TeamDrive Agent Documentation

Release 4.1.2

TeamDrive Systems GmbH

December 23, 2015

CONTENTS

CHAPTER

ONE

COPYRIGHT NOTICE

TeamDrive Systems GmbH
https://www.teamdrive.com
Max-Brauer-Allee 50
22765 Hamburg, Germany
Email: info@teamdrive.com

CHAPTER

TWO

INTRODUCTION

This manual will guide you through the installation of your TeamDrive Agent. This document is intended for system administrators and users, who need to install and configure a TeamDrive Agent.

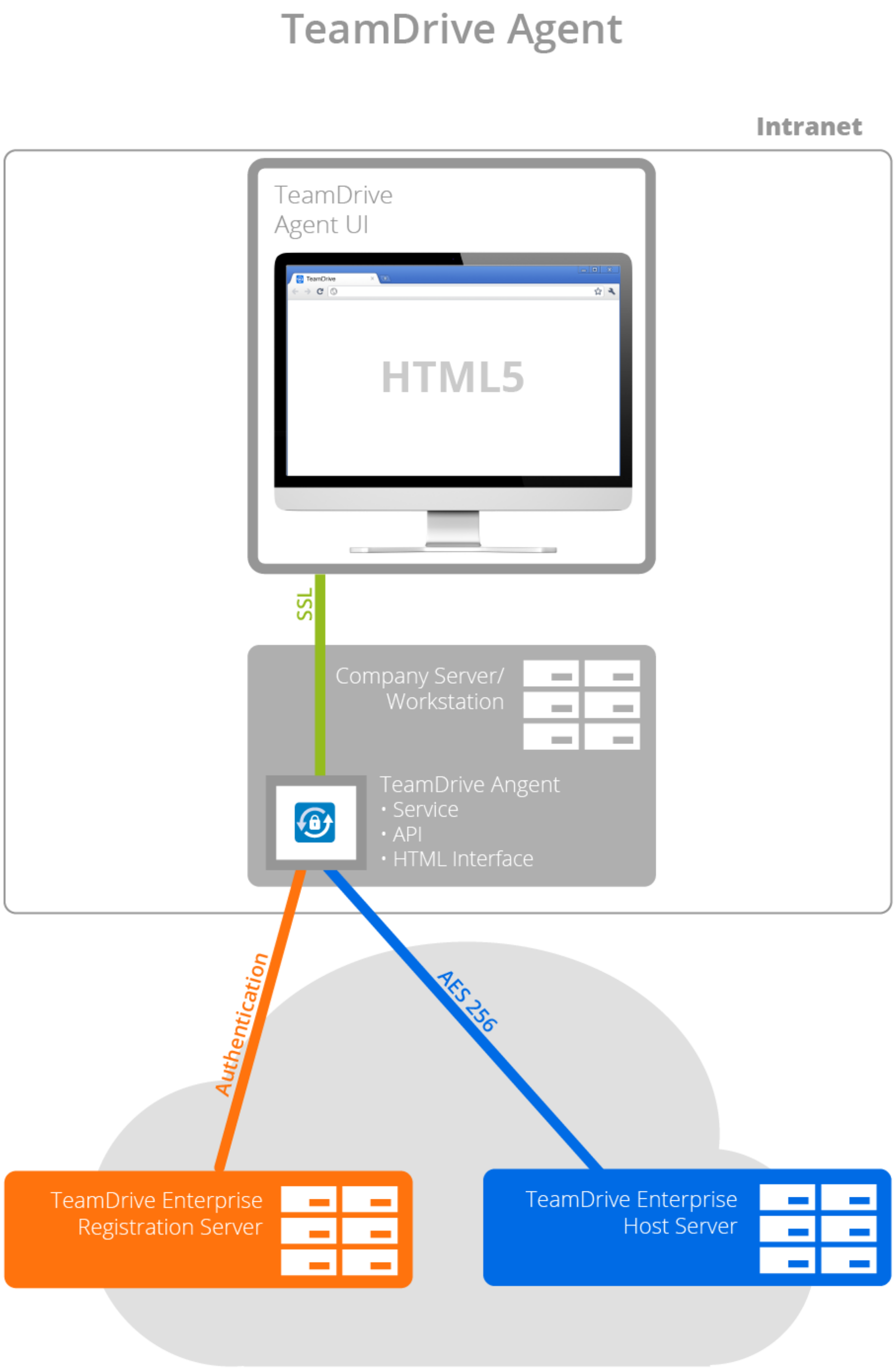
This Installation Guide outlines the deployment of a stand alone TeamDrive Agent, which is not part of a Web Portal.

2.1 Overview

The TeamDrive Agent is a full featured TeamDrive Client without a graphical user interface. An Agent provides:

- Synchronization of Spaces with the Cloud.
- Synchronization of Spaces with a local file system.
- Synchronization of Spaces with a remote file system.
- Management GUI using a Web App.
- API access using JSON calls.

This figure shows an overview of a TeamDrive Agent installation:


TeamDrive Agent
Intranet
TeamDrive
Agent UI
HTML5
SSL
Company Server/
Workstation
TeamDrive Angent
• Service
• API
• HTML Interface
Authentication
AES 256
TeamDrive Enterprise
Registration Server
TeamDrive Enterprise
Host Server

Typically, a TeamDrive Agent is installed on a server or network attached storage and communicates with the TeamDrive Cloud.

CHAPTER

THREE

WINDOWS INSTALLATION

3.1 Requirements

The *TeamDrive Agent* allows a TeamDrive Windows client to be installed and run as a Windows service. The *TeamDrive Agent* 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.

The support of older Windows versions has been dropped. The minimum supported Windows version is **Windows 7** or **Windows Server 2009**.

In order to install the *TeamDrive Agent*, you will need at least 50 MB of free disk space, a valid email address and Internet access. In the end, the total amount of disk space used will depend on the size of your created and/or joined Spaces.

If it is your first time installing and using TeamDrive you will need to create a TeamDrive account before installing the *TeamDrive Agent*. TeamDrive cannot be used without an account.

3.2 Required Skills

When installing the *TeamDrive Agent*, we assume that you have basic knowledge of:

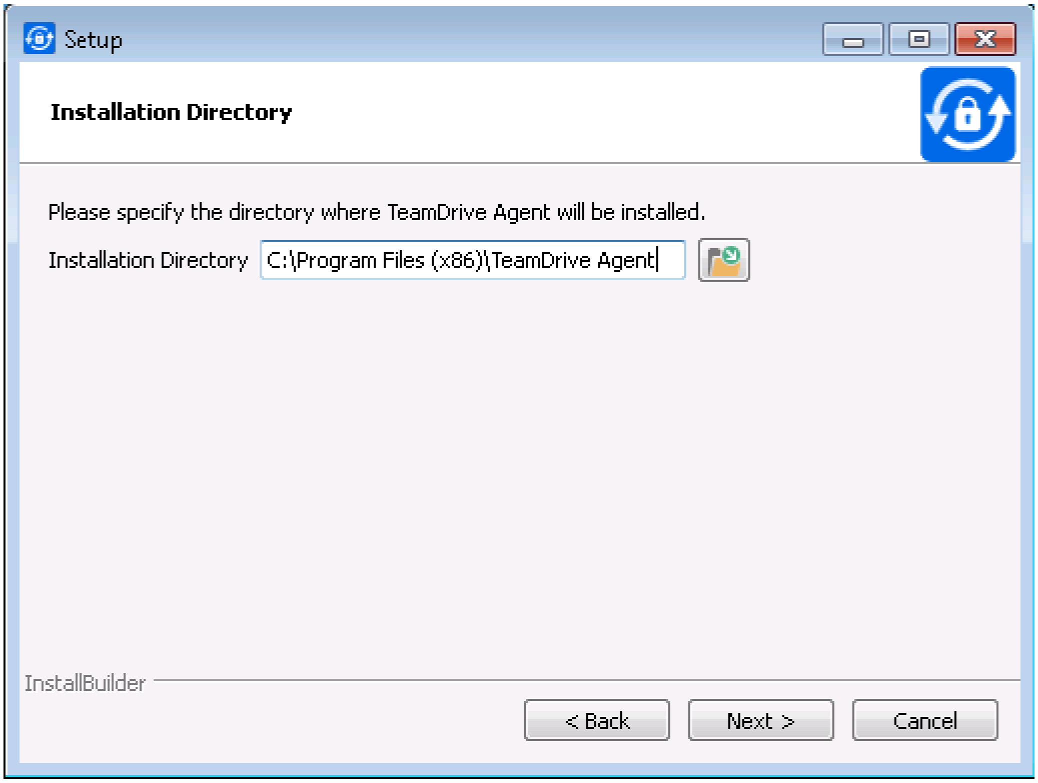
- Starting and stopping of Windows Services
- Configuration of a firewall

3.3 Installation

After you have successfully downloaded the *TeamDrive Agent*, double-click the installation file to start the installation.

Note: In order to install the *TeamDrive Agent* on a Windows terminal server, you need to put the server into a special installation mode by executing *Change user /install* at a command prompt.

1. To install the *TeamDrive Agent*, please specify an installation directory as seen in the following screen shot. We recommend not changing the installation directory to another location, as using the default directory will help with future version upgrades.

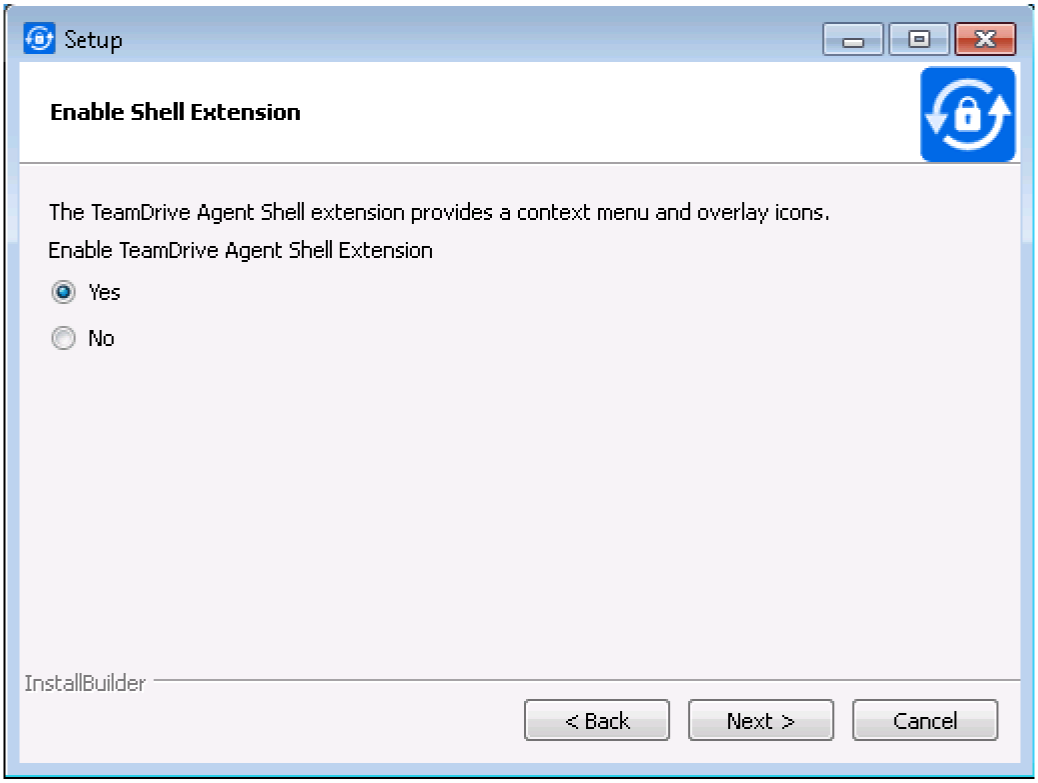


Installation Directory

Please specify the directory where TeamDrive Agent will be installed.

