



New Membership API

Version 2.5.0

This documentation is meant to be your technical guide for using the TK New Membership API.

The web services of this API can be used to automatically

- submit new customers to the TK health insurance
- change the address of a previously committed membership application

Please contact the TK sales team for an active cooperation status to use this API.



New Membership API

Contents

Release Notes	3
Overview.....	6
API Specification	7
Credentials.....	7
Code Examples	7
Operation /neuaufnahmeantrag/getApiAccessToken	7
Authentication and authorization	7
Sample Request.....	8
Operation /einreichen.....	10
Upload of documents	11
Sample request.....	12
Sample Response (ok case)	13
Operation /adressaenderung	14
Sample request.....	15
Sample Response (ok case)	15
Sample request.....	17
Sample Response.....	17
Operation /dokumenteNachreichen	18
Sample request.....	19
Sample Response (ok case)	19
Error Codes	20
Status Codes	20
Message Codes.....	20

New Membership API

Release Notes

Version	Date	Features
2.5.0	June 2022	New operation /dokumenteNachreichen - allows a later submit of documents for an existing membership application - added error codes 80000, 80010, 80011, 80012 operation /einreichen - new parameter 'vorlBescheinigung' provides a provisional certificate of membership within the response 'email' maximum length is now 241 characters - some minor changes in validation of 'versorgungsbezeuge' and 'kinder' - line feed ('\n') will be treated as illegal character for all string-fields
2.4.0	19.10.2021	New Endpoint for testing future/upcoming features - The new staging-endpoint https://tk.de/service/rest/public/staging/neuaufnahmeantrag/v2/next will provide a testing-opportunity for new and upcoming features. - The previously provided staging-endpoint https://tk.de/service/rest/public/staging/neuaufnahmeantrag/v2 is still provided and will represent a mirror of the current productive version Operation /einreichen - new field 'geburtsland' (land of birth) is mandatory for all customer types. This field will become effective in December 2021 and is accepted but ignored in this API version. - new error codes for 'geburtsland' 30076 and 30077. These error will only be triggered on the staging-next endpoint. Removed Jobtitle related error codes 50050, 50051, 50052
2.3.0	08.03.2021	Added validation and error code 30124 - field 'versicherungsbeginn' must be before start of employment/studies (only for member type 'BERUFSTAETIGE', 'SAISONBESCHAEFTIGTE', 'AUSZUBILDENDE', 'STUDIERENDE') Added new Kundengruppe 'SELBSTSTAENDIGE' (self employed) - new sub-datastructure "selbststaendige": {} (see yaml specification file) Changed field validation 'geburtsname', 'geburtsort' and 'staatsangehoerigkeit' are now mandatory for all member types. In previous versions those fields were mandatory for members coming from foreign countries and members not providing a 'versicherternummer'. Changed field 'maklervollmacht' - The field changed from boolean to an enum of 'BASIS' (which is identical to the older version of the mandate) or 'ERWEITERT' (which includes a passage that the broker is allowed to access status updates for this health insurance application). Note that that /status operation will only provide status information for an 'ERWEITERT' mandate - For backward compatibility, 'maklervollmacht' can still be given as Boolean value (true/false). A value of true will be mapped to 'BASIS' in all cases.
2.2.1	26.01.2021	Removed strict error checking behaviour. In previous versions it was an error to supply field values that were not necessary in the particular case context. Unnecessary field values are now simply ignored instead. Removed error codes 10011, 10021, 10032, 10041, 10051, 10061, 30131, 30141, 50081, 50091.
2.2.0	07.12.2020	Changes due to legal changes in Germany.

New Membership API

		- field 'innerhalbBindungsfrist' removed/ignored - field 'gekuendigtZum' removed/ignored - For Member type "BERUFSTAETIGE" with attributes 'imAuslandGelebt'=false, 'versicherungsArt'='gesetzlich' and 'selbstVersichert'='true' the fields 'kinder' and 'versorgungsbeuege' are removed/ignored - field 'pflegeversicherungBefreit' has been marked deprecated and replaced by 'kvPvBefreit'. See yaml documentation for further information. Additional changes - For Member type "BERUFSTAETIGE" with attribute 'selbststaendig'=true the following new fields are required: 'existenzgruender' 'beschaeftigtMehrereMinijobber' 'beschaeftigtArbeitnehmer' 'stundenSelbststaendigkeit' 'einkommenSelbststaendigkeit' 'stundenArbeitnehmer' 'entgeltArbeitnehmer' Validation of antragId Structure removed from /status operation
2.1.0	-	Operation /status - used to query the status of a given set of 'antragIds' - the response contains information to membership state actions - the response also contains information regarding the commission payment
2.0.1	22.06.2020	Operation /einreichen For member type "BERUFSTAETIGE" and "SAISONBESCHAEFTIGTE" the field 'jobtitle' has been removed.
2.0.0	-	Authentication has changed. User Credentials have been removed from the message payload. Instead, a token (Json Web Token) must be passed in the Authorization header of the request. The token has to be obtained with a separate request to an authentication endpoint, is valid for 30 minutes and can be reused for several requests. Your user credentials do not change. Operation /einreichen - Authorization-Header and nsj-Cookie is required 'vermittler' has been added to set a specific agent for commission - The field 'antragid' contained in the response has been renamed to 'antragId' to be consistent with using camel case Operation /adressaenderung - Authorization-Header and nsj-Cookie is required - We changed the name of to 'antragId' to be consistent with using camel case Operation /getApiAccessToken - new operation to acquire a token for use with other operations - a cookie will also be returned, which must be included in addition the JWT metaDaten - Structure - Removed 'partnerId' and 'token' - new optional field 'vermittler' for passing a different TK-Partner-Id than defined by your login for commission calculation - new optional field 'kooperationId' for marking an request as belonging to a special sales cooperation or promotion (Note that this field will currently not be processed but will be activated in near future) persDaten - Structure 'versicherungsbeginn' is now mandatory

