

TecAlliance

Tecalliance myITG API documentation

VERSION 6.2

28.07.2026

Copyright © TecAlliance GmbH – The transfer or transmission of this document in any form, also in part, requires the written approval of TecAlliance GmbH.

Contents

1. Authentication 2
2. Encryption/Decryption 2
3. ITG structure mapping 3
   3.1. Get all available ITGs from tenant 3
   3.2. Get all, one or multiple specific ITG incl. all children 3
   3.3. Get one or multiple ITG companies matched to customer data. 4
   3.4. Update a match between myITG and CustomerUniqueID 5
   3.5. Delete a match by CustomerUniqueID 6
4. Customer data 7
   4.1. Get all or some specific customer using CustomerUniqueID 7
   4.2. Add a customer 7
   4.3. Update a customer 8
   4.4. Delete a customer by unique ID 8
5. ITG member records 9
   5.1. Get all ITG or one specific ITG incl. all children 9
   5.2. Filter one or multiple ITG companies based on myITG ID 9
6. Sales data 11
   6.1. Defining sales data titles and structure 11
   6.2. Get sales data titles and definition 12
   6.3. Get all sales data 12
   6.4. Add sales data 13
   6.5. Update sales data by customer unique ID 13
   6.6. Delete sales data by customer unique ID 13
7. myMeeting 14
   7.1. Get all meetings 14
   7.2. List all appointment from a meeting 14
   7.3. Get appointment details 15
8. Reporting groups 17
   8.1. Get all reporting groups 17
9. Announcements 17
   9.1. Get announcements 17

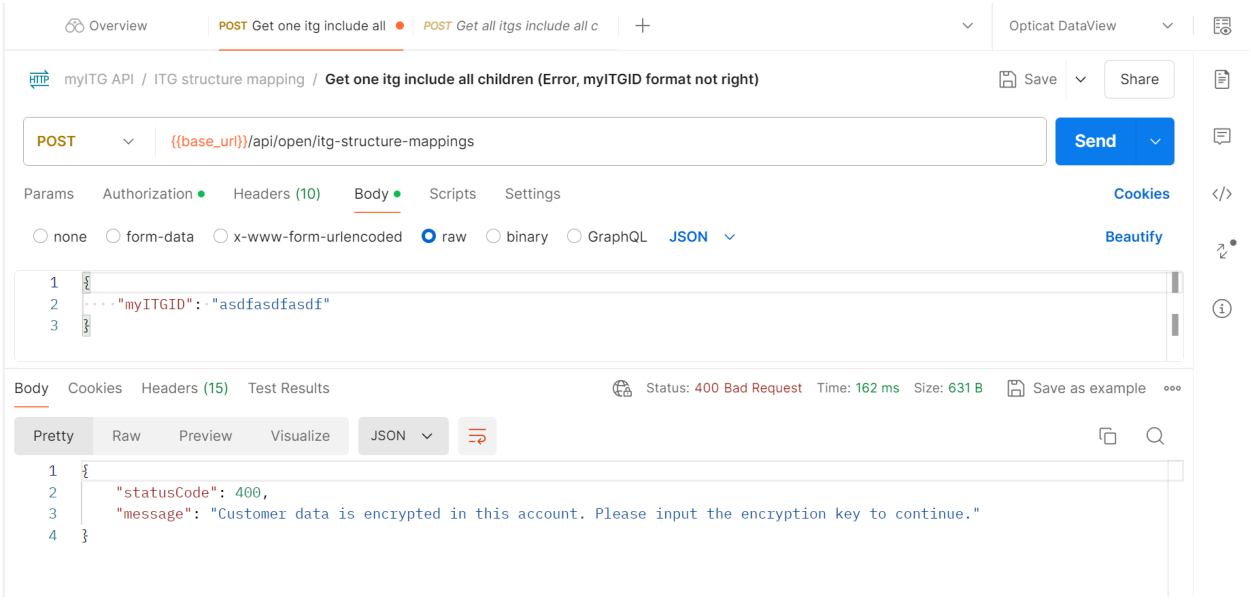
1. Authentication

These endpoints require a valid access token to be provided in the Authorization header as a bearer token.

```
curl -X POST -H "Authorization: Bearer {token}" -H "Content-Type: application/json" -d '{}' https://myitg.tecalliance.net/api/open/itgs
```

