
pynami

Release 0.3.3

May 14, 2023

Contents:

1	Installation	1
2	Usage	3
2.1	Authentication	3
2.2	Access default values	3
2.3	Get a specific mitglied	4
2.4	Search for a group of members	4
2.5	Send emails	4
2.6	Export data	4
3	Main class	7
4	Constants	17
4.1	Bundesländer	17
4.2	Countries	18
4.3	Nationalities	19
4.4	Payment details	24
4.5	Genders	26
4.6	Confessions	26
4.7	Member types and states	26
4.8	DPSG Bausteine	28
4.9	Subdivision	29
4.10	Activity types	31
4.11	Levels	33
5	REST URLs	35
5.1	Authentication	35
5.2	Search members	35
5.3	Update members	37
5.4	Activities	39
5.5	Default values	41
5.6	Dashboard	41
6	Search Mitglieder	45
6.1	Search Keys	46
7	API reference	49

7.1	pynami package	49
8	Changelog	97
8.1	Version 0.3.3 (14.05.2023)	97
9	Indices and tables	99
	Python Module Index	101
	HTTP Routing Table	103
	Index	105

CHAPTER 1

Installation

For this package to run you need at least Python 3.6. All other dependencies are installed automatically.

Just download and extract the repository or simply clone it, navigate to its top directory (where `setup.py` is located) and run

```
pip install [-e] .
```

Use the `-e` option if you want to edit the source files afterwards.

The package should be quite comfortable to use without loosing flexibility.

2.1 Authentication

Listing 1: Make use of the `with` statement for automatic login and logout

```
from pynami.nami import NaMi

with NaMi(username='MITGLIEDSNUMMER', password='PASSWORD') as nami:
    pass
```

Listing 2: Authenticate manually

```
from pynami.nami import NaMi

nami = NaMi()
nami.auth(username='MITGLIEDSNUMMER', password='PASSWORD')
# do stuff
nami.logout()
```

2.2 Access default values

Listing 3: Get gender names and their internal ids

```
from pynami.tools import tabulate2x

print(tabulate2x(nami.geschlechter()))
```

2.3 Get a specific mitglied

Listing 4: Get a mitglied and view its Mitgliedsnummer

```
id_ = nami.search(vorname='Max', nachname='Mustermann')[0].id
mgl = nami.mitglied(id_)
print(mgl.mitgliedsnummer)
```

2.4 Search for a group of members

Listing 5: Search for all Jungpfadfinder and Pfadfinder

```
search = {
    'mglStatusId': 'AKTIV',
    'mglTypeId': 'MITGLIED',
    'untergliederungId': [2, 3],
    'taetigkeitId': 1
}
result = nami.search(**search)
print(tabulate2x(result))
```

2.5 Send emails

Listing 6: Send emails to the members of the search result

```
from pynami.tools import send_emails

send_emails(result)
```

2.6 Export data

There are two functions for exporting data. The first is to export the data as a CSV (Comma-separated values) formatted string and save that as a file using the method `make_csv()`.

Listing 7: Export basic member data to a csv file

```
from pynami.tools import make_csv

keys = ['mitgliedsnummer', 'vorname', 'nachname', 'geschlecht', 'geburtsdatum']
with open('data.csv', 'w', newline='', encoding='utf-8-sig') as f:
    f.writelines(make_csv(result, keys, delimiter=';'))
```

Another option is to export the data as an Excel file using the function `export_xlsx()` which comes already with a possible write to file functionality.

Listing 8: Export basic member data to an Excel file

```
from pynami.tools import export_xlsx  
export_xlsx(result, keys, write_to_file=True, filepath='data.xlsx')
```


CHAPTER 3

Main class

class NaMi (*config*={}, ***kwargs*)

Main class for communication with the DPSG (Deutsche Pfadfinderschaft Sankt Georg) NAMI (Namentliche Mitgliederverwaltung)

Example

Listing 1: Connect to the NAMI, search for all active members and print them in a tabulated form.

```
from pynami.nami import NaMi
from pynami.tools import tabulate2x

with NaMi(username='MITGLIEDSNUMMER', password='PASSWORD') as nami:
    print(tabulate2x(nami.search()))
```

Parameters *config* (*dict*, optional) – Authorization configuration

_check_response (*response*)

Check a requests response object if the NAMI response looks ok. This currently checks some very basic things.

Raises

- *NamiHTTPError* – When HTTP (Hypertext Transfer Protocol) communication failes
- *NamiResponseSuccessError* – When the NAMI returns an error

_get_baseadmin (*key*, *grpId=None*, *mgldId=None*, *taetigkeitId=None*, ***kwargs*)

Base function for retrieving all core lists from the NAMI

Parameters

- **key** (*str*) – Name of the wanted items

- **grpId** (*int* or *str*, optional) – Group id
- **mgId** (*int* or *str*, optional) – Member id (not the DPSG Mitgliedsnummer)
- **taetigkeitId** (*int*, optional) – Id of an activity. A list of all possible ids can be found in the section *Activity types*. This is only required for the URLs which needs to be formatted with this value.

Keyword Arguments

- **gruppierung** (*int* or *str*, optional) – Group id, in case this differs from the group id in the URL (Uniform Resource Locator) (given by `grpId`)
- **mitglied** (*int* or *str*, optional) – Member id (not the DPSG Mitgliedsnummer). This overwrites `mgId`.

Returns The returned default values

Return type *list* of *Baseadmin*

activities

Functional activities

Type *list* of *Baseadmin*

auth (*username=None, password=None*)

Authenticate against the NAMI API (Application Programming Interface). This stores the `jsessionId` cookie in the requests session. Therefore this needs to be called only once.

This also stores your id (not the Mitgliedsnummer) for later pruposes.

Parameters

- **username** (*str*, optional) – The NAMI username. Which is your Mitgliedsnummer
- **password** (*str*, optional) – Your NAMI password

Returns The requests session, including the auth cookie

Return type *requests.Session*

bausteine

Choose DPSG training parts

Type *list* of *Baseadmin*

beitragsarten (*grpId=None, mgId=None*)

Get default values

Parameters

- **grpId** (*int* or *str*, optional) – Group id
- **mgId** (*int* or *str*, optional) – Member id (not the DPSG Mitgliedsnummer)

Returns Each possible fee type

Return type *list* of *Baseadmin*

beitragsarten_mgl (*grpId=None, mgId=None*)

Get default values

Parameters

- **grpId** (*int* or *str*, optional) – Group id
- **mgId** (*int* or *str*, optional) – Member id (not the DPSG Mitgliedsnummer)

Returns Each possible fee for a member

Return type `list` of `Baseadmin`

bescheinigungen (***kwargs*)

Get all certificates of inspection

Returns A list of all your certificates of inspection

Return type `list` of `SearchBescheinigung`

countries (*grpId=None, mglId=None*)

Get default values

Parameters

- **grpId** (*int* or *str*, optional) – Group id
- **mglId** (*int* or *str*, optional) – Member id (not the DPSG Mitgliedsnummer)

Returns List of all possible countries with their names and ids

Return type `list` of `Baseadmin`

download_beantragung (***kwargs*)

Open the application form for a certificate of good conduct as a PDF (Portable Document Format) file.

Parameters ****kwargs** – See `open_download_pdf()`.

download_bescheinigung (*id_, **kwargs*)

Open a certificate as a PDF file

Parameters

- **id** (*int*) – Internal id of the certificate
- ****kwargs** – See `open_download_pdf()`.

download_invoice (*id_, **kwargs*)

Downloads and opens an invoice as a pdf document.

Parameters

- **id** (*int*) – Id of the invoice (not the regular invoice number)
- ****kwargs** – See `open_download_pdf()`.

ebene1

You can choose a Diözese which you belong to.

Type `list` of `Baseadmin`

ebene2 (*ebene1*)

You can choose a Bezirk which you belong to.

Parameters **ebene1** (*int*) – Group id of a Diözese

Returns List of possible Bezirken you are associated with

Return type `list` of `Baseadmin`

ebene3 (*ebene2*)

You can choose a Stamm which you belong to.

Parameters **ebene2** (*int*) – Group id of a Bezirk

Returns List of possible Stämmen you are associated with

Return type `list` of `Baseadmin`

ebenen

Structural layers in the DPSG

Type `list` of `Baseadmin`

geschlechter (*grpId=None, mglId=None*)

Get default values

Parameters

- **grpId** (`int` or `str`, optional) – Group id
- **mglId** (`int` or `str`, optional) – Member id (not the DPSG Mitgliedsnummer)

Returns You can choose between three genders: male, female and diverse.

Return type `list` of `Baseadmin`

get_activity (*mgl, id_*)

Get an activity by its id.

Parameters

- **mgl** (`int`) – Member id (not DPSG Mitgliedsnummer)
- **id** (`int`) – Id of the activity. This will probably originate from an activity search result, e.g. by calling `mgl_activities()`.

Returns The Activity object containing all details.

Return type `Activity`

get_ausbildung (*mglId, id_*)

Get a training by its id.

Parameters

- **mglId** (`int`) – Member id (not DPSG Mitgliedsnummer)
- **id** (`int`) – Id of the training. This will probably originate from an training search result, e.g. by calling `mgl_ausbildungen()`.

Returns The Ausbildung object containing all details about the training.

Return type `Ausbildung`

get_bescheinigung (*id_*)

View a certificate of inspection by its id

Parameters **id** (`int`) – The internal id of the certificate

Returns An object holding all important details about the inspection

Return type `Bescheinigung`

get_mgl_history (*mglId, id_, ext=True*)

Get a member history entry by its id.

Parameters

- **mglId** (`int`) – Member id (not DPSG Mitgliedsnummer)
- **id** (`int`) – Id of the entry. May originate from a search result, e.g. by calling `mgl_history()`.
- **ext** (`bool`, optional) – If the extended history format should be used. Defaults to `True`.

Returns The object containing all vital information about this history entry.

Return type *MitgliedHistory*

get_tag (*mgld*, *tagId*)

Get a tag by its id

Parameters

- **mgld** (*int*) – Member id (not DPSG Mitgliedsnummer)
- **tagId** (*int*) – Tag id

Returns The tag object with all important details

Return type *Tag*

grpId

Group id of the user

Returns *int*

grpadmin_grps

Choose from a list of groups that you are associated with for group admin

Type *list* of *Baseadmin*

gruppierungen

Choose from a list of groups that you are associated with for member admin

Type *list* of *Baseadmin*

history (***kwargs*)

Dashboard function

Returns Last editing events like updating and creating members. In the NAMI these are displayed in the dashboard.

Return type *list* of *HistoryEntry*

invoice (*groupId*, *invId*)

Get an invoice by its id.

Parameters

- **groupId** (*int*) – Group id
- **invId** (*int*) – Id of the invoice. This will probably originate from a search result, e.g. by calling *invoices()*.

Returns The Invoice object containing all details.

Return type *Invoice*

invoices (*groupId=None*, ***kwargs*)

List of all invoices of a group

Parameters **groupId** (*int*, optional) – Group id

Returns All invoices of the specified group

Return type *list* of *SearchInvoice*

konfessionen (*grpId=None*, *mgld=None*)

Get default values

Parameters

- **grpId** (*int* or *str*, optional) – Group id

- **mgldId** (*int* or *str*, optional) – Member id (not the DPSG Mitgliedsnummer)

Returns All denominations

Return type *list* of *Baseadmin*

logout ()

This should be called at the end of the communication. It is called when exiting through the `__exit__()` method.

mgld_activities (*mgld*)

List of all activities of a member

Parameters **mgld** (*int*) – Member id (not DPSG Mitgliedsnummer)

Returns All activities of the member (even those which have already ended)

Return type *list* of *Activity*

mgld_ausbildungen (*mgld*)

Get all trainings from a Mitglied.

Parameters **mgldId** (*int*) – Member id (not DPSG Mitgliedsnummer)

Returns All trainings of the member

Return type *list* of *SearchAusbildung*

mgld_history (*mgld*, *ext=True*)

Get all history changes from a Mitglied.

Parameters

- **mgldId** (*int*) – Member id (not DPSG Mitgliedsnummer)
- **ext** (*bool*, optional) – If the extended history format should be used. Defaults to *True*.

Returns All history entries of the member

Return type *list* of *HistoryEntry*

mgltypes (*grpId=None*, *mgld=None*)

Get default values

Parameters

- **grpId** (*int* or *str*, optional) – Group id
- **mgldId** (*int* or *str*, optional) – Member id (not the DPSG Mitgliedsnummer)

Returns You can have one of three different member types

Return type *list* of *Baseadmin*

mitglied (*mgld=None*, *method='GET'*, *grpId=None*, ***kwargs*)

Gets or updates a Mitglied.

The keyword arguments are passed on to the HTTP communication

Parameters

- **mgldId** (*int*, optional) – ID of the Mitglied. This is not the DPSG Mitgliedsnummer. Defaults to the user.
- **method** (*str*) – HTTP Method. Should be 'GET' or 'PUT', defaults to 'GET'.
- **grpId** (*int*, optional) – The DPSG Stammesnummer, e.g. 131913. The default (*None*) takes the value from the internal attribute `__config`.

Returns The retrieved or respectively updated Mitglied. Note that the *austrittsDatum* attribute is not part of the returned data set.

Return type *Mitglied*

myId

NAMI internal id of the user

Returns int

notifications (*sortproperty=None, sortdirection='ASC', **kwargs*)

Dashboard function

Returns All current notifications (like tier changes of members). In the NAMI these are displayed in the dashboard.

Return type list of *Notification*

regionen (*grpId=None, mglId=None*)

Get default values

Parameters

- **grpId** (*int* or *str*, optional) – Group id
- **mglId** (*int* or *str*, optional) – Member id (not the DPSG Mitgliedsnummer)

Returns All possible Bundeslaender

Return type list of *Baseadmin*

search (***kwargs*)

Run a search for members

Todo:

- Check search terms and formatting. Also some search keys can only be used mutually exclusive.
-

Parameters ****kwargs** – Search keys and words. Be advised that some search words must have a certain formatting or can only take a limited amount of values.

Returns The search results

Return type list of *SearchMitglied*

See also:

SearchSchema for a complete list of search keys

search_all (*grpId=None, filterString=None, searchString=", sortproperty=None, sortdirection='ASC', **kwargs*)

Search function for filtering the whole member list with limited filter options.

It is also possible to sort the results.

Parameters

- **filterString** (*str*, optional) – Filter attribute.
- **searchString** (*str*, optional) – You can search within the chosen filter attribute. The Format must match the type of the filter attribute.
- **sortproperty** (*str*, optional) – Attribute by wich the results shall be sorted.

- **sortdirection** (*str*, optional) – Direction of sorting. Can take the values ASC (which is the default) and DESC.

Returns The search results

Return type *list* of *SearchMitglied*

staaten (*grpId=None, mglId=None*)

Get default values

Parameters

- **grpId** (*int* or *str*, optional) – Group id
- **mglId** (*int* or *str*, optional) – Member id (not the DPSG Mitgliedsnummer)

Returns A quite long list of different nationalities

Return type *list* of *Baseadmin*

stats

Contains counts from different tiers. The actual list of *StatCategory* is in the *statsCategories* attribute.

Type *Stats*

status_list

If you are active, inactive or already deleted

Type *list* of *Baseadmin*

subdivision

Which division you are associated with. This one is only used for searching.

Type *list* of *Baseadmin*

tagList

A different list of fee types but with basically the same content. This one is used for searching members.

Deprecated since version 0.3.3: Only returns an empty list and has therefore become useless.

Type *list* of *Baseadmin*

tags (*mglId, **kwargs*)

Get all tags of a member

Parameters **mglId** (*int*) – Member id (not DPSG Mitgliedsnummer)

Returns List of the search results

Return type *list* of *SearchTag*

tk_auf_grp (*grpId, mglId, **kwargs*)

Get all possible activities for a certain group

Parameters

- **grpId** (*int* or *str*) – Group id
- **mglId** (*int* or *str*) – Member id (not the DPSG Mitgliedsnummer)

Returns List of possible activities

Return type *list* of *Baseadmin*

tk_caea_grp (*grpId, mglId, taetigkeitId, **kwargs*)

Get all possible access rights for an activity

Parameters

- **grpId** (*int* or *str*) – Group id
- **mglId** (*int* or *str*) – Member id (not the DPSG Mitgliedsnummer)
- **taetigkeitId** (*int*) – Id of the activity. A list of all possible ids can be found in the section *Activity types*.

Returns List of possible access rights

Return type list of *Baseadmin*

tk_grp (*grpId*, *mglId*, ***kwargs*)

Get all possible groups for an activity

Parameters

- **grpId** (*int* or *str*) – Group id
- **mglId** (*int* or *str*) – Member id (not the DPSG Mitgliedsnummer)

Returns List of possible groups

Return type list of *Baseadmin*

tk_ug (*grpId*, *mglId*, *taetigkeitId*, ***kwargs*)

Get all possible subdivision for an activity

Parameters

- **grpId** (*int* or *str*) – Group id
- **mglId** (*int* or *str*) – Member id (not the DPSG Mitgliedsnummer)
- **taetigkeitId** (*int*) – Id of the activity. A list of all possible ids can be found in the section *Activity types*.

Returns List of possible subdivision

Return type list of *Baseadmin*

update_activity (*mgl*, *act*)

Update an activity

Parameters

- **mgl** (*int*) – Member id (not DPSG Mitgliedsnummer)
- **act** (*Activity*) – Updated data set. The activity id is taken form this data set.

Returns A new updated object

Return type *Activity*

Warning: This has not been tested yet!

update_ausbildung (*mglId*, *ausbildung*)

Update a training

Parameters

- **mgl** (*int*) – Member id (not DPSG Mitgliedsnummer)
- **ausbildung** (*Ausbildung*) – Updated data set. The training id is taken form this data set.

Returns A new updated object

Return type *Ausbildung*

Warning: This has not been tested yet!

zahlungskonditionen (*grpId=None, mglId=None*)

Get default values

Parameters

- **grpId** (*int* or *str*, optional) – Group id
- **mglId** (*int* or *str*, optional) – Member id (not the DPSG Mitgliedsnummer)

Returns Every paying method (either bank transfer or debit)

Return type *list* of *Baseadmin*

Here is an overview about the default values and their ids the NAMi uses.