New Membership API

1.4.0	18.02.2020	Operation /einreichen - Added support for new member type "BERUFSTAETIGE" (employees) - Added support for new member type "SAISONBESCHAEFTIGTE" (seasonal employment) - Added support for new member type "AUSZUBILDENDE" (trainees) Operation /adressaenderung - Extended internal status of the application for the acceptance of address changes. Address changes will now only be rejected when the new member has a confirmed membership Due to the changes in operation, we added validations and error codes
1.3.0	28.01.2020	New operation /adressaenderung for changing the resident address of a previously submitted membership request
1.2.0	07.01.2020	Removed the fields - fachrichtung - fachsemester - masterstudiengang - bachelorFachsemester - version Requests that contain the corresponding fields will still be processed. Added new Operation ADRESSAENDERUNG for changing the resident address of an existing membership application
1.1.0	17.09.2019	- Introduced new return value 'antragid' for EINREICHEN operation - Enabled /einreichen operation to have attached documents
1.0.2	09.09.2019	Introduced a length validation for 'hausnummer'
1.0.1	14.05.2019	Introduced staging endpoint for testing
1.0.1	12.03.2019	Changed validation for 'versicherungsbeginn' from "equals 'studienbeginn'" to "equals 'studienbeginn' or later"
1.0.0	10.12.2018	Initial Version

New Membership API

Overview

This is the technical documentation for using the TK New Membership API. By using this API you can automatically submit new membership requests to Techniker Krankenkasse.

The following operation is available on path **/neuaufnahmeantrag**:

/getAccessToken

Retrieves an Access Token which is used to authorize operational requests like **/einreichen** or **/adresaenderung**.

- the generated JWT is valid for 30 minutes
- it can be used to invoke multiple operations
- note that the cookie **nsj** also needs to be sent with each request

We have not implemented an option to refresh the token, as we think that 30 minutes should be long enough to do a fair number of requests. You can always obtain a new token (and cookie) by calling this endpoint.

The following operations are available on path **/neuaufnahmeantrag/v2**:

/einreichen

Creates a new TK membership application request. Currently the API supports

- only STUDENTS, EMPLOYEES, SEASONAL EMPLOYEES and TRAINEES
- simultaneous upload of documents
- simultaneous upload of a passport photo

/adresaenderung

Changes the resident address in the member's membership request.

The membership request is identified by the 'antragId' returned when the **/einreichen** operation was invoked. A change of the resident address is only possible until the membership has been confirmed by TK. Afterwards the address can only be changed by the new member himself.

/status

Queries the status of a set of antragIds.

The status can only be requested for antragId's owned by this user. The response will contain information about the current membership status, the commission status and if any actions need to be performed by the new member for further processing of his/her membership.

New Membership API

/dokumenteNachreichen

Allows to submit additional documents to an existing membership application identified by an antragId.

In addition to the document-attachments (passed as in /einreichen as multipart-mixed request) a comment can be added to give additional information on the added documents

API Specification

The API specification (Open API 3.0 yaml file) is available for download on the New Membership API section of tk.de. Note that there are two separate documentations, one about the authentication (/getApiAccessToken) and one about the new membership API (other operations).

Credentials

Your credentials for using this API are provided separately.

Code Examples

Code examples are available for download on the New Membership API section of tk.de.

Operation /neuaufnahmeantrag/getApiAccessToken

Authentication and authorization

With API 2.x the authentication and authorization has changed: "partnerId" and "token" are no longer included in the actual operation, instead a token has to be included in the "Authorization" header of the request.