2. Encryption/Decryption

myITG has on option to encrypt customer data. This data will be encrypted stored in the database. There is a key needed for encryption and decryption. This key will be known only to the account operator and his/her team. Every account operator can add encryption to their own account. If your account is encrypted, then you will see an error message similar to the one below.



Please add optional key “encryptedKey” into the header and submit together with the correct encryption/decryption key for your account.

3. ITG structure mapping

3.1. Get all available ITGs from tenant.

URL: /api/open/itgs
Method: *POST*
Request body: -
Response body: *A JSON array of objects that represents ITG information, where each object has following properties:*

```
myITGID: myITG-ID of ITG (string, max. 13)
companyName: company name (string, max. 255)
shortName: company short name (string, max. 255)
```

3.2. Get all, one or multiple specific ITG incl. all children.

URL: /api/open/itg-structure-mappings
Method: *POST*
Request body:

```
myITGID: myITG-ID of company (string, optional, max. 13) or (string array,
string max. 13). If not set or null, all companies will be returned.
ERPCustomer: The ERPCustomer number from matched customer (string, optional)
```

Response body: *A JSON array of objects that represents companies information and mapping customer information, where each object has following properties:*

```
itg: the ITG name (string, max. 255)
companyPath: the company name of parents (string array, string max. 255)
idPath: the ID of parents (string array, string max. 13)
parentCompanyName: the company name of parent (string, max. 255)
companyName: company name (string, max. 255)
hierarchylevel: hierarchy level (number)
countryName: country name (string, max. 255)
countryCode: country code (string, max. 2)
address: address of company (string, max. 255)
zip: zip code of company (string, max. 255)
city: city of company (string, max. 255)
IDCode: ID code of company (string, max. 255)
vat_m: VAT number of myITG company (string, max. 255)
duns_m: D-U-N-S number of myITG company (string, max. 255)
alternativeBusinessIdentifier_m: Alternative Business Identifier of myITG
company (string, max. 255)
myITGID: myITG-ID of company (string, max. 13)
ERPCustomer: unique ID of customer (string, max. 255)
vat_c: VAT number of customer (string, max. 255)
duns_c: D-U-N-S number of customer (string, max. 255)
```

```
alternativeBusinessIdentifier_c: Alternative Business Identifier of customer
(string, max. 255)
myITGGoogleAddress: google address of company (string, max. 255)
CRMName: customer name (string, max. 255)
CRMAddress: address of customer (string, max. 255)
CRMGoogleAddress: google address of customer (string, max. 255)
lastMappingDate: last mapping date (string, format “YYYY-MM-DDThh:mm:ssZ”)
mappingtype: mapping type (string, “GoogleAddress", “Manual” or “Import”)
processed: (boolean)
lastUpdate: last update date (string, format “YYYY-MM-DDThh:mm:ssZ”)
contractStartDate: contract start date in format “YYYY-MM-DDThh:mm:ssZ”
(date)
contractEndDate: contract end date in format “YYYY-MM-DDThh:mm:ssZ” (date)
join: date of company join parent (string, format “YYYY-MM-DDThh:mm:ssZ”)
leave: date of company leave parent (string, format “YYYY-MM-DDThh:mm:ssZ”)
revenueReportable: Field indicating if myITG company report revenue (boolean,
“true” or “false”)
DQS: DQS_AVG value from company (double, a number between 0 and 100)
```

Note: If you have matched one myITG member to multiple of your customer, then you will get this myITG multiple times in the response body associated to your matched customer.

