# Common Core Banking System API

The IA-CCBS API (Aka i-Apply Common Core Banking System Application Program Interface) aims to interconnect I-Apply with the core banking system and the card platform if it exists in the Bank's environment. For further information, please refer to paragraph "Architecture" below.

Contents

[Common Core Banking System API 1](#_Toc146211265)

[Terms & Technologies 2](#_Toc146211266)

[API 2](#_Toc146211267)

[Open API 2](#_Toc146211268)

[RESTful API 2](#_Toc146211269)

[Architecture 2](#_Toc146211270)

[Green layer – Bank's involvement 3](#_Toc146211271)

[iApply light CCBS Integration 4](#_Toc146211272)

[The Common CBS API 4](#_Toc146211273)

[Account Integration 4](#_Toc146211274)

[Collateral Maintenance Integration 5](#_Toc146211275)

[Credit Limit 5](#_Toc146211276)

[Customer Integration 5](#_Toc146211277)

[Customer Position Integration 6](#_Toc146211278)

[Deposit Account Integration 6](#_Toc146211279)

[Fees 6](#_Toc146211280)

[LG Beneficiary 6](#_Toc146211281)

[Mortgage Integration 7](#_Toc146211282)

[API Implementation 7](#_Toc146211283)

[Default values 7](#_Toc146211284)

[The elementary interface 7](#_Toc146211285)

[How to start & resources 8](#_Toc146211286)

[Recommended OpenAPI Tools 9](#_Toc146211287)

## Terms & Technologies

### API

An application programming interface (API) defines the rules you must follow in order to communicate with other software systems. Developers expose or create APIs so that other applications can communicate with their applications programmatically.

### Open API

The Common-CBS API is an Open API compliant specification. The definition of an Open API does not need further explanation as it is an industry-standard. In brief, Open API can provide a definition of i-Apply API to other organizations (such as the Banking Software providers) who have to implement their services following that specification. Open API provides a standardized method to do this. Our API is described in agnostic terms, decoupling them from any specific programming language. Consumers of our API specification (usually the Banking Software providers) do not need to understand the core of our application. They can understand precisely what they need from our API specification, written in a simple and expressive language. The OpenAPI Specification (OAS) enables this knowledge transfer from the API provider to the API consumer. It is an open standard for describing your APIs, allowing you to provide an API specification encoded in a JSON or YAML document. It provides a comprehensive dictionary of terms that reflects commonly-understood concepts in the world of APIs, thus embedding the fundamentals of HTTP and JSON. Teamed up with supporting tools can provide a rich experience based on a simple document.

### RESTful API

RESTful API is an interface that two computer systems use to exchange information securely over the internet or a controlled network. Most business applications must communicate with other internal and third-party applications to perform various tasks. RESTful APIs support this information exchange because they follow secure, reliable, and efficient software communication standards.

## Architecture

The diagram d1 presents the interfaces layer and the communication between the layers. The i-Apply application server (cyan color on the diagram) sends business requests to the i-Apply CCBS layer (orange). The IA-CCBS performs mappings based on a) the Country, b) the Bank, and c) the actual CBS and performs a sequence of validations. When necessary, it retrieves data from the i-Apply database (IADB) and the i-Apply file system (IAFS). Preparing and using a request invokes a RESTful web service call to the CCBS API (green) layer. The Bank usually develops the green layer with the help of the CBS and Cards provider(s). The typical action is to call the Core Banking Software and do what the method (e.g., SearchCustomer, OpenAccount, etc.) aims to do. Finally, prepare a valid response and return it to the IA-CCBS (orange) layer. At that point, the IA-CCBS evaluates the returned response, creates logs, errors, and data mappings to i-Apply terms and codes, and performs the necessary data persistence. The last step is returning a response to the IA-CBS (orange) and that layer to the initiator (cyan layer).

