REST - Schnittstelle

Beschreibung

Die REST-Schnittstelle ermöglicht es, verschiedene Konfigurationen der STARFACE aus einer eigenen Anwendung zu steuern. Sie basiert auf einem einfachen zustandslosem Protokoll, welches Zugriff auf verschiedene Ressourcen der STARFACE ermöglicht. Die Abfragen werden über unterschiedliche HTTP-Methoden durchgeführt.

HTTP-Methoden

Befehl	Beschreibung
GET	Fordert die angegebene Ressource vom Server an. Die Konfiguration am Server wird nicht verändert, weshalb GET als *sicher* bezeichnet wird
POST	Fügt eine neue Ressource hinzu und wird für die Authentifizierung verwendet
PUT	Die angegebene Ressource wird geändert
DELETE	Löscht die angegebene Ressource

Ressourcen

Typ	Beschreibung
Login	Wird benötigt zur Authentifizierung der Schnittstelle
Adressbuch	Abfrage von Adressbüchern, Schema und Adressen; Hinzufügen, ändern und löschen von Adressen
Benutzer	Abfrage von Nutzern und Nutzerstammdaten; Hinzufügen, ändern und löschen von Nutzern
Umleitungen	Abfrage und Änderung von Umleitungen für Benutzer
iFMC	Abfrage, hinzufügen, ändern und löschen von iFMC-Konfigurationen für Benutzer
Funktionstasten	Abfrage, hinzufügen, ändern und löschen von Funktionstasten für Benutzer
Telefone	Abfrage, hinzufügen, ändern und löschen von Telefonen zu Benutzern
Rufnummern	Lesender Zugriff auf die Rufnummernzuordnungen zu Benutzern

Zugriff

Der Zugriff auf die Schnittstelle ist über die Basis-URL der STARFACE, inklusive dem Zusatz */rest* möglich.

Beispiel

```
http://[IP/URL der STARFACE]:80/rest
https://[IP/URL der STARFACE]:443/rest
```

Authentifizierung

Zur Absicherung der REST-Schnittstelle, muss bei jeder Abfrage ein Authentifizierungstoken mitgesendet werden. Dieser Token muss vor der Verwendung weiterer Abfragen generiert werden und ist dann für 4 Stunden gültig. Diese Abfrage funktioniert in dieser Form, ab der Version 6.4.2.12 der STARFACE.

Schritt 1

Die Adresse der STARFACE für dieses Beispiel ist: https://example.starface-cloud.com

Beispiel 1

```
HTTP-GET zu https://example.starface-cloud.com/rest/login

Content-Type=application/json
X-Version=2

Rückgabe:
{
"loginType":"Internal",
"nonce":"pds24hmip1ctbogn1l8ujvs5u4",
"secret":null
}
```

Schritt 2

Die Generierung des verschlüsselten Passworts (secret) wird folgendermaßen aufgebaut:

LoginID:SHA512(LoginID+nonce+SHA512(password))

> **Hinweis:** Bei Nutzung eines Active Directory bildet sich das Passwort (secret) nicht aus den SHA Hashes sondern aus dem folgenden Aufbau:
>
> Base64Encode(LoginID+nonce+password)

Das folgene Beispiel geht davon aus das die folgenden Zugangsdaten genutzt werden:

Beschreibung	Wert
Benutzer	0001
Kennwort	password

Das SHA512 von „password“ ist b109f3bbbc244eb82441917ed06d618b9008dd09b3befd1b5e07394c706a8bb980b1d7785e5976ec049b46df5f1326af5a2ea6d103fd07c95385ffab0cacbc86

Unser Secret berechnet sich also so:

“0001:SHA512(0001pds24hmip1ctbogn1l8ujvs5u4b109f3bbbc244eb82441917ed06d618b9008dd09b3befd1b5e07394c706a8bb980b1d7785e5976ec049b46df5f1326af5a2ea6d103fd07c95385ffab0cacbc86)”

> **Hinweis:** Die farblichen Markierungen dienen außschließlich der optischen Trennung der verschiedene Teilstücke.

Was folgenden String ergibt:

0001:
8763072240d007e18b92ce58ce76bb244377e1f41bde6811ce7c17adab4977f0d00502a6a9a1b1d70a51824626b86df82699fe993b458a4818817375078983b3

Schritt 3

Der in Schritt 2 ermittelte String kann mit einem POST auf http://host/rest/login mit folgendem Body verwendet werden:

```
HTTP-POST zu https://example.starface-cloud.com/rest/login

Header:
{
Content-Type : application/json
X-Version : 2
}

Body:
{
"loginType": "Internal",
"nonce": "pds24hmip1ctbogn1l8ujvs5u4",
"secret":
"0001:
8763072240d007e18b92ce58ce76bb244377e1f41bde6811ce7c17adab4977f0d00502a6a9a1b1d70a51824626b86df82699fe993b458a48
18817375078983b3"
}


Rückgabe:
{
"authToken": "abcdef12345"
}
```

Bei erfolgreicher Authentifizierung findet sich im Response Body der Authentifizierungstoken

authToken=abcdef12345

Schritt 4

Der zurückgelieferte Authentifizierungstoken ist nun für 4 Stunden gültig und muss jeder weiteren Abfrage als Header vom Typ *authToken* hinzugefügt werden (z.B. authToken:abcdef12345).

Beispiel für die Authentifizierung

Für dieses Beispiel der Authentifizierung wurde Python 3.6.3 verwendet. Dabei wurden nur mitgelieferte Libraries genutzt.

Python-Login.py

In der folgenden Beispieldatei wird zusätzlich zum Login auch noch gezeigt, wie ein json-Element anlegt und als Payload an den Server mitgesendet wird.

Python-LoginAndCreateUser.py

Swagger-Dokumentation

Die Umfangreiche Dokumentation ist als Swagger-Datei verfügbar. Zum lesen der Doku, kann unter Anderem der Swagger-Editor verwendet werden. Diesen finden Sie unter http://editor.swagger.io/.

im Editor laden Sie Datei über den Menüpunkt File / Import File ...

Version	Download
6.6.0	STARFACE Rest V6_6_0_X.yaml
6.5.1	STARFACE Rest V6_5_1_X.yaml
6.5.0	STARFACE Rest V6_5_0_X.yaml