

Programmierung mit Groovy (für Experten)



Diese Seite richtet sich nur an Anwender mit tiefgreifender Erfahrung im Bereich Informatik und Anwendungsentwicklung.



Das ASV-Team, die LTMs und die MULTIs leisten hierzu keinen technischen Support!

1) Überblick: Datenzugriff in ASV

Zum technischen Verständnis: die Daten in ASV sind bei der Enterprise-Installation in einer relationalen Datenbank abgelegt. Der Zugriff auf die Datenschicht erfolgt ausschließlich (!) über die Business-Objekte in Form von Klassen, z.B. die Klasse Schueler, Lehrer, Unterricht oder SchuelerSchuljahr, SchuelerStamm usw. Ein Zugriff direkt auf die ASV-Datenbank ist nicht vorgesehen und kann, bedingt durch die mehrschichtige Systemarchitektur, zur Lieferung falscher Daten führen. Damit der Datenzugriff dennoch möglich ist, arbeitet die ASV mit einem Mehr-Schicht-Datenzugriffs-Modell:

Schicht	Bezeichnung	Zugriff möglich für	Anwendung
Schicht 0	Datenbank-Ebene	nur ASV-Kernteam und NUR im Supportfall auf Zuspiegelung einer schulscharfen Sicherung durch die Schule	nicht möglich (u.A. wegen Datenschutz)
Schicht 1	Java-Ebene	nur ASV-Kernteam, Anwendungsentwicklung	nicht möglich
Schicht 2	Java-Ebene	nur ASV-Kernteam, Berichte-Ersteller	Jasper-Reports
Schicht 3	Business-Objekt-Ebene	Erfahrene Benutzer (z.B. Zugriff auf item.schuelerStamm.geschlecht())	Textfeld mit onLoad- Event (Syntax: Groovy) im Serienbrief-, Listen-, Etiketten-Generator und im Exportformat
Schicht 4	Virtuelle Felder	jeder Benutzer	Virtuelle Felder (= Datenfeld) im Serienbrief-, Listen-, Etiketten-Generator und im Exportformat
Schicht 5	ASV-Anwendung	jeder Benutzer (= das, was man in der Oberfläche sieht)	Anwendung der ASV (= das, was man in den Modulen sieht)



- mit Groovy greifen Sie auf Schicht 3 zu
- normale Berichte, Listen, usw., die den Assistenten für Datenfelder nutzen, sind Schicht 3



- zentral bereitgestellte, komplexe Berichte, greifen auf Schicht 1 und 2 zu

2) Vorgehensweise: Programmierung mit Groovy

- vorab lesen: (nicht abschließende) [Liste der Datenfelder in ASV](#)
- vorab prüfen: [fehlende virtuelle Felder](#) in ASV
- technische Dokumentation: [Programmieren mit Groovy](#)

3) Datenzugriff auf spezielle Felder (Serienbriefe, Listengenerator)



Abhängig vom Bericht kann es sein, dass Sie `obj?.schuelerSchuljahr` durch `obj?` ersetzen müssen, weil Sie sich möglicherweise schon im Datenbereich `SchülerSchuljahr` befinden.

- Schüler Straße und Hausnummer ausgeben:

```
def st = obj?.schuelerStamm;
return st.schStrasse + " " + st.schHausnummer + "\n";
```

- Betrieb Straße und Hausnummer ausgeben:

```
def stb = obj?.schuelerStamm?.ausbildungBetrieb;
String strReturn = "";
if (stb != null) {
    if (stb?.betriebName1 != null) {
        strReturn = stb?.anschrStrasse + " " + stb?.anschriftHausnr +
"\n";
    }
}
```

- Schüler > Ein-Austritt > Höchster allgemeinbildender Schulabschluss

```
obj?.schuelerStamm.vorbildungSchulischM?.kurzform
```

- Schüler > Schuljahr XX/YY > Ausbildung > Ausbildungsberuf

```
obj?.schuelerSchuljahr.ausbildungsberuf?.getLangform()
obj?.schuelerSchuljahr.ausbildungsberuf?.getAnzeigeform()
obj?.schuelerSchuljahr.ausbildungsberuf?.getKurzform()
```

- Schüler > Schuljahr XX/YY > Ausbildung > Beschäftigungsart

```
obj?.schuelerStamm.getWlStellungImBeruf().getAnzeigeform()
obj?.schuelerStamm.getWlStellungImBeruf().getKurzform()
```

- Schüler > Schuljahr XX/YY > Ausbildung > Maßnahmenträger

```
obj?.schuelerStamm.wlKostentraeger.anzeigeform
obj?.schuelerStamm.wlKostentraeger.kurzform
```

