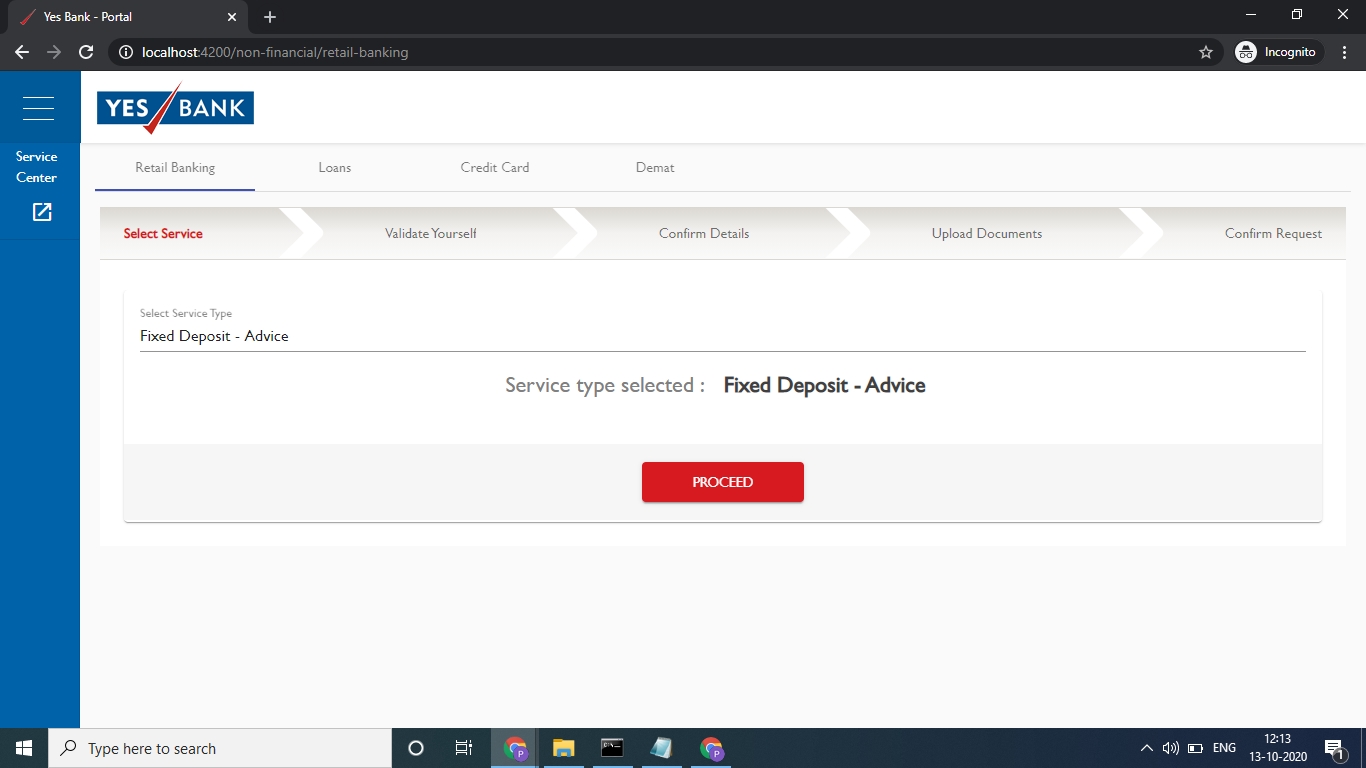
"sessionId": "2bcfbee1b41abff097667203537bac8b80f7b5de9520552a40a28bae65b0b4"

