

Script Callout

From iDempiere en

Contents

- 1 Creating the Rule
- 2 Configure Callout from Table/Column
- 3 Sample Code provided for copy/paste testing
- 4 Sample Code for Setting Payment Bank Account
- 5 Scripting Languages
- 6 See Also

Creating the Rule

Within the script you can use:

- Window context variables start with a W_ prefix
- Login context variables start with G_ prefix
- Parameters for callout start with A_ prefix
 - A_WindowNo
 - A_Tab
 - A_Field
 - A_Value
 - A_OldValue
 - A_Ctx

iDempiere Open Source ERP System

SuperUser@System.*/System Administrator

Feedback | Preference | Change Role | Log Out

Home Rule x

Record saved

Client* System Organization* *

Search Key beanshell:forceUpper

Name* beanshell:forceUpper

Description

Comment/Help To test this you can define callout in "Table and Column" for table "AD_Rule". In the callout field for Name column you must put @script:beanshell:forceUpper - then just open this window and fill the name to see it converted to uppercase

Event Type* Callout

Rule Type* JSR 223 Scripting APIs

Script

```
if (A_Value != null && A_Value instanceof String) {
    A_Tab.setValue(A_Field, ((String)A_Value).toUpperCase());
}
result = "";
```

Entity Type* User maintained

☒ Active

Configure Callout from Table/Column

The screenshot shows the iDempiere web application interface. The top navigation bar includes 'Home', 'Rule', and 'Table and Column' tabs. The 'Table and Column' tab is active, showing the configuration for the 'AD_Rule_Rule' table. The 'Name' column is selected, and its properties are being configured. The 'Name' field is set to 'Name', and the 'Description' is 'Alphanumeric identifier of the entity'. The 'Comment/Help' field contains the text: 'The name of an entity (record) is used as an default search option in addition to the search key. The name is up to 60 characters in length.' The 'Length' is set to 60, and the 'Reference' is set to 'String'. The 'Entity Type' is set to 'Dictionary'. The 'Value Format' is set to 'Autocomplete'. The 'Key column' checkbox is unchecked, while 'Mandatory' and 'Updatable' are checked. The 'Default Logic' field is empty. The 'Synchronize Column' button is visible. The 'Read Only Logic' and 'Mandatory Logic' fields are empty. The 'Identifier' checkbox is checked, and the 'Sequence' is set to 1. The 'Selection Column' is set to 'Sequence'. The 'Allow Logging' checkbox is checked, and the 'Allow Copy' checkbox is also checked. The 'Technical' section is expanded, showing the 'Translated' checkbox unchecked. The 'Callout' field is circled in orange and contains the text '@script:beanshell:forceUpper'. The 'Detail record' button is at the bottom left.

- This **Callout** can also be called from the **Report & Process Window**. Just remember to set in the **Rule** window from which **Event Type** are you calling this Rule Script.

Sample Code provided for copy/paste testing

On the Table and Column, callout reference:

```
@script:beanshell:forceUpper
```

On the Rule Search Key:

```
beanshell:forceUpper
```

On the Rule Script:

```
if (A_Value != null && A_Value instanceof String) {
    A_Tab.setValue(A_Field, ((String)A_Value).toUpperCase());
}
result = "";
```

Sample Code for Setting Payment Bank Account

This script will set the bank account on the Payment window based on the Tender Type and Credit Card Type.

On the Payment Table and Column, callout reference for Tender Type and Credit Card Type:

```
@script:beanshell:payment_setpaymentprocessor
```

On the Rule Search Key:

```
beanshell:payment_setpaymentprocessor
```

On the Rule Script:

```
import org.compiere.model.MPayment;

if(A_Tab.getValue("TenderType") != null && A_Tab.getValue("CreditCardType") != null && A_Tab.getValue("CreditCardType") != "")
{
    MPayment pmt = new MPayment(A_Ctx, 0,null);
    pmt.setTenderType(A_Tab.getValue("TenderType"));
    pmt.setC_Currency_ID(A_Tab.getValue("C_Currency_ID"));
    pmt.setCreditCardType(A_Tab.getValue("CreditCardType"));
    pmt.setPaymentProcessor();
    A_Tab.setValue("C_BankAccount_ID", pmt.getC_BankAccount_ID());
    pmt = null;
}
else
{
    A_Tab.setValue("C_BankAccount_ID", 0);
}
result="";
```

Scripting Languages

- Standard iDempiere has uploaded jars to work with groovy (<http://groovy.codehaus.org/>) , jython (<http://www.jython.org/Project/index.html>) and beanshell (<http://www.beanshell.org/>)
- to call a script from a **Callout** follow these sample syntax:
 - @script:beanshell:ValidateQtyOnHand
 - @script:groovy:ValidateQtyOnHand
 - @script:jython:ValidateQtyOnHand
- When you create the rule, you have to set in the **Search Key** such as:
 - Search Key : beanshell:ValidateQtyOnHand
 - Search Key : groovy:ValidateQtyOnHand
 - Search Key : jython:ValidateQtyOnHand
- Set the **Event Type** as *Callout* and **Rule Type** as *JSR 223 Scripting APIs*

See Also

- Script Process for event rule example from a **Report & Process** window.
- Script ModelValidatorLogin
- Script ModelValidator

Retrieved from "http://wiki.idempiere.org/w-en/index.php?title=Script_Callout&oldid=4142"

Categories: Development | Code snippets

- This page was last modified on 23 July 2013, at 21:07.
- This page has been accessed 959 times.
- Content is available under GNU-Lizenz für freie Dokumentation 1.3 oder höher.