



# Eigene Java-Klassen in Intrex

## 1. Einführung:

Velocity ist ein Open-Source Projekt der Jakarta Projektgruppe und birgt seine Hauptvorteile bei der Erstellung von dynamischen Webseiten und Webinhalten. Die Funktionsweise läuft über Skripts, die in der Skriptsprache Virtual Template Language (VTL) verfasst werden.

- ! Weitere Informationen finden Sie auf der Velocity Projektseite <http://velocity.apache.org/>

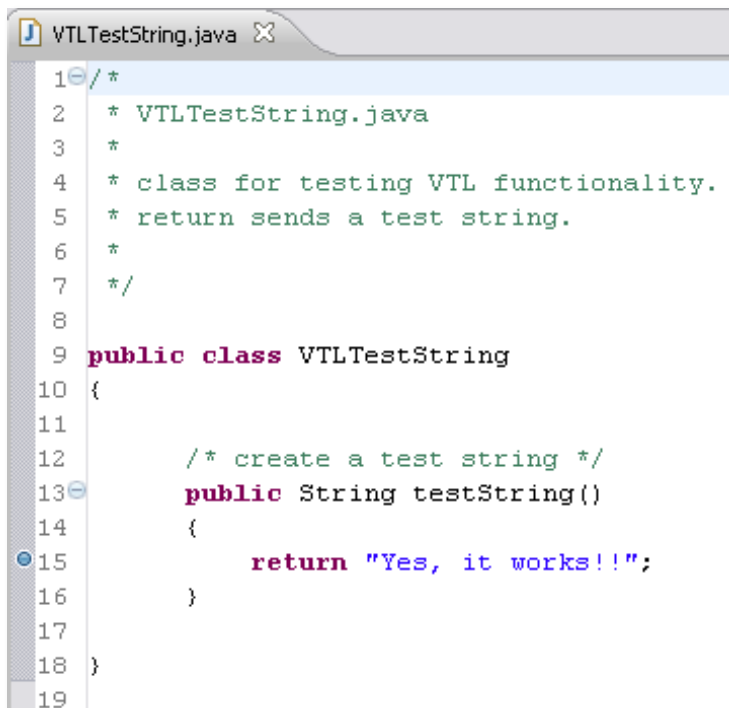
Mittels Velocity ist es Ihnen möglich, eigene erstellte Java-Klassen in Ihrem Portal einzubinden und zu verwenden.

## 2. Java-Projekt exportieren

Um eine eigenen Java-Klasse in Ihr Portal einzubinden, müssen Sie Ihr erstelltes Java-Projekt als JAR-File exportieren.

Die in diesem Beispiel eingesetzte Entwicklungsumgebung ist Eclipse 3.3.0. Sollten Sie eine andere Entwicklungsumgebung verwenden, unterscheiden sich die hier vorgestellten Schritte zum Exportieren.

Im hier genannten Beispiel haben wir eine Java-Klasse, die einen String ausgibt, wenn die Einbindung mittels Velocity funktioniert hat.




```
1 /*
2  * VTLTestString.java
3  *
4  * class for testing VTL functionality.
5  * return sends a test string.
6  *
7  */
8
9 public class VTLTestString
10 {
11
12     /* create a test string */
13     public String testString()
14     {
15         return "Yes, it works!!";
16     }
17
18 }
19
```

Haben Sie ihr Java-Projekt geöffnet, klicken Sie in der Eclipse Menüleiste auf *File / Export*. Wählen Sie als Exportformat *Java / JAR file* und klicken Sie auf *Next*.

Im nächsten Fenster markieren Sie den Projekt-Ordner, den Sie exportieren wollen. Darunter markieren Sie die Option *Export generated class files and resources*. Weiter können Sie den Speicherort wählen, an dem der JAR-Export gespeichert werden soll. Klicken Sie auf *Finish*.

Sie können nun Ihre Entwicklungsumgebung verlassen.


Stoppen Sie bitte alle laufenden Intrexx-Dienste über die Intrexx Service Konsole oder über die Windows Dienstverwaltung und schließen Sie bei Bedarf den Intrexx Portal Manager.

