Provisioning

Introduction

The provisioning API is an admin-only API used to create users and push synchronisations in MyTimetable. This API can be used in your user provisioning processes to support an push synchronisation 'opt-out' scenario. The API is supported from MyTimetable version 3.0, and only supports JSON output and form-encoded input.

User resources

/users

Method: GET

Description: Query all known usernames.

Parameters: none

Request URL

```
$base_url/api/v0/users
```

Example response body

```
{
    "users": [
        "alice@example.org",
        "bob@example.org",
        "carol@example.org"
    ]
}
```

/users/{username}

Method: GET

Description: Query a specific user.

Parameters

Name	Description	Default value
username (required)	The username to query.	-

Example request URL

```
$base_url/api/v0/users/alice@example.org
```

Example response body

Please note the `attributes` property can be `null`.

```
{
    "username": "alice@example.org",
    "displayName": "Alice",
    "locale": "nl",
    "attributes": [
        {
            "name": "urn:mace:dir:attribute-def:displayName",
            "values": [
                "Alice"
            ]
        },
        {
            "name": "urn:mace:dir:attribute-def:mail",
            "values": [
                "alice@example.org"
            ]
        },
        {
            "name": "urn:mace:dir:attribute-def:eduPersonAffiliation",
            "values": [
                "member",
                "student"
            ]
        },
        {
            "name": "urn:mace:dir:attribute-def:eduPersonPrincipalName",
            "values": [
                "alice@example.org"
            ]
        }
    ],
    "roles": [
        "ROLE_MEMBER",
        "ROLE_STUDENT"
    ]
}
```

Response code

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

/users

Method: POST

Description: Creates a user with the specified username.

Parameters: none

Request body form values:

Name	Description	Default value
username (required)	Username of the user.	-
displayName	Display Name of the user	`null`
locale	Locale of the user, or empty for the default locale.	`null`
attributes	Attribute for the user. This form value can be set multiple times. The format used is attributeName=attributeValue. When multiple values with the same attributeName are given, all attributeValues are persisted. See example request below.	`[]`

roles	Roles of the user. This form value can be set multiple times.	[]

Request URL

```
$base_url/api/v0/users
```

Request Accept header

```
application/json
```

Request Content-Type header

```
application/x-www-form-urlencoded
```

Example request body

```
username=alice%40example.org&displayName=Alice&locale=nl&attributes=urn%3Amace%3Adir%3Aattribute-def%
3AdisplayName%3DAlice&attributes=urn%3Amace%3Adir%3Aattribute-def%3AeduPersonAffiliation%
3Dmember&attributes=urn%3Amace%3Adir%3Aattribute-def%3AeduPersonAffiliation%3Dstudent&attributes=urn%3Amace%
3Adir%3Aattribute-def%3Amail%3Dalice%40example.org&attributes=urn%3Amace%3Adir%3Aattribute-def%
3AeduPersonPrincipalName%3Dalice%40example.org&roles=ROLE_MEMBER&roles=ROLE_STUDENT
```

Decoded example request body

```
username=alice@example.org
displayName=Alice
locale=nl
attributes=urn:mace:dir:attribute-def:displayName=Alice
attributes=urn:mace:dir:attribute-def:eduPersonAffiliation=member
attributes=urn:mace:dir:attribute-def:eduPersonAffiliation=student
attributes=urn:mace:dir:attribute-def:mail=alice@example.org
attributes=urn:mace:dir:attribute-def:eduPersonPrincipalName=alice@example.org
roles=ROLE_MEMBER
roles=ROLE_STUDENT
```

Example response Body

```
{
    "username": "alice@example.org",
    "displayName": "Alice",
    "locale": "nl",
    "attributes": [
        {
            "name": "urn:mace:dir:attribute-def:displayName",
            "values": [
                "Alice"
            ]
        },
        {
            "name": "urn:mace:dir:attribute-def:mail",
            "values": [
                "alice@example.org"
            ]
        },
        {
            "name": "urn:mace:dir:attribute-def:eduPersonAffiliation",
            "values": [
                "member",
                "student"
            ]
        },
        {
            "name": "urn:mace:dir:attribute-def:eduPersonPrincipalName",
            "values": [
                "alice@example.org"
            ]
        }
    ],
    "roles": [
        "ROLE_MEMBER",
        "ROLE_STUDENT"
    ]
}
```

Response code

201 (Created) if the user is created, 400 (Bad Request) if input validation fails, 409 (Conflict) if a user with the given username already exists

/users/{username}

Method: PUT

Description: Ensures a user with the specified username and properties exists.

> Please note that properties of a user can be changed by other actions as well. For example, if you provision the user with two attributes, a subsequent login by the user might overwrite these two attributes with new values or add more attributes (as provided by the authentication provider). Another example is the user changing its locale in the interface. Please keep this in mind when updating user properties.

Parameters

Name	Description	Default value
`username` (required)	The username for which a user profile should be created if it doesn't exist already.	-

Request body form values

Behaviour

If no request body is sent, the following behaviour is used:

- User does not exist: user is created
- User already exists: nothing is changed
- In either case, no response body is returned

If a request body is sent, the following behaviour is used:

- User does not exist: user is created with the properties set as requested via the request body
- User already exists: the properties of the user are updated (or unset) with the properties set as requested via the request body
- In either case, the user is returned as the response body

Name	Description	Default value
displayName	Display name of the user	`null`
locale	Locale of the user, or empty for the default locale.	`null`
attributes	Attribute for the user. This form value can be set multiple times. The format used is attributeName=attributeValue. When multiple values with the same attributeName are given, all attributeValues are persisted. See example request below.	`[]`
roles	Roles of the user. This form value can be set multiple times.	`[]`

Request URL

```
$base_url/api/v0/users/alice@example.org
```

Request Accept header

`application/json`

Request Content-Type header

`application/x-www-form-urlencoded`

Example request body

```
displayName=Alice&locale=nl&attributes=urn%3Amace%3Adir%3Aattribute-def%3AdisplayName%3DAlice&attributes=urn%
3Amace%3Adir%3Aattribute-def%3AeduPersonAffiliation%3Dmember&attributes=urn%3Amace%3Adir%3Aattribute-def%
3AeduPersonAffiliation%3Dstudent&attributes=urn%3Amace%3Adir%3Aattribute-def%3Amail%3Dalice%40example.
org&attributes=urn%3Amace%3Adir%3Aattribute-def%3AeduPersonPrincipalName%3Dalice%40example.
org&roles=ROLE_MEMBER&roles=ROLE_STUDENT
```

Decoded example request body

```
displayName=Alice
locale=nl
attributes=urn:mace:dir:attribute-def:displayName=Alice
attributes=urn:mace:dir:attribute-def:eduPersonAffiliation=member
attributes=urn:mace:dir:attribute-def:eduPersonAffiliation=student
attributes=urn:mace:dir:attribute-def:mail=alice@example.org
attributes=urn:mace:dir:attribute-def:eduPersonPrincipalName=alice@example.org
roles=ROLE_MEMBER
roles=ROLE_STUDENT
```

Example response body

If no request body is sent, no response body is returned.

If a request body is sent, the user is returned as the response body:

```
{
    "username": "alice@example.org",
    "displayName": "Alice",
    "locale": "nl",
    "attributes": [
        {
            "name": "urn:mace:dir:attribute-def:displayName",
            "values": [
                "Alice"
            ]
        },
        {
            "name": "urn:mace:dir:attribute-def:mail",
            "values": [
                "alice@example.org"
            ]
        },
        {
            "name": "urn:mace:dir:attribute-def:eduPersonAffiliation",
            "values": [
                "member",
                "student"
            ]
        },
        {
            "name": "urn:mace:dir:attribute-def:eduPersonPrincipalName",
            "values": [
                "alice@example.org"
            ]
        }
    ],
    "roles": [
        "ROLE_MEMBER",
        "ROLE_STUDENT"
    ]
}
```

Response code

- If no request body is sent: 204 No Content
- If a request body is sent: 201 Created (user newly created), 200 OK (user updated), or 400 (Bad Request) if input validation fails

/users/{username}

Method: DELETE

Description: Deletes the user profile with the specified username if it exists.

Parameters

Name	Description	Default value
username (required)	The username for which a user profile should be deleted if it exists.	-

Example request URL

```
$base_url/api/v0/users/alice
```

Example response body: none (status code 204 indicates success; 404 indicates user not found)

/users/{username}/synchronizations

Method: GET

Description: Query all synchronizations for a user.

Parameters:

Name	Description	Default value
username (required)	The username for which the synchronizations should be queried.	-

Example request URL

```
$base_url/api/v0/users/alice/synchronizations
```

Example response body

```
{
    "synchronizations": [
        {
            "id": 1,
            "type": "ews",
            "username": "alice",
                        "description": "alice@bob.test",
            "enabled": true,
                    "status": "OK",
            "lastResync": 1355320800000
        }
    ]
}
```

/users/{username}/synchronizations

Method: DELETE

Description: Delete all synchronizations for a user.

Parameters:

Name	Description	Default value
username (required)	The username for which the synchronizations should be deleted.	-
unlinkMode (required)	The unlink mode indicates what should happen to the events in the user's calendar: • **UNLINK_ONLY** Leaves all events in place. • **DELETE_FUTURE_EVENTS** Only events which have been pushed with this specific synchronization id and which have a start date in the future are removed. • **DELETE_ALL_EVENTS** All events which have been pushed with this specific synchronization id are removed. • **CLEANUP** All events which have been pushed by this MyTimetable instance are removed, even those with a different synchronization id.	-

Example request URL

```
$base_url/api/v0/users/alice/synchronizations?unlinkMode=DELETE_ALL_EVENTS
```

Example response body: none (status code 204 indicates success)

/users/{username}/synchronizations/{type}

Method: DELETE

Description: Delete synchronizations of a certain type for a certain user.

Parameters:

Name	Description	Default value
username (required)	The username for which the synchronizations should be deleted.	-

type (required)	The synchronization type, which indicates the calendaring service provider.	-
unlinkMode (required)	The unlink mode indicates what should happen to the events in the user's calendar: • **UNLINK_ONLY** Leaves all events in place. • **DELETE_FUTURE_EVENTS** Only events which have been pushed with this specific synchronization id and which have a start date in the future are removed. • **DELETE_ALL_EVENTS** All events which have been pushed with this specific synchronization id are removed. • **CLEANUP** All events which have been pushed by this MyTimetable instance are removed, even those with a different synchronization id.	-

Example request URL

```
$base_url/api/v0/users/alice/synchronizations/ews?unlinkMode=DELETE_ALL_EVENTS
```

Example response body: none (status code 204 indicates success; 404 indicates no synchronizations found; 422 indicates invalid synchronisation type)

Synchronization resources

/synchronizations

Method: GET

Description: Query all synchronizations.

Parameters: none

Request URL

```
$base_url/api/v0/synchronizations
```

Example response body

```
{
    "synchronizations": [
        {
            "id": 1,
            "type": "o365",
            "username": "alice",
                        "description": "alice@bob.test",
            "enabled": true,
                    "status": "OK",
            "lastResync": 1355320800000
        },
        {
            "id": 1,
            "type": "googlecalendar",
            "username": "bob",
                        "description": "bob@google.test",
            "enabled": true,
                    "status": "OK",
            "lastResync": 1355320800000
        }
    ]
}
```

/synchronizations

Method: PUT

Description: Sets up a synchronization with a user's external calendar.

Parameters

Name	Description	Default value

username (required)	The username for which the synchronization should be created.	-
type (required)	The synchronization type, which indicates the calendaring service provider.	-
provisioning_smtpAddress*	The user's mailbox address in Exchange or Office 365.	-

* Only applicable when using a *ProvisioningPreDelegatedEWSLinkingAdapter* or *ProvisioningPreAuthorizedOffice365LinkingAdapter*.

Example request URL

```
$base_url/api/v0/synchronizations
```

Example request body (application/x-www-form-urlencoded)

```
username=alice@example.org&type=ews&provisioning_smtpAddress=alice@bob.test
```

Example response body:

```
{
    "synchronization":
        {
            "id": 1,
            "type": "ews",
            "username": "alice@example.org",
                        "description": "alice@bob.test",
            "enabled": true,
                    "status": "OK",
            "lastResync": 1355320800000
        }
}
```

Response code: 200 OK, 422 Unprocessable Entity (username does not exist or synchronization type is not known), 550 Permission Denied (access not granted to mailbox)

/synchronizations/{id}

Method: DELETE

Description: Deletes the synchronization with the specified id.

Parameters

Name	Description	Default value
id (required)	The id of the synchronization which should be deleted.	-
unlinkMode (required)	The unlink mode indicates what should happen to the events in the user's calendar: • **UNLINK_ONLY** Leaves all events in place. • **DELETE_FUTURE_EVENTS** Only events which have been pushed with this specific synchronization id and which have a start date in the future are removed. • **DELETE_ALL_EVENTS** All events which have been pushed with this specific synchronization id are removed. • **CLEANUP** All events which have been pushed by this MyTimetable instance are removed, even those with a different synchronization id.	-

Example request URL

```
$base_url/api/v0/synchronizations/3?unlinkMode=UNLINK_ONLY
```

Example response body: none (status code 204 indicates success; 404 indicates synchronization not found)

Sample usage

If you want to create a large number of synchronisations, create a text file with the username and email address of the users. Then you can run the following shell script to create the users:

```
while read user mail
do
curl -v -X PUT -H apiToken:xxx https://mytimetable_host/api/v0/users/$user
done < users.txt
```

The following script will create an 'o365' synchronisation for each user:

```
while read user mail
do
curl -v -X PUT -H apiToken:xxx -d username=$user -d type=o365 -d provisioning_smtpAddress=$mail
https://mytimetable_host/api/v0/synchronizations
done < users.txt
```

A sample Powershell script:

```
$username = "username_in_MTT"
$type = "ews"
$smtpAddress = "user@host.nl"
$apiToken = "xxx"

# Create the user in MyTimetable if it does not exist
Invoke-RestMethod https://mytimetable_host/api/v0/users/${username} -Method PUT -Headers @{apiToken=$apiToken}

# Setup a synchronisation for the user
Invoke-RestMethod https://mytimetable_host/api/v0/synchronizations -Method PUT -Headers @{apiToken=$apiToken} -
ContentType "application/x-www-form-urlencoded" -Body @{username=$username;type=$type;
provisioning_smtpAddress=$smtpAddress}

# Delete the synchronisation for the user and remove all events from the calendar
Invoke-RestMethod https://mytimetable_host/api/v0/users/${username}/synchronizations/${type}?
unlinkMode=DELETE_ALL_EVENTS -Method DELETE -Headers @{apiToken=$apiToken}
```