Installation Directory: C:\Program Files (x86)\TeamDrive Agent

2. Afterwards, you can choose to install a Windows Explorer integration which provides some context menu entries and overlay icons that show the current synchronization status of Spaces, folders and files:



Enable Shell Extension

The TeamDrive Agent Shell extension provides a context menu and overlay icons.

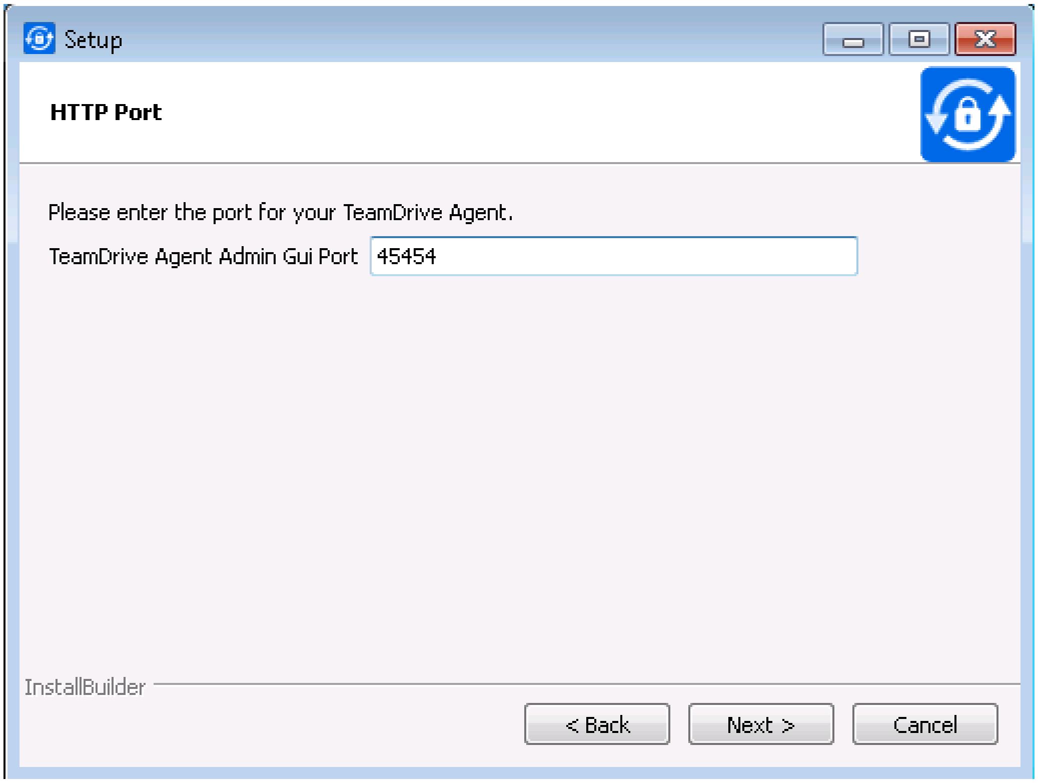
Enable TeamDrive Agent Shell Extension

● Yes

○ No

3. In order to administer the *TeamDrive Agent*, you will need to specify a TCP port. This port will be used to

provide an administration interface using your web browser. You can choose another port number, but further examples in this document will use the port *45454*:



Setup

HTTP Port

Please enter the port for your TeamDrive Agent.

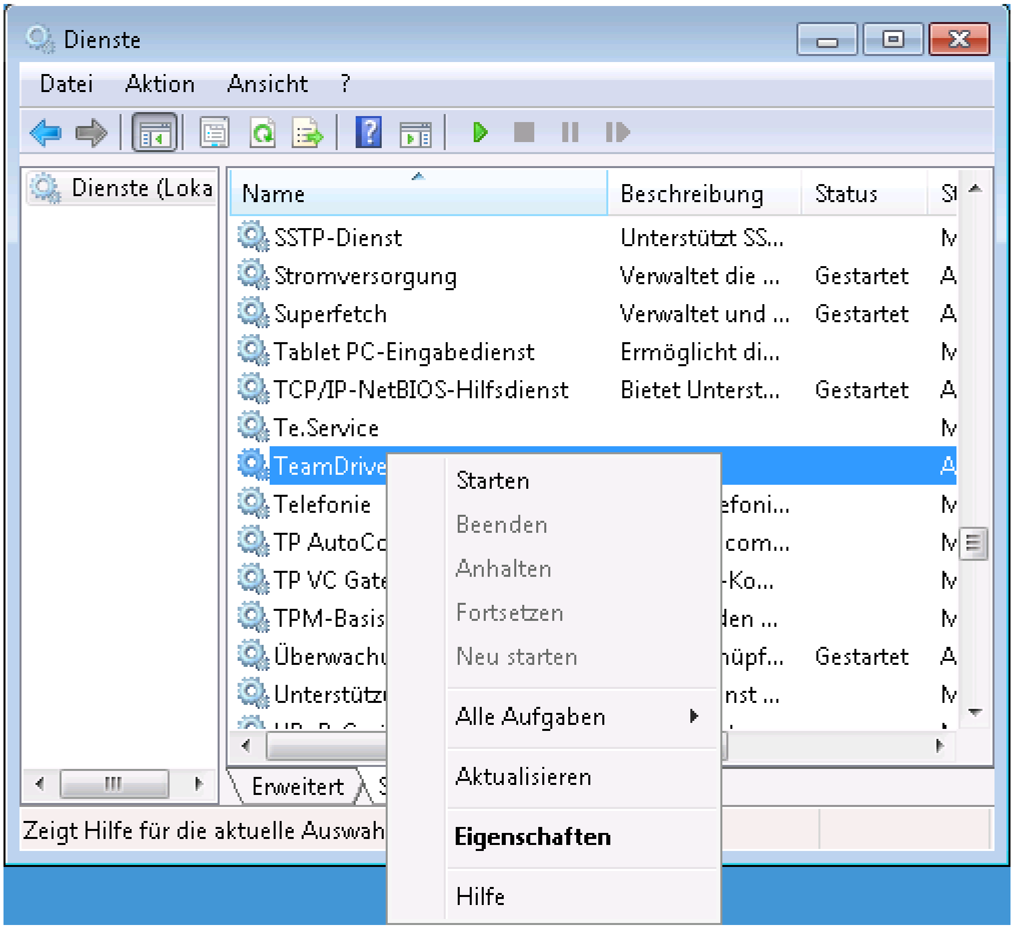
TeamDrive Agent Admin Gui Port: 45454

4. During the installation, the *TeamDrive Agent* will be registered as a Windows service under the name "*TeamDrive Agent Client*". 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 access the settings file of the *TeamDrive Agent*.

3.4 Post-Installation Tasks

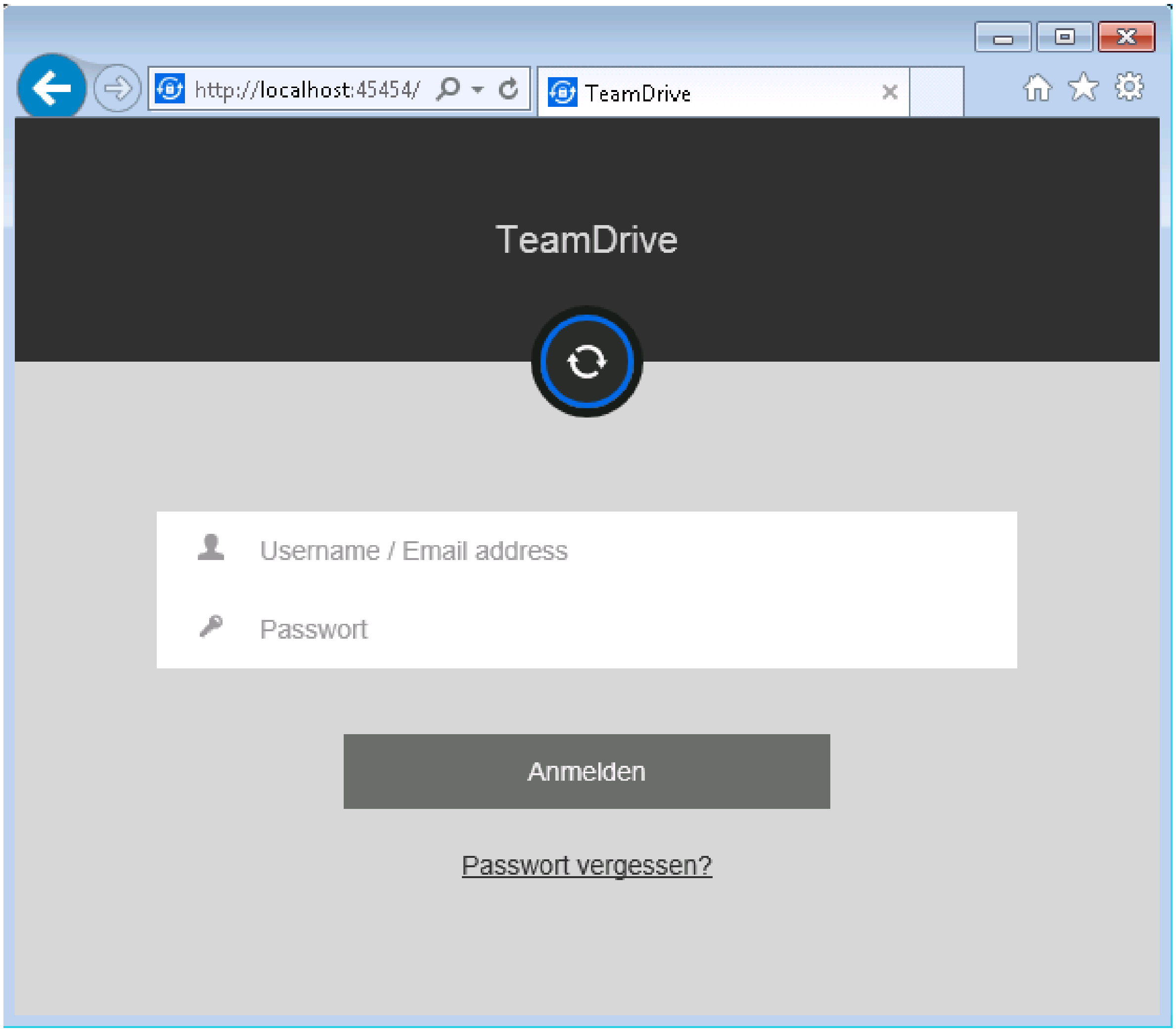
To be able to synchronize files with the *TeamDrive Agent*, some initial steps are required.