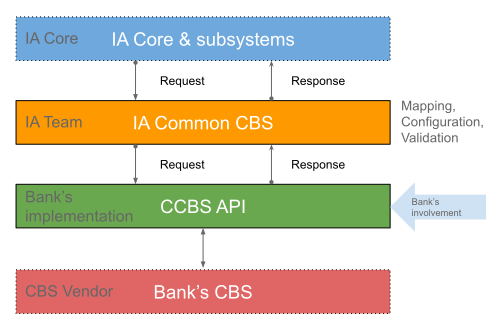


Diagram d1. Layers of CBS Interface

## Green layer – Bank's involvement

The green layer of the diagram is what the provider of the Banking Software must implement for the i-Apply to CBS interfacing. The orange layer of the architectural diagram d1 will do the following process:

1. Retrieve from the database further data if needed.
2. Map the i-Apply codes to Bank's-CBS-specific codes.
3. Invoke the RESTful API method call and handle the result.
4. In case of an error will return the error to the i-Apply
5. in case of a successful invocation, the Bank's-CBS result codes will be mapped to i-Apply codes and stored the results in the database (if necessary).
6. Will return that response back to the i-Apply system.

# iApply light CCBS Integration

The iApply CCBS Integration uses 99 API calls to offer integration between the Core Banking System (CBS) and the iApply platform. The light version uses only 53 integration points (API calls), organized into folders and priorities.

The **folder** indicates where the supplementary information of each call is stored on the file system, on GitHub, and in the OpenAPI interface.

The **Priority** indicates the sequence of the particular deliverables we expect from the side of the provider (in most cases, the vendor of the core banking system)—The Priority aims to offer a “Proof of Concept” and the elementary integrations between the two systems.

# The Common CBS API

The following text contains information regarding the Excel file and lists all the available API calls. Among others, it contains the following columns:

* **Colour:** only the green calls are available in the light iApply version.
* **Group** is the group's name in which this document's current chapter is organized.
* **Folder** is the name of the folder on the file systems (and other places) where the reader can find further information.
* **Priority** is the Priority mentioned above in the text.
* **iApply Interface** is the name of the API call.
* **Invocation** contains an indication of where the call is invocated by the iApply platform.
* **Client** contains some more information than the invocation regarding the API call.
* **Excel File Names** are the names of the Excel files the reader can find in the appropriate folder, containing the input and the output parameters of the API call (request and response)

## Account Integration

The account integration group contains interface API calls used to open and manipulate accounts to the CBS. In total, it contains 12 API calls organized into 3 priorities and the following folders:

1. **Repayment Schedule.** It is the set of calls aiming to create and retrieve repayment schedules as a simulation thus being an officials part of the application process.
2. **Drawdown.** Is the Partial disbursement of an amount of an application form.
3. **Account Integration.** Any other call is related to the account.
4. **LG.** Used to manipulate LG family of products. In some setups, it might be used to manipulate other similar products.

All the calls depend on a user action by selecting it on the CBS operations menu or by clicking a button.

The calls, marked as priority 2, are necessary for iApply/Light minimum CBS integration.

## Collateral Maintenance Integration

The collateral maintenance integration group contains interface API calls used to maintain the application form's collaterals. The collaterals can be movable, immovable, or any other type. In total, it contains 7 API calls organized in a single priority, and one folder.

All the calls depend on a user action by selecting it on the CBS operations menu. An exception is the API call "*DeleteCollateralMaintenanceDetails*," which executes automatically in the iApply operation "Delete Collateral Allocation, existing in CBS."

## Credit Limit

The credit limit group contains interface API calls to maintain the application form's collaterals. The collaterals can be movable, immovable, or any other type. In total, it contains 4 API calls organized in a single priority and one folder.

The limits are the same for all the products (e.g., Overdraft limit). Note that when iApply retrieves the limit simultaneously, it retrieves the available remaining amount.

All the calls depend on a user action by selecting it on the CBS operations menu.

## Customer Integration

The Customer Integration group contains interface API calls to retrieve and update or create the customer (applicant) and the other participants to the CBS. The integration can be simple or very complex when the iApply is used to manipulate all the customer details to the CBS. In total, it contains 14 API calls organized into 2 priorities and one folder. The term customer in iApply means any individual or legal entity participating in an application form with any of the roles of applicant, guarantor, representative, administrator, spouse, and many others. However, it is a matter of configuration and, in some cases, implementation of the kind of customer will be exchanged with the CBS.

The API calls under priority 1 are necessary for elementary cooperation between the CBS and the iApply/Light minimum CBS integration. There are mainly calls to search for an existing customer and retrieve back to the i-Apply the current instance of his details.

Most calls depend on a user action by selecting it on the CBS operations menu by clicking a button on a UI area. Only two calls are in-code orchestrated and executed automatically to maintain the customer addresses.

