
TeamDrive Agent Documentation

Release 3.2

TeamDrive Systems GmbH

September 09, 2014

CONTENTS

1	Copyright Notice	1
2	Installation	2
2.1	English	2
2.2	Deutsch	3
3	System Architecture and Performance	6
4	TeamDrive Http Api	7
4.1	English	7
4.2	German	7
4.3	Interface Description	8
	Index	14

COPYRIGHT NOTICE

Copyright © 2014, TeamDrive Systems GmbH. All rights reserved.

TeamDrive Systems GmbH

<https://www.teamdrive.com>

Max-Brauer-Allee 50

22765 Hamburg, Germany

Email: info@teamdrive.com

INSTALLATION

2.1 English

2.1.1 Windows

The *TeamDrive Agent Client* allows a TeamDrive Windows client to be installed and run as a service. The *TeamDrive Agent Client* can be run without a GUI and allows data to be synchronized in the background, even when a Windows user is not currently logged in.

Installation

1. During the installation the *TeamDrive Agent Client* will be registered as a Windows service under the name “*TeamDrive Agent Client*” and an entry is also added to the Windows Start menu. Please be aware that the Windows user logged in at the time of the *TeamDrive Agent Client*’s installation is automatically associated with this Windows service. Only this Windows user will be able to manage the *TeamDrive Agent Client*.
2. Launch the *TeamDrive Agent Client* GUI from the Start menu and activate the installation.
3. Exit the *TeamDrive Agent Client* via “File” → “Quit”.
4. Open the Windows services overview (e.g. on Windows 8 “Control Panel” → “Administrative Tools” → “Services”).
5. Start the service “*TeamDrive Agent Client*”. Your TeamDrive Spaces will now be synchronized in the background.

Management

1. To manage the *TeamDrive Agent Client* you must switch to the Windows user that was used to install the client.
2. After logging in with the necessary Windows user, start the *TeamDrive Agent Client* via the Start menu.
3. From here you can comfortably manage your Spaces using the *TeamDrive Agent Client*.

If at some point you wish to change which Windows user you would like to use to manage the *TeamDrive Agent Client*, you simply need to edit a configuration file “teamdrive.ini”.

You can navigate to the file using “%programdata%” → “TeamDrive3” → “teamdrive.ini” or simply “C:\ProgramData\TeamDrive3\teamdrive.ini”. Open the file with a text editor, locate the entry “teamdrive-home-agent=” and replace the old Windows user account with the new desired Windows user account in the given file path. Here is an example of what you should see: “teamdrive-home-agent=C:/Users/**Windows User Account Name**/AppData/Roaming/TeamDrive3”.

Afterwards, you can continue from **Step 2** of the section **Installation**.

2.1.2 Linux

Installation

1. Please extract the *TeamDrive Agent* with `tar -xvzf TeamDrive_<Version>.tar.gz`
2. Please execute the *TeamDrive Agent* by running `.teamdrived --http-api-port=45454`
3. Open <http://localhost:45454> and log-in to your user account.

Administration

Because of the missing graphical user interface, only a reduced set of administration tasks is possible with this *TeamDrive Agent*. If the HTML-Interface is insufficient, you can directly communicate with the HTTP-API by running a tool *TeamDriveApi.py*. Additionally, you can directly edit TeamDrive settings. These settings are located after the first start in your home-directory in `~/teamdrive/teamdrive.settings`.

2.1.3 Error Messages

Error Message	Solution
Could not stop Agent Client	Please open the Windows services overview and stop the service "TeamDrive Agent Client".
Could not start Agent Client	Please open the Windows services overview and stop the service "TeamDrive Agent Client". Afterwards, restart the TeamDrive Agent Client service.

2.2 Deutsch

Ziel des **TeamDrive Agent Clients** ist die Ausführung eines *TeamDrive Clients* unter Windows als Dienst ohne Oberfläche, um auch Dateien zu synchronisieren, wenn kein Windows-Benutzer angemeldet ist oder unter Linux als Daemon ohne Oberfläche.

2.2.1 Windows

Installation

Bei der Installation registriert der TeamDrive Agent Client ein Windows-Systemdienst mit dem Namen "*TeamDrive Agent Client*" und zusätzlich einen Startmenü-Eintrag. Bitte beachten Sie, dass der aktuelle Windows-Benutzer nun mit dem Systemdienst verbunden ist. Sie können den Systemdienst **nur** von diesem Windows-Benutzer aus

administrieren.