1. Open the Windows services overview (e.g. on Windows 8 "Control Panel" → "Administrative Tools" → "Services").
2. Please start the Windows service "*TeamDrive Agent Client*":



Name	Beschreibung	Status	St
SSTP-Dienst	Unterstützt SS...		M
Stromversorgung	Verwaltet die ...	Gestartet	A
Superfetch	Verwaltet und ...	Gestartet	A
Tablet PC-Eingabedienst	Ermöglicht di...		M
TCP/IP-NetBIOS-Hilfsdienst	Bietet Unterst...	Gestartet	A
Te.Service			M
TeamDrive			A
Telefonie	efoni...		M
TP AutoCo	com...		M
TP VC Gate	-Ko...		M
TPM-Basis	den ...		M
Überwachu	hüpf...	Gestartet	A
Unterstütz	nst ...		M

3. Open your browser and go to http://localhost:45454. Please replace the port with the one that was used during installation:



You can now activate Spaces in the administration interface. Your TeamDrive Spaces will now be synchronized in the background.

3.5 Changing the Windows User Account

If at some point you wish to change which Windows user you would like to use to manage the *TeamDrive Agent*, you simply need to edit the 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 a typical *teamdrive.ini* that was created by the installer:

```
[Settings]
teamdrive-home-agent=C:/Users/teamdrive/AppData/Roaming/TeamDrive3
http-api-port-agent=45454
shell-extension-user-agent=teamdrive
```

Please also look at the documentation of `teamdrived --teamdrive-home` (page 17), `teamdrived --http-api-port` (page 19) and *Platform specific Settings* (page 20).

The setting *shell-extension-user-agent* must be set to the same Windows username as in the *teamdrive-home* setting.

Afterwards, please continue with *Post-Installation Tasks* (page 7).

3.6 Accessing Network Shares

In order to access Spaces on network shares, the Windows account needs to be changed to a user, who has rights to access network shares. It is recommended to use the Windows account used to install the *TeamDrive Agent*.

Note: the *TeamDrive Agent* running as a Windows Services may not be able to access mapped network dries. It is recommended to use a *UNC* path directly as a Space root.

CHAPTER

FOUR

LINUX INSTALLATION

At the moment, you can either use the 64 bit RPM package or the 64 bit .tar.gz file.

4.1 Requirements

The Library Dependencies are moderate and should be provided by all recent linux distributions:

Library	Version
GLibc	2.9
GLibc++	3.4.11
libz	

Under Linux, TeamDrive is using the inotify [1] API of the Linux kernel to be notified about file modifications. Under some Linux distributions, there is a rather small limit regarding the number of files that can be watched. You can find out the current watch handle limit by executing `cat /proc/sys/fs/inotify/max_user_watches` in a terminal. If this number is lower than the number of files in your Spaces, please increase this value by adding for example `fs.inotify.max_user_watches=100000` to your */etc/sysctl.conf* file. Please reboot your system afterwards.

4.2 Installation on RPM based Distributions

If you are on a RPM based distribution, please install the TeamDrive Agent by installing the rpm file.

The TeamDrive Agent is hosted on a dedicated yum repository. This makes the installation and applying of future updates very easy — you can simply run yum update to keep your TeamDrive Agent up to date.

To enable the repository, you need to download the td-agent.repo file and place it into the directory */etc/yum.repos.d/*, e.g. by using *wget*:

```
[root@agent ~]# wget -O /etc/yum.repos.d/td-agent.repo \
http://repo.teamdrive.net/td-agent.repo
```

This will enable the “TeamDrive Agent Version 4” repository, which you can check by running *yum repolist* afterwards:

```
[root@agent ~]# yum repolist
Loaded plugins: security
repo id             repo name                                   status
td-agent-4          TeamDrive Agent Version 4                   4
base                CentOS-6 - Base                             6.367
extras              CentOS-6 - Extras                           14
```

[1] http://man7.org/linux/man-pages/man7/inotify.7.html

```
updates            CentOS-6 - Updates                        1.094
repolist: 7.477
```

To install the TeamDrive Agent, install the following package via yum from the “TeamDrive Agent” repository:

```
[root@agent ~]# yum install teamdrive-Agent
```

Please continue with *Starting and Stopping* (page 12).

4.3 Installation on other Distributions

To install the TeamDrive Agent on a non-RPM based distribution, please download the Agent manually for example by using wget:

```
wget http://s3download.teamdrive.net.s3.amazonaws.com/4.<version>/TMDR/linux-x86_64/TeamDrive_Agent_4
```

Please replace <*version*> with the latest release. Now, please extract the *TeamDrive Agent*:

```
tar -xvzf TeamDrive_Agent_4.<version>_x86_64.tar.gz
```

This will extract the TeamDrive Agent into a subdirectory *TeamDrive*.

Note: The *tar.gz* file does not register a *teamdrive-Agent* command on your system.

4.4 Starting and Stopping

Before starting the TeamDrive Agent for the first time, please also read *Initial Set-Up* (page 12).

Please start TeamDrive by using the shortcut in your desktop menu or bar. Alternatively, you can execute the global command “teamdrive-Agent” or ”./teamdrived” from the installation folder. TeamDrive requires some system settings that are not available by executing the teamdrived.bin executable directly.

Note: Running TeamDrive as root is not recommended. TeamDrive runs perfectly without privileges.

To stop the TeamDrive Agent, please execute:

```
/opt/teamdrive/TeamDriveApi.py '<agent-host>:<port>' quit False
```

Replace <*agent-host*> with the host where your TeamDrive Agent is installed and the port with the one you specified.

4.5 Initial Set-Up

To administrate your TeamDrive Agent, you need to enable the HTTP API by setting the `teamdrived --http-api-port` (page 19) option either as a command line argument or as a setting.

Then start the TeamDrive Agent, open your favorite browser and go to *http://<agent-host>:45454*. Replace <*agent-host*> with the host where your TeamDrive Agent is installed and the port with the one you specified. Log-in to your TeamDrive account now.

CHAPTER FIVE

PERFORMANCE CONSIDERATION

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. For example, you could have one Space per project or team.

CHAPTER SIX

MIGRATING FROM TEAMDRIVE 3

The application data directory (specified by the `teamdrived --teamdrive-home` (page 17) parameter) is compatible to TeamDrive 3.

1. Perform a full system backup in order to be able to restore your system if you encounter unexpected errors.
2. Shutdown the TeamDrive 3 Agent.
3. Install the new TeamDrive 4 Agent as described, but do not start the TeamDrive Agent.
4. Make sure that you have only one installation of the TeamDrive Agent. Manually delete older TeamDrive 3 installations in case they are not automatically deleted.
5. Start the TeamDrive 4 Agent. See *Initial Set-Up* (page 12).

> **Warning:** On Windows, the *uninstall.exe* will delete the application data directory. Be careful when removing old installations.

CHAPTER SEVEN

TROUBLESHOOTING

TeamDrive stores log output to `%APPDATA%\TeamDrive3\logs` or `~/.teamdrive/logs`. Please search for `[Critical]` messages that can be a hint to an error.

TeamDrive Agent is started, but administration interface is not available (Linux)

If you are not able to access the administration interface in you browser, please check that TeamDrive has opened a TCP port:

```
[root@agent ~] netstat -twaunp
```

Please edit the settings file and set TeamDrive to open the port 45454. This can be done by editing the file `~/.teamdrive/teamdrive.settings` file. This is an example:

```
[root@agent ~] vim ~/.teamdrive/teamdrive.settings
```

Add the following entries:

```
[Settings]
http-api-type=tcp
http-api-port=45454
```

Restart TeamDrive with the following commands:

```
[root@agent ~] killall teamdrived.bin
# Wait 10 seconds
[root@agent ~] ./teamdrived
```

CHAPTER EIGHT

SYNOPSIS

teamdrived [options] (autostart | shutdown)

CHAPTER

NINE

OPTIONS

This is a list of known settings or command line arguments that are used by TeamDrive.

The majority of these settings can also be added to **[Settings]** Section of , the `teamdrive.settings` file located in the *`teamdrived --teamdrive-home`* (page 17) folder, which typically `~/.teamdrive` on Linux or Mac OS and `%APPDATA%\TeamDrive3` on Windows.

This is a valid `teamdrive.settings` file that sets two settings on Linux:

```
[Settings]
language=en-US
spaces-path=/home/teamdrive/Spaces
```

And this is the corresponding settings file for Windows:

```
[Settings]
language=en-US
spaces-path=C:/Users/TeamDrive/Documents/Spaces
```

Please note the usage of slashes instead of backslashes on Windows. Also notice that a setting in the `teamdrive.settings` file is not prefixed by two dashes (`--`).

9.1 Local Installation Settings

`--spaces-path`
Default path for newly created Spaces by the user.

`--teamdrive-home`
Set TeamDrive application data path. Typically `~/.teamdrive` on Linux or Mac OS and `%APPDATA%\TeamDrive3` on Windows.

`--data-path`
Default location for temporary data and cache files. Defaults to *`teamdrived --teamdrive-home`* (page 17)

`--default-server`
Default host server to create spaces on

`--perform-backup``=true/false (default: true)`
Enable / Disable creation of Backups

`--backup-path`
Set the Backup path

`--language`
Set the user interface language

`--autostart-delay`
Delay background tasks to speedup system startup time

`--working-set-size`
Maximum cache size

`--enable-folder-decoration``=true/false (default: true)`
Enable / Disable file system folder decoration

`--enable-create-favorite-link-to-spaces``=true/false (default: true)`
Windows only: creates a shortcut in the Windows explorer favorites to the *`teamdrived --spaces-path`* (page 17) folder.

