

Erstellen eines Extended Container

Dieses Tutorial beschreibt die notwendigen Schritte zum Erstellen eines Extended Containers.

Table of contents

1 Anforderungen.....	2
2 Was ist ein Extended Container?.....	2
3 Vorbereitung.....	2
3.1 Die Datenbankstruktur.....	2
3.2 Erstellen des Connectors.....	4
3.3 Erstellen der Connectorrequests.....	4
4 Erstellen des Containers 'BookInfo'.....	5
5 Erstellen des Containers 'AuthorsBooks'.....	8
6 Erstellen des Containers 'AllBooks'.....	8

1. Anforderungen

Dieses Dokument setzt voraus, dass Galaxy bereits auf einem System installiert ist, eine Beschreibung der [Installationsprozedur](#) findet sich auf der [Galaxy Webseite](#), sowie im [Installationspaket](#), dass dort ebenfalls heruntergeladen werden kann.

Für dieses Tutorial wird eine zusätzlich nur Datenbank zum Speichern der Beispieldaten benötigt.

2. Was ist ein Extended Container?

Innerhalb von SAGA.M31 - Galaxy gibt es zwei verschiedene Arten von Containern:

- Einfache Container
- Extended Container

Einfache Container sind dazu gedacht, simple Abfragen zu realisieren. Es ist z.B. nicht möglich, Requests miteinander zu verknüpfen.

Innerhalb eines Extended Containers werden Requests in einer definierten Reihenfolge ausgeführt. Ein Request kann auf die Ausgabefelder der vorherigen Request zugreifen und diese weiterverarbeiten.

Ein Extended Container wird in mehrere Schritte unterteilt, in jedem Schritt wird genau ein Connectorrequest ausgeführt. Falls ein Request Eingabefelder benötigt, gibt es zwei Möglichkeiten, die Quelle für diese Felder zu definieren:

- **Ein Containerfeld** - Der Wert wird mit aus einem Feld des Containerrequests übernommen
- **Ein Feld eines vorherigen Schrittes** - Der Wert wird aus einem Ausgabefeld eines vorher durchlaufenen Schrittes ausgelesen.

Als Ausgabefeld für einen Container kann jedes Feld definiert werden, das von einem der Schritte erzeugt wird

3. Vorbereitung

Um dem Tutorial folgen zu können, müssen die folgenden Vorbereitungen getroffen werden:

3.1. Die Datenbankstruktur

Dieses Tutorial basiert auf einer simplen Tabellenstruktur, mit deren Hilfe die Funktionalität

der "Extended Containers" demonstriert wird.

Es existieren zwei Tabellen, eine speichert eine Liste von Büchern, die zweite die Autoren dieser Bücher. Diese Tabellen sind wie folgt aufgebaut:

Tabelle "books"

Tabelle "authors"

Die Verbindung dieser Tabellen wird über das Feld **author** innerhalb der Tabelle **books** hergestellt. Dort wird die ID eines Datensatzes aus der **authors** Tabelle hinterlegt, somit ist jedem Buch ein Autor zugeordnet.

Um diese Tabellenstruktur zu erstellen, kann das folgende SQL-Script genutzt werden:

```
# table 'books'

CREATE TABLE books (
id int(3) unsigned default '0',
title varchar(255) default '0',
subtitle varchar(255) default NULL,
isbn varchar(20) default '0',
author int(10) unsigned default NULL
) TYPE=MyISAM;

INSERT INTO books VALUES("1", "Practical C++ Programming", "-",
"1-56592-139-9", "1");
INSERT INTO books VALUES("2", "Java Threads", "-", "1-56592-418-5", "2");
INSERT INTO books VALUES("3", "Enterprise Java Beans", "-",
"1-56592-605-6", "3");
INSERT INTO books VALUES("4", "Java in a Nutshell", "A Desktop Quick
Reference",
"1-56592-262-x", "4");
INSERT INTO books VALUES("5", "Refactoring", "Improving the design of
existing code",
"0-201-48567-2", "5");
INSERT INTO books VALUES("6", "Analysis Patterns", "Reusable Object
Models",
"0-201-89542-0", "5");
INSERT INTO books VALUES("7", "Java Foundation Classes in a Nutshell",
"A Desktop Quick Reference", "1-56592-488-6", "4");
INSERT INTO books VALUES("8", "Java Examples In a Nutshell",
"A tutorial companion to Java in a Nutshell", "1-56592-371-5", "4");

# table 'authors'

CREATE TABLE authors (
id int(3) unsigned default '0',
name varchar(255) default '0'
) TYPE=MyISAM;
```

```

INSERT INTO authors VALUES("1", "Steve Oualline");
INSERT INTO authors VALUES("2", "Scott Oaks");
INSERT INTO authors VALUES("3", "Richard Monson-Haefel");
INSERT INTO authors VALUES("4", "David Flanagan");
INSERT INTO authors VALUES("5", "Martin Fowler");

