Introduction
Learn how to integrate our APIs into your application.

API Basics
The One Point API gives you access to features allowed by the bank and lets you extend them for use in your application. It strives to be REST and is organized around the main resources you would be interacting with - with a few notable exceptions.

Before you do anything: You will need an API user account that you can test the API against. Bank will create and provide username, password and X509 private certificate used to make API calls.

Request
Request model body remains same for all the requests where FunctionName is the name of the function to be called, Data is argument based upon the function and Signature is hash of the data generated with the private key.

Request Format

```
curl http://citzchannelmanagerapi.iis/api/v1/connect
-H "Authorization: Basic username:password"
-H "Content-Type: application/json"
-d
{
   "FunctionName": "FunctionName",
   "Data": "ew0KICAgICJBN0aW9uSWTIzIg0KfQ0K=",
   "Signature": "YdKYJ03M+J76a3vAe0Ndy8V0u7Ub7ZmdDdKs3RzWvlkywvg9aQ==",
   "TimeStamp": "2021-03-24T11:52:20.362"
}
-X POST
```

FunctionName	string	Name of Function (See API Functions).
Data	string	Base 64 encoded Json String based on Function Name.
Signature	string	Private Certificate (P.E.M file) is provided for signature generation. Signature is generated using SHA256 with RSA. Which is again encoded to Base 64
TimeStamp	string	Current time stamp of the request to be sent in yyyy-MM-ddTHH:mm:ss.fff format valid up to 5 mins

Signature generation:
Signature bytes is made up of signature generation model which includes the model based upon the function name and timestamp sent in the common parameter
requestModel={
"TransactionId": "123",
"AccountNumber": "1900000001"
}
signatureModel={
"Model": requestModel,
"TimeStamp": TimeStamp
}

Response

Both request body data and response data are formatted as JSON. Content type for responses will always be application/json. Generally, all responses will be in the following format.

Response Format

```
{
   "Code": [string],  // Only "0" if request is processed and no error occured while processing
    "Message": [string], // Explains why status is not success.
     ... ...  // contains actionable result of processing if present
}
```

Function name: All the request and response parameter keys should be set to PascalCase.

While HTTP status codes determines the general result of an API call, we have provided a handy Code key to let you know upfront if the request was successful or not.The Message key is a string which will contain a summary of the response and its status. For instance when trying to validate an account number, message might read “Account Validated Successfully”.In the event of an error, which could be because of "Invalid number" or "CBS connectivity error", the message key will contain a description of the error. Code and Message are the keys that is universal across requests.The apart from Code and Message key is where you want to look at for the result of your request. It can either be one or more primitive type(s) [string] or an object or an array depending on the request made.

Authentication

Authenticate your API calls by using basic authentication in the Authorization header of every request you make..

Bank will provide X509 private key certificate which is to be kept secret. If for any reason you believe your secret key has been compromised or you wish to reset them, you must inform the bank.

Do not commit your secret keys to git, or use them in client-side code.

Authorization headers should be in the following format: Authorization: Basic Base64(username:password)

Sample Authorization Header Authorization: Basic ew0KICAgICJBY2NvdW50TnVtYmVyIjogIjEyMzQ1NiIsDfQ==

API requests made without basic authentication will fail with the status code 401: Unauthorized

Errors

As stated earlier, OnePoint API is based on REST and as such, uses conventional HTTP response codes to indicate the general success or failure of requests. Plus Code key and Message key to undermine its specific cause of error.
Some of which are discussed below :

Common Parameter Error	Code
When attributes set to the common model is invalid such as invalid base64encoded or invalid json string formed.	1XX
Request Validation Error	Code
Request's model based on FunctioName validation fails when required fields are empty or invalid data types set to the properties.	2XX
Operation Error	Code
When provided with Transaction Ids or any other Ids which is either Duplicate/ Doesn't exists/ Already Updated/ Invalid.	3XX
Authorization Error	Code
1. Basic Authentication Error. 2. Signature Verification Failed. 3. No rights for the API.	4XX
Limits Check Error	Code
1. Checks for blocked accounts. 2. Checks for exception accounts. 3. Checks limits applied via the allocated channel. (for e.g: min amount,daily limit,daily frequency)	5XX
Custom API Error	Code
Error in the custom API made by the bank.	6XX
CBS Error	Code
This happens when there is connectivity issue from OnePoint to the CBS or when CBS doesn't provide response in time or not at all	7XXX
Service Call Error	Code
Failure,Pending or Unknown Status from Consuming Service	8XX
OnePoint's Exception	Code
Response message with Something went wrong or Contact with error Id means some expection occured in the OnePoint system Contact the vendor with the returned Id	999

STATUS PENDING CASE Pending or Unknown Status Occures when there is no response or timeout in response from Core Banking or Exception occurs in One Point system. The codes for which reconcilation or check status is required are: [71500],[71503],[999]

API Functions

Note: Exclude space when providing FunctionName for eg:- Balance Enquiry will be called as BalanceEnquiry (Case Sensitive).

CustomerEmailByForacid

Gets customer email by foracid for gump

Body Param

TransactionId	string(255)	Unique Request Identifier

(Post) Json String

{
"TransactionId": "CEBF-11",
"FORACID": "001010000XXXXXXX"
}

Response Param

Code	string	Success or Failure Indicator
Message	string	Success or Failure Reason
Data	object	Actionable Result

(Success) Sample Response
{ "Code": "0", "Message": "Operation Successfull", "Data": [{ "ACCT_NAME": "XXX XXX XXX", "SCHM_TYPE": "SBA", "SCHM_CODE": "SS011", "SCHM_DESCRIPTION": "CITIZENS STAFF SAVINGS ACCOUNT", "ACCT_CRNCY_CODE": "NPR",

"FREZ_CODE": " ",
"ACCT_CLS_FLG": "N",
"ACCT_STATUS": "A",
"PHONE_NUM": "98467XXXXX",
"EMAIL": "XXX.XXX@GMAIL.COM"
}
],
"DeveloperMessage": null,
"Errors": null
}

(Validation Error) Sample Response

{
"Code": "601",
"Message": "Validation Error",
"Data": "Account is Closed!",
"DeveloperMessage": null,
"Errors": null
}

(Validation Error) Sample Response

{
"Code": "601",
"Message": "Validation Error",
"Data": "Account is Dormant!",
"DeveloperMessage": null,
"Errors": null
}

(Validation Error) Sample Response

{
"Code": "601",
"Message": "Validation Error",
"Data": "Account is in Freeze Status!",
"DeveloperMessage": null,
"Errors": null
}