- Schüler > Schuljahr XX/YY > Gastschulverhältnis

```
obj?.schuelerSchuljahr.wlGastschuelerart?.anzeigeform
obj?.schuelerSchuljahr.wlGastschuelerart?.kurzform
```

- Schüler > Schuljahr XX/YY > Förderungsnummer

```
obj?.schuelerStamm.foerderungsnummer
```

- Schüler > Schuljahr XX/YY > Kammer-ID (nur BS)

```
obj?.schuelerSchuljahr?.kammer
```

- Schüler > Schuljahr XX/YY > zuständige Stelle, Kammer (nur BS, z.B. IHK, HWK, usw.) aus Betriebe > Ansprechpartner

```
obj?.schuelerSchuljahr?.ausb?.betriebBerufPerson?.wlBbig?.kurzform
```

alternativ, wenn Sie bereits im Datenbereich Schüler sind

```
obj?.ausb?.betriebBerufPerson?.wlBbig?.kurzform
```

- Betriebe > Ansprechpartner > Anrede des Ansprechpartners (liefert Herr/Frau)

```
obj?.schuelerSchuljahr?.ausbildung?.betriebBerufPerson?.person?.anredeAnzeigeform
```

alternativ, wenn Sie bereits im Datenbereich Schüler sind

```
obj?.ausbildung?.betriebBerufPerson?.person?.anredeAnzeigeform
```

- Klasse > Organisationsform

```
obj?.schuelerStamm?.klassengruppe?.klasse?.wlOrganisationsform?.getKurzform()
obj?.schuelerStamm?.klassengruppe?.klasse?.wlOrganisationsform?.getAnzeigeform()
```

- Klasse > Anzahl tatsächlich erteilter Unterrichtsstunden

```
obj?.schuelerStamm?.klassengruppe?.klasse?.getUnterrichtsstundenProWoche()
```

- Klasse > Klassenleiter mit Amtsbezeichnung

```
def tempGeschlechtAmtsb = "";
if(obj?.klassengruppe?.klasse?.getKlasseninformationenKlassenleiter()?.getLehrerStamm()?.getWlGeschlecht()?.kurzform == "W")
```

```
tempGeschlechtAmtsb =
obj?.klassengruppe?.klasse?.getKlasseninformationenKlassenleiter()?.get
LehrerStamm()?.getAmtsbezeichnung()?.getKurzformW();
else tempGeschlechtAmtsb =
obj?.klassengruppe?.klasse?.getKlasseninformationenKlassenleiter()?.get
LehrerStamm()?.getAmtsbezeichnung()?.getKurzform();
```

- Ersetzung je nach Geschlecht: „Der Schüler“ bzw. „Die Schülerin“:

```
return (obj.schuelerStamm.m_w.kurzform == "M") ? "Der Schüler" : "Die
Schülerin"
```

- Geburtsdatum des Schülers:

```
return obj?.schuelerStamm?.geburtsdatum.format("dd.MM.yyyy")
```

- Aktuelles Schuljahr:

```
return obj.schuljahr.kurzform
```

- Erster Tag des Schuljahres:

Anmerkung: Da das Datum für den letzten (Schul-)Tag des Schuljahres in der Datenbank nicht hinterlegt ist, kann für diesen Fall kein Code erstellt werden.

```
def Date ersterST = new Date()
ersterST = obj.schuljahr.ersterSchultag
return ersterST.format('dd.MM.yyyy')
```

- Vorname und Name des Klassenleiters/der Klassenleiterin:

```
return obj.klassengruppe.klasse.klassenleitungName
```

- Vollständige Schüleradresse mit Umbrüchen ausgeben:

```
def st = obj?.schuelerStamm;
String strReturn = "";
strReturn += st.familienname + "\n"
strReturn += st.rufname + "\n"
strReturn += st.schStrasse + " " + st.schHausnummer + "\n";
strReturn += st.schAnschrPLZ + " " + st.schAnschrOrt + "\n";
if (st.schStaat != null) {
    if (st.schStaat.anzeigeform.equals("Deutschland") == true) {
        strReturn += "\n";
    } else {
        strReturn += st.schStaat.anzeigeform + "\n";
    }
}
return strReturn;
```