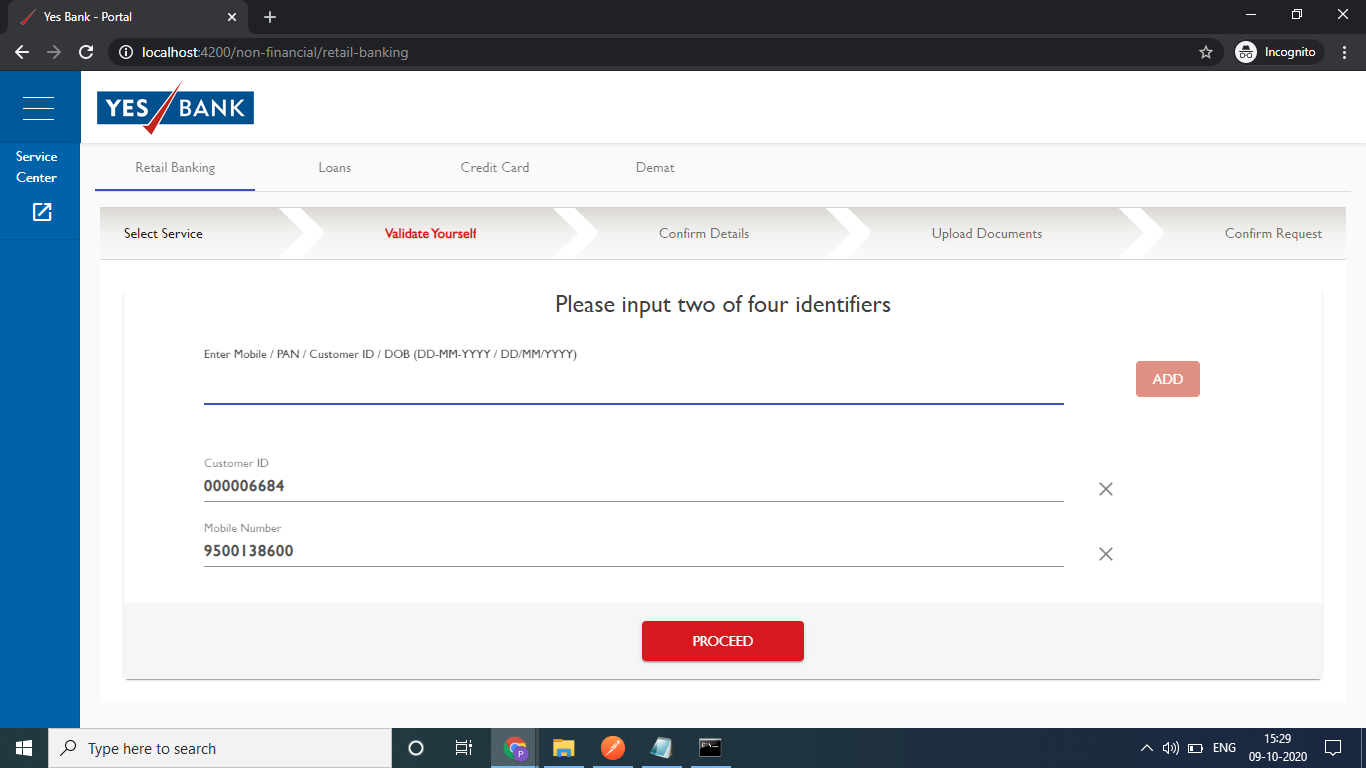
}

},

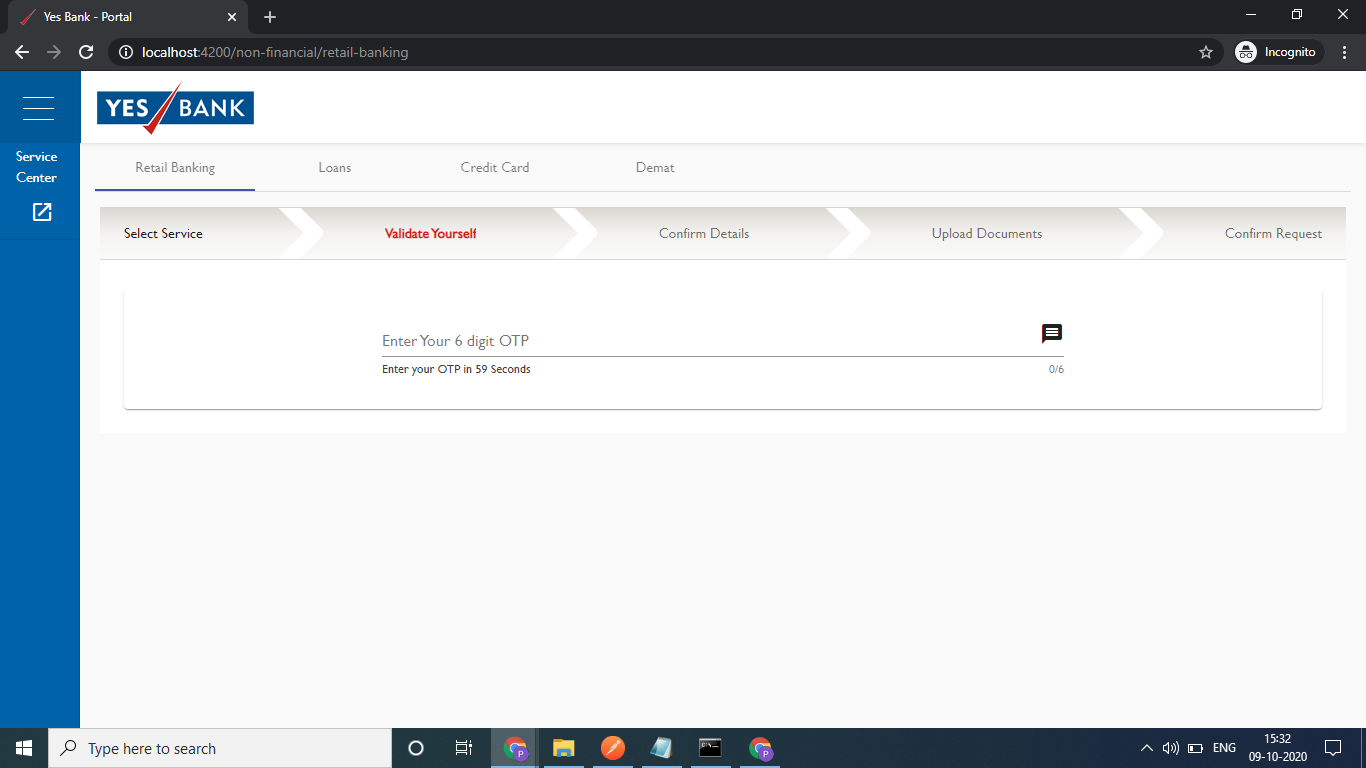
"status": true

}

2) 

3)

4)



**Api call:**

SSP Identify customer :

Service Name :- Validate Customer

Project :- YBLIGNITE

Module :- SSP

Workflow :- SSP:: Identify Customer

Process Name :- SSP :: Identify Customer

Nodes: -

1. Parse Input: - In this Script node we validate the input if it is not valid then we will show the error message and flow will be end.
2. Find If Customer Is Blocked: - In this call activity we check the customer is blocked or not by passing custId.
3. Construct Response Msg: - This script node is use to construct error message if user is blocked.
4. Find If Is An Existing Session: - In this db node we check the session is existing or not.
5. Validate Details: - In this script node we check the retry count.
6. Find Customer (MDM Search) - Capture RowId: - In this call activity we get the rowId of the customer from MDM search.
7. Parse API Response : - In this script node we parse the response that we get from MDM search then we will check the response if customer is identify successfully then flow will continue if not then flow will end.
8. Find If Customer Is Blocked: - In this call activity we check the customer is blocked or not if it is blocked then end the flow otherwise continue the flow.
9. MDM Details Wrapper API -Capture Additional Details: - In this call activity if we want to fetch additional data then we use this call activity we send refId,rowId and srType as input and we get the additional details.
10. Construct DB Query:- In this script node we are construct the query for update the accountNumber,pan etc in ss\_session and ss\_session\_masked table.
11. Execute DB Query : - This db node is use for execute the query for ss\_session.
12. Execute Masked Query : - This db node is use for execute the query for ss\_session\_masked.
13. Initiate OTP : - In this call activity we use for initiate otp we send custId,message as input.
14. Parse API Response: - In this script node we parse the response that we get from initiate OTP call activity and set errorcode and errorResponse and End the flow.