1. Im nächsten Schritt starten Sie bitte die *TeamDrive Agent* Oberfläche, die Sie im Startmenü finden und aktivieren diese Installation.
2. Beenden Sie jetzt bitte den *TeamDrive Agent Client* über *Datei* → *Beenden*
3. Öffnen Sie die Windows-Dienste Übersicht (Beispielsweise, indem Sie im Startmenü "Dienste" eingeben)
4. Starten Sie jetzt bitte den Dienst "*TeamDrive Agent Client*". Jetzt synchronisiert der *TeamDrive Agent Client* im Hintergrund Ihre Spaces.

Administration

1. Wenn Sie den *TeamDrive Agent Client* administrieren wollen, wechseln Sie bitte zu dem Benutzer, mit dem Sie den *TeamDrive Agent Client* installiert haben.
2. Nach dem Einloggen starten Sie bitte den *TeamDrive Agent Client* über das Startmenü.
3. Jetzt können Sie wie gewohnt über den *TeamDrive Client* Ihre Spaces administrieren.

Falls Sie im Nachhinein einen anderen Windows Benutzer verwenden wollen, mit dem Sie Ihren *TeamDrive Agent Client* administrieren wollen, müssen Sie eine Konfigurationsdatei editieren. Sie finden die Datei im Ordner `%PROGRAMDATA%\TeamDrive3\` (oder `C:\ProgramData\TeamDrive3\`) mit dem Namen `TeamDrive3.ini`. Bitte ändern Sie jetzt im angegebenen Pfad unter dem Schlüssel `"teamdrive-home-agent"` den angegebenen Benutzer. Anschließend fahren Sie bitte wie im Abschnitt Installation unter Punkt 2 beschrieben fort.

2.2.2 Linux

Installation

Der TeamDrive Agent ist in einer 32-Bit und in einer 64-Bit Version verfügbar. Bitte wählen Sie die Architektur aus, die zu Ihrem System passt. Sie können die Systemarchitektur mit dem Kommando `uname -p` erfragen.

Nachdem Sie den TeamDrive Agent heruntergeladen haben, führen Sie bitte die nachfolgenden Schritte aus:

1. Bitte entpacken Sie den *TeamDrive Agent* mit `tar -xvzf TeamDrive_<Version>.tar.gz`
2. Bitte starten Sie den *TeamDrive Agent* mit `./teamdrived --http-api-port=45454`
3. Öffnen Sie bitte in Ihrem Browser die `http://localhost:45454` und loggen Sie sich mit Ihrem TeamDrive Benutzernamen ein.

Der TeamDrive Agent ist nicht automatisch als Dienst registriert, da dies Distributionabhängig ist. Bitte lesen Sie in Ihrem Benutzerhandbuch Ihrer Distribution, wie sie ein Programm als Dienst registrieren.

Sie können die Einstellung des HTTP-Ports direkt in den TeamDrive Einstellungen vornehmen, indem Sie `[Settings]` (den Sie gegebenenfalls erstellen müssen) den Eintrag:

```
http-api-port=45454
```

hinzufügen.

Administration

Auf Grund der fehlenden Oberfläche ist es nur beschränkt möglich, den TeamDrive Agent für Linux zu Administrieren. Sollte Ihnen die HTML-Oberfläche nicht ausreichen, können Sie auch direkt mit der HTTP-API vom *TeamDrive Agent* sprechen. Dafür steht Ihnen Kommandozeilenprogramm mit dem Namen `TeamDriveApi.py` zur Verfügung. Des Weiteren können Sie direkt die TeamDrive-Einstellungen editieren. Diese finden Sie nach dem ersten Start in Ihrem Home-Verzeichnis unter `~/teamdrive/teamdrive.settings`.

2.2.3 Fehlermeldungen

Meldung	Aktion
Could not stop Agent Client	Bitte öffnen Sie wie beschrieben die Dienste-Übersicht von Windows und beenden Sie dem Dienst "TeamDrive Agent Client"
Could not start Agent Client	Bitte öffnen Sie wie beschrieben die Dienste-Übersicht von Windows und beenden Sie dem Dienst "TeamDrive Agent Client" anschließend starten Sie bitte den Dienst wieder.
[Critical] ... Zugriff verweigert //<server>/<share>/<spaces>	Der Agent kann mit dem System Account nicht auf Netzlaufwerke zugreifen. Hierzu muss ein Domain Account in den Eigenschaften unter dem Reiter "Anmelden" hinterlegt werden.

SYSTEM ARCHITECTURE AND PERFORMANCE

The TeamDrive Agent is typically used to synchronize data on a central server. These are some guidelines to optimize the performance of the TeamDrive Agent in this situation:

1. Do not mount the file system over the network, which contains the data that will be synchronized. Network shares are much slower than local file systems. Using a network share will limit the number of files and directories per Space.
2. Use more than one TeamDrive Space. Using more than one Space can increase the performance. You should split your data to into at least four Spaces.

TEAMDRIVE HTTP API

4.1 English

The *TeamDrive Agent* offers a JSON interface via HTTP, which allows you to operate the *TeamDrive Agent* without using the GUI. The JSON interface can be activated entering the following text under the *[Settings]* section in the TeamDrive settings file (teamdrive.settings):

```
http-api-port=<host>
```

where *<host>* equals one of the following:

1. a port number (e.g. *45454*)
2. an IP address including the port number (e.g. *127.0.0.1:45454*)
3. an IPv6 address including the port number (e.g. *:::0]:45454*)

The command line application *TeamDriveApi.py* is located in the *TeamDrive Agent* installation directory (e.g. *C:\Program Files (x86)\TeamDrive 3 Agent*). Running this file allows you to use the TeamDrive API directly. Use *TeamDriveApi.py -help* for a list of the call functions the can be used.

If you would like to use the TeamDrive HTTP API without Python, HTTP *GET* requests can be made directly through *TeamDrive Agent*. The URL will need to be constructed as follows:

```
http://<host>/api/<command>?<parameters>
```

For example:

```
http://[::1]:45454/api/getSpace?id=1
```

The result is a JSON document.

4.2 German

Der TeamDrive Agent bietet eine JSON-Schnittstelle über HTTP an, über den Sie den TeamDrive Agent auch ohne Gui bedienen können. Diese können Sie aktivieren, indem Sie in der TeamDrive Konfigurationsdatei einen Eintrag:

```
http-api-port=<host>
```

hinzufügen, wobei *<host>* eines der folgenden Elemente sein kann:

1. Ein port. Beispielsweise *45454*
2. eine IP mit Port. Beispielsweise *127.0.0.1:45454*
3. eine IPv6 mit Port. Beispielsweise *:::0]:45454*

Im Installationsverzeichnis befindet sich ein Kommandozeilenprogramm *TeamDriveApi.py*, mit dem Sie die TeamDrive Api direkt ansprechen können. Die Aufrufkonvention entnehmen Sie bitten der Ausgabe von *TeamDriveApi.py -help*.

Falls Sie *TeamDrive Http Api* ohne Python verwenden wollen, könne Sie direkt Http *GET* Anfragen an den *TeamDrive Agent* richten, wobei die URL wie Folgt aufgebaut ist:

```
http://<host>/api/<command>?<parameters>
```

Beispielsweise:

```
http://[::1]:45454/api/getSpace?id=1
```

das Ergebnis ist ein *JSON*-Dokument.

4.3 Interface Description

```
class TeamDriveApi.TeamDriveApi (server='[::1]:45454')
```

about ()

Returns information about TeamDrive.

Return type dict

Raises **httplib.HTTPException** a Connection Error.

addAddressbook (*name*)

Adds a new Addressbook entry. TeamDrive will lookup that name on the RegServer. If that fails, an exception is thrown. You cannot add email addresses.

Parameters **name** (*str*) – the new Addressbook's name.

Returns the id of the new entry.

Return type dict

Raises

- **httplib.HTTPException** – a Connection Error.
- **TeamDriveException** – user is unknown.

createSpace (*spaceName, disableFileSystem, spacePath=None, importExisting=None*)

Creates a new Space. Throws **TeamDriveException**, if the call fails.

Parameters

- **spaceName** (*str*) – The Name of the new Space
- **disableFileSystem** (*bool*) – This will disable the synchronization into the file system
- **spacePath** (*str*) – File Path, in which the new Space will be created.
- **importExisting** (*bool*) – Tells the TeamDrive Agent to import an existing directory.

Return type dict

Raises

- **httplib.HTTPException** – a Connection Error.
- **TeamDriveException** – Space creation failed.

deleteSpace (*spaceId, delInFs, delOnServer*)

Deletes a Space. Throws `TeamDriveException`, if the call fails.

Parameters

- **spaceId** (*int*) – The Id of the Space
- **delInFs** (*bool*) – Delete the Space in the File System
- **delOnServer** (*bool*) – Deletes the Space on the server. (requires Administrator rights)

Return type dict

Raises

- **httplib.HTTPException** – a Connection Error.
- **TeamDriveException** – Deletion failed

getAddressbook (*addressId*)

Returns information about a given Addressbook Id.

Parameters **addressId** (*int*) – the Address Id.

Return type dict

Raises **httplib.HTTPException** a Connection Error.

getAddressbookByName (*addressName*)

Returns an Addressbook by a given name. Throws a `TeamDriveException`, if there is no Space with this name.

Parameters **addressName** (*str*) – The Name.

Return type dict

Raises

- **httplib.HTTPException** – a Connection Error.
- **TeamDriveException** – There is no Space with this name

getAddressbookIds ()

Returns all known Addressbook Ids.

Return type list[int]

Raises **httplib.HTTPException** a Connection Error.

getFile (*fileId*)

Returns information about the given File Id.

Parameters **fileId** (*int*) – The File id

Return type dict

Raises **httplib.HTTPException** a Connection Error.

getFiles (*spaceId, filePath, trashed*)

Returns a list of files matching Space Id, file path and trashed flag.

Parameters

- **spaceId** (*int*) – The Id of the Space
- **filePath** (*str*) – the path of the file in the Space.
- **trashed** (*bool*) – Is the file in the trash?

Return type list[dict]

Raises `httplib.HTTPException` a Connection Error.

getFolderContent (*spaceId*, *filePath*, *trashed*)

Returns a list of file Ids in a folder of a Space.

Parameters

- **spaceId** (*int*) – The Id of the Space
- **filePath** (*str*) – The parent folder path.
- **trashed** (*bool*) – In Trash?

Return type list[int]

Raises `httplib.HTTPException` a Connection Error.

getLoginInformation ()

Returns information about the current user.

Return type dict

Raises `httplib.HTTPException` a Connection Error.

getSettings ()

Returns a dict of all known settings. Some of them are not meant to be set by the user.

Return type dict

Raises `httplib.HTTPException` a Connection Error.

getSpace (*spaceId*)

Will return informations about that Space.

example:

```
>>> api = TeamDriveApi("127.0.0.1:45454")
>>> map(api.getSpace, api.getSpaceIds())
[ {... } ]
```

Parameters **spaceId** (*int*) – The Id of the Space

Return type dict

Raises `httplib.HTTPException` A Connection Error.

getSpaceByName (*spaceName*)

Returns a Space by a given Space name. Throws a `TeamDriveException`, if there is no Space with this name.

Parameters **spaceName** (*str*) – The Name.

Return type dict

Raises

- `httplib.HTTPException` – a Connection Error.
- `socket.error` – a Connection Error.
- `TeamDriveException` – There is no Space with this name

getSpaceIds ()

Returns a list of all known Spaces Ids.

example:

```
>>> api = TeamDriveApi("127.0.0.1:45454")
>>> map(api.getSpace, api.getSpaceIds())
[ {... } ]
```

Return type list[int]**Raises** `httplib.HTTPException` A Connection Error.**getSpaceMemberIds** (*spaceId*)

Returns all Addressbook Ids in a given Space.

Return type list[int]**Raises** `httplib.HTTPException` a Connection Error.**inviteMember** (*spaceId*, *addressId*, *text*)Invites a user into a Space. Throws `TeamDriveException`, if the call fails.**Parameters**

- **spaceId** (*int*) – The Id of the Space
- **addressId** (*int*) – Already known Addressbook Id.
- **text** (*str*) – Invitation Text

Return type dict**Raises**

- `httplib.HTTPException` – a Connection Error.
- `TeamDriveException` – Inviting that member failed

joinSpace (*spaceId*, *disableFileSystem*, *spacePath=None*)Joins an archived Space or rejoins an existing. Throws `TeamDriveException`, if the call fails.**Parameters**

- **spaceId** (*int*) – The Id of the Space
- **disableFileSystem** (*bool*) – This will disable the synchronization into the file system
- **spacePath** (*str*) – Optional Space root in the file system.

Return type dict**Raises**

- `httplib.HTTPException` – a Connection Error.
- `TeamDriveException` – Joining failed

kickMember (*spaceId*, *addressId*)

Removes a member form this Space.

Throws `TeamDriveException`, if the call fails. For example, if the one doesn't have the rights to remove a user**Parameters**

- **spaceId** (*int*) – The Id of the Space
- **addressId** (*int*) – Already known Addressbook Id.

Return type dict

Raises

- **httplib.HTTPException** – a Connection Error.
- **TeamDriveException** – Removing this user failed

login (*username, password*)

Will login a user with a given password. Throws `TeamDriveException`, if the call fails.

Parameters

- **username** (*str*) – The Username
- **password** (*str*) – The Password

Return type dict

Raises

- **httplib.HTTPException** – a Connection Error.
- **TeamDriveException** – Logging failed.

moveFile (*spaceId, filePath, trashed, newFilePath*)

Moves a file from `filePath` to `newFilePath` in the given Space. Throws `TeamDriveException`, if the call fails.