- Vollständige Betriebsadresse mit Umbrüchen ausgeben:

```
String strBetriebName1 = "";
String strBetriebeAdresse = "";
```

```

def stb = obj?.schuelerStamm?.ausbildungBetrieb;
if (stb != null) {
    if (stb?.betriebName1 != null) {
        strBetriebName1 = stb?.betriebName1;

        strBetriebeAdresse = stb?.betriebName1 + "\n"
        if (stb?.betriebName2 != null) {
            if (stb?.betriebName2.length() > 1) {
                strBetriebeAdresse += stb?.betriebName2 + "\n"
            }
        }
        if (stb?.ansprVORNAMEN.get(0) != null) {
            strBetriebeAdresse += "zu Hd."
            if (stb?.ansprAkad.get(0) != null) {
                strBetriebeAdresse += " " + stb?.ansprAkad.get(0);
            }
            strBetriebeAdresse += " " + stb?.ansprVORNAMEN.get(0) + " "
+ stb?.ansprName.get(0) + "\n";
        }
        strBetriebeAdresse += stb?.anschrStrasse + " " +
stb?.anschriftHausnr + "\n";
        strBetriebeAdresse += stb?.anschriftPLZ + " " +
stb?.anschriftOrt + "\n";
        if (stb?.anschriftStaat != null) {
            if (stb?.anschriftStaat?.anzeigeform.equals("Deutschland")
== false) {
                strBetriebeAdresse += +
stb?.anschriftStaat?.anzeigeform + "\n";
            }
        }
    }
}
return strBetriebeAdresse;

```

- Name des aktuell eingeloggtten Benutzers

Dieser Code muss unter Modulbezogene-Funktionen > Globales Scripting eingefügt werden:

```

public class Tools {
    public static Benutzer getBenutzer() {
        String login = AuthUtils.getCurrentUserLogin();
        List benutzerList =
Locator.getInstance(BenutzerDAO.class).findByKennung(login);
        if (benutzerList.size() > 0) {
            Benutzer benutzer = (Benutzer) benutzerList.get(0);
            return benutzer;
        }
        return null;
    }
}

```

Dieser Code muss im Felddaufruf genutzt werden:

```
Tools.getBenutzer.getBenutzerName();
Tools.getBenutzer.getBenutzerVorname();
Tools.getBenutzer.getFullname();
Tools.getBenutzer.getVorname();
Tools.getBenutzer.getName();
Tools.getBenutzer.getNameVorname()
```

4) Ersetzen, Umformatieren, Arbeiten mit Datumswerten (Serienbriefe, Listengenerator)

Eine Gute Informationsquelle über Groovy ist auch eine [Online-Dokumentation](#).

- Aktuelles Datum ausgeben:

```
return new Date().format("dd.MM.yyyy")
```

- Aktuelles Datum + 10 Tage ausgeben:

```
datum = new Date()
datum = datum + 10
return datum.format("dd.MM.yyyy")
```

- Datum umformatieren

```
new SimpleDateFormat("dd.MM.yyyy").format(feld.getDate())
```

- Klammern entfernen

```
item.value = "alter Wert";
item.value = item.value.replace("]", "");
item.value = item.value.replace("[", "");
```

- Null-Werte abfangen (null wird von ASV immer dann ausgegeben, wenn ein Wert nicht vorhanden ist)

```
if (obj?.schuelerStamm.feldname != null) {
    // Feldwert zurückgeben
    item.value = obj?.schuelerStamm.feldname;
} else {
    // Entwertungszeichen oder Text zurückgeben
    item.value = "-";
}
```

5) Aufrufe im Globalen Scripting (Serienbriefe, Etiketten)

Serienbriefe oder Etiketten benötigen, um auf alle Felder der ASV zugreifen zu können, einige Imports im Global Scripting. Diese finden Sie im Serienbrief-Generator > Modulbezogene Funktionen > **Globales Scripting**.

```
import javax.swing.*;
import de.isb.svp.client.ze.scripting.objects.*;

import de.isb.svp.domain.pojo.SchuleStamm;
import de.isb.svp.domain.pojo.SchuleSchuljahr;
import de.isb.svp.domain.pojo.WlSchuljahr;
import de.isb.svp.domain.pojo.LehrerStamm;
import de.isb.svp.domain.pojo.Benutzer;
import de.isb.svp.domain.pojo.Anschrift;
import de.isb.svp.domain.pojo.Aussenstelle;
import de.isb.svp.domain.BenutzerDAO;
import de.isb.svp.domain.Schulart;
import de.isb.svp.auth.AuthUtils;
import de.isb.base.business Locator;
import de.isb.base.utils.Strings;
import bayern.asv.api.tools.SchulnameFormatter;
import bayern.asv.reports.api.st.schulwechsel.SchulwechselHelper;
import bayern.asv.reports.api.tools.unterzeichner.UnterzeichnerDatenHelper;
import java.util.*;
import java.lang.*
import java.text.*
import javax.swing.*
```

Quelle:

<https://www.asv.bayern.de/doku/> - **Amtliche Schulverwaltung - Dokumentation**

Permanenter Link:

<https://www.asv.bayern.de/doku/bers/berichte/groovy>

Letzte Änderung: **22.03.2022 08:17**