Request:

{

"projectId": "ff0ae4a6884711e9b16676fb2f2488b6",

"workflowId": "cc847fb2c95e11e9933d8e7a151d5229",

"processId": "dbba74dcc95e11e9933d8e7a151d5229",

"ProcessVariables": {

"custId": "000006651",

"mobileNo": "9500138600",

"sessionId": "2bcfbee1b41abff097667203537bac8b80f7b5de9520552a40a28bae65b0b4",

"srType": 1016

}

}

Response:

{

"payload": {

"processResponse": {

"ProcessVariables": {

"mdmCashWithdrawal": "",

"mdmPoliticalExposure": "",

"mdmIncome": "",

"mdmCashDeposit": "",

"isValidRelation": false,

"custId": "000006651",

"mdmFunds": "",

"mdmResidenceType": "",

"mdmCustDetails": {

"firstName": null,

"lastName": null,

"dob": null,

"entityType": null,

"activationDt": null,

"fullName": "000006651",

"fcrId": null,

"srcUcic": null,

"mdmId": null

},

"maskedMobileNo": "\*\*\*\*\*\*8600",

"cId": 6651,

"mdmQualification": "",

"panResponse": null,

"fcrResponse": null,

"identifiers": "Mobile = \*\*\*\*\*\*8600; Customer Id = \*\*\*\*\*6651; ",

"mdmAcctNo": "",

"vLogs": [

{

"typeTag": "Mongo",

"moduleName": "Module 1",

"description": "sample desc",

"type": 3,

"versionNumber": "v1.0",

"timestamp": "2020-01-20T13:59:46Z"

},

{

"typeTag": "App",

"moduleName": "Module 1",

"description": "sample desc",

"type": 2,

"versionNumber": "v1.0",

"timestamp": "2020-01-20T13:59:46Z"

},

{

"typeTag": "Web",

"moduleName": "Module 1",

"description": "sample desc",

"type": 1,

"versionNumber": "v1.0",

"timestamp": "2020-01-20T13:59:46Z"

}

],

"mdmMaritalStatus": "",

"mdmContactDetails": {

"mobile": "9500138600",

"landline": "022-44449956",

"fax": null,

"email": "rajesh.sadhu@appiyo.com"

},

"sessionId": "2bcfbee1b41abff097667203537bac8b80f7b5de9520552a40a28bae65b0b4",

"env": "0",

"mdmOutwardRemittance": "",

"otpKey": "abcdef",

"mdmTransactions": "",

"mdmInwardRemittance": "",

"isActiveAccount": false,

"mdmAadhar": "",

"pvHttpRequestHeaders": "{\"x-request-id\":\"576978b516e417fb27bb9a5eab0e0f8d\",\"content-length\":\"552\",\"referer\":\"http:\\/\\/localhost:4200\\/\",\"requestedIp\":\"171.51.213.103\",\"accept-language\":\"en-US,en;q=0.9\",\"x-auth-sessionid\":\"2bcfbee1b41abff097667203537bac8b80f7b5de9520552a40a28bae65b0b4\",\"origin\":\"http:\\/\\/localhost:4200\",\"x-appiyo-ts\":\"2020-10-13T06:47:37.047Z\",\"x-appiyo-hash\":\"3784e559c4e173d341c86c0b979797ccace80b26691ba1bbc26110eaa5e30680\",\"x-forwarded-for\":\"171.51.213.103, 172.17.0.1\",\"accept\":\"application\\/json, text\\/plain, \*\\/\*\",\"x-real-ip\":\"172.17.0.1\",\"host\":\"128.199.227.253\",\"connection\":\"close\",\"content-type\":\"text\\/html\",\"x-appiyo-key\":\"RVU3N3FFTWdpR1ppQUpzVm5rOUpSWFk4NU5LaVROWTRySGYwRngyYTgxNGlVYXI2UmNPOGlHR3FKeTY4NEVUWHBHY3c1MnVqKy9VYkExdnNCejhnREs1aFZRakdUTEdvRVFqZ1pESGExY0F6dlNYUUYvaEJMQS9QWE9tb2RxcmxlbHZTdDdRS01pTU13eDIwZlk1M3dHeHd4VHl2VHNRN21DU1haVnJhcnNNenNEZE9jM1FNdHRqbDllcDBtaTdPYUpPTVNrZXhVRDNBTDZYL2QxVTJzK3I4RVd2R2ZTS3pNMy9WSk1KNEdxQjl5aUdaYkpjSEZmbzFyK3oxV3BXNXRoUlY5cjJ3WENaVEs4L0UyT0syY1Npd1E0T050dHh0cmR2MFpENkhZaEtTMzF2TkhuZktXZVIwbFk5dVZhQnVYYllXL2gxNVdqcU5VY1c3Wng4emh6c2xmdzlseGM2aA==\",\"accept-encoding\":\"gzip, deflate\",\"user-agent\":\"Mozilla\\/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit\\/537.36 (KHTML, like Gecko) Chrome\\/86.0.4240.75 Safari\\/537.36\"}",