Warning: These values may change over time (e.g. when a new AG for a annual campaign comes up) Current state: 12.05.2023

4.1 Bundesländer

Table 1: Bundesländer and their ids

Beschreibung	ID
Baden-Württemberg (Deutschland)	1
Bayern (Deutschland)	2
Berlin (Deutschland)	3
Brandenburg (Deutschland)	4
Bremen (Deutschland)	5
Hamburg (Deutschland)	6
Hessen (Deutschland)	7
Mecklenburg-Vorpommern (Deutschland)	8
Niedersachsen (Deutschland)	9
Nordrhein-Westfalen (Deutschland)	10
Rheinland-Pfalz (Deutschland)	11
Saarland (Deutschland)	12
Sachsen (Deutschland)	13
Sachsen Anhalt (Deutschland)	14
Schleswig-Holstein (Deutschland)	15
Thüringen (Deutschland)	16
Nicht-DE (Ausland)	23

See also:

regionen()

4.2 Countries

Table 2: Countries for address information

Beschreibung	ID
Deutschland	1
Algerien	2
Belgien	3
Brasilien	4
Dänemark	5
Finnland	6
Frankreich	7
Grossbritannien	8
Irland	9
Island	10
Israel	11
Italien	12
Japan	13
Kanada	14
Litauen	15
Luxemburg	16
Niederlande	17
Norwegen	18
Österreich	19
Polen	20
Rumänien	21
Schweiz	22
Schweden	23
Slowenien	24
Spanien	25
Südafrika	26
Thailand	27
Ungarn	28
Vatikanstadt	29
USA	30
Indien	31
Polen	32
CV	33
Ausland	34
Malaysia	35
Tunesien	36

See also:

countries()

4.3 Nationalities

Beschreibung	ID
Bosnisch	1284
USA/SLO	1286
serb.montenegro	1287
serbisch	1288
französisch	1289
A	1291
holländisch	1292
Kenia	1293
madegaschi	1294
• amerikanisch	1295
St. Lucianer	1296
lettisch	1297
Iran	1298
japanisch	1299
öster. + südafr.	1304
italiano	1306
Ukrainerin	1308
deutsch	1054
Serbien-Montenegro	1310
Jugoslawisch/Kosovo	1311
	1056
ungarisch	1312
andere	1057
chinesisch	1058
österreichisch	1059
Thailand	1315
italienisch	1060
EU	1061
maltesisch	1317
Portugiesisch	1318
niederländisch	1064
schweizerisch	1320
kenianisch	1065
Tschechische Republik	1321
kroatisch	1066
brasilianisch	1067
weißrussisch	1323
polnisch	1069
Bolivien	1326
türkisch	1071
öster.	1327
Grieche	1329
portugiesisch	1074
Burundi	1331
amerikanisch	1076

Continued on next page

Table 3 – continued from previous page

Beschreibung	ID
Brittisch	1333
österreich	1081
Albanien	1338
Nigeria	1339
GB	1340
mazedonisch	1341
britisch	1086
Togolaise	1343
Schweizer	1088
pakistanisch	1344
Thai	1089
österreichschich/norwegisch	1345
litauisch	1091
Portugal	1347
griechisch	1092
Italienisch-Österr.	1348
schweizerisch	1093
guatemalteisch	1349
belgisch	1094
nicht bekannt	1350
Portugiesisch	1352
lettisch	1102
französisch	1103
spanisch	1104
british	1360
marokkanisch	1361
Togo	1106
albanisch	1107
Niederländer	1364
Sri Lanka	1110
estnisch	1366
Indien	1111
Eritrea	1112
slowakisch	1370
Serbe	1115
Algerisch	1372
indonesisch	1117
griech.	1373
yugoslawisch	1118
englisch/plonisch	1119
irisch	1375
bosnisch-herzegow.	1120
Jugoslawisch	1376
Italienisch	1121
Irish	1122
libanesisch	1123
iranisch	1124
Bulgarien/Belgien	1380
polnisch/italiensch	1381

Continued on next page

Table 3 – continued from previous page

Beschreibung	ID
amerikanisch	1126
salvadorianisch	1127
Russische Föderation	1384
US	1129
Rumänisch	1385
Ungarn	1130
U.S. Amerikaner	1386
SERBIEN	1387
Spanisch	1132
ukrainisch	1133
Belgier	1389
eritreisch	1134
philippinisch	1135
Ghana	1391
jugoslawisch	1137
Amerikaner	1393
tschechisch	1138
kongolesisch	1394
portugisiesch	1395
Bosnien und Herzegowina	1140
Portugisiesch	1396
Türkisch	1141
nigerianisch	1397
bosnisch	1142
Indonesisch	1398
British	1143
griechisch-amerikanisch	1399
portugisisch	1144
CH	1146
Malaysian	1405
Filipina	1150
vietnam	1151
Kroatien	1152
venezuelanisch	1408
Brasilien	1410
niederländer	1155
U.S.Amerikaner	1411
vietnamesisch	1156
ECU	1412
Luxemburgisch	1413
Britisch	1158
Österreich	1159
amerikanisch / polnisch	1415
kanadisch	1160
schweiz	1416
Thailänder	1417
Irakisch	1162
Peru	1418
Italien	1163

Continued on next page

Table 3 – continued from previous page

Beschreibung	ID
Kirgystan	1164
australisch	1420
GB (EU) / USA	1421
Tschad	1166
bolivisch	1422
Bosnien	1167
ital./portug.	1168
frz	1426
Portugisisch	1427
croatisch	1173
Haiti	1429
Kroatisch	1174
Belgisch	1175
russisch	1176
Mexikaner	1432
Kongolese	1433
irakisch	1179
mexikanisch	1435
Französisch	1180
brasilianische	1436
Kanadisch / Niederländisch	1437
rumänisch	1182
moldawisch	1183
US-Amerikaner	1439
Albanisch	1184
niederländiscch	1440
Neiderländisch / Englisch	1185
Ägypten	1441
jugosl.	1186
und spanisch	1442
bolivianisch	1187
Amerikanerin	1443
+italienisch	1444
Brasilien / Österreich	1446
chilenisch	1191
Polen	1447
Serbisch	1192
kazachstan	1193
US-amerikanisch	1194
Kongo	1450
thailändisch	1451
Amerikanisch	1196
und australisch	1452
togoisch	1453
Slowenien	1199
Kurdisch	1455
Russisch	1200
Philippiner	1456
tschetschenisch	1457

Continued on next page

Table 3 – continued from previous page

Beschreibung	ID
Mosambik	1458
Schweiz	1203
Guatemala	1459
Brasilianisch	1460
Kroate	1205
Äthopien	1461
usbekisch	1462
Indisch	1463
USA / GB(EU)	1465
vietnamesisch	1210
Britisch	1466
syrisch	1211
Japanisch	1467
Tai	1468
Türkei	1213
F	1214
Ukraine	1215
Weißrussland	1471
slowenisch	1216
Äthiopisch	1472
luxemburgisch	1218
Rumänien	1219
• Spanisch	1475
Serbien	1220
Kenianer	1476
Griechisch	1221
kroatische	1477
kosovarisch	1222
Litauisch	1479
kasachisch	1225
Südkorea	1481
tunesisch	1482
kamerunesisch	1229
philipino	1233
phillipino	1234
port.	1235
luxemb.	1236
N-amerikanisch	1237
F - CH	1238
NL	1239
theiländisch	1243
Vietnam	1246
Bulgarien	1247
bulgarisch	1250
norwegisch	1251
rwander	1252
ägypt.	1254

Continued on next page

Table 3 – continued from previous page

Beschreibung	ID
B	1255
dänisch	1259
türk.	1261
Syrisch	1262
domenikanisch	1263
tamilisch	1265
äthiopisch	1266
pl	1268
griechisch	1269
Togolesisch	1271
kamerunisch	1272
Polnisch	1273
persisch	1276
Irak	1277
Ukrainisch	1278
Tschechisch	1279

See also:

`staaten()`

4.4 Payment details

4.4.1 Member management

Table 4: Payment methods

Beschreibung	ID
Std Lastschrift	1
Std Überweisung	2

See also:

Here `zahlungskonditionen()`

Table 5: Fee types

Beschreibung	ID
DPSG Bundesverband 000000 (Familienermäßigt - Stiftungseuro - VERBANDSBEITRAG)	5
DPSG Bundesverband 000000 (Familienermäßigt - VERBANDSBEITRAG)	2
DPSG Bundesverband 000000 (Sozialermäßigt - Stiftungseuro - VERBANDSBEITRAG)	6
DPSG Bundesverband 000000 (Sozialermäßigt - VERBANDSBEITRAG)	3
DPSG Bundesverband 000000 (Übernahme - VERBANDSBEITRAG)	7
DPSG Bundesverband 000000 (Voller Beitrag - Stiftungseuro - VERBANDSBEITRAG)	4
DPSG Bundesverband 000000 (Voller Beitrag - VERBANDSBEITRAG)	1

See also:

`beitragsarten_mgl()`

The following table can be obtained with a different function and is given here for the sake of completeness.

Table 6: Fee types (Yes, the same ones)

Beschreibung	ID
Familienermäßigt	2
Übernahme	7
Voller Beitrag	1
Voller Beitrag - Stiftungseuro	4
Familienermäßigt - Stiftungseuro	5
Sozialermäßigt	3
Sozialermäßigt - Stiftungseuro	6

See also:

`beitragsarten()`

See also:

Default values How to get these values

4.4.2 Ids for searching

Deprecated since version 0.3.3: No longer in use. You will just get an empty list.

GET `//tagging/getTagList`

Get default values and their ids.

Query Parameters

- **page** (*int*) – The page number of result display
- **start** (*int*) – Show search results form this index
- **limit** (*int*) – Show only up to this many entries

Response JSON Object

- **success** (*boolean*) – If everything was ok
- **responseType** (*string*) – NAMI response type: Can be OK, INFO, WARN, ERROR, EXCEPTION or null
- **totalEntries** (*int*) – Number of returned entries
- **data** (*list*) – The search results. Each returned tag is a list entry in the form of a json array.

See also:

BaseadminSchema JSON (JavaScript Object Notation) schema of the returned data

Status Codes

- **200 OK** – No error

See also:

`tagList()`

4.5 Genders

Beschreibung	ID
divers	24
keine Angabe	23
männlich	19
weiblich	20

See also:

geschlechter()

4.6 Confessions

Beschreibung	ID
römisch-katholisch	1
evangelisch / protestantisch	2
orthodox	3
freikirchlich	4
andere christliche	5
jüdisch	6
muslimisch	7
sonstige	8
ohne Konfession	9
neuapostolisch	11

See also:

konfessionen()

4.7 Member types and states

4.7.1 Member types

GET `/nami/enum/mgltype`

Get default values and their ids.

Query Parameters

- **gruppierung** (*int*) – A chosen group id (usually your Stammesnummer) (optional)
- **mitglied** (*int*) – Member id (not the DPSG Mitgliedsnummer) (optional)
- **page** (*int*) – The page number of result display
- **start** (*int*) – Show search results form this index
- **limit** (*int*) – Show only up to this many entries

Response JSON Object

- **success** (*boolean*) – If everything was ok

- **responseType** (*string*) – NAMI response type: Can be OK, INFO, WARN, ERROR, EXCEPTION or null
- **totalEntries** (*int*) – Number of returned entries
- **data** (*list*) – The search results. Each returned member type is a list entry in the form of a json array.

Table 7: Latest search results

Beschreibung	ID
Nicht-Mitglied	NICHT_MITGLIED
Schnuppermitglied	SCHNUPPER_MITGLIED
Mitglied	MITGLIED

See also:

mgltypes()

See also:

BaseadminSchema JSON schema of the returned data

Status Codes

- 200 OK – No error

4.7.2 Member states

GET /nami/search-multi/status-list

Get default values and their ids.

Query Parameters

- **page** (*int*) – The page number of result display
- **start** (*int*) – Show search results form this index
- **limit** (*int*) – Show only up to this many entries

Response JSON Object

- **success** (*boolean*) – If everything was ok
- **responseType** (*string*) – NAMI response type: Can be OK, INFO, WARN, ERROR, EXCEPTION or null
- **totalEntries** (*int*) – Number of returned entries
- **data** (*list*) – The search results. Each returned member state is a list entry in the form of a json array.

Table 8: Latest search results

Beschreibung	ID
Aktiv	AKTIV
Inaktiv	INAKTIV
archiviert	GELOESCHT_ALT

See also:

`status_list`

See also:

`BaseadminSchema` JSON schema of the returned data

Status Codes

- **200 OK** – No error

4.8 DPSG Bausteine

GET `/module/baustein`

Get default values and their ids.

Query Parameters

- **gruppierung** (*int*) – A chosen group id (usually your Stammesnummer) (optional)
- **mitglied** (*int*) – Member id (not the DPSG Mitgliedsnummer) (optional)
- **page** (*int*) – The page number of result display
- **start** (*int*) – Show search results form this index
- **limit** (*int*) – Show only up to this many entries

Response JSON Object

- **success** (*boolean*) – If everything was ok
- **responseType** (*string*) – NAMI response type: Can be OK, INFO, WARN, ERROR, EXCEPTION or null
- **totalEntries** (*int*) – Number of returned entries
- **data** (*list*) – The search results. Each returned baustein is a list entry in the form of a json array.

Table 9: Baustein tags (latest search results)

Beschreibung	ID
Schritt 1 - Motivation und Grundlagen zum Leiten, Ausbildung in der DPSG	21
Schritt 2 - Gestaltung und Organisation von Gruppenstunden	22
Baustein 1a - Identität und Führungsstil	3
Baustein 1b - Teamarbeit	4
Baustein 1c - Gesellschaftliches Engagement	5
Baustein 1d (Bis 2016 3d) - Spiritualität	14
Baustein 2a - Lebenswelt von Kindern und Jugendlichen, Pädagogik der DPSG	6
Baustein 2b - Mädchen und Jungen, Geschlechtsbewusste Gruppenarbeit	7
Baustein 2c - Pfadfinderische Grundlagen: Pfadfinderische Methodik	8
Baustein 2d - Gewalt gegen Kinder und Jugendliche: Sensibilisierung, Intervention	9
Baustein 2e - Gewalt gegen Kinder und Jugendliche: Vertiefung, Prävention	10
Baustein 3a - Pfadfinderische Grundlagen: Geschichte und Hintergründe	11
Baustein 3b - Erste Hilfe	12
Baustein 3c - Finanzen, Haftung und Versicherung	13
Baustein 3e - Pfadfindertechniken	1
Baustein 3f - Planung und Durchführung von Maßnahmen	2

Continued on next page

Table 9 – continued from previous page

Beschreibung	ID
Abgeschlossene Modulausbildung - oder Woodbadge-Kurs I	25
Woodbadge-Kurs oder Woodbadge-Kurs II - Wölflinge	17
Woodbadge-Kurs oder Woodbadge-Kurs II - Jungpfadfinder	18
Woodbadge-Kurs oder Woodbadge-Kurs II - Pfadfinder	19
Woodbadge-Kurs oder Woodbadge-Kurs II - Rover	20
Woodbadge-Kurs oder Woodbadge-Kurs II - Vorstände	23
Modulleitungstraining (MLT) - Modulleitungstraining (MLT)	26
Teamer Training I (TT I) - Wölflinge	27
Teamer Training I (TT I) - Jungpfadfinder	29
Teamer Training I (TT I) - Pfadfinder	30
Teamer Training I (TT I) - Rover	31
Teamer Training I (TT I) - Vorstände	32
Teamer Training II (TT II) - Wölflinge	33
Teamer Training II (TT II) - Jungpfadfinder	28
Teamer Training II (TT II) - Pfadfinder	34
Teamer Training II (TT II) - Rover	35
Teamer Training II (TT II) - Vorstände	36
Assistant Leadertrainer Training (ALT) - Assistant Leadertrainer Training (ALT)	24
Kuratenausbildung - Ausbildung der Kuratinnen und Kuraten	37
Sonstiges - Kurse ausserhalb der Woodbadge-Ausbildung (Kein Baustein!)	38

See also:

bausteine

See also:

BaseadminSchema JSON schema of the returned data

- **metaData** (*json*) – Additional information about the data fields but not on the data itself.

Status Codes

- **200 OK** – No error

4.9 Subdivision

GET /orgadmin/untergliederung

Get default values and their ids.

Query Parameters

- **gruppierung** (*int*) – A chosen group id (usually your Stammesnummer)
- **mitglied** (*int*) – Member id (not the DPSG Mitgliedsnummer)
- **page** (*int*) – The page number of result display
- **start** (*int*) – Show search results form this index
- **limit** (*int*) – Show only up to this many entries

Response JSON Object

- **success** (*boolean*) – If everything was ok

- **responseType** (*string*) – NAMI response type: Can be OK, INFO, WARN, ERROR, EXCEPTION or null
- **totalEntries** (*int*) – Number of returned entries
- **data** (*list*) – The search results. Each returned subdivision is a list entry in the form of a json array.

Table 10: Subdivision types for searching (latest search results)

Beschreibung	ID
AG Ausbildung	24
AG Ausbildung Pfadfinderstufe	75
AG Digitale Infrastruktur und Tools	74
AG DPSG App	50
AG Freiwilligendienst	51
AG Friedenslicht	52
AG Geschlechtergerechtigkeit	53
AG Jahresaktion 2018 - Lebendig. Kraftvoll. Schärfer. Glaubste?	55
AG Jahresaktion 2019 - vollKostbar	54
AG Jahresaktion 2020 - No waste - ohne Wenn und Abfall	69
AG Jahresaktion 2021 - Pfadfinderinnen/Pfadfinder sind MITTENDrin	70
AG Jahresaktion 2022 - Pfadfinden ist bunt	73
AG Jahresaktion 2023 - Unter Strom	77
AG Jahresaktion 2024 - Antirassismus	81
AG Kompass	79
AG NaMi Community Management NCM	57
AG Netzwerk DPSG	56
AG Öffentlichkeitsarbeit	22
AG Ordnung	62
AG Pfingsten in Westernohe PIW	58
AG Politische Bildung	59
AG Satzungsfragen	31
AG Spiritualität	42
AG Stärkung Strukturschwacher Regionen	76
AG Transparente Finanzen	60
AG WBK-Rahmenkonzept	61
Archivbeirat	64
Aufarbeitung - Beirat	80
BDKJ	17
Biber	49
Bundesamt	38
Bundesebene	39
Campstaff PIW	72
Elternbeirat	14
Entwicklungsfragen	6
Freunde und Förderer	35
Hauptausschuss	29
Inklusion	8
Interkulturelles Lernen	7
Internationale Gerechtigkeit	41
Internationales	21
Jugendhilfeausschuss	20

Continued on next page

Table 10 – continued from previous page

Beschreibung	ID
Jugendring	18
Junge Delegierte rdp	78
Jungpfadfinder	2
Kuratorium Westernohe	65
Landesstelle	37
MoViS - Beauftragte/Beauftragter für ehrenamtliches Engagement	71
Ökologie	26
Pfadfinder	3
PGR	19
RdP	16
Rechnungsprüfungsausschuss RPA	66
Rechtsträger	15
Rover	4
SG internationale Ausbildung	68
SG Nahost	67
Sonstige	48
Sonstige Mitarbeitende (mit Versicherungsschutz)	13
Sonstige Mitarbeitende (ohne Versicherungsschutz)	47
StG Deutsch-Französisch	40
Stiftung	36
Vorstand	5
Wahlausschuss	30
Westernohe	34
Wölfling	1

See also:

BaseadminSchema JSON schema of the returned data

See also:

subdivision

- **metaData** (*json*) – Additional information about the data fields but not on the activities themselves.

Status Codes

- **200 OK** – No error

4.10 Activity types

GET `/nami/search-multi/all-visible-taetigkeiten`

Get default values and their ids.

Query Parameters

- **gruppierung** (*int*) – A chosen group id (usually your Stammesnummer)
- **mitglied** (*int*) – Member id (not the DPSG Mitgliedsnummer)
- **page** (*int*) – The page number of result display
- **start** (*int*) – Show search results from this index

- **limit** (*int*) – Show only up to this many entries

Response JSON Object

- **success** (*boolean*) – If everything was ok
- **responseType** (*string*) – NAMI response type: Can be OK, INFO, WARN, ERROR, EXCEPTION or null
- **totalEntries** (*int*) – Number of returned entries
- **data** (*list*) – The search results. Each returned activity is a list entry in the form of a json array.

Table 11: All activity types for searching

Beschreibung	ID
Mitgliederservice	42
Erstellen von SGBVIII Bescheinigungen	43
Datenschutzerklärung EAs abgegeben	45
Präventionsschulung EAs	46
soz. erm. Beiträge zuweisen	47
€ Helfer AK/AG/StG/SG	48
€ Mitglied	1
x € AdministratorIn	14
€ Mitglied AK/AG/StG/SG	18
€ passive Mitgliedschaft	39
Beitragsabrechnung	36
€ BeobachterIn	8
€ Delegierte(r)	7
– ElternvertreterIn	4
EmpfängerIn Freiexemplar	37
Empfänger Freiexemplare	28
€ Ersatzdelegierte(r)	31
€ GeschäftsführerIn	19
– hauptberufliche Mitarbeiter	22
€ KassenprüferIn	21
€ KassiererIn	20
KontakterIn	32
€ KuratIn	11
€ LeiterIn	6
€ Leitungsteam-SprecherIn	5
€ MaterialwartIn	23
sonst. MitarbeiterIn	16
Ref. Ersatzdelegierte(r)	30
Ref.-Delegierte(r)	25
€ ReferentIn	10
€ SprecherIn	2
stellv. Vorsitzende(r)	26
stellvertr. Mitglied	34
Stufendelegierte(r)	33

Continued on next page

Table 11 – continued from previous page

Beschreibung	ID
Versandanschrift	29
– VertreterIn (BDKJ/RDP etc.)	24
€ Vorsitzende(r)	13
€ sonst. Mitglied	40
– sonst. ext. MitarbeiterIn	41
Stammessuche	44
Schnuppermitgliedschaft	35

See also:

activities

See also:

BaseadminSchema JSON schema of the returned data

- **metaData** (*json*) – Additional information about the data fields but not on the activities themselves.

