

# NEVV: Schnittstellendokumentation für OData

## Allgemeines

Die Service-Strukturen sind nicht komplett fixiert und können sich je nach neuen Anforderungen ändern. Da Konsumenten des Service auf diese Änderungen entsprechend reagieren können sollten, empfehlen wir ggf. einen halbjährlichen Abgleich.

## Antworten

An jeden der Aufrufe jederzeit `$format=json` angefügt werden, um statt XML dann JSON als Antwort zu erhalten.

## URL

Servicepfad:

[https://studentservices.uzh.ch/sap/opu/odata/uzh/vvz\\_data\\_srv/](https://studentservices.uzh.ch/sap/opu/odata/uzh/vvz_data_srv/)

Der Service benötigt für den Aufruf immer ein Entity, das an den Pfad angehängt wird. Im Folgenden wird dies beispielhaft so aussehen:

**PSearchSet?\$skip=0&\$top=100&\$orderby=LastName  
asc&\$filter=OrgId eq '50000008' and PiqYear eq '2016' and  
PiqSession eq '003'&\$inlinecount=allpages**

Die Funktionalität dieses Beispiels wird weiter unten erklärt.

# Schnittstellendetails

## Vorwort

Die Detailsichten sind im Grunde filterlos.

Es gibt keine Operation, die eine Menge von Detailseiten liefert. Das ist auch nicht unbedingt sinnvoll für die Performance, die Details sind eine nicht unerhebliche Datenmenge.

Natürlich könnte das nachgerüstet werden, Anfangs wurde in Proof-of-Concepts die Sinnhaftigkeit dazu ermittelt um zum Beispiel die Modulliste auf der linken Seite mit allen Details zu füllen. Es stellte sich heraus, dass aus Performancegründen darauf verzichtet werden sollte, denn es würde hier so einiges gesendet.

## Ermittlung einer Objektmenge mit Details

Es wird hier eine Lösung in zwei Schritten angeraten:: Aufruf der Suche mit Fakultätsfilter für die jeweiligen Objekte. Danach aus den Antworten das Laden der einzelnen Details.

## Suche mittels NEVV-OData

Vorab für die Suche: Alle Dozierenden der MNF wären über 600 Ergebnisse. Die Parameter \$skip und \$stop erzwingen hier ein Paging.

Für die hier beschriebene Suche wurde das Paging vollständig gefordert, das bedeutet, dass ein Zugriff ohne \$skip und \$stop unmöglich ist. Diese Entscheidung wurde getroffen, um die Serverlast durch externe Aufrufer etwas zu begrenzen.

Für die beispielhaft hier aufgeführten Aufrufe bedeutet das, dass jeweils 100 Ergebnisse (Wert in \$stop) geliefert und mit \$skip 0 Ergebnisse dabei gemäß der Sortierung in \$orderby überprungen werden. Sollen die nächsten 100 Ergebnisse ermittelt, ändert sich also einfach der Wert von \$skip auf 100 usw.

Der \$inlinecount parameter sorgt dafür, dass ausgegeben wird, wieviele Ergebnisse insgesamt vorhanden wären – d.h. hier lässt sich programmatisch herausfinden, wieviele Ergebnisse die Suche haben könnte und damit ein seitenweises nachladen implementieren. Das ist ratsam für die Performance.

Im NEVV werden jeweils nur immer 20 Ergebnisse vorgeladen und bei Scrolling weitere geholt. Das hat sich als recht effektiv erwiesen und reduziert die Serverlast und den Speicherbedarf im Browser enorm. Die Fakultätsfilter sind relativ simpel. Hier wird in \$filter nur **OrgId** mitübergeben (also die Objektid der Fakultät).

Für die Suche wäre das beispielhaft an den Servicepfad angehängt:

```
PSearchSet?$skip=0&$stop=100&$orderby=LastName  
asc&$filter=OrgId eq '50000008' and PiqYear eq '2016' and  
PiqSession eq '003'&$inlinecount=allpages
```

## Suche nach Personen

```
PSearchSet?$skip=0&$stop=100&$orderby=LastName  
asc&$filter=OrgId eq '50000008' and PiqYear eq '2016' and
```

*PiqSession eq '003'&\$inlinecount=allpages*

Das wäre ein beispielhafter Aufruf für die Dozierenden der MNF.

## Filterkriterien (\$filter)

Allgemein können die Filterkriterien bestimmt werden, indem

**SearchPageConfigSet?\$expand=Values** gelesen wird.

- Seark - Freitextsuche, zu verwenden mit **substring(...)**
- FirstName - Freitextsuche, zu verwenden mit **substring(...)**
- LastName - Freitextsuche, zu verwenden mit **substring(...)**
- Function - Eine Id der Funktion einer Person (mit *eq*)
- OrgId - Zugehörigkeit, die Id einer Organisationseinheit (mit *eq*)

## Sortierung

- \$orderby=LastName asc
- \$orderby=FirstName asc
- \$orderby=Title asc - alphabetische Sortierung nach Titel
- \$orderby=FunctionTxt - alphabetische Sortierung nach Funktion

## Suche nach Studienprogrammen

*CgSearchSet?\$skip=0&\$top=100&\$orderby=CgStext*

*asc&\$filter=OrgId eq '50000008' and PiqYear eq '2016' and*

*PiqSession eq '003'&\$inlinecount=allpages*

Das wären die Studienprogramme der MNF.

