

REST-APIs

- [Definition](#)
- [Einrichtung](#)
- [Verwendung](#)

Definition

Nuclos bietet die Möglichkeit aus einer OpenAPI-Spezifikation mittels des `OpenAPI Generator` einen Java-Client zur Anbindung von externen REST-Services zu generieren. Dieser Client kann dann in den Regeln verwendet werden.



Dieses Feature steht ab der Version 4.2023.29 zur Verfügung.

Einrichtung

Um einen Client für einen REST-Service erzeugen zu lassen, muss die Servicespezifikation unter *Administration/REST APIs* registriert werden. Danach werden der Struktur entsprechende Klassen im Verzeichnis `<NUCLOS-HOME>/data/codegenerator/src-conf-rest/` generiert, die in Regeln verwendet werden können.

Feld	Bedeutung
Name	Name der REST-API: eindeutig; wird u.a. als Name für die Adapter-Klasse verwendet, bspw <code>example.rest.NuclosRESTAdapter</code> (liegt im Package des Nuclets)
Beschreibung	optional
Haupt-Package	Name für das api - Package; Default: <code><nuclet_package>.invoker</code>
API-Package	Name für das api - Package; Default: <code><nuclet_package>.api</code>
Modell-Package	Name für das models - Package; Default: <code><nuclet_package>.models</code>
Bibliothek für HTTP-Client-Template	Bibliothek für HTTP-Client; Default: <code>jersey2</code>
Bibliothek für Date-Objekte	Bibliothek für Date-Objekte; Default: <code>java8</code>
BigDecimal als String-Objekte behandeln	<input type="checkbox"/>
Serializable implementieren?	<input type="checkbox"/>
Datei	<input type="text" value="NuclosDE.json"/>

ListeDetails

REST APIs

Name

Nuclos

Beschreibung

Haupt-Package

de.nuclos.rest

API-Package

de.nuclos.rest.api

Modell-Package

de.nuclos.rest.models

Bibliothek für Http-Client-Template

jersey2

Bibliothek für Date-Objekte

java8

BigDecimal als String-Objekte behandeln

☐

Serializable implementieren?

☐

Datei

NuclosDE.json

Datei	OpenAPI-Spezifikation als JSON oder YAML
-------	---

<https://openapi-generator.tech/docs/generators/java> bietet weitere Informationen zu den Einstellungsmöglichkeiten des Generators.



Mglw. weitere Abhängigkeiten

Bei der Verwendung einer anderen HTTP-Client-Bibliothek oder Date-Bibliothek als den Standard, werden u.U. weitere Bibliotheken benötigt, die als serverseitige Extension zur Verfügung gestellt werden müssen.

Verwendung

Die Erzeugung der Client-Klassen erfolgt ohne Abhängigkeit zu einem Endpunkt. Daher muss vor dem Aufruf der API-Methoden ein `ApiClient` für den anzusprechenden Endpunkt erzeugt werden. Zur Vereinfachung bietet Nuclos hier eine Adapter-Klasse (bspw. `example.rest.NuclosRESTAdapter`) an, die mit dem Endpunkt initialisiert werden kann.

Der Aufruf der API-Methoden erfolgt über die `Api`-Klassen (s. API-Package). Diese können auch über die Adapter-Klasse bezogen werden, bspw. durch den Aufruf von `adapter.getNucloSystemApi()`.

Nachfolgend findet sich eine Beispiel-Implementation zum Aufruf mehrerer API-Methoden (GET und POST) mit vorherigem Login und Authentifizierung über ein Session-Cookie.

```
String restAPIsBasePath = ParameterProvider.getNucletParameter(TestOtherNucletParameter.RestAPIsBasePath);
if (StringUtils.isBlank(restAPIsBasePath) || "null".equals(restAPIsBasePath)) {
    restAPIsBasePath = "http://localhost/nuclos-war";
}
final NuclosRESTAdapter nuclosRESTAdapter = NuclosRESTAdapter.newInstance(restAPIsBasePath);
try {
    StringBuilder sb = new StringBuilder();
    final Serverstatus serverstatus = nuclosRESTAdapter.getNucloSystemApi().restServerstatusGet();
    sb.append(new ObjectMapper().writeValueAsString(serverstatus));
    sb.append("\n");

    //LOGIN
    final LoginParams loginParams = new LoginParams();
    loginParams.setUsername("nuclos");
    loginParams.setPassword("");
    final LoginInfo loginInfo = nuclosRESTAdapter.getNuclosAuthenticationApi().login1(loginParams);
    final String sessionId = loginInfo.getSessionId();
    nuclosRESTAdapter.getApiClient().addDefaultCookie("JSESSIONID", sessionId);

    DebugSql debugSQL = new DebugSql();
    debugSQL.setDebugSQL("false");
    nuclosRESTAdapter.getNuclosMaintenanceApi().setDebugSQL(debugSQL);

    debugSQL = nuclosRESTAdapter.getNuclosMaintenanceApi().getDebugSQL();
    sb.append(new ObjectMapper().writeValueAsString(debugSQL));
    sb.append("\n");

    debugSQL = new DebugSql();
    debugSQL.setDebugSQL("true");
    nuclosRESTAdapter.getNuclosMaintenanceApi().setDebugSQL(debugSQL);
    debugSQL = nuclosRESTAdapter.getNuclosMaintenanceApi().getDebugSQL();
    sb.append(new ObjectMapper().writeValueAsString(debugSQL));
    sb.append("\n");

    testBO.setResult(sb.toString());
} catch (ApiException | JsonProcessingException e) {
    throw new BusinessException(e);
}
```