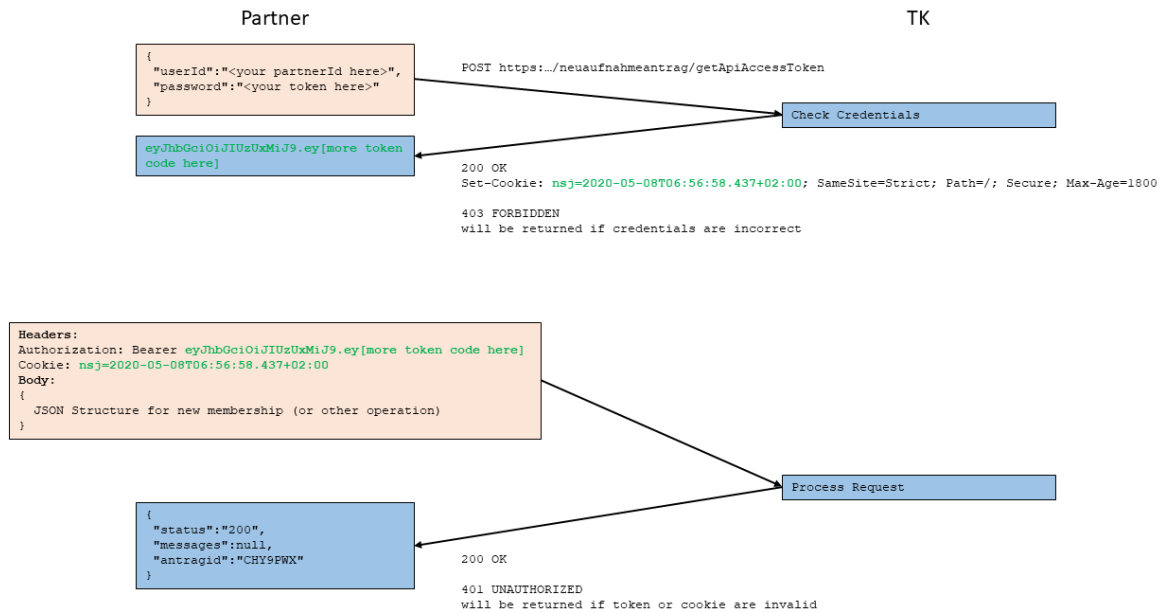
The token follows the specification of Json Web Tokens (<https://tools.ietf.org/html/rfc7519>). To improve security and prevent token sidejacking attacks a request to this endpoint also sends a cookie that must be included in the subsequent requests.

To obtain the token the new operation /getApiAccessToken can be used. Note that existing user credentials do not change, but the names of the fields have changed for this operation:

API v1 data field	API v2 data field
partnerId	userId
token	password

The new flow for requests is as follows:

New Membership API



Token and cookie are valid for 30 minutes and can be used for all operation types and for as many requests as desired.

The request for retrieving a token requires

- Data transmission via POST request
- ContentType application/json
- Accept header is mandatory
- User-Agent must be set

Returned Content-Type is 'text/plain;charset=utf-8'



Note that our server infrastructure also will set an INSTSESSIONID-Cookie, that needs to be passed with every request/operation to avoid server-hopping and token signature issues!

Sample Request

```
Request-Method: POST
Status Code: 200 OK
Accept: text/plain
Accept-Encoding: gzip, deflate, br
Accept-Language: de-DE,de;q=0.9,en-US;q=0.8,en;q=0.7
Connection: keep-alive
User-Agent: Mozilla/5.0
Content-Type: application/json
Content-Length: 45

{
  "userId":"your partner id (credentials)",
  "password":"your token you used for API v1 (credentials)"
}
```

Response:

```
HTTP/1.1 200 OK
Set-Cookie: nsj=2020-05-08T08:09:12.102+02:00; SameSite=Strict; Path=/; Secure; Max-Age=1800
Strict-Transport-Security: max-age=1800
Content-Length: 339
Content-Security-Policy: frame-ancestors 'self'
Keep-Alive: timeout=5, max=96
Connection: Keep-Alive
```




New Membership API

Content-Type: text/plain; charset=utf-8

eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJFSU5SRU1DSEVOIEFEUEkVU0FFTkRFU1VORyIsImF1ZCI6I1Q3MjA2MTAzSDkiLCJuc2oiOiIOMGEwNDRkYmQyOGM1ODlmMmJlNGM0NmUyNTdjZWRIYTdjZjdkMWI2OWYzNjYlM2JkZTEuODFjZWVzYjNjYzFmIiwiaXNzIjoiaHR0cHM6XC9cL3d3dy50ay5kZSIsImV4cCI6MTU4ODkxOTk1Mn0.WynC6iXRK6v10D-eH5h8Lecod-gyksrOJXIKxlLNK4UogqbB1Xf0rTVOH9BWYNmjW9hH4V2Ms-lNJugKCFrPyA

The relevant parts for submitting requests are the token itself and the cookie nsj.

New Membership API

Operation /einreichen

After receiving your request, we first check your credentials (token and cookie). Invalid credentials will be rejected with a return code 401 unauthorized.

A request with valid token and cookie will be validated technically and professionally. This is the general structure of a request:

- Data transmission via POST request
- ContentType multipart/mixed
- Accept header is mandatory
- User-Agent must be set
- Exactly one dataset as structured JSON object
- optional document uploads

Technically invalid requests will be rejected with code 400 bad request.

The professional validation checks the consistency of the request dataset. Any validation errors will also lead to a rejected request with code 400 bad request.

In any case the request will return a JSON structure containing further information on the validation issues.

Note that the message property contains an error description (in German) which is deprecated and will be removed with the next API version.

```
{
  "status": "400",
  "antragid": null,
  "messages": [
    {
      "code": "10020",
      "message": "Angabe von 'krankenversicherungName' ist Pflicht, wenn 'imAuslandGelebt' false oder Abkommenstaat"
    },
    {
      "code": "10030",
      "message": "Angabe von 'versicherungsart' ist Pflicht, wenn 'imAuslandGelebt' false oder Abkommenstaat"
    }
  ]
}
```

An antragId will only be returned if the request is accepted. This ID can later be used for identification of the membership request within the system and is valid until the membership is fully processed within TK.