`--device``=<device>`
if *device* is "softwarecontext", TeamDrive will use the QtQuick2dRenderer as the rendering backend (Windows only)

`--enable-multi-processes``=true/false (default: false)`
Allow spawning of multiple TeamDrive instances.

`--enable-shell-extension``=true/false (default: true)`
Enables or disables the Windows Explorer integration or the Finder Extension.

New in version 4.1.2.

`autostart`
Don't show the main window.

`shutdown`
Shutdown the currently running TeamDrive process.

9.2 Registration Server Communication

`--check-for-updates``=true/false (default: true)`
The TeamDrive Client will check for software updates on the Registration Server. Set this value to false, if a software distribution tool will be used to deploy Client installations to your users.

`--auto-accept-invitation``=true/false (default: false)`
When set to true, the TeamDrive Client will accept all Space invitations automatically and join these Spaces.

`--auto-accept-invitation-mode` `(default: archived)`
The mode of operation when joining Spaces automatically. Possible values are: current-version-only, all-versions, directory-information-only, no-sync-to-filesystem, offline-available, archived.

`--auto-invite-users``=list`
A list of user names to be automatically invited into newly created Spaces with specified DefaultInvitationRights. The list has to be separated by semicolons and enclosed with double quotes in the settings file. Example: auto-invite-users="abc;def"

`--enable-key-repository``=true/false (default: true)`
Enable / Disable the Key Repository.

9.3 Synchronization Settings

`--default-file-permissions`
Default unix file permissions. Default is 644.

`--default-dir-permissions`
Default unix directory permissions. Default is 755.

`--scan-enabled``=true/false (default: true)`
On startup, the internal database will be compared with the file system using a file system scan to detect Space changes while TeamDrive was not running.

9.4 Agent Settings

`--http-api-port``=<host>`
where <*host*> equals one of the following:

1.a port number (e.g. `45454`)

2.an IPv4 address including the port number (e.g. `127.0.0.1:45454`)

3.an IPv6 address including the port number (e.g. `[::0]:45454`)

4.a pipe or socket name. Requires `http-api-type=local`.

`--http-api-type`
Sets the type of the HTTP API.

1.`local`: Opens a Named pipe on Windows or a Socket file on Linux

2.`tcp`: Opens a TCP port.

3.`ssl`: Opens a SSL/TLS encrypted TCP socket. See *`teamdrived --http-api-private-key`* (page 19) and *`teamdrived --http-api-certificate`* (page 19)

`--http-api-certificate`
SSL-Server only: A path to a certificate file for the api (PEM format).

`--http-api-private-key`
SSL-Server only: A path to a RSA private file for the certificate (PEM format). See also *`teamdrived --http-api-certificate`* (page 19).

`--idle-shutdown-timeout`
The number of idle seconds, until the TeamDrive Agent shuts itself down. Typically used with automatically shutdown of Linux containers. Default: no timeout.

New in version 4.1.1.

`--http-api-external-register-url`
This url will be used to show a “Register Now” link in the Web UI. This setting is mainly useful for Web Portals.

New in version 4.1.1.

`--http-api-external-login-url`
This url will be used to redirect the Browser if a login is required. This setting is intended to be used for external authentication.

New in version 4.1.1.

`--http-api-compress-replies``=true/false (default: true)`
Use a zipped HTTP content-encoding for JSON replies.

New in version 4.1.1.

9.5 Platform specific Settings

It is possible to limit a setting to a specific domain. For example if you add -linux to a setting, this setting will only affect Linux clients.

Suffix	Description
-mac	Mac OS X only.
-win	Windows only.
-linux	Linux Clients only.
-agent	Used by the Headless Agent.

CHAPTER

TEN

TEAMDRIVE HTTP API

10.1 Overview

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 by specifying the `teamdrived --http-api-port` (page 19) option.

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. See *Options* (page 17) for a list of Options and Settings.

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.

10.2 Requirements

In order to run use the interface, you will need the these requirements:

- TeamDrive Desktop Client or *TeamDrive Agent*

The `Python module and Command line interface` requires

- *Python* in your *PATH* variable.
- Recommended: docopt for Python.
- `TeamDriveApi.TeamDriveApi.sse()` (page 51) requires *Python 3*.

10.3 Security Considerations

Communicating with the *TeamDrive Http Api* contains sensitive information, for example user names, passwords and file data. There are mechanisms to provide authentication and privacy.

10.3.1 Restricting the availability of the Api

By restricting the api to the local host, any attempt to access the api from another host will fail:

```
http-api-port=127.0.0.1:<port>
```

Although, this does not prevent the usage of the api from another user of this system.

10.3.2 Using a local socket or a named pipe

One can further reduce the availability by specifying the type of the api like this:

```
http-api-type=socket
http-api-port=<name>
```

<name> can be a user defined string. The Unix socket file will be placed in */tmp* on Mac OS X and on Linux.

10.3.3 Using HTTPS for communication with the TeamDrive Agent

HTTPS can be enabled by setting *type* to *ssl* like this:

```
http-api-type=ssl
http-api-port=443
```

It is recommended to use port 443 for communication with browsers, because using a non-standard port for HTTPS is considered insecure. Please also have a look at the `teamdrived --http-api-certificate` (page 19) and `teamdrived --http-api-private-key` (page 19) settings.

10.4 Authentication

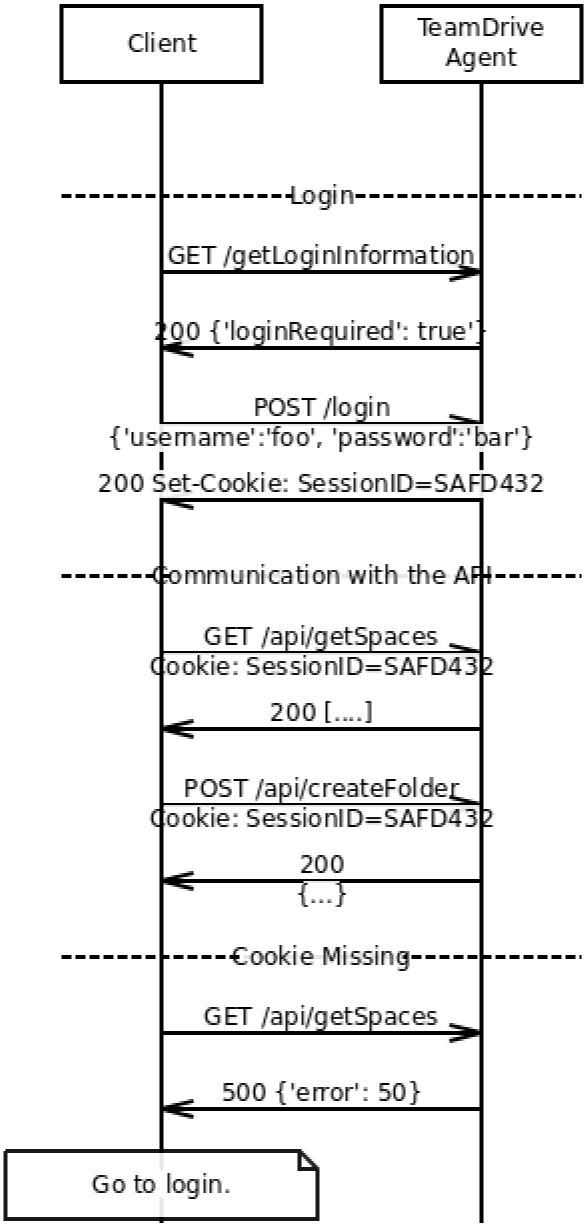
In order to prevent unauthorized users from accessing the *TeamDrive Http Api*, an authentication is **required** to access the api. There are two ways to authenticate clients:

1. Statefull Session Cookies
2. Stateless Authorization Header

10.4.1 Session Cookies

Session Cookies are mainly used to authenticate Web Browsers. Using a session requires a call to `TeamDriveApi.TeamDriveApi.login()` (page 43) to retrieve a cookie from the server. `TeamDriveApi.TeamDriveApi.getLoginInformation()` (page 35) can be used to determine if a login is required.

Login with the Api using Cookies should follow this protocol:



In the case of a missing Cookie, a `TeamDriveApi.TeamDriveException.Error_Missing_Session` (page 53) is returned

10.4.2 Authorization Header

This stateless alternative to Cookies uses HTTP Basic Authentication. **Every** request to the Api requires an *Authorization Header* following **RFC 2617**.

Currently, The Agent does *not* answer requests without Authentication Header appropriately. It is expected, that clients send the Authorization Header unconditionally.

10.5 Interface Overview

TeamDrive provides two API endpoints:

- The `/api` endpoint provides meta data about all elements of TeamDrive
- The `/files` endpoint provides an interface for reading and writing the file content.

10.5.1 Communicating with the /api endpoint

This is the reference documentation of the TeamDrive Client API. As a convenience, there is a Python module that provides accessor functions for each API call, which is also usable as a command line interface. Please have a look at all possible usages.

Example using curl:

```
$ curl '127.0.0.1:45454/api/getLoginInformation'
{ "isLoginRequired" : true }
```

Example using the Python module:

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

Example using the command line:

```
./TeamDriveApi.py '127.0.0.1:45454' getLoginInformation
{
    "username": "My Username",
    "apiUrl": "/",
    "isLoginRequired": false
}
```

Example Session (Initial Login):

```
# Start TeamDrive:
exec /opt/teamdrive/teamdrived

# Wait for it:
sleep 5

# Log-in
./TeamDriveApi.py '127.0.0.1:45454' --user myUser --pass myPass login

# Create a new Space
./TeamDriveApi.py '127.0.0.1:45454' --user myUser --pass myPass createSpace

# Call Functions
./TeamDriveApi.py '127.0.0.1:45454' --user myUser --pass myPass createFolder

# Quit TeamDrive without logout
./TeamDriveApi.py '127.0.0.1:45454' --user myUser --pass myPass quit False
```

Example Session:

```
exec /opt/teamdrive/teamdrived
sleep 5

# No need to login now
./TeamDriveApi.py '127.0.0.1:45454' --user myUser --pass myPass createFolder

# Quit TeamDrive without logout
./TeamDriveApi.py '127.0.0.1:45454' --user myUser --pass myPass quit False
```

Example Session (After previous Logout:

```
# If the previous session ended with a logout:
./TeamDriveApi.py '127.0.0.1:45454' --user myUser --pass myPass quit True

# The next call should start with a login:
exec /opt/teamdrive/teamdrived
sleep 5
./TeamDriveApi.py '127.0.0.1:45454' --user myUser --pass myPass login
./TeamDriveApi.py '127.0.0.1:45454' --user myUser --pass myPass createFolder
```

See also:

`TeamDriveApi.TeamDriveException` (page 52) for Errors returned by the API.