Parameters

- **spaceId** (*int*) – The Id of the Space
- **filePath** (*str*) – the path of the file in the Space.
- **trashed** (*bool*) – Is the file in the trash?
- **newFilePath** (*str*) – The destination file path.

Return type dict

Raises

- **httplib.HTTPException** – a Connection Error.
- **TeamDriveException** – Failed to queue command.

putFile (*spaceId, path, data*)

Uploads a file into a Space. The file creation is done asynchronously, a successful return code does not indicate a successful import. Use `getFiles()` to request the current state of that file.

Parameters

- **spaceId** (*int*) – The Id of the Space
- **path** (*str*) – the path of the file in the Space.
- **data** – The binary data

Return type dict

Raises **httplib.HTTPException** a Connection Error.

putFileContent (*spaceId, spacePath, filePath*)

Uploads “`filePath`” into a Space under the filename “`spacePath`”. The file creation is done asynchronously, a successful return code does not indicate a successful import. Use `getFiles()` to request the current state of that file.

Parameters

- **spaceId** (*int*) – The Id of the Space
- **spacePath** (*str*) – the path of the file in the Space.
- **filePath** (*str*) – the path to the local file

Return type dict

Raises **httplib.HTTPException** a Connection Error.

quit (*logout*)

Quits TeamDrive. Throws `TeamDriveException`, if the call fails.

Parameters **logout** (*bool*) – Quit and Logout

Return type dict

Raises **httplib.HTTPException** a Connection Error.

removeLocallyFile (*fileId, recursive=False*)

Removes a file form the local file system. A successful return code does not indicate a successful execution, because it is done asynchronously.

Parameters

- **fileId** (*int*) – the file to be removed
- **recursive** (*bool*) – Directories only: recursive remove

Return type dict

Raises

- **httplib.HTTPException** – a Connection Error.
- **TeamDriveException** – Failed to queue command.

restoreLocallyFile (*fileId, recursive=False*)

Restores a file form the local file system. A successful return code does not indicate a successful execution, because it is done asynchronously.

Parameters

- **fileId** (*int*) – the file to be restored
- **recursive** (*bool*) – Directories only: recursive restore

Return type dict

Raises

- **httplib.HTTPException** – a Connection Error.
- **TeamDriveException** – Failed to queue command.

setSetting (*key, value*)

set a Setting. must be a valid key as in `getSettings`. NOTE:

Parameters

- **key** (*str*) – The name of the setting
- **value** – The new value of the setting

Return type dict

Raises **httplib.HTTPException** a Connection Error.

A

about() (TeamDriveApi.TeamDriveApi method), 8
 addAddressbook() (TeamDriveApi.TeamDriveApi method), 8

C

createSpace() (TeamDriveApi.TeamDriveApi method), 8

D

deleteSpace() (TeamDriveApi.TeamDriveApi method), 8

G

getAddressbook() (TeamDriveApi.TeamDriveApi method), 9
 getAddressbookByName() (TeamDriveApi.TeamDriveApi method), 9
 getAddressbookIds() (TeamDriveApi.TeamDriveApi method), 9
 getFile() (TeamDriveApi.TeamDriveApi method), 9
 getFiles() (TeamDriveApi.TeamDriveApi method), 9
 getFolderContent() (TeamDriveApi.TeamDriveApi method), 10
 getLoginInformation() (TeamDriveApi.TeamDriveApi method), 10
 getSettings() (TeamDriveApi.TeamDriveApi method), 10
 getSpace() (TeamDriveApi.TeamDriveApi method), 10
 getSpaceByName() (TeamDriveApi.TeamDriveApi method), 10
 getSpaceIds() (TeamDriveApi.TeamDriveApi method), 10
 getSpaceMemberIds() (TeamDriveApi.TeamDriveApi method), 11

I

inviteMember() (TeamDriveApi.TeamDriveApi method), 11

J

joinSpace() (TeamDriveApi.TeamDriveApi method), 11

K

kickMember() (TeamDriveApi.TeamDriveApi method), 11

L

login() (TeamDriveApi.TeamDriveApi method), 12

M

moveFile() (TeamDriveApi.TeamDriveApi method), 12

P

putFile() (TeamDriveApi.TeamDriveApi method), 12
 putFileContent() (TeamDriveApi.TeamDriveApi method), 12

Q

quit() (TeamDriveApi.TeamDriveApi method), 13

R

removeLocallyFile() (TeamDriveApi.TeamDriveApi method), 13
 restoreLocallyFile() (TeamDriveApi.TeamDriveApi method), 13

S

setSetting() (TeamDriveApi.TeamDriveApi method), 13

T

TeamDriveApi (class in TeamDriveApi), 8