3.3. Get one or multiple ITG companies matched to customer data.

URL: /api/open/itg-matches
Method: *POST*
Request body:

```
lastMappingDate: Mapping date in format “YYYY-MM-DDThh:mm:ssZ” (date,
optional). Renders all myITG companies matched after that date.
```

Response body: *A JSON array of objects that represents companies information and mapping customer information, where each object has following properties:*

```
itg: the ITG name (string, max. 255)
companyPath: the company name of parents (string array, string max. 255)
idPath: the ID of parents (number array, string max. 13)
parentCompanyName: the company name of parent (string, max. 255)
companyName: company name (string, max. 255)
hierarchylevel: hierarchy level (number)
countryName: country name (string, max. 255)
countryCode: country code (string, max. 2)
address: address of company (string, max. 255)
zip: zip code of company (string, max. 255)
city: city of company (string, max. 255)
IDCode: ID code of company (string, max. 255)
vat_m: VAT number of myITG company (string, max. 255)
```

```
duns_m: D-U-N-S number of myITG company (string, max. 255)
alternativeBusinessIdentifier_m: Alternative Business Identifier of myITG
company (string, max. 255)
myITGID: myITG-ID of company (string, max. 13)
ERPCustomer: unique ID of customer (string, max. 255)
vat_c: VAT number of customer (string, max. 255)
duns_c: D-U-N-S number of customer (string, max. 255)
alternativeBusinessIdentifier_c: Alternative Business Identifier of customer
(string, max. 255)
myITGGoogleAddress: google address of company (string, max. 255)
CRMName: customer name (string, max. 255)
CRMAddress: address of customer (string, max. 255)
CRMGoogleAddress: google address of customer (string, max. 255)
lastMappingDate: last mapping date (string, format “YYYY-MM-DDThh:mm:ssZ”)
mappingtype: mapping type (string, “GoogleAddress", “Manual” or “Import”)
processed: (boolean)
lastUpdate: last update date (string, format “YYYY-MM-DDThh:mm:ssZ”)
contractStartDate: contract start date in format “YYYY-MM-DDThh:mm:ssZ”
(date)
contractEndDate: contract end date in format “YYYY-MM-DDThh:mm:ssZ” (date)
join: date of company join parent (string, format “YYYY-MM-DDThh:mm:ssZ”)
leave: date of company leave parent (string, format “YYYY-MM-DDThh:mm:ssZ”)
revenueReportable: Field indicating if myITG company report revenue (boolean,
“true” or “false”)
DQS: DQS_AVG value from company (double, a number between 0 and 100)
```

