Document Type and Meta Feed

XWikiGuest
2011/01/19 13:11

Table of Contents

Document Type and Meta Feed 3
Data model 3
Document Type 3
Meta Feed 4
Implementation 5
MetaFeedImpl 5
EnumerationValueFactory 6

Document Type and Meta Feed

How to create Document Types and Meta Feeds.

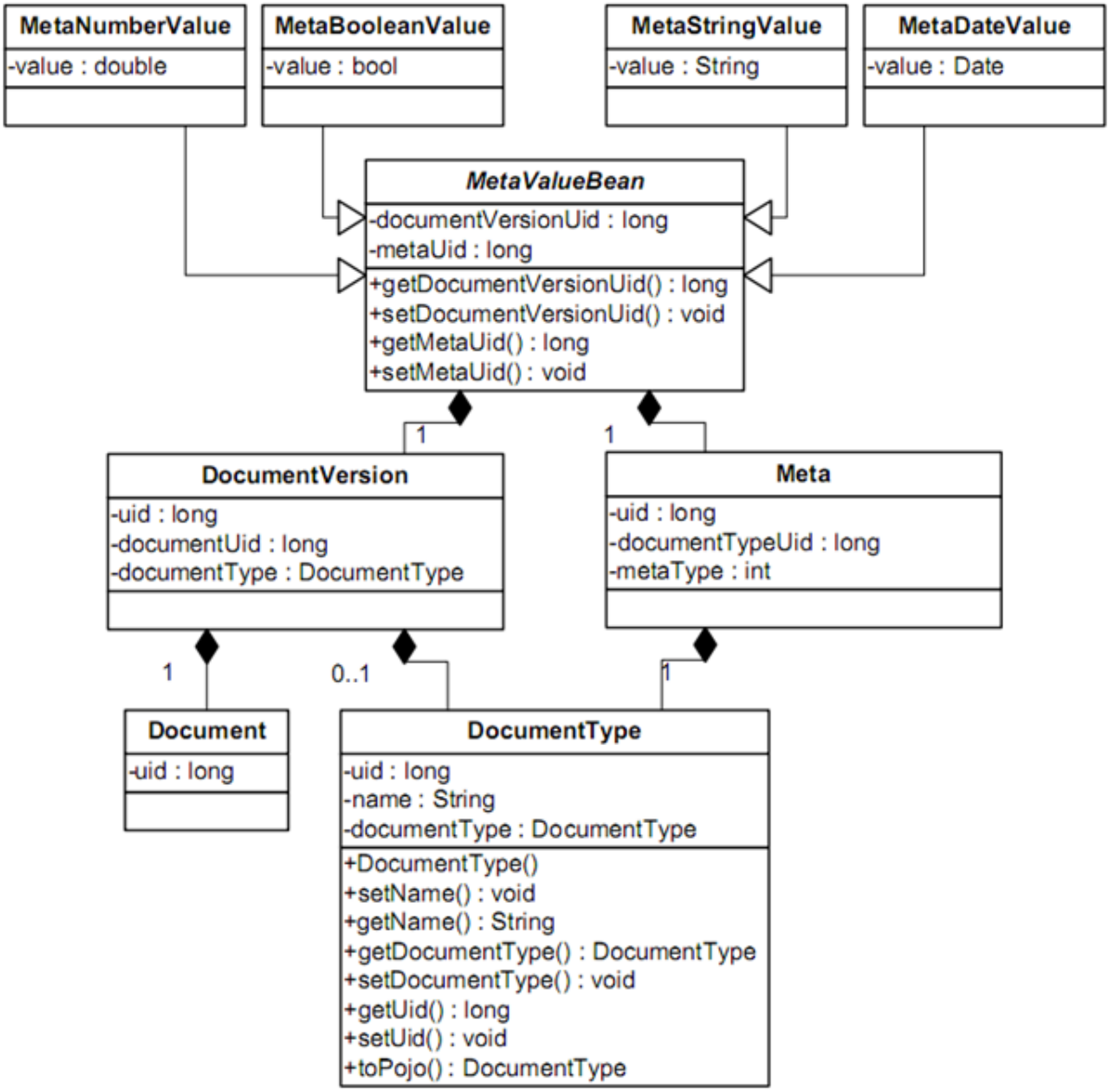
Data model

The Document Type and Meta Feed data model.