Status Codes

- 200 OK – No error

4.11 Levels

Table 12: DPSG levels for searching

Beschreibung	ID
Bundesebene	1
Diözese	2
Bezirk	3
Stamm	4
Siedlung	6

See also:

ebenen

The default server URL for the NAMI is

<https://nami.dpsg.de/ica/rest>

5.1 Authentication

POST /nami/auth/manual/sessionStartup

Login

Query Parameters

- **username** (*string*) – The DPSG member id
- **password** (*string*) – Your password

GET /nami/auth/logout

Logout. Ends the current session.

5.2 Search members

GET /nami/mitglied/filtered-for-navigation/gruppierung/gruppierung/ (*groupId*) /
flist

List all group members though filtering by one value is possible

Parameters

- **groupId** (*int*) – A chosen group id (usually your Stammesnummer)

Query Parameters

- **filterString** (*string*) – The filter attribute (optional)
- **searchString** (*string*) – Search value. If the filter attribute refers to a date the format is YYYY-mm-dd HH:MM:SS.

- **page** (*int*) – The page number of result display
- **start** (*int*) – Show search results form this index
- **limit** (*int*) – Show only up to this many entries

Response JSON Object

- **success** (*boolean*) – If everything was ok
- **responseType** (*string*) – NAMI response type: Can be OK, INFO, WARN, ERROR, EXCEPTION or null
- **totalEntries** (*int*) – Number of returned members
- **data** (*list*) – The search results. Each returned member is a list entry in the form of a json array.

See also:

[*SearchMitgliedSchema*](#) JSON schema of the returned members

Status Codes

- 200 OK – No error

GET /nami/search-multi/result-list

Search members by multiple keys

Query Parameters

- **searchedValues** (*string*) – A json formatted string of all search values
- **page** (*int*) – The page number of result display
- **start** (*int*) – Show search results form this index
- **limit** (*int*) – Show only up to this many entries

Response JSON Object

- **success** (*boolean*) – If everything was ok
- **responseType** (*string*) – NAMI response type: Can be OK, INFO, WARN, ERROR, EXCEPTION or null
- **totalEntries** (*int*) – Number of returned members
- **data** (*list*) – The search results. Each returned member is a list entry in the form of a json array.

See also:

[*SearchMitgliedSchema*](#) JSON schema of the returned members

Status Codes

- 200 OK – No error

GET /nami/mitglied/filtered-for-navigation/gruppierung/gruppierung/{groupId}/{mgId}

Get a member by its id (not the DPSG Mitgliedsnummer)

Parameters

- **groupId** (*int*) – A chosen group id (usually your Stammesnummer)
- **mgId** (*int*) – Member id (not the DPSG Mitgliedsnummer)

Response JSON Object

- **success** (*boolean*) – If everything was ok
- **responseType** (*string*) – NAMI response type: Can be OK, INFO, WARN, ERROR, EXCEPTION or null
- **message** (*string*) – Should be null.
- **title** (*string*) – Should be null.
- **data** (*json*) – Contains all relevant information about the member

See also:

MitgliedSchema JSON schema of the returned member

Status Codes

- 200 OK – No error

5.3 Update members

PUT /nami/mitglied/filtered-for-navigation/gruppierung/gruppierung/{groupId}/{mglId}
 Update information about a member.

Parameters

- **groupId** (*int*) – A chosen group id (usually your Stammesnummer)
- **mglId** (*int*) – Member id (not the DPSG Mitgliedsnummer)

JSON Parameters

- **beitragsArt** (*string*) – Fee type
- **beitragsArtId** (*id*) – Id of the fee type
- **eintrittsdatum** – Begin of association (Format: YYYY-mm-dd HH:MM:SS)
- **email** (*string*) – Primary email address
- **emailVertretungsberechtigter** (*string*) – Email address of an authorized representative.
- **ersteTaetigkeit** (*string*) – First activity. Defaults to null.
- **ersteUntergliederung** (*string*) – First tier.
- **fixBeitrag** (*string*) – Defaults to null.
- **geburtsDatum** (*string*) – Birth date (Format: YYYY-mm-dd HH:MM:SS)
- **genericField1** (*string*) – Not sure why these even exist.
- **genericField1** – Not sure why these even exist.
- **geschlecht** (*string*) – Gender
- **geschlechtId** (*int*) – Corresponding id to the gender
- **gruppierung** (*string*) – Group name including its id
- **gruppierungId** (*string*) – Group id as a string
- **id** (*int*) – Member id

- **jungpfadfinder** (*string*) –
- **konfession** (*string*) – Confession
- **confessionId** (*int*) – Id corresponding to the confession
- **kontoverbindung** (*string*) – JSON formatted string of the payment details
- **land** (*string*) – Country the member lives in
- **landId** (*int*) – Id corresponding to the address country
- **mglType** (*string*) – Member type
- **nachname** (*string*) – Surname
- **nameZusatz** (*string*) – Additional name
- **ort** (*string*) – Address city
- **pfadfinder** (*string*) –
- **plz** (*string*) – Postal code
- **region** (*string*) – Name of the address state
- **regionId** (*int*) – Id of the address state
- **rover** (*string*) –
- **sonst01** (*boolean*) – Defaults to false.
- **sonst02** (*boolean*) – Defaults to false.
- **spitzname** (*string*) – Nickname
- **staatsangehoerigkeit** (*string*) – Citizenship
- **staatsangehoerigkeitId** (*string*) – Id of the citizenship
- **status** (*string*) – If the member is active or inactive
- **strasse** (*string*) – Address street
- **stufe** (*string*) – Current tier of the member
- **telefax** (*string*) – Fax number
- **telefon1** (*string*) – First telephone number
- **telefon2** (*string*) – Second telephone number
- **telefon3** (*string*) – Third telephone number
- **version** (*int*) – History version number
- **vorname** (*string*) – First name
- **wiederverwendenFlag** (*boolean*) – If the member data may be used after the membership ends
- **woelfing** (*string*) –
- **zeitschriftenversand** (*boolean*) – If the member gets the DPSG newspaper.

Response JSON Object

- **success** (*boolean*) – If everything was ok
- **responseType** (*string*) – NAMI response type: Can be OK, INFO, WARN, ERROR, EXCEPTION or null

- **message** (*string*) – Should be Update successful.
- **title** (*string*) – Should be null.
- **data** (*json*) – Contains all information about the updated member

See also:

[*MitgliedSchema*](#) JSON schema of the returned member

Status Codes

- 200 OK – No error

5.4 Activities

GET /nami/zugeordnete-taetigkeiten/filtered-for-navigation/gruppierung-mitglied/mitglied/{n
Retrieve all activities of a member. There is also an option for filtering the results.

Parameters

- **mgliId** (*int*) – Member id (not the DPSG Mitgliedsnummer)

Query Parameters

- **page** (*int*) – The page number of result display
- **start** (*int*) – Show search results form this index
- **limit** (*int*) – Show only up to this many entries
- **filterString** (*string*) – The filter attribute (optional)
- **searchString** (*string*) – Search value. If the filter attribute refers to a date the format is YYYY-mm-dd HH:MM:SS.

Response JSON Object

- **success** (*boolean*) – If everything was ok
- **responseType** (*string*) – NAMI response type: Can be OK, INFO, WARN, ERROR, EXCEPTION or null
- **totalEntries** (*int*) – Number of returned members
- **data** (*list*) – The search results. Each returned activity is a list entry in the form of a json array.

See also:

[*SearchActivitySchema*](#) JSON schema of the returned activity

- **metaData** (*json*) – Additional information about the data fields but not on the activities themselves.

Status Codes

- 200 OK – No error

GET /nami/zugeordnete-taetigkeiten/filtered-for-navigation/gruppierung-mitglied/mitglied/{n
Get a single activity by its id.

Parameters

- **mgliId** (*int*) – Member id (not the DPSG Mitgliedsnummer)
- **actId** (*int*) – Id of the desired activity

Response JSON Object

- **success** (*boolean*) – If everything was ok
- **responseType** (*string*) – NAMI response type: Can be OK, INFO, WARN, ERROR, EXCEPTION or null
- **message** (*string*) – Should be null.
- **title** (*string*) – Should be null.
- **data** (*json*) – Contains all relevant information about the activity.

See also:

[*ActivitySchema*](#) JSON schema of the returned activity

Status Codes

- 200 OK – No error

PUT /nami/zugeordnete-taetigkeiten/filtered-for-navigation/gruppierung-mitglied/mitglied/{namiId}/taetigkeit/{actId}
Update an activity

Parameters

- **mgliId** (*int*) – Member id (not the DPSG Mitgliedsnummer)
- **actId** (*int*) – Id of the desired activity

JSON Parameters

- **aktivBis** (*string*) – End date of the activity (Format: YYYY-mm-dd HH:MM:SS)
- **aktivVon** (*string*) – Start date of the activity (Format: YYYY-mm-dd HH:MM:SS)
- **beitragsArtId** (*int*) – Id of the fee type (default: null). It has not been observed to be anything else than null.
- **caeaGroup** (*string*) – Access rights for the group.
- **caeaGroupForGf** (*string*) – Access rights for the sub group.
- **caeaGroupId** (*int*) – Corresponding id for caeaGroup.
- **caeaGroupForGfId** (*int*) – Corresponding id for caeaGroupForGf.
- **gruppierung** (*string*) – Group Name including its id.
- **id** (*int*) – Id of the activity.
- **taetigkeit** (*string*) – Type of the activity.
- **untergliederung** (*string*) – Department or tier the activity is associated with.

Response JSON Object

- **success** (*boolean*) – If everything was ok
- **responseType** (*string*) – NAMI response type: Can be OK, INFO, WARN, ERROR, EXCEPTION or null
- **message** (*string*) – Should be Update successful.
- **title** (*string*) – Should be null.

- **data** (*json*) – Contains all information about the updated activity.

See also:

ActivitySchema JSON schema of the returned activity

Status Codes

- 200 OK – No error

5.5 Default values

GET /baseadmin/{key}/

Get default values and their ids from the baseadmin database.

Parameters

- **key** (*string*) – Which type of values you want to get. Possible values are geschlecht, staatsangehoerigkeit, konfession, region, land and zahlungskondition.

Query Parameters

- **gruppierung** (*string*) – Group id
- **mitglied** (*string*) – Member id (not the DPSG Mitgliedsnummer)
- **page** (*int*) – The page number of result display
- **start** (*int*) – Show search results form this index
- **limit** (*int*) – Show only up to this many entries

Response JSON Object

- **success** (*boolean*) – If everything was ok
- **responseType** (*string*) – NAMI response type: Can be OK, INFO, WARN, ERROR, EXCEPTION or null
- **totalEntries** (*int*) – Number of returned members
- **data** (*list*) – The search results. Each returned activity is a list entry in the form of a json array.

See also:

BaseadminSchema JSON schema of the returned data

- **metaData** (*json*) – Additional information about the data fields but not on the data itself.

Status Codes

- 200 OK – No error

5.6 Dashboard

GET /dashboard/notification-entries/flist

Retrieve all recent notifications.

Query Parameters

- **page** (*int*) – The page number of result display
- **start** (*int*) – Show search results form this index
- **limit** (*int*) – Show only up to this many entries
- **filterString** (*string*) – The filter attribute (optional). So far only the value `interval` has been observed.
- **searchString** (*string*) – Search value. If the filter attribute refers to a date the format is `YYYY-mm-dd HH:MM:SS`. For the `filterString interval` use an integer to view the notifications of the specified number of weeks.

Response JSON Object

- **success** (*boolean*) – If everything was ok
- **responseType** (*string*) – NAMI response type: Can be `OK`, `INFO`, `WARN`, `ERROR`, `EXCEPTION` or `null`
- **totalEntries** (*int*) – Number of returned members
- **data** (*list*) – The search results. Each returned notification is a list entry in the form of a json array.

See also:

[*NotificationSchema*](#) JSON schema of the returned notification

- **metaData** (*json*) – Additional information about the data fields but not on the activities themselves.

Status Codes

- `200 OK` – No error

GET /dashboard/history-entries/flist

Retrieve all recent history entries.

Query Parameters

- **page** (*int*) – The page number of result display
- **start** (*int*) – Show search results form this index
- **limit** (*int*) – Show only up to this many entries
- **filterString** (*string*) – The filter attribute (optional). So far only the value `interval` has been observed.
- **searchString** (*string*) – Search value. If the filter attribute refers to a date the format is `YYYY-mm-dd HH:MM:SS`. For the `filterString interval` use an integer to view the history entries of the specified number of weeks.

Response JSON Object

- **success** (*boolean*) – If everything was ok
- **responseType** (*string*) – NAMI response type: Can be `OK`, `INFO`, `WARN`, `ERROR`, `EXCEPTION` or `null`
- **totalEntries** (*int*) – Number of returned members

- **data** (*list*) – The search results. Each returned history entry is a list entry in the form of a json array.

See also:

HistoryEntrySchema JSON schema of the returned history entry

- **metaData** (*json*) – Additional information about the data fields but not on the activities themselves.

Status Codes

- 200 OK – No error

GET /dashboard/stats/stats

Get basic statistics about your group

Response JSON Object

- **success** (*boolean*) – If everything was ok
- **responseType** (*string*) – NAMI response type: Can be OK, INFO, WARN, ERROR, EXCEPTION or null
- **message** (*string*) – Should be null.
- **title** (*string*) – Should be null.
- **data** (*json*) – Contains the statistical data about your group

See also:

StatsSchema JSON schema of the statistical data

Status Codes

- 200 OK – No error

Search Mitglieder

For validation of the search parameters the following class is used:

```
class SearchSchema (*, only: types.StrSequenceOrSet | None = None, exclude: types.StrSequenceOrSet
                    = (), many: bool = False, context: dict | None = None, load_only:
                    types.StrSequenceOrSet = (), dump_only: types.StrSequenceOrSet = (), partial:
                    bool | types.StrSequenceOrSet = False, unknown: str | None = None)
```

Base class for search parameters

You could just use `json.dumps()` on a dictionary to create the search string but this would bypass all the type and spell checking.

This also takes care of the miss-spelled search key 'mitgliedsNummber' so that the user of this API can use the correct spelling.

Example

Listing 1: Generate the search string

```
from pynami.search import SearchSchema

searchKeys = {'vorname': 'Max',
              'nachname': 'Mustermann'}
searchedValues = SearchSchema().dumps(searchKeys,
                                      separators=(',', ':'))
print(searchedValues)
```

Listing 2: Make direct use of the search keys by searching for all
Wölflinge and Jungpfadfinder

```
from pynami.tools import tabulate2x
from pynami.nami import NaMi

with NaMi(username='MITGLIEDSNUMMER', password='PASSWORD') as nami:
```

(continues on next page)

(continued from previous page)

```
searchResults = nami.search(untergliederungId=[1,2])
print(tabulate2x(searchResults))
```

`__model__`
alias of `pynami.schemas.base.BaseModel`

6.1 Search Keys

`SearchSchema.vorname`

First Name

Type `str`

`SearchSchema.funktion`

Unused tag

Type `str`

`SearchSchema.organisation`

Unused tag

Type `str`

`SearchSchema.nachname`

Surname

Type `str`

`SearchSchema.alterVon`

Minimal age

Type `str`

`SearchSchema.alterBis`

Maximal age

Type `str`

`SearchSchema.mglWohnort`

City

Type `str`

`SearchSchema.mitgliedsNummer`

The DPSG id

Type `str`

`SearchSchema.mglStatusId`

If the member is active

Type `str`

`SearchSchema.mglTypeId`

Kind of membership

Type `list of str`

`SearchSchema.tagId`

Kinds of fees

Type `list of int`

`SearchSchema.bausteinIncludeId`
Possible completed trainings

Type `list of int`

`SearchSchema.spitzname`
Nickname

Type `str`

`SearchSchema.zeitschriftenversand`
If the member gets the DPSG newspaper

Type `bool`

`SearchSchema.untergliederungId`
Tiers in which the members are active

Type `int`

`SearchSchema.taetigkeitId`
List of activities the members are committed to

Type `list of int`

`SearchSchema.mitAllenTaetigkeiten`
Whether to search in all active activities

Type `bool`

`SearchSchema.withEndedTaetigkeiten`
Whether to search also in activities that have already ended

Type `bool`

6.1.1 Restrict search to some group or level

There are two way to search only in a specified group or level which are mutually exclusive:

1. Choose the level and specify the group id and/or the group name

`SearchSchema.ebeneId`
Represents a DPSG structural level

Type `int`

`SearchSchema.grpNummer`
Group id

Type `str`

`SearchSchema.grpName`
Group name

Type `str`

2. Choose group ids from different levels from a automated list

`SearchSchema.gruppierung1Id`
Group id of a Diözese

Type `int`

`SearchSchema.gruppierung2Id`
Group id of a Bezirk

Type `int`

`SearchSchema.gruppierung3Id`
Group id of a Stamm

Type `int`

`SearchSchema.inGrp`
Whether to search in the group

Type `bool`

`SearchSchema.unterhalbGrp`
Whether to search in subgroups

Type `bool`

6.1.2 Unused search keys

`SearchSchema.gruppierung4Id`
Unused

Type `int`

`SearchSchema.gruppierung5Id`
Unused

Type `int`

`SearchSchema.gruppierung6Id`
Unused

Type `int`

`SearchSchema.privacy`
Unused

Type `str`

`SearchSchema.searchName`
Unused

Type `str`

`SearchSchema.searchType = 'MITGLIEDER'`
Default search type. Not even strictly necessary.

Type `str`

7.1 pynami package

7.1.1 Subpackages

pynami.schemas package

Submodules

pynami.schemas.activity module

Schemas for activities

class Activity (***kwargs*)

Bases: *pynami.schemas.base.BaseModel*

Main class for activities directly obtained by their id.

This class is intended to be instantiated by calling the `load()` method on a corresponding data dictionary.

class ActivitySchema (*, *only: types.StrSequenceOrSet | None = None, exclude: types.StrSequenceOrSet = (), many: bool = False, context: dict | None = None, load_only: types.StrSequenceOrSet = (), dump_only: types.StrSequenceOrSet = (), partial: bool | types.StrSequenceOrSet = False, unknown: str | None = None*)

Bases: *pynami.schemas.base.BaseSchema*

Schema class for the *Activity* class

__model__

alias of *Activity*

aktivBis = None

End date

Type `datetime`

aktivVon = None
Start date

Type `datetime`

beitragsArtId = None
Fee type

Type `int`

caeaGroup = None
Access rights for the group

Type `str`

caeaGroupForGf = None
Access rights for sub group

Type `str`

caeaGroupForGfId = None
Access right id

Type `int`

caeaGroupId = None
Access right id

Type `int`

gruppierung = None
Group associated wiht the activity

Type `str`

gruppierungId = None
Group id

Type `int`

id = None
Id

Type `int`

taetigkeit = None
Kind of activity

Type `str`

taetigkeitId = None
Activity id

Type `int`

untergliederung = None
Tier or group association

Type `str`

untergliederungId = None
tier or sub group id

Type `int`

```
class SearchActivity (**kwargs)
```

Bases: `pynami.schemas.base.BaseSearchModel`

Main class for activities wich come up as a search result.

