



Elektronischer Rechtsverkehr (ERV)

Spezifikation des Validierungsmoduls für FB-Eingaben

Dateiname: FB_Spezifikation_Validierungsmodul

Version: 1.2.1 vom 24.11.2009

Ersteller: Robert Kleedorfer (robert.kleedorfer@brz.gv.at)

Letzte Änderung: Sabine Reichl

1 Dokumentinformation

1.1 Inhaltsverzeichnis

1	Dokumentinformation	2
1.1	Inhaltsverzeichnis	2
1.2	Tabellenverzeichnis	2
1.3	Änderungsverlauf.....	2
2	Einleitung	3
2.1	Zweck des Dokuments.....	3
3	Validierungsmodul	4
3.1	Deployment Units	4
3.2	Application Programming Interface (API)	4
3.2.1	Class ErsteValidierung	4
3.2.2	Interface Message.....	4
3.2.3	Interface Part	5
3.2.4	Interface Result	5
3.2.5	Class Version	5
3.3	Mitgelieferte Komponenten	5
3.4	Externe Abhängigkeiten	5
3.5	Selbsttest	5

1.2 Tabellenverzeichnis

Tabelle 1: Verwendete externe Bibliotheken.....	5
---	---

1.3 Änderungsverlauf

Version	Datum	Ersteller	Kommentar
1.0	19.01.2007	Robert Kleedorfer	Erste Version
1.1	01.03.2007	Robert Kleedorfer	Ausbessern Aufrufparameter. Referenz auf neues erv-Framework
1.2	01.10.2009	Sabine Reichl	Hinzufügen von iText.jar
1.2.1	23.11.2009	Sabine Reichl	Erweitern um 2 externe Bibliotheken für verschlüsselte PDFs

2 Einleitung

2.1 Zweck des Dokuments

Dieses Dokument enthält die Beschreibung des Validierungsmoduls für Eingaben an die Applikation Firmenbuch (FB) im Rahmen des ERV.

Das Validierungsmodul wird sowohl bei den Übermittlungsstellen als auch in der BRZG zur Prüfung von sämtlichen FB-Eingaben eingesetzt.

Dieses Dokument ist nur eine Ergänzung zum entsprechenden Dokument für die VJ (VJ_Spezifikation_Validierungsmodul.doc), hier werden nur firmenbuchspezifische Unterschiede dokumentiert.

3 Validierungsmodul

3.1 Deployment Units

Das Validierungsmodul wird als Java-Programm realisiert. Die Validierung (für applikationsspezifische Prüfungen) besteht aus einer Validierungsklasse, einigen Hilfsklassen und den dazu benötigten Ressourcen, die in einem JAR-File mit dem Namen `fb_validierung.jar` gebündelt werden.

Der Name des JAR-Files enthält den Anwendungsnamen (also „FB“), jedoch keine Versionskennung, weil es nicht möglich sein soll, mehrere Versionen des Validierungsmoduls parallel zu verwenden. Stattdessen kann die Versionsinformation über ein API abgefragt werden.

3.2 Application Programming Interface (API)

Im folgenden Kapitel werden die öffentlichen APIs des Validierungsmoduls näher beschrieben. Damit soll es möglich sein, die Validierung einer Nachricht aufrufen und das zurückgegebene Ergebnis interpretieren zu können.

3.2.1 Class *ErsteValidierung*

Die Validierungsklasse *ErsteValidierung* enthält die Prüfmethode `validate()`, welche applikationsspezifischen Inhalt der eingehenden Nachricht (gekapselt in einer *Message*) prüft.

Es wird gegen jene Geschäftsregeln der Anwendung FB geprüft, die im Dokument *FB_Geschäftsregeln* enthalten sind.

```
Package at.gv.justiz.erv.fb.validierung;  
  
public class ErsteValidierung {  
    public static ValidationResult validate(Message msg)  
        throws ValidationException;  
  
    public static Version getVersion();  
}
```

Die Prüfmethode `validate()` ist synchron und re-entrant und liefert normalerweise ein *ValidationResult* (welches eine Liste von Validierungsergebnissen des Typs *Result* enthält) zurück. Im Ausnahmefall wird eine *ValidationException* geworfen.