New Membership API

Upload of documents

Documents can be attached by embedding them into the request. There is a limit of 10 documents in total with a maximum size of 10MB for each file. Additionally a passport photo can be attached, which is later used for the TK eHealth card (Gesundheitskarte).

All binary data has to be encoded as BASE64 and an appropriate Content-Type must be set.

The *Content-Length* Header needs to reflect the total size of the request (including attachments).

The following headers need to be set within the request-boundary:

```
Content-Disposition: name="PASSBILD"; filename="passport_photo.jpg"
```

```
Content-Transfer-Encoding: BASE64
```

```
Content-Type: "image/jpeg"
```

Only used for passport photo. The picture size must be 300x400 px minimum.

Accepted formats: jpg, jpeg, tif, png

```
Content-Disposition: name="DOKUMENT"; filename="meinDokument.pdf"
```

```
Content-Transfer-Encoding: BASE64
```

```
Content-Type: "application/pdf"
```

For all other documents. Accepted formats: plain text, pdf, doc, docx, jpeg, png, bmp, tiff

Sample request

```
Request Method: POST
Status Code: 200 OK
Accept: application/json
Accept-Encoding: gzip, deflate, br
Accept-Language: de-DE,de;q=0.9,en-US;q=0.8,en;q=0.7
Authorization: Bearer
eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJFSU5SRUldSEVOIEFEUkVtU0FFTkRlVORyIsImFlZCI6IiQ3MjA2MTAzSDkiLCJuc2oiOiIOMGEwNDRkYmQyOGM1ODlmMmJlNGM0NmUyNTdjZWRIYkdjZjZkdWVzYjY1M2JkZTEwODFjZWZlYjYjYzFmIiwiaXNzIjoiaHR0cHM6XC9cL3d3dy50ay5kZSIsImV4cCI6MTU4ODkxOTk1Mn0.WynC6iXRXK6v10D-eH5h8Lecod-gyksrOJXIKx1LNK4UogqbBlXf0rTVOH9BWNmJw9hH4V2Ms-1NjUGKCFrPyA
Cookie: nsj=2020-05-08T08:09:12.102+02:00
Connection: keep-alive
Content-Length: 1648086
Content-Type: multipart/mixed; boundary=-----249072686915163
User-Agent: Mozilla/5.0
-----249072686915163
Content-Disposition: form-data; name="neuaufnahme"
Content-Type: application/json

{
  "metaDaten": {
    "vermittler": null,
    "vorgangsId": null
  },
  "kundengruppe": "STUDIERENDE",
  "sprache": "EN",
  "maklervollmacht": true,
  "bestehendeVersicherung": {
    "imAuslandGelebt": true,
    "landLetzteVersicherung": "CN",
    "krankenversicherungName": null,
    "versicherungsart": null,
    "selbstVersichert": null,
    "pflichtversichert": null,
    "innerhalbBindungsfrist": null,
    "gekuendigtZum": null
  },
  "studierende": {
    "befreitKv": false,
    "hochschule": "Universität Münster",
    "studienbeginn": "01.04.2019"
  },
  "persDaten": {
    "name": {
      "anrede": "Herr",
      "titel": null,
      "vorname": "Lee",
      "nachname": "Wang",
      "namenszusatz": null
    },
    "adresse": {
      "strasse": "Pekingweg",
      "hausnummer": "66",
      "adresszusatz": null,
      "plz": "55433",
      "ort": "Münster",
      "land": "DE"
    },
    "email": "lee.wang@uni-muenster.de",
    "telefon": "+86 139-1099 8888",
    "geburtsdatum": "24.02.1996",
    "versichertennummer": null,
    "geburtsname": "Wang",
    "geburtsort": "Shenzen",
    "geburtsland": "CN",
    "staatsangehoerigkeit": "CN",
    "versorgungsbezüge": false,
    "kinder": false,
    "mitversicherungVonAngehoerigen": false,
    "versicherungsbeginn": "01.04.2019"
  }
}
-----249072686915163
Content-Disposition: name="PASSBILD"; filename="my_passport_photo.jpg";
Content-Transfer-Encoding: BASE64
Content-Type: "image/jpeg"

/9j/4AAQSkZJRgABAQEAdgB2AAD/4RIERXhpZgAAT... (more Base64 data)
-----249072686915163
Content-Disposition: name="DOKUMENT"; filename="test.pdf";
Content-Transfer-Encoding: BASE64
Content-Type: "application/pdf"

JVBERi0xLjQNCiV1tbWlDQoxIDAgb2JqDQo8PC9Ue... (even more Base64 data)
-----249072686915163--
```

New Membership API

Sample Response (ok case)

This is a sample expected response with no validation errors and

```
"metaDaten": {  
  ...  
  "vorlBescheinigung": false  
}
```

Note that the returned 'antragId' will vary, as it is the Id that identifies this specific membership application.

```
{  
  "status": "200",  
  "messages": null,  
  "antragId": "L772GBW"  
}
```

Setting the "vorlBescheinigung" property to true will include a BASE64 encoded file within the property "vorlBescheinigungData":

TKeasy-System ist DT02

Status ist 200

```
{  
  "status": "200",  
  "messages": null,  
  "vorlBescheinigungData": "JVBERi0xLjQKJfbk/N8KMSAwIG9iago8PAovZSAvQ... [more BASE64 data]",  
  "antragId": "L772GBW"  
}
```

The size of the PDF File is about 115kB (BASE64 encoded) or 85kB (as decoded binary PDF file).

