QXmlMoodle - Developer

Abstract

QXmlMoodle is a open-source library host on github (https://github.com/roddevu/QXmlMoodle). This library permit you to import and export the Moodle quiz in XML.

The following document is provide to help future developer to understand how the project work and theses specificities.

In order to have the highest visibility all the project and even the documentation are in English. But it's a French project made only by French people for the moment ;)

Quick usage

If you don't want to "put your hand in the code", follow this 3 steps to use **QXmlMoodle** in your project:

1. Add this two jar to your project:
   a. **XOM 1.2.7** (or higher) findable here: http://www.cafeconleche.org/XOM/xom-1.2.7.jar
   b. **QXmlMoodle 1.0**: https://github.com/roddevu/QXmlMoodle/blob/master/qxmlmoodle.jar
2. Look javadoc and samples
3. Enjoy :)

Project

With **QXmlMoodle** developers can: **import**, **change**, **create**, **export** a moodle quiz in an XML valid document.

Furthermore a **powerful XML debugger** is included! Indeed, for each exceptions and warning **QXmlMoodle** provide you all the necessary to correct them: line, xml marker and a short description. Sometimes you will also have a proposition of correction!

All you have to do is load a document, using the load method of Quiz class, and call **getErrors()** or **getWarnings()** method to get the list of error(s) and warning(s).

Very easy to use, isn't it?

Strict compliant

QXmlMoodle is fully PMD and CheckStyle compliant.

To participate to this project, your code need to raise **0 warning after a CheckStyle and a PMD scan**! PMD information warnings are tolerate, but please avoid them.

Usage

To use the actual version of **QXmlMoodle**, you need to add XOM 1.2.7 (or higher) to your project. You can download it jar here: http://www.cafeconleche.org/XOM/xom-1.2.7.jar.

QXmlMoodle provide a class that covers all needs: question list, error list, warning list.

- To create a quiz:
  - simply instantiate the "**class fr.qxmlmoodle.Quiz**"
- To import a Moodle quiz:
  - call the method: **load(file_url : string)**
  - return true if no important error have been encounter, return false otherwise
- To manipulate the quiz:
  - adding a question
  - get the question list
  - change a question
  - etc.
- To test your quiz:
  - use **getErrors()** to retrieve the error list
  - use **getWarnings()** to retrieve warnings
- To export your quiz, you have two methods:
  - use **exportToString() : string**
  - use **save(file_url : string) : boolean**

QXmlMoodle support (for the moment) 9 type of questions. It represent all the questions type available in the actual Moodle quiz format.

Each sort of question have its methods, for example in a Type Matching question you can manage the sub questions with: **addSubQuestion()** and **getSubQuestions()**.

You can find more information in the Javadoc, here: https://github.com/roddevu/QXmlMoodle/tree/master/JavaDoc

Architecture

Common

You should use the **MoodleXMLList** instead of the java list object to maintains a code coherence.

This generic class is a container of item, all the class which contains many element extend or use it.

Question

All the questions are in one main class: **Quiz**.

This class contains:

- A **QuestionList** which extend **MoodleXMLList**. This represent the **Question Factory**.
- All the question type extend **AbstractQuestion**.

An **AbstractQuestion** extended class provide the **createAnswer()** method. This method permit you to return your own answer type.

If your question type doesn't need more than the classic answer you can return **new AnswerGeneral()**.

To sump-up, if you want to create or add a new question type:

1. Create a class **QuestionYYY** which extend **AbstractQuestion**, in the *fr.qxmlmoodle.question* package. This class represent the new question type YYY.
2. If your question need specific answer:
   a. Create a class **AnswerYYY** which extend **AbstractAnswer**, in the *fr.qxmlmoodle.question.answer* package.
   b. Override methods: **doSpecificImport()** and **doSpecificExport()**. View javadoc for more information: https://github.com/roddevu/QXmlMoodle/tree/master/JavaDoc. -> return false to stop the import/export.
   c. Override the abstract method **createAnswer()** from your question type by returning **new AnswerYYY().**
3. If your question use generic answer:
   a. Override the abstract method **createAnswer()** by returning **new AnswerGeneral()**.
4. Override methods: **doSpecificImport()** and **doSpecificExport()**. View javadoc for more information: https://github.com/roddevu/QXmlMoodle/tree/master/JavaDoc. -> **return false only on important error** because this will stop the import/export. Preferably, throw warning or error.
5. Add your type YYY to the enumeration QuestionType.

6. Change the class **QuestionList** (the question factory in *fr.qxmlmoodle* package) to add your question. Don't forget to be PMD compliant! So if PMD throw a warning for complexity create another method **createQuestion***(I+1)* (with I represent the number of **createQuestion** method existing in the class).
   Normally you could add your new type in the **createQuestion2** method. But if PMD throw a warning simply create a new method call **createQuestion3**.
7. That's all ;)

Error/Warning

QXmlMoodle have its own exception and warning manager, this allow you to not use *try....catch* if you don't want.

For each exceptions and warning **QXmlMoodle** provide you all the necessary to correct them: line, xml marker and a short description. Sometimes you will also have a proposition of correction!

In this version, only an import of a xml file can throw errors or warnings. Since **QXmlMoodle** export in an correct XML format.

All error implement **IError** and all the import error extend **ImportError**.

All warning implement **IWarning** and all the import warning extend **ImportWarning**.

If you want to add an import Warning YYY [or an import Error YYY], you have to:

1. Create a class **ImportWarningYYY** [**ImportErrorYYY**] which extend **ImportWarning** [**ImportError**], in the *fr.qxmlmoodle.warning* [*fr.qxmlmoodle.exception*] package.
2. Add to the enumeration ImportWarningType [ImportErrorType] your type YYY.
3. Create constructors with pertinent parameter and precise the type of your Warning [Error] by using the mother constructor method. View javadoc for more information: https://github.com/roddevu/QXmlMoodle/tree/master/JavaDoc.
4. That's all ;)

If you want to throw an import Warning, add the following code in the **doSpecificImport()** method:

importer.getWarnings().add(new ImportWarningYYY(...))

If you want to throw an import Error, add the following code in the **doSpecificImport()** method:

importer.getErrors().add(new ImportErrorYYY(...))

In both case you can **return false to stop the import**.

Last note

QXmlMoodle is fully PMD (3.2.6) and CheckStyle (5.5.0) compliant.

To participate to this project, your code need to raise **0 warning after a CheckStyle (5.5.0) and a PMD (3.2.6) scan**! PMD information warnings are tolerate, but please avoid them.

For more information look at:

- Samples: https://github.com/roddevu/QXmlMoodle/tree/master/samples
- Javadoc: https://github.com/roddevu/QXmlMoodle/tree/master/JavaDoc