Experiments to Understand Avro Schema:

Generation, Compatibility and Schema Evolution

Motivation

For developers it is important to know how to handle data serialization and data modelling. Although a lot of examples and hints are provided on the internet, a systematic approach helps us to remember the good features and the pitfalls.
Doing the right things fast does not protect you from falling. But how to continue? How to get up again? If you have seen the error messages, they won't surprise you.
This project gives you a playground for working with and learning how handle Avro Schema in your projects.

Objectives

Module 1: Avro Basics

- Generation of a Java class from a given Schema
- Generate an Avro schema from private Java classes
- Generate Avro schema from Java class from any package
- Generate Avro schema from generated Java class
- Test schema compatibility for 2 persisted schemas
- Test schema compatibility for modified schemas

Summary:

- Generation of fields depends on the chosen datatype.
- Primitives can't be NULL, objects can be NULL, thus, the field is nullable for objects only.
- Public fields appear in the schema always.
- Private or protected fields, which have a Getter/Setter methods appear in the schema.
- References to the same class work.
- References to other classes work.
- In Avro 1.9, the change of the namespace is not an incompatible change any longer. In older Avro version this was an incompatible change.

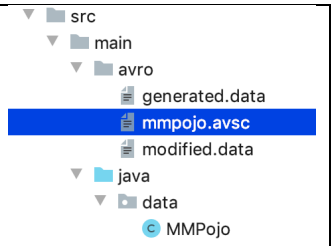
Module 1: Avro Basics

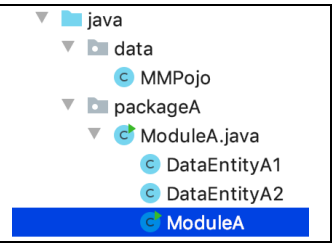
Generation of a Java class from a given Schema

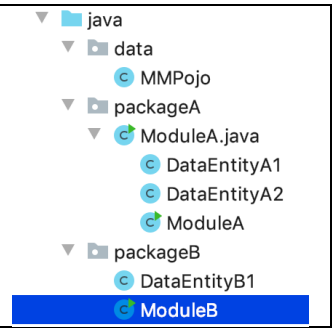
- We use a local copy of the schema file in folder: `src/avro/mmpojo.avsc`
- The Avro-Maven plugin processes only files with extension `[.avsc]`

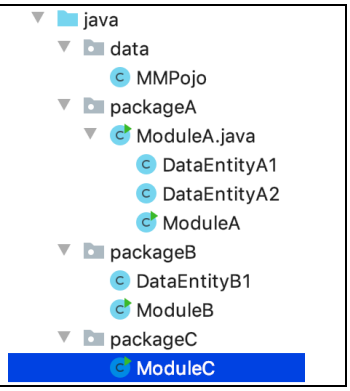
MMPojo.java is the result of the Schema generation via Maven plugin.

```
mvn generate-sources
```



Generate Avro schema from private Java classes	

Generate Avro schema from Java class from any package	

Generate Avro schema from generated Java class	

Test schema compatibility for 2 persisted schemas Test schema compatibility for modified schemas