## Customer Position Integration

The Customer Position Integration group contains interface API calls used to retrieve the customer's existing position in the Bank. That information contains the existing exposures and any other obligations to the Bank related to the principal applicant or parties. Also, the information regarding the collaterals is already known to the bank products. Additionally, the loan insurance is expected to affect the results. In total, it contains 4 API calls organized in a single priority and one folder.

The API calls are under priority 1 and are necessary for elementary cooperation between the CBS and the iApply/Light minimum CBS integration. They are only retrieving information and do not perform any data modification.

All the calls depend on a user action by selecting it on the CBS operations menu by clicking a button or on a UI area.

## Deposit Account Integration

The Deposit Account Integration group contains interface API calls used to retrieve and maintain the deposit account on the side of the CBS. Only one call was used to retrieve information, and the others used to modify the accounts. In total, it contains 4 API calls organized in a single priority and one folder.

The same calls are used for similar types of accounts, such as the Service and the Overdraft accounts.

All the calls depend on a user action by selecting it on the CBS operations menu and clicking. The exception is one call to retrieve the existing account information when the user selects the UI as an overdraft account.

## Fees

The Fees group contains a few interface API calls used to maintain the fees on the side of the CBS. In total, it contains 2 API calls organized in a single priority and one folder.

All the calls depend on a user action by selecting it on the CBS operations menu.

## LG Beneficiary

The LG beneficiary group contains a few interface API calls used to retrieve and update the LG type of product beneficiaries on the side of the CBS. In total, it contains 2 API calls organized in a single priority and one folder.

All the calls depend on a user action by selecting it on the CBS operations menu.

## Mortgage Integration

The Mortgage Integration group contains interface API calls used to retrieve and update the information needed by the type of the mortgage family of products on the side of the CBS. In total, it contains 4 API calls organized in a single priority and one folder.

All the calls depend on a user action by selecting it on the CBS operations menu.

# API Implementation

Each API call has one expected response structure and returns one specific result structure. Both are described in two points:

* The Open API specification covers the total technical needs. Using online tools, the developer can convert iApply's request and response structures into programming language-specific data structures.
* The Excel files are organized in folders where the reader will find further information in texts and "last minute" corrections or clarifications.

## Default values

In both responses and results and when a field value is null, the CCBS subsystem expects that the vendor will specify (on request) and provide (on results) the default values coming up from the configuration of the CBS system for the specific Bank (solution) and the product-specific default values when this is applicable. Consider that note as crucial as the iApply does not reproduce the CBS setup or know the entire product factory.

## The elementary interface

With the term elementary interface, i-Apply means the necessary API methods that the Bank must implement to start working with the i-Apply. That crude interface is needed to set up and work in a fundamental way and a startup version of the i-Apply. Does not provide any of the necessary automation available in the entire API version but offers a starting point for the Bank. Later the rest of the API calls must be implemented by the Bank (or the CBS provider).

The list of the supported API calls is organized into groups, and there are the following:

* Group G00, with informative API calls. Used to explain to the i-Apply the CBS system and as a test for the connectivity.
* Group G01. Customer basic data. It is a set of methods to search a customer in the CBS and retrieve the primary data (such as demographics).
* Group G02. Position-related methods. Used to extract the customer's current obligations (position) to the Bank.

All the other API calls to the elementary interface are characterized as Group 99 (other calls).