See also:

Chapter *Authentication* (page 22) regarding Authentication.

10.5.2 Communicating with the /files endpoint

See `TeamDriveApi.TeamDriveApi.putFile()` (page 45) for Details.

10.5.3 Passing Parameters to the API

The format of parameters depend on the HTTP method:

- Parameters for GET-requests should be passed by a query string in the url. For example `curl --user myTeamDriveUser http://127.0.0.1:45454/api/getSpace?id=310`
- Parameters to POST-requests can either be a JSON object or data encoded using `application/x-www-form-urlencoded`.

10.6 Interface Description

10.6.1 TeamDriveApi.TeamDriveApi

class `TeamDriveApi.`**`TeamDriveApi`**(*server=’[::1]:45454’*, *username=’‘*, *password=’‘*)

Return type *TeamDriveApi* (page 25)

Raises

- **`httplib.HTTPException`** – a Connection Error.
- **`socket.error`** – a Connection Error.
- **`ConnectionRefusedError`** – a Connection Error.

about ()

Returns information about TeamDrive.

Returns

```
{
  "version": "4.0.12 (Build: 1294)",
}
```

Changed in version 4.1.1: no longer returns license infos. call *getLicense* (page 35) instead.

Return type dict

Raises

- **httplib.HTTPException** – a Connection Error.
- **socket.error** – a Connection Error.
- **ConnectionRefusedError** – a Connection Error.
- **ValueError** – no JSON returned. Wrong URL?

addAddressbook (*name*)

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

Using this function requires the HTTP POST method.

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

Returns the id of the new entry.

Return type dict

Raises

- **httplib.HTTPException** – a Connection Error.
- **socket.error** – a Connection Error.
- **ConnectionRefusedError** – a Connection Error.
- **TeamDriveException** – user is unknown.
- **ValueError** – no JSON returned. Wrong URL?

addComment (*fileId*, *comment*)

Attaches the comment to the file.

Using this function requires the HTTP POST method.

See also:

getComments (page 31), *getSpaceComments* (page 38), *addSpaceComment* (page 27) and *removeComment* (page 46)

New in version 4.1.1.

Parameters

- **fileId** (*int*) – The corresponding id of the file
- **comment** (*str*) – The content of the new comment

Return type dict

Raises

- **`httplib.HTTPException`** – a Connection Error.
- **`socket.error`** – a Connection Error.
- **`ConnectionRefusedError`** – a Connection Error.
- **`TeamDriveException`** – Adding this comment failed.
- **`ValueError`** – no JSON returned. Wrong URL?

`addSpaceComment`(*spaceId*, *comment*)

Adds this comment to the Space.

Using this function requires the HTTP POST method.

See also:

`getComments` (page 31), *`getSpaceComments`* (page 38), *`addComment`* (page 26) and *`removeComment`* (page 46)

New in version 4.1.1.

Parameters

- **`spaceId`** (*int*) – The Space id.
- **`comment`** (*str*) – The content of the new comment

Return type dict

Raises

- **`httplib.HTTPException`** – a Connection Error.
- **`socket.error`** – a Connection Error.
- **`ConnectionRefusedError`** – a Connection Error.
- **`TeamDriveException`** – Adding this comment failed
- **`ValueError`** – no JSON returned. Wrong URL?

`clearProfilePicture`()

Clears the profile picture of the current user.

See also:

`getProfilePicture` (page 36) to retrieve profile pictures and *`setProfilePicture`* (page 50)

New in version 4.1.1.

Return type dict

Raises

- **`httplib.HTTPException`** – a Connection Error.
- **`socket.error`** – a Connection Error.
- **`ConnectionRefusedError`** – a Connection Error.
- **`ValueError`** – no JSON returned. Wrong URL?

`createFolder`(*spaceId*, *filePath*, *trashed*)

Creates a new folder in that Space

Example:

```
$ ./TeamDriveApi.py '127.0.0.1:45454' createFolder 3 '/my new folder' False
```

```
{
  "result": true,
  "file": {
        "isDir": true,
        "name": "my new folder",
        "currentVersionId": 0,
        "spaceId": 3,
        "isInFileSystem": false,
        "path": "/",
        "id": 756,
        "permissions": "rwxr-xr-x"
  }
}
```

See also:

`getFullFolderContent` (page 34) or `getFolderContent` (page 33) for getting the content of this folder

New in version 4.0.

Parameters

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

Return type dict

Raises

- **`httplib.HTTPException`** – a Connection Error.
- **`socket.error`** – a Connection Error.
- **`ConnectionRefusedError`** – a Connection Error.
- **`ValueError`** – no JSON returned. Wrong URL?

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

Creates a new Space with the given name.

To convert an existing folder into a TeamDrive Space, you need to set the *importExisting* parameter to True.

In case the url is called directly, the default for *disableFileSystem* is False and the *spacePath* is mandatory.

Using this function requires the HTTP POST method.

Throws `TeamDriveException` (page 52), if the call fails.

Parameters

- **`spaceName`** (*str*) – The Name of the new Space
- **`disableFileSystem`** (*bool*) – This will disable the synchronization into the file system. TeamDrive will only synchronize meta-data if this is *True*.
- **`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.
- **`socket.error`** – a Connection Error.
- **`ConnectionRefusedError`** – a Connection Error.
- **`TeamDriveException`** – Space creation failed.
- **`ValueError`** – no JSON returned. Wrong URL?

`deleteFileFromTrash`(*fileId*)

Deletes a file from Trash by physically deleting the file from the server. This call fails, if you do not have the rights to delete files, it is a directory or the file is not in the trash.

Using this function requires the HTTP POST method.

Example:

```
>>> api = TeamDriveApi()
>>> for inTrash in api.getFullFolderContent(1, "/", True):
>>>     api.deleteFileFromTrash(inTrash["id"])
```

Throws *`TeamDriveException`* (page 52), if the call fails.

See also:

`removeLocallyFile` (page 47) and *`setTrashed`* (page 51).

New in version 4.0.

Parameters **`fileId`** (*int*) – The Id of the Space

Return type dict

Raises

- **`httplib.HTTPException`** – a Connection Error.
- **`socket.error`** – a Connection Error.
- **`ConnectionRefusedError`** – a Connection Error.
- **`TeamDriveException`** – Failed to queue command.
- **`ValueError`** – no JSON returned. Wrong URL?

`deleteSpace`(*id*, *delInFs*, *delOnServer*, *delInDB*)

Deletes a Space.

Using this function requires the HTTP POST method.

Throws *`TeamDriveException`* (page 52), if the call fails.

Parameters

- **`id`** (*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)
- **`delInDB`** (*bool*) – Deletes the Space in the local Database

Return type dict

Raises

- **httplib.HTTPException** – a Connection Error.
- **socket.error** – a Connection Error.
- **ConnectionRefusedError** – a Connection Error.
- **TeamDriveException** – Deletion failed
- **ValueError** – no JSON returned. Wrong URL?

getActivities()

Returns a list of all activities in the last 30 days. If `hasDetails` is `true`, a call to *getActivityDetails* (page 30) with the given activityId will return a list of details.

Possible activity types are:

- `userInvited`
- `userJoined`
- `userLeft`
- `userKicked`
- `commentCreated`
- `commentDeleted`,
- `spaceCommentCreated`
- `spaceCommentDeleted`
- `versionAdded`
- `versionDeleted`
- `versionPublished`
- `versionUnpublished`
- `versionExpired`

See also:

getActivityDetails (page 30)

New in version 4.1.1.

Returns A list of activities.

Return type list[dict]

Raises

- **httplib.HTTPException** – a Connection Error.
- **socket.error** – a Connection Error.
- **ConnectionRefusedError** – a Connection Error.
- **ValueError** – no JSON returned. Wrong URL?

getActivityDetails(*activityId*)

Returns a list of all activity details for an activity.

See also:

getActivities (page 30)

New in version 4.1.1.

Parameters **activityId** (*str*) – The activityId as returned by *getActivities* (page 30)

Returns A list of activityDetails.

Return type list[dict]

Raises

- **`httplib.HTTPException`** – a Connection Error.
- **`socket.error`** – a Connection Error.
- **`ConnectionRefusedError`** – a Connection Error.
- **`ValueError`** – no JSON returned. Wrong URL?

`getAddressbook`(*id*)

Returns information about a given Addressbook Id.

See also:

`getFullAddressbook` (page 33) for an example

Parameters **`id`** (*int*) – the Address Id.

Return type dict

Raises

- **`httplib.HTTPException`** – a Connection Error.
- **`socket.error`** – a Connection Error.
- **`ConnectionRefusedError`** – a Connection Error.
- **`ValueError`** – no JSON returned. Wrong URL?

`getAddressbookByName`(*addressName*)

This is convenience wrapper of the Python module that is not part of the official API.

Returns an Addressbook entry by a given user 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.
- **`socket.error`** – a Connection Error.
- **`ConnectionRefusedError`** – a Connection Error.
- **`TeamDriveCallFailed`** – There is no Space with this name
- **`ValueError`** – no JSON returned. Wrong URL?

`getAddressbookIds`()

Returns all knows Addressbook Ids.

Return type list[int]

Raises

- **`httplib.HTTPException`** – a Connection Error.
- **`socket.error`** – a Connection Error.
- **`ConnectionRefusedError`** – a Connection Error.
- **`ValueError`** – no JSON returned. Wrong URL?

`getComments` (*fileId*)

Returns a list of all comments that are attached to a specific file in a space.

Example

```
>>> api = TeamDriveApi()
>>> api.getComments(4183)
```

Returns

```
[
  {
        "creator": "test",
        "creatorAddressId": 1,
        "text": "file baz",
        "created": "2015-09-01T15:15:31Z",
        "commentId": 16,
        "initials": "te"
  },
  {
        "creator": "test",
        "creatorAddressId": 1,
        "deleted": true,
        "created": "2015-09-01T15:15:28Z",
        "commentId": 15,
        "initials": "te"
  }
]
```

See also:

`getSpaceComments` (page 38), *`addComment`* (page 26), *`addSpaceComment`* (page 27) and *`removeComment`* (page 46)

New in version 4.1.1.

Parameters **`fileId`** (*int*) – The corresponding id of the file

Returns A list of comments.