This class is intended to be instantiated by calling the `load()` method on a corresponding data dictionary.

```
get_activity (nami, mglId)
```

Create a real `Activity` form the search result by getting the corresponding data set through the activity id.

Parameters

- **nami** (`NaMi`) – Main NAMI class
- **mglId** (`int`) – Member id (not DPSG Mitgliedsnummer)

Returns The activity object corresponding to this search result.

Return type `Activity`

```
class SearchActivitySchema (*, only: types.StrSequenceOrSet | None = None, exclude:
    types.StrSequenceOrSet = (), many: bool = False, context: dict |
    None = None, load_only: types.StrSequenceOrSet = (), dump_only:
    types.StrSequenceOrSet = (), partial: bool | types.StrSequenceOrSet =
    False, unknown: str | None = None)
```

Bases: `pynami.schemas.base.BaseSearchSchema`

Schema class for the `SearchActivity` class

```
__model__
```

alias of `SearchActivity`

```
entries_aktivBis = None
```

End date

Type `datetime`

```
entries_aktivVon = None
```

Start date

Type `datetime`

```
entries_anlagedatum = None
```

Creation date

Type `datetime`

```
entries_beitragsArt = None
```

Fee type

Type `str`

```
entries_caeaGroup = None
```

Access rights for the group

Type `str`

```
entries_caeaGroupForGf = None
```

Access rights for sub group

Type `str`

```
entries_gruppierung = None
```

Group associated wiht the activity

Type `str`

entries_mitglied = None
Member
Type `str`

entries_taetigkeit = None
Kind of activity
Type `str`

entries_untergliederung = None
Tier or group association
Type `str`

pynami.schemas.base module

This module contains some base classes

exception `AccessError`
Bases: `AttributeError`

class `BaseModel` (***kwargs*)
Bases: `object`

Base class for all the main classes.

It stores all data entries as instance attributes.

__getstate__ ()
Enable saving data through pickling
Returns Returns the `__dict__` attribute of the class.
Return type `dict`

__setattr__ (*name, value*)
Set certain fields to read-only

Parameters

- **name** (`str`) – Attribute name.
- **value** – New value.

Raises `AccessError` – In case of invalid write access.

Returns `None`

__setstate__ (*state*)
Enable loading data with the `pickle` module

Parameters **state** (`dict`) – Loaded data.

Returns `None`

_field_blacklist = []
Attribute names which are to be skipped while preparing tabulated output
Type `list`

_tabkeys = []
Default attributes names for tabulating and data export. This attribute is inherited and adapted to all Schema classes from `BaseModel`.

Type `list of str`

table_view (*field_blacklist=None*)

Prepare nicely formatted output

Parameters **field_blacklist** (`list of str`, optional) – List of attributes to be skipped

Returns All data entries which are not in the blacklist

Return type `dict`

tabulate (*elements=None*)

Prepare ordered tabulated output

Parameters **elements** (`list of str`, optional) – List of keys which shall be included in the table

Returns Specified data entries

Return type `OrderedDict`

```
class BaseSchema (*, only: types.StrSequenceOrSet | None = None, exclude: types.StrSequenceOrSet
                  = (), many: bool = False, context: dict | None = None, load_only:
                  types.StrSequenceOrSet = (), dump_only: types.StrSequenceOrSet = (), partial: bool
                  | types.StrSequenceOrSet = False, unknown: str | None = None)
```

Bases: `marshmallow.schema.Schema`

Base class for all Schemas in this module

It handles the formatting of dates so that the fields in the derived classes can be standard `DateTime` field.

Note: This class can not be used on its own but only as a derived class.

class Meta

Bases: `object`

This Base Meta class defines the default date and time format.

dateformat = `'%Y-%m-%d %H:%M:%S'`

Default NAMI date format. Note that in most cases the time part is only zeroes. However it can happen that the incoming value has time information as well.

Type `str`

datetimeformat = `'%Y-%m-%d %H:%M:%S'`

Default NAMI datetime format

Type `str`

__model__

`class`: Main class which this Schema is modelling. Each derived class must define this attribute.

alias of `BaseModel`

_declared_fields = {}

_hooks = {('post_load', False): ['make_object'], ('pre_load', False): ['correctEmptyStrings']}

correctEmptyStrings (*data, **kwargs*)

Replace empty strings with `None`

This method loops over all fields of the derived Schema class and where it finds a `DateTime` field and an empty string in the data `dict` it replaces this value with `None`.

To achieve that this is applied before loading this method is decorated with `pre_load()` decorator.

Parameters **data** (*dict*) – Data dictionary which is about to be loaded

Returns Adjusted data dictionary

Return type *dict*

make_object (*data*, ***kwargs*)

Create the object associated with this Schema by making use of the `post_load()` decorator.

Parameters

- **data** (*dict*) – Data dictionary containing all keywords and their values
- ****kwargs** – Possible keyword arguments passed through to this method during the loading procedure.

Returns Main class corresponding to this Schema

Return type *class*

class BaseSearchModel (***kwargs*)

Bases: *pynami.schemas.base.BaseModel*

Base class for all classes that are loaded from a *BaseSearchSchema*

_tabkeys = ['id', 'descriptor']

type

NAMI class without the hierarchy

Type *str*

```
class BaseSearchSchema (*, only: types.StrSequenceOrSet | None = None, exclude:
    types.StrSequenceOrSet = (), many: bool = False, context: dict |
    None = None, load_only: types.StrSequenceOrSet = (), dump_only:
    types.StrSequenceOrSet = (), partial: bool | types.StrSequenceOrSet =
    False, unknown: str | None = None)
```

Bases: *pynami.schemas.base.BaseSchema*

Base class for all schemas that describe search results.

All search results share the same three attributes.

__model__

alias of *BaseSearchModel*

_declared_fields = {'descriptor': <fields.String(dump_default=<marshmallow.missing>,

_hooks = {('post_load', False): ['make_object'], ('pre_load', False): ['correctEmpty

descriptor = None

Object description. In some cases the same as *representedClass*.

Type *str*

id = None

Internal id of the object This is an integer in most cases.

Type *int* ore *str*

representedClass = None

NAMI class structure

Type *str*

pynami.schemas.cogc module

This module contains classes to handle stuff with the certificates of good conduct.

class Bescheinigung (**kwargs)

Bases: `pynami.schemas.base.BaseModel`

Base data class for a certificate about the inspection of a certificate of goos conduct.

This class is intended to be instantiated by calling the `load()` method on a corresponding data dictionary.

download_fz (nami, **kwargs)

Open the certificate as a PDF file

Parameters

- **nami** (*NaMi*) – Main NAMi class
- ****kwargs** – See `open_download_pdf()`.

Returns `None`

class BescheinigungSchema (*, only: `types.StrSequenceOrSet` | `None` = `None`, exclude: `types.StrSequenceOrSet` = (), many: `bool` = `False`, context: `dict` | `None` = `None`, load_only: `types.StrSequenceOrSet` = (), dump_only: `types.StrSequenceOrSet` = (), partial: `bool` | `types.StrSequenceOrSet` = `False`, unknown: `str` | `None` = `None`)

Bases: `pynami.schemas.base.BaseSchema`

Schema class for the *Bescheinigung* class

__model__

alias of *Bescheinigung*

autor = `None`

Person who did the inspection

Type `str`

datumEinsicht = `None`

Inspection date. May be empty.

Type `datetime`

download = `None`

Relative download URL

Type `str`

empfaenger = `None`

Receiver

Type `str`

erstelltAm = `None`

Entry creation date

Type `datetime`

fzDatum = `None`

Date of the certificate of good conduct

Type `datetime`

fzNummer = `None`

Number of the certificate of good conduct

Type `str`

get_download_url (*data*, ***kwargs*)

Extract the HTML (Hypertext Markup Language) enclosed URL string from the pdf download field

Parameters *data* (*dict*) – Data dictionary

Returns The updated data dictionary

Return type `dict`

id = `None`

Id of this certificate

Type `int`

class **SearchBescheinigung** (***kwargs*)

Bases: `pynami.schemas.base.BaseSearchModel`

Base data class for a certificate about the inspection of a certificate of goos conduct that came up as a search result.

This class is intended to be instantiated by calling the `load()` method on a corresponding data dictionary.

download_fz (*nami*, ***kwargs*)

Open the certificate as a PDF file

Parameters

- **nami** (*NaMi*) – Main NAMi class
- ****kwargs** – See `open_download_pdf()`.

Returns `None`

get_bescheinigung (*nami*)

Create a real *Bescheinigung* form the search result by getting the corresponding data set through the certificate id.

Parameters **nami** (*NaMi*) – Main NAMi class

Returns The certificate object corresponding to this search result.

Return type *Bescheinigung*

class **SearchBescheinigungSchema** (*, *only: types.StrSequenceOrSet | None = None, exclude: types.StrSequenceOrSet = (), many: bool = False, context: dict | None = None, load_only: types.StrSequenceOrSet = (), dump_only: types.StrSequenceOrSet = (), partial: bool | types.StrSequenceOrSet = False, unknown: str | None = None*)

Bases: `pynami.schemas.base.BaseSearchSchema`

Schema class for the *SearchBescheinigung* class

__model__

alias of *SearchBescheinigung*

entries_autor = `None`

Person who did the inspection

Type `str`

entries_datumEinsicht = `None`

Inspection date. May be empty.

Type `datetime`

```

entries_empfGebDatum = None
    Birth date

    Type datetime

entries_empfNachname = None
    Surname

    Type str

entries_empfVorname = None
    First name

    Type str

entries_empfaenger = None
    Receiver

    Type str

entries_erstelltAm = None
    Entry creation date

    Type datetime

entries_fzDatum = None
    Date of the certificate of good conduct

    Type datetime

entries_fzNummer = None
    Number of the certificate of good conduct

    Type str

```

pynami.schemas.dashboard module

Schemas for dashboard functionalities

```

class Notification (**kwargs)
    Bases: pynami.schemas.base.BaseSearchModel

```

Main class for notification like tier changes of a mitglied.

In the NAMI the notifications are displayed in the dashboard.

This class is intended to be instantiated by calling the `load()` method on a corresponding data dictionary.

```

class NotificationSchema (*, only: types.StrSequenceOrSet | None = None, exclude:
    types.StrSequenceOrSet = (), many: bool = False, context: dict |
    None = None, load_only: types.StrSequenceOrSet = (), dump_only:
    types.StrSequenceOrSet = (), partial: bool | types.StrSequenceOrSet =
    False, unknown: str | None = None)
    Bases: pynami.schemas.base.BaseSearchSchema

```

Schema class for the `Notification` class

```

__model__
    alias of Notification

```

```

entries_actor = None
    The person who started the event

    Type str

```

entries_actorId = None

Id of the person who started the event

Type `int`

entries_changedFields = None

Which fields have been changed. This may be empty.

Type `str`

entries_completeChanges = None

More details about the changes. This may be empty.

Type `str`

entries_entryDate = None

Date of the event

Type `datetime`

entries_id = None

Id of the event

Type `int`

entries_newObject = None

New object

Type `str`

entries_objectClass = None

NAMi class

Type `str`

entries_objectId = None

Object id

Type `int`

entries_operation = None

The nature of the change

Type `str`

entries_originalObject = None

Old object. This may be empty

Type `str`

```
class StatCatSchema(*, only: types.StrSequenceOrSet | None = None, exclude:
    types.StrSequenceOrSet = (), many: bool = False, context: dict | None = None,
    load_only: types.StrSequenceOrSet = (), dump_only: types.StrSequenceOrSet =
    (), partial: bool | types.StrSequenceOrSet = False, unknown: str | None = None)
```

Bases: `pynami.schemas.base.BaseSchema`

Schema class for the `StatCategory` class.

This only contains the name of the tier and its number of members.

__model__

alias of `StatCategory`

count = None

Number of members in the group

Type `int`

name = None
Name of the tier/group

Type *str*

class StatCategory (***kwargs*)
Bases: *pynami.schemas.base.BaseModel*

Main class for statistical tier numbers.

This class is intended to be instantiated by calling the `load()` method on a corresponding data dictionary.

class Stats (***kwargs*)
Bases: *pynami.schemas.base.BaseModel*

Main class for basic statistical entries.

The information in this class is displayed in the NAMI dashboard in a pie chart.

This class is intended to be instantiated by calling the `load()` method on a corresponding data dictionary.

class StatsSchema (*, *only: types.StrSequenceOrSet | None = None, exclude: types.StrSequenceOrSet = (), many: bool = False, context: dict | None = None, load_only: types.StrSequenceOrSet = (), dump_only: types.StrSequenceOrSet = (), partial: bool | types.StrSequenceOrSet = False, unknown: str | None = None*)
Bases: *pynami.schemas.base.BaseSchema*

Schema class for the *Stats* class

__model__
alias of *Stats*

nrMitglieder = None
Total number of members in the group

Type *int*

statsCategories = None
Detailed information on the partitions

Type *list of StatCategory*

pynami.schemas.default module

Schema for default values

class Baseadmin (***kwargs*)
Bases: *pynami.schemas.base.BaseSearchModel*

Base data class for all default values and their id mapping.

This class is intended to be instantiated by calling the `load()` method on a corresponding data dictionary.

class BaseadminSchema (*, *only: types.StrSequenceOrSet | None = None, exclude: types.StrSequenceOrSet = (), many: bool = False, context: dict | None = None, load_only: types.StrSequenceOrSet = (), dump_only: types.StrSequenceOrSet = (), partial: bool | types.StrSequenceOrSet = False, unknown: str | None = None*)
Bases: *pynami.schemas.base.BaseSearchSchema*

Schema class for the *Baseadmin* class

All the default values only consist of the same four attributes

```
__model__  
    alias of Baseadmin  
  
name = None  
    Name of this value. This will be empty in many cases.  
  
Type str
```

pynami.schemas.grpadmin module

This module contains classes for group admin (*Gruppierungsverwaltung*) stuff

```
class Invoice (**kwargs)  
    Bases: pynami.schemas.base.BaseModel  
  
    Represents an invoice  
  
    This class is intended to be instantiated by calling the load() method on a corresponding data dictionary.  
  
    download (nami, **kwargs)  
        Download this invoice as PDF
```

Parameters

- **nami** (*NaMi*) – Main NAMI class
- ****kwargs** – See `open_download_pdf()`.

Returns *None*

```
class InvoiceSchema (*, only: types.StrSequenceOrSet | None = None, exclude:  
    types.StrSequenceOrSet = (), many: bool = False, context: dict | None = None,  
    load_only: types.StrSequenceOrSet = (), dump_only: types.StrSequenceOrSet =  
    (), partial: bool | types.StrSequenceOrSet = False, unknown: str | None = None)  
    Bases: pynami.schemas.base.BaseSchema
```

Schema class for the *Invoice* class

```
__model__  
    alias of Invoice  
  
currency = None  
    Currency (e.g. 'EUR')  
  
Type str  
  
debitor = None  
    Id of the debitor  
  
Type str  
  
debitorKonto = None  
    Some account id  
  
Type str  
  
debitorName = None  
    Debitor, e.g. a group  
  
Type str  
  
debitor_document_id = None  
    Some other internal id  
  
Type int
```


displayName = None
Human-readable string describing the invoice
Type `str`

einzugsDatum = None
Date of money collection
Type `datetime`

erloesKtoName = None
This may be empty
Type `str`

freigabeDatum = None
When the invoice was released
Type `datetime`

get_download_url (*data*, ***kwargs*)
Extract the HTML enclosed URL string from the pdf download field
Parameters *data* (*dict*) – Data dictionary
Returns The updated data dictionary
Return type `dict`

id = None
NAMI id
Type `int`

pdf = None
Relative download URL
Type `str`

reCreated = None
Creation date of the invoice
Type `datetime`

reDatum = None
Date of the invoice
Type `datetime`

reNr = None
Official invoice number
Type `str`

rechnungsLauf = None
Internal invoice number
Type `int`

status = None
If the invoice has been released
Type `str`

total = None
Total amount

Type `str`

zahlungsweise = `None`

Way of payment (e.g. 'Lastschrift')

Type `str`

zahlungsweiseId = `None`

Id of the way of payment (e.g. 'LASTSCHRIFT')

Type `str`

class SearchInvoice (***kwargs*)

Bases: `pynami.schemas.base.BaseSearchModel`

Represents an invoice that came up as a search result

This class is intended to be instantiated by calling the `load()` method on a corresponding data dictionary.

download (*nami, **kwargs*)

Download this invoice as PDF

Parameters

- **nami** (*NaMi*) – Main NAMi class
- ****kwargs** – See `open_download_pdf()`.

Returns `None`

get_invoice (*nami, grpId=None*)

Create a real *Invoice* form the search result by getting the corresponding data set through the training id.

Parameters

- **nami** (*NaMi*) – Main NAMi class
- **grpId** (*str*, optional) – DPSG Group id. Defaults to group id of the user

Returns The invoice object corresponding to this search result.

Return type *Invoice*

```
class SearchInvoiceSchema (*, only: types.StrSequenceOrSet | None = None, exclude:
    types.StrSequenceOrSet = (), many: bool = False, context: dict |
    None = None, load_only: types.StrSequenceOrSet = (), dump_only:
    types.StrSequenceOrSet = (), partial: bool | types.StrSequenceOrSet =
    False, unknown: str | None = None)
```

Bases: `pynami.schemas.base.BaseSearchSchema`

Schema class for the *SearchInvoice* class

__model__

alias of *SearchInvoice*

entries_debitor = `None`

Id of the debtor

Type `str`

entries_debitorId = `None`

e.g. a group id

Type `int`

entries_debitorName = None
Debitor, e.g. a group
Type `str`

entries_debitorType = None
NAMI class
Type `str`

entries_debitor_document_id = None
Some other internal id
Type `int`

entries_displayName = None
Human-readable string describing the invoice
Type `str`

entries_einzugsDatum = None
Date of money collection
Type `datetime`

entries_erloesKtoName = None
This may be empty
Type `str`

entries_fibuDebitorKonto = None
Some account id
Type `str`

entries_fibuErloesKonto = None
This may be empty
Type `str`

entries_freigabeDatum = None
Type `datetime`

entries_kontoOwnerId = None
Id of the account owner
Type `int`

entries_kontoOwnerTyp = None
NAMI class
Type `str`

entries_kontoverbindung = None
Account details
Type `str`

entries_kreditor = None
May be empty
Type `str`

entries_reCreated = None
Creation date of the invoice

Type `datetime`
entries_reDatum = `None`
Date of the invoice

Type `datetime`
entries_reMwst = `None`
VAT (Value added tax) amount including currency

Type `str`
entries_reNetto = `None`
Netto amount including currency

Type `str`
entries_reNr = `None`
Official invoice number

Type `str`
entries_rechnungsEmpfaenger = `None`
Recipient

Type `str`
entries_rechnungsLauf = `None`
Internal invoice number

Type `int`
entries_status = `None`
If the invoice has been released

Type `str`
entries_zahlungsEmpfaenger = `None`
Recipient of the payment

Type `str`

pynami.schemas.history module

Schemas for history entries

class **HistoryEntry** (***kwargs*)
Bases: `pynami.schemas.base.BaseSearchModel`

Main class for history entries

This contains information about an update of a Mitglied which is displayed on the NAMI dashbiard.

This class is intended to be instantiated by calling the `load()` method on a corresponding data dictionary.

get_history (*nami, mglId, ext=True*)
Create a real *MitgliedHistory* form the search result by getting the corresponding data set through the history entry id.

Parameters

- **nami** (*NaMi*) – Main NAMI class
- **mglId** (*int*) – Member id (not DPSG Mitgliedsnummer)

- **ext** (*bool*, optional) – If the extended history format should be used. Defaults to *True*.