New Membership API

Operation /adressaenderung

After receiving your request we first check your credentials (token and cookie). Invalid credentials will be rejected with a return code 401 unauthorized.

A request with valid credentials will be validated technically and professionally. This is the general structure of a request:

- Data transmission via POST request
- ContentType application/json
- Accept header is mandatory
- User-Agent must be set
- Exactly one dataset as a structured JSON object

Technically invalid Requests will be rejected with code 400 bad request.

The professional validation checks the consistency of the request dataset. Any validation errors will also lead to a rejected request with code 400 bad request.

In any case, the request will return a JSON structure containing further information on the validation issues.

Note that the message property contains an error description (in German) which is deprecated and will be removed with the next API version.

```
{
  "status": "401",
  "messages": [
    {
      "code": "99071",
      "message": "Die 'antragid' ist ungültig oder gehört nicht zur 'partnerId'"
    }
  ]
}
```



New Membership API

Sample request

```
Request Method: POST
Status Code: 200 OK
Accept: application/json
Accept-Encoding: gzip, deflate, br
Accept-Language: de-DE,de;q=0.9,en-US;q=0.8,en;q=0.7
Authorization: Bearer
eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJFSU5SRUldSEVOIEFEUkVtU0FFTkRlVjVORyIsImF1dCI6IiQ3MjA2MTAzSDkiLCJuc2oiOiIOMGEwNDRkYmQyOGM1ODlmMmJlNGM0NmUyNTdjZWRIYTdjZjdkMWI2OWYzNjY1M2JkZTEwODFjZWZzYjNjYzFmIiwiaXNzIjoiaHR0cHM6XC9cL3d3dy50ay5kZSIsImV4cCI6MTU4ODkxOTk1Mn0.WynC6iXRXK6v10D-eH5h8Lecod-gyksrOJXIKx1LNK4UogqbB1Xf0rTVOH9BWNmjW9hH4V2Ms-1NJugKCFrPyA
Cookie: nsj=2020-05-08T08:09:12.102+02:00
Connection: keep-alive
Content-Length: 371
Content-Type: application/json
User-Agent: Mozilla/5.0

{
  "antragId": "ABC1234",
  "metaDaten": {
    "vorgangsId": null
  },
  "adresse": {
    "strasse": "Borgweg",
    "hausnummer": "1",
    "adresszusatz": null,
    "plz": "22303",
    "ort": "Hamburg",
    "land": "DE"
  }
}
```

Sample Response (ok case)

```
{
  "status": "200",
  "messages": null
}
```

New Membership API

Operation /status

After receiving your request we first check your credentials (token and cookie). Invalid credentials will be rejected with a return code 401 unauthorized.

This is the general structure of a request:

- Data transmission via POST request
- ContentType application/json
- Accept header is mandatory
- User-Agent must be set
- Maximum of 10 antragIds passed in the antragIds-Array.

Technically invalid Requests will be rejected with code 400 bad request.

Out of technical reasons, we can only deliver status information for applications that have been issued after June 21th 2020.

If one of the given IDs is not owned by the issuer of this request, the server will response with 401.

New Membership API

Sample request

```
Request URL: https://localhost/service/rest/public/neuaufnahmeantrag/v2/status
Request Method: POST
Status Code: 200 OK
Accept: application/json
Authorization: Bearer ... token ...
Content-Length: 98
Content-Type: application/json
Cookie: INSTSESSIONID=094d24ad-8dc6-42b7-8749-16708776823f.mx-pfilmu07; nsj=2020-11-10T07:09:50.336+01:00
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.183
Safari/537.36
```

```
{
  "antragIds": ["F2ABCDE", "B2ABCDE", "B5ABCDE", "M3ABCDE", "S2ABCDE"]
}
```

Sample Response

```
{
  "status": "200",
  "messages": [
  ],
  "statusResponse": [
    {
      "antragId": "F2ABCDE",
      "mitgliedsStatus": "Offen",
      "verguetungsStatus": "Offen",
      "aktionen": [
      ]
    },
    {
      "antragId": "B2ABCDE",
      "mitgliedsStatus": "InBearbeitung",
      "verguetungsStatus": "Offen",
      "aktionen": [
        {
          "aktion": "KVdSPruuefung",
          "erstellDatum": "09.11.2020"
        }
      ]
    },
    {
      "antragId": "B5ABCDE",
      "mitgliedsStatus": "InBearbeitung",
      "verguetungsStatus": "Offen",
      "aktionen": [
        {
          "aktion": "UploadPassbild",
          "erstellDatum": "09.11.2020"
        }
      ]
    },
    {
      "antragId": "M3ABCDE",
      "mitgliedsStatus": "Mitglied",
      "verguetungsStatus": "ZahlungVorbereitet",
      "aktionen": [
      ]
    },
    {
      "antragId": "S2ABCDE",
      "mitgliedsStatus": "Storniert",
      "verguetungsStatus": "Storniert",
      "aktionen": [
      ]
    }
  ]
}
```

New Membership API

Operation /dokumenteNachreichen

After receiving your request we first check your credentials (token and cookie). Invalid credentials will be rejected with a return code 401 unauthorized.