Return type list[dict]

Raises

- **`httplib.HTTPException`** – a Connection Error.
- **`socket.error`** – a Connection Error.
- **`ConnectionRefusedError`** – a Connection Error.
- **`ValueError`** – no JSON returned. Wrong URL?

`getFile` (*id*)

Returns information about the given File Id.

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

Return type dict

Raises

- **`httplib.HTTPException`** – a Connection Error.
- **`socket.error`** – a Connection Error.

- **`ConnectionRefusedError`** – a Connection Error.
- **`ValueError`** – no JSON returned. Wrong URL?

`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.
- **`socket.error`** – a Connection Error.
- **`ConnectionRefusedError`** – a Connection Error.
- **`ValueError`** – no JSON returned. Wrong URL?

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

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

Example:

```
python ./TeamDriveApi.py '127.0.0.1:45454' getFolderContent 3 / False
[756,767,737,769,629,763,628,630,830,765]
```

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.
- **`socket.error`** – a Connection Error.
- **`ConnectionRefusedError`** – a Connection Error.
- **`ValueError`** – no JSON returned. Wrong URL?

`getFullAddressbook` ()

Returns all Addressbook entries at once.

Example:

```
$ python ./TeamDriveApi.py '127.0.0.1:45454' getFullAddressbook
```

```
[
  {
        "id": 1,
        "email": "info@teamdrive.net",
        "name": "My Name",
        "icon": "self"
  }
]
```

Note: As this function was added with TeamDrive 4, the fall back for TeamDrive 3 is to iterate over all Addressbook Ids, therefore this function is much slower.

New in version 4.0.

Return type list[dict]

Raises

- **`httplib.HTTPException`** – a Connection Error.
- **`socket.error`** – a Connection Error.
- **`ConnectionRefusedError`** – a Connection Error.
- **`ValueError`** – no JSON returned. Wrong URL?

`getFullFolderContent` (*spaceId*, *filePath*, *trashed*)

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

Example:

```
$ python ./TeamDriveApi.py '127.0.0.1:45454' getFullFolderContent 3 / False
```

```
[
  {
        "isDir": true,
        "name": "My folder",
        "currentVersionId": 0,
        "spaceId": 3,
        "isInFileSystem": true,
        "path": "/",
        "id": 756,
        "permissions": "rwxr-xr-x"
  },
  {
        "isDir": false,
        "name": "My File",
        "currentVersionId": 0,
        "spaceId": 3,
        "isInFileSystem": true,
        "path": "/",
        "id": 767,
        "permissions": "rwxr-xr-x"
  }
]
```

Note: As this function was added with TeamDrive 4, the fall back for TeamDrive 3 is to iterate over getFolderContent, therefore this function is much slower.

See also:

`getFile` (page 32) and `getFolderContent` (page 33)

New in version 4.0.

Parameters

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

Return type list[dict]

Raises

- **`httplib.HTTPException`** – a Connection Error.
- **`socket.error`** – a Connection Error.
- **`ConnectionRefusedError`** – a Connection Error.
- **`ValueError`** – no JSON returned. Wrong URL?

`getLicense`()

Returns information about the current License.

Returns

```
{
  "distributor": "DOCU",
  "licenseKey": "DOCU-8782-5485-7266",
  "licenses": [
        "WebDAV",
        "Professional"
  ]
}
```

New in version 4.1.1.

See also:

`setLicenseKey` (page 48) to set the license.

Return type dict

Raises

- **`httplib.HTTPException`** – a Connection Error.
- **`socket.error`** – a Connection Error.
- **`ConnectionRefusedError`** – a Connection Error.
- **`ValueError`** – no JSON returned. Wrong URL?

`getLoginInformation`()

Returns information about the current user.

Return type dict

Raises

- **`httplib.HTTPException`** – a Connection Error.

- **socket.error** – a Connection Error.
- **ConnectionRefusedError** – a Connection Error.
- **ValueError** – no JSON returned. Wrong URL?

getMember (*spaceId*, *addressId*)

Returns information about a member of a Space.

Example:

```
$ python ./TeamDriveApi.py '127.0.0.1:45454' getMember 3 1
```

Returns

```
{
        "status": "active",
        "spaceId": 3,
        "addressId": 1
}
```

Parameters

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

Return type dict

Raises

- **httplib.HTTPException** – a Connection Error.
- **socket.error** – a Connection Error.
- **ConnectionRefusedError** – a Connection Error.
- **ValueError** – no JSON returned. Wrong URL?

getProfilePicture (*id*)

Returns the profile picture of the address.

See also:

setProfilePicture (page 50) to set the profile picture of the current user.

New in version 4.1.1.

Parameters **id** (*int*) – The addressId.

Returns The picture.

Return type bytes | dict

Raises

- **httplib.HTTPException** – a Connection Error.
- **socket.error** – a Connection Error.
- **ConnectionRefusedError** – a Connection Error.

getSettings ()

Returns a *list* of settings, with each setting a *dict*. An incomplete list of settings can be found in the *Options* (page 17) Section.

Example :

```
        {
                "key": "identifier",
                "label": "readable label",
                "help": "help text",
                "type": "string|path|integer|bool|enum",
                "value": "current value",
                "options": ["enum option 1", "enum option 2"]
        }
```

setSetting (page 50) can be used to set a specific setting.

Return type list

Raises

- **httplib.HTTPException** – a Connection Error.
- **socket.error** – a Connection Error.
- **ConnectionRefusedError** – a Connection Error.
- **ValueError** – no JSON returned. Wrong URL?

getSpace (*id*)

Will return information about that Space.

See also:

getSpaceStatistics (page 40) for statistics.

Example using Python:

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

Example using curl:

```
curl --user myTeamDriveUser http://127.0.0.1:45454/api/getSpace?id=310
```

Returns

```
        {
                "name": "Api",
                "creator": "...",
                "spaceRoot": "/home/teamdrive/Spaces/Api",
                "icon": "space",
                "status": "active",
                "status_message": "OK",
                "id": 8,
                "fileId": 1
        }
```

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

Return type dict

Raises

- **httplib.HTTPException** – a Connection Error.
- **socket.error** – a Connection Error.

- **ConnectionRefusedError** – a Connection Error.
- **ValueError** – no JSON returned. Wrong URL?

getSpaceByName (*spaceName*)

This is convenience wrapper of the Python module that is not part of the official API.

Returns a Space by a given Space name. Throws a *TeamDriveException* (page 52), 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.
- **ConnectionRefusedError** – a Connection Error.
- **TeamDriveCallFailed** – There is no Space with this name
- **ValueError** – no JSON returned. Wrong URL?

getSpaceComments (*spaceId*)

Returns a list of all comments that are attached to the Space itself.

Example

```
>>> api = TeamDriveApi()
>>> api.getSpaceComments(13)
```

Returns

```
[
  {
        "creator": "test",
        "creatorAddressId": 1,
        "text": "file baz",
        "created": "2015-09-01T15:15:31Z",
        "commentId": 16,
        "initials": "te"
  },
  {
        "creator": "test",
        "creatorAddressId": 1,
        "deleted": true,
        "created": "2015-09-01T15:15:28Z",
        "commentId": 15,
        "initials": "te"
  }
]
```

See also:

getComments (page 31), *addComment* (page 26), *addSpaceComment* (page 27) and *removeComment* (page 46)

New in version 4.1.1.

Parameters **spaceId** (*int*) – The Space id.

Returns A list of comments.

Return type list[dict]

Raises

- **httplib.HTTPException** – a Connection Error.
- **socket.error** – a Connection Error.
- **ConnectionRefusedError** – a Connection Error.
- **ValueError** – no JSON returned. Wrong URL?

getSpaceIds()

Returns a list of all known Spaces Ids.

Example:

```
>>> api = TeamDriveApi()
>>> api.getSpaceIds()
[1,4,5,6]
```

Return type list[int]

Raises

- **httplib.HTTPException** – a Connection Error.
- **socket.error** – a Connection Error.
- **ConnectionRefusedError** – a Connection Error.
- **ValueError** – no JSON returned. Wrong URL?

getSpaceMemberIds(*spaceId*)

Returns all Addressbook Ids in a given Space.

Example:

```
$ python ./TeamDriveApi.py '127.0.0.1:45454' getSpaceMemberIds 3
[1,2]
```

New in version 4.0.

Deprecated since version 4.1.1: Use *getSpaceMembers* (page 39) instead.

Return type list[int]

Raises

- **httplib.HTTPException** – a Connection Error.
- **socket.error** – a Connection Error.
- **ConnectionRefusedError** – a Connection Error.
- **ValueError** – no JSON returned. Wrong URL?

getSpaceMembers(*spaceId*)

Returns all Member infos in a given Space.

New in version 4.1.

Return type list[dict]

Raises

- **httplib.HTTPException** – a Connection Error.
- **socket.error** – a Connection Error.
- **ConnectionRefusedError** – a Connection Error.
- **ValueError** – no JSON returned. Wrong URL?

getSpacePermissionLevels()

Get a list of possible space permission "levels" (ie. read, read/write, admin, etc.) and their capabilities. Permission levels are returned in order of increasing "magnitude". The reply of this call is constant and won't change until the agent is restarted.

Returns

```
[
        {
                "id": "none",
                "name: "None",
                "rights": {
                                "memberInvite": False,
                                "memberKick": False,
                                "memberSetRights": False,
                                "spaceRename": False,
                                "spaceDeleteOnServer": False,
                                "filePublish": False,
                                "fileDeleteOnServer": False,
                                "fileRename": False,
                                "fileRestoreFromTrash": False,
                                "fileChange": False
                                }
        },
        {
                "id": "read",
                "name": "Read"
                "rights": { ... }
        },
        ...
]
```

New in version 4.1.1.

Return type dict

Raises

- **httplib.HTTPException** – a Connection Error.
- **socket.error** – a Connection Error.
- **ConnectionRefusedError** – a Connection Error.
- **ValueError** – no JSON returned. Wrong URL?

getSpaceStatistics(*id*)

Returns Space statistics

New in version 4.1.1.

See also:

`getSpace` (page 37), `getSpaces` (page 41) for other information.

Parameters **`id`** (*int*) – The Id of the Space

Return type dict

Raises

- **`httplib.HTTPException`** – a Connection Error.
- **`socket.error`** – a Connection Error.
- **`ConnectionRefusedError`** – a Connection Error.
- **`ValueError`** – no JSON returned. Wrong URL?

`getSpaces`()

Returns a list of all known Spaces.

Note: As this function was added with TeamDrive 4, the fall back for TeamDrive 3 is to iterate over all Space Ids, therefore this function is much slower.

See also:

`getSpaceStatistics` (page 40) for statistics.

Returns

```
[
  {
        "name": "My Space",
        "creator": "$TMDR-1007",
        "icon": "archived",
        "status": "archived",
        "status_message": "Space Archived",
        "id": 1,
        "spaceRoot": "/home/teamdrive/Spaces/My Space",
        "time": "2015-09-15T11:17:48Z",
        "permissionLevel": "read"
  },
  {
        "name": "Api",
        "creator": "...",
        "spaceRoot": "/home/teamdrive/Spaces/Api",
        "icon": "space",
        "status": "active",
        "status_message": "OK",
        "id": 8,
        "fileId": 1,
        "time": "2015-09-15T11:17:48Z",
        "permissionLevel": "readWrite"
  }
]
```

New in version 4.0.

Return type dict

Raises

- **`httplib.HTTPException`** – a Connection Error.
- **`socket.error`** – a Connection Error.

- **ConnectionRefusedError** – a Connection Error.
- **ValueError** – no JSON returned. Wrong URL?

inviteMember(*id*, *addressId*, *text*, *permissionLevel=None*, *password=None*)

Invites a user into a Space.

Using this function requires the HTTP POST method.

Throws *TeamDriveException* (page 52), if the call fails.

Parameters

- **id** (*int*) – The Id of the Space
- **addressId** (*int*) – Already known Addressbook Id.
- **text** (*str*) – Invitation Text
- **permissionLevel** (*str* | *None*) – The new user permission Level. Optional.

  New in version 4.1.1.
- **password** (*str* | *None*) – Optional. The password for the invitation.

  New in version 4.1.1.

Return type dict

Raises

- **httplib.HTTPException** – a Connection Error.
- **socket.error** – a Connection Error.
- **ConnectionRefusedError** – a Connection Error.
- **TeamDriveException** – Inviting that member failed
- **ValueError** – no JSON returned. Wrong URL?

joinSpace(*id*, *disableFileSystem*, *spacePath=None*, *password=None*)

Accepts an invitation, joins an archived Space or rejoins an active one.

Note: Password protected Invitations: Invitations may be protected by a password and accepting these invitations require the password parameter.If an invitation is password protected, *getSpace* (page 37) adds `"isPasswordProtected": true` to the reply.

Using this function requires the HTTP POST method.

Throws *TeamDriveException* (page 52), if the call fails.

Parameters

- **id** (*int*) – The Id of the Space
- **disableFileSystem** (*bool*) – This will disable the synchronization into the file system. TeamDrive will only synchronize meta-data if this is *True*.
- **spacePath** (*str* | *None*) – Optional, Space root in the file system.
- **password** (*str* | *None*) – Optional, Password for accepting this invitation.

  New in version 4.1.1.

Return type dict

Raises

- **httplib.HTTPException** – a Connection Error.

- **socket.error** – a Connection Error.
- **ConnectionRefusedError** – a Connection Error.
- **TeamDriveException** – Joining failed
- **ValueError** – no JSON returned. Wrong URL?

kickMember (*spaceId*, *addressId*)

Removes a member form this Space.

Removing a member from a Space will not remove this user from *getSpaceMemberIds* (page 39), but will set a this member to status “kicked” in *getMember* (page 36).

Using this function requires the HTTP POST method.

Throws *TeamDriveException* (page 52), if the call fails. For example, if the one doesn’t have the rights to remove a user

New in version 4.0.

Parameters

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

Return type dict

Raises

- **httplib.HTTPException** – a Connection Error.
- **socket.error** – a Connection Error.
- **ConnectionRefusedError** – a Connection Error.
- **TeamDriveException** – Removing this user failed
- **ValueError** – no JSON returned. Wrong URL?

login ()

A TeamDrive Client installation is bound to a single TeamDrive account. Calling *login* (page 43) authenticates TeamDrive user and creates a new device. this is typically the first call to an Agent.

Using this function requires the HTTP POST method.

Throws *TeamDriveException* (page 52), if the call fails.

Example as Command line:

```
./TeamDriveApi.py '127.0.0.1:45454' --user myUser --pass myPass login
```

Example with Curl:

```

    curl 'https://127.0.0.1:45454/login' --data-binary '{"username":"My User","password":"My
```

Example:

```
>>> api = TeamDriveApi("127.0.0.1:45454", "My Username", "My Password")
>>> api.login()
```

Note: On the desktop client, this call is not available, as the API will not be available until the user has successfully logged in. This is a technical limitation of the Gui Client.

See also:

Chapter *Authentication* (page 22) regarding Authentication.

Parameters

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

Return type dict

Raises

- **`httplib.HTTPException`** – a Connection Error.
- **`socket.error`** – a Connection Error.
- **`ConnectionRefusedError`** – a Connection Error.
- **`TeamDriveException`** – Logging failed.
- **`ValueError`** – no JSON returned. Wrong URL?

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

Moves a file from *filePath* to *newFilePath* in the given Space.

Using this function requires the HTTP POST method.

Throws `TeamDriveException` (page 52), 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.
- **`socket.error`** – a Connection Error.
- **`ConnectionRefusedError`** – a Connection Error.
- **`TeamDriveException`** – Failed to queue command.
- **`ValueError`** – no JSON returned. Wrong URL?

`publishFile` (*fileId*, *password=None*, *expiraryDate=None*)

Publishes the recent version of that file. The parameters *password* and *expiraryDate* are optional. Throws `TeamDriveException` (page 52), if the call fails.

New in version 4.1.

Parameters

- **`fileId`** (*int*) – The Id of the Space
- **`password`** (*str*) – Is the file in the trash?
- **`expiraryDate`** (*str*) – ISO 8601 extended format: either YYYY-MM-DD for dates or YYYY-MM-DDTHH:mm:ss, YYYY-MM-DDTHH:mm:ssTZD (e.g., 1997-07-16T19:20:30+01:00) for combined dates and times.

Return type dict

Raises

- **httplib.HTTPException** – a Connection Error.
- **socket.error** – a Connection Error.
- **ConnectionRefusedError** – a Connection Error.
- **TeamDriveException** – Failed to queue command.
- **ValueError** – no JSON returned. Wrong URL?

putFile (*spaceId*, *path*, *data*)

This is a Python specific accessor to the "files" API. The files API is another API endpoint to handle file content instead of meta data. Urls used by the files API are generated by concatenating the API endpoint `files/` plus the local Space Id (eg. `3`) plus the full file path relative to the Space root (eg. `/my/path/to/the/file`). Such a url could look like this: `http://127.0.0.1:45454/files/3/my/path/to/the/file`. This API does not support reading and writing files that are in the trash.

This Uploads a file into a Space. The file creation is done asynchronously, a successful return code does not indicate a successful upload to the Server. Use *getFiles* (page 33) to request the current state of that file.

Example using telnet:

```
telnet 127.0.0.1 45454
PUT /files/3/testfile HTTP/1.1
Host: 127.0.0.1:45454
Accept-Encoding: identity
Content-Length: 4

test
```

Example using curl:

```
echo "test" | curl -X PUT -d @- http://127.0.0.1:45454/files/3/testfile
```

Returns

```
{
  "file": {
        "confirmed": false,
        "creator": "local",
        "currentVersionId": 6156,
        "hasComments": false,
        "icon": "default",
        "id": 7195,
        "isDir": false,
        "name": "testfile",
        "path": "/",
        "permissions": "rw-rw-r--",
        "size": 4,
        "spaceId": 3,
        "time": "2015-10-22T14:09:45"
  },
  "result": true
}
```

Note: In TeamDrive 3, this API endpoint was called `/webdav`.

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.
- **`socket.error`** – a Connection Error.
- **`ConnectionRefusedError`** – a Connection Error.

`putFileContent`(*spaceId*, *spacePath*, *filePath*)

Note: This is not part of the HTTP API. See *`putFile`* (page 45) instead.

Example:

```
$ python ./TeamDriveApi.py '127.0.0.1:45454' putFileContent 3 /testfile ./testfile
```

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.

Using this function requires the HTTP POST method.

Throws *`TeamDriveException`* (page 52), if the call fails.

Parameters `logout` (*bool*) – Quit and also log out the current user. See *`login`* (page 43) to log in afterwards.

Return type dict

Raises

- **`httplib.HTTPException`** – a Connection Error.
- **`socket.error`** – a Connection Error.
- **`ConnectionRefusedError`** – a Connection Error.
- **`ValueError`** – no JSON returned. Wrong URL?

`removeComment`(*commentId*)

Marks this comment as "removed"

Using this function requires the HTTP POST method.

See also:

`getComments` (page 31), *`getSpaceComments`* (page 38), *`addComment`* (page 26) and *`addSpaceComment`* (page 27)

New in version 4.1.1.

Parameters **`commentId`** (*int*) – The id of the comment that will be deleted.

Return type dict

Raises

- **`httplib.HTTPException`** – a Connection Error.
- **`socket.error`** – a Connection Error.
- **`ConnectionRefusedError`** – a Connection Error.
- **`TeamDriveException`** – Removing this comment failed
- **`ValueError`** – no JSON returned. Wrong URL?

`removeLocallyFile`(*id*, *recursive=False*)

Removes a file form the local file system by disabling the offline-available flag, but does not move this file into the trash or removes the file from the Space. This operation is not synchronized to other clients.

A successful return code does not indicate a successful execution, because it is done asynchronously.

Using this function requires the HTTP POST method.

See also:

`deleteFileFromTrash` (page 29) and *`setTrashed`* (page 51).

Parameters

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

Return type dict

Raises

- **`httplib.HTTPException`** – a Connection Error.
- **`socket.error`** – a Connection Error.
- **`ConnectionRefusedError`** – a Connection Error.
- **`TeamDriveException`** – Failed to queue command.
- **`ValueError`** – no JSON returned. Wrong URL?

`requestResetPassword`(*username*)

This call will reset the password of the user.

New in version 4.1.1.

Parameters **`username`** (*str*) – The Username

Return type dict

Raises

- **`httplib.HTTPException`** – a Connection Error.
- **`socket.error`** – a Connection Error.
- **`ConnectionRefusedError`** – a Connection Error.
- **`TeamDriveException`** – Logging failed.
- **`ValueError`** – no JSON returned. Wrong URL?

`restoreLocallyFile`(*id*, *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. This operation is not synchronized to other clients.

Using this function requires the HTTP POST method.

Parameters

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

Return type dict

Raises

- **`httplib.HTTPException`** – a Connection Error.
- **`socket.error`** – a Connection Error.
- **`ConnectionRefusedError`** – a Connection Error.
- **`TeamDriveException`** – Failed to queue command.
- **`ValueError`** – no JSON returned. Wrong URL?

`setAccountEmail`(*email*)

Sets the account e-mail.