Returns The activity object corresponding to this search result.

Return type *MitgliedHistory*

```
class HistoryEntrySchema(*, only: types.StrSequenceOrSet | None = None, exclude:
    types.StrSequenceOrSet = (), many: bool = False, context: dict |
    None = None, load_only: types.StrSequenceOrSet = (), dump_only:
    types.StrSequenceOrSet = (), partial: bool | types.StrSequenceOrSet =
    False, unknown: str | None = None)
```

Bases: *pynami.schemas.base.BaseSearchSchema*

Schema class for the *HistoryEntry* class

__model__

alias of *HistoryEntry*

entries_actor = None

Name of the person who created the change including the id

Type *str*

entries_actorId = None

Id of the person who created the change

Type *int*

entries_author = None

Who did this

Type *str*

entries_changedFields = None

Which fields have been changed. This may be empty.

Type *str*

entries_completeChanges = None

More details about the changes. This may be empty.

Type *str*

entries_entryDate = None

Date of the event

Type *datetime*

entries_gruppierung = None

Group name including its id

Type *str*

entries_id = None

NAMi id

Type *int*

entries_mitglied = None

Almost the same as *entries_author*

Type *str*

entries_newObject = None

Updated object ('e.g. a Mitglied)

Type *str*

entries_objectClass = None

NAMi class

Type `str`

entries_objectId = None

Object id (not the NAMi id for addressing the entry)

Type `int`

entries_operation = None

What kind of action has been done.

Type `str`

entries_originalObject = None

The object before the change. This may be empty for a `GruppierungsHistoryEntry`.

Type `str`

class MitgliedHistory (***kwargs*)

Bases: `pynami.schemas.base.BaseModel`

Main class for a member revision history entry obtained directly from its id.

This class is intended to be instantiated by calling the `load()` method on a corresponding data dictionary.

class MitgliedHistorySchema (*, *only: types.StrSequenceOrSet | None = None, exclude: types.StrSequenceOrSet = (), many: bool = False, context: dict | None = None, load_only: types.StrSequenceOrSet = (), dump_only: types.StrSequenceOrSet = (), partial: bool | types.StrSequenceOrSet = False, unknown: str | None = None*)

Bases: `pynami.schemas.base.BaseSchema`

Schema class for the `MitgliedHistory` class

__model__

alias of `MitgliedHistory`

actor = None

Name of the person who created the change including the id

Type `str`

changedFields = None

Which fields have been changed. This may be empty.

Type `str`

entryDate = None

Date of the event

Type `datetime`

gruppierung = None

Group name including its id

Type `str`

id = None

NAMi id

Type `int`

operation = None

What kind of action has been done.

Type `str`

pynami.schemas.mgl module

Schemas for operations on members

class `Mitglied` (**kwargs)

Bases: `pynami.schemas.base.BaseModel`

Main class representing a NAMI Mitglied

This class overwrites the `__getattr__()` and `__setattr__()` methods so that attributes of this class can be handled in a convenient way. It is intended to be instantiated by calling the `load()` method on a corresponding data dictionary.

update (*nami*)

Writes the possibly changed values to the NAMI

Parameters `nami` (*NaMi*) – Main class for communication with the NAMI

Returns The new Mitglied as it is returned by the NAMI

Return type *Mitglied*

```
class MitgliedSchema(*, only: types.StrSequenceOrSet | None = None, exclude:
    types.StrSequenceOrSet = (), many: bool = False, context: dict | None = None,
    load_only: types.StrSequenceOrSet = (), dump_only: types.StrSequenceOrSet
    = (), partial: bool | types.StrSequenceOrSet = False, unknown: str | None =
    None)
```

Bases: `pynami.schemas.base.BaseSchema`

Schema class for a *Mitglied*

class `Meta`

Bases: `pynami.schemas.base.Meta`

Extended `marshmallow.Schema.Meta` class for further configuration

ordered = `True`

This is nessecary for the payment deatils to be dumped in the corrent order.

Type `bool`

__model__

alias of *Mitglied*

austrittsDatum = `None`

Date of the end of membership

Type `date`

beitragsart = `None`

Fee type

Type `str`

beitragsartId = `None`

Id of the fee type

Type `int`

eintrittsdatum = `None`

Start of membership

Type `date`

email = None
Primary member email
Type `str`

emailVertretungsberechtigter = None
Email address of an authorized representative.
Type `str`

ersteTaetigkeit = None
First activity. May be empty.
Type `str`

ersteTaetigkeitId = None
Id of the first activity. Defaults to `null`.
Type `int`

ersteUntergliederung = None
First tier. May be empty
Type `str`

ersteUntergliederungId = None
Id of the first tier. This may be empty.
Type `int`

fixBeitrag = None
Defaults to `null`.
Type `str`

geburtsDatum = None
Birth date
Type `date`

genericField1 = None
Not sure why these even exist.
Type `str`

genericField2 = None
Not sure why these even exist.
Type `str`

geschlecht = None
Gender
Type `str`

geschlechtId = None
Gender id
Type `int`

gruppierung = None
Group name including its id
Type `str`

gruppierungId = None

Group id

Note: In the search result (see [SearchMitgliedSchema](#)) this comes as a `str`.

Type `int`**id = None**

NAMI internal id (not DPSG member nummer).

Type `int`**jungpfadfinder = None**

Tier field. Not sure what it is for.

Type `str`**konfession = None**

Confession

Type `str`**konfessionId = None**

Id of the confession

Type `int`**kontoverbindung = None**Account details. In a search result this comes as a `str`.**Type** `NamiKontoSchema`**land = None**

Address country

Type `str`**landId = None**

Id of the address country

Type `int`**lastUpdated = None**Date of the last update. This value is not dumped when updating a `Mitglied`.**Type** `datetime`**mglType = None**

Type of membership (e.g. 'Mitglied')

Type `str`**mglTypeId = None**

Id of the type of membership (typically just the value in uppercase).

Type `str`**mitgliedsNummer = None**

DPSG member number

Type `int`

nachname = None
Surname
Type `str`

nameZusatz = None
Extra name info
Type `str`

ort = None
Address city
Type `str`

pfadfinder = None
Tier field. Not sure what it is for.
Type `str`

plz = None
Postal code
Type `str`

region = None
A *Bundesland* or foreign country
Type `str`

regionId = None
Region id
Type `int`

rover = None
Tier field. Not sure what it is for.
Type `str`

sonst01 = None
Another generic field. Defaults to `False`.
Type `bool`

sonst02 = None
Another generic field. Defaults to `False`.
Type `bool`

spitzname = None
Nickname
Type `str`

staatsangehoerigkeit = None
Nationality
Type `str`

staatsangehoerigkeitId = None
Id of the nationality
Type `int`

staatsangehoerigkeitText = None
Extra nationality info. Empty in most cases.

```

    Type str
    status = None
        If the member is active or not

    Type str
    strasse = None
        Address information

    Type str
    stufe = None
        Current tier

    Type str
    telefax = None
        Fax number. Who uses these anyway today?

    Type str
    telefon1 = None
        First phone number

    Type str
    telefon2 = None
        Second phone number

    Type str
    telefon3 = None
        Third phone number

    Type str
    version = None
        History version number

    Type int
    vorname = None
        First name

    Type str
    wiederverwendenFlag = None
        If the member data may be used after the membership ends

    Type bool
    woelfling = None
        Tier field. Not sure what it is for.

    Type str
    zeitschriftenversand = None
        If the member gets the DPSG newspaper.

    Type bool

class NamiKonto (**kwargs)
    Bases: pynami.schemas.base.BaseModel
    Holds information about bank account and payment method of the member.

```

This class is intended to be instantiated by calling the `load()` method on a corresponding data dictionary.

```
class NamiKontoSchema(*, only: types.StrSequenceOrSet | None = None, exclude:
    types.StrSequenceOrSet = (), many: bool = False, context: dict | None = None,
    load_only: types.StrSequenceOrSet = (), dump_only: types.StrSequenceOrSet
    = (), partial: bool | types.StrSequenceOrSet = False, unknown: str | None =
    None)
```

Bases: `pynami.schemas.base.BaseSchema`

Schema class for bank account and payment method information.

This Schema will be `nested` inside a `MitgliedSchema` and there should be no other use for it.

`__model__`

alias of `NamiKonto`

bankleitzahl = None

Bank sort code

Type `str`

bic = None

BIC (Bank Identifier Code)

Type `str`

double_dump (*data*, ***kwargs*)

Incoming data sets are nicely formatted `json` strings which can be loaded easily into the Schema but when you update a `Mitglied` all payment details have to be formatted into a `json` string and all attributes have to be in a certain order. To achieve this the `post_dump()` decorator is used.

Parameters *data* (*dict*) – Already dumped data set

Returns A `json` formatted string

Return type `str`

iban = None

IBAN (International Bank Account Number)

Type `str`

id = None

NAMI internal id of this payment details

Type `int`

id_to_str (*data*, ***kwargs*)

For some reason the NAMI gives the id of the payment details as an `integer`, but when you want to update a member it has to be a `string`. Therefore this method converts incoming ids to a `str` object before loading them by making use of the `pre_load()` decorator.

Parameters *data* (*dict*) – Data dictionary to be loaded

Returns Corrected data dictionary

Return type `dict`

institut = None

Bank

Type `str`

kontoinhaber = None

Account holder

```

    Type str
    kontonummer = None
        Account number

    Type str
    mitgliedsNummer = None
        DPSG member id

    Type int
    zahlungsKondition = None
        Kind of payment (e.g. 'Std Lastschrift'). This attribute is not dumped when updating a Mitglied.

    Type str
    zahlungsKonditionId = None
        Id corresponding to the kind of payment

    Type int

class SearchMitglied(**kwargs)
    Bases: pynami.schemas.base.BaseSearchModel

    Main class for a Mitglied which came up as a search result. Unfortunately there cannot be just one Mitglied
    class because the search results lack crucial information (e.g. payment details).

    get_mitglied(nami)
        Create a real Mitglied from the search result by getting the corresponding data set through the member
        id.

        Parameters nami (NaMi) – Main NAMI class

        Returns The Mitglied object corresponding to this search result.

        Return type Mitglied

class SearchMitgliedSchema(*, only: types.StrSequenceOrSet | None = None, exclude:
    types.StrSequenceOrSet = (), many: bool = False, context: dict |
    None = None, load_only: types.StrSequenceOrSet = (), dump_only:
    types.StrSequenceOrSet = (), partial: bool | types.StrSequenceOrSet =
    False, unknown: str | None = None)
    Bases: pynami.schemas.base.BaseSearchSchema

    Schema class for SearchMitglied.

    For some reason attribute naming in the NAMI is a bit inconsistent between a Mitglied which comes up as a
    search result and one that is addressed directly by its id.

    __model__
        alias of SearchMitglied

    entries_austrittsDatum = None
        Date of the end of membership

        Type date

    entries_beitragsarten = None
        Fee type

        Type str

    entries_eintrittsdatum = None
        Start of membership

```

Type `date`

entries_email = None

Primary member email

Type `str`

entries_emailVertretungsberechtigter = None

Email address of an authorized representative.

Type `str`

entries_ersteTaetigkeitId = None

Id of the first activity. Defaults to `null`.

Type `int`

entries_ersteUntergliederungId = None

Id of the first tier. This may be empty.

Type `int`

entries_fixBeitrag = None

Defaults to `null`.

Type `str`

entries_geburtsDatum = None

Birth date

Type `date`

entries_genericField1 = None

Not sure why these even exist.

Type `str`

entries_genericField2 = None

Not sure why these even exist.

Type `str`

entries_geschlecht = None

Gender

Type `str`

entries_gruppierung = None

Group name including its id

Type `str`

entries_gruppierungId = None

Group id as a string

Type `str`

entries_id = None

NAMi internal id (not DPSG member nummer).

Type `int`

entries_jungpfadfinder = None

Tier field. Not sure what it is for.

Type `str`

entries_konfession = None

Confession

Type `str`

entries_kontoverbindung = None

Account details. For some reason this is not always transmitted and may therefore be empty.

Type `str`

entries_lastUpdated = None

Date of the last update

Type `datetime`

entries_mglType = None

Type of membership (e.g. 'Mitglied')

Type `str`

entries_mitgliedsNummer = None

DPSG member number

Type `int`

entries_nachname = None

Surname

Type `str`

entries_pfadfinder = None

Tier field. Not sure what it is for.

Type `str`

entries_rover = None

Tier field. Not sure what it is for.

Type `str`

entries_rowCssClass = None

Unused. The purpose could not be found out.

Type `str`

entries_spitzname = None

Nickname

Type `str`

entries_staatangehorigkeitText = None

Extra nationality info. Empty in most cases.

Type `str`

entries_staatsangehorigkeit = None

Nationality

Type `str`

entries_status = None

If the member is active or not

Type `str`

entries_stufe = None

Current tier

```
    Type str
entries_telefax = None
    Fax number. Who uses these anyway today?

    Type str
entries_telefon1 = None
    First phone number

    Type str
entries_telefon2 = None
    Second phone number

    Type str
entries_telefon3 = None
    Third phone number

    Type str
entries_version = None
    History version number

    Type int
entries_vorname = None
    First name

    Type str
entries_wiederverwendenFlag = None
    If the member data may be used after the membership ends

    Type bool
entries_woelfling = None
    Tier field. Not sure what it is for.

    Type str
```

pynami.schemas.search module

This module contains the classes for a search for members

```
class SearchSchema (*, only: types.StrSequenceOrSet | None = None, exclude: types.StrSequenceOrSet
                    = (), many: bool = False, context: dict | None = None, load_only:
                    types.StrSequenceOrSet = (), dump_only: types.StrSequenceOrSet = (), partial:
                    bool | types.StrSequenceOrSet = False, unknown: str | None = None)
Bases: pynami.schemas.base.BaseSchema
```

Base class for search parameters

You could just use `json.dumps()` on a dictionary to create the search string but this would bypass all the type and spell checking.

This also takes care of the miss-spelled search key `'mitgliedsNummber'` so that the user of this API can use the correct spelling.

Example

Listing 1: Generate the search string

```

from pynami.search import SearchSchema

searchKeys = {'vorname': 'Max',
              'nachname': 'Mustermann'}
searchedValues = SearchSchema().dumps(searchKeys,
                                      separators=(',', ':'))
print(searchedValues)

```

Listing 2: Make direct use of the search keys by searching for all
Wölflinge and Jungpfadfinder

```

from pynami.tools import tabulate2x
from pynami.nami import NaMi

with NaMi(username='MITGLIEDSNUMMER', password='PASSWORD') as nami:
    searchResults = nami.search(untergliederungId=[1,2])
    print(tabulate2x(searchResults))

```

class MetaBases: `pynami.schemas.base.Meta`Extended `marshmallow.Schema.Meta` class for further configuration**ordered = True**

All attributes shall be dumped in code order

Type `bool`**alterBis = None**

Maximal age

Type `str`**alterVon = None**

Minimal age

Type `str`**bausteinIncludeId = None**

Possible completed trainings

Type `list of int`**correct_spelling (data, **kwargs)**Check for misspelled or incorrect data before dumping. Realized by the `pre_dump()` decorator.**Parameters** `data (dict)` – Data set to be dumped**Returns** Corrected data**Return type** `dict`**Raises** `KeyError` – When there is a search key that is not an attribute of this class**ebeneId = None**

Represents a DPSG structural level

Type `int`

funktion = None
Unused tag
Type `str`

grpName = None
Group name
Type `str`

grpNummer = None
Group id
Type `str`

gruppierung1Id = None
Group id of a Diözese
Type `int`

gruppierung2Id = None
Group id of a Bezirk
Type `int`

gruppierung3Id = None
Group id of a Stamm
Type `int`

gruppierung4Id = None
Unused
Type `int`

gruppierung5Id = None
Unused
Type `int`

gruppierung6Id = None
Unused
Type `int`

inGrp = None
Whether to search in the group
Type `bool`

mglStatusId = None
If the member is active
Type `str`

mglTypeId = None
Kind of membership
Type `list of str`

mglWohnort = None
City
Type `str`

mitAllenTaetigkeiten = None
Whether to search in all active activities

```

    Type bool
mitgliedsNummer = None
    The DPSG id

    Type str
nachname = None
    Surname

    Type str
organisation = None
    Unused tag

    Type str
privacy = None
    Unused

    Type str
searchName = None
    Unused

    Type str
searchType = None
    Default search type. Not even strictly neccessary.

    Type str
spitzname = None
    Nickname

    Type str
taetigkeitId = None
    List of activities the mambers are comitted to

    Type list of int
tagId = None
    Kinds of fees

    Type list of int
untergliederungId = None
    Tiers in which the members are active

    Type int
unterhalbGrp = None
    Whether to search in subgroups

    Type bool
vorname = None
    First Name

    Type str
withEndedTaetigkeiten = None
    Whether to search also in activities that have already ended

    Type bool

```

zeitschriftenversand = None

If the member gets the DPSG newspaper

Type `bool`

pynami.schemas.training module

Schemas for trainings

class `Ausbildung` (***kwargs*)

Bases: `pynami.schemas.base.BaseModel`

Main class for a training obtained directly from its id.

This class is intended to be instantiated by calling the `load()` method on a corresponding data dictionary.

class `AusbildungSchema` (*, *only: types.StrSequenceOrSet | None = None, exclude: types.StrSequenceOrSet = (), many: bool = False, context: dict | None = None, load_only: types.StrSequenceOrSet = (), dump_only: types.StrSequenceOrSet = (), partial: bool | types.StrSequenceOrSet = False, unknown: str | None = None*)

Bases: `pynami.schemas.base.BaseSchema`

Schema class for the `Ausbildung` class

__model__

alias of `Ausbildung`

baustein = None

Name of the training (e.g. 'Baustein 3a')

Type `str`

bausteinId = None

Id of the training

Type `int`

id = None

NAMi id

Type `int`

lastModifiedFrom = None

Who did the last change to this entry

Type `str`

mitglied = None

Who absolved the training

Type `str`

veranstalter = None

Who organized the event (e.g. a *Bezirk*)

Type `str`

vstgName = None

Name of the event

Type `str`

vstgTag = None
Day of the training event

Type `date`

class SearchAusbildung (***kwargs*)
Bases: `pynami.schemas.base.BaseSearchModel`

Main class for a training obtained as a search result.

This class is intended to be instantiated by calling the `load()` method on a corresponding data dictionary.

get_ausbildung (*nami, mglId*)
Create a real `Ausbildung` form the search result by getting the corresponding data set through the training id.

Parameters

- **nami** (`NaMi`) – Main NAMI class
- **mglId** (`int`) – Member id (not DPSG Mitgliedsnummer)

Returns The activity object corresponding to this search result.

Return type `Ausbildung`

class SearchAusbildungSchema (*, *only: types.StrSequenceOrSet | None = None, exclude: types.StrSequenceOrSet = (), many: bool = False, context: dict | None = None, load_only: types.StrSequenceOrSet = (), dump_only: types.StrSequenceOrSet = (), partial: bool | types.StrSequenceOrSet = False, unknown: str | None = None*)
Bases: `pynami.schemas.base.BaseSearchSchema`

Schema class for the `SearchAusbildung` class

__model__
alias of `SearchAusbildung`

entries_baustein = None
Name of the training (e.g. 'Baustein 3a')

Type `str`

entries_id = None
NAMI id of this training entry

Type `int`

entries_mitglied = None
Who absolved the training

Type `str`

entries_veranstalter = None
Who organized the event (e.g. a *Bezirk*)

Type `str`

entries_vstgName = None
Name of the event

Type `str`

entries_vstgTag = None
Day of the training event

Type `date`

pynami.schemas.tags module

Schemas for tags

```
class SearchTag (**kwargs)
```

Bases: `pynami.schemas.base.BaseSearchModel`

Base data class for a tag that came up as a search result.