This is the general structure of a request:

- Data transmission via POST request
- ContentType multipart/mixed
- Accept header is mandatory
- User-Agent must be set
- Exactly one dataset as structured JSON object
- document uploads

Technically invalid Requests will be rejected with code 400 bad request.

If one of the given IDs is not owned by the issuer of this request, the server will response with 401.

The technical transmission of documents is identical to /einreichen operation, see Upload of documents.

Note that you need to provide at least 1 attached document when calling this operation.



New Membership API

Sample request

```
Request URL: https://localhost/service/rest/public/neuaufnahmeantrag/v2/status
Request Method: POST
Status Code: 200 OK
Accept: application/json
Authorization: Bearer ... token ...
Content-Length: 72222
Content-Type: multipart/mixed; boundary=-----116891789827210
Cookie: INSTSESSIONID=094d24ad-8dc6-42b7-8749-16708776823f.mx-pfilmu07; nsj=2020-11-10T07:09:50.336+01:00
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.183 Safari/537.36

-----116891789827210
Content-Disposition: form-data; name="neuaufnahme"
Content-Type: application/json

{
  "antragId": "USGD7XF",
  "nachreichenKommentar": "Certificate of enrollment"
}
-----116891789827210
Content-Disposition: name="DOCUMENT"; filename="document.pdf";
Content-Transfer-Encoding: BASE64
Content-Type: "application/pdf"

JVBERi0xLjQNCiWlbtbWlDQoxIDAgb2JqDQo8...(BASE64 File Data)

-----116891789827210--
```

Sample Response (ok case)

```
{
  "status": "200",
  "messages": null
}
```

New Membership API

Error Codes

Status codes are returned as result codes of your http requests as well as part of the JSON response.

Message Codes are results of the professional validation of your request and name the reason why a request was rejected by the TK server.

Status Codes

Status Code	Description
200	The request was accepted and processed
400	The request has validation errors. Please fix those errors and resend the request.
401	The Bearer-Token is invalid (or nsj-Cookie does not match), your request has been rejected.
403	Only getApiAccessToken-Operation: the provided PartnerId or Token is invalid.

Message Codes

Status Code	Description
10000	'imAuslandGelebt' is mandatory.
10010	'landLetzteVersicherung' is mandatory when 'imAuslandGelebt' true.
10012	'landLetzteVersicherung' must be provided as ISO-3166-1.
10020	'krankenversicherungName' is mandatory when 'imAuslandGelebt' false or the country has a social security agreement with Germany.
10022	'krankenversicherungName' is limited to 45 characters.
10023	'krankenversicherungName' contains invalid characters.
10030	'versicherungsart' is mandatory when 'imAuslandGelebt' false or the country has a social security agreement with Germany.
10031	'versicherungsart' must be 'gesetzlich' or 'privat'.
10040	'selbstVersichert' is mandatory when 'imAuslandGelebt' false and versicherungsart 'gesetzlich'.
10050	'pflichtversichert' is mandatory when 'imAuslandGelebt' false and versicherungsart 'gesetzlich' and 'selbstVersichert' true.
10060	'innerhalbBindungsfrist' is mandatory when 'imAuslandGelebt' false and versicherungsart 'gesetzlich' and 'selbstVersichert' true.
10080	'gekuendigtZum' must be provided in German date format 'DD.MM.YYYY'.
10081	'gekuendigtZum' must be at least 70 years in the past and no more than 18 months in the future.
20000	'befreitKv' is mandatory.
20010	'hochschule' is mandatory.
20011	'hochschule' is limited to 45 characters.
20012	'hochschule' contains invalid characters.
20040	'studienbeginn' is mandatory.
20041	'studienbeginn' must be provided in German date format 'DD.MM.YYYY'.
20042	'studienbeginn' must be at least 20 years in the past and no more than 18 months in the future.
20043	'versicherungsbeginn' must be at or after 'studienbeginn' when kundengruppe 'STUDIERENDE'.
21010	'ausbildungsbeginn' is mandatory.
21011	'ausbildungsbeginn' must be at least 42 months in the past and no more than 18 months in the future.

