Bank Hapoalim's **BANKAPP** Open API Specification Doc

Version	Date	Author	Description
1.0	02-Oct-2014	Bank Hapoalim	Initial API
1.2	07-Oct-2014	Bank Hapoalim	Lamas API
1.3	10-Mar-2015	Bank Hapoalim	CMC version

bank hapoalim

Overview

Bank Hapoalim's Open API offers an easy-to-use RESTful JSON API.
Connect in minutes and start using it. With the Open API you can :

- Integrate bank account information in minutes via our REST API
- Leverage a consistent and bank-agnostic interface

Access to the data is possible in two ways:

1) Directly querying the data through DynamoDB, a fully managed NoSQL database service (on AWS).
2) Using the APIs via REST.

Please NOTE:
Some of the data is only accessible through the REST API:

1. Geo location for branches/ATMs("bankats").
2. Real accounts data.
3. Retrieving a statistical geographical area ("agas" code) and polygon from a geo-point.

DynamoDB

DynamoDB is a NoSQL database as a service run by Amazon Cloud Services.

We chose it because of its simplicity, scalability and robustness.

As part of the Open API you can access bank data either directly through DynamoDB in the language of your choice, or through our REST APIs. It is up to you to choose your preference.

To learn more about DynamoDB, you can visit this link.

Amazon provides full documentation and SDKs for many platforms which explain how to connect and query DynamoDB.

You can read more about DynamoDB for Developers, and find code examples and API documentation here.

Once you sign up to Bank Hapoalim's Open API you will receive a set of access and secret keys with which you can use to connect to our DynamoDB and start querying the database.

In this document, we describe both the tables available in DynamoDB alongside REST queries that you can make and their corresponding responses.

Important

For each table structure we give you details about indexed columns on DynamoDB so that you could better utilize DynamoDB for your queries.

Also please note you have to initialize the AWS DynamoDB client to region **eu-west-1**.

DynamoDB API Code Example

```
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List; import
java.util.Map;

import org.apache.commons.lang.StringUtils;

import com.amazonaws.auth.AWSCredentials;
import
com.amazonaws.auth.BasicAWSCredentials;
import com.amazonaws.regions.Region; import
com.amazonaws.regions.Regions;
import com.amazonaws.services.dynamodbv2.AmazonDynamoDBClient; import
com.amazonaws.services.dynamodbv2.model.AttributeValue; import
com.amazonaws.services.dynamodbv2.model.ComparisonOperator; import
com.amazonaws.services.dynamodbv2.model.Condition;
```

```
import com.amazonaws.services.dynamodbv2.model.QueryRequest;
import com.amazonaws.services.dynamodbv2.model.QueryResult; import
com.amazonaws.services.dynamodbv2.model.ScanRequest; import
com.amazonaws.services.dynamodbv2.model.ScanResult; import
com.poalim.dynamodb.util.Util;


public class PoalimAWSTest {

      private final static String CREDIT_CARD_TRANSACTION_TABLE = "CreditCards4";
      private final static String PURCHASE_DATE_INDEX = "Purchase_Date-Index";

      private final static String CC_TABLE_HASH_KEY = "Account_Number"; private
      final static String CC_PURCHASE_DATE_KEY = "Purchase_Date";


      private final static String ACCOUNTS_TABLE = "Accounts"; private final static
      String ACCOUNTS_CUSTOMER_GENDER_KEY = "Customer_Gender"; private final static
      String ACCOUNTS_BRANCH_CITY_KEY = "Branch_City";

      private final static String testAccountNumber = "16998";

      private final static String accessKey = "<accessKey>"; private
      final static String secretKey = "<secretKey>";

      private AmazonDynamoDBClient client;


      public static void main(String[] args)
      {

            PoalimAWSTest  aPoalimAWSTest = new PoalimAWSTest();

            System.out.println(aPoalimAWSTest.getCreditCardsTransactionsByDate(test
            ountNumber, "20140101", "20140131"));

            System.out.println(aPoalimAWSTest.getAllAccountsByGenderAndCity(“Holon”, “M"));
      }

      public PoalimAWSTest (){

            AWSCredentials creds = new BasicAWSCredentials(accessKey, secretKey);
```

```
            client = new AmazonDynamoDBClient(creds);
            client.setRegion(Region.getRegion(Regions.EU_WEST_1));
      }
         public   String   getCreditCardsTransactionsByDate(String   accountNumber,   String
fromDate, String       toDate)
  {

   List<Map<String,     AttributeValue>>     itemsMapList     =     new     ArrayList<Map<String
AttributeValue>>()

   Map<String, AttributeValue> lastEvaluatedKey = null;

   do
   {

      Map<String, Condition> keyConditions = new HashMap<String, Condition>();

      Condition hashKeyCondition = new
      Condition().withComparisonOperator(ComparisonOperator.EQ.toString()).withAttr
      teValueList(new AttributeValue().withN(accountNumber));

      keyConditions.put(CC_TABLE_HASH_KEY, hashKeyCondition);

      Condition dateCondition = new Condition();

      if( (!StringUtils.isBlank(fromDate) && Util.isNumeric(fromDate)) &&
      (!StringUtils.isBlank(toDate)  && Util.isNumeric(toDate)))
      {

            dateCondition.withComparisonOperator(ComparisonOperator.BETWEEN.toString())
            .withAttributeValueList(new AttributeValue().withS(fromDate), new
            AttributeValue().withS(toDate));

            keyConditions.put(CC_PURCHASE_DATE_KEY, dateCondition);
      } else
      { return null;
      }
```

```
      QueryRequest queryRequest = new
      QueryRequest().withTableName(CREDIT_CARD_TRANSACTION_TABLE).withIndexName(PURCHASE_DATE_
      X).withKeyConditions(keyConditions).withExclusiveStartKey(lastEvaluatedKey);

      QueryResult result = client.query(queryRequest);


      itemsMapList.addAll(result.getItems());

      lastEvaluatedKey = result.getLastEvaluatedKey();
   }
   while (lastEvaluatedKey != null);

   return itemsMapList.toString();
  }



  public String getAllAccountsByGenderAndCity(String gender, String branchCity)
{

   List<Map<String, AttributeValue>> itemsMapList = new ArrayList<Map<String
AttributeValue>>()

   Map<String, AttributeValue> lastEvaluatedKey = null;

   int counter = 0;
do {

            Map<String, Condition> conditions = new HashMap<String, Condition>();

            Condition genderFilterCondition = new
            Condition().withComparisonOperator(ComparisonOperator.EQ.toString()).wi
            ttributeValueList(new AttributeValue().withS(gender));

            conditions.put(ACCOUNTS_CUSTOMER_GENDER_KEY, genderFilterCondition);

            Condition branchCityFilterCondition = new
            Condition().withComparisonOperator(ComparisonOperator.EQ.toString()).wi
            ttributeValueList(new AttributeValue().withS(branchCity));

            conditions.put(ACCOUNTS_BRANCH_CITY_KEY, branchCityFilterCondition);
```

```
            ScanRequest scanRequest = new ScanRequest()
                  .withTableName(ACCOUNTS_TABLE)
                  .withScanFilter(conditions)
                  .withExclusiveStartKey(lastEvaluatedKey);;

            ScanResult result = client.scan(scanRequest);


            itemsMapList.addAll(result.getItems());

            lastEvaluatedKey = result.getLastEvaluatedKey();

            System.out.println("current counter in scan =["+ ++counter+"]");

      }
      while (lastEvaluatedKey != null);

      return itemsMapList.toString();
  }

}
```