"fcrPan": ""

},

"Status": "Execution Completed",

"ProcessName": "SSP :: Identify Customer",

"Error": "0",

"ErrorCode": "200",

"ProcessId": "dbba74dcc95e11e9933d8e7a151d5229",

"ApplicationId": "ff0ae4a6884711e9b16676fb2f2488b6",

"ErrorMessage": "Execution successfull, initiated send an otp",

"ProcessInstanceId": "fa1643420d1f11ebac940242ac110003",

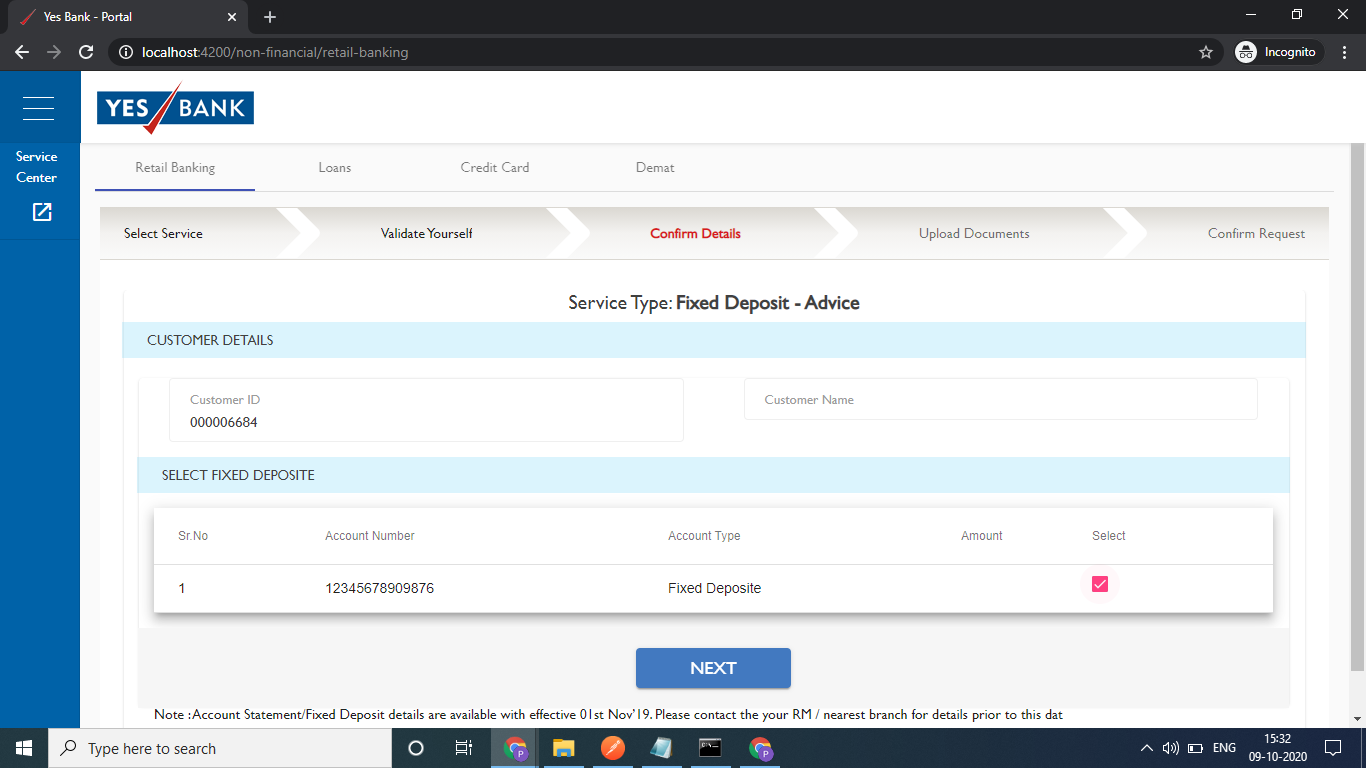
"WorkflowId": "cc847fb2c95e11e9933d8e7a151d5229"

}

},

"status": true

}

5)

**Api called :**

1. Ssp auth validate:

Service Name :- Validate Customer

Project :- YBLIGNITE

Module :- SSP

Workflow :- SSP:: Authenticate Customer

Process Name :- SSP :: Auth Validate

Nodes: -

1. Valid Input: - In this script node we validate otpkey,otpValue,sessionId if it is not valid then we directly end the flow.
2. Validate Session: - In this db node we check the current session is valid session or not.
3. Validate Session Details: - In this script node we get the value of max retry and assign to variable.
4. Find If Customer Is Blocked: - In this Call Activity we will check the customer is blocked or not if it is blocked then we show the error message or else flow will continue.
5. Authorize Customer: - In this call activity we check the customer is authorize or not.
6. Parse Response: - In this script node we get response from Authorize Customer call activity and we check the otp and response is 200 or not if not valid response then we show error message and end the flow.
7. Increment Retry Count: - In this db node we update the incremented retry count in ss\_session table.
8. Check If Reached Max Limit: - In this db node we execute query to get the max number of customer retry.If he not reached the max limit then we end the flow or if he reached the max limit of retry then we blocked that customer.
9. Mark The Customer As Blocked For Current Date: - This call activity is use to block the Customer.
10. Set Error Code: - In this script node we construct the block message and the end the flow.

