



*Elektronischer Rechtsverkehr (ERV)*

# **Spezifikation des Validierungsmoduls für GB-Eingaben**

ERV Version 15.1

Dateiname: GB\_Spezifikation\_Validierungsmodul

Dokument-Version: 15.1.0 vom 09.12.2014

Ersteller: Markus Michalitsch ([grundbuch@brz.gv.at](mailto:grundbuch@brz.gv.at))

# 1 Dokumentinformation

## 1.1 Inhaltsverzeichnis

1	Dokumentinformation.....	2
1.1	Inhaltsverzeichnis.....	2
1.2	Tabellenverzeichnis.....	2
1.3	Änderungsverlauf.....	2
2	Einleitung.....	4
2.1	Zweck des Dokuments.....	4
3	Validierungsmodul.....	5
3.1	Deployment Units.....	5
3.2	Stand der Implementierung.....	5
3.3	Application Programming Interface (API).....	5
3.3.1	Class ErsteValidierung.....	5
3.3.2	Interface ValidationResult.....	6
3.3.3	Interface Message.....	6
3.3.4	Interface Part.....	6
3.3.5	Interface Result.....	6
3.3.6	Class Version.....	6
3.4	Mitgelieferte Komponenten.....	6
3.5	Externe Abhängigkeiten.....	6
3.6	Selbsttest.....	7
3.7	Fehlerhilfe.....	7

## 1.2 Tabellenverzeichnis

Tabelle 1:	Verwendete externe Bibliotheken.....	7
------------	--------------------------------------	---

## 1.3 Änderungsverlauf

Version	Datum	Ersteller	Kommentar
1.0	19.01.2007	Kleedorfer	Ursprüngliche Version für FB-Validierung
1.0.1	28.05.2008	Michalitsch	Adaptierung für GB-Validierung: Erste Version
1.0.2	24.06.2008	Michalitsch	Anpassungen bei 3.2 und 3.3.2
1.0.3	30.09.2008	Michalitsch	JAXB Package auf jaxb-ri-1.0.6 geändert (Siehe: Externe Abhängigkeiten)
1.0.4	11.11.2008	Michalitsch	Anpassung der Regelliste für die aktuelle Freigabe Package-Pfad zur „ErsteValidierung“ Klasse angepasst.

1.0.5	05.12.2008	Michalitsch	Anpassung bei 3.2 und 3.5
1.1.1	09.03.2009	Michalitsch	Nur Hochversionierung V1_0 → V1_1, keine inhaltlichen Änderungen
1.2.1	17.06.2009	Michalitsch	Kapitel: Fehlerhilfe eingefügt
2.0.1	01.02.2010	Michalitsch	nur hochversioniert, keine inhaltlichen Änderungen (Schermann)
2.0.2	12.09.2014	Wimmer	Validierungsmodul aktualisiert – plus jar für iban-bic
15.1.0	09.12.2014	Dönz	Validierungsmodul 2.0.25, Externe Abhängigkeiten aktualisiert

## 2 Einleitung

### 2.1 Zweck des Dokuments

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

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

Dieses Dokument ist nur eine Ergänzung zum entsprechenden Dokument für die VJ ([\*VJ\\_Spezifikation\\_Validierungsmodul\*](#)), hier werden nur Grundbuchbuchspezifische 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 einer JAR-Datei mit dem Namen `erv_gb_validation_<Version>.jar` gebündelt werden.

Der Name der JAR-Datei enthält den Anwendungsnamen (also "GB"), und eine Versionskennung. Zusätzlich kann die Versionsinformation über ein API abgefragt werden.

### 3.2 Stand der Implementierung

Es wird grundsätzlich gegen jene Geschäftsregeln der Anwendung GB geprüft, die im Dokument [GB\\_Geschaeftsregeln](#) enthalten sind und nicht als intern (i) gekennzeichnet sind. Ausgenommen davon sind die folgenden Geschäftsregeln:

- Anhänge allgemein: GR-0001 bis GR-0004 (alle)
- Payload: GR-1003, GR-1143

### 3.3 Application Programming Interface (API)

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

#### 3.3.1 Class *ErsteValidierung*

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

```
package at.gv.brz.gb.erv;

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.

### 3.3.2 Interface ValidationResult

Für die genaue Beschreibung siehe [VJ\\_Spezifikation\\_Validierungsmodul](#).

Für die GB-Validierung sind folgende Listen implementiert:

- `getSatisfiedRules()` – Liefert ein Set mit den erfüllten Regeln.
- `getFailedRules()` – Liefert eine Map mit den Regeln die eine Exception ausgelöst haben. Key: Result Objekt, Value: Throwable Objekt.
- `getViolatedRules()` – Liefert ein Set mit den verletzten Regeln.
- `getResults()` – Liefert alle Resultate (Satisfied und Violated)

### 3.3.3 Interface Message

Für die genaue Beschreibung siehe [VJ\\_Spezifikation\\_Validierungsmodul](#).

### 3.3.4 Interface Part

Für die genaue Beschreibung siehe [VJ\\_Spezifikation\\_Validierungsmodul](#).

### 3.3.5 Interface Result

Für die genaue Beschreibung siehe [VJ\\_Spezifikation\\_Validierungsmodul](#).

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

### 3.3.6 Class Version

Für die genaue Beschreibung siehe [VJ\\_Spezifikation\\_Validierungsmodul](#).

## 3.4 Mitgelieferte Komponenten

Die Validierung selbst wird als JAR-Datei ausgeliefert:

- `erv_gb_validation_2_0_25.jar`.

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

- `erv_commons-1.3.0.jar`

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

Zusätzlich ist für die IBAN- und BIC-Daten die folgende JAR-Datei notwendig:

- `bic-iban-1.1.0.jar`

## 3.5 Externe Abhängigkeiten

Das Validierungsmodul verwendet Java in der Version 1.6 und darüber hinaus einige andere Open Source Bibliotheken, die in folgender Tabelle aufgelistet sind.

Diese werden nicht mitgeliefert.

Bibliothek	Version	JAR-Datei	Kommentar
Apache Commons Lang	2.4	commons-lang-2.4.jar	<a href="http://commons.apache.org/">http://commons.apache.org/</a>
Apache Commons Logging	1.1	commons-logging-1.1.jar	<a href="http://commons.apache.org/">http://commons.apache.org/</a>
Apache Commons Validator	1.3.1	commons-validator-1.3.1.jar	<a href="http://commons.apache.org/">http://commons.apache.org/</a>
Log4j	1.2.14	log4j-1.2.14.jar	<a href="http://logging.apache.org/log4j">http://logging.apache.org/log4j</a>
Spring Framework	3.0.4	Je nach Quelle: asm, beans, context, core, expression notwendig	<a href="http://springframework.org/">http://springframework.org/</a>
jUnit	3.8.2	junit.jar	<a href="http://www.junit.org/">http://www.junit.org/</a> (Wird benötigt um den Selbsttest auszuführen.)

**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 im CLASSPATH eingetragen sein.

### 3.6 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.

```
import at.gv.brz.gb.validation.testsuite.ValidierungTestSuite;
public class SelfTest {
    public static void main(String[] args) {
        ValidierungTestSuite.main(args);
    }
}
```

### 3.7 Fehlerhilfe

Während der Verarbeitung kann es bei größeren Texten unter Umständen zu einem „*StackOverflowError*“ kommen. Abhilfe schafft hier bis auf weiteres eine Erhöhung der „Stack per Thread Size“ der Java Virtual Machine.

Der Parameter dazu lautet: Xss- und die Größe des Stacks.

Zum Beispiel: java -Xss512k ...