BIS

Fusion Metadata Registry
A schema registry for Python DataFrames

When working with Pandas in Python data pipelines, it's useful to check whether DataFrames match the expected schema. It allows errors to be detected early and avoids the propagation of bad data.

The *pandera* Python package provides a way to do just that: specify the expected schema and validate a DataFrame for compliance.

```
import pandas as pd
import pandera.pandas as pa

schema = pa.DataFrameSchema(
  {
    "IDENTIFIER": pa.Column(int),
    "OBS_VALUE": pa.Column(float, pa.Check(lambda s: s > 500)),
    "COUNTRY": pa.Column(str, [
      pa.Check(lambda s: s.str.startswith("CTY_")),
      pa.Check(lambda s: s.str.split("_", expand=True).shape[1] == 2)
    ]),
  },
  strict=True
)

schema.validate(my_dataframe)
```

However, schemas change.

A good approach is to define and manage schemas outside of the Python code.

This **separation of concerns** avoids entangling data quality rules with pipeline code, making each part easier to understand, test and evolve independently.

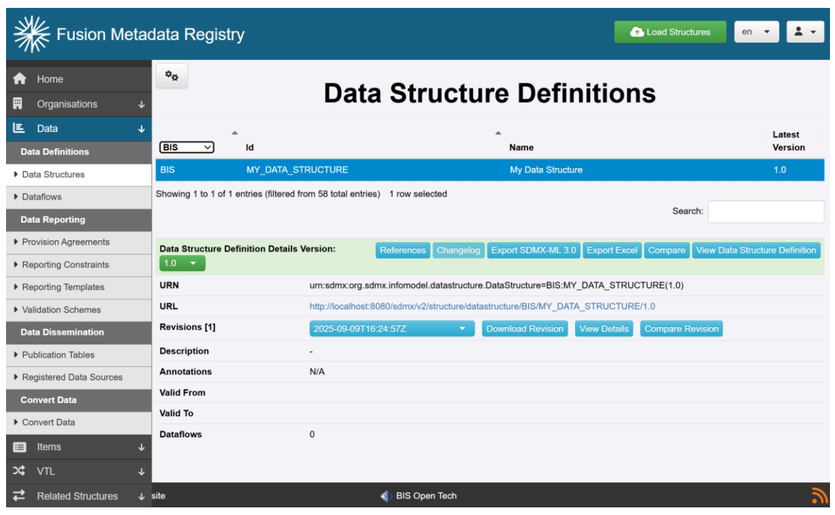
We can achieve this separation by using SDMX Data Structure Definitions (DSDs) to define the schemas, and Fusion Metadata Registry (FMR) as a *schema registry* – a repository where schemas are centralised, stored and managed.

Business users can take control, modifying schemas as and when required like adding new values to code lists. Moreover, changes to the Python are avoided, making data management more agile and relieving the burden on IT specialists and data engineers.

```
import pandas as pd
import pandera.pandas as pa
import pyfmr.panderafmr as fp


schema = pa.DataFrameSchema(
      fp.getColumns(
            "https://fmr.mydomain.org",
            "datastructure=BIS:MY_DATA_STRUCTURE(1.0)"
      ),
      strict=True
)

schema.validate(my_dataframe)
```



Try it on GitHub