

Core-Connector template

To communicate with Payment Manager or Mojaloop Connector, DFSPs are required to implement Core Connector in alignment with their backend technology so that their backend can talk to the Payment Manager API. To help API implementation efforts, DFSPs are provided with:

- the specification of the Mojaloop Connector API
- a Core Connector template

What is a Core Connector template? DFSP Core Backends have their own custom API or enterprise integration solutions so multiple variants of Core Connector have been built to align with common Core Banking Solution (CBS) products. These Core Connector variants are all built using a standard template. The template provides a placeholder codebase for the API endpoints that need to be developed.

Making use of participation tools to reduce time and effort

The effort for a DFSP to customise a Core Connector template and build and deploy an integration, will differ depending on their chosen design, and how much of the participant tools they choose to make use of.

Core-connector Requirements

The core connector will need to support both Payer DFSP and Payee DFSP integrations. Mojaloop Connector API is a Mojaloop-centric API built around the resource types of the Mojaloop FSPIOP API but applying a use-case-oriented approach. The key resources of the Mojaloop FSPIOP API are the party lookup, quotes, and transfers services:

- **Party lookup service:** Identifying the DFSP serving the Payee and the Payee itself (= the recipient of funds in a transaction) based on a Payee identifier (typically a MSISDN, that is, a mobile number).
- **Quotes service:** Requesting a quote and exchanging cryptographic proof to prepare and secure the transfer. A quote is a contract between a Payer DFSP and Payee DFSP for a particular financial transaction before the transaction is performed. It guarantees the agreement set by the Payer and Payee DFSPs about the Payer, the Payee, and transfer amount, and is valid during the lifetime of a quote and transfer of a specified financial transaction.
- **Transfers service:** Executing the transaction as per the agreed details and cryptographic proof.

Within the context of the Mojaloop FSPIOP API, the transfer happens in three main steps corresponding to these three phases:

- Identifying the Payee (party lookup or discovery phase)
- Agreeing the transfer (quote or agreement phase)

- Executing the transfer (transfer phase)

Using Mojaloop Connector and the Mojaloop Connector API, the DFSP has the choice to perform the transfer process in one combined step, in two steps, or three steps.

For more information on the three phases of a transfer in Mojaloop, see Module 2: Mojaloop in greater detail of [Mojaloop training course](#) DFSP-101.

Design Documentation

It best practice to design integrations using sequence diagrams. [Here](#) is a repository of sequence diagrams that can be used as as starting point to build the detailed integration design. It is best practice to include these designs in the core-connector repository along side the code that implement the designs.

Usage guide

It is best practice to include a Readme.md file that describes how the use implement and contribute to the core-connector being created.

Aligning with deployments

The deployment stack and integrations into Payment manager rely on a pre-built docker image of the core connector. The template project create an easy mechanism for doing this.

Aligning with standardized quality assurance

Another participation tool is the core-connector testing framework. This is useful as it support a style of test driven development. It allows a review of the technical requirements, and then a tests with visibility and asserts, the integration being developed. It is likely that the your scheme will use the same tool for testing your integration.