

Anschluss

- [Definition](#)
- [Schnittstellen](#)
- [Anschluss](#)
 - [Beispiel](#)
 - [PhoneCallRequestContext](#)
 - [Schnittstellen entwickeln](#)

Definition

Menüaufruf: (Administration) - (Anschluss)

Mit den Anschlüssen wurde in Nuclos eine neue Möglichkeit geschaffen, einen Nuclos Server noch weiter in seine Systemlandschaft zu integrieren, mittels sogenannter "Nuclos-Schnittstellen". Erstes Anwendungsszenario in einem Kundenprojekt war die Anbindung einer Telefonanlage, welche inzwischen auch produktiv genutzt wird. Theoretisch sollten aber auch andere Systeme angebunden werden können, wie z.B. SMS-Dienste, oder alles was über Java erreichbar ist (einschließlich RESTful Services).

Schnittstellen

Der Unterschied zu einer Regel liegt darin, dass über eine Nuclos-Server-Extension (in Java entwickelt, als JAR zur Verfügung gestellt: [siehe auch Entwicklungsumgebung](#)) dauerhaft eine Schnittstelle läuft, die z.B. die Events der Telefonanlage entgegennimmt. Die Integration der Events dieser Schnittstellen in die eigene Businesslogik (Nuclet: BO, Regeln, etc.) erfolgt dann aber wieder einfach und flexibel per Regeln. Somit könnte man solche eine Schnittstelle auch als Treiber bezeichnen, wie man sie für Hardware kennt. Existiert eine Schnittstelle, wie in unserem ersten Anwendungsszenario eine Schnittstelle zu ACHAT (<http://www.authensis.de/de/achat>), kann diese theoretisch auch für andere Systeme und Nuclets interessant werden, und wird sicherlich auch demnächst in einem Nucletshop angeboten.

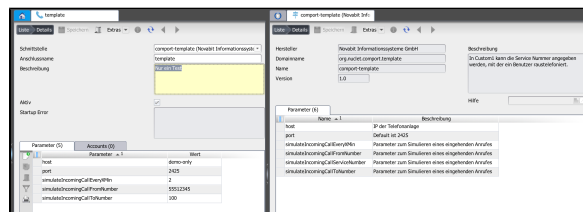
Die Implementierung solch einer Schnittstelle erfolgt in Java und basiert normalerweise nur auf der Nuclos-API, womit eine Abwärtskompatibilität wie bei Regeln geschaffen werden kann. "Kann" deshalb, weil Nuclos selbst nicht ausschließen kann, dass die Extension auch auf andere Klassen von Nuclos, als die der API zugreift.

Anschluss

Ein Anschluss ist die Verwendung einer Schnittstelle unter Angabe der für die Schnittstelle notwendigen Parameter, wie Hostadressen, Ports, Usernamen etc. Eine Schnittstelle kann somit beliebig oft als Anschluss in einem System konfiguriert werden. Neben den Parameter können auch Nuclos-Benutzer mit Accounts des Anschlusses verknüpft werden. Die Verwendung dieser Informationen steht dann intern der Schnittstelle zur Verfügung.

Beispiel

Am Beispiel einer Dummy-Schnittstelle "comport-template" sollte die Theorie etwas verständlicher werden. Der Sourcecode dieses Templates steht [hier \(https://bitbucket.org/nuclos/com-interface-template\)](https://bitbucket.org/nuclos/com-interface-template) zur Verfügung und kann frei verwendet werden. Zusätzlich hängt diesem Artikel eine fertig kompilierte Schnittstelle an.

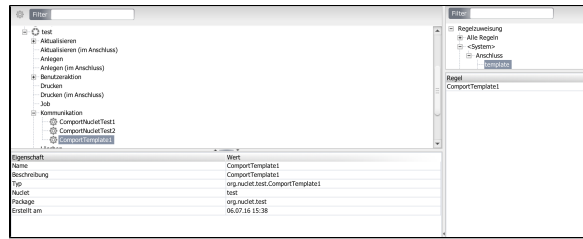


Wird der Server mit der comport-template Server-Extension gestartet, steht diese Schnittstelle im Schnittstellen Dropdown bei einem neuen Anschluss zur Auswahl (links im Bild). Details anzeigen auf die ausgewählte Schnittstelle zeigt ein paar Informationen zur Schnittstelle selbst (rechts im Bild).