New Membership API

21012	'ausbildungsbeginn' must be provided in German date format 'DD.MM.YYYY'.
21020	'arbeitgeber' is mandatory.
21021	'arbeitgeber' is limited to 45 characters.
21022	'arbeitgeber' contains invalid characters.
21030	'arbeitgeberAdresse' is mandatory.
21040	'sozAusweisBeantr' is mandatory.
30000	'name' is mandatory.
30010	'adresse' is mandatory
30020	'email' has an invalid format. Expected format: max.mustermann@some.domain
30021	'email' is limited to 65 characters.
30030	'telefon' has an invalid format. Expected format: +1 123 456 789.
30031	'telefon' is limited to 20 characters.
30040	'geburtsdatum' is mandatory.
30041	'geburtsdatum' must be in the past.
30042	'geburtsdatum' must be provided in German date format 'DD.MM.YYYY'.
30043	'geburtsdatum' may not be more than 120 years in the past.
30045	'geburtsdatum' may not exceed 55 years (relative to the given 'versicherungsbeginn') in the past when kundengruppe 'SAISONBESCHAEFTIGTE' and 'imAuslandGelebt' true.
30050	'versichertennummer' is invalid (checksum mismatch).
30051	'versichertennummer' must be null when 'imAuslandGelebt' true.
30060	'geburtsname' is mandatory when 'imAuslandGelebt' true or ('imAuslandGelebt' false and 'versichertennummer' is null).
30061	'geburtsname' is limited to 45 characters.
30062	'geburtsname' contains invalid characters.
30070	'geburtsort' is mandatory when 'imAuslandGelebt' true or ('imAuslandGelebt' false and 'versichertennummer' is null).
30071	'geburtsort' is limited to 24 characters.
30072	'geburtsort' contains invalid characters.
30076	'geburtsland' is mandatory
30077	'geburtsland' must be provided as ISO-3166-1.
30080	'staatsangehoerigkeit' is mandatory when 'imAuslandGelebt' true or ('imAuslandGelebt' false and 'versichertennummer' is null).
30081	'staatsangehoerigkeit' must be provided as ISO-3166-1.
30090	'versorgungsbezeuge' is mandatory.
30100	'kinder' is mandatory.
30110	'mitversicherungVonAngehoerigen' is mandatory.
30120	'versicherungsbeginn' is mandatory.
30121	'versicherungsbeginn' must be provided in German date format 'DD.MM.YYYY'.
30123	'versicherungsbeginn' must be at least 12 months in the past and no more than 18 months in the future.
30124	'versicherungsbeginn' must be before or equal 'studienbeginn' or 'ausbildungsbeginn' or 'beschaeftigtSeitAb' (depending on given 'kundengruppe')
30130	'rentenbezug' is mandatory when 'kundengruppe' is not student-type
30140	'kvPvBefreit' is mandatory when 'kundengruppe' is 'STUDIERENDE'. Note that this field might contain the value of the deprecated field 'pflegeversicherungBefreit'
40000	'antragId' is mandatory
40001	'antragId' is invalid
40010	'adresse' is mandatory.
50000	'entgeltklasse' is mandatory.
50001	'entgeltklasse' is invalid. Valid values are 'versicherungspflichtig', 'ueber-jaeg'.
50010	'ersteBeschaeftigung' is mandatory.
50020	'rechtsbelehrung' is mandatory.
50021	'rechtsbelehrung' must be TRUE when ueberBBG is TRUE.
50030	'arbeitgeber' is mandatory.

New Membership API

50031	'arbeitsgeber' is limited to 45 characters.
50032	'arbeitsgeber' contains invalid characters.
50040	'arbeitsgeberAdresse' is mandatory.
50060	'beschaeftigtSeitAb' is mandatory.
50061	'beschaeftigtSeitAb' must be provided in German date format 'DD.MM.YYYY'.
50062	'beschaeftigtSeitAb' may not be more than 70 years in the past or 18 months in the future.
50080	'selbststaendig' is mandatory
50090	'geschaeftsfuehrer' is mandatory
50100	'existenzgruender' is mandatory when 'selbststaendig' = true
50110	'beschaeftigtMehrereMinijobber' is mandatory when 'selbststaendig' = true
50111	'beschaeftigtArbeitnehmer' is mandatory when 'selbststaendig' = true
50120	'stundenSelbststaendigkeit' is mandatory when 'selbststaendig' = true
50121	'stundenSelbststaendigkeit' must be between 0 and 999.99
50130	'einkommenSelbststaendigkeit' is mandatory when 'selbststaendig' = true
50131	'einkommenSelbststaendigkeit' must be between 0 and 999999.99
50140	'stundenArbeitnehmer' is mandatory when 'selbststaendig' = true
50141	'stundenArbeitnehmer' must be between 0 and 999.99
50150	'entgeltArbeitnehmer' is mandatory when 'selbststaendig' = true
50151	'entgeltArbeitnehmer' must be between 0 and 999999.99
60000	'antragIds' is mandatory
60002	Too many 'antragIds' provided. The maximum count for 'antragIds' is 10
70000	'selbststaendigSeit' is mandatory
70001	'selbststaendigSeit' may not be more than 70 years in the past or more than 18 months in the future.
70002	'selbststaendigSeit' must be provided in German date format 'DD.MM.YYYY'.
70010	'einkommensteuerbescheidVorhanden' is mandatory
70020	'monatlicheEinkuenfte' is mandatory when 'einkommensteuerbescheidVorhanden' = false
70021	'monatlicheEinkuenfte' must be between 0 and 999999.99
70030	'argeLeistungen' is mandatory
70040	'weiteresEinkommen' must be between 0 and 999999.99
70041	'weiteresEinkommenArt' is mandatory when 'weiteresEinkommen' is not null
70042	'weiteresEinkommenArt' is limited to 50 characters
70050	'partnerNichtGkv' is mandatory
70060	'krankengeldAnspruch43' is mandatory
70070	'krankengeldAb' is mandatory when 'krankengeldAnspruch43' = true
70071	'krankengeldAb' may not be in the past or more than 18 months in the future.
70072	'krankengeldAb' must be provided in German date format 'DD.MM.YYYY'.
70080	'aktuellArbeitsunfaehigSeit' may not be more than 50 years in the past or anywhere in the future.
70081	'aktuellArbeitsunfaehigSeit' must be provided in German date format 'DD.MM.YYYY'.
70090	'aktuellKrankengeldAnspruchSeit' may not be more than 50 years in the past or later than 18 months in the future.
70091	'aktuellKrankengeldAnspruchSeit' must be provided in German date format 'DD.MM.YYYY'.
80000	'antragId' is mandatory
80010	'nachreichenKommntar' is limited to 250 characters
80011	'nachreichenKommntar' contains invalid characters
80012	at least 1 document is mandatory
90000	'anrede' is mandatory.
90001	'anrede' must be 'Herr' or 'Frau'.
90010	'vorname' is mandatory.
90011	'vorname' must be between 2 and 27 characters.