Bank API

The bank API encapsulates data access to the underlying DynamoDB.
In every API access you must provide the security credentials you have received in your registration (as headers).
The API allows you to query the data on most of the indexes, however:

1. It doesn't allow paging.
2. It doesn't offer a scan table operation which is possible through direct DynamoDB operation.

Base URL for API Calls

For every API call, the base URL is https://api.bankapp.co.il/

Agas Code (Statistical Geographical Area)

The Agas codes provided in the Tables and API (as parameters and response elements) represents demographical areas as defined by the CBS (Central Bureau of Statistics Israel). They are in fact polygons on the map of Israel.

The Israel CBS divided Israel's map into around 3600 geographical regions depicted by polygons. In these geographic statistical areas, each denoted by an 8 digits code and known as “Agas”, they aggregated valuable sociological and demographic data.

Some of the fields that the CBS provides contain socio-demographic ranking, age distribution, average rooms per household and much more.

This information can be crossed and analyzed with data from the Bank API which is based on location, to produce interesting results which can be then be used and visualized on a map layer.

The codes can be retrieved from DynamoDB directly (“***LamasLocations***” table) or by invoking our REST API (to get the polygon of a specific code, or retrieve the code of a geo-point).

- ***Lamas*** *are Hebrew initials for the CBS (Central Bureau of Statistics Israel).*

Accounts Table

Contains account related data.

Each record represents a BANK HAPOALIM account owner and his administrative and demographical details

Field Name	Description
Account_Number	Unique customer identifier

Branch_Number	Customer bank account and branch number unique identifier
Branch_City	Customer Branch City
Branch_Name	Branch commercial name
Account_Owner_Address	Demographical Area branch address of the oldest client in the account (According to Central Bureau of Statistics Israel)
Account_Open_Date	Account establishment year date (yyyy)
Account_Type_Description	Used to distinguish between private account or corporate account (Private / Business)
Customer_Name	Customer full name – the name is encrypted for confidentiality purposes
Customer_Age	The age is rounded up for confidential purposes.
Customer_Gender	M / F If several account owners exist, the older is selected.
Language_Preference	Customer preferred correspondence language: Hebrew / English / Arabic

Example data

Field Name	Value
Account_Number	211687
Branch_Number	236631234
Branch_City	Tel Aviv Yafo
Branch_Name	Tel Aviv Yafo branch

Account_Owner_Address	50000121
Account_Open_Date	2005
Account_Type_Description	Private
Customer_Name	Customer 211687
Customer_Age	43
Customer_Gender	Female
Language_Preference	Hebrew

DynamoDB table name

Accounts

Indexed Columns

Primary Hash Key:	Account_Number (Number)		
Global Secondary Indices			
Index Name	**Hash Key**	**Range Key**	**Projected Attributes**
Account_Owner_Address-Account_Type_Description-Index	Account_Owner_Address (String)	Account_Type_Description (String)	All
Account_Owner_Address-Customer_Age-Index	Account_Owner_Address (String)	Customer_Age (Number)	All
Account_Type_DescriptionCustomer_Age-Index	Account_Type_Description (String)	Customer_Age (Number)	All

1. Get Sample Accounts

Retrieve top X accounts as defined in the server

Request

Method	URL
`GET`	`api/account/sample/`

Type	Params	Values
HEAD HEAD	accessKey secretKey	string string

accessKey
The `accessKey` that was given by the Bank Open API

secretKey
The `secretKey` that was given by the Bank Open API

Response

Status	Response

200	[{ "accountNumber": 154270, "branchNumber": "226106123", "branchCity": "Herzliya", "branchName": "Herzliya branch", "accountOwnerAddress": "64000066", "accountOpenDate": "2011", "accountTypeDescription": "Private", "customerName": "Customer 154270", "customerAge": 44, "customerGender": "Female", "languagePreference": "Hebrew", "customerType": "Private" }, { "accountNumber": 225407, "branchNumber": "161225123", "branchCity": "Kfar Tavor ", "branchName": "Kfar Tavor branch", "accountOwnerAddress": "00470001", "accountOpenDate": "2002", "accountTypeDescription": "Private", "customerName": "Customer 225407", "customerAge": 50, "customerGender": "Female", "languagePreference": "Hebrew", "customerType": "Private" }, { "accountNumber": 35176, "branchNumber": "236831234", "branchCity": "Tel Aviv Yafo", "branchName": "Tel Aviv Yafo branch", "accountOwnerAddress": "50000423", "accountOpenDate": "2011", "accountTypeDescription": "Private", "customerName": "Customer 35176 ", "customerAge": 25, "customerGender": "Male", "languagePreference": "Hebrew", "customerType": "Private"

	} ...]
200	{"errorMessage":"Request failed: server error"}

2. Account Details for Account

Get account details by account number

Request

Method	URL
GET	api/account/{account_number}/details

Type	Params	Values
HEAD HEAD	accessKey secretKey	string string

accessKey
The `accessKey` that was given by the Bank Open API

secretKey
The `secretKey` that was given by the Bank Open API

account_number
The account number for which we want to get the account details.