3.4. Update a match between myITG and CustomerUniqueID

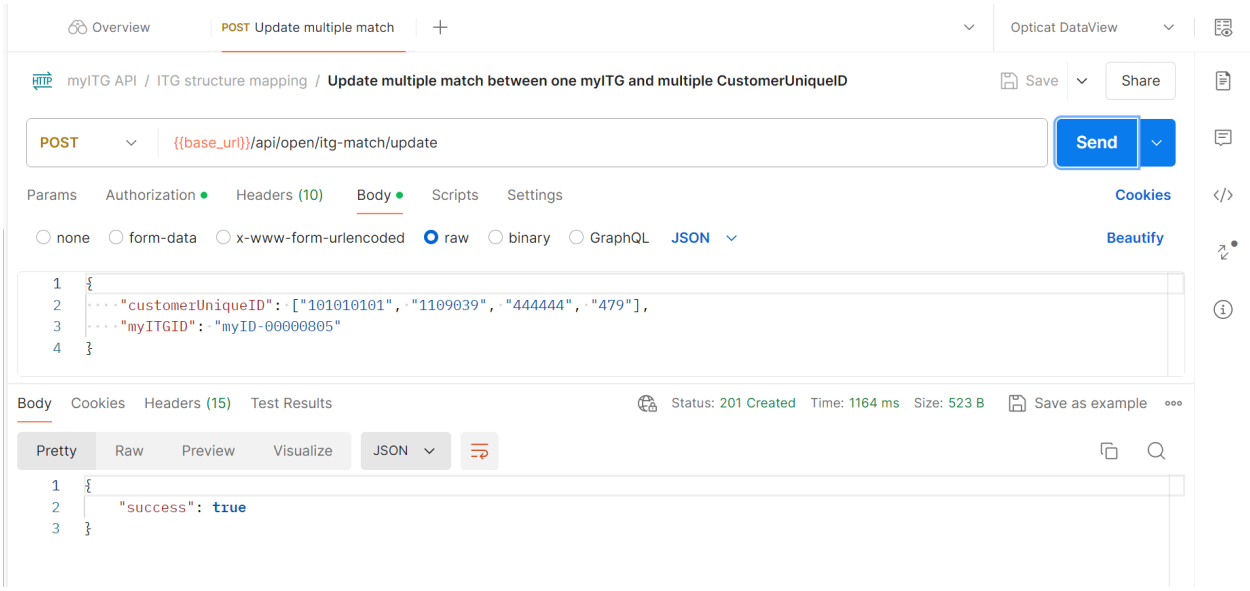
URL: api/open/itg-match/update
Method: *POST*
Request body:

```
customerUniqueID: the unigue ID of customer (string, max. 255) or (array of
string, max. 255) when matching one myITG company to multiple customers.
myITGID: myITG-ID of company (string, max. 13)
contractStartDate: contract start date in format “YYYY-MM-DDThh:mm:ssZ”
(date,optional)
contractEndDate: contract end date in format “YYYY-MM-DDThh:mm:ssZ” (date,
optional)
```

Response body: *A JSON object has following properties:*

```
success: indicates whether the request was successful (boolean)
message: a description of the problem when not success (string, optional)
```

Note: If you match one myITG company to multiple customer, then you need to add an array of string to the customerUniqueID like the request body in the screenshot below.



3.5. Delete a match by CustomerUniqueID

URL: /api/open/itg-match/delete
Method: *POST*
Request body:

```
customerUniqueID: the unique ID of customer (string, max. 255)
```

Response body: *A JSON object has following properties:*

```
success: indicates whether the request was successful (boolean)
message: a description of the problem when not success (string, optional)
```

Note: If you matched one myITG company to multiple customer, then the delete request will remove only the customer with the ID from your match. If you want to remove all matches, then you need to repeat the request for each matched customer.

4. Customer data

4.1. Get all or some specific customer using CustomerUniqueID

URL: /api/open/customers
Method: *POST*
Request body:

```
uniqueIDs: the unique IDs of customers (string array, string max. 255). If
not set or null, all customers will return.
```

Response body: *A JSON array of objects that represents ITG companies information, where each object has following properties:*

```
customerUniqueID: unique ID of customer (string, max. 255)
name: name of customer (string, max. 255)
name2: another name of customer (string, max. 255)
address1: address of customer (string, max. 255)
address2: another address (string, max. 255)
zip: zip of customer (string, max. 255)
city: city of customer (string, max. 255)
countryISO2: 2 chars country ISO code (string, max. 2)
VAT: VAT number (string, max. 255)
DUNS: D-U-N-S number (string, max. 255)
alternativeBusinessIdentifier: Alternative Business Identifier (string, max.
255)
processed: is processed (boolean)
lastUpdate: last update date (string, format “YYYY-MM-DDThh:mm:ssZ”)
```

4.2. Add a customer

URL: /api/open/customer/add
Method: *POST*
Request body:

```
customerUniqueID: unique ID of customer (string, max. 255)
name: name of customer (string, max. 255)
name2: another name of customer (string, max. 255)
address1: address of customer (string, max. 255)
address2: another address (string, max. 255)
zip: zip of customer (string, max. 255)
city: city of customer (string, max. 255)
countryISO2: 2 chars country ISO code (string, max. 2)
VAT: VAT number (string, max. 255)
DUNS: D-U-N-S number (string, max. 255)
alternativeBusinessIdentifier: Alternative Business Identifier (string, max.
255)
processed: is processed (boolean)
```