This class is intended to be instantiated by calling the `load()` method on a corresponding data dictionary.

```
class SearchTagSchema (*, only: types.StrSequenceOrSet | None = None, exclude:
    types.StrSequenceOrSet = (), many: bool = False, context: dict | None = None,
    load_only: types.StrSequenceOrSet = (), dump_only: types.StrSequenceOrSet
    = (), partial: bool | types.StrSequenceOrSet = False, unknown: str | None =
    None)
```

Bases: `pynami.schemas.base.BaseSearchSchema`

Schema class for the SearchBescheinigung class

```
__model__
```

alias of `SearchTag`

```
entries_identitaet = None
```

Identity of the member with the DPSG Mitgliedsnummer

Type `str`

```
entries_tag = None
```

Actual value of the tag

Type `str`

```
class Tag (**kwargs)
```

Bases: `pynami.schemas.base.BaseModel`

Base data class for a tag

This class is intended to be instantiated by calling the `load()` method on a corresponding data dictionary.

```
class TagSchema (*, only: types.StrSequenceOrSet | None = None, exclude: types.StrSequenceOrSet = (),
    many: bool = False, context: dict | None = None, load_only: types.StrSequenceOrSet
    = (), dump_only: types.StrSequenceOrSet = (), partial: bool | types.StrSequenceOrSet
    = False, unknown: str | None = None)
```

Bases: `pynami.schemas.base.BaseSchema`

Schema class for the `Tag` class

```
__model__
```

alias of `Tag`

```
id = None
```

Id of this tag

Type `int`

```
mitglied = None
```

Surname and first name of the member this tag belongs to

Type `str`

```
mitgliedId = None
```

Member id (not DPSG Mitgliedsnummer)

Type `int`

tag = None
 Actual value of the tag
 Type `str`

tagId = None
 Same as `id`
 Type `int`

7.1.2 Submodules

pynami.constants module

This module defines some constants needed for communication with the NAMI.

BASE_URL = 'https://nami.dpsg.de/ica/'
 Base URL of the server

Type `str`

DEFAULT_PARAMS = {'limit': 10000, 'page': 1, 'start': 0}
 Default parameters to avoid pagination. The NAMI uses 5000 as limit when using the *show all* option.

Type `dict`

class URLMetaClass

Bases: `type`

Meta class for `URLS`

Makes the class accessible like a dict and prepends the base url when this is done. The relative URL string can be retrieved in the usual way as attributes.

class URLS

Bases: `object`

Contains all relative REST (Representational State Transfer) URLs.

The class attributes can be accessed like a dict in order to get the full URL (prepending the *base URL*). The relative URL strings can be retrieved like normal class attributes.

Example

```
>>> URLS.AUTH
'/nami/auth/manual/sessionStartup'
>>> URLS['AUTH']
'https://nami.dpsg.de/ica/rest/nami/auth/manual/sessionStartup'
```

This is not just a dict so that each URL can be documented.

Note: Some URL strings have to be formatted with some variables.

ALLE_TAETIGKEITEN = '/nami/search-multi/all-visible-taetigkeiten'

AUSBILDUNG = '/nami/mitglied-ausbildung/filtered-for-navigation/mitglied/mitglied/'

AUTH = '/nami/auth/manual/sessionStartup'

```
BAUSTEIN = '/module/baustein'
BEANTRAGUNG = '/fz-beantragen/download-beantragung'
BEITRAGSART = '/namiBeitrag/beitragsart/gruppierung/{grpId}/'
BEITRAGSART_MGL = '/namiBeitrag/beitragsartmgl/gruppierung/{grpId}/'
EBENE = '/orgadmin/ebene/'
EBENE1 = '/nami/search-multi/ebene/1'
EBENE2 = '/nami/search-multi/ebene/2/gruppierung1/{grpId}/'
EBENE3 = '/nami/search-multi/ebene/3/gruppierung2/{grpId}/'
FZ = '/nami/fz/eigene-bescheinigungen/'
GESCHLECHT = '/baseadmin/geschlecht/'
GETMGL = '/nami/mitglied/filtered-for-navigation/gruppierung/gruppierung/{gruppierung}'
GET_TAG = '/nami/mitglied-tagged-item/filtered-for-navigation/identitaet/identitaet/{mglId}'
GRPADMIN_GRP = '/nami/gruppierungen-for-grpadmin/parentgruppierung/node/root'
GRUPPIERUNGEN = '/nami/gruppierungen/filtered-for-navigation/gruppierung/node/root'
HISTORY = '/dashboard/history-entries/flist'
INVOICE = '/nami/rechin-for-grpadmin/rechin/gruppierung/'
INVOICE_PDF = '/nami/rechin-for-grpadmin/pdf'
KONFESSION = '/baseadmin/konfession/'
LAND = '/baseadmin/land/'
LOGIN = '/nami/auth/login'
LOGOUT = '/nami/auth/logout'
MGLTYPE = '/nami/enum/mgltype'
MGL_HISTORY = '/nami/mitglied-history/filtered-for-navigation/mitglied/mitglied/'
MGL_HISTORY_EXT = '/nami/mitglied-history-with-values/filtered-for-navigation/mitglied/mitglied/'
MGL_TAETIGKEITEN = '/nami/zugeordnete-taetigkeiten/filtered-for-navigation/gruppierung/gruppierung/{mglId}'
NOTIFICATIONS = '/dashboard/notification-entries/flist'
REGION = '/baseadmin/region/'
SEARCH = '/nami/search-multi/result-list'
SEARCH_ALL = '/nami/mitglied/filtered-for-navigation/gruppierung/gruppierung/{gruppierung}'
SERVER = ''
STAAT = '/baseadmin/staatsangehoerigkeit/'
STATS = '/dashboard/stats/stats'
STATUS_LIST = '/nami/search-multi/status-list'
STUFENWECHSEL = '/mgl-verwaltungS/stufenwechsel'
TAGLIST = '//tagging/getTagList'
TAGS = '/nami/mitglied-tagged-item/filtered-for-navigation/identitaet/identitaet/{mglId}'
TK_AUF_GRP = '//nami/taetigkeittaufgruppierung/filtered/gruppierung/gruppierung/{grpId}'
```



```

TK_CAEA_GRP = '//nami/caea-group/filtered-for-navigation/taetigkeit/taetigkeit/{taetig
TK_GRP = '//nami/gf/gruppierung'
TK_UG = '//nami/untergliederungauftaetigkeit/filtered/untergliederung/taetigkeit/{taet
UNTERGLIEDERUNG = '/orgadmin/untergliederung'
ZAHLUNGSKONDITION = '/baseadmin/zahlungskondition/'

```

```

class UgId
    Bases: enum.Enum
    Untergliederungs ID
    JUFFI = 2
    PFADI = 3
    ROVER = 4
    STAVO = 5
    WOE = 1

```

pynami.nami module

NAMI library in python

This module contains the main class *NaMi* and a few simple exception definitions.

```

class NaMi (config={}, **kwargs)
    Bases: object
    Main class for communication with the DPSG NAMI

```

Example

Listing 3: Connect to the NAMI, search for all active members and print them in a tabulated form.

```

from pynami.nami import NaMi
from pynami.tools import tabulate2x

with NaMi (username='MITGLIEDSNUMMER', password='PASSWORD') as nami:
    print (tabulate2x (nami.search ()))

```

Parameters *config* (dict, optional) – Authorization configuration

__check_response (*response*)

Check a requests response object if the NAMI response looks ok. This currently checks some very basic things.

Raises

- *NamiHTTPError* – When HTTP communication failes
- *NamiResponseSuccessError* – When the NAMI returns an error

__get_baseadmin (*key*, *grpId=None*, *mglId=None*, *taetigkeitId=None*, ***kwargs*)

Base function for retrieving all core lists from the NAMI

Parameters

- **key** (*str*) – Name of the wanted items
- **grpId** (*int* or *str*, optional) – Group id
- **mgId** (*int* or *str*, optional) – Member id (not the DPSG Mitgliedsnummer)
- **taetigkeitId** (*int*, optional) – Id of an activity. A list of all possible ids can be found in the section *Activity types*. This is only required for the URLs which needs to be formatted with this value.

Keyword Arguments

- **gruppierung** (*int* or *str*, optional) – Group id, in case this differs from the group id in the URL (given by *grpId*)
- **mitglied** (*int* or *str*, optional) – Member id (not the DPSG Mitgliedsnummer). This overwrites *mgId*.

Returns The returned default values

Return type *list* of *Baseadmin*

activities

Functional activities

Type *list* of *Baseadmin*

auth (*username=None, password=None*)

Authenticate against the NAMI API. This stores the *jsessionId* cookie in the requests session. Therefore this needs to be called only once.

This also stores your id (not the Mitgliedsnummer) for later pruposes.

Parameters

- **username** (*str*, optional) – The NAMI username. Which is your Mitgliedsnummer
- **password** (*str*, optional) – Your NAMI password

Returns The requests session, including the auth cookie

Return type *requests.Session*

bausteine

Choose DPSG training parts

Type *list* of *Baseadmin*

beitragsarten (*grpId=None, mgId=None*)

Get default values

Parameters

- **grpId** (*int* or *str*, optional) – Group id
- **mgId** (*int* or *str*, optional) – Member id (not the DPSG Mitgliedsnummer)

Returns Each possible fee type

Return type *list* of *Baseadmin*

beitragsarten_mgl (*grpId=None, mgId=None*)

Get default values

Parameters

- **grpId** (*int* or *str*, optional) – Group id
- **mgId** (*int* or *str*, optional) – Member id (not the DPSG Mitgliedsnummer)

Returns Each possible fee for a member

Return type *list* of *Baseadmin*

bescheinigungen (***kwargs*)

Get all certificates of inspection

Returns A list of all your certificates of inspection

Return type *list* of *SearchBescheinigung*

countries (*grpId=None*, *mgId=None*)

Get default values

Parameters

- **grpId** (*int* or *str*, optional) – Group id
- **mgId** (*int* or *str*, optional) – Member id (not the DPSG Mitgliedsnummer)

Returns List of all possible countries with their names and ids

Return type *list* of *Baseadmin*

download_beantragung (***kwargs*)

Open the application form for a certificate of good conduct as a PDF file.

Parameters ***kwargs* – See *open_download_pdf()*.

download_bescheinigung (*id_*, ***kwargs*)

Open a certificate as a PDF file

Parameters

- **id** (*int*) – Internal id of the certificate
- ***kwargs* – See *open_download_pdf()*.

download_invoice (*id_*, ***kwargs*)

Downloads and opens an invoice as a pdf document.

Parameters

- **id** (*int*) – Id of the invoice (not the regular invoice number)
- ***kwargs* – See *open_download_pdf()*.

ebene1

You can choose a Diözese which you belong to.

Type *list* of *Baseadmin*

ebene2 (*ebene1*)

You can choose a Bezirk which you belong to.

Parameters **ebene1** (*int*) – Group id of a Diözese

Returns List of possible Bezirken you are associated with

Return type *list* of *Baseadmin*

ebene3 (*ebene2*)

You can choose a Stamm which you belong to.

Parameters **ebene2** (*int*) – Group id of a Bezirk

Returns List of possible Stämmen you are associated with

Return type `list` of `Baseadmin`

ebenen

Structural layers in the DPSG

Type `list` of `Baseadmin`

geschlechter (`grpId=None, mglId=None`)

Get default values

Parameters

- **grpId** (`int` or `str`, optional) – Group id
- **mglId** (`int` or `str`, optional) – Member id (not the DPSG Mitgliedsnummer)

Returns You can choose between three genders: male, female and diverse.

Return type `list` of `Baseadmin`

get_activity (`mgl, id_`)

Get an activity by its id.

Parameters

- **mgl** (`int`) – Member id (not DPSG Mitgliedsnummer)
- **id** (`int`) – Id of the activity. This will probably originate from an activity search result, e.g. by calling `mgl_activities()`.

Returns The Activity object containing all details.

Return type `Activity`

get_ausbildung (`mglId, id_`)

Get a training by its id.

Parameters

- **mglId** (`int`) – Member id (not DPSG Mitgliedsnummer)
- **id** (`int`) – Id of the training. This will probably originate from an training search result, e.g. by calling `mgl_ausbildungen()`.

Returns The Ausbildung object containing all details about the training.

Return type `Ausbildung`

get_bescheinigung (`id_`)

View a certificate of inspection by its id

Parameters **id** (`int`) – The internal id of the certificate

Returns An object holding all important details about the inspection

Return type `Bescheinigung`

get_mgl_history (`mglId, id_, ext=True`)

Get a member history entry by its id.

Parameters

- **mglId** (`int`) – Member id (not DPSG Mitgliedsnummer)
- **id** (`int`) – Id of the entry. May originate from a search result, e.g. by calling `mgl_history()`.

- **ext** (*bool*, optional) – If the extended history format should be used. Defaults to *True*.

Returns The object containing all vital information about this history entry.

Return type *MitgliedHistory*

get_tag (*mgld*, *tagId*)

Get a tag by its id

Parameters

- **mgld** (*int*) – Member id (not DPSG Mitgliedsnummer)
- **tagId** (*int*) – Tag id

Returns The tag object with all important details

Return type *Tag*

grpId

Group id of the user

Returns *int*

grpadmin_grps

Choose from a list of groups that you are associated with for group admin

Type *list* of *Baseadmin*

gruppierungen

Choose from a list of groups that you are associated with for member admin

Type *list* of *Baseadmin*

history (***kwargs*)

Dashboard function

Returns Last editing events like updating and creating members. In the NAMI these are displayed in the dashboard.

Return type *list* of *HistoryEntry*

invoice (*groupId*, *invId*)

Get an invoice by its id.

Parameters

- **groupId** (*int*) – Group id
- **invId** (*int*) – Id of the invoice. This will probably originate from a search result, e.g. by calling *invoices()*.

Returns The Invoice object containing all details.

Return type *Invoice*

invoices (*groupId=None*, ***kwargs*)

List of all invoices of a group

Parameters **groupId** (*int*, optional) – Group id

Returns All invoices of the specified group

Return type *list* of *SearchInvoice*

konfessionen (*grpId=None*, *mgld=None*)

Get default values

Parameters

- **grpId** (*int* or *str*, optional) – Group id
- **mglId** (*int* or *str*, optional) – Member id (not the DPSG Mitgliedsnummer)

Returns All denominations

Return type *list* of *Baseadmin*

logout ()

This should be called at the end of the communication. It is called when exiting through the `__exit__()` method.

mgl_activities (*mgl*)

List of all activities of a member

Parameters **mgl** (*int*) – Member id (not DPSG Mitgliedsnummer)

Returns All activities of the member (even those which have already ended)

Return type *list* of *Activity*

mgl_ausbildungen (*mglId*)

Get all trainings from a Mitglied.

Parameters **mglId** (*int*) – Member id (not DPSG Mitgliedsnummer)

Returns All trainings of the member

Return type *list* of *SearchAusbildung*

mgl_history (*mglId*, *ext=True*)

Get all history changes from a Mitglied.

Parameters

- **mglId** (*int*) – Member id (not DPSG Mitgliedsnummer)
- **ext** (*bool*, optional) – If the extended history format should be used. Defaults to *True*.

Returns All history entries of the member

Return type *list* of *HistoryEntry*

mgltypes (*grpId=None*, *mglId=None*)

Get default values

Parameters

- **grpId** (*int* or *str*, optional) – Group id
- **mglId** (*int* or *str*, optional) – Member id (not the DPSG Mitgliedsnummer)

Returns You can have one of three different member types

Return type *list* of *Baseadmin*

mitglied (*mglId=None*, *method='GET'*, *grpId=None*, ***kwargs*)

Gets or updates a Mitglied.

The keyword arguments are passed on to the HTTP communication

Parameters

- **mglId** (*int*, optional) – ID of the Mitglied. This is not the DPSG Mitgliedsnummer. Defaults to the user.
- **method** (*str*) – HTTP Method. Should be 'GET' or 'PUT', defaults to 'GET'.

- **grpId** (*int*, optional) – The DPSG Stammesnummer, e.g. 131913. The default (*None*) takes the value from the internal attribute `__config`.

Returns The retrieved or respectively updated Mitglied. Note that the *austrittsDatum* attribute is not part of the returned data set.

Return type *Mitglied*

myId

NAMI internal id of the user

Returns *int*

notifications (*sortproperty=None*, *sortdirection='ASC'*, ***kwargs*)

Dashboard function

Returns All current notifications (like tier changes of members). In the NAMI these are displayed in the dashboard.

Return type *list* of *Notification*

regionen (*grpId=None*, *mgId=None*)

Get default values

Parameters

- **grpId** (*int* or *str*, optional) – Group id
- **mgId** (*int* or *str*, optional) – Member id (not the DPSG Mitgliedsnummer)

Returns All possible Bundeslaender

Return type *list* of *Baseadmin*

search (***kwargs*)

Run a search for members

Todo:

- Check search terms and formatting. Also some search keys can only be used mutually exclusive.
-

Parameters ***kwargs* – Search keys and words. Be advised that some search words must have a certain formatting or can only take a limited amount of values.

Returns The search results

Return type *list* of *SearchMitglied*

See also:

SearchSchema for a complete list of search keys

search_all (*grpId=None*, *filterString=None*, *searchString=""*, *sortproperty=None*, *sortdirection='ASC'*, ***kwargs*)

Search function for filtering the whole member list with limited filter options.

It is also possible to sort the results.

Parameters

- **filterString** (*str*, optional) – Filter attribute.
- **searchString** (*str*, optional) – You can search within the chosen filter attribute. The Format must match the type of the filter attribute.

- **sortproperty** (*str*, optional) – Attribute by which the results shall be sorted.
- **sortdirection** (*str*, optional) – Direction of sorting. Can take the values ASC (which is the default) and DESC.

Returns The search results

Return type *list* of *SearchMitglied*

staaten (*grpId=None, mglId=None*)

Get default values

Parameters

- **grpId** (*int* or *str*, optional) – Group id
- **mglId** (*int* or *str*, optional) – Member id (not the DPSG Mitgliedsnummer)

Returns A quite long list of different nationalities

Return type *list* of *Baseadmin*

stats

Contains counts from different tiers. The actual list of *StatCategory* is in the *statsCategories* attribute.

Type *Stats*

status_list

If you are active, inactive or already deleted

Type *list* of *Baseadmin*

subdivision

Which division you are associated with. This one is only used for searching.

Type *list* of *Baseadmin*

tagList

A different list of fee types but with basically the same content. This one is used for searching members.

Deprecated since version 0.3.3: Only returns an empty list and has therefore become useless.

Type *list* of *Baseadmin*

tags (*mglId, **kwargs*)

Get all tags of a member

Parameters **mglId** (*int*) – Member id (not DPSG Mitgliedsnummer)

Returns List of the search results

Return type *list* of *SearchTag*

tk_auf_grp (*grpId, mglId, **kwargs*)

Get all possible activities for a certain group

Parameters

- **grpId** (*int* or *str*) – Group id
- **mglId** (*int* or *str*) – Member id (not the DPSG Mitgliedsnummer)

Returns List of possible activities

Return type *list* of *Baseadmin*

tk_caea_grp (*grpId*, *mglId*, *taetigkeitId*, ***kwargs*)

Get all possible access rights for an activity

Parameters

- **grpId** (*int* or *str*) – Group id
- **mglId** (*int* or *str*) – Member id (not the DPSG Mitgliedsnummer)
- **taetigkeitId** (*int*) – Id of the activity. A list of all possible ids can be found in the section [Activity types](#).