Request

{

"projectId": "ff0ae4a6884711e9b16676fb2f2488b6",

"workflowId": "0d95fa3ac95f11e9933d8e7a151d5229",

"processId": "2b184018c95f11e9933d8e7a151d5229",

"ProcessVariables": {

"sessionId": "2bcfbee1b41abff097667203537bac8b80f7b5de9520552a40a28bae65b0b4",

"otpKey": "abcdef",

"otpValue": "222222"

}

}

Response:

{

"payload": {

"processResponse": {

"ProcessVariables": {

"isAuthValidated": true,

"custId": "000006651"

},

"Status": "Execution Completed",

"ProcessName": "SSP :: Auth Validate",

"authentication-token": "0JjKdDQSUIbxzrfUzYhgS8ux6I3KIOZCehMHj1OnFX0fXZJCEMLuKFgxM9RtZPcl",

"Error": "0",

"ErrorCode": "200",

"ProcessId": "2b184018c95f11e9933d8e7a151d5229",

"ApplicationId": "ff0ae4a6884711e9b16676fb2f2488b6",

"ErrorMessage": "Execution successfull, valid otp",

"ProcessInstanceId": "226fdb000d2011eb818b0242ac110003",

"WorkflowId": "0d95fa3ac95f11e9933d8e7a151d5229"

}

},

"status": true

}

b) GET SESSION DETAILS

Service Name :- Get Session Details

Project :- YBLIGNITE

Module :- SSP

Workflow :- SSP:: Get Session Details

Process Name :- SSP :: Get Session Details

Nodes: -

1. Validate Input: - In this script node we are construct query to fetch customer details like custId,pan,dob using sessionId.
2. Fetch Session Details: - In this db node we are execute query to fetch customer details like custId,pan,dob using sessionId.
3. Validate Session: - in this script note we validate the custId.
4. Get Session Details (Level 1): - In this call activity we call Get Session Details level 1 process and pass custId,sessionId, srType to perform further operation.
5. Get Account Details For FD : - In this call activity we call the Get Session Details FD Statement process custId,sessionId, srType to perform further operation.
6. Get FCR Account details for fd: - In this call activity we call the FCR get All Account details process and pass custId, sessionId and accountType Flag then we get the account details.
7. Parse response : - In this script node we parse the response that we get from FCR process.

**Request :**

{

"projectId": "ff0ae4a6884711e9b16676fb2f2488b6",

"workflowId": "2dcb7546b14411eaa34acea256932e5c",

"processId": "5995bfc0cf1711e9933d8e7a151d5229",

"ProcessVariables": {

"sessionId": "e1701fb12640c8b1dca19bb798801041b090a53bbc8d77b2863795a3a59f69b",

"srType": "1016"

}

}

**Response :**

{

"ApplicationId": "ff0ae4a6884711e9b16676fb2f2488b6",

"Error": "0",

"ErrorCode": "200",

"ErrorMessage": "",

"ProcessId": "5995bfc0cf1711e9933d8e7a151d5229",

"ProcessInstanceId": "b250b0160a1211eb8e1e0242ac110003",

"ProcessName": "SSP :: Get Session Details",

"ProcessVariables": {

"addressData": null,

"annualIncome": null,

"cKycDt": "",

"casaAccountDetails": null,

"countryData": null,

"custId": "000006651",

"custName": "",

"custType": "",

"emailUpdate": {

"maskedOldEmail": null,

"oldEmail": "rajesh.sadhu@appiyo.com"

},

"fdAccountDetails": [

{

"accountNumber": "12345678909876",

"accountType": "Fixed Deposite",

"balance": null

}

],

"guardianRelationData": null,

"interestInstructions": null,

"isAuthorizationRequired": false,

"isCkycLatest": false,

"isExpiredPoa": false,

"isExpiredPoi": false,

"isPanRequired": false,

"isValidPoa": false,

"isValidPoi": false,

"landLineUpdate": {

"maskedLandLineNumber": null,

"oldNumber": "022-44449956"