- Die Nachricht (*Message*) gilt nur dann als valide, wenn `ValidationResult.isOk()` dies bestätigt.

Die Methode `getVersion()` gibt in einem Objekt der Klasse *Version* die strukturierte Versionsinformation für das Validierungsmodul zurück.

3.2.2 Interface *Message*

Für die genaue Beschreibung siehe *VJ_Spezifikation_Validierungsmodul*.

3.2.3 Interface Part

Für die genaue Beschreibung siehe [VJ_Spezifikation_Validierungsmodul](#).

3.2.4 Interface Result

Für die genaue Beschreibung siehe [VJ_Spezifikation_Validierungsmodul](#).

Die Methode `getMessage()` liefert eine zur Anzeige geeignete Fehlermeldung. Die Fehlermeldungen sind in [FB_Fehlermeldungen](#) beschrieben und werden durch ihre Kennung (vgl. `getPublishedId()`) identifiziert.

3.2.5 Class Version

Für die genaue Beschreibung siehe [VJ_Spezifikation_Validierungsmodul](#).

3.3 Mitgelieferte Komponenten

Die Validierung selbst wird als jar Datei ausgeliefert:

- `erv_fb_validation.jar`.

Die passenden Teile unseres „ERV Commons“ Frameworks werden ebenfalls mitgeliefert:

- `erv_commons-1.0.0-BUILD_009.jar`

Der Name kann je nach Version und Buildnummer variieren.

Dieser Teil des ERV Commons Frameworks ist für VJ und Firmenbuch identisch, er wird in der entsprechenden Zielumgebung nur einmal benötigt.

3.4 Externe Abhängigkeiten

Das Validierungsmodul verwendet Java in der Version 1.4.2_01 und darüber hinaus einige andere Open Source Bibliotheken, die in folgender Tabelle aufgelistet sind.

Bibliothek	Version	Jar	Kommentar
Xerces	2.8.0	xml-apis.jar xercesImpl.jar	http://xerces.apache.org/xerces-j
Xalan	2.7.0	xalan.jar serializer.jar	http://xml.apache.org/xalan-j
jUnit	3.8.2	junit.jar	http://junit.sourceforge.net/junit3.8.2/index.html (Wird benötigt um den Selbsttest auszuführen.)
iText	2.1.7	iText.jar	http://www.lowagie.com/iText/ Wird benötigt um die PDF Prüfungen auszuführen
Bouncycastle	138	bcmail-jdk14-138.jar bcprov-jdk14-138.jar	PDF Prüfungen mit verschlüsselten PDFs

Tabelle 1: Verwendete externe Bibliotheken

In der Spalte *Kommentar* findet man die Seiten im Internet, von denen die Bibliotheken bezogen werden können. Alle Bibliotheken müssen für eine einwandfreie Funktion des Validierungsmoduls in der Reihenfolge der Auflistung in obiger Tabelle im CLASSPATH eingetragen sein.

3.5 Selbsttest

Als Hilfsmittel um die Korrekte Einrichtung zu verifizieren steht eine Selbsttestfunktion zur Verfügung. Diese macht nichts anderes als die Unittests der

Eingabevalidierung durchzuführen. Sollte dieses ohne Fehler gelingen, ist die Einrichtung (CLASSPATH usw.) stimmig.

```
package at.gv.justiz.erv.fb.validierung.testsuite;  
public class ValidierungTestSuite extends TestSuite  
{  
    public static void main(String[] args) { ... }  
}
```

Ein Aufruf von der *Linux* Commandline sieht so aus:

```
# wir stehen in einem Verzeichniss mit den benötigten jars  
$ ls -l  
erv_commons-1.0.0-BUILD_009.jar  
erv_fb_validation.jar  
junit-3.8.1.jar  
serializer-2.7.0.jar  
xalan-2.7.0.jar  
xercesImpl-2.8.1.jar  
xml-apis-2.8.1.jar  
iText-2.1.7.jar  
  
# mit paste wird aus der liste der jars eine classpath gebaut.  
$ java -cp `ls -l *.jar | paste -s -d : -` \  
at.gv.justiz.erv.fb.validierung.testsuite.ValidierungTestSuite
```