## How to start & resources

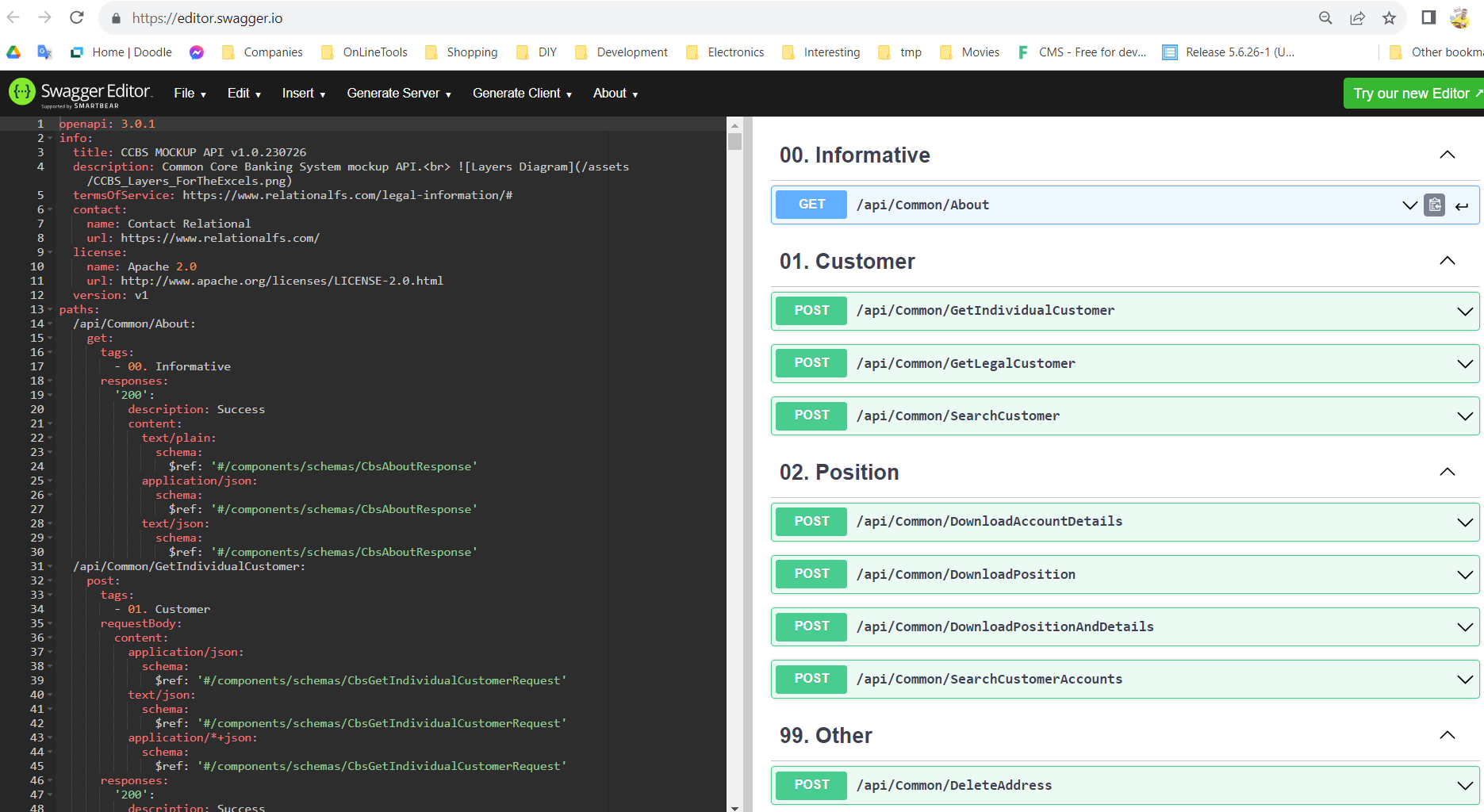
At the GitHub project iApply\_CCBS\_API you will find the OpenAPI JSON file.

Visit link: <https://github.com/aafent/iApply_CCBS_API> and download the file **CCBS\_API.json**

*Notice that we update the repository occasionally, so a git on it is recommended.*

Then we are strongly recommending to use the free service: <https://editor.swagger.io> to import the file or to import the following URL: <https://raw.githubusercontent.com/aafent/iApply_CCBS_API/main/CCBS_API.json>

You will get a picture similar to the following (screen s1):



Screen s1.

As you can see, the API Calls are organized into groups. In that screen shoot, the four (4) groups of the elementary interface are visible.

Actually, there are just 8 (eight) methods to implement for the implementation of the elementary interfaces.

The G00 contains just one method, the "about" The only you must do to return the following response:

**{**

**"title": "CBS Interface,"**

**"apiVersion": "1.0",**

**"code": "ACMEBank.SoftCompany.S1",**

**"message": null,**

**"success": true**

**}**

***Note*** *that in the title field, you can specify whatever you want to name that works, but the code has to be as in the example.*

For Groups G01 and G02, you must implement the described responses based on the given requests. Each method has a specific request and expects either an HTTP Error or a response of the particular structure (class). The responses and the requests are also available in the form of Excel files containing further information.

## Recommended OpenAPI Tools

Open API Tools. An extended list with several tools.

* <https://openapi.tools>

Swagger Tools. The industry standard to the OpenAPI presentation & authoring

* <https://swagger.io/tools/swagger-editor>
* <https://swagger.io/tools/swagger-codegen>

A collection of open-source and commercial tools for creating your APIs with OpenAPI

Sourced from and published for the community.

* <https://tools.openapis.org>