Returns List of possible access rights

Return type *list* of *Baseadmin*

tk_grp (*grpId*, *mglId*, ***kwargs*)

Get all possible groups for an activity

Parameters

- **grpId** (*int* or *str*) – Group id
- **mglId** (*int* or *str*) – Member id (not the DPSG Mitgliedsnummer)

Returns List of possible groups

Return type *list* of *Baseadmin*

tk_ug (*grpId*, *mglId*, *taetigkeitId*, ***kwargs*)

Get all possible subdivision for an activity

Parameters

- **grpId** (*int* or *str*) – Group id
- **mglId** (*int* or *str*) – Member id (not the DPSG Mitgliedsnummer)
- **taetigkeitId** (*int*) – Id of the activity. A list of all possible ids can be found in the section [Activity types](#).

Returns List of possible subdivision

Return type *list* of *Baseadmin*

update_activity (*mgl*, *act*)

Update an activity

Parameters

- **mgl** (*int*) – Member id (not DPSG Mitgliedsnummer)
- **act** (*Activity*) – Updated data set. The activity id is taken form this data set.

Returns A new updated object

Return type *Activity*

Warning: This has not been tested yet!

update_ausbildung (*mglId*, *ausbildung*)

Update a training

Parameters

- **mgl** (*int*) – Member id (not DPSG Mitgliedsnummer)

- **ausbildung** (*Ausbildung*) – Updated data set. The training id is taken form this data set.

Returns A new updated object

Return type *Ausbildung*

Warning: This has not been tested yet!

zahlungskonditionen (*grpId=None, mglId=None*)

Get default values

Parameters

- **grpId** (*int* or *str*, optional) – Group id
- **mglId** (*int* or *str*, optional) – Member id (not the DPSG Mitgliedsnummer)

Returns Every paying method (either bank transfer or debit)

Return type *list* of *Baseadmin*

exception NamiHTTPError

Bases: *Exception*

Raised when the HTTP status code was not as expected!

exception NamiResponseSuccessError

Bases: *Exception*

This is being raised when the response ‘success’ field is not *True*.

exception NamiResponseTypeError

Bases: *Exception*

This is raised when the response type from the NAMI is not in list of allowed values or more specifically when the NAMI returns an error, a warning or an exception.

pynami.util module

Some utility functions that are used by other classes and methods from this # package but not directly connected to the NAMI.

class ExtractHrefParser (*, *convert_charrefs=True*)

Bases: *html.parser.HTMLParser*

HTML parser for extracting a URL

handle_starttag (*tag, attrs*)

Handles a HTML tag

Parameters

- **tag** (*str*) – The HTML tag
- **attrs** (*list*) – Attributes inside the tag as name and value tuples

url = *None*

URL embedded in HTML tags

Type *str*

extract_url (*htmlstr*)

Extract a URL from a HTML string.

Parameters **htmlstr** (*str*) – The HTML string which will be parsed.

Returns The URL

Return type *str*

open_download_pdf (*content*, *open_file=True*, *save_file=False*, *timeout=10*, *filename=""*)

Open and/or save a downloaded PDF file.

When you only want to open the file: To avoid having to deal with graphical save-the-file dialogues a temporary file inside a temporary directory is created which is deleted after a timeout that is used to open the file. During this timeout the program is blocked.

Parameters

- **content** (*bytes*) – Content of the downloaded file
- **open_file** (*bool*, optional) – Wether to directly open the downloaded file. Defaults to *True*.
- **save_file** (*bool*, optional) – Wether to save the downloaded file to disc. Defaults to *False*.
- **timeout** (*float*, optional) – Time the system has for opening the file. Defaults to 10 seconds.
- **filename** (*str*, optional) – Full path to save file

validate_iban (*value*)

Validate an IBAN

Parameters **value** (*str*) – Value to check. Spaces are allowed.

Raises *ValidationError* – In case of invalid IBAN.

Returns The IBAN in compact form.

Return type *str*

pynami.tools module

Some utility functions for user convenience

export_xlsx (*data*, *attrs=None*, *includeheader=True*, *tableName='Tabelle1'*, *sheetName='Data'*, *write_to_file=False*, *filepath=""*)

Create a Microsoft Excel Wokbook from a given dataset. The data can optionally be saved to a file.

Parameters

- **data** (*list*) – Data objects. They should all belong to the same class.
- **attrs** (*list* of *str*, optional) – Attribute names for the CSV table. If left empty (*None*) the value of the first *_tabkeys* attribute in the list is taken.
- **includeheader** (*bool*, optional) – Whether to include headers in the output. Defaults to *True*.
- **tableName** (*str*, optional) – Name of the table. Defaults to *'Tabelle1'*.
- **sheetName** (*str*, optional) – Name of the worksheet. Defaults to *'Data'*.

- **write_to_file** (*bool*, optional) – If the workbook should be saved to a file. Defaults to *False*.
- **filepath** (*str*, optional) – Full path to the Excel file where the data should be saved. If left empty the default savefile dialog will be invoked via the module *tkinter*.

Returns The created workbook.

Return type *Workbook*

make_csv (*data*, *attrs=None*, *includeheader=True*, *delimiter=', '*)

Makes a CSV formatted string from a data set

Parameters

- **data** (*list*) – Data objects. They should all belong to the same class.
- **attrs** (*list* of *str*, optional) – Attribute names for the CSV table. If left empty (*None*) the value of the first *_tabkeys* attribute in the list is taken.
- **includeheader** (*bool*, optional) – Whether to include headers in the output. Defaults to *True*.

Returns CSV formatted data

Return type *str*

send_emails (*mitglieder*, *to=*”, *method='bcc'*, *email1=True*, *email2=True*, *open_browser=True*)

Send emails to several members.

Parameters

- **mitglieder** (*list*) – The List contents can be either *SearchMitglied* or *Mitglied*
- **to** (*str*, optional) – Primary recipient
- **method** (*str*, optional) – If you want to send your mails as bcc or something else. Currently only bcc is supported.
- **email1** (*bool*, optional) – If emails should be send to the primary address of the members.
- **email2** (*bool*, optional) – If emails should be send to the email account of the member’s parent.
- **open_browser** (*bool*, optional) – If *True* the link is opened directly by the system. On a computer this may open your default mail program.

Returns The mailto link

Return type *str*

tabulate2x (*objs*, *elements=None*)

Tabulate a list of objects by tabulating each object first

Parameters

- **obj** (*list*) – The list of objects to tabulate. If they are not from the same class this may not work.
- **elements** (*list* of *str*, optional) – List of keys which should be displayed

Returns Nicely formatted tabulated output

Return type *str*

8.1 Version 0.3.3 (14.05.2023)

- Added this changelog
- Added function for Excel export
- Reorganized doc files
- Automated table generation of constants
- Updated tables of constants
- Fixed rare bug where received dates would not be converted and lead to an error

CHAPTER 9

Indices and tables

- `genindex`
- `modindex`
- `search`

p

- `pynami`, [49](#)
- `pynami.constants`, [83](#)
- `pynami.nami`, [85](#)
- `pynami.schemas`, [49](#)
- `pynami.schemas.activity`, [49](#)
- `pynami.schemas.base`, [52](#)
- `pynami.schemas.cogc`, [55](#)
- `pynami.schemas.dashboard`, [57](#)
- `pynami.schemas.default`, [59](#)
- `pynami.schemas.grpadmin`, [60](#)
- `pynami.schemas.history`, [64](#)
- `pynami.schemas.mgl`, [67](#)
- `pynami.schemas.search`, [76](#)
- `pynami.schemas.tags`, [82](#)
- `pynami.schemas.training`, [80](#)
- `pynami.tools`, [95](#)
- `pynami.util`, [94](#)

HTTP Routing Table

/baseadmin

GET /baseadmin/{key}/, 41

/dashboard

GET /dashboard/history-entries/flist,

42

GET /dashboard/notification-entries/flist,

41

GET /dashboard/stats/stats, 43

/module

GET /module/baustein, 28

/nami

GET /nami/auth/logout, 35

GET /nami/enum/mgltype, 26

GET /nami/mitglied/filtered-for-navigation/gruppierung/gruppierung/{groupId}/flist,

35

GET /nami/mitglied/filtered-for-navigation/gruppierung/gruppierung/{groupId}/{mglId},

36

GET /nami/search-multi/all-visible-taetigkeiten,

31

GET /nami/search-multi/result-list, 36

GET /nami/search-multi/status-list, 27

GET /nami/zugeordnete-taetigkeiten/filtered-for-navigation/gruppierung-mitglied/mitglied/{n

39

GET /nami/zugeordnete-taetigkeiten/filtered-for-navigation/gruppierung-mitglied/mitglied/{n

39

POST /nami/auth/manual/sessionStartup,

35

PUT /nami/mitglied/filtered-for-navigation/gruppierung/gruppierung/{groupId}/{mglId},

37

PUT /nami/zugeordnete-taetigkeiten/filtered-for-navigation/gruppierung-mitglied/mitglied/{n

40

/orgadmin

GET /orgadmin/untergliederung, 29

/tagging

GET //tagging/getTagList, 25

Symbols

__getstate__() (*BaseModel* method), 52
 __model__ (*ActivitySchema* attribute), 49
 __model__ (*AusbildungSchema* attribute), 80
 __model__ (*BaseSchema* attribute), 53
 __model__ (*BaseSearchSchema* attribute), 54
 __model__ (*BaseadminSchema* attribute), 59
 __model__ (*BescheinigungSchema* attribute), 55
 __model__ (*HistoryEntrySchema* attribute), 65
 __model__ (*InvoiceSchema* attribute), 60
 __model__ (*MitgliedHistorySchema* attribute), 66
 __model__ (*MitgliedSchema* attribute), 67
 __model__ (*NamiKontoSchema* attribute), 72
 __model__ (*NotificationSchema* attribute), 57
 __model__ (*SearchActivitySchema* attribute), 51
 __model__ (*SearchAusbildungSchema* attribute), 81
 __model__ (*SearchBescheinigungSchema* attribute), 56
 __model__ (*SearchInvoiceSchema* attribute), 62
 __model__ (*SearchMitgliedSchema* attribute), 73
 __model__ (*SearchSchema* attribute), 46
 __model__ (*SearchTagSchema* attribute), 82
 __model__ (*StatCatSchema* attribute), 58
 __model__ (*StatsSchema* attribute), 59
 __model__ (*TagSchema* attribute), 82
 __setattr__() (*BaseModel* method), 52
 __setstate__() (*BaseModel* method), 52
 _check_response() (*NaMi* method), 7, 85
 _declared_fields (*BaseSchema* attribute), 53
 _declared_fields (*BaseSearchSchema* attribute), 54
 _field_blacklist (*BaseModel* attribute), 52
 _get_baseadmin() (*NaMi* method), 7, 85
 _hooks (*BaseSchema* attribute), 53
 _hooks (*BaseSearchSchema* attribute), 54
 _tabkeys (*BaseModel* attribute), 52
 _tabkeys (*BaseSearchModel* attribute), 54

A

AccessError, 52
 activities (*NaMi* attribute), 8, 86
 Activity (class in *pynami.schemas.activity*), 49
 ActivitySchema (class in *pynami.schemas.activity*), 49
 actor (*MitgliedHistorySchema* attribute), 66
 aktivBis (*ActivitySchema* attribute), 49
 aktivVon (*ActivitySchema* attribute), 50
 ALLE_TAETIGKEITEN (*URLS* attribute), 83
 alterBis (*SearchSchema* attribute), 46, 77
 alterVon (*SearchSchema* attribute), 46, 77
 Ausbildung (class in *pynami.schemas.training*), 80
 AUSBILDUNG (*URLS* attribute), 83
 AusbildungSchema (class in *pynami.schemas.training*), 80
 austrittsDatum (*MitgliedSchema* attribute), 67
 AUTH (*URLS* attribute), 83
 auth() (*NaMi* method), 8, 86
 autor (*BescheinigungSchema* attribute), 55

B

bankleitzahl (*NamiKontoSchema* attribute), 72
 BASE_URL (in module *pynami.constants*), 83
 Baseadmin (class in *pynami.schemas.default*), 59
 BaseadminSchema (class in *pynami.schemas.default*), 59
 BaseModel (class in *pynami.schemas.base*), 52
 BaseSchema (class in *pynami.schemas.base*), 53
 BaseSchema.Meta (class in *pynami.schemas.base*), 53
 BaseSearchModel (class in *pynami.schemas.base*), 54
 BaseSearchSchema (class in *pynami.schemas.base*), 54
 baustein (*AusbildungSchema* attribute), 80
 BAUSTEIN (*URLS* attribute), 83
 bausteine (*NaMi* attribute), 8, 86
 bausteinId (*AusbildungSchema* attribute), 80

bausteinIncludeId (*SearchSchema attribute*), 47, 77
BEANTRAGUNG (*URLS attribute*), 84
beitragsart (*MitgliedSchema attribute*), 67
BEITRAGSART (*URLS attribute*), 84
BEITRAGSART_MGL (*URLS attribute*), 84
beitragsarten() (*NaMi method*), 8, 86
beitragsarten_mgl() (*NaMi method*), 8, 86
beitragsArtId (*ActivitySchema attribute*), 50
beitragsartId (*MitgliedSchema attribute*), 67
Bescheinigung (*class in pynami.schemas.cogc*), 55
bescheinigungen() (*NaMi method*), 9, 87
BescheinigungSchema (*class in pynami.schemas.cogc*), 55
bic (*NamiKontoSchema attribute*), 72

C

caeaGroup (*ActivitySchema attribute*), 50
caeaGroupForGf (*ActivitySchema attribute*), 50
caeaGroupForGfId (*ActivitySchema attribute*), 50
caeaGroupId (*ActivitySchema attribute*), 50
changedFields (*MitgliedHistorySchema attribute*), 66
correct_spelling() (*SearchSchema method*), 77
correctEmptyStrings() (*BaseSchema method*), 53
count (*StatCatSchema attribute*), 58
countries() (*NaMi method*), 9, 87
currency (*InvoiceSchema attribute*), 60

D

dateformat (*BaseSchema.Meta attribute*), 53
datetimeformat (*BaseSchema.Meta attribute*), 53
datumEinsicht (*BescheinigungSchema attribute*), 55
debitor (*InvoiceSchema attribute*), 60
debitor_document_id (*InvoiceSchema attribute*), 60
debitorKonto (*InvoiceSchema attribute*), 60
debitorName (*InvoiceSchema attribute*), 60
DEFAULT_PARAMS (*in module pynami.constants*), 83
descriptor (*BaseSearchSchema attribute*), 54
displayName (*InvoiceSchema attribute*), 61
double_dump() (*NamiKontoSchema method*), 72
download (*BescheinigungSchema attribute*), 55
download() (*Invoice method*), 60
download() (*SearchInvoice method*), 62
download_beantragung() (*NaMi method*), 9, 87
download_bescheinigung() (*NaMi method*), 9, 87
download_fz() (*Bescheinigung method*), 55
download_fz() (*SearchBescheinigung method*), 56
download_invoice() (*NaMi method*), 9, 87

E

EBENE (*URLS attribute*), 84
ebene1 (*NaMi attribute*), 9, 87
EBENE1 (*URLS attribute*), 84
EBENE2 (*URLS attribute*), 84
ebene2() (*NaMi method*), 9, 87
EBENE3 (*URLS attribute*), 84
ebene3() (*NaMi method*), 9, 87
ebeneId (*SearchSchema attribute*), 47, 77
ebenen (*NaMi attribute*), 9, 88
eintrittsdatum (*MitgliedSchema attribute*), 67
einzugsDatum (*InvoiceSchema attribute*), 61
email (*MitgliedSchema attribute*), 68
emailVertretungsberechtigter (*MitgliedSchema attribute*), 68
empfaenger (*BescheinigungSchema attribute*), 55
entries_actor (*HistoryEntrySchema attribute*), 65
entries_actor (*NotificationSchema attribute*), 57
entries_actorId (*HistoryEntrySchema attribute*), 65
entries_actorId (*NotificationSchema attribute*), 57
entries_aktivBis (*SearchActivitySchema attribute*), 51
entries_aktivVon (*SearchActivitySchema attribute*), 51
entries_anlagedatum (*SearchActivitySchema attribute*), 51
entries_austrittsDatum (*SearchMitgliedSchema attribute*), 73
entries_author (*HistoryEntrySchema attribute*), 65
entries_autor (*SearchBescheinigungSchema attribute*), 56
entries_baustein (*SearchAusbildungSchema attribute*), 81
entries_beitragsArt (*SearchActivitySchema attribute*), 51
entries_beitragsarten (*SearchMitgliedSchema attribute*), 73
entries_caeaGroup (*SearchActivitySchema attribute*), 51
entries_caeaGroupForGf (*SearchActivitySchema attribute*), 51
entries_changedFields (*HistoryEntrySchema attribute*), 65
entries_changedFields (*NotificationSchema attribute*), 58
entries_completeChanges (*HistoryEntrySchema attribute*), 65
entries_completeChanges (*NotificationSchema attribute*), 58
entries_datumEinsicht (*SearchBescheinigungSchema attribute*), 56
entries_debitor (*SearchInvoiceSchema attribute*), 62

entries_debitor_document_id (*SearchInvoiceSchema attribute*), 63
 entries_debitorId (*SearchInvoiceSchema attribute*), 62
 entries_debitorName (*SearchInvoiceSchema attribute*), 62
 entries_debitorType (*SearchInvoiceSchema attribute*), 63
 entries_displayName (*SearchInvoiceSchema attribute*), 63
 entries_eintrittsdatum (*SearchMitgliedSchema attribute*), 73
 entries_einzugsDatum (*SearchInvoiceSchema attribute*), 63
 entries_email (*SearchMitgliedSchema attribute*), 74
 entries_emailVertretungsberechtigter (*SearchMitgliedSchema attribute*), 74
 entries_empfaenger (*SearchBescheinigungSchema attribute*), 57
 entries_empfGebDatum (*SearchBescheinigungSchema attribute*), 56
 entries_empfNachname (*SearchBescheinigungSchema attribute*), 57
 entries_empfVorname (*SearchBescheinigungSchema attribute*), 57
 entries_entryDate (*HistoryEntrySchema attribute*), 65
 entries_entryDate (*NotificationSchema attribute*), 58
 entries_erloesKtoName (*SearchInvoiceSchema attribute*), 63
 entries_erstelltAm (*SearchBescheinigungSchema attribute*), 57
 entries_ersteTaetigkeitId (*SearchMitgliedSchema attribute*), 74
 entries_ersteUntergliederungId (*SearchMitgliedSchema attribute*), 74
 entries_fibuDebitorKonto (*SearchInvoiceSchema attribute*), 63
 entries_fibuErloesKonto (*SearchInvoiceSchema attribute*), 63
 entries_fixBeitrag (*SearchMitgliedSchema attribute*), 74
 entries_freigabeDatum (*SearchInvoiceSchema attribute*), 63
 entries_fzDatum (*SearchBescheinigungSchema attribute*), 57
 entries_fzNummer (*SearchBescheinigungSchema attribute*), 57
 entries_geburtsDatum (*SearchMitgliedSchema attribute*), 74
 entries_genericField1 (*SearchMitgliedSchema attribute*), 74
 entries_genericField2 (*SearchMitgliedSchema attribute*), 74
 entries_geschlecht (*SearchMitgliedSchema attribute*), 74
 entries_gruppierung (*HistoryEntrySchema attribute*), 65
 entries_gruppierung (*SearchActivitySchema attribute*), 51
 entries_gruppierung (*SearchMitgliedSchema attribute*), 74
 entries_gruppierungId (*SearchMitgliedSchema attribute*), 74
 entries_id (*HistoryEntrySchema attribute*), 65
 entries_id (*NotificationSchema attribute*), 58
 entries_id (*SearchAusbildungSchema attribute*), 81
 entries_id (*SearchMitgliedSchema attribute*), 74
 entries_identitaet (*SearchTagSchema attribute*), 82
 entries_jungpfadfinder (*SearchMitgliedSchema attribute*), 74
 entries_konfession (*SearchMitgliedSchema attribute*), 74
 entries_kontoOwnerId (*SearchInvoiceSchema attribute*), 63
 entries_kontoOwnerTyp (*SearchInvoiceSchema attribute*), 63
 entries_kontoverbindung (*SearchInvoiceSchema attribute*), 63
 entries_kontoverbindung (*SearchMitgliedSchema attribute*), 75
 entries_kreditor (*SearchInvoiceSchema attribute*), 63
 entries_lastUpdated (*SearchMitgliedSchema attribute*), 75
 entries_mglType (*SearchMitgliedSchema attribute*), 75
 entries_mitglied (*HistoryEntrySchema attribute*), 65
 entries_mitglied (*SearchActivitySchema attribute*), 52
 entries_mitglied (*SearchAusbildungSchema attribute*), 81
 entries_mitgliedsNummer (*SearchMitgliedSchema attribute*), 75
 entries_nachname (*SearchMitgliedSchema attribute*), 75
 entries_newObject (*HistoryEntrySchema attribute*), 65
 entries_newObject (*NotificationSchema attribute*), 58
 entries_objectClass (*HistoryEntrySchema attribute*), 66
 entries_objectClass (*NotificationSchema attribute*), 58
 entries_objectId (*HistoryEntrySchema attribute*),