```

3.2. Erstellen des Connectors

Um die Verbindung zur Datenbank innerhalb von Galaxy herzustellen, muss zunächst ein Connector erstellt werden. Hierzu muss im Menü "Connectors" der Unterpunkt "Create Connector" ausgewählt werden. Der Type für den neuen Connector ist SQL, da eine Datenbank per SQL angesprochen werden soll:

Nachdem die Parameter eingetragen wurden, wird der Connector per "Save Connector" gespeichert.

3.3. Erstellen der Connectorrequests

Nach erfolgreichem Speichern des Connectors wird die Liste aller Connectoren angezeigt:

Nach Auswahl des Connectors kann mit Hilfe des "Create!" Buttons ein neuer Request für diesen Connector definiert werden. Für die Beispielcontainer werden insgesamt 5 Requests benötigt:

Requestname	Beschreibung
getBookInfo	Liefert die Felder <i>id</i> , <i>title</i> , <i>subtitle</i> , <i>isbn</i> und <i>author</i> aus der Tabelle books . Als Eingabefeld muss die ID des Buches angegeben werden
getAuthor	Liefert die Felder <i>id</i> und <i>name</i> aus der Tabelle authors . Als Eingabefeld muss die ID des Autors angegeben werden
getBooksForAuthor	Liefert die Tabelle <i>books</i> mit den Feldern <i>id</i> , <i>title</i> , <i>subtitle</i> , <i>isbn</i> und <i>author</i> aus der Tabelle books , als Eingabefeld wird die ID eines Autors erwartet. Der Request liefert alle Bücher dieses Autors.
findAuthorIdByName	Erwartet den Namen eines Autors als Eingabe und liefert seine ID zurück
getAllBooks	Liefert die Tabelle <i>books</i> mit den Feldern <i>id</i> , <i>title</i> , <i>subtitle</i> , <i>isbn</i> und <i>author</i> aus der Tabelle books .

	Dieser Request benötigt keine Eingabefelder, es werden alle Einträge der Tabelle zurückgeliefert.
--	---

Die Konfiguration der einzelnen Requests wird in den folgenden Screenshots beschrieben:

getBookInfo

getAuthor

getBooksForAuthor

findAuthorIdByName

getAllBook

4. Erstellen des Containers 'BookInfo'

Dieser Container benötigt eine Buch-ID als Eingabefeld, und liefert die folgenden Ausgabefelder:

- **title** - Der Titel des Buches
- **subtitle** - Der Untertitel des Buches
- **isbn** - Die entsprechende ISBN
- **author** - Den Namen des Autors

Da der Request 'getBookInfo' lediglich die ID des Autors liefert, ist es notwendig, einen weiteren Request ('getAuthor') auszuführen, um den Namen des Autors zu erhalten. Als Eingabefeld für diesen Request muss das Feld **author** aus dem Request 'getBookInfo' übernommen werden.

Um einen Extended Container zu erstellen muss zunächst der Menüpunkt "Container" angewählt werden. Der Unterpunkt "Create Extended Container" führt zu einem Formular, mit dessen Hilfe der Container konfiguriert wird:

Die Werte für 'Container Name', 'Handle' und 'Description' können übernommen werden.

Am einfachsten ist es, den Container nach der "Bottom Up" Methode zu definieren, wobei die Requests, die als letztes ausgeführt werden sollen, zuerst definiert werden.

Zunächst wird mit Hilfe des Links ' [Add output fields from request]' ein neues Ausgabefeld hinzugefügt. Dies führt zur Feldauswahl, die der Connectorübersicht ähnlich ist:

Links werden die Connectoren und Ihre Requests aufgelistet.

Bei Auswahl eines Request werden rechts seine Ausgabefelder aufgelistet.

Da wir den Namen des Autors zurückliefern wollen, wird dieses Feld selektiert. Das Feld wird mit Hilfe des Buttons "Add Selected Fields" in die Liste der Ausgabefelder übernommen:

Das neu definierte Ausgabefeld übernimmt den Namen des Connectorfeldes. Dieser kann manuell überschrieben werden.

Im Abschnitt "Open Dependencies" wird nun das Eingabefeld ('authorid') angezeigt, das benötigt wird, um den Namen eines Autors zu lesen. Für Felder, die in diesem Abschnitt angezeigt werden, wurde noch keine Datenquelle zugeordnet. Eine solche Datenquelle kann entweder ein Eingabefeld des Containers sein oder aber eines der Ausgabefelder eines vorherigen Schrittes.