Document Type

The class diagram shows relations between document type and meta values classes.

NB: A DocumentType may contain another. In this case, this Document Type inherit settings from it.



Currently, each Meta refer to a documentTypeUid corresponding to existing Document Type.

Document Type does not contain Meta, but each Meta has its own documentTypeUid referring to a Document Type instance.

For each Document Type, it's possible to register n Meta Data. 4 types are available for each meta data. Each type is symbolized by an integer attribute metaType of the Meta class:

- MetaStringValue* (metaType: 1)

- MetaNumberValue (metaType: 2)
- MetaDateValue (metaType: 3)
- MetaBooleanValue (metaType: 4)

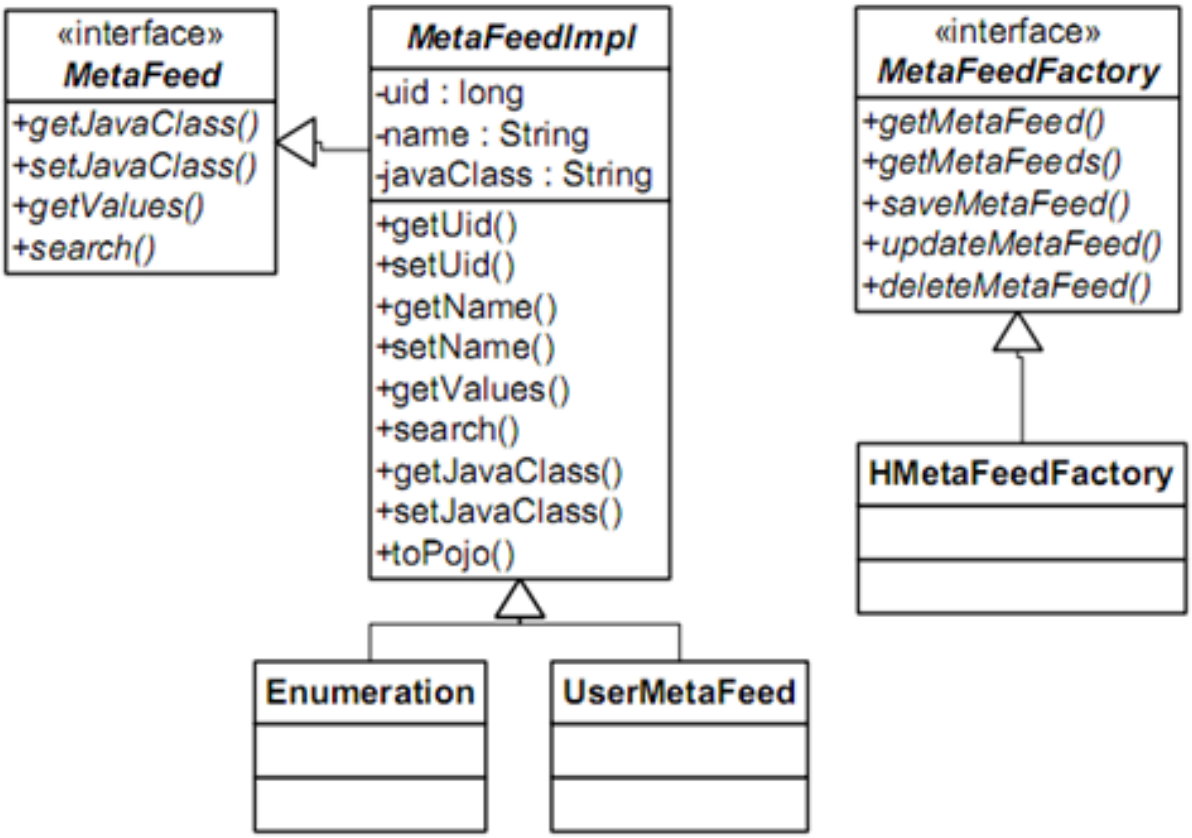
Each of them are instantiated when setting a Document Type to a particular Document Version (please see Quotero source code, DocumentVersionController, method updateMetasValue).