Response body: *A JSON object has following properties:*

```
success: indicates whether the request was successful (boolean)
message: a description of the problem when not success (string, optional)
```

4.3. Update a customer

URL: /api/open/customer/update
Method: *POST*
Request body:

```
customerUniqueID: unique ID of customer (string, max. 255)
name: name of customer (string, max. 255)
name2: another name of customer (string, max. 255)
address1: address of customer (string, max. 255)
address2: another address (string, max. 255)
zip: zip of customer (string, max. 255)
city: city of customer (string, max. 255)
countryISO2: 2 chars country ISO code (string, max. 2)
vat: VAT number (string, max. 255)
duns: D-U-N-S number (string, max. 255)
alternativeBusinessIdentifier: Alternative Business Identifier (string, max.
255)
processed: is processed (boolean)
```

Response body: *A JSON object has following properties:*

```
success: indicates whether the request was successful (boolean)
message: a description of the problem when not success (string, optional)
```

4.4. Delete a customer by unique ID

URL: /api /open/customer/delete
Method: *POST*
Request body:

```
uniqueID: the unique ID of the customer (string, max. 255)
```

Response body: *A JSON object has following properties:*

```
success: indicates whether the request was successful (boolean)
message: a description of the problem when not success (string, optional)
```

5. ITG member records

5.1. Get all ITG or one specific ITG incl. all children

URL: /api/open/itg-member-records
Method: *POST*
Request body:

```
myITGID: myITG-ID of company (string, optional, max. 13). If not set or null, all companies will return.
```

Response body: *A JSON array of objects that represents companies information and mapping customer information, where each object has following properties:*

```
myITGID: myITG-ID of company (string, max. 13)
companyName: company name (string, max. 255)
parentCompanyID: parent myITG-ID to display (string, max. 13)
countryCode: country code (string, max. 2)
address: address of company (string, max. 255)
IDCode: ID code of company (string, max. 255)
shortName: short name of company (string, max. 255)
vat: VAT number (string, max. 255)
duns: D-U-N-S number (string, max. 255)
alternativeBusinessIdentifier: Alternative Business Identifier (string, max. 255)
website: website url of company (string, max. 255)
contact: contact email of company (string, max. 255)
join: date of company join parent (string, format “YYYY-MM-DDThh:mm:ssZ”)
leave: date of company leave parent (string, format “YYYY-MM-DDThh:mm:ssZ”)
type: company type (string, “MEMBER”, “ITG”, “NTG”, “CUSTOMER_GROUP” or “AFFILIATE”)
status: the active status (string, “ON” or “OFF” or “PARKING”)
created: date of company created (string, format “YYYY-MM-DDThh:mm:ssZ”)
updated: date of company updated (string, format “YYYY-MM-DDThh:mm:ssZ”)
ITG: the ITG short name which company under (string, max. 255)
lastUpdate: last update date (string, format “YYYY-MM-DDThh:mm:ssZ”)
revenueReportable: Field indicating if myITG company report revenue (boolean, “true” or “false”)
DQS: DQS_AVG value from company (double, a number between 0 and 100)
```

5.2. Filter one or multiple ITG companies based on myITG ID

URL: /api/open/itg-member-datas
Method: *POST*
Request body:

```
myITGID: myITG-IDs of company (string array, string max. 13).
```