Using this function requires the HTTP POST method.

Returns True, if the license key was successfully changed otherwise raises *`TeamDriveException`* (page 52) Exception

See also:

`setLicenseKey` (page 48) to set the license key.

New in version 4.1.1.

Parameters **`email`** (*str*) – The new e-mail address

Return type dict

Raises

- **`httplib.HTTPException`** – a Connection Error.
- **`socket.error`** – a Connection Error.
- **`ConnectionRefusedError`** – a Connection Error.
- **`ValueError`** – no JSON returned. Wrong URL?
- **`TeamDriveException`** – Failed to set email.

setLicenseKey(*licenseKey*)

Overwrite the license key of this agent.

Using this function requires the HTTP POST method.

Returns True, if the license key was successfully changed otherwise raises *TeamDriveException* (page 52) Exception

See also:

setAccountEmail (page 48) to set the registration e-mail address. *getLicense* (page 35) to retrieve license infos.

Parameters **licenseKey** (*str*) – The new license key

Return type dict

Raises

- **httplib.HTTPException** – a Connection Error.
- **socket.error** – a Connection Error.
- **ConnectionRefusedError** – a Connection Error.
- **ValueError** – no JSON returned. Wrong URL?

setMemberPermissionLevel(*spaceId*, *addressId*, *permissionLevel*)

Changes the permission level of another member.

Using this function requires the HTTP POST method.

Throws *TeamDriveException* (page 52), if the call fails. For example, if the one doesn't have the rights to remove a user

New in version 4.1.1.

Parameters

- **spaceId** (*int*) – The Id of the Space
- **addressId** (*int*) – Already known Addressbook Id.
- **permissionLevel** (*str*) – the new permission level

Return type dict

Raises

- **httplib.HTTPException** – a Connection Error.
- **socket.error** – a Connection Error.
- **ConnectionRefusedError** – a Connection Error.
- **TeamDriveException** – Removing this user failed
- **ValueError** – no JSON returned. Wrong URL?

setProfileInfo(*email=None*, *phone=None*, *mobile=None*)

Sets profile information of the current user.

See also:

setProfilePicture (page 50) to set the profile picture.

New in version 4.1.1.

Parameters

- **`email`** (*str | None*) – Optional, the new e-mail.
- **`phone`** (*str | None*) – Optional, the new phone number.
- **`mobile`** (*str | None*) – Optional, the new mobile number.

Raises

- **`httplib.HTTPException`** – a Connection Error.
- **`socket.error`** – a Connection Error.
- **`ConnectionRefusedError`** – a Connection Error.
- **`ValueError`** – no JSON returned. Wrong URL?

`setProfilePicture`(*data*)

Sets the profile picture of the current user.

See also:

`getProfilePicture` (page 36) to retrieve profile pictures and *`clearProfilePicture`* (page 27) to clear the picture.

New in version 4.1.1.

Parameters **`data`** (*bytes*) – The blob.

Return type dict

Raises

- **`httplib.HTTPException`** – a Connection Error.
- **`socket.error`** – a Connection Error.
- **`ConnectionRefusedError`** – a Connection Error.
- **`ValueError`** – no JSON returned. Wrong URL?

`setSetting`(*key*, *value*)

Set a Setting. must be a valid key as in *`getSettings`* (page 36). An incomplete list of settings can be found in the *Options* (page 17) Section.

Using this function requires the HTTP POST method.

Parameters

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

Return type dict

Raises

- **`httplib.HTTPException`** – a Connection Error.
- **`socket.error`** – a Connection Error.
- **`ConnectionRefusedError`** – a Connection Error.
- **`ValueError`** – no JSON returned. Wrong URL?

`setSpaceSetting`(*spaceId*, *key*, *value*)

set a Space Setting. must be a valid key as in `getSpaceSettings`.

Using this function requires the HTTP POST method.

New in version 4.1.2.

Parameters

- **`spaceId`** (*int*) – The Space id.
- **`key`** (*str*) – The name of the setting
- **`value`** – The new value of the setting

Return type dict

Raises

- **`httplib.HTTPException`** – a Connection Error.
- **`socket.error`** – a Connection Error.
- **`ConnectionRefusedError`** – a Connection Error.
- **`ValueError`** – no JSON returned. Wrong URL?

`setTrashed`(*fileId*, *inTrash*)

Moves a file into or out of the trash. Trashing a file also removes the file from the local file system.

Throws *`TeamDriveException`* (page 52), if the call fails.

Call *`deleteFileFromTrash`* (page 29) to delete the file physically afterwards.

See also:

`removeLocallyFile` (page 47) and *`deleteFileFromTrash`* (page 29).

New in version 4.0.

Parameters

- **`fileId`** (*int*) – The Id of the Space
- **`inTrash`** (*bool*) – Is the file in the trash?

Return type dict

Raises

- **`httplib.HTTPException`** – a Connection Error.
- **`socket.error`** – a Connection Error.
- **`ConnectionRefusedError`** – a Connection Error.
- **`TeamDriveException`** – Failed to queue command.
- **`ValueError`** – no JSON returned. Wrong URL?

`sse`()

This is the Server Side Events api listening on */sse*. This API sends events according to the Server Side Events specification.

The implementation of the python wrapper requires Python 3.

Example Usage:

```
$ python3 TeamDriveApi.py '127.0.0.1:45454' sse
data: { "event" : "FileEntryChanged", "fileId" : 0, "hasConflict" : 0, "isDir" :
                false, "isTrashed" : false, "name" : "my file", "path" :
                "/home/teamdrive/Spaces/My Space", "spaceId" : 3, "synced" : true }

data: { "event" : "SpaceModified", "spaceId" : 3 }
```

Possible Event types are: *FileEntryChanged*, *VersionEntryChanged*, *FileVersionPublished*, *FileVersionUnpublished* and *SpaceModified*

Raises

- **`httplib.HTTPException`** – a Connection Error.
- **`socket.error`** – a Connection Error.
- **`ConnectionRefusedError`** – a Connection Error.
- **`ValueError`** – no JSON returned. Wrong URL?

`unpublishFile` (*fileId*)

Unpublishes the file. Throws `TeamDriveException` (page 52), if the call fails.

New in version 4.1.

Parameters **`fileId`** (*int*) – The Id of the Space

Return type dict

Raises

- **`httplib.HTTPException`** – a Connection Error.
- **`socket.error`** – a Connection Error.
- **`ConnectionRefusedError`** – a Connection Error.
- **`TeamDriveException`** – Failed to queue command.
- **`ValueError`** – no JSON returned. Wrong URL?

10.6.2 TeamDriveApi.TeamDriveException

class `TeamDriveApi.`**`TeamDriveException`** (*name*, *result*)

In case of an error, the TeamDrive API returns a specific JSON document. Thy python Module will parse this error.

Example using curl:

```
curl -s '127.0.0.1:45454/api/getSpaces'
{
  "error": 50,
  "error_message": "CJsonApi::ApiError (Error_Missing_Session)",
  "result": false,
  "status_code": 500
}
```

Example using the command line:

```
$ ./TeamDriveApi.py '127.0.0.1:45454' getSpaces
Traceback (most recent call last):
  File "./TeamDriveApi.py", line 1178, in <module>
        main()
  File "./TeamDriveApi.py", line 1174, in main
        result = (getattr(TeamDriveApi, command)(api, *params))
  File "./TeamDriveApi.py", line 307, in getSpaces
        return self._call("getSpaces")
  File "./TeamDriveApi.py", line 204, in _call
        return self._call(name, params, method)
  File "./TeamDriveApi.py", line 205, in _call
```

```
        raise TeamDriveException(name, jsonRes)
__main__.TeamDriveException: Call getSpaces failed with {u'status_code': 500, u'error_message':
```

Error_HTTP_Method_Not_Allowed = 2

Error_Insufficient_Rights = 30

Error_Internal_Error = 1

Error_Invalid_File_Key = 43

Error_Invalid_Parameter = 21

Error_Missing_Parameter = 20

Error_Missing_Session = 50
No or invalid credentials supplied.

See also:

Chapter *Authentication* (page 22).

Error_Unknown_AddressId = 42

Error_Unknown_Error = 3

Error_Unknown_Space = 40

Error_Unknown_User = 41

Error_Wrong_Login = 51

getError()
The TeamDrive API Exception code :rtype: int

getErrorString()
A human readable version of this error :rtype: str

getStatusCode()
The HTTP status code of this exception. :rtype: int

CHAPTER

ELEVEN

CHANGE LOG

11.1 TeamDrive 4.1.2

- added *setSpaceSetting* (page 50).
- added *teamdrived --enable-shell-extension* (page 18).

11.2 TeamDrive 4.1.1

- added *setAccountEmail* (page 48)
- *about* (page 25) no longer returns license info. call *getLicense* (page 35) instead.
- added *requestResetPassword* (page 47)
- added *getProfilePicture* (page 36), *setProfilePicture* (page 50), *clearProfilePicture* (page 27) and *setProfileInfo* (page 49)
- added *getComments* (page 31), *getSpaceComments* (page 38), *addComment* (page 26), *addSpaceComment* (page 27) and *removeComment* (page 46)
- deprecated *getSpaceMemberIds* (page 39)
- added *getSpacePermissionLevels* (page 40), *setMemberPermissionLevel* (page 49), added permission levels to Member related calls.
- added *getSpaceStatistics* (page 40)
- added ability to handle password protected invitations in *joinSpace* (page 42) and *inviteMember* (page 42)
- added *getActivities* (page 30) and *getActivityDetails* (page 30)
- added *teamdrived --idle-shutdown-timeout* (page 19), *teamdrived --http-api-external-register-url* (page 19) and *teamdrived --http-api-external-login-url* (page 19)

11.3 TeamDrive 4.1

- Added *createFolder* (page 27), *publishFile* (page 44) and *unpublishFile* (page 52).
- Added documentation of the */sse* API

11.4 TeamDrive 4.0

- All modifying calls require HTTP POST requests, for example createFolder or deleteSpace
- Client Authentication is **required**
- Added *getSpaces* (page 41), *getFullFolderContent* (page 34), *getFullAddressbook* (page 33), *setTrashed* (page 51) and *createFolder* (page 27)
- Changed the Exception response.
- added *getMember* (page 36) to retrieve the status of a member

11.5 TeamDrive 3.3.1

- added *getSpaceMemberIds* (page 39)

11.6 Web User Interface 1.0.2

- Create Space Dialog: Added Server selection. (Requires Agent 4.1.2).
- Improved error message if there is no depot usage available.
- Added automatic redirect to the external login url on the login page.
- Fixed a bug regarding custom user rights (Only TeamDrive 3 Spaces affected)