Das soeben exportierte JAR-File kopieren Sie anschließend in das Verzeichnis  `<intrexx>\lib\custom`.

### 3. customcallables.cfg

Um die Java-Klasse korrekt aufrufen zu können, muss nun in der Datei *customcallables.cfg* manuelle ein entsprechender Verweis eingetragen werden.

Wechseln Sie hierzu in das Verzeichnis

 `<intrexx>\org<portalname>\internal\cfg\callables\`.

Öffnen Sie nun die Datei *customcallables.cfg* mit einem beliebigen Texteditor

```
<?xml version="1.0" encoding="UTF-8"?>
<callables
  xmlns="urn:schemas-unitedplanet-de:lucy:server:velocity:callables"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:schemas-unitedplanet-de:lucy:server:velocity:callables callables.xsd">
</callables>
```

Fügen Sie hier nun folgenden Eintrag unterhalb von `xsi:schemaLocation` ein:

```
<item contextName="VTLTest" className="VTLTestString"
use="instance" />
```

`contextName` kann hierbei frei von Ihnen gewählt werden. Er dient zur Item-Referenzierung beim späteren Methodenaufruf.

Bei `className` ist exakt der Name der eingebundenen Java-Klasse einzugeben.

```
<?xml version="1.0" encoding="UTF-8"?>
<callables
  xmlns="urn:schemas-unitedplanet-de:lucy:server:velocity:callables"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:schemas-unitedplanet-de:lucy:server:velocity:callables callables.xsd">
  <item contextName="VTLTest" className="VTLTestString" use="instance" />
</callables>
```

Speichern und schließen Sie die Datei.

Nach dem Kopiervorgang des JAR-Files und der Anpassung der *customcallables.cfg* können Sie die Intrexx Dienste wieder starten.

## 4. VTL-Skript erstellen

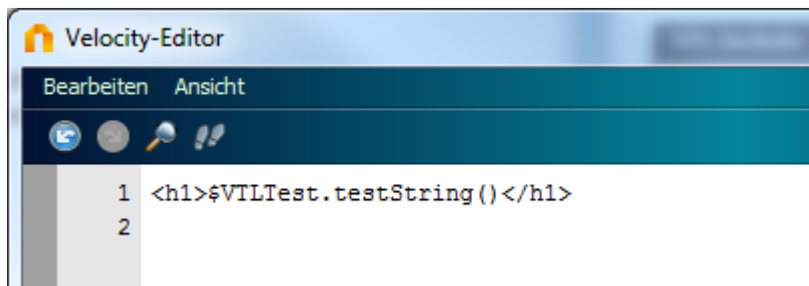
Die Einbindung der eigenen Java-Klasse(n) erfolgt durch Velocity. Zu diesem Zweck muss nun noch ein Skript in der Sprache Virtual Template Language (VTL) erstellt werden.

Platzieren Sie hierzu auf einer Intrexx Applikationsseite das Ansichtselement *VTL Include*. Sie finden dies in der Kategorie *Weitere Kontrollen*.

Erstellen Sie eine neue VM-Datei, falls noch nicht vorhanden und öffnen Sie den internen Editor von Intrexx. Erfassen Sie nun folgendes Skript:

```
$VTLTest.testString()
```

Mit dem vorangestellten Dollarzeichen \$ beginnt eine Anweisung in VTL. `VTLTest` ist der in diesem Beispiel verwendete und in `customcallables.cfg` definierte `contextName`. Mit `testString()` wird die in der Java-Klasse definierte Funktion aufgerufen, die den Teststring bei erfolgreicher Einbindung zurückgeben soll.



Das hier verwendete HTML-Tag `<h1>` hat keinen Einfluss auf die Funktionalität. Es dient lediglich zur größeren Darstellung des Strings im Portal. Speichern und schließen Sie die Datei.

Speichern und veröffentlichen Sie Ihre Applikation und öffnen Sie ihr Portal im Browser. Wählen Sie ihre erstellte Applikation.

Sie sollten folgende Ausgabe erhalten:

