API Documentation

Introduction

The MyTimeTable API allows you to retrieve timetables and alter user's MyTimetable profiles. Most of the data in MyTimetable can be accessed using the API, and the possibilities are expanded continuously. The API is based on REST principles, and is currently on version 0. Requests can be done using the various available HTTP verbs (GET, POST, PUT and DELETE).

The API documentation uses the variable **`$base_url`** to refer to the base URL of MyTimetable. This URL depends on the customer and can be found on the Customer-specific API information page.

Not all MyTimetable sites have enabled API access. Some customers choose to enable API access only for certain trusted applications. Please contact us if you have questions about the API or if you would like to receive an access token for a certain API.

Data structure

A student's timetable is made up of one or more individual timetables. These timetables are all of a certain type. The exact meaning of a type depends on the customer (see Customer-specific API information), but the table below gives an idea of the options:

Type	Meaning
student	Personal timetable of a student (when available, most institutions do not have personal timetables available in their timetabling application).
staff	Personal timetable of a staff member.
module	Timetable for a module or course.
pos	Timetable for a programme of study (group of modules).
posss	Timetable for a programme of study (group of studentsets and associated modules). This is a different implementation that the *pos* type, and the exact type available depends on the institution.
studentset	Timetable for a certain group of students (usually the timetable for a class or study group).
studentsetgroup	Timetable for a group of student sets (most customers use either studentset or studentsetgroup to the outside world).
modulepos	Timetable for a certain module, but limited to the studentsets of a certain study programme. This is used if a module is shared between study programmes, but contains activities that are specific to a certain study programme.

user	Personal timetable for a user (can be both a student or a staff member).
location	Timetable for a certain (class)room.
zone	Timetable for a building or other grouping of rooms.
tag	Timetable for all events that are marked with a specific tag.

The API gives the option to either query a user's complete personal timetable, based on his or her MyTimetable subscriptions, or a single timetable which is not based on a user's profile. The first option always requires a certain form of authentication, whereas the second option may be available without any authentication or only requires an API token. An example workflow for both scenarios:

- **Personal timetable**
  - Request access to the user's personal timetable by using OAuth, or use impersonation using an API token (if available).
  - Retrieve a list of a user's subscriptions using the **`/subscriptions`** call.
  - Retrieve the timetable of a user using the **`/timetable`** call.
  - Add or remove subscriptions to a user's profile by using the **`/subscriptions`** calls. To retrieve a list of timetables that can be added, one can use the **`/timetables`** call.
- **Single timetable**
  - Retrieve a list of timetables for a certain type by using the **`/timetables`** call.
  - The list of the previous call can be filtered using several filter options. The values for these filters can be retrieved using the **`/timetablefilters`** call.
  - After choosing the right timetable, the contents of the timetable can be retrieved using the **`/timetables/{key}`** call.

A timetable always consists of a list of **Event** objects and is sorted by event start and end date. The events contain the following values:

Name	Description
activityDescription	The description of the activity, could be the course, module or activity name. This should be shown to the user as 'main description'.
moduleCode	Some customers have a separate module code available which is a (semi-) unique identifier for a certain course of module.
activityTypeName	The name of the activity type this event belongs to. Key value which can be used to group and colour activities. Should not be shown to the end-user directly.
activityTypeDescription	The description of the activity type of this event. This value should be shown to the user and indicates whether the activity is a lecture, practical, exam, etc. Not all institutions provide (sensible) activity type information.
startDate	Start date and time of the event. See below for the date format.
endDate	End date and time of the event. See below for the date format.
studentSets	List of student sets (e.g., classes, groups) associated with the event.
locations	List of locations associated with the event (consists of identifier, name, key, capacity, url and avoidConcurrencyLocationIds (MyTimetable v2.5+, list of id's of locations that are also occupied if this location is occupied)).
staffMembers	List of staff members associated with the event.
notes	Some notes or additional information about the event, not all customers use this field.
timetableKeys	Contains the list of subscriptions (timetables) this event is part of (MyTimetable v2.5+)
highlighted	True iff the event should be highlighted in the interface (reason is customer or department specific) (MyTimetable v2.5+)
tags	List of tags assigned to the event (MyTimetable 2.7+)

New fields may be added to objects without incrementing the version number. See API Documentation#Version numbering for details.

Input and output format

The resources support both JSON and XML output. By default, the returned format depends on the HTTP Accept header. Use `application/json` for JSON and `application/xml` for XML. It is also possible to specify the output format in the URL, by appending `.json` or `.xml` to the resource name, before the query string. Dates and times are returned in ISO 8601 format (`yyyy-MM-ddTHH:mm:ss±HH:mm`) when retrieving results in XML, and in Unix timestamp format when retrieving JSON (in milliseconds, instead of the default seconds).

In the future, the API will only support JSON output, so we strongly recommend to implement your API client based on the JSON output.

Output can be available in multiple languages. Use the `Accept-Language` HTTP header to specify the locale, for example `nl-NL`. When the specified locale is not available, the default locale will be used.

Input can be provided in the form of HTTP GET URL parameters (which should be URL encoded properly) or an HTTP POST/PUT body that is in `application/x-www-form-urlencoded`. Most parameters are optional, when this is not the case this is mentioned in the documentation. Dates and times can be specified in one of the following format:

- 'today' or 'tomorrow' for the current or next day at 0:00;
- Unix timestamp format (in milliseconds sinds 1970, so don't forget to multiply/divide by 1000);
- ISO 8601 format (`yyyy-MM-ddTHH:mm:ss±HH:mm`): elements on the right side (time, time zone) are optional;
- Legacy format (`yyyy/MM/dd HH:mm`): the time is optional.

Authentication and authorisation

Depending on the wishes of the institution, the MyTimetable API can require several forms of authentication and authorisation.

First of all, it can be necessary for the client application to provide some form of authentication. This makes sure only authorised applications can access the API data. This authorisation is usually done using an API token. Next to the client authentication, sometimes the client application wants to access the personal data of a user. The user can either provide authorise the application to access this data, using OAuth, or the client can use a special API token with elevation privileges. Some MyTimetable environment only allow API access for requests that can be linked to a real user. In these cases an OAuth token is always necessary.

In the requests below we have not included the authorisation information (like the `access_token` query parameter or the `apiToken` header). The page with Customer-specific API information lists the requirements of the different MyTimetable environments available.

Recommendations

Please keep some things in mind when using the API:

- The API-token is a secret, and should only be used in back-end services. Do not include it in your end-user application.
- Enable GZIP-compression in your API library. Support for non-compressed responses may be dropped in the future.
- Make sure your client supports TLS v1.2, older versions may not be available.
- Do not cache data for long periods of time (or at all), but do make sure end-users cannot trigger useless API requests (e.g., cache while the user is in the same view, and do not let the user refresh 100's of times by simply spamming a button).
- The schema may change as fields may be added. Be sure to ignore unknown fields in your data mapper.
- Do not 'scrape' the API and store data locally. It causes a high load and stale data.
- Do not perform unnecessarily large requests or a large number of requests. E.g., if you want to show the upcoming 5 activities of a location, just request the upcoming 5 activities and not the timetable for the whole year. Also, do not refresh that view every minute, as the timetable will not change every minute.
- In the future, the API will only support JSON output, so we strongly recommend to implement your API client based on the JSON output.

Version numbering

The API uses a version numbering scheme and the base URL of the API is: `$base_url/api/v$version`. The current API version is version 0. Non-breaking changes (like the addition of extra information or methods) will be done without incrementing the version number. On breaking changes the version number will be increased. Multiple API versions may exist at the same time. Eveoh will try to support older API versions when technically and economically feasible, but users are encouraged to always use the latest API version.

Multiple databases

Multiple datasources with timetabling data may be available (e.g., when retrieving data from multiple years). By specifying a GET parameter **d**, it is possible to select a certain database for the request in certain requests. The list of possible databases can be retrieved with the API Documentation#/databases call. If no database is specified, the default database will be used, which is usually the most sensible option. For older MyTimetable versions (<v2.4) it is necessary to select the right database with every call, if not using the default database.

Resources

/timetabletypes

Method: GET

Description: Query all possible timetable types, these types serve as input for the **/timetables**, **/timetable** and **/timetablefilters** calls. See API Documentation#Data structure for a discussion on the meaning of the types. MyTimetable versions before v2.3 return all possible types supported by the back-end, in that case consult Customer-specific API information to find out what types the customer uses.

Parameters: none

Request URL

```
$base_url/api/v0/timetabletypes
```

Example response body

```
{
    "timetableType": [
        "location",
        "zone",
        "module",
        "student",
        "staff",
        "pos",
        "posss",
        "studentset",
        "studentsetgroup",
        "modulegroup",
        "user",
                "tag"
    ]
}
```

/timetabletypesdetails *(since MyTimetable 3.0)*

Method: GET

Description: Query all possible timetable types and return them including some details regards naming, whether or not they can include child timetables and their ordering in the menus.

Parameters: none

Request URL

```
$base_url/api/v0/timetabletypesdetails
```

Example response body

```
{
    "timetableTypes": [
        {
            "childOptionSelectable": false,
            "description": "Course",
            "name": "module",
            "optionSelectable": false,
            "parent": false,
            "weight": 1
        },
        {
            "childOptionSelectable": false,
            "description": "Programme of study",
            "name": "pos",
            "optionSelectable": false,
            "parent": true,
            "weight": 2
        },
                ...
}
```

/timetables

Method: GET

Description: Query a list of timetables for a specified type. The filters that can be specified can be retrieved using the **/timetablefilters** call. Returns a list of descriptions and identifiers. The identifiers can be used in subsequent calls to **/timetables/{key}** and **/subscriptions/{key}**.

Parameters

Name	Description	Default value

type (required)	The type of the timetables, e.g. 'module'.	-
`limit`	The maximum amount of records returned.	0
`offset`	Starting position of the records returned (e.g. offset=5 skips the first 5 records).	0
`*Filter`	Filter results on a specific filter. If the filtername, as returned by the `/timetablefilters` call is `department`, the parameter name is `departmentFilter`.	-
q	Search string to filter results on (available from MyTimetable v2.3+).	-
d	Datasource to search in. Highest priority datasource by default (usually the most current database).	Highest priority datasource

Example request URL

```
$base_url/api/v0/timetables?type=module
```

Example response body

```
{
    "timetable": [
        {
            "description": "VMT for Informatics",
            "value": "2011!module!49CBD432F01E8C5057B344E02604A710"
        }
    ]
}
```

/timetables/{key}

Method: GET

Description: Query the schedule for the specified timetable identifier. The `activityType*` properties are deprecated since MyTimetable 3.4, in favour of the `eventType` object.

Parameters

Name	Description	Default value
key (required)	The key corresponding with the specific timetable, as retrieved using the **/timetables** call.	-
`fetchBy`	Use 'hostKey' to retrieve the events by it's timetable host key instead of timetable key. Do not include this parameter to retrieve by timetable key. (MyTimetable v2.3 and higher)	-
`type`	The type of the timetable to retrieve, only used in combination with `fetchBy=hostKey`.	
`startDate`	The start date to retrieve events from, or 'today' for the current date at 0:00, or 'tomorrow' for tomorrow at 0:00, or none for the beginning of times.	-
`endDate`	The end date to retrieve events till, or 'today' for the current date at 0:00, or 'tomorrow' for tomorrow at 0:00, or none for the end of times.	-
`limit`	The maximum amount of records returned, 0 for no limit.	0

Example request URLs

```
$base_url/api/v0/timetables/2011!module!49CBD432F01E8C5057B344E02604A710
```

Example response body

```
[
    {
        "activityDescription": "Tsjechische letterkunde 2",
        "activityTypeColor": "#0000ff",
        "activityTypeDescription": "Werkcollege",
        "activityTypeName": "Werkcollege",
        "allDay": false,
        "conferenceUrl": null,
        "customAttributes": {},
        "department": {
            "id": "2010!1E65DFEB58C56AAF042109ECF86B3149",
            "name": "Department of Languages"
        },
        "draft": false,
        "endDate": 1355320800000,
        "enrolled": null,
        "eventType": {
            "color": "#0000ff",
            "description": "Lecture",
            "eventTypeGroup": {
                "description": "Group Session",
                "id": "Group Session"
            },
            "id": "lecture"
        },
        "filtered": false,
        "highlighted": false,
        "locations": [
            {
                "avoidConcurrencyLocationIds": [],
                                "bookUrl": null,
                "capacity": 118,
                "customAttributes": {},
                "examCapacity": null,
                "capacity": 16,
                "customAttributes": {},
                "id": "492E26C5A6F3BDE74C11240C046C3257",
                "key": "908K04SEM1",
                "name": "Bungehuis K.04",
                "sipAddress": null,
                "url": null
            }
        ],
        "moduleCode": "UvA/FGW/TLK_133214266",
        "notes": "Group 14 only",
        "notes2": "",
        "notes3": "",
        "recorded": false,
        "recordingUrl": null,
        "staffMembers": [
            "dr. E.R.G. Metz"
        ],
        "startDate": 1355313600000,
        "students": [],
        "studentSets": [],
                "tags": [
            {
                "key": "weblecture",
                "name": "Web Lecture"
            }
        ],
        "timetableKeys": [
            "2010!module!40B7FB6233A4AD9834C7B999276851F5"
        ]
    }
]
```

/timetables/{key}/options *(since MyTimetable 3.1)*

Method: GET

Description: Query the available options for the specified timetable. These options can be set on the subscription using the PUT call on a subscription. The `allEquivalent` boolean in the output will be true if any choice of options results in the same timetable.

Parameters

Name	Description	Default value
key (required)	The key corresponding with the specific timetable, as retrieved using the **/timetables** or **/subscriptions** call.	-

Example request URLs

```
$base_url/api/v0/timetables/2011!module!49CBD432F01E8C5057B344E02604A710/options
```

Example response body

```
{
    "allEquivalent": false,
    "options": [
        {
            "id": "3772C4D1BFA1033EDD5E0539C89DD331",
            "name": "MODULE A - GROEP 1"
        },
        {
            "id": "3772C4D1BFA1033EDD5E0539C89DD336",
            "name": "MODULE A - GROEP 2"
        },
        {
            "id": "3772C4D1BFA1033EDD5E0539C89DD334",
            "name": "MODULE A - GROEP 3"
        },
        {
            "id": "3772C4D1BFA1033EDD5E0539C89DD335",
            "name": "MODULE A - GROEP 4"
        },
        {
            "id": "3772C4D1BFA1033EDD5E0539C89DD333",
            "name": "MODULE A - GROEP 5"
        }
    ]
}
```

/timetables/{parent}/{key}/options *(since MyTimetable 3.1)*

Method: GET

Description: Query the available options for the specified child of a timetable. These options can be set on the subscription using the PUT call on a subscription. The `allEquivalent` boolean in the output will be true if any choice of options results in the same timetable.

Parameters

Name	Description	Default value
key (required)	The key of the child subscription you wish to retrieve the options for.	-
`parent` (required)	The key of the parent of the subscription.	-

Example request URLs

```
$base_url/api/v0/timetables/2011!pos!49CBD432F01E8C5057B344E02604A710/6473C4D1BFD103FEDD5E0539C89DD334/options
```

Example response body

```
{
    "allEquivalent": false,
    "options": [
        {
            "id": "3772C4D1BFA1033EDD5E0539C89DD331",
            "name": "MODULE A - GROEP 1"
        },
        {
            "id": "3772C4D1BFA1033EDD5E0539C89DD336",
            "name": "MODULE A - GROEP 2"
        },
        {
            "id": "3772C4D1BFA1033EDD5E0539C89DD334",
            "name": "MODULE A - GROEP 3"
        },
        {
            "id": "3772C4D1BFA1033EDD5E0539C89DD335",
            "name": "MODULE A - GROEP 4"
        },
        {
            "id": "3772C4D1BFA1033EDD5E0539C89DD333",
            "name": "MODULE A - GROEP 5"
        }
    ]
}
```

/timetable

Method: GET

Description: Query the personal timetable for the current user (as identified by the OAuth token or elevated API token). The `activityType*` properties are deprecated since MyTimetable 3.4, in favour of the `eventType` object.

Parameters

Name	Description	Default value
`startDate`	The start date to retrieve events from, or 'today' for the current date at 0:00, or 'tomorrow' for tomorrow at 0:00, or none for the beginning of times.	-
`endDate`	The end date to retrieve events till, or 'today' for the current date at 0:00, or 'tomorrow' for tomorrow at 0:00, or none for the end of times.	-
`limit`	The maximum amount of records returned, 0 for no limit.	0
`type`	Specifies the type of the subscriptions to include when returning the timetable, can be specified multiple times, useful to exclude zone and location subscriptions when requesting the timetable. If not specified all subscriptions will be included in the result.	-
`excludeResourceTimetables`	Set this to `true` to exclude any resource (location, zone, equipment) timetables. Generally you will want this for portal integrations etc. (added in MyTimetable 3.1)	`false`

Example request URL

```
$base_url/api/v0/timetable?startDate=2012-07-10&endDate=2012-07-10&type=location&type=zone
```

Example response body

```
[
    {
        "activityDescription": "Tsjechische letterkunde 2",
        "activityTypeColor": "#0000ff",
        "activityTypeDescription": "Werkcollege",
        "activityTypeName": "Werkcollege",
        "allDay": false,
        "conferenceUrl": null,
        "customAttributes": {},
        "department": {
            "id": "2010!1E65DFEB58C56AAF042109ECF86B3149",
            "name": "Department of Languages"
        },
        "draft": false,
        "endDate": 1355320800000,
        "enrolled": null,
        "eventType": {
            "color": "#0000ff",
            "description": "Lecture",
            "eventTypeGroup": {
                "description": "Group Session",
                "id": "Group Session"
            },
            "id": "lecture"
        },

        "filtered": false,
        "highlighted": false,
        "locations": [
            {
                "avoidConcurrencyLocationIds": [],
                                "bookUrl": null,
                "capacity": 118,
                "customAttributes": {},
                "examCapacity": null,
                "capacity": 16,
                "customAttributes": {},
                "id": "492E26C5A6F3BDE74C11240C046C3257",
                "key": "908K04SEM1",
                "name": "Bungehuis K.04",
                "sipAddress": null,
                "url": null
            }
        ],
        "moduleCode": "UvA/FGW/TLK_133214266",
        "notes": "Group 14 only",
        "notes2": "",
        "notes3": "",
        "recorded": false,
        "recordingUrl": null,
        "staffMembers": [
            "dr. E.R.G. Metz"
        ],
        "startDate": 1355313600000,
        "students": [],
        "studentSets": [],
                "tags": [
            {
                "key": "weblecture",
                "name": "Web Lecture"
            }
        ],
        "timetableKeys": [
            "2010!module!40B7FB6233A4AD9834C7B999276851F5"
        ]
    }
]
```

/timetablefilters

Method: GET

Description: Query all filters available for a specific timetable type. Using the results it is possible to filter the output of this call and of the **/timetables** call. Returns a list of possible filtertypes. For each type a list of possible filter descriptions and identifiers is provided.

Parameters

Name	Description	Default value
`type` **(required)**	The type of the timetables to retrieve the filters for.	-
`*Filter`	Filter results on a specific filter. If the filtername, as returned by the `/timetablefilters` call is `department`, the parameter name is `departmentFilter`.	-
d	Datasource to search in. Highest priority datasource by default (usually the most current database).	Highest priority datasource

Example request URL

```
$base_url/api/v0/timetablefilters?type=pos
```

Example response Body

```
{
    "filterattribute": [
        {
            "option": [
                {
                    "name": " Faculty of Informatics",
                    "value": "646ADCA666D4A88402CA46C26A73803C"
                },
                {
                    "name": " Faculty of Law",
                    "value": "646ADCA666D4A88402CA46C26A738046"
                }
            ],
            "type": "department"
        }
    ]
}
```

/subscriptions

Method: GET

Description: Retrieves all subscriptions for the current user (as identified by the OAuth token or elevated API token). For older MyTimetable versions (pre-2.4) only the subscriptions of the currently selected datasource are returned. Newer MyTimetable versions return all subscriptions, use the `database` value in the returned subscriptions to determine which datasource a certain subscription belongs to. Since MyTimetable 3.0 the type name of a subscription is also returned.

Parameters:

Name	Description	Default value
`type`	Specifies the type of the subscriptions to include, can be specified multiple times, useful to exclude zone and location subscriptions when requesting a personal list of subscriptions. If not specified all subscriptions will be included in the result.	-
`excludeResourceTimetables`	Set this to `true` to exclude any resource (location, zone, equipment) timetables. Generally you will want this for portal integrations etc. (added in MyTimetable 3.1)	`false`

Request URL

```
$base_url/api/v0/subscriptions
```

Example response Body

```
[
        {
                "key":"2011!module!49CBD432F01E8C5057B344E02604A711",
        "database": "2015",
                "description":"Basic chemical reactions",
                "childSubscriptions":[],
                "enabled":false,
        "enabledOptions": [
            {
                "id": "3772C4D1BFA1033EDD5E0539C89DD335",
                "name": "MODULE A - GROEP 4"
            }
        ],
                "removable":true,
        "type": "pos"
        }
]
```

Method: DELETE

Description: Delete all subscriptions

Parameters: none

Request URL

```
$base_url/api/v0/subscriptions
```

Response code: 204 (No content)

/subscriptions/{key}

Method: POST

Description: Add a new subscription

Parameters

Name	Description	Default value
key (required)	The key of the timetable you wish to add.	-
option	The id of the option to select for this timetable, retrieved using the **/timetables/{key}/options** call. Can be specified multiple times, or not specified to select all options (default). Should be specified as a form parameter. Available since MyTimetable 3.1.	None (all options selected)

Example request URL

```
$base_url/api/v0/subscriptions/2011!pos!0F2C927DF37F3A2BEF1F1713768E4EE6
```

Response code: 204 (No content)

Method: DELETE

Description: Delete a subscription.

Parameters

Name	Description	Default value
key (required)	The key of the subscription you wish to delete.	-

Example request URL

```
$base_url/api/v0/subscriptions/2011!pos!0F2C927DF37F3A2BEF1F1713768E4EE6
```

Response code: 204 (No content)

Method: PUT

Description: Edit a subscription. Use boolean values to trigger the subscription on or off.

Parameters

Name	Description	Default value
key (required)	The key of the subscription you wish to edit.	-
`state`	Boolean value indicating the state of the subscription.	`false`

Example request URL

```
$base_url/api/v0/subscriptions/2011!pos!0F2C927DF37F3A2BEF1F1713768E4EE6
```

Response Code: 204 (No content)

/subscriptions/{key}/options (since MyTimetable 3.1)

Method: PUT

Description: Set selected options of a subscription. Specifying all possible options will cause the list of selected options to be cleared, showing the complete timetable to the user.

Parameters

Name	Description	Default value
key (required)	The key of the subscription you with to set the options for.	-
option (required)	The id of the option, retrieved using the **/timetables/{key}/options** call. Can be specified multiple times. Should be specified as a form parameter.	-

Example request URL

```
$base_url/api/v0/subscriptions/2011!pos!0F2C927DF37F3A2BEF1F1713768E4EE6/options
```

Response code: 200 (OK)

Method: DELETE

Description: Clear the list of selected options of a subscription, which will cause all activities of the timetable to be shown to the user.

Parameters

Name	Description	Default value
key (required)	The key of the subscription you with to clear the options for.	-

Example request URL

```
$base_url/api/v0/subscriptions/2011!pos!0F2C927DF37F3A2BEF1F1713768E4EE6/options
```

Response code: 204 (No content)

/subscriptions/{parent}/{key}

Method: PUT

Description: Edit a subscription nested under a parent. Use boolean values to trigger the subscription on or off.

Parameters

Name	Description	Default value
key (required)	The key of the subscription you wish to edit.	-
`parent` (required)	The key of the parent of the subscription.	-
`state`	Boolean value indicating the state of the subscription.	`false`

Example request URL

```
$base_url/api/v0/subscriptions/2011!location!D2D0C6A8E334B242289FBAA7F69759C3/2011!module!
49CBD432F01E8C5057B344E02604A773
```

Response Code: 204 (No content)

/subscriptions/{parent}/{key}/options (since MyTimetable 3.1)

Method: PUT

Description: Set selected options of a subscription nested under a parent. Specifying all possible options will cause the list of selected options to be cleared, showing the complete timetable to the user.

Parameters

Name	Description	Default value
key (required)	The key of the subscription you with to set the options for.	-
`parent` (required)	The key of the parent of the subscription.	-
option (required)	The id of the option, retrieved using the **/timetables/{parent}/{key}/options** call. Can be specified multiple times. Should be specified as a form parameter.	-

Example request URL

```
$base_url/api/v0/subscriptions/2011!pos!0F2C927DF37F3A2BEF1F1713768E4EE6/646ADCA666D4A88402CA46C26A738046
/options
```

Response code: 200 (OK)

Method: DELETE

Description: Clear the list of selected options of a subscription nested under a parent, which will cause all activities of the timetable to be shown to the user.

Parameters

Name	Description	Default value

key (required)	The key of the subscription you with to clear the options for.	-
`parent` (required)	The key of the parent of the subscription.	-

Example request URL

```
$base_url/api/v0/subscriptions/2011!pos!0F2C927DF37F3A2BEF1F1713768E4EE6/646ADCA666D4A88402CA46C26A738046
/options
```

Response code: 204 (No content)

/user

Method: GET

Description: Query information on the current user.

Requires scope `username` or `user_read` (since MyTimetable 2019.21, includes user roles). To retrieve the iCalendar feed URL for the current user, scope `profile_read` is required.

Parameters: none

Request URL

```
$base_url/api/v0/user
```

Example response body

```
{
    "feedUrl": "http://myuniversity/ical?eu=FKW92Fawier&t=556cf8a1-704a-4997-bc4d-5acf395eccaf",
    "username": "testuser",
    "roles": [
        "ROLE_STUDENT",
        "ROLE_MEMBER"
    ]
}
```

/weeklabels *(deprecated since MyTimetable 2.7, replaced by: /weeklabelmaps)*

Method: GET

Description: Query the weeklabels. This provides custom labels for the weeks when available. The response includes a value (the custom label) and a week (year - 1900 + the ISO week number).

Parameters: none

Request URL

```
$base_url/api/v0/weeklabels
```

Example response body

```
{
    weeklabel: [
        {
            "week": 11223,
            "value": "23"
        },
        {
            "week": 11224,
            "value": "24"
        }
    ]
}
```

/weeklabelmaps *(since MyTimetable 2.7)*

Method: GET

Description: Query the weeklabel maps. This provides the available maps with custom labels for the weeks. Per weeklabel map, the response includes a key and a boolean which indicates whether the map is the default weeklabel map for the requested user.

Parameters: none

Request URL

```
$base_url/api/v0/weeklabelmaps
```

Example response body

```
{
    "weeklabelmaps": [
        {
            "key": "SYLLABUS",
            "isDefault": true
        }
    ]
}
```

/weeklabelmaps/{key} *(since MyTimetable 2.7)*

Method: GET

Description: Retrieve the specified weeklabel map. This provides the key, a boolean which indicates if this is the default weeklabel map for the requested user and the actiual map with custom labels for the weeks.

Parameters

Name	Description	Default value
key (required)	The key of the weeklabel you wish to retrieve.	-

Example request URL

```
$base_url/api/v0/weeklabelmaps/SYLLABUS
```

Example response body

```
{
    "key": "SYLLABUS",
    "isDefault": true,
    "weeklabels": [
        {
            "year": 2014,
            "week": 4,
            "description": "wk 4"
        },
        {
            "year": 2014,
            "week": 5,
            "description": "wk 5"
        }
    ]
}
```

/databases *(deprecated since MyTimetable 3.1, replaced by: /databasedetails)*

Method: GET

Description: Query the available datasources, see API Documentation#Multiple databases for more information. For MyTimetable 3.1+, please use the `/databasedetails` call, which also provides the labels of the databases.

Parameters: none

Request URL

```
$base_url/api/v0/databases
```

Example response body

```
{
    "database": [
        "2011"
    ]
}
```

/databasedetails *(since MyTimetable 3.1)*

Method: GET

Description: Query the available datasources, see API Documentation#Multiple databases for more information. The key (to be used in the query string of other calls) and the label are returned.

Parameters: none

Request URL

```
$base_url/api/v0/databasedetails
```

Example response body

```
{
    "databases": [
        {
            "key": "2015",
            "label": "2015/2016"
        },
        {
            "key": "2016",
            "label": "2016/2017"
        },
        {
            "key": "TimetableDB",
            "label": "Custom"
        }
    ]
}
```

/eventtypegroups *(since MyTimetable 3.0)*

Method: GET

Description: Query the event type groups and event filtering settings.

Parameters: none

Request URL

```
$base_url/api/v0/eventtypegroups
```

Example response body

```
{
  "filteringEnabled": true,
  "eventtypegroups": [
    {
      "id": "Excursie",
      "description": "Excursie",
      "isFilteredByDefault": false,
      "isFiltered": false,
      "isFilteredForSync": false
    },
    {
      "id": "Hoorcollege",
      "description": "Hoorcollege",
      "isFilteredByDefault": false,
      "isFiltered": false,
      "isFilteredForSync": false
    },
    {
      "id": "Other",
      "description": "Other",
      "isFilteredByDefault": false,
      "isFiltered": false,
      "isFilteredForSync": false
    }
  ]
}
```

Method: PATCH

Parameters: none

Request URL

```
$base_url/api/v0/eventtypegroups
```

Request Content-Type

application/x-www-form-urlencoded

Example request body

```
eventTypeGroups%5BExcursie%5D%5BisFiltered%5D=true&eventTypeGroups%5BHoorcollege%5D%5BisFiltered%
5D=false&eventTypeGroups%5BOther%5D%5BisFiltered%5D=false
```

Decoded example request body

```
eventTypeGroups[Excursie][isFiltered]=true
eventTypeGroups[Hoorcollege][isFiltered]=false
eventTypeGroups[Other][isFiltered]=false
```

Response code

204 (No content), 403 (Forbidden) if filtering is disabled.

/terms *(since MyTimetable 3.4)*

Method: GET

Description: Query the available terms. Returns a list of terms, each containing of a unique id, a key (may be null), a name and a list of date ranges.

Parameters: none

Request URL

```
$base_url/api/v0/terms
```

Example response body

```
{
    "terms": [
        {
            "dateRanges": [
                {
                    "endDate": 1513983600000,
                    "startDate": 1504476000000
                },
                {
                    "endDate": 1535925600000,
                    "startDate": 1514847600000
                }
            ],
            "id": "2017!492E26C5A6F3BDE74C11240C046C2FB1",
            "key": null,
            "name": "[2017/2018] All year"
        },
        {
            "dateRanges": [
                {
                    "endDate": 1535925600000,
                    "startDate": 1504476000000
                }
            ],
            "id": "TermTime!13",
            "key": "All",
            "name": "[Term Time] All year"
        }
    ]
}
```

/user/messages *(since MyTimetable 4.0)*

Note: this API endpoint always returns `application/json` as Content Type. XML is not supported.

Method: GET

Description: Query the active messages for the current user (as identified by the OAuth token or elevated API token). Requires scope `messages_read`.

Parameters: none

Request URL

```
$base_url/api/v0/user/messages
```

Request Accept header

`application/json`

`application/vnd.mytimetable.html+json` to include a property `bodyHtml` for each returned message

Example response body

```
[
    "messages": [
            {
                "id": 1,
                "title": "This is the title of a non-expiring message targeted to students marked as
important.",
                "body": "# This is a header formatted with Markdown",
                "important": true,
                "publicationDate": 1550155320000,
                "expirationDate": null,
                "targetRoles": [
                    "ROLE_STUDENT"
                ],
                "bodyHtml": "<h1>This is a header formatted with Markdown</h1>"
            },
            {
                "id": 2,
                "title": "This is an expiring message targeted to all users.",
                "body": "Some content.",
                "important": false,
                "publicationDate": 1550155320000,
                "expirationDate": 1553252400000,
                "targetRoles": [],
                "bodyHtml": "<p>Some content.</p>"
            }
    ]
]
```

Response code

200 (OK)

/messages *(since MyTimetable 4.0)*

Note: this API endpoint always returns `application/json` as Content Type. XML is not supported.

Method: GET

Description: Retrieves messages. Requires API token scope `ROLE_API_MESSAGES`.

Parameters:

Name	Description	Default value
`limit`	The maximum amount of records returned, 0 for no limit.	`0`
`offset`	Starting position of the records returned (e.g. offset=5 skips the first 5 records).	`0`
`q`	Search string to filter results on.	-

Request URL

```
$base_url/api/v0/messages
```

Request Accept header

`application/json`

`application/vnd.mytimetable.html+json` to include a property `bodyHtml` for each returned message

Example response Body

```
[
    "messages": [
            {
                "id": 1,
                "title": "This is the title of a non-expiring message targeted to students and staff marked as important.",
                "body": "# This is a header formatted with Markdown",
                "important": true,
                "publicationDate": 1550155320000,
                "expirationDate": null,
                "targetRoles": [
                    "ROLE_STUDENT",
                    "ROLE_STAFF"
                ],
                "bodyHtml": "<h1>This is a header formatted with Markdown</h1>"
            },
            {
                "id": 2,
                "title": "This is a draft message targeted to all users.",
                "body": "Some content.",
                "important": false,
                "publicationDate": null,
                "expirationDate": null,
                "targetRoles": [],
                "bodyHtml": "<p>Some content.</p>"
            }
    ]
]
```

Response code

200 (OK)

Method: POST

Parameters: none

Request body form values:

Name	Description	Default value
`title (required)`	Title of the message.	-
`body (required)`	Body of the message. Markdown formatting can be used.	-
`important`	Sets if a message should be considered important.	false
`publicationDate`	Publication date. The message will be shown as of this date. See below for the date format.	null
`expirationDate`	Expiration date. The message will be shown until this message, or forever if set to null. See below for the date format.	null
`targetRoles`	User roles to target this message to. If not set, all users can view the message.	[ ]

Request URL

```
$base_url/api/v0/messages
```

Request Accept header

`application/json`

`application/vnd.mytimetable.html+json` to include a property `bodyHtml` for the returned message

Request Content-Type header

application/x-www-form-urlencoded

Example request body

```
title=This%20is%20the%20title%20of%20a%20non-expiring%20message%20targeted%20to%20students%20marked%20as%
20important.&body=%23%20This%20is%20a%20header%20formatted%20with%20Markdown%0AThis%20is%20some%20content.
&important=true&publicationDate=2019-01-02&expirationDate=&targetRoles=ROLE_STAFF&targetRoles=ROLE_STUDENT
```

Decoded example request body

```
title=This is the title of a non-expiring message targeted to students marked as important.
body=# This is a header formatted with Markdown
This is some content.
important=true
publicationDate=2019-12-12
expirationDate=
targetRoles=ROLE_STAFF
targetRoles=ROLE_STUDENT
```

Example response Body

```
{
    "id": 1,
    "title": "This is the title of a non-expiring message targeted to students marked as important.",
    "body": "# This is a header formatted with Markdown
This is some content.",
    "important": true,
    "publicationDate": 1552258800000,
    "expirationDate": null,
    "targetRoles": [
        "ROLE_STAFF",
        "ROLE_STUDENT"
    ],
    "bodyHtml": "<h1>This is a header formatted with Markdown</h1>
<p>This is some content.</p>"
}
```

Response code

201 (Created) if the message is created, 400 (Bad Request) if input validation fails

/messages/{id} *(since MyTimetable 4.0)*

Note: this API endpoint always returns `application/json` as Content Type. XML is not supported.

Method: GET

Description: Retrieves the message with the given ID. Requires API token scope `ROLE_API_MESSAGES`.

Parameters:

Name	Description	Default value
`id`	The ID of the message to retrieve.	-

Example request URL

```
$base_url/api/v0/messages/1
```

Request Accept header

`application/json`

`application/vnd.mytimetable.html+json` to include a property `bodyHtml` for the returned message

Example response Body

```
{
    "id": 1,
    "title": "This is the title of a non-expiring message targeted to students marked as important.",
    "body": "# This is a header formatted with Markdown",
    "important": true,
    "publicationDate": 1550155320000,
    "expirationDate": null,
    "targetRoles": [
        "ROLE_STUDENT"
    ],
    "bodyHtml": "<h1>This is a header formatted with Markdown</h1>"
}
```

Response code

200 (OK), 404 (Not Found) if the message does not exist

Method: DELETE

Description: Deletes the message with the given ID.

Parameters:

Name	Description	Default value
`id (required)`	ID of the message to delete.	-

Example request URL

```
$base_url/api/v0/messages/1
```

Response code

204 (No Content) if the message is successfully deleted, or 404 (Not Found) when message with given ID does not exist.

Method: PATCH

Parameters:

Name	Description	Default value
`id (required)`	The ID of the message to patch.	-

Request body form values:

Only parameters set in the body of the request will be updated.

Name	Description	Default value
`title`	Title of the message.	-
`body`	Body of the message. Markdown formatting can be used.	-
`important`	Sets if a message should be considered important.	false
`publicationDate`	Publication date. The message will be shown as of this date. See below for the date format.	null
`expirationDate`	Expiration date. The message will be shown until this message, or forever if set to null. See below for the date format.	null
`targetRoles`	User roles to target this message to. If not set, all users can view the message.	[ ]

Example request URL

```
$base_url/api/v0/messages/1
```

Request Accept header

`application/json`

`application/vnd.mytimetable.html+json` to include a property `bodyHtml` for the returned message

Request Content-Type header

application/x-www-form-urlencoded

Example request body

```
title=This%20is%20the%20edited%20title%20of%20a%20non-expiring%20message%20targeted%20to%20students%20marked%
20as%20important.&body=%23%20This%20is%20an%20edited%20header%20formatted%20with%20Markdown%0AThis%20is%20some%
20content.&important=false&publicationDate=2019-03-
11&expirationDate=&targetRoles=ROLE_STAFF&targetRoles=ROLE_STUDENT
```

Decoded example request body

```
title=This is the edited title of a non-expiring message targeted to students marked as important.
body=# This is an edited header formatted with Markdown
This is some content.
important=false
publicationDate=2019-03-11
expirationDate=
targetRoles=ROLE_STAFF
```

Example response Body

```
{
    "id": 1,
    "title": "This is the title of a non-expiring message targeted to students marked as important.",
    "body": "# This is a header formatted with Markdown
This is some content.",
    "important": true,
    "publicationDate": 1552258800000,
    "expirationDate": null,
    "targetRoles": [
        "ROLE_STAFF"
    ],
    "bodyHtml": "<h1>This is a header formatted with Markdown</h1>
<p>This is some content.</p>"
}
```

Response code

200 (OK), 400 (Bad Request) if input validation fails