

# Informationscontainer

## Table of contents

1 Was ist ein Container?.....	2
2 Normaler Container.....	2
2.1 Erstellen eines Container.....	2
3 Extended Container.....	4
3.1 Erstellen eines Extended Containers.....	4
4 Testen des Containers.....	4

## 1. Was ist ein Container?

Informationscontainer stellen das Kernkonzept hinter SAGA.M31 - Galaxy - dar. Sie bieten eine einheitliche, konsistente und definierte Schnittstelle für die Integration von Daten in beliebige Anwendungen.

Ein Container wird definiert durch seine Ein- und Ausgabefelder, die die Parameter für eine Abfrage, sowie deren Rückgabewert darstellen. Container selbst definieren hierbei lediglich, die Schnittstelle, nicht die eigentliche Quelle für die Daten.

Um einen Container nun mit Daten zu bestücken, werden seine Felder mit denen eines beliebigen Connectoren verknüpft. Durch diese Abstraktion bietet der Container einen flexiblen Datenzugriff, ohne die Notwendigkeit, die eigentliche Quelle der Daten kennen zu müssen. Bei einer Migration müssen z.B. lediglich die Felder des Containers mit anderen Connectorfeldern verknüpft werden. Da sich die Schnittstelle nach außen hin nicht verändert, müssen die Applikationen, die auf Galaxy aufsetzen, nicht verändert werden.

SAGA.M31 -Galaxy - Informationscontainer können mit Hilfe des Galaxy wui (**Web User Interface**) komfortabel erstellt, konfiguriert und gewartet werden.

Container sind mit Hilfe einer **WSDL** Definition beschrieben und als **Web Service** bereitgestellt, dies ermöglicht die einfache Integration in beliebige Programmiersprachen und Anwendungen.

Bei der Implementierung der Web Services Schnittstelle wurde besonderen Wert auf Kompatibilität gelegt, so wurde beispielsweise der **Document/Literal - Style** gewählt, um die Integration in verschiedene Frameworks wie z.B. **Microsoft's .NET** zu unterstützen.

Seit der Version 1.4 von SAGA.M31 - Galaxy stehen zwei verschiedene Typen von Containern zur Verfügung. Neu hinzugekommen sind die sogenannten "Extended Container", mit deren Hilfe verschachtelte Abfragen realisiert werden können.

## 2. Normaler Container

Ein Normaler Container wird zum Ausführen von einem oder mehreren Connectorrequest genutzt. Eingabefelder für die Requests werden bei diesem Containertyp als Containerfelder bereitgestellt, d.H. ein verschachteln von Requests ist nicht möglich.

### 2.1. Erstellen eines Container

Um einen Container zu erstellen, muss zunächst der Menüpunkt "Container", anschließend der Unterpunkt "Create Conainer" ausgewählt werden. Es wird Formular angezeigt, mit

dessen Hilfe der Container erstellt wird:

Feld	Beschreibung
Container Name	Ein aussagekräftiger Name für diesen Container
Handle	Ein eindeutiger Bezeichner für diesen Container. Dieser wird beim Aufruf über die verschiedenen Service-Schnittstellen von Galaxy verwendet, um den Zielcontainer für die Abfrage zu definieren.
Description	Eine Beschreibung für diesen Container
WSDL Public available	Ist diese Checkbox aktiviert, so ist die WSDL, die diesen Container beschreibt, über eine HTTP Anfrage verfügbar. Dies hat den Vorteil, dass sie ohne Umwege in die meisten Toolkits für Webservices eingebunden werden kann. Hierdurch wird allerdings die Schnittstelle des Containers öffentlich bekannt gegeben, was unter Umständen aus Gründen der Sicherheit nicht erwünscht ist. Ist die Checkbox nicht aktiviert, muss die WSDL über das Benutzerinterface von Galaxy heruntergeladen, lokal gespeichert und dann eingebunden werden.
Open Dependencies	Hier werden alle Eingabefelder aufgelistet, die für die ausgewählten Requests benötigt wurden, aber noch keinem Containerfeld zugeordnet wurden. Um ein neues Containerfeld für eine der Abhängigkeiten anzulegen, muss zunächst die entsprechende Abhängigkeit durch aktivieren der Checkbox ausgewählt werden. Mit einem Klick auf den Button "Add Selected Fields" wird ein neues Containerfeld erstellt, dem das ausgewählte Eingabefeld zugeordnet ist.
Output Fields	<p>Hier wird die Liste aller Felder angezeigt, die dieser Container zurückliefert. Neue Ausgabefelder werden mit Hilfe des Links "[Add more Outputfields]" hinzugefügt, hinter dem sich eine Übersicht aller Requests und deren Felder verbirgt. Dort können die gewünschten Felder ausgewählt und anschließend hinzugefügt werden.</p> <p>Der Link "[Remap]" dient dazu, das Containerfeld einem neuen Connectorfeld zuzuordnen, ein Klick</p>