Response body: *A JSON array of objects that represents companies information and mapping customer information, where each object has following properties:*

```
myITGID: myITG-ID of company (string, max. 13)
companyName: company name (string, max. 255)
parentCompanyID: parent myITG-ID to display (string, max. 13)
countryCode: country code (string, max. 2)
address: address of company (string, max. 255)
IDCode: ID code of company (string, max. 255)
shortName: short name of company (string, max. 255)
website: website url of company (string, max. 255)
contact: contact email of company (string, max. 255)
join: date of company join parent (string, format “YYYY-MM-DDThh:mm:ssZ”)
leave: date of company leave parent (string, format “YYYY-MM-DDThh:mm:ssZ”)
type: company type (string, “MEMBER”, “ITG”, “NTG”, “CUSTOMER_GROUP” or
“AFFILIATE”)
status: the active status (string, “ON” or “OFF” or “PARKING”)
created: date of company created (string, format “YYYY-MM-DDThh:mm:ssZ”)
updated: date of company updated (string, format “YYYY-MM-DDThh:mm:ssZ”)
ITG: the ITG short name which company under (string, max. 255)
lastUpdate: last update date (string, format “YYYY-MM-DDThh:mm:ssZ”)
revenueReportable: Field indicating if myITG company report revenue (boolean,
“true” or “false”)
DQS: DQS_AVG value from company (double, a number between 0 and 100)
```

6. Sales data

6.1. Defining sales data titles and structure

URL: */api/open/sales-data/definition*
Method: *Post*
Request body

```
{
    "params": [
        {
            "column": “B”,
            "name": "Brand",
            "value": “String”
        },
        {
            "column": “C”,
            "name": "Sales 2021",
            "value": “Currency”
        },
        {
            "column": “D”,
            "name": "Sales 2022",
            "value": “Currency”
        },
        {
            "column": “E”,
            "name": "Absolute, ABS(D-C)",
            "value": “”
        },
        {
            "column": “F”,
            "name": " Relative, REL(D-C)",
            "value": “”
        }
    ]
}
```

Each object in “params” should has following properties:
column: column name of table (string, A-Z, max. 1)
name: the name of this column data (string, max. 50)
value: the type of this column data (string, should be “String”, “Currency” or “”)

```
The objects and parameters above have to follow the requirements of the sales
data table. The first object and the first column A are reserved for
customerUniqueID. All other objects and columns are flexible. The function
ABS(X-Y) and REL(X-Y) have to linked to an object and column set previously.
X and Y are representing columns.
```

Response body:

```
success: indicates whether the request was successful (boolean)
message: a description of the problem when not success (string, optional)
```

6.2. Get sales data titles and definition

URL: /api/open/sales-data/get-definition
Method: *POST*
Request body: -

Response body:
lastUpdate: timestamp from last update (number)
definition: *A JSON array of objects based on the definition made under 6.1*

example:

```
{
    lastUpdate: 2025-09-23T07:37:26Z,
    definition: [
        {
            "column": “B”,
            "name": "Brand",
            "value": “String”
        },
        {
            "column": “C”,
            "name": "Sales 2021",
            "value": “Currency”
        },
    ]
}
```

6.3. Get all sales data

URL: /api/open/sales-data/list
Method: *POST*
Request body: -

Response body: *A JSON array of objects, where each object has following properties:*

```
customerUniqueID: customer unique ID (string, max. 255)
lastUpdate: last update date (string, format “YYYY-MM-DDThh:mm:ssZ”)
(other data user uploaded, like "Product": "xxxx", "Sales 2021": 123123)
```

6.4. Add sales data

URL: /api/open/sales-data/add
Method: *POST*
Request body:

```
customerUniqueID: customer unique ID (string, max. 255)
(other data like "Product": "xxxx", "Sales 2021": 123123)
```

Important: Only customerUniqueID is a fixed key/value in the sales data request. All other keys/values are flexible. They have to follow the definition in function 6.1.