},

"maritalStatus": "",

"maskedCustId": "\*\*\*\*\*\*651",

"maskedEmailId": "",

"mdmEmail": "rajesh.sadhu@appiyo.com",

"mdmPan": "",

"natureOfIndustry": null,

"natureOfIndustry2": null,

"natureOfIndustry3": null,

"nomineeRelationData": null,

"panExist": false,

"politicalExposure": null,

"principleInstructions": null,

"productTest": 0,

"profession": null,

"reKycDt": "",

"residenceType": null,

"seniorCitizen": false,

"sourceOfFunds": null,

"srcOfFunds2": null,

"srcOfFunds3": null,

"totalFDBalance ": 0,

"transactionData": null

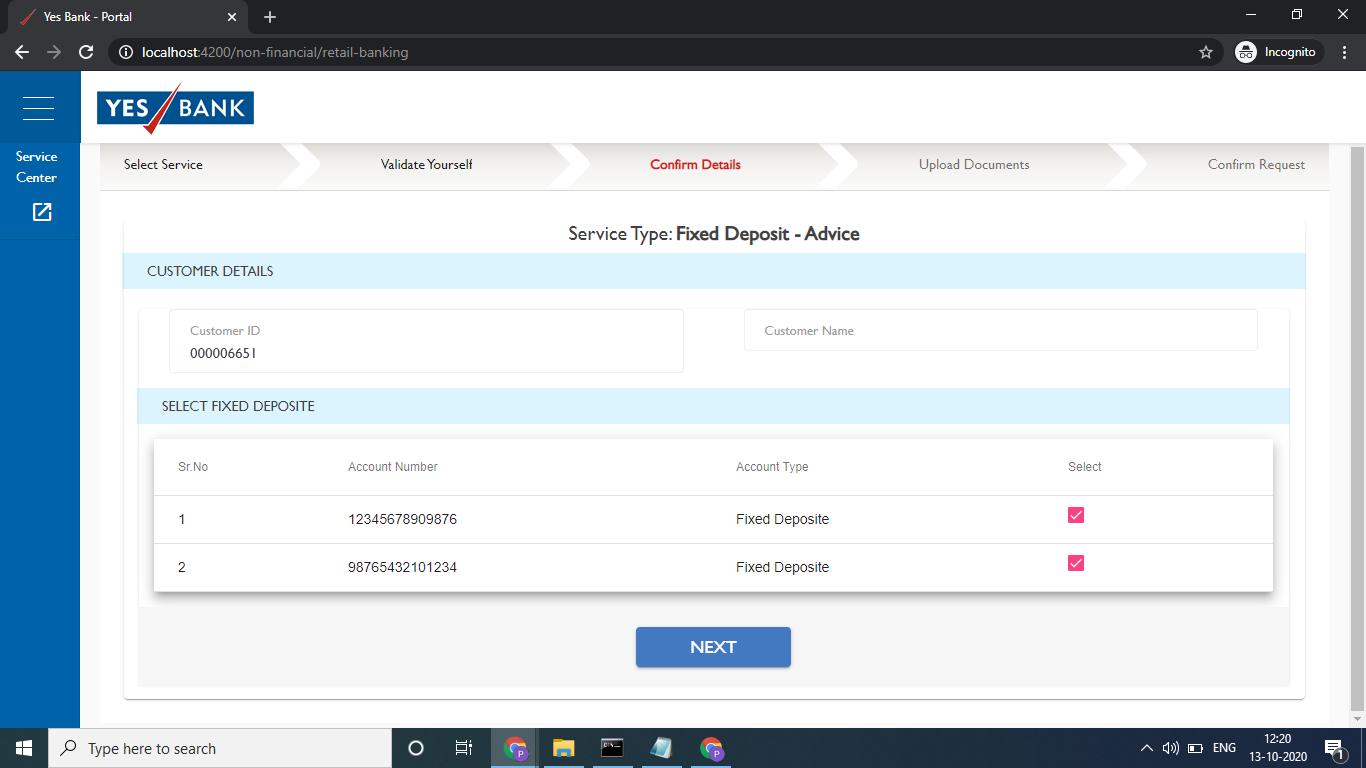
},

"Status": "Execution Completed",

"WorkflowId": "f73014cac99911e9933d8e7a151d5229"

}

6)



**Api Called**:

Service Name :- Get session details for casa and fd

Project :- YBLIGNITE

Module :- SSP

Workflow :- Get session details for casa and fd

Process Name :- Get session details for casa and fd

Nodes: -

1. Basic Validation : This node account no and customer id pass as a json array
2. Intense API Call : This is the call activity used to perform call intense api . Input is account no and customer id and output of this call activity is fileID, statementType, statementName, productType, accountNumber, fromDate, toDate
3. Parse details : intenseData is bind into this
4. Set response : if data is coming the error code 200 else error code 413 which is

Regret inconvenience, We are unable to retrieve the statement for the selected account no. Please contact your RM/nearest branch for further assistance

Internal api call (Intense service)

Service Name :- Intense service

Project :- YBLIGNITE

Module :- API Integration

Workflow :- Intense service

Process Name :- Intense service

1. Construct api request : This node account no and customer id pass as input
2. Add log data req : This is a call activity which is perform to store a log data
3. prepare mAuth req : This node mAuthRequestEndPoint, mAuthRequestMethod, mAuthRequestHeaders values are set which is required to call third party api
4. M Auth Call : This node third party api is called
5. Set Response : This node If response is not 200 then status , status code and failure message is set
6. Add log data resp : This is a call activity which is perform to store a log data

Parse response : This node set the values of the variables is fileID, statementType, statementName, productType, accountNumber, fromDate, toDate

Get additional data for fd and casa

Request :

{

"projectId": "ff0ae4a6884711e9b16676fb2f2488b6",

"workflowId": "2d588146fca611ea9abd0242ac110003",

"processId": "2d8189f6fca611eaa0c30242ac110003",

"ProcessVariables": {

"sessionId": "e1701fb12640c8b1dca19bb798801041b090a53bbc8d77b2863795a3a59f69b",

"srType": "1016"

}

}

Response :

{

"ApplicationId": "ff0ae4a6884711e9b16676fb2f2488b6",

"Error": "0",

"ErrorCode": "200",

"ErrorMessage": "",

"ProcessId": "2d8189f6fca611eaa0c30242ac110003",

"ProcessInstanceId": "8d01a1200a1311ebac940242ac110003",

"ProcessName": "SSP:: Get Additional Data",

"ProcessVariables": {

"errorCode": "200",

"intenseData": [

{

"accountNumber": "12345678909876",

"fileID": "66666666",

"fromDate": "13/03/2020",

"toDate": "12/04/2020"

},

{

"accountNumber": "98765432101234",

"fileID": "7777777",

"fromDate": "01/04/2020",

"toDate": "02/05/2020"

}

]

