JlingoProposal

Multi Lingual Message Framework

Aim: *To write a framework that will help programmers to use multi language with ease. The idea is to make it usable through all possible languages. To start with , we will consider our scope to be JAVA (Java EE) to be precise. The end product should be usable by anyone by including their .JAR files in their project*

XML files: Standard XML Schema : LanguageConfig.xml Standard Individual language files: LangEng.xml etc.

Compulsory Attributes in the Individual XML *Each message packet should contain these compulsory attributes:* ID : Unique ID of the message
msgText : message content

Optional Attributes in the Individual XML It can contain optional fields like:
csvName: comma separated metadata names of the fields associated with the message csvValue: comma separated metadata value for the fields associated with the message.

Examples:
severity: type of the message: INFO/DEBUG/ERROR/FATAL wrapStyle: will be wrapped by this css style while display serviceName: name of the service. applicationID: ID of the application

Details of each field:

1. ID:
a. It can be alphanumeric/ numeric, to be decided.
b. Should be unique but same throughout different language files
c. Should be of certain min length, we need to fix this

2. msgText
a. Should be able to handle UNICODE
b. Should be able to handle parameters:
For example: the message “hello ?%name%?” then ?%name%? should be replaceable by a variable.

3. csvName: variable name for the metadata for the use of this message. 4. csvValue: value of the above defined variable.

---~~Some csvName and cvsValue, examples~~ --

5. WrapStyle: Optional a. When entered, it should wrap it with a style of that css. Assume that the stylesheet is linked. b. They can also give stylesheet syntax there itself.

6. serviceName:optional Class which invokes this, helpful when the message is of DEBUG type.

7. applicationID: optional : ID of the application. Global Module

8. severity: Optional a. Can be Info/debug/error/warning/fatal. b. These will serve as a category of the message for the programmer to identify

1. Info:

1. Just simple static text, label etc. : Eg: “Please Enter Name”

2. something Developers can set info flag true to display these messages 3.
ii. Debug:

1. Debug information, to be used like simple echo or sysout etc

2. should be something like set a flag and debug message can be displayed.
iii. Error:

1. Pure Business Logic errors. Like validation failure.

example: “Please enter your name”
iv. Warn:

1. Business / System level warnings
   v. Fatal:
2. System level fatal errors: Something not working etc.

Example LanguageConfig.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<Languages>
     <Language name="English" file="LangEng.xml" load="true"/>
</Languages>
```

Example LangEng.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<Language version="1.0" name="English">
     <!--  Global Common Data -->
     <MetaData>
          <ServerName></ServerName>
          <ServerEnv></ServerEnv>
     </MetaData>
     <!--  Application Specific data -->
     <Messages>
          <msg ID="I00001" msgText="This is text1"
                        csvName="severity,wrapStyle"
                             csvValue="INFO,.style1"
                           />
          <msg ID="E00002" msgText="This is a second text Example"
                        csvName="severity,wrapStyle"
                    csvValue="ERROR,.style2"
                           />
     </Messages>
</Language>
```