Response

Status	Response

200	<pre>{
	"accountNumber": 225407,
	"branchNumber": "161225123",
	"branchCity": "Kfar Tavor ",
	"branchName": "Kfar Tavor branch",
	"accountOwnerAddress": "00470001",
	"accountOpenDate": "2002",
	"accountTypeDescription": "Private",
	"customerName": "Customer 225407",
	"customerAge": 50,
	"customerGender": "Female",
	"languagePreference": "Hebrew",
	"customerType": "Private"
	}</pre>

200	<pre>{"errorMessage":"Request failed: server error"}</pre>

3. Complete Account Data for Account

Get all account data for a specific account number

Request

Method	URL
GET	api/account/{account_number}/all

Type	Params	Values
HEAD HEAD	accessKey secretKey	string string

`accessKey`
The `accessKey` that was given by the Bank Open API

`secretKey`
The `secretKey` that was given by the Bank Open API

account_number
The account number for which we want to get the complete account data.

Response

Status	Response

200	{
	"account": {
	"accountNumber": 225407,
	"branchNumber": "161225123",
	"branchCity": "Kfar Tavor ",
	"branchName": "Kfar Tavor branch",
	"accountOwnerAddress": "00470001",
	"accountOpenDate": "2002",
	"accountTypeDescription": "Private",
	"customerName": "Customer 225407",
	"customerAge": 50,
	"customerGender": "Female",
	"languagePreference": "Hebrew",
	"customerType": "Private"
	},
	"balance": {
	"accountNumber": 225407,
	"balanceId": "57361",
	"validityDate": "20140630",
	"nisBalance": "0",
	"dollarBalance": "0",
	"euroBalance": "0",
	"creditBalance": "0",
	"depositBalance": "0",
	"capitalMarketBalance": "0"
	},
	"creditCardsList": [
	{
	"accountNumber": 225407,
	"chargeId": 5547802,

```
                    "purchaseAmount": 200,
                        "paymentAmount": "200.00",
        "numberOfPaymentsInPurchase": "1",
                    "numberOfPayment": "1",
                        "purchaseDate": "20140625",
                        "businessName": "Business 4742991",
        "paymentCategory": "Pharmacies",
        "businessLocation": "Supplier_address missing"
    },
    {
        "accountNumber": 225407,
                    "chargeId": 5547803,
                    "purchaseAmount": 148.59,
                        "paymentAmount": "148.59",
        "numberOfPaymentsInPurchase": "1",
                    "numberOfPayment": "1",
                        "purchaseDate": "20140622",
                        "businessName": "Business 2238489",
        "paymentCategory": "Life Insurance Companies",
        "businessLocation": "Supplier_address missing"
    }
                        ...
],
"transactionsList": [
  {
        "accountNumber": 225407,
                        "transactionId": 140108090001277630,
                        "transactionDate": "20140108",
                        "transactionDescription": "Salary",
        "transactionAmount": 4777.45,
                        "transactionCurrency": "NIS",
        "creditDebitCode": "1",
        "transactionGroupDescription": "Current Accounts"
    },
    {
        "accountNumber": 225407,
                        "transactionId": 140110090002861550,
                        "transactionDate": "20140110",
                        "transactionDescription": "Isracard",
        "transactionAmount": 1792.78,
                        "transactionCurrency": "NIS",
        "creditDebitCode": "2",
```

	"transactionGroupDescription": "Credit Cards" } ...] }
200	{"errorMessage":"Request failed: server error"}

4. Accounts in Age Range for Agas Area

Get all accounts where the account owner lives in a specified Agas area and his age is in the specified range.

Request

Method	URL
GET	api/account/age-range-and-agas-area

Type	Params	Values	
HEAD HEAD GET GET GET	accessKey secretKey fromAge toAge agasArea	string string number number number	[optional] [optional] [mandatory]

accessKey
The `accessKey` that was given by the Bank Open API

secretKey
The `secretKey` that was given by the Bank Open API

fromAge
Start of age range

toAge
End of age range

agasArea agasArea
as defined.

Response

Status	Response

200	[
	{
	"accountNumber": 154639,
	"branchNumber": "226106123",
	"branchCity": " Herzliya",
	"branchName": "Herzliya branch",
	"accountOwnerAddress": "64000066",
	"accountOpenDate": "2004",
	"accountTypeDescription": "Private",
	"customerName": "Customer 154639",
	"customerAge": 56,
	"customerGender": "Female",
	"languagePreference": "Hebrew"
	},
	{
	"accountNumber": 154281,
	"branchNumber": "226106123",
	"branchCity": " Herzliya",
	"branchName": "Herzliya branch",
	"accountOwnerAddress": "64000066",
	"accountOpenDate": "2010",
	"accountTypeDescription": "Private",
	"customerName": "Customer 154281",
	"customerAge": 56,
	"customerGender": "Female",
	"languagePreference": "Hebrew"
	}
	...
	]

200	{"errorMessage":"Request failed: server error"}

5. Accounts with accountTypeDescription and AgasArea

Get all accounts where the account owner lives in the specified AgasArea and holds the given accountTypeDescription.

Request

Method	URL
GET	api/account/type-description-and-agas-area

Type	Params	Values	
HEAD HEAD GET GET	accessKey secretKey accountTypeDescription agasArea	string string string number	[optional] [mandatory]

accessKey
The `accessKey` that was given by the Bank Open API

secretKey
The `secretKey` that was given by the Bank Open API

accountTypeDescription
Business / Private

agasArea agasArea
as defined.

Response

Status	Response
200	[

	{
	"accountNumber": 70569,
	"branchNumber": "628731234",
	"branchCity": "Pardes Hana”,
	“branchName”: “Pardes Hana branch”,
	"accountOwnerAddress": "78000003",
	"accountOpenDate": "2005",
	"accountTypeDescription": "Private",
	"customerName": "Customer 70569",
	"customerAge": 64,
	"customerGender": “M”,
	"languagePreference": "Hebrew"
	},
	{
	"accountNumber": 111571,
	"branchNumber": "628731234",
	"branchCity": "Pardes Hana”,
	“branchName”: “Pardes Hana branch”,
	"accountOwnerAddress": "78000003",
	"accountOpenDate": "2005",
	"accountTypeDescription": "Private",
	"customerName": "Customer 111571",
	"customerAge": 60,
	"customerGender": “M”,
	"languagePreference": "Hebrew"
	}
	...
	]

200	{"errorMessage":"Request failed: server error"}

Balances Table

Contains data regarding the customer and accounts balances of the account

Each record specifies the balances of the account for the entire relevant banking domain used by the customer