	führt ebenfalls zur Auswahl der Connectorfelder. Dort kann das entsprechende Feld nun ausgewählt werden.
Input Fields	Hier werden die Eingabefelder aufgelistet. Ähnlich wie schon bei den Ausgabefelder existiert ein Link ("Add more input fields"), über den weitere Eingabefelder hinzugefügt werden können.

### 3. Extended Container

Um komplexere Abfragen realisieren zu können stellt SAGA.M31 - Galaxy die "Extended Container" bereit. Dort ist es möglich eines der Ausgabefelder eines Requests als Eingabefeld für einen weiteren Request zu definieren. Auf diese Art lassen sich beliebig tiefe Abfragen definieren.

In Version 1.5 wurden den "Extended Containern" eine neue Funktion hinzugefügt, das sogenannte "Inlining". Hierbei wird eine Tabelle aus einem der vorherigen Request als Eingabefeld für einen weiteren Request definiert. In diesem Fall wird der geschachtelte Request genau einmal für jede Zeile der Tabelle ausgeführt. Die Ausgabefelder werden in die übergelagerte Tabelle als neue Spalten hinzugefügt, so dass nach aussen hin nur eine Tabellenstruktur sichtbar ist. Diese Funktion ist mit dem JOIN-Befehl von SQL vergleichbar.

#### 3.1. Erstellen eines Extended Containers

Einen Extended Container kann im Container-Menü über den Eintrag "Create Extended Container" erstellen. Das Formular zum Erstellen eines solchen Containers unterscheidet sich nur geringfügig von dem eines normalen Containers.

### 4. Testen des Containers

In der Containersicht (erreichbar über den Menülink "Manage Containers") steht für jeden verfügbaren Container ein Link für den "Sample Run" zur Verfügung. Ein Klick darauf führt zur Testseite.

Dort werden zunächst alle Eingabefelder aufgezählt, die für den Containerlauf erforderlich sind. Für jedes Feld wird ein Textfeld bereitgestellt, in das ein Wert für den Probelauf eingegeben werden muss.

Ein Klick auf den "Fire!"-Button startet die Abfrage und die zurückgelieferten Felder werden in einer HTML-Tabelle dargestellt.

Um sich die Strukturen für die XML-Anfrage und die XML-Antwort für diesen Container anzuschauen, muss die Checkbox "Show XML Request/Response" ausgewählt werden. Nach der Ausführung werden die Strukturen unterhalb der zurückgelieferten Containerfelder dargestellt.

Um Fehler, die während der Ausführung auftreten analysieren zu können, gibt es noch die Checkbox "Show the Trace". Ist diese aktiviert, so werden detaillierte Informationen über die Verarbeitung innerhalb der Connectoren angezeigt. Im Falle eines SQL Connectoren z.B. welche SQL Statements konkret abgesetzt werden, sowie die zurückgelieferten Daten. Das Format der Ausgabe variiert zwischen den verschiedenen Connectortypen.