# Filterkriterien (\$filter)

TBD

## Sortierung

TBD

## Suche nach Modulen

*SmSearchSet?\$skip=0&\$top=100&\$orderby=SmStext asc&\$filter=(OrgId eq '500000008' or OrgId eq '500000007' or OrgId eq '500000003') and PiqYear eq '2016' and PiqSession eq '003'&\$inlinecount=allpages*

Die Module für MNF, PhF und WWF. Hier ist auch der Aufbau für den \$filter interessant, da das lediglich eine or-Verknüpfung ist.

Anstelle des Filterattributs **OrgId** für Fakultäten, kann auch mit dem Filterattribut **OfferingOrgId** nach anbietenden Organisationen gefiltert werden.

## Filterkriterien

Allgemein können die Filterkriterien bestimmt werden, indem

**SearchPageConfigSet?\$expand=Values** gelesen wird.

- Seark - Freitextsuche, zu verwenden mit **substring(...)**
- SmShort - Freitextsuche, zu verwenden mit **substring(...)**
- SmStext - Freitextsuche, zu verwenden mit **substring(...)**
- Description - Freitextsuche, zu verwenden mit **substring(...)**
- ScType - Id einer Studienstufe (Bachelor/Master/...) (mit *eq*)

- Points - Eine Möglichkeit auf ein Intervall für Credits zu filtern (mit *le*, *ge*, *eq*, usw)
- OrgId - Id einer Fakultät (mit *eq*)
- OfferingOrgId - Id einer Organisationseinheit (mit *eq*)

## Sortierungen

- \$orderby=SmStext desc - Sortierung nach Titel
- \$orderby=SmShort asc - Sortierung nach Kürzel
- \$orderby=CategoryText asc - Sortierung nach Kategorie
- \$orderby=OrgText asc - Sortierung nach Fakultät
- \$orderby=Points asc - Sortierung nach ECTS

## Suche nach Veranstaltungen

*ESearchSet?\$skip=0&\$top=100&\$orderby=EStext asc&\$filter=OrgId eq '50000008' and PiqYear eq '2016' and PiqSession eq '003'&\$expand=Persons&\$inlinecount=allpages*

Hier die Veranstaltungen für MNF noch zum Schluss.

## Filterkriterien

TBD

## Sortierungen

- \$orderby=VstNr asc - Sortierung nach Veranstaltungsnummer
- \$orderby=EStext asc - Sortierung nach Titel
- \$orderby=EShort asc - Sortierung nach Kürzel
- \$orderby=CategoryText asc - Sortierung nach Kategorie

- \$orderby=ScheduleSummary asc - Sortierung nach Termine

## Allgemeines

Bei diesen Suchanfragen kann natürlich jeweils PiqYear und PiqSession weggelassen werden, dann wir das Standardsemester gelesen.

## Details

### Studienprogramme

```
CgDetailsSet(CgObjId='50734856',PiqYear='2016',PiqSession='003')?
$expand=Organizations,Head,Coordination,Scs,Cgs
```

Das liefert die Details für ein Studienprogramm. Variabel sind die Werte zwischen ( und ), es sind die Schlüsselattribute. Das gilt auch für die anderen Abfragen.

## Module

```
SmDetailsSet(SmObjId='50410843',PiqYear='2016',PiqSession='003')?
$expand=Partof,Organizations,Responsible,Events,Events/Persons,OfferF
```

Das liefert die Details für ein Modul.

## Veranstaltungen

```
EDetailsSet(EObjId='50792095',PiqYear='2016',PiqSession='003')?
$expand=Rooms,Persons,Schedule,Schedule/Rooms,Schedule/Persons,M
```

Das liefert die Details für eine Veranstaltung.

## Personen

*PDetailsSet(Objid='01060015',PiqYear='2016',PiqSession='003')?  
\$expand=Events,Events/Persons,Responsible*

Das liefert die Details für eine Person.

## Allgemeines für Detailabrufe

PiqYear und PiqSession sind (primäre) Schlüsselprädikaten und *müssen* immer gefüllt sein. Das ist leider eine technische Beschränkung, um konsistent zu bleiben. Wenn es wirklich unbedingt ohne Semester gehen muss, ist es aber möglich hier noch Dummy-Werte nachzurüsten. Ratsam und konform ist es allerdings nicht unbedingt.