NB: A meta can received a MetaFeed, as type. In this case, the meta type is String.

** The string value must be specified with a Java Class full name (which is an implementation of meta feed as enumeration).*

Meta Feed

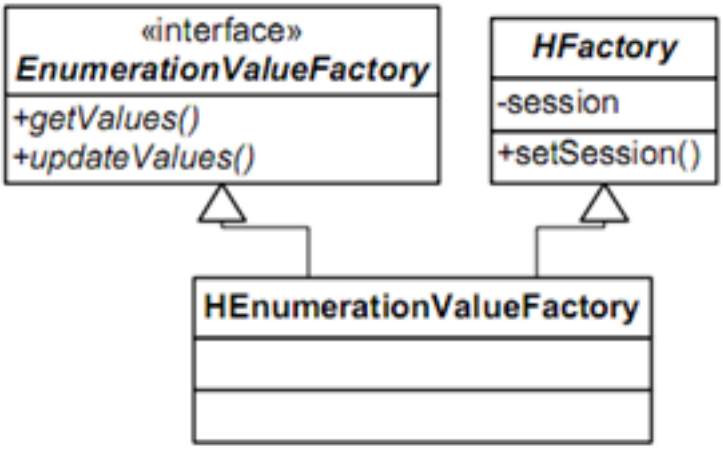
Here is the class diagram showing how to implement Meta Feed:



If you want to create a new Meta Feed type, you must build an new MetaFeedImpl inherited class which implements methods from the interface: org.coretechs.quotero.dms.MetaFeedImpl.

Method	Description
getUid()	Return the UID of meta feed record
setUid()	Set the UID of meta feed record
getName()	Return the meta feed name
setName()	Set the meta feed name
setJavaClass()	Set the meta feed implementation class
getJavaClass()	Return the meta feed implementation class
toPojo()	Convert meta feed to plain old Java object from meta feed
getValues()	Return all meta feed values

Example with Enumeration meta feed implementation:



Implementation

A concrete example with Enumeration Meta Feed implementation. Methods have been deliberately simplified for understanding (don't use as it).

MetaFeedImpl

```
public class Enumeration extends MetaFeedImpl {

        /**

            Get values of Enumeration meta feed delegating to EnumerationValueFactory.

            Return a Vector of String.

        */

    public Vector<String> getValues() {

        EnumerationValueFactory ef =

                                        new FactoryInstantiator().getEnumerationValueFactory();

        return ef.getValues(this.uid);

    }


        /**

            Make searches from criteria into current meta feed

                and return array of string as result.

        */

    public String[] search(String criteria) {

        Vector<String> r = new Vector<String>();

        Vector<String> values = this.getValues();

        for (String s : values) {

            if (s.contains(criteria))

                r.add(s);

        }

        String[] a = new String[r.size()];

        return r.toArray(a);
```

```
    }
}
```

EnumerationValueFactory

```
public class HEnumerationValueFactory extends HFactory implements EnumerationValueFactory {
    /**
            Get values of Enumeration meta feed.
        Return a Vector of String.
    */
        public Vector<String> getValues(long uid) {
        Vector<String> vValues = new Vector<String>();
        List<String> lValues = session.createSQLQuery(
                                "SELECT ... FROM ... WHERE ... ORDER BY ...");
        for(String st: lValues)
            vValues.add(st);
        return vValues;
    }

    /**
         Update values of Enumeration meta feed with its uid and its values.
    */
    public void updateValues(long uid, Vector<String> values) {
        session.createSQLQuery("DELETE FROM ... WHERE ...").executeUpdate();
        for(String b: values){
            session.createSQLQuery(
                                "INSERT INTO ... (...) VALUES (...)")
                                .executeUpdate();
        }
    }
}
```