Response body: *A JSON object has following properties:*

```
success: indicates whether the request was successful (boolean)
message: a description of the problem when not success (string, optional)
```

6.5. Update sales data by customer unique ID

URL: /api/open/sales-data/update
Method: *POST*
Request body:

```
customerUniqueID: customer unique ID (string, max. 255)
data: a JSON array of objects likes [{ "Product": "aaa", "Sales 2021":
12111, ...}, { "Product": "bbb", "Sales 2021": 3333, ...}, ....]
```

Response body: *A JSON object has following properties:*

```
success: indicates whether the request was successful (boolean)
message: a description of the problem when not success (string, optional)
```

6.6. Delete sales data by customer unique ID

URL: /api/open/sales-data/delete
Method: *POST*
Request body:

```
customerUniqueID: customer unique ID (string, max. 255)
```

Response body: *A JSON object has following properties:*

```
success: indicates whether the request was successful (boolean)
message: a description of the problem when not success (string, optional)
```

7. myMeeting

7.1. Get all meetings

URL: /api/open/myMeeting
Method: POST
Request body:

```
keyword: meeting related keywords (string, optional, max. 255)
myITGID: myITG-ID of ITG (string, optional, string max. 13)
startDate: meeting start date format “YYYY-MM-DDThh:mm:ssZ” (string,
optional, max. 10)
endDate: meeting end date in format “YYYY-MM-DDThh:mm:ssZ” (string, optional,
max. 10)
sort: a fix value for sorting with “date ASC“, “date DESC“, “meetingID ASC”
or “meetingID DESC” (string, optional)
```

Response body: *A JSON array of objects, where each object has following properties:*

```
meetingID: meeting ID (integer),
meetingTitle: meeting name or title (string, max. 255)
description: meeting description (string, max 255)
myITGID: myITG-ID of ITG (string, max. 13)
itg: name of ITG
appointment: number of appointments (integer)
startDate: starting date of meeting in format “YYYY-MM-DDThh:mm:ssZ”
endDate: ending date of meeting in format “YYYY-MM-DDThh:mm:ssZ”
location: name of the location (string)
city: city name (string)
country: country code as ISO standard (string, max 2)
```

7.2. List all appointment from a meeting

URL: /api/open/myMeeting/appointment
Method: POST
Request body:

```
meetingID: ID of the meeting (integer)
status: status of an appointment with PREPARATION, PREPARATION_COMPLETED,
MEETING_COMPLETED, FOLLOW_UP_COMPLETED (string, optional)
timeZone: available time zones of a meeting with “Asia/Bangkok” as example
(string, optional)
```

Response body: *A JSON array of objects, where each object has following properties:*

```
appointmentID: ID of an appointment (integer),
myITGID: myITG-ID of ITG or member (string, max. 13)
```

```
companyName: The name of the ITG or ITG member (string)
topics: number of topics inside the appointment (integer)
date: date of the meeting in format “YYYY-MM-DDThh:mm:ssZ”
startTime: starting time of meeting in format hh-mm-ss (string)
endTime: ending time of meeting in format hh-mm-ss (string)
location: name of the location (string)
status: status of the appointment (string)
assignedTo: name of user when appointment was assigned to them (array,
string)

country: country code as ISO standard (string, max 2)
```

7.3. Get appointment details

URL: /api/open/myMeeting/appointmentDetail
Method: POST
Request body:

```
meetingID: ID of the meeting (integer, required)
appointmentID: ID of an appointment (integer, required)
```

