

# Comparing 3 ways to add account aggregation offering to your clients

TYPICAL TASKS OF AN ACCOUNT AGGREGATION APPLICATION	DO EVERYTHING ON YOUR OWN	USE AGGREGATOR <b>UNU<sub>API</sub> PLUS</b>	USE ADVANCED AGGREGATOR <b>UNU<sub>API</sub> PREMIUM</b>
1 Read and understand each one of the developers portals for the banks you will aggregate	Very high effort	Read just the authentication part	Read just the authentication part
2 Build a sample application to test the APIs	High effort	High effort	High effort
3 Register your sample application in the sandbox of each bank	High effort	Low - You will register just one set of APIs	Low - You will register just one set of APIs
4 Understand in detail the APIs of each bank (fields, Rules, Flows, replies, exceptions)	High effort	Low effort - Deal with just one API	Low effort - Deal with just one API
5 Understand in detail the APIs of UNUapi (fields, Rules, Flows, replies, exceptions)	No actions required	Medium effort	Medium effort
6 Integrate your application with the APIs of the connected banks	High effort	Low effort - Integrate just one set of APIs	Low effort - Integrate just one set of APIs
7 Tests your application with the APIs	Very high effort	Low effort - Test just one set of APIs	Low effort - Test just one set of APIs
8 Repeat above tests for the LIVE environment	Very high effort	Low effort - Test just one set of APIs	Low effort - Test just one set of APIs
9 Add more Banks in the future	Medium effort per bank	Low effort	Low effort
10 Periodically visit the developer portals of the connected banks for changes and maintain your applications	Very high effort	No actions required	No actions required
11 Build advanced functionality in your application (reconciliation, authorization workflows, currency conversions, categorization etc)	Very high effort	Very high effort	Low effort - APIs provided

## Which is the right choice for your application?

### DO EVERYTHING ON YOUR OWN

It can be a fitting option if your application needs to connect to one or two banks, at most. It can also be an option if you have a powerful developers team that will be dedicated to the banking API maintenance.

### USING **UNU<sub>API</sub> PLUS**

It is the best choice if you plan to build an aggregation application that will connect to banks from different geographies. It is your only choice if you have limited resources and you prefer to direct them to more advanced or strategic development tasks.

### USING **UNU<sub>API</sub> PREMIUM**

It is perfect for financial institutions or banks that need to go beyond account aggregation. It is the premium choice for those that need a quick implementation of a Business Financial Managers (BFM or PFM) on their portals, while at the same time they retain control on their code.

Onboard with us