Field Name	Description

Balance_Id	Unique identifier
Account_Number	Unique customer identifier
Validity_Date	Balance validity date specifies the date in which the balance was accurate in the specific account (yyyyMMdd)
NIS_Balance	Account NIS balance (double)
Dollar_Balance	Account USD balance (double)
Euro_Balance	Account Euror balance (double)
Credit_Balance	Account credit level (double)
Deposit_Balance	Account daily interest deposit (double)
Capital_Market_Balance	Sum of capital market balance of the account (double)

Example data

Field Name	Value
Balance_Id	1
Account_Number	9999
Validity_Date	20140601
NIS_Balance	10000
Dollar_Balance	10000
Euro_Balance	10000
Credit_Balance	10000
Deposit_Balance	10000
Capital_Market_Balance	10000

DynamoDB table name
Balances

1. Account Balance
Retrieve customer balance details by account number

Request

Method	URL
GET	api/account/{account_number}/balance

Type	Params	Values
HEAD HEAD	accessKey secretKey	string string

accessKey
The `accessKey` that was given by the Bank Open API

secretKey
The `secretKey` that was given by the Bank Open API

account_number
The account number for which we want to get the account balance.

Response

Status	Response
200	{ "accountNumber": 209963, "balanceId": "62264", "validityDate": "20140601",

	<pre>"nisBalance": "10000",
	"dollarBalance": "10000",
	"euroBalance": "10000",
	"creditBalance": "10000",
	"depositBalance": "10000",
	"capitalMarketBalance": "10000"
	}</pre>

200	{"errorMessage":"Request failed: server error"}

Transactions Table

Contains customer account financial transactions.
Each record represents one transaction.

Field Name	Description
Transaction_Id	Transaction unique identifier
Account_Number	Unique customer identifier
Transaction_Date	Transaction Date: yyyyMMdd
Transaction_Description	Short transaction description
Transaction_Time	hhmm
Transaction_Amount	double
Transaction_Currency	NIS / USD / EUR / etc…
Credit_Debit_Code	Positive / Negative (1 / 2 accordingly)

Transaction_Group_Description	Transaction type specifies the banking domain in which the transaction is associated with. Values: Capital market, Credit ,Deposits, Foreign Exchange conversion, etc For example: transaction of acquisition of a stock market share will associated to capital market domain Current / capital markets / checking / Loans

Example data

Field Name	Value
Transaction_Id	140101090001520413
Account_Number	280696
Transaction_Date	20140101
Transaction_Description	Cash Withdrawls
Transaction_Time	2352
Transaction_Amount	200.00
Transaction_Currency	NIS
Credit_Debit_Code	1
Transaction_Group_Description	Current Accounts

DynamoDB table name

Transactions

Indexed Columns

Primary Hash Key:	Account_Number (Number)		
Primary Range Key:	Transaction_Id (Number)		

Local Secondary Indices			
Index Name	**Hash Key**	**Range Key**	**Projected Attributes**
Transaction_Amount-Index	Account_Number (Number)	Transaction_Amount (Number)	All
Transaction_Date-Index	Account_Number (Number)	Transaction_Date (String)	All
Transaction_Group_Description-Index	Account_Number (Number)	Transaction_Group_Description (String)	All

1. Transactions for Account by Dates

Returns a JSON Array containing the transactions for an account with Account_Number, between "fromDate" to "toDate"

Request

Method	URL
`GET`	`api/account/{account_number}/transaction/by-date`

Type	Params	Values	
HEAD HEAD GET GET	accessKey secretKey fromDate toDate	string string string string	[optional] [optional]

accessKey

The `accessKey` that was given by the Bank Open API

secretKey
The `secretKey` that was given by the Bank Open API

account_number
The account number for which we want to get the data.

fromDate
Start of date range (yyyyMMdd)

toDate
End of date range (yyyyMMdd)

Response

Status	Response

200	[
	{
	"accountNumber": 179577,
	"transactionId": 140102090004860830,
	"transactionDate": "20140102",
	"transactionDescription": "Isracard",
	"transactionAmount": 2945.71,
	"transactionCurrency": "NIS",
	"creditDebitCode": "2",
	"transactionGroupDescription": “Capital Markets"
	},
	{
	"customerId": 179577,
	"transactionId": 140103090005657540,
	"transactionDate": "20140103",
	"transactionDescription": "Interest",
	"transactionAmount": 13.67,
	"transactionCurrency": "NIS",
	"creditDebitCode": "1",
	"transactionGroupDescription": "Loans"
	}
	]

200	{"errorMessage":"Request failed: server error"}

2. Transactions for Account by Group Description

Returns a JSON Array containing the transactions for an account with Account_Number, with the given group description.

Request

Method	URL
GET	api/account/{account_number}/transaction/by-group-description

Type	Params	Values
HEAD HEAD GET	accessKey secretKey transactionGroupDescriptio	string string string

accessKey

The `accessKey` that was given by the Bank Open API

secretKey

The `secretKey` that was given by the Bank Open API

account_number

The account number for which we want to get the data.

transactionGroupDescription

Current / capital markets / checking / Loans / Credit Cards

Response

Status	Response

200	<pre>[
	{
	"accountNumber": 35176,
	"transactionId": 140418090001979220,
	"transactionDate": "20140418",
	"transactionDescription": "Securities-Sale",
	"transactionAmount": 45487.67,
	"transactionCurrency": "NIS",
	"creditDebitCode": "1",
	"transactionGroupDescription": “Capital Markets”</pre>

	<pre>},
	{
	"accountNumber": 35176,
	"transactionId": 140418090001979220,
	"transactionDate": "20140418",
	"transactionDescription": "Securities-Sale",
	"transactionAmount": 88373.92,
	"transactionCurrency": "NIS",
	"creditDebitCode": "1",
	"transactionGroupDescription": "Capital Markets"
	}
	]</pre>

200	<pre>{"errorMessage":"Request failed: server error"}</pre>

3. Transactions for Account by Amount

Returns a JSON Array containing the transactions for an account with amount between the given values.