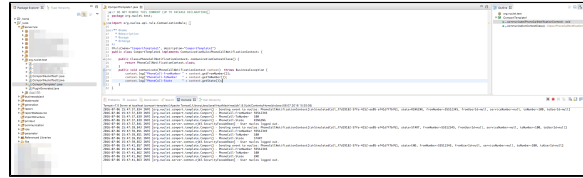
Wenn ein Anschluss wie im Bild gezeigt angelegt wird, nimmt die Schnittstelle gleich ihren Dienst auf. In unserem Fall würde zur Demonstration alle 2 Min. ein Telefonanruf eingeben (siehe eingestellte Parameter).

Communication-Rule

Nun können wir auch eine Communication-Rule unserem Anschluss über den Regelmanager hinzufügen.



Der Inhalt besteht nur aus einem Logging des Events.



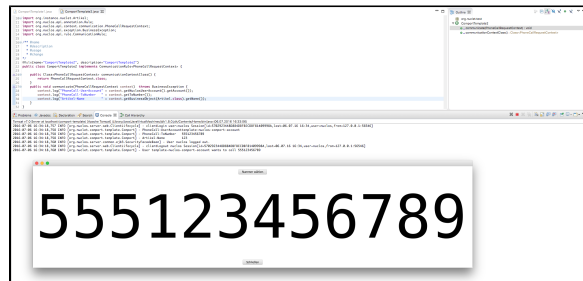
PhoneCallRequestContext

Eine besondere Aufmerksamkeit kann man dem `PhoneCallRequestContext` widmen. Dieser ermöglicht nämlich einem Benutzer eine Telefonnummer direkt aus Nuclos heraus anzurufen. Dabei telefoniert er nicht über Nuclos direkt, sondern es wird ein Request für ein neues Telefonat an die Telefonanlage oder dem Endgerät übermittelt (sofern diese so etwas unterstützt). Hierfür sind einige Voraussetzungen zu erfüllen:

1. Das Interface arbeitet mit dem [PhoneCallRequestContext](#) (wie unser Template)
2. Es wird der BO-Attributstyp `PhoneNumber` für die Ablage von Telefonnummer verwendet
3. Dem aktuellen Benutzer ist für am Anschluss ein Account hinterlegt

Die vergrößerte Anzeige der Telefonnummer (siehe folgenden Screenshot) ist für diese Attribute zwar immer zu sehen, jedoch erscheint dann zusätzlich der Button "Nummer wählen".

Im Screenshot ist auch eine weitere Regel zu sehen, die speziell mit dem `PhoneCallRequestContext` arbeitet. Zusätzlich zum Account des Benutzer und der anzurufenden Nummer steht auch das BO zur Verfügung, aus dem die Rufnummer gewählt wurde.



Schnittstellen entwickeln

Derzeit enthält unsere API 2 Typen von Schnittstellen:

1. `PhoneCall` (für Telefonanlagen)
2. `GenericNotification` (für Sonderfälle)

Weitere Typen können ggf. hinzugefügt werden. Sprechen Sie uns an!

Implementiert eine Schnittstelle `PhoneCall`, kann die Schnittstelle sehr leicht ausgetauscht werden wenn man z.B. die Anlage wechselt, ohne das die Businesslogik an sich angefasst werden muss. Man legt nur einen neuen Anschluss an, administriert ggf. die Accounts und weist die Regeln zu.

Communication-Context

Neben dem im Beispiel gezeigten [PhoneCallNotificationContext](#) bzw. [PhoneCallRequestContext](#) gibt es auch noch einen universellen [GenericNotificationContext](#) der nur mit einer Key-Value-Map arbeitet.

Einige Hinweise

- Die Implementierung richtet sich nach dem System, zu dem eine Schnittstelle aufgebaut werden soll. Möglicherweise sind Drittbibliotheken zu verwenden und zu erwerben. Oder das System verfügt über einen RESTful Service.
- Die Art und Weise wie die API Interfaces implementiert werden müssen kann dem template entnommen werden [org.nuclet.comport.template](#). Wichtig ist auch die Angabe des verwendeten Java-Package in der [comport-template-server-beans.xml](#)

Datei	Geändert
Java-Archiv comport-template-server-1.0-SNAPSHOT.jar	Juli 06, 2016 by M aik Stüker