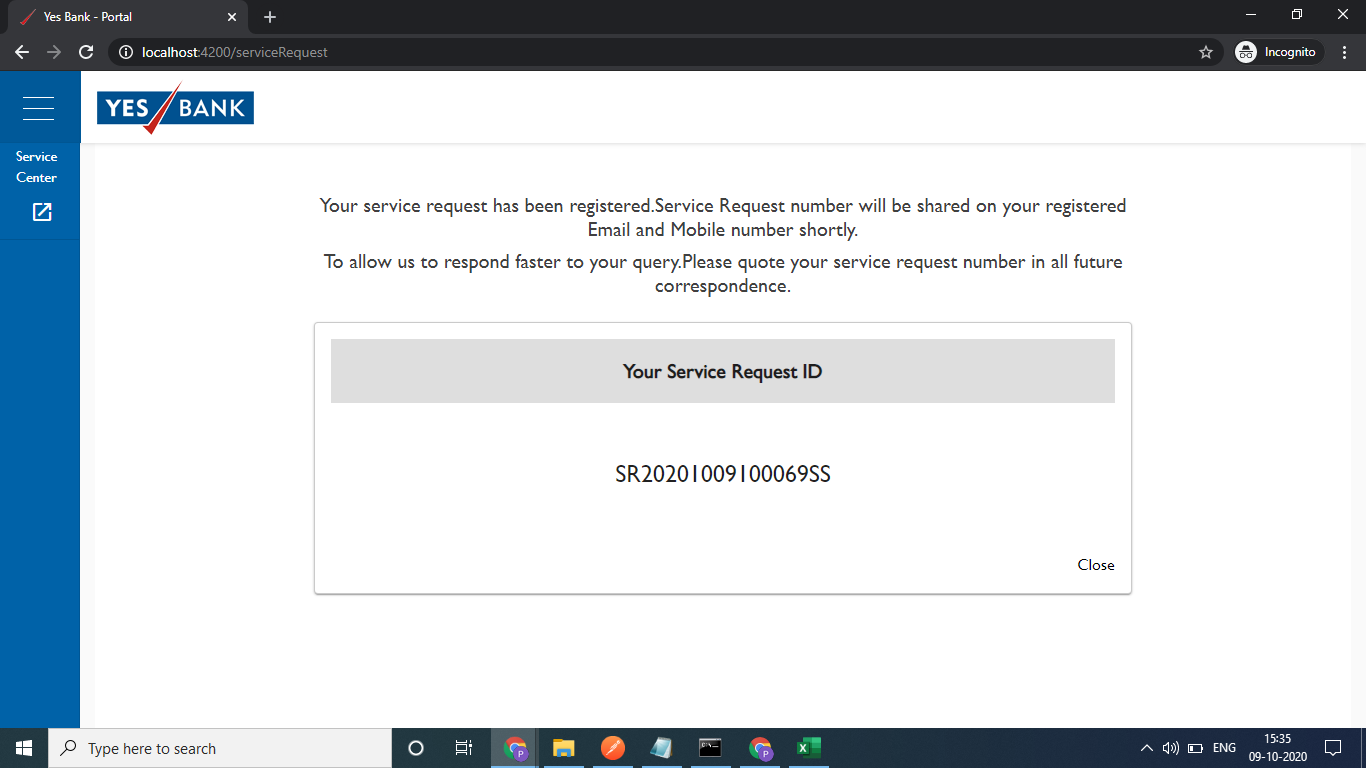
},

"Status": "Execution Completed",

"WorkflowId": "2d588146fca611ea9abd0242ac110003"

}

7)



**Api call:**

1. Create Sr :

Service Name :- Create SR

Project :- YBLIGNITE

Module :- SSP

Workflow :- SSP:: CreateSR

Process Name :- SSP\_Create\_SR

Nodes:-

1. Validate Header :- This Script node is to validate Header and Header.refId if it is there then we set the referenceId for further operation.

2. Get Session Details: - This is db node for fetch the session Details of the customer by refId like custName,Pan,SrType etc.

3. Basic Data Validations: - This is a Call Activity used to perform the basic validations.

4. Construct Error Message: - This node is come into picture if basic data validations call activity is not return success response then we show the error message.

5. Sr Specific Data Validations: - This call Activity is used to perform Sr Specific validations.

6. Construct Error Message: - This node is come into picture if Sr Specific data validations call activity is not return success response then we show the error message.

7. Validate Reference Id: - This node is used to count the refId.

8. Construct Error Message: - This node is come into picture if Validate Reference Id node is return refId count > 0 then we show the error message.

9. Any SR Is InProgress?: - In this call Activity we Checked any SR is in progress for the current Customer ID.

10. Construct Error Message: - This node is come into picture if Any SR Is InProgress is return true (inProgress) then we show the error message.

11. Construct DB Query: - This script node is to construct query for inserting in Service Request table.

12. Populate Data Into DB: - This db node is used to execute query to insert data in Service Request Table.

13. GENERATE SR ID: - This db node is used to update sr\_id as per given pattern.

14. Get SR Id:- This db node is used to get the updated SrId.

15. Create SR Others (Level 1): - This call activity is used for perform the operation for respective SrType we pass the srType and respective data to this call activity.

16. Update SR\_ID Into Session Schema: - In this db node we update the srId in ss\_session table based on refId.

17. Add SR Into Pending Bucket: - In this db node we insert srId in pending\_sr table.

18. Mark SR As Created: - This call activity is used to mark the SR as created.

19. Set Response Message: - This script node is used for set errorCode and message as SR created successfully.

20. Create CRM Service: - In this call activity we call the crm case creation process for creating crm case by passing srId,SrType,CustId then End the Flow

Request :

{

"projectId": "ff0ae4a6884711e9b16676fb2f2488b6",

"workflowId": "75e5596ac95e11e9933d8e7a151d5229",

"processId": "01ea2356c99a11e9933d8e7a151d5229",

"ProcessVariables": {

"headers": {

"refId": "2bcfbee1b41abff097667203537bac8b80f7b5de9520552a40a28bae65b0b4",

"channel": "SS",

"appId": "app2"

},

"srDetails": {

"srType": 1016

},

"fdAdvice": [

{

"accountNumber": "12345678909876",

"fileId": "66666666",

"fromDate": "2020-03-13",

"toDate": "2020-04-12",

"accountType": "Fixed Deposite"