Request

Method	URL
GET	api/account/{account_number}/transaction/by-amount

Type	Params	Values	
HEAD HEAD GET GET	accessKey secretKey fromAmount toAmount	string string double double	[optional] [optional]

accessKey
The `accessKey` that was given by the Bank Open API

secretKey
The `secretKey` that was given by the Bank Open API

account_number
The account number for which we want to get the data.

fromAmount
Start of amount range

toAmount End of
amount range

Response

Status	Response

200	[
	{
	"accountNumber": 35176,
	"transactionId": 140528090004232380,
	"transactionDate": "20140528",
	"transactionDescription": "Checkbooks",
	"transactionAmount": 18,
	"transactionCurrency": "NIS",
	"creditDebitCode": "2",
	"transactionGroupDescription": "Checks"
	}
	]

200	{"errorMessage":"Request failed: server error"}

4. Net Income for Account for Month

Retrieve the net income for an account for a specific month

Request

Method	URL
GET	api/account/{account_number}/net-income-for-month

Type	Params	Values	
HEAD HEAD GET	accessKey secretKey date	string string string	[mandatory]

accessKey
The `accessKey` that was given by the Bank Open API

secretKey
The `secretKey` that was given by the Bank Open API

account_number
The account number for which we want to get the net income for month for.

date
The month for which we want the net income (yyyyMM).

Response

Status	Response

200	{
	"accountNumber": "225407",
	"date": "201401",
	"netIncome": 2471.7703
	}

200	{"errorMessage":"Request failed: server error"}

Credit Cards Table

Contains data regarding the customer credit card purchases.

Each record specifies a credit card purchase of the customer. The purchase data that includes the purchase amount, the number of credit payments payment category allows you to get a full understanding of the customer expenses in credit card wise

Field Name	Description
Charge_Id	Unique identifier
Account_Number	Unique customer identifier
Purchase_Amount	Total purchase amount (double)
Number_Of_Payments_In_Purchase	Specifies how many equal credit payments will be in the specific purchase
Purchase_Date	Credit card purchase date (yyyyMMdd)
Purchase_Time	Credit card purchase time (hh:mm:ss)
Business_Name	Specifies the point of sale in which the purchase was made. Notice that the name is encrypted for confidential purposes

Payment_Category	Classify the customer expenses according to main expenditure types: Recreation, entertainment and restaurants, education and family, culture and leisure, food and beverages, clothing and footwear, beauty and fashion accessories, health, electronics and communications, Home & Garden, insurance, automotive and transportation, taxes and fees, loans and mortgages, commissions, various checks, cash, gifts and souvenirs
Business_Location	Indicates The location in which the purchase was made based on h business the demographical Area (based on the Central Bureau of Statistics Israel

Example data

Field Name	Value
Charge_Id	1
Account_Number	975
Purchase_Amount	275.03
Number_Of_Payments_In_Purchase	1
Purchase_Date	20140416
Purchase_Time	2316
Business_Name	Business 6422583
Payment_Category	Gas Stations
Business_Location	50000913

DynamoDB table name

CreditCards4

Indexed Columns

Primary Hash Key:	Account_Number (Number)		
Primary Range Key:	Charge_Id (Number)		
Local Secondary Indices			
Index Name	**Hash Key**	**Range Key**	**Projected Attributes**
Purchase_Date-Index	Account_Number (Number)	Purchase_Date (String)	All
Payment_Category-Index	Account_Number (Number)	Payment_Category (String)	All
Purchase_Amount-Index	Account_Number (Number)	Purchase_Amount (Number)	All

1. Credit Card Transactions for Account by Dates

Returns a JSON Array containing the credit card transactions for an account with account number, between "fromDate" to "toDate"

Request

Method	URL
`GET`	`api/account/{account_number}/credit-card-transaction/by-date`

Type	Params	Values	
HEAD HEAD GET GET	accessKey secretKey fromDate toDate	string string string string	[optional] [optional]

accessKey
The `accessKey` that was given by the Bank Open API

secretKey
The `secretKey` that was given by the Bank Open API

account_number
The account number for which we want to get the credit card transactions.

fromDate
Start of date range (yyyyMMdd)

toDate
End of date range (yyyyMMdd)

Response

Status	Response
200	[{ "accountNumber": 209963, "chargeId": 6608999, "purchaseAmount": 50, "numberOfPaymentsInPurchase": "1", "purchaseDate": "20141201", "purchaseTime": "1652", "businessName": "Business 4972210", "paymentCategory": "Delis", "businessLocation": "50000913"

<table>
<tr><td></td><td><pre>
        },
        {
              "accountNumber ": 209963,
              "chargeId": 6609000,
              "purchaseAmount": 289.96,
              "numberOfPaymentsInPurchase": "1",
              "purchaseDate": "20141201",
              "purchaseTime": "0934",
              "businessName": "Business 3986054",
              "paymentCategory": "Gas Stations",
              "businessLocation": "50000913"
        },
        {
              "accountNumber ": 209963,
              "chargeId": 6609001,
              "purchaseAmount": 186.23,
              "numberOfPaymentsInPurchase": "1",
              "purchaseDate": "20141201",
         "purchaseTime": "1722",
         "businessName": "Business 1870835 ",
         "paymentCategory": "Life Insurance Companies",
         "businessLocation": "50000913"
        }
]
</pre></td></tr>
<tr><td>200</td><td><pre>{"errorMessage":"Request failed: server error"}</pre></td></tr>
</table>

2. Credit Card Transactions for Account by Amount

Returns a JSON Array containing the credit card transactions for an account with account number , with amounts between fromAmount and toAmount

Request

Method	URL
GET	api/account/{account_number}/credit-card-transactions/by-amount

Type	Params	Values	

HEAD HEAD GET GET	accessKey secretKey fromAmount toAmount	string string double double	 [optional] [optional]

accessKey
The `accessKey` that was given by the Bank Open API

secretKey
The `secretKey` that was given by the Bank Open API

account_number
The account number for which we want to get the credit card transactions

fromAmount
Start of amount range

toAmount
End of amount range

Response

Status	Response

200	[
	{
	"accountNumber": 35176,
	"chargeId": 2860125,
	"purchaseAmount": 10.41,
	"paymentAmount": "10.41",
	"numberOfPaymentsInPurchase": "0",
	"numberOfPayment": "0",
	"purchaseDate": "20140629",
	"businessName": "Business 4733602",
	"paymentCategory": "Parking Lots",
	"businessLocation": "Supplier_address missing"
	},
	{
	"accountNumber": 35176,
	"chargeId": 2860090,
	"purchaseAmount": 15.9,
	"paymentAmount": "15.90",
	"numberOfPaymentsInPurchase": "0",
	"numberOfPayment": "0",
	"purchaseDate": "20140429",
	"businessName": "Business 4733602",
	"paymentCategory": "Parking Lots",
	"businessLocation": "Supplier_address missing"
	}
	...
	]

200	{"errorMessage":"Request failed: server error"}

3. Credit Card Transactions for Account by Payment Category

Returns a JSON Array containing the credit card transactions for an account with account number, in a specific paymentCategory.