In diesem Fall soll das Feld mit der Autoren-ID aus dem Request 'getBookInfo' bestückt werden. Im einfachsten Fall kann dies mit Hilfe des Links '[Link with Outputfield]' geschehen. Nach einem Klick auf diesen Link wird erneut die Requestübersicht angezeigt:

Nach Auswahl des Requests 'getBookInfo' auf der linken Seite werden seine Ausgabefelder angezeigt.

Anstelle der Checkbox zur Auswahl wird nun ein Link angezeigt, da nur ein Feld als Quelle gewählt werden kann. Durch Klicken des "Select" Links des Feldes 'author' wird dieses als Quelle für das Feld authorid des Requests 'getAuthor' definiert.

Nach der Auswahl wird erneut ein Feld im Abschnitt "Open Dependencies" angezeigt:

Der neu hinzugefügte Request, besitzt weitere Abhängigkeiten.

Es handelt sich um die bookid, die für den Request 'getBookInfo' benötigt wird. Diesmal soll ein Containerfeld als Quelle definiert werden. Hierzu muss lediglich die Checkbox der Abhängigkeit aktiviert und anschließend der Button "Add Selected Fields" betätigt werden. Anschließend ist die Abhängigkeit aufgelöst und dem Container wurde ein neues Eingabefeld hinzugefügt:

Das neu erstellte Eingabefeld.

Ein Klick auf "[Edit Container Flow]" führt zur Übersicht der in diesem Container verwendeten Requests:

Die einzelnen Schritte im Container Flow

Jeder Schritt wird durch eine der Boxen repräsentiert. In dieser Ansicht werden die folgenden Informationen dargestellt:

- **Name** - Jeder Schritt hat einen Namen, dieser kann über die Textbox in der Linken

oberen Ecke verändert werden.

- **ConnectorRequest** - Zentriert wird der Request angezeigt, der in diesem Schritt ausgeführt wird. (Format: CR [ConnectorName : RequestName])
- **Position** - Mit Hilfe der Pfeile rechts oben kann der Request nach oben oder unten verschoben werden. Das rote X löscht einen Schritt.
- **Abhängigkeiten** - Im unteren Bereich werden die Felder angezeigt, die der Request zur Ausführung benötigt. Die Quelle kann mit Hilfe der Dropdown-Listen ausgewählt werden und über den Button "Add Selected as new Containerfield" können Felder einem neuen Containerfeld zugewiesen werden.

Im Abschnitt "Add a new Step" kann durch Auswahl des Connectors und des Requests ein neuer Schritt hinzugefügt werden.

Für diesen Container ist es nicht notwendig, in dieser Ansicht Einstellungen vorzunehmen, ein Klick auf "Go Back" führt zurück zur Basismaske des Containers.

Dort müssen nun noch die Ausgabefelder aus dem Schritt 'BookDB_getBookInfo' erstellt werden. Dies geschieht mit Hilfe des Links "[Add Unmapped Outputfield]", der ein neues Feld erstellt. Dem neuen Containerfeld wird nun noch eine Datenquelle (ein Resultat aus einem der Schritte) zugeordnet. Hierzu muss zunächst der entsprechende Schritt aus der Liste 'source' ausgewählt werden:

Auswahl des Schrittes

Nach Auswahl des Schrittes wird eine zweite Liste angezeigt, in der alle Felder dieses Schrittes angezeigt werden:

Auswahl des Feldes

Nach Auswahl des Quellfeldes wurde dem Containerfeld eine Datenquelle zugeordnet. Der Name in der ersten Spalte wird aus dem Quellfeld übernommen, kann aber beliebig angepasst werden.

Mit dem selben Verfahren werden nun noch die Felder "isbn" und "subtitle" zugeordnet:

Containerfelder nach Zuordnung

Der Container ist nun erstellt und kann gespeichert werden. Hierzu kann entweder der Button "Save" verwendet werden, oder auch "Save and Sample run". Letztere speichert den Container und wechselt direkt zur Testseite:

SampleRun des Containers

Dort wird ein Eingabefeld bereitgestellt, in das die ID eines der Bücher eingetragen werden muss. Ein Klick auf "Run" liefert dann das folgende Ergebnis:

SampleRun des Containers - Ergebnis

Der Container ist nun fertiggestellt, und kann über die verschiedenen Schnittstellen von Galaxy abgefragt werden.

5. Erstellen des Containers 'AuthorsBooks'

[TODO]

6. Erstellen des Containers 'AllBooks'

[TODO]