# Common Core Banking System API

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

## Terms & Technologies

### API

An application programming interface (API) defines the rules you must follow 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. What is 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 means 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 guts 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, 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 the a) country, b) the Bank, and c) the actual CBS and performs a sequence of validations. When necessary, 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).

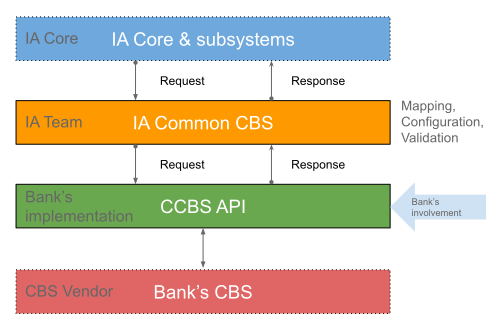


Diagram d1. Layers of CBS Interface

## 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.

## 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

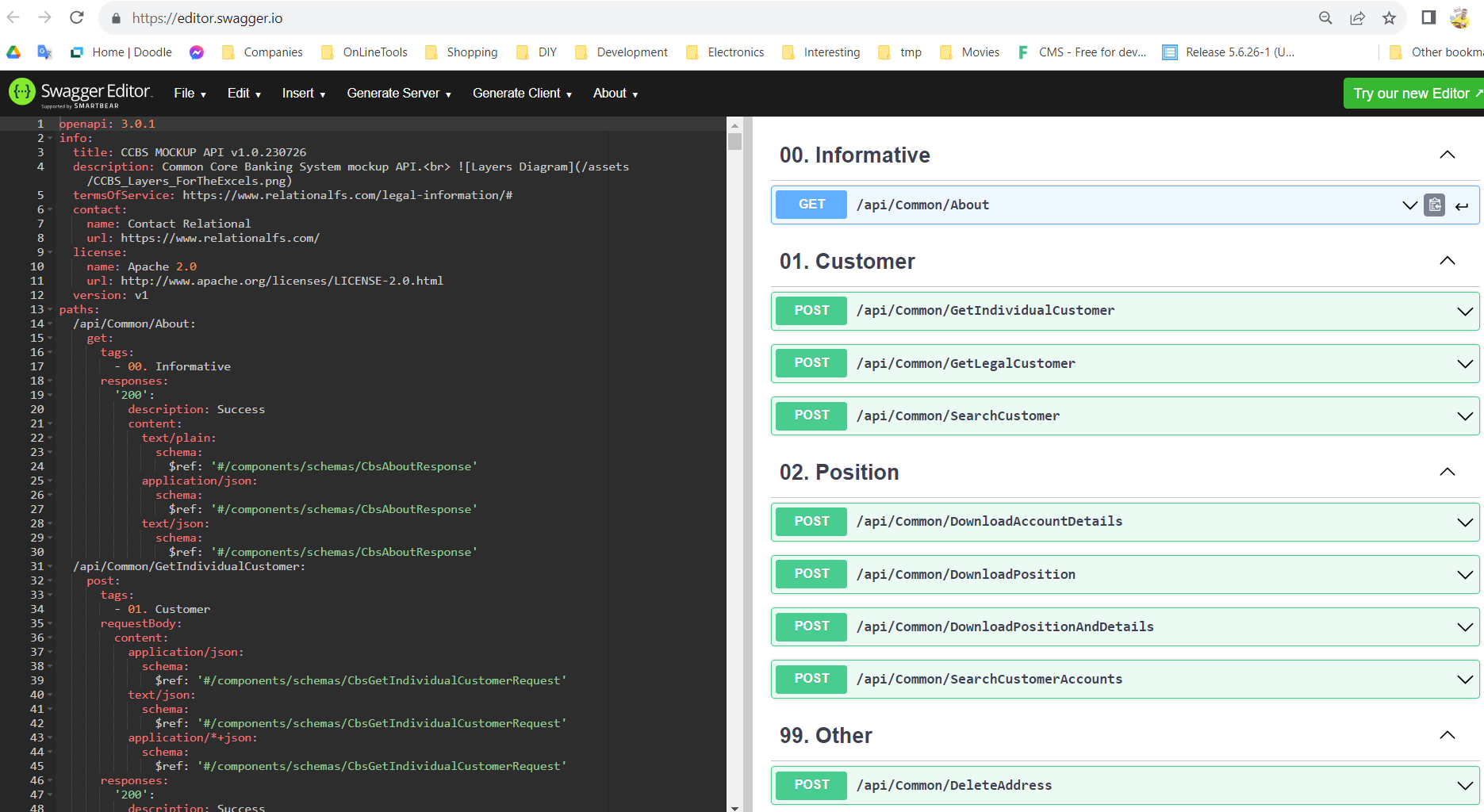
At the GitHub project iApply\_CCBS\_API you will find the OpenAPI JSON file.

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": "C-Soft Interface,"**

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

**"code": "ASSETBANK.CSOFT.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.