Request

Method	URL

GET	api/account/{account_number}/transaction/by-payment-category

Type	Params	Values
HEAD HEAD GET	accessKey secretKey paymentCategory	string string string

accessKey
The `accessKey` that was given by the Bank Open API

secretKey
The `secretKey` that was given by the Bank Open API

account_number
The account number for which we want to get the credit card transactions.

paymentCategory
Expeniture type category for account expenses

Response

Status	Response

200	[{ "accountNumber": 35176, "chargeId": 2860085, "purchaseAmount": 264.8, "paymentAmount": "264.80", "numberOfPaymentsInPurchase": "0", "numberOfPayment": "0", "purchaseDate": "20140413", "businessName": "Business 4593343", "paymentCategory": "Footwear", "businessLocation": "Supplier_address missing"
	}]
200	{"errorMessage":"Request failed: server error"}

Lams Locations Table

A simple table containing the polygons of each Agas code.

Field Name	Description
locationId	The Unique Agas code
coordinates	The polygon coordinates

Example data

Field Name	Value
locationId	96000013
coordinates	35.07767208,32.85059184 35.077743,32.850654569999989 ...

DynamoDB table name
LamasLocations

1. Get polygon by Agas code.
Returns a JSON containing the locationId (Agas code) and polygon's coordinates.

Request

Method	URL
`GET`	`api/lamas/{agas_code}`

Type	Params	Values	
HEAD HEAD	accessKey secretKey	string string	

accessKey
The `accessKey` that was given by the Bank Open API

secretKey
The `secretKey` that was given by the Bank Open API

agas_code
The agas code we want to get the coordinates to

Response

Status	Response

200	{ "locationId": 700426, "coordinates": "34.63060068,31.79030085 34.631128079999989,31.7899323 …. }
200	{"errorMessage":"Request failed: server error"}

2. Get polygons by multiple Agas codes.

Returns a JSON array containing the locationIds (Agas codes) and polygons' coordinates. It is mostly useful when one would like to draw multi regions on a map.

Request

Method	URL
GET	api/lamas/by-multiple-numbers

Type	Params	Values	
HEAD HEAD GET	accessKey secretKey agasNumbers	string string string	

accessKey
The `accessKey` that was given by the Bank Open API

secretKey
The `secretKey` that was given by the Bank Open API

agasNumbers comma separated values, each one representing the Agas code

Response

Status	Response

200	[
	{
	"locationId": 700426,
	"coordinates": "34.63060068,31.79030085
	34.631128079999989,31.7899323 ….
	},
	{
	"locationId": 251,
	"coordinates": "34.623549,31.776386039999988
	34.623662759999988,31.77640359
	…
	}
	]

200	{"errorMessage":"Request failed: server error"}

3. Get polygon and Agas code by geo-point.

Returns a JSON containing the locationId (Agas code) and polygon's coordinates where the point is located.
This is mainly useful when one is developing a location aware up and wants to query for the Agas code where the user is or something similar.

Request

Method	URL
GET	api/lamas/by-point

Type	Params	Values	

HEAD HEAD GET GET	accessKey secretKey lon lat	string string double double	

accessKey
The `accessKey` that was given by the Bank Open API

secretKey
The `secretKey` that was given by the Bank Open API

lon
The longtitude part of the coordinate

lat
The latitude part of the coordinate

Response

Status	Response

200	{
	"location": {
	"coordinates": [
	[
	[
	34.63060068,
	31.79030085
	],
	[
	34.63112807999999,
	31.79030085
	],
	...
	]
	],
	"type": "Polygon"
	},
	"_id": 248
	}

200	{"errorMessage":"Request failed: server error"}

Geo-Location Branches/Bankats Queries

1. Branches Nearby

Returns a JSON Array containing the all branches nearby sorted by distance

Request

Method	URL
GET	api/geo/branch/by-point

Type	Params	Values	
HEAD HEAD GET GET	accessKey secretKey lat lon	string string double double	[mandatory] [mandatory]

accessKey
The `accessKey` that was given by the Bank Open API

secretKey
The `secretKey` that was given by the Bank Open API

lat
latitude of the point

lon longitude of the point

Response

Status	Response
200	[
	{ "longitude": 34.78228, "latitude": 32.06285, "street": "Rival", "city": "Tel Aviv", "houseNum": 7, "shemSnif": "Hagalil", "distanceTo": 0.33081773678119775, "snifNMumber": 507 }, { "longitude": 34.77538, "latitude": 32.06405, "street": "Rotschild Avenue", "city": "Tel Aviv", "houseNum": 50, "shemSnif": "Main Branch", "distanceTo": 0.437194545224093, "snifNMumber": 170 } ...]
200	{"errorMessage":"Request failed: server error"}

2. Branches in City

Returns a JSON Array containing all the branches in the specified city.

Request

Method	URL
GET	api/geo/branch/by-city

Type	Params	Values
HEAD	accessKey	string
HEAD GET	secretKey city	string string

accessKey
The `accessKey` that was given by the Bank Open API

secretKey
The `secretKey` that was given by the Bank Open API

city
The city for which we want to see the branches

Response

Status	Response

200	[
	{
	"longitude": 34.77538,
	"latitude": 32.06405,
	"street": "Rotschild Ave.",
	"city": "Tel Aviv",
	"houseNum": 50,
	"shemSnif": "Main Branch",
	"distanceTo": 0,
	"snifNMumber": 170
	},
	{
	"longitude": 34.76388,
	"latitude": 32.06435,
	"street": "Hamered",
	"city": "Tel Aviv",
	"houseNum": 29,
	"shemSnif": "Bet Ha-Ta'asiya",
	"distanceTo": 0,
	"snifNMumber": 208
	}
	...
	]

200	{"errorMessage":"Request failed: server error"}

3. Bankats Nearby

Returns a JSON Array containing all the bankats nearby sorted by distance

Request

Method	URL
GET	api/geo/bankat/by-point

Type	Params	Values	
HEAD HEAD GET GET	accessKey secretKey lat lon	string string double double	[mandatory] [mandatory]

accessKey
The `accessKey` that was given by the Bank Open API

secretKey
The `secretKey` that was given by the Bank Open API

lat
latitude of the point

lon longitude of the point

Response

Status	Response
200	[

	{
	"longitude": 34.78015,
	"latitude": 32.06367,
	"street": "Menahem Begin",
	"city": “Tel Aviv Yafo",
	"houseNum": 23,
	"distanceTo": 0.24595808473881153,
	"snifNMumber": 159,
	"addressString": “23 Menahem Begin St.,Tel Aviv, 66183",
	"withdrawSms": "1"
	},
	{
	"longitude": 34.78228,
	"latitude": 32.06285,
	"street": "Rival",
	"city": "Tel Aviv",
	"houseNum": 7,
	"distanceTo": 0.33081773678119775,
	"snifNMumber": 507,
	"addressString": "7 Rival St., Tel Aviv, 67778”,
	"withdrawSms": "1"
	}
	...
	]

200	{"errorMessage":"Request failed: server error"}

4. Bankats in City

Returns a JSON Array containing all the bankats in the specified city.