Response body: *A JSON array of objects, where each object has following properties:*

```
id: ID of an appointment (integer, required)
status: status of an appointment with PREPARATION, PREPARATION_COMPLETED,
MEETING_COMPLETED, FOLLOW_UP_COMPLETED (string)
start: starting date/time from appointment in “YYYY-MM-DDThh:mm:ssZ” format
(date)
end: end date/time from appointment in “YYYY-MM-DDThh:mm:ssZ” format (date)
meetingTopics.title: title of a meeting topic (string)
meetingTopics.topic: topic of a meeting topic (string)
meetingTopics.subTopic: sub topic from a meeting topic (string)
meetingTopics.description: text description from a meeting topic (string)
meetingTopics.created: date/time when user created meeting topic in “YYYY-MM-
DDThh:mm:ssZ” format (date)
meetingTopics.updated: date/time when user updated meeting minute in “YYYY-
MM-DDThh:mm:ssZ” format (date)
meetingTopics.createdUser: name of user who created meeting topic (string)
meetingTopics.editedUser: name of user who edited meeting topic (string)
meetingMinutes.text: text/description from meeting minutes (string)
meetingMinutes.createdUser: The name of the user who created the meeting
minute.
meetingMinutes.editedUser: The name of the user who edited the meeting.
meetingMinutes.created: date/time when user created meeting minute in “YYYY-
MM-DDThh:mm:ssZ” format (date)
meetingMinutes.updated: date/time when user edited meeting minute in “YYYY-
MM-DDThh:mm:ssZ” (date)
```

meetingFollowUps.text: text/description from meeting minutes (string)
meetingFollowUps.createdUser: The name of the user who created the follow up meeting content (string)
meetingFollowUps.editedUser: The name of the user who edited the follow up meeting content (string)
meetingFollowUps.created: date/time when user created follow up meeting in “YYYY-MM-DDThh:mm:ssZ” format (date)
meetingFollowUps.updated: date/time when user edited follow up meeting in “YYYY-MM-DDThh:mm:ssZ” format (date)
meetingAttachments.name: name of the file which was attached to the meeting (string)
meetingAttachments.src: URL to download meeting attachment (URL)
meetingHistory: array of objects when existing (array/objects)
timeZone: time zone of meeting, e.g. Asia/Bangkok (string)
location: location of appointment when added and existing (string)
myITGID: myITG-ID of ITG or member (string, max. 13)
companyName: The name of the ITG or ITG member (string)
companyType: type of myITG company, e.g. MEMBER etc. (string)
companySite: url of myITG member website when existing (url)
companyEmail: email from myITG member when existing (email)
companyExtData: array of information when added by customer into appointment. (array)
salesFigure: array of information when sales data exists. (array)

8. Reporting groups

8.1. Get all reporting groups

URL: /api/open/reporting-groups
Method: POST
Request body:

Response body: *A JSON array of objects, where each object has following properties:*

```
repGroupId: reporting group ID (integer)
mgId: main group ID (integer)
maingroup: main group name (string)
sgId: sub group ID (integer)
subgroup: sub group name
gaNo: TecAlliance TecDoc GA (product group) ID (integer)
gaDescEn: GA name in English (string)
ga: TecDoc GA (string))
```

9. Announcements

9.1. Get announcements

URL: /api/open/announcements
Method: POST
Request body:

```
date: Date of announcement (“YYYY-MM-DDThh:mm:ssZ”, optional)
updateDate: Date when the announcement was added/edited (“YYYY-MM-
DDThh:mm:ssZ”, optional),
pageNo: Page which should be rendered (integer, optional, default 1)
pageSize: Number of items per page (integer, optional, max. 1.000, default
100)
```

Response body: *A JSON array of objects, where each object has following properties:*

```
list: Array of announcements (array)
     ITG: ITG name (string)
     ITG ID: ITG ID (integer)
     Parent: Parent name of the ITG (string)
     Parent ID: Parent ID (integer)
     Company: ITG member related to announcement (string)
     myID: ITG member ID related to announcement (string starting with myID-)
     Country: Country name of ITG member (string)
     Action: Announcement related action (string)
     Date: Date of announcement (“YYYY-MM-DDThh:mm:ssZ”)
     UpdateDate: Date when the announcement was added/edited (“YYYY-MM-
DDThh:mm:ssZ”)
```

```
Total: Total number of announcements (integer)
pageNo: Page which should be rendered (integer)
pageSize: Number of items per page (integer)
totalPage: Max. number of pages based on pageSize and pageNo (integer)
```