},

{

"accountNumber": "98765432101234",

"fileId": "7777777",

"fromDate": "2020-04-01",

"toDate": "2020-05-02",

"accountType": "Fixed Deposite"

}

]

}

}

Response:

{

"ApplicationId" : "ff0ae4a6884711e9b16676fb2f2488b6",

"Error" : "0",

"ErrorCode" : "200",

"ErrorMessage" : "Request created successfully",

"ProcessId" : "01ea2356c99a11e9933d8e7a151d5229",

"ProcessInstanceId" : "a6f6da7c0d2011eb94cc0242ac110003",

"ProcessName" : "SSP\_Create\_SR",

"ProcessVariables" : {

"apiUniqueReqId" : "",

"businessNew" : "",

"businessOld" : "",

"caseId" : "",

"comments" : "SR Created. Mobile = \*\*\*\*\*\*8600; Customer Id = \*\*\*\*\*6651; ",

"custDetails" : {

"accountNumber" : null,

"custFirstName" : null,

"custId" : "000006651",

"custLastName" : null,

"custName" : "000006651",

"email" : "rajesh.sadhu@appiyo.com",

"landline" : "022-44449956",

"mobileNumber" : "9500138600"

},

"dematOtpKey" : "",

"identifiers" : "Mobile = \*\*\*\*\*\*8600; Customer Id = \*\*\*\*\*6651; ",

"industryNew" : "",

"industryOld" : "",

"insertQuery" : "SET LANGUAGE British;INSERT INTO service\_request(ref\_id,app\_id,channel,cust\_id,cust\_name,account\_number,mobile,channel\_mobile,identifiers,sr\_type,created\_by,client\_ip,branch\_id,email,created\_on,updated\_on,updated\_by,current\_status) VALUES ('2bcfbee1b41abff097667203537bac8b80f7b5de9520552a40a28bae65b0b4','app2','SS','000006651','000006651',NULL,'9500138600','9500138600','Mobile = \*\*\*\*\*\*8600; Customer Id = \*\*\*\*\*6651; ',1016,'101','172.17.0.1','1','rajesh.sadhu@appiyo.com',SYSUTCDATETIME(),SYSUTCDATETIME(),'101',1)",

"moduleName" : "RL",

"needDocs" : true,

"needSupportedDocs" : false,

"profileOld" : null,

"roleId" : 0,

"sourceNew" : "",

"sourceOld" : "",

"srDetails" : {

"branchId" : "1",

"srId" : "SR20201013100074SS",

"srName" : null,

"srType" : 1016

},

"srId" : "SR20201013100074SS",

"userId" : ""

},

"Status" : "Execution Completed",

"WorkflowId" : "75e5596ac95e11e9933d8e7a151d5229"

}

1. Submit Sr:

Service Name :- Submit SR

Project :- YBLIGNITE

Module :- SSP

Workflow :- SSP:: SubmitSR

Process Name :- SSR :: Submit SR

Nodes: -

1. Get Session Details: - This is db node for fetch the session Details of the customer by refId like currentStatus,custId,SrType etc.
2. Basic Validations On SR Details: - In this script node we do validations based on SR details.
3. Check If SR In-progress?? :- In this call Activity we check if any SR in Progress or not for current srType.
4. Construct Response : - In this node we construct the response if any SR is already in progress.
5. Mark SR As Submitted By Customer: - In this call Activity we mark Status of SR as Submited by Customer.
6. Send Notifications To Customer: - This is the Intermediate Events using that we send message to customer with SR status.
7. Update With MDM And Mark As Completed :- This is the Intermediate Events inside this we call send verification link workflow.
8. Fetch SR Details: - In this db node we fetch SR details like srTpe, email etc by refId.
9. Check SR Type: - In this script node we check the srTpe and set the value for emailVerificationStatus.
10. MDM Update & Close SR: - In this call Activity we call the process MDM Update & Close SR for updating the SR and Close.
11. CASA Statement & FD Advise: - In this call activity we call the process FD Statement (CRM Request Handler) and passing srId.
12. get fileId from FD table: - In this db script we fetch the caseId, fileId from respective table.
13. Basic Validation: - in this script node we extract the fileId from Object for further operation.
14. Crm Request Handler Call: - In this call activity we call the CRM Request Handler process and pass the file Id as input.
15. Parse Details: - In this script node we parse the response of crm Request Handler.
16. Audit Log: - This call activity use for logging the response in audit logger table.
17. Set Response: - in this script node we update the status of crm\_request\_handler\_status in fd\_statement\_details.
18. CRM case Status: - in this call Activity we call the crm case status process and set the crm case status.
19. End node : - finally the SubmitSR process is End.

Request :

{

"projectId": "ff0ae4a6884711e9b16676fb2f2488b6",

"workflowId": "e254d6ead55f11e9933d8e7a151d5229",

"processId": "ee966c98d55f11e9933d8e7a151d5229",

"ProcessVariables": {

"srId": "SR20201013100074SS",

"apiUniqueReqId": "1602571948657"

}

}

Response:

{

"ApplicationId": "ff0ae4a6884711e9b16676fb2f2488b6",

"Error": "0",

"ErrorCode": "200",

"ErrorMessage": "",

"ProcessId": "ee966c98d55f11e9933d8e7a151d5229",

"ProcessInstanceId": "a7c799500d2011ebba780242ac110003",

"ProcessName": "SSR :: Submit SR",

"ProcessVariables": {

"apiUniqueReqId": "1602571948657",

"isApprovedByDvu": false,

"roleId": 0,

"setCaseResponse": null,

"statusExternal": "Rejected By DVU",

"statusText": "1",

"userId": ""

},

"Status": "Execution Completed",

"WorkflowId": "e254d6ead55f11e9933d8e7a151d5229"

}