Request

Method	URL
GET	api/geo/branch/by-city

Type	Params	Values

HEAD HEAD GET	accessKey secretKey city	string string string

accessKey
The `accessKey` that was given by the Bank Open API

secretKey
The `secretKey` that was given by the Bank Open API

city
The city for which we want to see the bankats

Response

Status	Response

200	[{ "longitude": 34.77538, "latitude": 32.06405, "street": "Rotschild Ave.", "city": "Tel Aviv", "houseNum": 50, "distanceTo": 0, "snifNMumber": 170, "addressString": " 50 Rotschild Ave., Tel Aviv, 61000", "withdrawSms": "1" }, { "longitude": 34.76388, "latitude": 32.06435, "street": "Hamered", "city": "Tel Aviv", "houseNum": 29, "distanceTo": 0, "snifNMumber": 208, "addressString": "29 Hamered St., Tel Aviv, 68125",
	"withdrawSms": "1" } ...]
200	{"errorMessage":"Request failed: server error"}

Real World Account Details

The Open API exposes real-world financial details of 10 representative bank account holders.

bank hapoalim

The account archetypes are as follows:

Account Numbers– 200004 / 200007
Male / Female – Pension age (~67), "Private" Account Type with a "High" Credit Rate.

Account Numbers – 200009 / 200010 / 200012
Male / Female – Young adult (age 23), Student with "Single" as the marital status.

Account Numbers – 200001 / 200003 / 200008 / 200013
Male / Female – Adult (age 30), with "Single" as the marital status.

Account Number - 200011
Male / Female – Adult (age 45), married with a standing loan.

Field Name	Description
Account_Number	Random Value
Branch_Number	Random Value
Branch_Name	
Branch_City	
Branch_Address	
Account_Open_Date	In format DDMMYYYY
Account_Type_Description	Account Classification: Private / Business
Customer_Salary	Salary
Customer_Name	
Customer_Age	
Customer_Gender	
Language_Preference	

Field Name	Description
Balance_Id	Unique identifier
Account_Number	Unique customer identifier
Validity_Date	Balance validity date specifies the date in which the balance was accurate in the specific account (yyyyMMdd)
NIS_Balance	Account NIS balance (double)
Dollar_Balance	Account USD balance (double)
Euro_Balance	Account Euror balance (double)
Credit_Balance	Account credit level (double)
Deposit_Balance	Account daily interest deposit (double)
Capital_Market_Balance	Sum of capital market balance of the account (double)

Field Name	Description
Transaction_Id	Transaction unique identifier
Account_Number	Unique customer identifier
Transaction_Date	Transaction Date: yyyyMMdd

Transaction_Description	Short transaction description
Transaction_Time	hhmm
Transaction_Amount	double
Transaction_Currency	NIS / USD / EUR / etc…
Credit_Debit_Code	Positive / Negative (1 / 2 accordingly)
Transaction_Group_Description	Transaction type specifies the banking domain in which the transaction is associated with. Values: Capital market, Credit ,Deposits, Foreign Exchange conversion, etc For example: transaction of acquisition of a stock market share will associated to capital market domain Current / capital markets / checking / Loans

Field Name	**Description**
Purchase_Number	Unique identifier
Account_Number	Unique customer identifier
Purchase_Amount	Total purchase amount (double)
Payment_Amount	The current payment amount
Number_Of_Payments_In_Purchase	Specifies how many equal credit payments will be in the specific purchase
Number_Of_Payment	The current payment number
Purchase_Date	Credit card purchase date (yyyyMMdd)
Purchase_Time	Credit card purchase time (hh:mm:ss)

Business_Name	Specifies the point of sale in which the purchase was made. Notice that the name is encrypted for confidential purposes
Payment_Category	Classify the customer expenses according to main expenditure types Recreation, entertainment and restaurants, education and family, culture and leisure, food and beverages, clothing and footwear, beauty and fashion accessories, health, electronics and communications, Home & Garden, insurance, automotive and transportation, taxes and fees, loans and mortgages, commissions, various checks, cash, gifts and souvenirs
Business_Location	Indicates The location in which the purchase was made based on h business the demographical Area (based on the Central Bureau of Statistics Israel
Card_Type	The type of card (Isracard hybrid)

1. Get Real Account Details

Returns a JSON Object containing the real details of a given account

Request

Method	URL
GET	api/real/{account_number}/details

Type	Params	Values
HEAD HEAD	accessKey secretKey	string string

accessKey
The `accessKey` that was given by the Bank Open API

secretKey
The `secretKey` that was given by the Bank Open API

account_number
The account number for which we want to get the real account details.

Response

Status	Response

200	{
	"accountNumber": 200013,
	"branchNumber": "557",
	"branchCity": "Ramat Gan",
	"branchAddress": "Yossef Sapir 9",
	"branchName": "Kiryat Kernitzki branch",
	"accountOwnerAddress": "86000414",
	"accountOpenDate": "1999",
	"accountTypeDescription": "Private",
	"customerName": "Customer 13",
	"customerAge": 30,
	"customerSalary": null,
	"customerGender": "M",
	"languagePreference": "Hebrew"
	}

200	{"errorMessage":"Request failed: server error"}

2. Real Account Balance

Returns a JSON Object containing the real balance details of a given account

Request

Method	URL
GET	api/real/{account_number}/balance

Type	Params	Values
HEAD HEAD	accessKey secretKey	string string

accessKey
The `accessKey` that was given by the Bank Open API

secretKey
The `secretKey` that was given by the Bank Open API

account_number
The account number for which we want to get the account balance.