66
 entries_objectId (*NotificationSchema attribute*),
 58
 entries_operation (*HistoryEntrySchema attribute*), 66
 entries_operation (*NotificationSchema attribute*),
 58
 entries_originalObject (*HistoryEntrySchema attribute*), 66
 entries_originalObject (*NotificationSchema attribute*), 58
 entries_pfadfinder (*SearchMitgliedSchema attribute*), 75
 entries_rechnungsEmpfaenger (*SearchInvoiceSchema attribute*), 64
 entries_rechnungsLauf (*SearchInvoiceSchema attribute*), 64
 entries_reCreated (*SearchInvoiceSchema attribute*), 63
 entries_reDatum (*SearchInvoiceSchema attribute*),
 64
 entries_reMwst (*SearchInvoiceSchema attribute*),
 64
 entries_reNetto (*SearchInvoiceSchema attribute*),
 64
 entries_reNr (*SearchInvoiceSchema attribute*), 64
 entries_rover (*SearchMitgliedSchema attribute*), 75
 entries_rowCssClass (*SearchMitgliedSchema attribute*), 75
 entries_spitzname (*SearchMitgliedSchema attribute*), 75
 entries_staatangehoerigkeitText (*SearchMitgliedSchema attribute*), 75
 entries_staatsangehoerigkeit (*SearchMitgliedSchema attribute*), 75
 entries_status (*SearchInvoiceSchema attribute*),
 64
 entries_status (*SearchMitgliedSchema attribute*),
 75
 entries_stufe (*SearchMitgliedSchema attribute*), 75
 entries_taetigkeit (*SearchActivitySchema attribute*), 52
 entries_tag (*SearchTagSchema attribute*), 82
 entries_telefax (*SearchMitgliedSchema attribute*),
 76
 entries_telefon1 (*SearchMitgliedSchema attribute*), 76
 entries_telefon2 (*SearchMitgliedSchema attribute*), 76
 entries_telefon3 (*SearchMitgliedSchema attribute*), 76
 entries_untergliederung (*SearchActivitySchema attribute*), 52
 entries_veranstalter (*SearchAusbildungSchema attribute*), 81
 entries_version (*SearchMitgliedSchema attribute*),
 76
 entries_vorname (*SearchMitgliedSchema attribute*),
 76
 entries_vstgName (*SearchAusbildungSchema attribute*), 81
 entries_vstgTag (*SearchAusbildungSchema attribute*), 81
 entries_wiederverwendenFlag (*SearchMitgliedSchema attribute*), 76
 entries_woelfling (*SearchMitgliedSchema attribute*), 76
 entries_zahlungsEmpfaenger (*SearchInvoiceSchema attribute*), 64
 entryDate (*MitgliedHistorySchema attribute*), 66
 erloesKtoName (*InvoiceSchema attribute*), 61
 erstelltAm (*BescheinigungSchema attribute*), 55
 ersteTaetigkeit (*MitgliedSchema attribute*), 68
 ersteTaetigkeitId (*MitgliedSchema attribute*), 68
 ersteUntergliederung (*MitgliedSchema attribute*), 68
 ersteUntergliederungId (*MitgliedSchema attribute*), 68
 export_xlsx() (in module *pynami.tools*), 95
 extract_url() (in module *pynami.util*), 94
 ExtractHrefParser (class in *pynami.util*), 94

F

fixBeitrag (*MitgliedSchema attribute*), 68
 freigabeDatum (*InvoiceSchema attribute*), 61
 funktion (*SearchSchema attribute*), 46, 77
 FZ (*URLS attribute*), 84
 fzDatum (*BescheinigungSchema attribute*), 55
 fzNummer (*BescheinigungSchema attribute*), 55

G

geburtsDatum (*MitgliedSchema attribute*), 68
 genericField1 (*MitgliedSchema attribute*), 68
 genericField2 (*MitgliedSchema attribute*), 68
 geschlecht (*MitgliedSchema attribute*), 68
 GESCHLECHT (*URLS attribute*), 84
 geschlechter() (*NaMi method*), 10, 88
 geschlechtId (*MitgliedSchema attribute*), 68
 get_activity() (*NaMi method*), 10, 88
 get_activity() (*SearchActivity method*), 51
 get_ausbildung() (*NaMi method*), 10, 88
 get_ausbildung() (*SearchAusbildung method*), 81
 get_bescheinigung() (*NaMi method*), 10, 88
 get_bescheinigung() (*SearchBescheinigung method*), 56
 get_download_url() (*BescheinigungSchema method*), 56
 get_download_url() (*InvoiceSchema method*), 61

get_history() (*HistoryEntry* method), 64
 get_invoice() (*SearchInvoice* method), 62
 get_mgl_history() (*NaMi* method), 10, 88
 get_mitglied() (*SearchMitglied* method), 73
 GET_TAG (*URLS* attribute), 84
 get_tag() (*NaMi* method), 11, 89
 GETMGL (*URLS* attribute), 84
 grpadmin_grps (*NaMi* attribute), 11, 89
 GRPADMIN_GRPES (*URLS* attribute), 84
 grpId (*NaMi* attribute), 11, 89
 grpName (*SearchSchema* attribute), 47, 78
 grpNummer (*SearchSchema* attribute), 47, 78
 gruppierung (*ActivitySchema* attribute), 50
 gruppierung (*MitgliedHistorySchema* attribute), 66
 gruppierung (*MitgliedSchema* attribute), 68
 gruppierung1Id (*SearchSchema* attribute), 48, 78
 gruppierung2Id (*SearchSchema* attribute), 48, 78
 gruppierung3Id (*SearchSchema* attribute), 48, 78
 gruppierung4Id (*SearchSchema* attribute), 48, 78
 gruppierung5Id (*SearchSchema* attribute), 48, 78
 gruppierung6Id (*SearchSchema* attribute), 48, 78
 gruppierungen (*NaMi* attribute), 11, 89
 GRUPPIERUNGEN (*URLS* attribute), 84
 gruppierungId (*ActivitySchema* attribute), 50
 gruppierungId (*MitgliedSchema* attribute), 68

H

handle_starttag() (*ExtractHrefParser* method), 94
 HISTORY (*URLS* attribute), 84
 history() (*NaMi* method), 11, 89
 HistoryEntry (*class* in *pynami.schemas.history*), 64
 HistoryEntrySchema (*class* in *pynami.schemas.history*), 65

I

iban (*NamiKontoSchema* attribute), 72
 id (*ActivitySchema* attribute), 50
 id (*AusbildungSchema* attribute), 80
 id (*BaseSearchSchema* attribute), 54
 id (*BescheinigungSchema* attribute), 56
 id (*InvoiceSchema* attribute), 61
 id (*MitgliedHistorySchema* attribute), 66
 id (*MitgliedSchema* attribute), 69
 id (*NamiKontoSchema* attribute), 72
 id (*TagSchema* attribute), 82
 id_to_str() (*NamiKontoSchema* method), 72
 inGrp (*SearchSchema* attribute), 48, 78
 institut (*NamiKontoSchema* attribute), 72
 Invoice (*class* in *pynami.schemas.grpadmin*), 60
 INVOICE (*URLS* attribute), 84
 invoice() (*NaMi* method), 11, 89
 INVOICE_PDF (*URLS* attribute), 84
 invoices() (*NaMi* method), 11, 89

InvoiceSchema (*class* in *pynami.schemas.grpadmin*), 60

J

JUFFI (*UgId* attribute), 85
 jungpfadfinder (*MitgliedSchema* attribute), 69

K

konfession (*MitgliedSchema* attribute), 69
 KONFESSION (*URLS* attribute), 84
 konfessionen() (*NaMi* method), 11, 89
 konfessionId (*MitgliedSchema* attribute), 69
 kontoinhaber (*NamiKontoSchema* attribute), 72
 kontonummer (*NamiKontoSchema* attribute), 73
 kontoverbindung (*MitgliedSchema* attribute), 69

L

land (*MitgliedSchema* attribute), 69
 LAND (*URLS* attribute), 84
 landId (*MitgliedSchema* attribute), 69
 lastModifiedFrom (*AusbildungSchema* attribute), 80
 lastUpdated (*MitgliedSchema* attribute), 69
 LOGOUT (*URLS* attribute), 84
 logout() (*NaMi* method), 12, 90

M

make_csv() (*in module pynami.tools*), 96
 make_object() (*BaseSchema* method), 54
 mgl_activities() (*NaMi* method), 12, 90
 mgl_ausbildungen() (*NaMi* method), 12, 90
 MGL_HISTORY (*URLS* attribute), 84
 mgl_history() (*NaMi* method), 12, 90
 MGL_HISTORY_EXT (*URLS* attribute), 84
 MGL_TAETIGKEITEN (*URLS* attribute), 84
 mglStatusId (*SearchSchema* attribute), 46, 78
 mglType (*MitgliedSchema* attribute), 69
 MGLTYPE (*URLS* attribute), 84
 mglTypeId (*MitgliedSchema* attribute), 69
 mglTypeId (*SearchSchema* attribute), 46, 78
 mgltypes() (*NaMi* method), 12, 90
 mglWohnort (*SearchSchema* attribute), 46, 78
 mitAllenTaetigkeiten (*SearchSchema* attribute), 47, 78
 mitglied (*AusbildungSchema* attribute), 80
 Mitglied (*class* in *pynami.schemas.mgl*), 67
 mitglied (*TagSchema* attribute), 82
 mitglied() (*NaMi* method), 12, 90
 MitgliedHistory (*class* in *pynami.schemas.history*), 66
 MitgliedHistorySchema (*class* in *pynami.schemas.history*), 66
 mitgliedId (*TagSchema* attribute), 82
 MitgliedSchema (*class* in *pynami.schemas.mgl*), 67

MitgliedSchema.Meta (class in py-nami.schemas.mgl), 67
 mitgliedsNummer (MitgliedSchema attribute), 69
 mitgliedsNummer (NamiKontoSchema attribute), 73
 mitgliedsNummer (SearchSchema attribute), 46, 79
 myId (NaMi attribute), 13, 91

N

nachname (MitgliedSchema attribute), 69
 nachname (SearchSchema attribute), 46, 79
 name (BaseadminSchema attribute), 60
 name (StatCatSchema attribute), 59
 nameZusatz (MitgliedSchema attribute), 70
 NaMi (class in pynami.nami), 7, 85
 NamiHTTPError, 94
 NamiKonto (class in pynami.schemas.mgl), 71
 NamiKontoSchema (class in pynami.schemas.mgl), 72
 NamiResponseSuccessError, 94
 NamiResponseTypeError, 94
 Notification (class in pynami.schemas.dashboard), 57
 NOTIFICATIONS (URLS attribute), 84
 notifications () (NaMi method), 13, 91
 NotificationSchema (class in py-nami.schemas.dashboard), 57
 nrMitglieder (StatsSchema attribute), 59

O

open_download_pdf () (in module pynami.util), 95
 operation (MitgliedHistorySchema attribute), 66
 ordered (MitgliedSchema.Meta attribute), 67
 ordered (SearchSchema.Meta attribute), 77
 organisation (SearchSchema attribute), 46, 79
 ort (MitgliedSchema attribute), 70

P

pdf (InvoiceSchema attribute), 61
 pfadfinder (MitgliedSchema attribute), 70
 PFADI (UgId attribute), 85
 plz (MitgliedSchema attribute), 70
 privacy (SearchSchema attribute), 48, 79
 pynami (module), 49
 pynami.constants (module), 83
 pynami.nami (module), 85
 pynami.schemas (module), 49
 pynami.schemas.activity (module), 49
 pynami.schemas.base (module), 52
 pynami.schemas.cogc (module), 55
 pynami.schemas.dashboard (module), 57
 pynami.schemas.default (module), 59
 pynami.schemas.grpadmin (module), 60
 pynami.schemas.history (module), 64
 pynami.schemas.mgl (module), 67
 pynami.schemas.search (module), 76

pynami.schemas.tags (module), 82
 pynami.schemas.training (module), 80
 pynami.tools (module), 95
 pynami.util (module), 94

R

rechnungsLauf (InvoiceSchema attribute), 61
 reCreated (InvoiceSchema attribute), 61
 reDatum (InvoiceSchema attribute), 61
 region (MitgliedSchema attribute), 70
 REGION (URLS attribute), 84
 regionen () (NaMi method), 13, 91
 regionId (MitgliedSchema attribute), 70
 reNr (InvoiceSchema attribute), 61
 representedClass (BaseSearchSchema attribute), 54
 rover (MitgliedSchema attribute), 70
 ROVER (UgId attribute), 85

S

SEARCH (URLS attribute), 84
 search () (NaMi method), 13, 91
 SEARCH_ALL (URLS attribute), 84
 search_all () (NaMi method), 13, 91
 SearchActivity (class in pynami.schemas.activity), 50
 SearchActivitySchema (class in py-nami.schemas.activity), 51
 SearchAusbildung (class in py-nami.schemas.training), 81
 SearchAusbildungSchema (class in py-nami.schemas.training), 81
 SearchBescheinigung (class in py-nami.schemas.cogc), 56
 SearchBescheinigungSchema (class in py-nami.schemas.cogc), 56
 SearchInvoice (class in pynami.schemas.grpadmin), 62
 SearchInvoiceSchema (class in py-nami.schemas.grpadmin), 62
 SearchMitglied (class in pynami.schemas.mgl), 73
 SearchMitgliedSchema (class in py-nami.schemas.mgl), 73
 searchName (SearchSchema attribute), 48, 79
 SearchSchema (class in pynami.schemas.search), 45, 76
 SearchSchema.Meta (class in py-nami.schemas.search), 77
 SearchTag (class in pynami.schemas.tags), 82
 SearchTagSchema (class in pynami.schemas.tags), 82
 searchType (SearchSchema attribute), 48, 79
 send_emails () (in module pynami.tools), 96
 SERVER (URLS attribute), 84
 sonst01 (MitgliedSchema attribute), 70

sonst02 (*MitgliedSchema* attribute), 70
 spitzname (*MitgliedSchema* attribute), 70
 spitzname (*SearchSchema* attribute), 47, 79
 STAAT (*URLS* attribute), 84
 staaten() (*NaMi* method), 14, 92
 staatsangehoerigkeit (*MitgliedSchema* attribute), 70
 staatsangehoerigkeitId (*MitgliedSchema* attribute), 70
 staatsangehoerigkeitText (*MitgliedSchema* attribute), 70
 StatCategory (class in *pynami.schemas.dashboard*), 59
 StatCatSchema (class in *pynami.schemas.dashboard*), 58
 Stats (class in *pynami.schemas.dashboard*), 59
 stats (*NaMi* attribute), 14, 92
 STATS (*URLS* attribute), 84
 statsCategories (*StatsSchema* attribute), 59
 StatsSchema (class in *pynami.schemas.dashboard*), 59
 status (*InvoiceSchema* attribute), 61
 status (*MitgliedSchema* attribute), 71
 status_list (*NaMi* attribute), 14, 92
 STATUS_LIST (*URLS* attribute), 84
 STAVO (*UgId* attribute), 85
 strasse (*MitgliedSchema* attribute), 71
 stufe (*MitgliedSchema* attribute), 71
 STUFENWECHSEL (*URLS* attribute), 84
 subdivision (*NaMi* attribute), 14, 92

T

table_view() (*BaseModel* method), 53
 tabulate() (*BaseModel* method), 53
 tabulate2x() (in module *pynami.tools*), 96
 taetigkeit (*ActivitySchema* attribute), 50
 taetigkeitId (*ActivitySchema* attribute), 50
 taetigkeitId (*SearchSchema* attribute), 47, 79
 Tag (class in *pynami.schemas.tags*), 82
 tag (*TagSchema* attribute), 82
 tagId (*SearchSchema* attribute), 46, 79
 tagId (*TagSchema* attribute), 83
 tagList (*NaMi* attribute), 14, 92
 TAGLIST (*URLS* attribute), 84
 TAGS (*URLS* attribute), 84
 tags() (*NaMi* method), 14, 92
 TagSchema (class in *pynami.schemas.tags*), 82
 telefax (*MitgliedSchema* attribute), 71
 telefon1 (*MitgliedSchema* attribute), 71
 telefon2 (*MitgliedSchema* attribute), 71
 telefon3 (*MitgliedSchema* attribute), 71
 TK_AUF_GRP (*URLS* attribute), 84
 tk_auf_grp() (*NaMi* method), 14, 92
 TK_CAEA_GRP (*URLS* attribute), 84

tk_caea_grp() (*NaMi* method), 14, 92
 TK_GRP (*URLS* attribute), 85
 tk_grp() (*NaMi* method), 15, 93
 TK_UG (*URLS* attribute), 85
 tk_ug() (*NaMi* method), 15, 93
 total (*InvoiceSchema* attribute), 61
 type (*BaseSearchModel* attribute), 54

U

UgId (class in *pynami.constants*), 85
 untergliederung (*ActivitySchema* attribute), 50
 UNTERGLIEDERUNG (*URLS* attribute), 85
 untergliederungId (*ActivitySchema* attribute), 50
 untergliederungId (*SearchSchema* attribute), 47, 79
 unterhalbGrp (*SearchSchema* attribute), 48, 79
 update() (*Mitglied* method), 67
 update_activity() (*NaMi* method), 15, 93
 update_ausbildung() (*NaMi* method), 15, 93
 url (*ExtractHrefParser* attribute), 94
 URLMetaClass (class in *pynami.constants*), 83
 URLs (class in *pynami.constants*), 83

V

validate_iban() (in module *pynami.util*), 95
 veranstalter (*AusbildungSchema* attribute), 80
 version (*MitgliedSchema* attribute), 71
 vorname (*MitgliedSchema* attribute), 71
 vorname (*SearchSchema* attribute), 46, 79
 vstgName (*AusbildungSchema* attribute), 80
 vstgTag (*AusbildungSchema* attribute), 80

W

wiederverwendenFlag (*MitgliedSchema* attribute), 71
 withEndedTaetigkeiten (*SearchSchema* attribute), 47, 79
 WOE (*UgId* attribute), 85
 woelfling (*MitgliedSchema* attribute), 71

Z

zahlungskondition (*NamiKontoSchema* attribute), 73
 ZAHLUNGSKONDITION (*URLS* attribute), 85
 zahlungskonditionen() (*NaMi* method), 16, 94
 zahlungskonditionId (*NamiKontoSchema* attribute), 73
 zahlungsweise (*InvoiceSchema* attribute), 62
 zahlungsweiseId (*InvoiceSchema* attribute), 62
 zeitschriftenversand (*MitgliedSchema* attribute), 71
 zeitschriftenversand (*SearchSchema* attribute), 47, 79