Konnektoren Allgemein

The idea behind connectors is to have a facility that acts as a data adapter between some source data and a destination output, which allows for easy export/import of data from/to Webdesk.

This structure is built around two basic types of connectors:

- source connectors
- destination connectors

which allow a wide range of configurations to be set up, and a lot of flexibility when importing/exporting data.

Source connectors take their input data and transform it to a generic output. In other words, they transform data from a well known structure and create an output in a generic form to be consumed by others. The input data can come from a variety of sources:

- a CSV file
- a persistence entity in the system
- a specific table in a database
- etc.

The output will always take the form of a generic map <field name, field value>.

On the other hand, **destination connectors** do the opposite and take a generic input and transform it to a well known output structure.

The output structure can take several forms:

- a file
- a specific entity
- a specific table in a database
- etc.

The input is a generic map of the form <field name, field value>.

With these two components we have the basics to set up a powerful tool for data transfers from/to Webdesk in a wide variety of formats. The source and destination connectors are bound to each other using a **connector link**, which specifies the source and destination connector to be used and, specifically, the mapping between the fields that allows for data in the input format to be expressed as data in the output format.

This means that the configuration for an end-to-end connector is as follows

SOURCE # SourceConnector # DestinationConnector # DESTINATION

Example:

let's say that we want to import data of persons stored in a file. In this context, the SOURCE is the file we're reading the data from, and the DESTINATION is the PoPerson entity in the Webdesk system. So, for this import we could set up a SeparatorFileConnector with the appropriate configuration to read the data from a CSV file, then we would need to link the created configuration to a PoPersonConnector (since this connector is not configureable, no configuration is needed in this step). Once the link is created, we need to map the fields that we specified in the file connector to the appropriate fields in the PoPerson connector, and only then can we synchronise this connector link to perform the actual import of data. Furthermore, a connector link can be synchronised periodically using a job.