New Membership API

90012	'vorname' contains invalid characters.
90020	'nachname' is mandatory.
90021	'nachname' must be between 2 and 27 characters.
90022	'nachname' contains invalid characters.
90030	'titel' must be between 2 and 15 characters.
90031	'titel' contains invalid characters.
90040	'nameszusatz' must be between 2 and 15 characters.
90041	'nameszusatz' contains invalid characters.
91000	'strasse' is mandatory.
91001	'strasse' must be between 2 and 23 characters.
91002	'strasse' contains invalid characters.
91010	'plz' is mandatory.
91011	'plz' must be between 2 and 10 characters.
91012	'plz' contains invalid characters.
91013	'plz' for German addresses is limited to 5 characters
91020	'ort' is mandatory.
91021	'ort' must be between 2 and 35 characters.
91022	'ort' contains invalid characters.
91030	'land' is mandatory.
91031	'land' must be provided as ISO-3166-1.
91032	'land' must be Germany ('DE') because we only accept German addresses to be changed to.
91040	'hausnummer' is limited to 8 characters.
91041	'hausnummer' contains invalid characters.
91042	'hausnummer' may only contain 4 characters as number and 4 characters as number appendix.
91050	'adresszusatz' is limited to 35 characters.
91051	'adresszusatz' contains invalid characters.
98000	'partnerId' is mandatory.
98010	'token' is mandatory.
98016	The access token is not valid for the desired operation
98015	'token' or 'partnerId' is invalid.
98017	'vorgangId' is limited to 25 characters.
98018	'vermittler' is limited to 10 characters.
99000	'metaDaten' is mandatory.
99010	'studierender' is mandatory when kundengruppe 'STUDIERENDE'.
99011	'studierender' must be null when kundengruppe is not 'STUDIERENDE'.
99012	'beschaeftigter' is mandatory when kundengruppe 'BERUFSTAETIGE' or 'SAISONBESCHAEFTIGTE'.
99013	'beschaeftigter' must be null when kundengruppe is not 'BERUFSTAETIGE' or 'SAISONBESCHAEFTIGTE'.
99014	'auszubildende' is mandatory when kundengruppe 'AUSZUBILDENDE'.
99015	'auszubildende' must be null when kundengruppe is not 'AUSZUBILDENDE'.
99020	'bestehendeVersicherung' is mandatory.
99030	'persDaten' is mandatory.
99040	'sprache' is mandatory.
99041	'sprache' must be 'DE' or 'EN'.
99050	'kundengruppe' is mandatory.
99051	'kundengruppe' is invalid. For now use 'STUDIERENDE', 'BERUFSTAETIGE', 'SAISONBESCHAEFTIGTE', 'AUSZUBILDENDE'.
99060	No dataset (JSON) was found in the request.
99061	More than 1 dataset was found in the request. Exactly 1 is expected.
99062	The dataset (JSON) could not be parsed.
99064	One of the attached files is too large. Maximum size for attachments is 10MB.

New Membership API

99065	Too many attachments. The maximum number of allowed attachments is 10.
99066	The attached passport photo does not meet the requirements. Resolution must be 300x400 point minimum.
99070	A confirmation of holding the mandate is mandatory. 'maklervollmacht' must be true.
99071	The address of 'antragsId' cannot be changed. Access denied.
99072	'maklervollmacht' must be either 'BASIS' or 'ERWEITERT'
99080	The operation you are calling is currently not active for this endpoint.
99081	The upload function for passport photos is not available for this 'kundengruppe' at this time. It may be available in near future.
99090	'selbststaendige' is mandatory when kundengruppe 'SELBSTSTAENDIGE'.
99091	'selbststaendige' must be null when kundengruppe is not 'SELBSTSTAENDIGE'.