Response

Status	Response

200	{
	"balanceId": "11",
	"accountNumber": 200013,
	"validityDate": "20140630",
	"nisBalance": "₪ -624.20",
	"dollarBalance": "$254.51",
	"euroBalance": "€ 0.00",
	"creditBalance": "₪ 0.00",
	"depositBalance": "₪ 0.00",
	"capitalMarketBalance": "₪ 15,497.38"
	}

200	{"errorMessage":"Request failed: server error"}

3. Real Account Credit Card Transactions

Returns a JSON Array containing the real credit card transactions' details of a given account

Request

Method	URL
GET	api/real/{account_number}/credit-cards

Type	Params	Values
HEAD HEAD	accessKey secretKey	string string

accessKey
The `accessKey` that was given by the Bank Open API

secretKey
The `secretKey` that was given by the Bank Open API

account_number
The account number for which we want to get the real account credit card transactions.

Response

Status	Response
200	[{ "purchaseNumber": 1, "accountNumber": 200013, "purchaseAmount": "2,800.00", "paymentAmount": "2,800.00", "numberOfPaymentsInPurchase": 0, "numberOfPayment": 0,

	"purchaseDate": "20140331",
	"paymentDate": "20140420",
	"businessName": "Business 4593343",
	"paymentCategory": "Assorted",
	"businessLocation": "Supplier_address missing",
	"cardType": "Isracard Hybrid"
	},
	{
	"purchaseNumber": 2,
	"accountNumber": 200013,
	"purchaseAmount": "36",
	"paymentAmount": "36",
	"numberOfPaymentsInPurchase": 0,
	"numberOfPayment": 0,
	"purchaseDate": "20140501",
	"paymentDate": "20140520",
	"businessName": "Business 6694891",
	"paymentCategory": "Entertainment",
	"businessLocation": "Supplier_address missing",
	"cardType": "Isracard Hybrid"
	}
	...
	]

200	{"errorMessage":"Request failed: server error"}

4. Real Account Transactions

Returns a JSON Array containing the real transactions' details of a given account

Request

Method	URL
GET	api/real/{account_number}/transactions

Type	Params	Values

HEAD HEAD	accessKey secretKey	string string

accessKey
The `accessKey` that was given by the Bank Open API

secretKey
The `secretKey` that was given by the Bank Open API

account_number
The account number for which we want to get the real account transactions.

Response

Status	Response

200	[
	{
	"transactionId": 140407090001681000,
	"accountNumber": 200013,
	"transactionDate": "20140407",
	"transactionTime": "1111",
	"transactionDescription": “Deposit-Withdrawals",
	"transactionAmount": "4,406.67",
	"transactionCurrency": "NIS",
	"creditDebitCode": "1",
	"transactionGroupDescription":"Deposits"
	},
	{
	"transactionId": 140420090001942000,
	"accountNumber": 200013,
	"transactionDate": "20140420",
	"transactionTime": "0",
	"transactionDescription": "Isracard",
	"transactionAmount": "2,801.00",
	"transactionCurrency": "NIS",
	"creditDebitCode": "2",
	"transactionGroupDescription": "Credit Cards"
	}
	...
	]

200	{"errorMessage":"Request failed: server error"}

5. Real Account Full Data

Returns a JSON Object containing all available real data for a given account number

Request

Method	URL
GET	api/real/{account_number}/all

Type	Params	Values
HEAD HEAD	accessKey secretKey	string string

accessKey
The `accessKey` that was given by the Bank Open API

secretKey
The `secretKey` that was given by the Bank Open API

account_number
The account number for which we want to get the real full account data.

Response

Status	Response

200	{
	"account": {
	"accountNumber": 200013,
	"branchNumber": "557",
	"branchCity": "Ramat Gan",
	"branchAddress": "9 Yossef Sapir",
	"branchName": "Kiryat Kernizki branch",
	"accountOwnerAddress": "86000414",
	"accountOpenDate": "1999",

```
        "accountTypeDescription": "Private",
 "customerName": "Customer 13",
 "customerAge": 30,
 "customerSalary": null,
        "customerGender": “M”,
 "languagePreference": "Hebrew"
},
"balance": {
 "balanceId": "11",
 "accountNumber": 200013,
 "validityDate": "20140630",
  "nisBalance": "₪ -624.20",
 "dollarBalance": "$254.51",
 "euroBalance": "€ 0.00",
 "creditBalance": "₪ 0.00",
        "depositBalance": "₪ 0.00",
        "capitalMarketBalance": "₪ 15,497.38"
},
"creditCardsList": [
 {
            "purchaseNumber": 1,
             "accountNumber": 200013,
             "purchaseAmount": "2,800.00",
             "paymentAmount": "2,800.00",
             "numberOfPaymentsInPurchase": 0,
             "numberOfPayment": 0,
             "purchaseDate": "20140331",
            "paymentDate": "20140420",
             "businessName": "Business 4593343",
          "paymentCategory": "Assorted",
        "businessLocation": "Supplier_address missing
              ”cardType”: “Isracard Hybrid”"
 },
 {
            "purchaseNumber": 2,
             "accountNumber": 200013,
             "purchaseAmount": "36",
             "paymentAmount": "36",
             "numberOfPaymentsInPurchase": 0,
             "numberOfPayment": 0,
             "purchaseDate": "20140501",
            "paymentDate": "20140520",
```

```
                "businessName": "Business 6694891",
              "paymentCategory": "Entertainment",
            "businessLocation": "Supplier_address missin
          “cardType”: “Isracard Hybrid”
       }
                              ...
      ],
     "transactionsList": [
    {
           "transactionId": 140407090001681000,
            "accountNumber": 200013,
            "transactionDate": "20140407",
            "transactionTime": "1111",
            "transactionDescription": "Deposit-
Withdrawals",
           "transactionAmount": "4,406.67",
            "transactionCurrency": "NIS",
           "creditDebitCode": "1",
            "transactionGroupDescription": "Deposits"
    },
    {
           "transactionId": 140420090001942000,
            "accountNumber": 200013,
            "transactionDate": "20140420",
            "transactionTime": "0",
            "transactionDescription": "Isracard",
           "transactionAmount": "2,801.00",
            "transactionCurrency": "NIS",
           "creditDebitCode": "2",
            "transactionGroupDescription": "Credit Cards"
    }
                              ...
      ]
}
```

200	{"errorMessage":"Request failed: server error"}

Glossary

Conventions

- **Status** - HTTP status code of response.
- All the possible responses are listed under ‘Responses’ for each method. Only one of them is issued per request server.
- All response are in JSON format.
- All request parameters are mandatory unless explicitly marked as `[optional]`