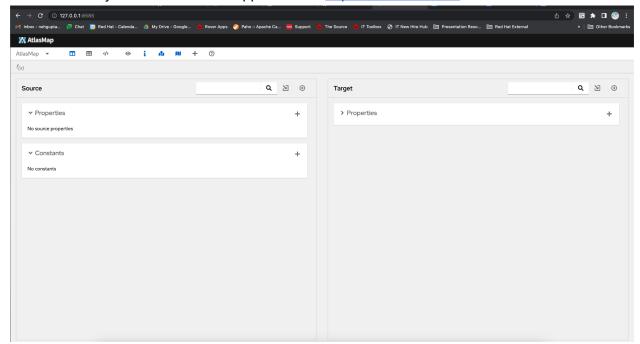
You can install the standalone version of AtlasMap from Git Hub to run locally and work with your application.

Reference Link: https://github.com/atlasmap/atlasmap

Once installed you can access the application at http://127.0.0.1:8585

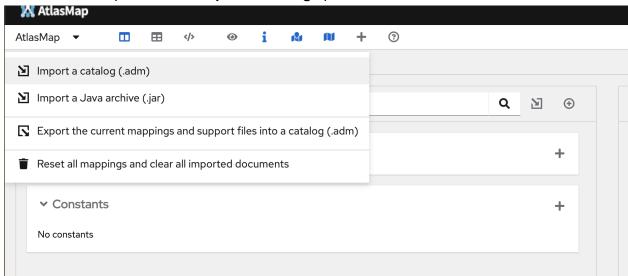


Once launched you can create your data conversion logic here or import the existing ones as well.

The AtlasMap is a data mapping solution with an interactive web-based user interface, that simplifies configuring integrations between Java, XML, and JSON data sources. You can design your data mapping on the AtlasMap Data Mapper UI canvas, and then run that data mapping via the runtime engine. AtlasMap Data Mapper UI is primarily designed to work within the Syndesis UI, and now we're exploring to improve the standalone user experience.

Importing an Existing Data Mapper File.

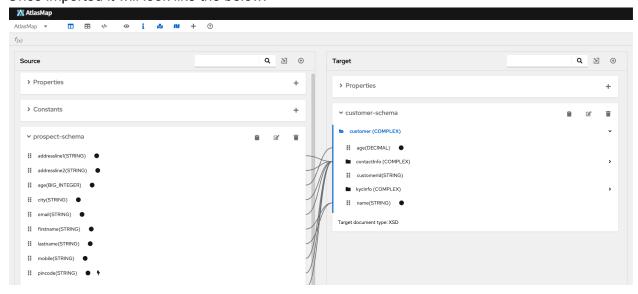
Click on Atlas Map and select Import a catalog option from the list



Select the path of your .adm file.

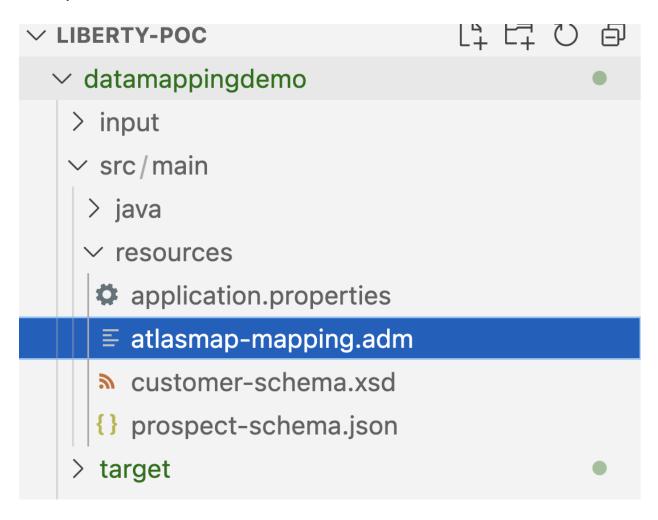


Once imported it will look like the below:



Using Data Mapping files inside the application.

To use the data mapper file inside your application put it under the **src/main/java/resources** directory.



Once you have added the file, You can refere this file inside the route of your application.

```
datamappingdemo > src > main > java > com > redhat > demo > dataformatdemo > J SimpleRouteBuilder.
∨ LIBERTY-POC
                              日の甘口
 \lor datamappingdemo
                                               27
  > input
                                                              restConfiguration()
                                               28
  ∨ src/main
                                               29
                                                              .component("servlet");

y java / com / redhat / demo / dataformatdemo

√

                                               30
                                               31
                                                              rest()
    J DataformatdemoApplication.java
                                               32
                                                              .consumes(MediaType.APPLICATION_JSON_VALUE)

■ SimpleRouteBuilder.java

                                               33
                                                              .produces(MediaType.APPLICATION_JSON_VALUE)

√ resources

                                               34
    application.properties
                                               35
                                                             //.get("/opportunity").to("{{route.findAllOpportunities}}")
                                               36
                                                             .post("/prospect").to("direct:getCustomer");
    = atlasmap-mapping.adm
                                                             //.get("/myfile").to("direct:viewfile");
                                               37
    a customer-schema.xsd
                                               38
    {} prospect-schema.json
                                               39
                                                              from("direct:getCustomer")
  > target
                                                             .to("log:response")
                                               40
  41
                                                              .to("atlasmap:atlasmap-mapping.adm")
                                               42
                                                                      .convertBodyTo(String.class)
  • mvnw
                                               43
                                                                      //.unmarshal(getJacksonDataFormat(OpportunityList.class))
  mvnw.cmd
                                                                     //.marshal().json()
                                               44
  nom.xml
                                               45
                                                                      .to("log:response");
                                               46
                                               47
                                               48
                                               49
                                               50
                                                         private JacksonDataFormat getJacksonDataFormat(Class<?> unmarshalType)
                                               51
                                                             JacksonDataFormat format = new JacksonDataFormat();
                                               52
                                                              format.setUnmarshalType(unmarshalType);
                                               53
                                                              return format;
                                               54
                                               55
> OUTLINE
                                               56
```

The above example does the JSON to XML conversion for the given input data.

Below is an example of data conversion:

