

# Teedy Installation and Configuration

## About Teedy Document Management System

Welcome to the unofficial documentation space for Teedy. Teedy (prior "Sismics Docs") is an open source enterprise content management system (ECM) and/or document management system (DMS) with a lot of features and a modern user interface. We use it for different purposes.

Source Code: [github.com/sismics/docs](https://github.com/sismics/docs)

Homepages: [sismics.com](https://sismics.com) and [teedy.io](https://teedy.io)

- [Multilingual Support](#)
- [Known Limitations](#)
- [Downloads](#)
- [Security](#)
- [Automatic Mail Importing \(Inbox Scanning\)](#)
- [Environment Configuration](#)
- [Migration from H2 to PostgreSQL](#)
- [Apache Reverse Proxy and Firewall](#)
- [Teedy with PostgreSQL](#)
- [Access to the H2 Database](#)
- [Basic installation with Jetty and H2 Database](#)
- [Build web application server from Source](#)
- [Backup and restore strategies](#)
- [App for Android](#)

# Multilingual Support

Teedy is available in different languages. Please help to make some commits to fix your own language. You can use the following tool to validate your modified json files which contain translatable strings: <https://jsonlint.com>

Language file directories from git project:

- `teedy\docs-web\src\main\webapp\src\locale`
- `teedy\docs-core\src\main\resources`

# Known Limitations

- General issues can be found in <https://github.com/sismics/docs/issues>
- If you delete a user, all his assigned documents get deleted without asking. The proper way to omit data loss, just deactivate the user and properly assign permissions to all tags, so other people can continue working with these documents.
- standard H2 database gets really slow when having thousands of documents or documents with thousands of files inside → [Migration from H2 to PostgreSQL](#) helps! **Really**, PostgreSQL is the only useful way to be productive with Teedy!

# Downloads

Teedy consists of Java Server Application for Windows, MacOS or Linux Server and additionally:

- Android App
- Document Importer for Windows/Linux/MacOS

You can find precompiled [files in our Cloud](#) or you compile it yourself from sources.

There's also a Grafana Dashboard for more deep analysis available. See [Grafana Monitoring / Statistics](#)

You can make your own executable with source code. See [Build web application server from Source](#)

# Security

## General Tips

- run Teedy in Intranet
- use SSL
- adjust apache2 to hide /api/app information (json output which contains information like used version, user count, etc.)

```
<Location "/api/app">
    AllowOverride None
    Order deny,allow
    Deny from All
</Location>

<Location ~ "/api/app/.*">
    AllowOverride None
    Allow from All
</Location>

#rewrite /api/app/ to /api/app and so on. Otherwise api/app will be blocked but api/app/ will
not be blocked
RewriteEngine on
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)/$ /$1 [R=301,L]
```

- enable 2FA for each user account
- Change admin user name to something else

```
UPDATE t_user SET use_username_c = 'yournewusername' WHERE use_username_c = 'admin';
```

On Linux you can use <https://wiki.ubuntuusers.de/QtQR> to import a QR code image from Desktop in case you just want to generate TOTP tokens with tools like [KeePassXC](#) for instance. Just run qtqr and select "Decode from File" to open such QR code.

```
apt install qtqr
qtqr #run the app
```

The screenshot shows the 'QtQR: QR Code Generator' application window. It has a standard macOS title bar with minimize, maximize, and close buttons. The window is divided into several sections. On the left, under 'Select data type:', there is a dropdown menu currently set to 'Text' and a large empty rectangular box labeled 'Text to be encoded:'. On the right, under 'Parameters:', there are three settings: 'Pixel Size' with a numeric spinner set to 3, 'Error Correction' with a dropdown menu set to 'Lowest', and 'Margin Size' with a numeric spinner set to 4. Below these parameters is a large rectangular area containing the text 'Start typing to create QR Code or drop here image files for decoding.' At the bottom of the window, there are three buttons: 'Save QRCode' with a floppy disk icon, 'Decode from File' with a folder icon, and 'Decode from Webcam'.

# Automatic Mail Importing (Inbox Scanning)

## Inbox scanning **Enabled**

By enabling this feature, the system will scan the specified inbox every minute for **unread** emails and automatically import them.

After importing an email, it will be marked as read.

Configuration settings for [Gmail](#), [Outlook.com](#), [Yahoo](#).

Last synchronization: Jan 9, 2020 10:29:08 PM, 0 message imported

Enable inbox scanning ☒

IMAP hostname


IMAP port (143 or 993)

IMAP username

IMAP password

Tag added to imported documents

 Save

 Test the parameters

This feature can be configured by environment variables! → [Environment Configuration](#)

# Environment Configuration

We use Fedora 38 Workstation, but the following steps should be similar on other systems.

We also use a timer to delay the startup of jetty11 at bootup, because it sometimes struggles to perform well at the first start.

```
vim /lib/systemd/system/jetty11.timer
```

```
[Unit]
Description=Timer for jetty11 Startup Delay

[Timer]
OnBootSec=1min

[Install]
WantedBy=timers.target
```

```
vim /lib/systemd/system/jetty11.service
```

```
[Unit]
Description=Jetty 11 Web Application Server
Documentation=https://www.eclipse.org/jetty/documentation/current/
After = syslog.target network.target

[Service]
# Configuration
Environment="JETTY_HOME=/opt/jetty-home-11.0.15"
Environment="JETTY_BASE=/opt/jetty-home-11.0.15/jetty-base"
Environment="JETTY_USER=jetty"
Environment="JETTY_HOST=127.0.0.1"
Environment="JETTY_ARGS=jetty.port=8080"
Environment="JETTY_STATE=/var/lib/jetty11/jetty.state"

Environment="JAVA_OPTS=-Xms1024m -Xmx3584m -Djava.awt.headless=true"

#Configure Jetty Service to use database connection instead of H2 local DB
Environment="DATABASE_URL=jdbc:postgresql://127.0.0.1:5432/teedy_db"
```



```
Environment="DATABASE_USER=teedy"
Environment="DATABASE_PASSWORD=password"

#set base url for password reset
Environment="DOCS_BASE_URL=https://your.domain.tld"

#Configure tesseract performance
Environment="OMP_THREAD_LIMIT=1"

# Lifecycle
Type=forking
ExecStart = /opt/jetty-home-11.0.15/bin/jetty.sh start
ExecStop = /opt/jetty-home-11.0.15/bin/jetty.sh stop
ExecReload = /opt/jetty-home-11.0.15/bin/jetty.sh restart

# Logging
SyslogIdentifier=jetty11

# Security
User=jetty
Group=jetty
PrivateTmp=yes
AmbientCapabilities=CAP_NET_BIND_SERVICE
NoNewPrivileges=true
WorkingDirectory=/usr/share/jetty11/
LogsDirectory=jetty11
LogsDirectoryMode=750
ProtectSystem=
ReadWritePaths=/var/lib/jetty9/
ReadWritePaths=/mnt/data/sismics/

[Install]
WantedBy=multi-user.target
```

More environment vars can be found at

<https://github.com/sismics/docs/blob/dd36e08d7d6cd8248f12a9570694b4631be3b04d/README.md#available-environment-variables>

```
systemctl daemon-reload
systemctl enable jetty11.timer #we do not enable jetty11.service too, because that service is
just fully controlled by our timer
systemctl start jetty11.service
systemctl status jetty11.service
```

## Information about Teedy file structure

Directory	Notes
<code>/var/docs/theme</code>	Background image stored here
<code>/var/docs/storage</code>	place for alle the files and their automatically generated thumbnails (*_thumb) and web previews. Note that alle the files inside this dir are encrypted and got hashed file names. You can access the file content only inside a running Teedy instance which decrypts the files for you
<code>/var/docs/log</code>	log files. Same output which you get from <code>journalctl -u jetty9.service</code> (if you use systemd)
<code>/var/docs/db</code>	H2 database files. Empty but existing if PostgreSQL is used
<code>/var/docs</code>	Root Directory

# Migration from H2 to PostgreSQL

Note that the migration will make you losing all the OCR extracted text content from files in your documents. If you want to have those OCR strings back you will need to run a re-indexing for all of them.

## Stop Teedy and make backup of recent H2 file structure

```
service jetty9 stop
cd /var/
cp -R docs/ docs.bak/
chown -R jetty:jetty docs.bak/
chmod 777 docs.bak/
```

## Create some test instance of Teedy which is setup with PostgreSQL

Please see [Teedy with PostgreSQL](#) to get some points on how to do this.

## Dumping scheme data from (filled) existing Teedy PostgreSQL instance to get a starting point for migration

You will need to configure some temporary instance of Sismics to let it create the desired structure for you. This step is only required if you don't want to use the provided SQL statements in this documentation. Do that step if you want to do it for newer releases that will come up.

```
su - postgres #from root user
pg_dump --schema-only teedy_db > teedy_db.sql #that dump gets splitted in step1 and step2
later on
```

## Wipe away the psql test instance database and create a fresh/empty one again (for production use)

```
psql
drop database teedy_db;
CREATE DATABASE teedy_db WITH ENCODING 'UNICODE' LC_COLLATE 'C' LC_CTYPE 'C' TEMPLATE
template0;
GRANT ALL PRIVILEGES ON DATABASE teedy_db TO teedy;
\q
```

```

/*step1.sql*/
SET statement_timeout = 0;
SET lock_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SELECT pg_catalog.set_config('search_path', '', false);
SET check_function_bodies = false;
SET xmloption = content;
SET client_min_messages = warning;
SET row_security = off;CREATE EXTENSION IF NOT EXISTS plpgsql WITH SCHEMA pg_catalog;
COMMENT ON EXTENSION plpgsql IS 'PL/pgSQL procedural language';
SET default_tablespace = '';SET default_with_oids = false;CREATE TABLE public.t_acl (
    acl_id_c character varying(36) NOT NULL,
    acl_perm_c character varying(30) NOT NULL,
    acl_sourceid_c character varying(36) NOT NULL,
    acl_targetid_c character varying(36) NOT NULL,
    acl_deletedate_d timestamp without time zone,
    acl_type_c character varying(30) DEFAULT 'USER'::character varying NOT NULL
);
ALTER TABLE public.t_acl OWNER TO teedy;CREATE TABLE public.t_audit_log (
    log_id_c character varying(36) NOT NULL,
    log_identity_c character varying(36) NOT NULL,
    log_classentity_c character varying(50) NOT NULL,
    log_type_c character varying(50) NOT NULL,
    log_message_c character varying(1000),
    log_createdate_d timestamp without time zone,
    log_iduser_c character varying(36) DEFAULT 'admin'::character varying NOT NULL
);
ALTER TABLE public.t_audit_log OWNER TO teedy;CREATE TABLE public.t_authentication_token (
    aut_id_c character varying(36) NOT NULL,
    aut_iduser_c character varying(36) NOT NULL,
    aut_longlasted_b boolean NOT NULL,
    aut_creationdate_d timestamp without time zone NOT NULL,
    aut_lastconnectiondate_d timestamp without time zone,
    aut_ip_c character varying(45),
    aut_ua_c character varying(1000)
);
ALTER TABLE public.t_authentication_token OWNER TO teedy;CREATE TABLE public.t_base_function (
    baf_id_c character varying(20) NOT NULL
);

```

```

ALTER TABLE public.t_base_function OWNER TO teedy;CREATE TABLE public.t_comment (
    com_id_c character varying(36) NOT NULL,
    com_iddoc_c character varying(36) NOT NULL,
    com_iduser_c character varying(36) NOT NULL,
    com_content_c character varying(4000) NOT NULL,
    com_createdate_d timestamp without time zone,
    com_deletedate_d timestamp without time zone
);
ALTER TABLE public.t_comment OWNER TO teedy;CREATE TABLE public.t_config (
    cfg_id_c character varying(50) NOT NULL,
    cfg_value_c character varying(250) NOT NULL
);
ALTER TABLE public.t_config OWNER TO teedy;CREATE TABLE public.t_contributor (
    ctr_id_c character varying(36) NOT NULL,
    ctr_iduser_c character varying(36) NOT NULL,
    ctr_iddoc_c character varying(36) NOT NULL
);
ALTER TABLE public.t_contributor OWNER TO teedy;CREATE TABLE public.t_document (
    doc_id_c character varying(36) NOT NULL,
    doc_iduser_c character varying(36) NOT NULL,
    doc_title_c character varying(100) NOT NULL,
    doc_description_c character varying(4000),
    doc_createdate_d timestamp without time zone,
    doc_deletedate_d timestamp without time zone,
    doc_language_c character varying(7) DEFAULT 'eng'::character varying NOT NULL,
    doc_subject_c character varying(500),
    doc_identifier_c character varying(500),
    doc_publisher_c character varying(500),
    doc_format_c character varying(500),
    doc_source_c character varying(500),
    doc_type_c character varying(500),
    doc_coverage_c character varying(500),
    doc_rights_c character varying(500),
    doc_updatedate_d timestamp without time zone NOT NULL,
    doc_idfile_c character varying(36)
);
ALTER TABLE public.t_document OWNER TO teedy;CREATE TABLE public.t_document_metadata (
    dme_id_c character varying(36) NOT NULL,
    dme_iddocument_c character varying(36) NOT NULL,
    dme_idmetadata_c character varying(36) NOT NULL,

```

```

    dme_value_c character varying(4000)
);
ALTER TABLE public.t_document_metadata OWNER TO teedy;CREATE TABLE public.t_document_tag (
    dot_id_c character varying(36) NOT NULL,
    dot_iddocument_c character varying(36) NOT NULL,
    dot_idtag_c character varying(36) NOT NULL,
    dot_deletedate_d timestamp without time zone
);
ALTER TABLE public.t_document_tag OWNER TO teedy;CREATE TABLE public.t_file (
    fil_id_c character varying(36) NOT NULL,
    fil_iddoc_c character varying(36),
    fil_iduser_c character varying(36) NOT NULL,
    fil_mimetype_c character varying(100) NOT NULL,
    fil_createdate_d timestamp without time zone,
    fil_deletedate_d timestamp without time zone,
    fil_order_n integer,
    fil_content_c text,
    fil_name_c character varying(200),
    fil_version_n integer DEFAULT 0 NOT NULL,
    fil_latestversion_b boolean DEFAULT true NOT NULL,
    fil_idversion_c character varying(36)
);
ALTER TABLE public.t_file OWNER TO teedy;CREATE TABLE public.t_group (
    grp_id_c character varying(36) NOT NULL,
    grp_idparent_c character varying(36),
    grp_name_c character varying(50) NOT NULL,
    grp_idrole_c character varying(36),
    grp_deletedate_d timestamp without time zone
);
ALTER TABLE public.t_group OWNER TO teedy;CREATE TABLE public.t_metadata (
    met_id_c character varying(36) NOT NULL,
    met_name_c character varying(50) NOT NULL,
    met_type_c character varying(20) NOT NULL,
    met_deletedate_d timestamp without time zone
);
ALTER TABLE public.t_metadata OWNER TO teedy;CREATE TABLE public.t_password_recovery (
    pwr_id_c character varying(36) NOT NULL,
    pwr_username_c character varying(50) NOT NULL,
    pwr_createdate_d timestamp without time zone,
    pwr_deletedate_d timestamp without time zone

```

```

);
ALTER TABLE public.t_password_recovery OWNER TO teedy;CREATE TABLE public.t_relation (
    rel_id_c character varying(36) NOT NULL,
    rel_iddocfrom_c character varying(36) NOT NULL,
    rel_iddocto_c character varying(36) NOT NULL,
    rel_deletedate_d timestamp without time zone
);
ALTER TABLE public.t_relation OWNER TO teedy;CREATE TABLE public.t_role (
    rol_id_c character varying(36) NOT NULL,
    rol_name_c character varying(36) NOT NULL,
    rol_createdate_d timestamp without time zone NOT NULL,
    rol_deletedate_d timestamp without time zone
);
ALTER TABLE public.t_role OWNER TO teedy;CREATE TABLE public.t_role_base_function (
    rbf_id_c character varying(36) NOT NULL,
    rbf_idrole_c character varying(36) NOT NULL,
    rbf_idbasefunction_c character varying(20) NOT NULL,
    rbf_createdate_d timestamp without time zone NOT NULL,
    rbf_deletedate_d timestamp without time zone
);
ALTER TABLE public.t_role_base_function OWNER TO teedy;CREATE TABLE public.t_route (
    rte_id_c character varying(36) NOT NULL,
    rte_iddocument_c character varying(36) NOT NULL,
    rte_name_c character varying(50) NOT NULL,
    rte_createdate_d timestamp without time zone NOT NULL,
    rte_deletedate_d timestamp without time zone
);
ALTER TABLE public.t_route OWNER TO teedy;CREATE TABLE public.t_route_model (
    rtm_id_c character varying(36) NOT NULL,
    rtm_name_c character varying(50) NOT NULL,
    rtm_steps_c character varying(5000) NOT NULL,
    rtm_createdate_d timestamp without time zone NOT NULL,
    rtm_deletedate_d timestamp without time zone
);
ALTER TABLE public.t_route_model OWNER TO teedy;CREATE TABLE public.t_route_step (
    rtp_id_c character varying(36) NOT NULL,
    rtp_idroute_c character varying(36) NOT NULL,
    rtp_name_c character varying(200) NOT NULL,
    rtp_type_c character varying(50) NOT NULL,
    rtp_transition_c character varying(50),

```

```

    rtp_comment_c character varying(500),
    rtp_idtarget_c character varying(36) NOT NULL,
    rtp_idvalidatoruser_c character varying(36),
    rtp_order_n integer NOT NULL,
    rtp_createdate_d timestamp without time zone NOT NULL,
    rtp_enddate_d timestamp without time zone,
    rtp_deletedate_d timestamp without time zone,
    rtp_transitions_c character varying(2000)
);
ALTER TABLE public.t_route_step OWNER TO teedy;CREATE TABLE public.t_share (
    sha_id_c character varying(36) NOT NULL,
    sha_name_c character varying(36),
    sha_createdate_d timestamp without time zone,
    sha_deletedate_d timestamp without time zone
);
ALTER TABLE public.t_share OWNER TO teedy;CREATE TABLE public.t_tag (
    tag_id_c character varying(36) NOT NULL,
    tag_iduser_c character varying(36) NOT NULL,
    tag_name_c character varying(36) NOT NULL,
    tag_createdate_d timestamp without time zone,
    tag_deletedate_d timestamp without time zone,
    tag_color_c character varying(7) DEFAULT '#3a87ad'::character varying NOT NULL,
    tag_idparent_c character varying(36)
);
ALTER TABLE public.t_tag OWNER TO teedy;CREATE TABLE public.t_user (
    use_id_c character varying(36) NOT NULL,
    use_idrole_c character varying(36) NOT NULL,
    use_username_c character varying(50) NOT NULL,
    use_password_c character varying(60) NOT NULL,
    use_email_c character varying(100) NOT NULL,
    use_createdate_d timestamp without time zone NOT NULL,
    use_deletedate_d timestamp without time zone,
    use_privatekey_c character varying(100) DEFAULT ''::character varying NOT NULL,
    use_storagequota_n bigint DEFAULT '10000000000'::bigint NOT NULL,
    use_storagecurrent_n bigint DEFAULT 0 NOT NULL,
    use_totpkey_c character varying(100),
    use_disabledate_d timestamp without time zone,
    use_onboarding_b boolean DEFAULT true NOT NULL
);
ALTER TABLE public.t_user OWNER TO teedy;CREATE TABLE public.t_user_group (

```



```

    ugp_id_c character varying(36) NOT NULL,
    ugp_iduser_c character varying(36) NOT NULL,
    ugp_idgroup_c character varying(36) NOT NULL,
    ugp_deletedate_d timestamp without time zone
);
ALTER TABLE public.t_user_group OWNER TO teedy;CREATE TABLE public.t_vocabulary (
    voc_id_c character varying(36) NOT NULL,
    voc_name_c character varying(50) NOT NULL,
    voc_value_c character varying(500) NOT NULL,
    voc_order_n integer NOT NULL
);
ALTER TABLE public.t_vocabulary OWNER TO teedy;CREATE TABLE public.t_webhook (
    whk_id_c character varying(36) NOT NULL,
    whk_event_c character varying(50) NOT NULL,
    whk_url_c character varying(1024) NOT NULL,
    whk_createdate_d timestamp without time zone NOT NULL,
    whk_deletedate_d timestamp without time zone
);

```

```
psql -d teedy_db -U teedy -f step1.sql -L teedy_db_dump_step1.log
```

## Export data from H2 to CSV

Access the H2 database and run the following call statements. See this tutorial on how to open your existing H2 database → [Access / view H2 Database](#)

```

/*get an overview of all tables*/
SHOW TABLES;

```

The following call statements were built up from the "SHOW TABLES;" output above.

```

call CSVWRITE ('C:/users/mario/Downloads/h2data/T_ACL.csv', 'SELECT * FROM T_ACL',
'charset=utf8');
call CSVWRITE ('C:/users/mario/Downloads/h2data/T_AUDIT_LOG.csv', 'SELECT * FROM T_AUDIT_LOG',
'charset=utf8');
call CSVWRITE ('C:/users/mario/Downloads/h2data/T_AUTHENTICATION_TOKEN.csv', 'SELECT * FROM
T_AUTHENTICATION_TOKEN', 'charset=utf8');
call CSVWRITE ('C:/users/mario/Downloads/h2data/T_BASE_FUNCTION.csv', 'SELECT * FROM
T_BASE_FUNCTION', 'charset=utf8');
call CSVWRITE ('C:/users/mario/Downloads/h2data/T_COMMENT.csv', 'SELECT * FROM T_COMMENT',

```

```
'charset=utf8');
call CSVWRITE ('C:/users/mario/Downloads/h2data/T_CONFIG.csv', 'SELECT * FROM T_CONFIG',
'charset=utf8');
call CSVWRITE ('C:/users/mario/Downloads/h2data/T_CONTRIBUTOR.csv', 'SELECT * FROM
T_CONTRIBUTOR', 'charset=utf8');
call CSVWRITE ('C:/users/mario/Downloads/h2data/T_DOCUMENT.csv', 'SELECT * FROM T_DOCUMENT',
'charset=utf8');
call CSVWRITE ('C:/users/mario/Downloads/h2data/T_DOCUMENT_METADATA.csv', 'SELECT * FROM
T_DOCUMENT_METADATA', 'charset=utf8');
call CSVWRITE ('C:/users/mario/Downloads/h2data/T_DOCUMENT_TAG.csv', 'SELECT * FROM
T_DOCUMENT_TAG', 'charset=utf8');
/*the column "fil_content_c" is ignored because it contains a lot of weird plain text which is
almost impossible to export to CSV in a clean fashion*/
call CSVWRITE ('C:/users/mario/Downloads/h2data/T_FILE.csv', 'SELECT FIL_ID_C, FIL_IDDOC_C,
FIL_IDUSER_C, FIL_MIMETYPE_C, FIL_CREATEDATE_D, FIL_DELETEDATE_D, FIL_ORDER_N, NULL AS
FIL_CONTENT_C, FIL_NAME_C, FIL_VERSION_N, FIL_LATESTVERSION_B, FIL_IDVERSION_C FROM T_FILE',
'charset=utf8');
call CSVWRITE ('C:/users/mario/Downloads/h2data/T_GROUP.csv', 'SELECT * FROM T_GROUP',
'charset=utf8');
call CSVWRITE ('C:/users/mario/Downloads/h2data/T_METADATA.csv', 'SELECT * FROM T_METADATA',
'charset=utf8');
call CSVWRITE ('C:/users/mario/Downloads/h2data/T_PASSWORD_RECOVERY.csv', 'SELECT * FROM
T_PASSWORD_RECOVERY', 'charset=utf8');
call CSVWRITE ('C:/users/mario/Downloads/h2data/T_RELATION.csv', 'SELECT * FROM T_RELATION',
'charset=utf8');
call CSVWRITE ('C:/users/mario/Downloads/h2data/T_ROLE.csv', 'SELECT * FROM T_ROLE',
'charset=utf8');
call CSVWRITE ('C:/users/mario/Downloads/h2data/T_ROLE_BASE_FUNCTION.csv', 'SELECT * FROM
T_ROLE_BASE_FUNCTION', 'charset=utf8');
call CSVWRITE ('C:/users/mario/Downloads/h2data/T_ROUTE.csv', 'SELECT * FROM T_ROUTE',
'charset=utf8');
call CSVWRITE ('C:/users/mario/Downloads/h2data/T_ROUTE_MODEL.csv', 'SELECT * FROM
T_ROUTE_MODEL', 'charset=utf8');
call CSVWRITE ('C:/users/mario/Downloads/h2data/T_ROUTE_STEP.csv', 'SELECT * FROM
T_ROUTE_STEP', 'charset=utf8');
call CSVWRITE ('C:/users/mario/Downloads/h2data/T_SHARE.csv', 'SELECT * FROM T_SHARE',
'charset=utf8');
call CSVWRITE ('C:/users/mario/Downloads/h2data/T_TAG.csv', 'SELECT * FROM T_TAG',
'charset=utf8');
call CSVWRITE ('C:/users/mario/Downloads/h2data/T_USER.csv', 'SELECT * FROM T_USER',
```

```
'charset=utf8');  
call CSVWRITE ('C:/users/mario/Downloads/h2data/T_USER_GROUP.csv', 'SELECT * FROM  
T_USER_GROUP', 'charset=utf8');  
call CSVWRITE ('C:/users/mario/Downloads/h2data/T_VOCABULARY.csv', 'SELECT * FROM  
T_VOCABULARY', 'charset=utf8');  
call CSVWRITE ('C:/users/mario/Downloads/h2data/T_WEBHOOK.csv', 'SELECT * FROM T_WEBHOOK',  
'charset=utf8');
```

## Upload csv files to server (by SSH/SFTP)

```
mkdir ~/h2data  
#put csv files here
```

## Move files to postgres directory (to grant access)

```
#as root  
mv ~/h2data /var/lib/postgresql/  
cd /var/lib/postgresql/  
chown -R postgres:postgres h2data/
```

## SQL Import Copying

The folloing copy statements were built up bases on the "call CSVWRITE" statements in the H2 export.

```
psql  
\c teedy_db;  
#You are now connected to database "teedy_db" as user "postgres".  
COPY T_ACL FROM '/var/lib/postgresql/h2data/T_ACL.csv' DELIMITER ',' CSV HEADER ENCODING 'utf-8';  
COPY T_CONFIG FROM '/var/lib/postgresql/h2data/T_CONFIG.csv' DELIMITER ',' CSV HEADER ENCODING 'utf-8';  
COPY T_GROUP FROM '/var/lib/postgresql/h2data/T_GROUP.csv' DELIMITER ',' CSV HEADER ENCODING 'utf-8';  
COPY T_ROLE FROM '/var/lib/postgresql/h2data/T_ROLE.csv' DELIMITER ',' CSV HEADER ENCODING 'utf-8';  
COPY T_USER FROM '/var/lib/postgresql/h2data/T_USER.csv' DELIMITER ',' CSV HEADER ENCODING 'utf-8';  
COPY T_CONTRIBUTOR FROM '/var/lib/postgresql/h2data/T_CONTRIBUTOR.csv' DELIMITER ',' CSV  
HEADER ENCODING 'utf-8';
```

```
COPY T_METADATA FROM '/var/lib/postgresql/h2data/T_METADATA.csv' DELIMITER ',' CSV HEADER
ENCODING 'utf-8';
COPY T_WEBHOOK FROM '/var/lib/postgresql/h2data/T_WEBHOOK.csv' DELIMITER ',' CSV HEADER
ENCODING 'utf-8';
COPY T_VOCABULARY FROM '/var/lib/postgresql/h2data/T_VOCABULARY.csv' DELIMITER ',' CSV HEADER
ENCODING 'utf-8';
COPY T_BASE_FUNCTION FROM '/var/lib/postgresql/h2data/T_BASE_FUNCTION.csv' DELIMITER ',' CSV
HEADER ENCODING 'utf-8';
COPY T_AUDIT_LOG FROM '/var/lib/postgresql/h2data/T_AUDIT_LOG.csv' DELIMITER ',' CSV HEADER
ENCODING 'utf-8';
COPY T_RELATION FROM '/var/lib/postgresql/h2data/T_RELATION.csv' DELIMITER ',' CSV HEADER
ENCODING 'utf-8';
COPY T_USER_GROUP FROM '/var/lib/postgresql/h2data/T_USER_GROUP.csv' DELIMITER ',' CSV HEADER
ENCODING 'utf-8';
COPY T_ROLE_BASE_FUNCTION FROM '/var/lib/postgresql/h2data/T_ROLE_BASE_FUNCTION.csv' DELIMITER
',' CSV HEADER ENCODING 'utf-8';
COPY T_ROUTE_MODEL FROM '/var/lib/postgresql/h2data/T_ROUTE_MODEL.csv' DELIMITER ',' CSV
HEADER ENCODING 'utf-8';
COPY T_AUTHENTICATION_TOKEN FROM '/var/lib/postgresql/h2data/T_AUTHENTICATION_TOKEN.csv'
DELIMITER ',' CSV HEADER ENCODING 'utf-8';
COPY T_TAG FROM '/var/lib/postgresql/h2data/T_TAG.csv' DELIMITER ',' CSV HEADER ENCODING 'utf-
8';
COPY T_PASSWORD_RECOVERY FROM '/var/lib/postgresql/h2data/T_PASSWORD_RECOVERY.csv' DELIMITER
',' CSV HEADER ENCODING 'utf-8';
COPY T_SHARE FROM '/var/lib/postgresql/h2data/T_SHARE.csv' DELIMITER ',' CSV HEADER ENCODING
'utf-8';
COPY T_COMMENT FROM '/var/lib/postgresql/h2data/T_COMMENT.csv' DELIMITER ',' CSV HEADER
ENCODING 'utf-8';
COPY T_DOCUMENT FROM '/var/lib/postgresql/h2data/T_DOCUMENT.csv' DELIMITER ',' CSV HEADER
ENCODING 'utf-8';
COPY T_DOCUMENT_METADATA FROM '/var/lib/postgresql/h2data/T_DOCUMENT_METADATA.csv' DELIMITER
',' CSV HEADER ENCODING 'utf-8';
COPY T_DOCUMENT_TAG FROM '/var/lib/postgresql/h2data/T_DOCUMENT_TAG.csv' DELIMITER ',' CSV
HEADER ENCODING 'utf-8';
COPY T_FILE FROM '/var/lib/postgresql/h2data/T_FILE.csv' DELIMITER ',' CSV HEADER ENCODING
'utf-8';
COPY T_ROUTE FROM '/var/lib/postgresql/h2data/T_ROUTE.csv' DELIMITER ',' CSV HEADER ENCODING
'utf-8';
COPY T_ROUTE_STEP FROM '/var/lib/postgresql/h2data/T_ROUTE_STEP.csv' DELIMITER ',' CSV HEADER
ENCODING 'utf-8';
```

## SQL Step 2 - Adjust foreign keys and indexes

```
/*step2.sql*/  
  
ALTER TABLE public.t_webhook OWNER TO teedy; ALTER TABLE ONLY public.t_acl  
    ADD CONSTRAINT t_acl_pkey PRIMARY KEY (acl_id_c);  
  
ALTER TABLE ONLY public.t_audit_log  
    ADD CONSTRAINT t_audit_log_pkey PRIMARY KEY (log_id_c);  
  
ALTER TABLE ONLY public.t_authentication_token  
    ADD CONSTRAINT t_authentication_token_pkey PRIMARY KEY (aut_id_c);  
  
ALTER TABLE ONLY public.t_base_function  
    ADD CONSTRAINT t_base_function_pkey PRIMARY KEY (baf_id_c);  
  
ALTER TABLE ONLY public.t_comment  
    ADD CONSTRAINT t_comment_pkey PRIMARY KEY (com_id_c);  
  
ALTER TABLE ONLY public.t_config  
    ADD CONSTRAINT t_config_pkey PRIMARY KEY (cfg_id_c);  
  
ALTER TABLE ONLY public.t_contributor  
    ADD CONSTRAINT t_contributor_pkey PRIMARY KEY (ctr_id_c);  
  
ALTER TABLE ONLY public.t_document_metadata  
    ADD CONSTRAINT t_document_metadata_pkey PRIMARY KEY (dme_id_c);  
  
ALTER TABLE ONLY public.t_document  
    ADD CONSTRAINT t_document_pkey PRIMARY KEY (doc_id_c);  
  
ALTER TABLE ONLY public.t_document_tag  
    ADD CONSTRAINT t_document_tag_pkey PRIMARY KEY (dot_id_c);  
  
ALTER TABLE ONLY public.t_file  
    ADD CONSTRAINT t_file_pkey PRIMARY KEY (fil_id_c);  
  
ALTER TABLE ONLY public.t_group  
    ADD CONSTRAINT t_group_pkey PRIMARY KEY (grp_id_c);  
  
ALTER TABLE ONLY public.t_metadata  
    ADD CONSTRAINT t_metadata_pkey PRIMARY KEY (met_id_c);  
  
ALTER TABLE ONLY public.t_password_recovery  
    ADD CONSTRAINT t_password_recovery_pkey PRIMARY KEY (pwr_id_c);  
  
ALTER TABLE ONLY public.t_relation  
    ADD CONSTRAINT t_relation_pkey PRIMARY KEY (rel_id_c);  
  
ALTER TABLE ONLY public.t_role_base_function  
    ADD CONSTRAINT t_role_base_function_pkey PRIMARY KEY (rbf_id_c);  
  
ALTER TABLE ONLY public.t_role  
    ADD CONSTRAINT t_role_pkey PRIMARY KEY (rol_id_c);  
  
ALTER TABLE ONLY public.t_route_model
```

```

    ADD CONSTRAINT t_route_model_pkey PRIMARY KEY (rtm_id_c);
ALTER TABLE ONLY public.t_route
    ADD CONSTRAINT t_route_pkey PRIMARY KEY (rte_id_c);
ALTER TABLE ONLY public.t_route_step
    ADD CONSTRAINT t_route_step_pkey PRIMARY KEY (rtp_id_c);
ALTER TABLE ONLY public.t_share
    ADD CONSTRAINT t_share_pkey PRIMARY KEY (sha_id_c);
ALTER TABLE ONLY public.t_tag
    ADD CONSTRAINT t_tag_pkey PRIMARY KEY (tag_id_c);
ALTER TABLE ONLY public.t_user_group
    ADD CONSTRAINT t_user_group_pkey PRIMARY KEY (ugp_id_c);
ALTER TABLE ONLY public.t_user
    ADD CONSTRAINT t_user_pkey PRIMARY KEY (use_id_c);
ALTER TABLE ONLY public.t_vocabulary
    ADD CONSTRAINT t_vocabulary_pkey PRIMARY KEY (voc_id_c);
ALTER TABLE ONLY public.t_webhook
    ADD CONSTRAINT t_webhook_pkey PRIMARY KEY (whk_id_c);
CREATE INDEX idx_acl_sourceid_c ON public.t_acl USING btree (acl_sourceid_c);
CREATE INDEX idx_acl_targetid_c ON public.t_acl USING btree (acl_targetid_c);
CREATE INDEX idx_doc_createdate_d ON public.t_document USING btree (doc_createdate_d);
CREATE INDEX idx_doc_language_c ON public.t_document USING btree (doc_language_c);
CREATE INDEX idx_doc_title_c ON public.t_document USING btree (doc_title_c);
CREATE INDEX idx_dot_composite ON public.t_document_tag USING btree (dot_iddocument_c,
dot_idtag_c, dot_deletedate_d);
CREATE INDEX idx_log_identity_c ON public.t_audit_log USING btree (log_identity_c);
ALTER TABLE ONLY public.t_authentication_token
    ADD CONSTRAINT fk_aut_iduser_c FOREIGN KEY (aut_iduser_c) REFERENCES
public.t_user(use_id_c) ON UPDATE RESTRICT ON DELETE RESTRICT;
ALTER TABLE ONLY public.t_comment
    ADD CONSTRAINT fk_com_iddoc_c FOREIGN KEY (com_iddoc_c) REFERENCES
public.t_document(doc_id_c) ON UPDATE RESTRICT ON DELETE RESTRICT;
ALTER TABLE ONLY public.t_comment
    ADD CONSTRAINT fk_com_iduser_c FOREIGN KEY (com_iduser_c) REFERENCES
public.t_user(use_id_c) ON UPDATE RESTRICT ON DELETE RESTRICT;
ALTER TABLE ONLY public.t_document_metadata
    ADD CONSTRAINT fk_dme_iddocument_c FOREIGN KEY (dme_iddocument_c) REFERENCES
public.t_document(doc_id_c) ON UPDATE RESTRICT ON DELETE RESTRICT;
ALTER TABLE ONLY public.t_document_metadata
    ADD CONSTRAINT fk_dme_idmetadata_c FOREIGN KEY (dme_idmetadata_c) REFERENCES
public.t_metadata(met_id_c) ON UPDATE RESTRICT ON DELETE RESTRICT;

```

```

ALTER TABLE ONLY public.t_document
    ADD CONSTRAINT fk_doc_idfile_c FOREIGN KEY (doc_idfile_c) REFERENCES
public.t_file(fil_id_c) ON UPDATE RESTRICT ON DELETE RESTRICT;
ALTER TABLE ONLY public.t_document
    ADD CONSTRAINT fk_doc_iduser_c FOREIGN KEY (doc_iduser_c) REFERENCES
public.t_user(use_id_c) ON UPDATE RESTRICT ON DELETE RESTRICT;
ALTER TABLE ONLY public.t_document_tag
    ADD CONSTRAINT fk_dot_iddocument_c FOREIGN KEY (dot_iddocument_c) REFERENCES
public.t_document(doc_id_c) ON UPDATE RESTRICT ON DELETE RESTRICT;
ALTER TABLE ONLY public.t_document_tag
    ADD CONSTRAINT fk_dot_idtag_c FOREIGN KEY (dot_idtag_c) REFERENCES public.t_tag(tag_id_c)
ON UPDATE RESTRICT ON DELETE RESTRICT;
ALTER TABLE ONLY public.t_file
    ADD CONSTRAINT fk_fil_iddoc_c FOREIGN KEY (fil_iddoc_c) REFERENCES
public.t_document(doc_id_c) ON UPDATE RESTRICT ON DELETE RESTRICT;
ALTER TABLE ONLY public.t_file
    ADD CONSTRAINT fk_fil_iduser_c FOREIGN KEY (fil_iduser_c) REFERENCES
public.t_user(use_id_c) ON UPDATE RESTRICT ON DELETE RESTRICT;
ALTER TABLE ONLY public.t_role_base_function
    ADD CONSTRAINT fk_rbf_idbasefunction_c FOREIGN KEY (rbf_idbasefunction_c) REFERENCES
public.t_base_function(baf_id_c) ON UPDATE RESTRICT ON DELETE RESTRICT;
ALTER TABLE ONLY public.t_role_base_function
    ADD CONSTRAINT fk_rbf_idrole_c FOREIGN KEY (rbf_idrole_c) REFERENCES
public.t_role(rol_id_c) ON UPDATE RESTRICT ON DELETE RESTRICT;
ALTER TABLE ONLY public.t_route
    ADD CONSTRAINT fk_rte_iddocument_c FOREIGN KEY (rte_iddocument_c) REFERENCES
public.t_document(doc_id_c) ON UPDATE RESTRICT ON DELETE RESTRICT;
ALTER TABLE ONLY public.t_route_step
    ADD CONSTRAINT fk_rtp_idroute_c FOREIGN KEY (rtp_idroute_c) REFERENCES
public.t_route(rte_id_c) ON UPDATE RESTRICT ON DELETE RESTRICT;
ALTER TABLE ONLY public.t_route_step
    ADD CONSTRAINT fk_rtp_idvalidatoruser_c FOREIGN KEY (rtp_idvalidatoruser_c) REFERENCES
public.t_user(use_id_c) ON UPDATE RESTRICT ON DELETE RESTRICT;
ALTER TABLE ONLY public.t_tag
    ADD CONSTRAINT fk_tag_iduser_c FOREIGN KEY (tag_iduser_c) REFERENCES
public.t_user(use_id_c) ON UPDATE RESTRICT ON DELETE RESTRICT;
ALTER TABLE ONLY public.t_user
    ADD CONSTRAINT fk_use_idrole_c FOREIGN KEY (use_idrole_c) REFERENCES
public.t_role(rol_id_c) ON UPDATE RESTRICT ON DELETE RESTRICT;
REVOKE ALL ON SCHEMA public FROM PUBLIC;

```

```
REVOKE ALL ON SCHEMA public FROM postgres;  
GRANT ALL ON SCHEMA public TO postgres;  
GRANT ALL ON SCHEMA public TO PUBLIC;
```

```
psql -d teedy_db -U teedy -f step2.sql -L teedy_db_dump_step2.log
```

## Configure Jetty Service to use database connection instead of H2 local DB

See [Teedy with PostgreSQL](#) on how to do this

## Restart Jetty

```
service jetty9 restart && journalctl -f -u jetty9.service
```

## Reindex all files with Tesseract OCR scanning libraries (the OCR data from is H2 is lost)

There is no way to click some button for this. It has to be scripted. The provided API deals with it. Please see this page for re-processing files by API → [API Scripts / database queries](#)

Before executing this please move (or remove) the \*\_thumb and \*\_web files away because they will be re-processed too. If you don't do this the following will occur: [Fix Preview Bug](#)



# Apache Reverse Proxy and Firewall

## Install apache2

```
sudo apt install apache2
```

## Activate modules

```
sudo a2enmod headers rewrite proxy proxy_html proxy_http ssl vhost_alias
```

## Apache Reverse Proxy Configuration

```
sudo vim /etc/apache2/sites-available/dms.yourdomain.de_httpd.conf
```

```
<VirtualHost YOURPUBLICIP:7080 127.0.0.1:7080>
    ServerName dms.yourdomain.de
    RewriteEngine On
    RewriteCond %{HTTPS} off
    RewriteRule (.*?) https://%{HTTP_HOST}%{REQUEST_URI} [R=301,L]
</VirtualHost>
<VirtualHost YOURPUBLICIP:7081 127.0.0.1:7081>
    ServerName dms.YOURDOMAIN.de
    ServerAdmin info@YOURDOMAIN.de

    ErrorLog ${APACHE_LOG_DIR}/error-sismics.log
    CustomLog ${APACHE_LOG_DIR}/access-sismics.log combined

    SSLEngine on
    SSLCertificateFile /etc/letsencrypt/live/YOURDOMAIN.de/cert.pem
    SSLCertificateKeyFile /etc/letsencrypt/live/YOURDOMAIN.de/privkey.pem
    SSLCertificateChainFile /etc/letsencrypt/live/YOURDOMAIN.de/chain.pem

    SSLCipherSuite EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH
    SSLProtocol All -SSLv2 -SSLv3 -TLSv1 -TLSv1.1
    SSLHonorCipherOrder On
    Header always set Strict-Transport-Security "max-age=63072000; includeSubDomains;
preload"
    #Header always set X-Frame-Options DENY
```

```
Header always set X-Content-Type-Options nosniff
#Header set Content-Security-Policy "default-src 'none'; script-src 'self'; connect-
src 'self'; img-src 'self'; style-src 'self';"
Header unset X-Powered-By
Header set Referrer-Policy "origin-when-cross-origin"
Header always edit Set-Cookie (.*) "$1; HttpOnly; Secure"
Header set X-XSS-Protection "1; mode=block"
Header always set Content-Security-Policy "upgrade-insecure-requests;" #upgrade unsafe
gravatar icons to load from https instead of http
```

```
# Requires Apache >= 2.4
SSLCompression off
#SSLUseStapling on
#SSLStaplingCache "shmcb:logs/stapling-cache(150000)"
# Requires Apache >= 2.4.11
SSLSessionTickets Off
```

```
ProxyRequests Off
```

```
# Auth changes in 2.4 - see http://httpd.apache.org/docs/2.4/upgrading.html#run-time
<Proxy *>
    Require all granted
</Proxy>
```

```
ProxyPass / http://localhost:8080/dms/
ProxyPassReverse / http://localhost:8080/dms/
<Location />
    SSLRenegBufferSize 100000000
    Require all granted
</Location>
```

```
<Location "/api/app">
    AllowOverride None
    Order deny,allow
    Deny from All
</Location>
```

```
<Location ~ "/api/app/.*">
    AllowOverride None
    Allow from All
```

```
</Location>
```

```
RewriteEngine on
```

```
RewriteCond %{REQUEST_FILENAME} !-d
```

```
RewriteRule ^(.*)/$ /$1 [R=301,L]
```

```
</VirtualHost>
```

## Firewall Blocking Rule

Block direct access to Jetty9 on Port 8080 (ingoing and outgoing TCP traffic) to allow access only on SSL secured domain. Use iptables or similar.

# Teedy with PostgreSQL

Teedy 1 was tested successfully with PostgreSQL Version 16 and lower

## Install and configure PostgreSQL

Teedy requires at least PSQL 9.4 (PostgreSQL94Dialect)

PostgreSQL 10 and upwards are configured to deliver SSL by standard! You will need to configure it's SSL cert!

```
sudo apt install -y postgresql postgresql-client libpq-dev postgresql-contrib
```

```
sudo vim /etc/postgresql/<VERSION>/main/pg_hba.conf
```

```
#add local to trust to omit password input. If you change to md5 you will need to enter
passwords if you run scripts (e.g. bash)
# "local" is for Unix domain socket connections only
local        all        all        trust
#host        all        all        0.0.0.0/0 md5
hostssl      all        all        0.0.0.0/0 md5
```

```
sudo vim /etc/postgresql/<VERSION>/main/postgresql.conf
```

```
listen_addresses = '*'           # what IP address(es) to listen on;
ssl = on
ssl_cert_file = '/etc/ssl/yourdomain.de.pem'
ssl_key_file = '/etc/ssl/private/yourdomain.de.key'
```

```
#login as postgres user
su - postgres
psql

CREATE USER teedy WITH PASSWORD 'password';
CREATE DATABASE teedy_db WITH ENCODING 'UNICODE' LC_COLLATE 'C' LC_CTYPE 'C' TEMPLATE
template0;
```

```
GRANT ALL PRIVILEGES ON DATABASE teedy_db TO teedy ;

#remove old database if required
#service postgresql restart #kick old connections
#REVOKE ALL PRIVILEGES ON DATABASE teedy_db FROM teedy;
#DROP DATABASE teedy_db;
#DROP USER teedy;
```

## PostgreSQL SSL

```
cd /etc/letsencrypt/live/yourdomain.de/
cp privkey.pem /etc/ssl/private/yourdomain.de.key

(cat privkey.pem; printf "\n\n"; cat cert.pem; printf "\n\n"; cat chain.pem; printf "\n\n") >>
/etc/ssl/yourdomain.de.pem
cd /etc/ssl/

chgrp ssl-cert /etc/ssl/private/yourdomain.de.key
chmod 640 /etc/ssl/private/yourdomain.de.key
chgrp ssl-cert /etc/ssl/yourdomain.de.pem
chmod 640 /etc/ssl/yourdomain.de.pem

less /var/log/postgresql/postgresql-9.5-main.log #check for occuring errors belonging to SSL
cert
```

## Configure dms.xml (optional)

```
vim /opt/jetty-home-11.0.15/jetty-base/webapps/dms.xml
```

```
<?xml version="1.0"?>
<!DOCTYPE Configure PUBLIC "-//Jetty//Configure//EN"
"http://www.eclipse.org/jetty/configure_10_0.dtd">
<Configure class="org.eclipse.jetty.webapp.WebAppContext">
  <Set name="contextPath">/dms</Set>
  <Set name="war"><SystemProperty name="jetty.data" default="."/>/webapps/dms.war</Set>
  <Call class="java.lang.System" name="setProperty">
    <Arg>docs.home</Arg>
    <Arg>/var/docs</Arg>
  </Call>
```

## Configuration for usage of PostgreSQL instead H2

Note: The database connection is set via a central environment variable configuration for the entire Jetty service and cannot be set for individual WebApplicationContext.

Have a look at [Environment Configuration](#) on how to swap to PostgreSQL

## External connection test with Oracle SQL Developer

- enable ports and configure firewall correctly
- download SQL Developer driver for PostgreSQL → <https://jdbc.postgresql.org/download.html#current> (Treiber = postgresql-42.2.6)
- add the driver in system settings
- User and database differ from the name. Therefore, the entry must be entered as follows:

The screenshot shows the 'New' connection dialog in Oracle SQL Developer. The 'Name' field is set to 'dms.yourdomain.de'. The 'Datenbanktyp' (Database type) is set to 'PostgreSQL'. The 'Benutzername' (Username) is 'teedy'. The 'Kennwort' (Password) field is masked with dots. The 'Kennwort speichern' (Save password) checkbox is checked. The 'Hostname' field contains 'nain.de:5432/teedy\_db?'. The 'Port' field is empty. At the bottom, there is a 'Datenbank wählen' (Select database) button and a dropdown menu showing 'te'.

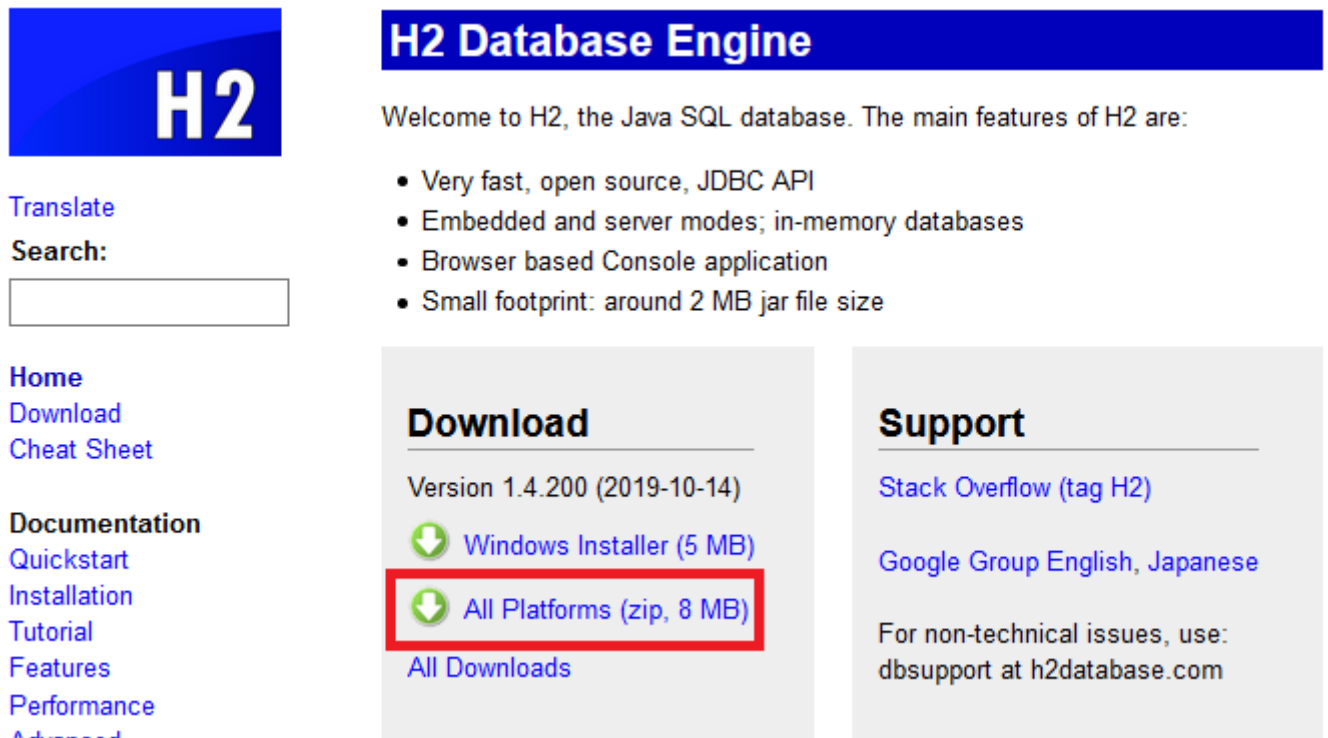
- user: teedy
- hostname: yourdomain.de:5432/teedy\_db?
- port: stays empty
- database: teedy\_db

# Access to the H2 Database

External Database access for H2 can be granted on different ways. Teedy stores its H2 database in standard path `/var/docs/db`

## Variant 1: Download H2 Database Tool

1. Download → <http://www.h2database.com/h2-2019-03-13.zip>



**H2 Database Engine**

Welcome to H2, the Java SQL database. The main features of H2 are:

- Very fast, open source, JDBC API
- Embedded and server modes; in-memory databases
- Browser based Console application
- Small footprint: around 2 MB jar file size

**Download**

Version 1.4.200 (2019-10-14)

- Windows Installer (5 MB)
- All Platforms (zip, 8 MB)**

[All Downloads](#)

**Support**

[Stack Overflow \(tag H2\)](#)

[Google Group English, Japanese](#)

For non-technical issues, use: [dbsupport at h2database.com](mailto:dbsupport@h2database.com)

2. Put your database **copy/backup** files (docs.mv.db and dovs.trace.db) into some directory where you can access it
3. Use it

**Run the h2.bat file and connect via Web Interface, or**

Login

Gespeicherte Einstellung: Generic H2 (Embedded) ▼

Einstellungs-Name: Generic H2 (Embedded) Speichern Entfernen

---

Datenbank-Treiber Klasse: org.h2.Driver

JDBC URL: jdbc:h2:~/Downloads/docs

Benutzername: sa

Passwort:

Verbinden Verbindung testen

## Connect via Console (Windows)

```
cmd
cd C:\Users\mario\Downloads\h2-2019-03-13\h2\bin
#note: ignore the *.db ending. H2 will add it automatically. If you do it yourself it will fail!
java -jar h2-1.4.199.jar -url
"jdbc:h2:file:~/Downloads/docs;CACHE_SIZE=65536;LOCK_TIMEOUT=10000;IFEXISTS=TRUE;" -driver
"org.h2.Driver" -user "sa" -password ""
```

## Connect via Console (Linux)

Good tips: <https://o7planning.org/en/11895/installing-h2-database-and-using-h2-console>

```
cd opt/
wget https://h2database.com/h2-2019-03-13.zip
unzip h2-2019-03-13.zip
rm h2-2019-03-13.zip
cd /opt/h2/bin/

#open Database with h2 driver - enable X11 forward to recieve graphical user interface (GUI)
export DISPLAY=localhost:10.0 && java -jar h2-1.4.199.jar -url
```



```
"jdbc:h2:file:/var/docs.bak/db/docs;CACHE_SIZE=65536;LOCK_TIMEOUT=10000;IFEXISTS=TRUE;" -  
driver "org.h2.Driver" -user "sa" -password ""
```

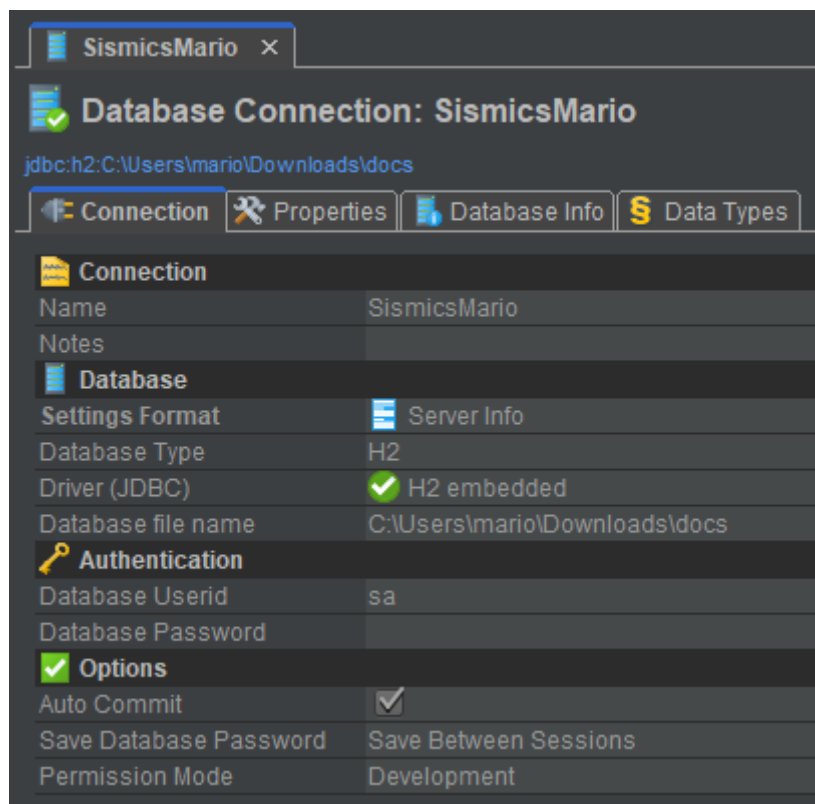
#note that version h2-2019-10-14 failed on the Teedy H2 database. So i used older version h2-2019-03-13

## Variant 2: DBVisualizer

DBVisualizer comes with bundled support for H2 including the driver →

<https://www.dbvis.com/features/h2-database-features>

Some example configuration looks like this:



Ignore the \*.db ending. H2 will add it automatically. If you do it yourself it will fail!  
So the database "docs" contains of two files and you only need to enter the principal name.



docs.mv.db



docs.trace.db

# Basic installation with Jetty and H2 Database

## Install Required Software

```
#install a lot of stuff
sudo apt update
sudo apt install tesseract-ocr tesseract-ocr-deu tesseract-ocr-eng libtesseract-dev ffmpeg
mediainfo mediainfo-gui openjdk-11-jdk

#install Jetty Web Server
sudo apt install jetty11

#check versions
ffmpeg -version
tesseract -v
mediainfo --version
dpkg -l | grep jetty11
dpkg -l | grep jdk
```

You can also install jetty manually (not by `apt`) with ease and full control:

```
cd /opt
wget https://repo1.maven.org/maven2/org/eclipse/jetty/jetty-home/11.0.15/jetty-home-11.0.15.tar.gz
tar -xvzf jetty-home-11.0.15.tar.gz
mkdir -p /opt/jetty-home-11.0.15/jetty-base/
cd /opt/jetty-home-11.0.15/jetty-base/
java -jar ../start.jar --add-modules=deploy,http
cp /opt/teedy/docs-web/target/docs-web-1.*.war /opt/jetty-home-11.0.15/jetty-base/webapps/dms.war #copy the compiled deployment war to target dir
chown jetty:adm /opt/jetty-home-11.0.15/jetty-base/webapps/dms.war
chown jetty:adm /opt/jetty-home-11.0.15/jetty-base/webapps/dms.xml
```

## Create dms.xml configuration files

This allows to change the default docs home dir and other things. Have a look at

<https://github.com/sismics/docs/blob/master/docs.xml>

```
vim /opt/jetty-home-11.0.15/jetty-base/webapps/dms.xml
```

```
<?xml version="1.0"?>
<!DOCTYPE Configure PUBLIC "-//Jetty//Configure//EN"
"http://www.eclipse.org/jetty/configure_10_0.dtd">
<Configure class="org.eclipse.jetty.webapp.WebAppContext">
  <Set name="contextPath">/dms</Set>
  <Set name="war"><SystemProperty name="jetty.data" default="."/>/webapps/dms.war</Set>
  <Call class="java.lang.System" name="setProperty">
    <Arg>docs.home</Arg>
    <Arg>/var/docs</Arg>
  </Call>
</Configure>
```

Adjust the following lines according to your configured XML argument `docs.home` (if changed)

```
mkdir -p /var/docs/
chmod -T 770 /var/docs/
chown -R jetty:jetty /var/docs/
```

Take the pre-built `dms.war` file or [compile on your own](#) and put it to `/opt/jetty-home-11.0.15/jetty-base/webapps/dms.war`

## Tuning

Raise the heap space `Xmx` to prevent "java heap space error" - this often occurs when OCR'ing a lot of files or uploading multiple files at once. This causes to restart jetty9 completely

```
sudo vim /lib/systemd/system/jetty11.service
```

```
Environment="JAVA_OPTS=-Xms1024m -Xmx3584m -Djava.awt.headless=true"
```

See also [Environment Configuration](#) for reference

## JDK/JRE - Permissions Policy Adjustments (optional)

that might be dangerous / unsecure

```
sudo vim /usr/lib/jvm/java-11-openjdk-amd64/lib/security/default.policy
```

Add to top of file:

```
grant {  
    permission java.security.AllPermission "", "";  
};
```

## Fix jetty read-only filesystem (since Jetty 9.4.15)

Sympton → "Caused by: [java.io.FileNotFoundException](#): /var/docs/db/docs.trace.db (file system is readonly - but it is not!)"

The newest jetty package was changed to contain restricted settings in `/lib/systemd/system/jetty9.service`. You need to add another `ReadWritePath` for the `/var/docs` directory

```
sudo vim /lib/systemd/system/jetty11.service
```

```
ProtectSystem=strict  
ReadWritePaths=/var/lib/jetty11  
ReadWritePaths=/var/docs/
```

## Restart Jetty Service

```
sudo systemctl restart jetty11.service
```

## Check the logs

```
sudo less /var/log/jetty11/  
sudo journalctl -f -u jetty11
```

## Access to Teedy Web Interface

<https://YOURHOST:8080>

# Build web application server from Source

This short tutorial shows how to build Teedy. Finally you will get a ready-to-deploy dms.war java archive file.

## Pre-Requisites

```
sudo add-apt-repository ppa:openjdk-r/ppa
sudo apt update
sudo apt install openjdk-11-jdk maven npm jetty11

sudo ln -s /usr/bin/nodejs /usr/bin/node
npm install -g grunt-cli
npm install -g npm@latest
```

## Building

Additional steps are included in the following bash script.

- Adjust code for GDPR (data privacy → add some custom URLs to satisfy german TMG)
- adjusting hibernate pool size to allow more connections
- some CSS fix for bigger preview images

```
#!/bin/bash
BASE="/opt/teedy"

#altes Repo löschen
#rm -rf /opt/teedy/

#clone
git clone https://github.com/sismics/docs.git ${BASE}
cd ${BASE}/

#to update existing repo:
git stash
git pull

#GDPR adjustments
sed -i 's/<li>v{{ app.current_version }}</li>/<li><a
```

```

href=\"https://stadtfabrikanten.org/impressum">Impressum</a></li><li><a
href=\"https://stadtfabrikanten.org/datenschutz">Datenschutz</a></li><li>v{{
app.current_version }}</li>/g' ${BASE}/docs-web/src/main/webapp/src/share.html
sed -i 's/<li>v{{ app.current_version }}</li>/<li><a
href=\"https://stadtfabrikanten.org/impressum">Impressum</a></li><li><a
href=\"https://stadtfabrikanten.org/datenschutz">Datenschutz</a></li><li>v{{
app.current_version }}</li>/g' ${BASE}/docs-web/src/main/webapp/src/index.html

#CSS fix
sed -i 's/42px/100px/g' ${BASE}/docs-web/src/main/webapp/src/style/main.less

#Adjust hibernate.pool_size from 10 to 50
sed -i 's/"hibernate.connection.pool_size", "10"/"hibernate.connection.pool_size",
"50"/g' ${BASE}/docs-core/src/main/java/com/sismics/util/jpa/EMF.java

#Building
cd ${BASE}/
mvn clean -DskipTests install

cd ${BASE}/docs-web
mvn -Pprod -DskipTests clean install

#Apply
JETTY_WEBAPPS_DIR="/opt/jetty-home-11.0.15/jetty-base/webapps"
service jetty11 stop
mv ${JETTY_WEBAPPS_DIR}/dms.war ${JETTY_WEBAPPS_DIR}/dms.war.bak
cp /opt/teedy/docs-web/target/docs-web-1.*-SNAPSHOT.war ${JETTY_WEBAPPS_DIR}/dms.war
chown jetty:adm ${JETTY_WEBAPPS_DIR}/dms.war

systemctl restart jetty11
journalctl -f -u jetty11.service

```

# Backup and restore strategies

## Simple: Make backup of server app and data dir (H2 database)

```
#!/bin/bash
BUP_PATH="/backup/teedy/"
mkdir -p $BUP_PATH
rsync -lrptR /var/docs $BUP_PATH
rsync -lrptR /opt/jetty-home-11.0.15/jetty-base/webapps $BUP_PATH
cd $BUP_PATH
FILENAME=$(date +%Y-%m-%d)-teedy.tar.gz
tar -zcvf $FILENAME $BUP_PATH/var/
chown TARGETUSER:TARGETUSER $FILENAME
```

## Simple: Restore backup data dir (H2 database)

```
systemctl stop jetty11.service
cd /var; rm -rf docs
mkdir -p /var/docs
cp /backup/teedy/2018-12-30-teedy.tar.gz /var/docs
cd /var/docs
tar -xvzf 2018-12-30-teedy.tar.gz
chmod 777 /var/docs
systemctl start jetty11.service
```

## Simple: Backup PostgreSQL (not if you are using H2)

```
sudo -iu postgres bash -c "pg_dump YOURDATABASE > YOURDATABASE.sql" && mv
/var/lib/postgresql/YOURDATABASE.sql "$BUP_POSTGRES"/YOURDATABASE.sql
```

## Simple: Restore PostgreSQL dump

The database you want to restore must exist in your psql instance! If you move your DB from one server to another you need to recreate an empty DB first. See [Teedy with PostgreSQL](#).



```
#move the database dump file to a location where postgres user can read it, for example
/var/lib/postgres/
chown postgres /var/lib/postgresql/teedy_db.sql
su - postgres

#drop old database and create a new one
psql
drop database teedy_inventory_db;
CREATE DATABASE teedy_db WITH ENCODING 'UNICODE' LC_COLLATE 'C' LC_CTYPE 'C' TEMPLATE
template0;
GRANT ALL PRIVILEGES ON DATABASE teedy_inventory_db TO teedy_inventory;

#now import the backup db dump
psql teedy_inventory_db < teedy_inventory_db.sql
```

## API Use: Bash script for API export of all documents and tags with cURL

```
#!/bin/bash
BASE_URL="https://dms.yourdomain.de"
AUTH_TOKEN=$(curl -i -X POST -d username="THEUSERNAME" -d password="THEPASSWORD"
"$BASE_URL/api/user/login" -k|grep "auth_token"|cut -c24-59)
BACKUP_DIR="/backup/teedy"
TARGET_JSON_FILE=$BACKUP_DIR"/documentlist.json"
TARGET_JSON_FILE_PARSED=$BACKUP_DIR"/documentlist.txt"
TARGET_JSON_FILE_SORTED=$BACKUP_DIR"/documentlist.sorted.txt"
DIR_ZIP=$BACKUP_DIR/"ZIP"
DIR_PDF=$BACKUP_DIR/"PDF"

#create backup directory
mkdir -p "$BACKUP_DIR"
mkdir -p "$DIR_ZIP"
mkdir -p "$DIR_PDF"

#get the documents list
#curl --silent -X GET -H "Cookie: auth_token=$AUTH_TOKEN"
"$BASE_URL/api/document/list?limit=0" -k | jq . > "$TARGET_JSON_FILE"
curl --silent -X POST -H "Cookie: auth_token=$AUTH_TOKEN" "$BASE_URL/api/document/list" -d
```

```

"limit=999999" -k | jq . > "$TARGET_JSON_FILE"

#get the tags list
curl --silent -X GET -H "Cookie: auth_token=$AUTH_TOKEN" "$BASE_URL/api/tag/list?limit=0" -k |
jq . > "$BACKUP_DIR"/taglist.json
#read the complete list of documents
jq -c '.{documents}|.[]|.[]|{id}+{title}+{create_date}+{tags}' "$TARGET_JSON_FILE" >
"$TARGET_JSON_FILE_PARSED"

#make sorted list which shows number of duplicates
jq -c '.{documents}|.[]|.[]|{title}' "$TARGET_JSON_FILE" | sort | uniq -c | sort >
"$TARGET_JSON_FILE_SORTED"

COUNT=0
TOTAL=$(jq -r '.total' $TARGET_JSON_FILE)
jq -c '.{documents}|.[]|.[]|{id}+{title}+{create_date}' "$TARGET_JSON_FILE" | while read -r
i; do
    COUNT=$((COUNT + 1))

    #parse the line and get parameters from the line
    DOC_ID=$(jq -c '.{id}|.id' <<< $(printf '%s\n' "$i"))
    DOC_NAME=$(jq -c '.{title}|.title' <<< $(printf '%s\n' "$i"))
    DOC_DATE=$(jq -c '.{create_date}|.create_date' <<< $(printf '%s\n' "$i"))

    #EXPORT_FILE_NAME=$(date -d@${DOC_DATE:0:-3} +%Y-%m-%d)_${DOC_ID:1:-1}_${DOC_NAME:1:-
1}.zip
    EXPORT_FILE_NAME=$(date -d@${DOC_DATE:0:-3} +%Y-%m-%d)_${DOC_ID:1:-1}
    echo $COUNT OF $TOTAL = $EXPORT_FILE_NAME ____ ${DOC_NAME:1:-1}

    #Export ZIP
    curl --silent -X GET -H "Cookie: auth_token=$AUTH_TOKEN"
"$BASE_URL"/api/file/zip?id="${DOC_ID:1:-1}" -k -o "$DIR_ZIP"/"$EXPORT_FILE_NAME".zip

    #Export PDF
    curl --silent -X GET -H "Cookie: auth_token=$AUTH_TOKEN"
"$BASE_URL"/api/document/"${DOC_ID:1:-
1}"/pdf?margin=10\&metadata=false\&comments=true\&fitimagetopage=true -k -o
"$DIR_PDF"/"$EXPORT_FILE_NAME".pdf
done

```

```
#logout if finished
curl --silent -X POST -H "Cookie: auth_token=$AUTH_TOKEN" "$BASE_URL/api/user/logout" -k
```

## API Use: Flat Hirarchy Export + File List Overview

```
#!/bin/bash

BASE_URL="https://dms.yourdomain.de"
AUTH_TOKEN=$(curl -i -X POST -d username="THEUSER" -d password="THEPASSWORD"
"$BASE_URL/api/user/login" -k|grep "auth_token"|cut -c24-59)
BACKUP_DIR="/backup/teedy"
TARGET_DOCLIST_JSON=$BACKUP_DIR"/documentlist_forfiles.json"
TARGET_FILELIST_JSON=$BACKUP_DIR"/filelist.json"

mkdir -p "$BACKUP_DIR"
rm $TARGET_DOCLIST_JSON
rm $TARGET_FILELIST_JSON

echo "Retrieving document list"
#curl --silent -X GET -H "Cookie: auth_token=$AUTH_TOKEN"
"$BASE_URL/api/document/list?limit=0" -k | jq . > "$TARGET_DOCLIST_JSON"
curl --silent -X POST -H "Cookie: auth_token=$AUTH_TOKEN" "$BASE_URL/api/document/list" -d
"limit=999999" -k | jq . > "$TARGET_JSON_FILE"

echo "Retrieving file list based on document list"
COUNT=0
jq -c '.{documents}|.[].[]|.[]|{id}+{title}+{create_date}' "$TARGET_DOCLIST_JSON" | while read -
r i; do
    COUNT=$((COUNT + 1))
    DOC_ID=$(jq -c '.{id}|.id' <<< $(printf '%s\n' $i))
    DOC_ID=${DOC_ID:1:-1}
    curl --silent -X GET -H "Cookie: auth_token=$AUTH_TOKEN"
"$BASE_URL/api/file/list?id=$DOC_ID" >> "$TARGET_FILELIST_JSON"
    #echo -e "\n" >> "TARGET_JSON_FILE"
    echo Getting $COUNT : $DOC_ID
done

echo "Dumping files into flat hirarchy"
```

```

mkdir "$BACKUP_DIR"/flat_hirarchy/
COUNT=0
jq -c '.[[]|.[]|{create_date}+{name}+{id}+{document_id}+{mimetype}' "$TARGET_FILELIST_JSON" |
while read -r i; do
    COUNT=$((COUNT + 1))
    DOC_ID=$(jq -c '.|{document_id}|.document_id' <<< $(printf '%s\n' "$i"))
    DOC_ID=${DOC_ID:1:-1}
    FILE_NAME=$(jq -c '.|{name}|.name' <<< $(printf '%s\n' "$i"))
    FILE_NAME=${FILE_NAME:1:-1}
    FILE_DATE=$(jq -c '.|{create_date}|.create_date' <<< $(printf '%s\n' "$i"))
    FILE_ID=$(jq -c '.|{id}|.id' <<< $(printf '%s\n' "$i"))
    FILE_ID=${FILE_ID:1:-1}
    #MIMETYPE=$(jq -c '.|{mimetype}|.mimetype' <<< $(printf '%s\n' "$i")|sed 's#/#_#g')
    #MIMETYPE=${MIMETYPE:1:-1}
    #FILE_TYPE=$(echo "$FILE_NAME"|awk -F. '{print $(NF)}')
    #FILE_TYPE=${FILE_TYPE:0:-1}
    #EXPORT_FILE_NAME=$(date -d@${FILE_DATE:0:-3} +%Y-%m-%
%d)_"$FILE_ID"."$FILE_ID"."$MIMETYPE"."$FILETYPE"
    EXPORT_FILE_NAME=$(date -d@${FILE_DATE:0:-3} +%Y-%m-%d)_"$FILE_ID"."$FILE_ID"."$FILE_NAME"
    echo Getting $COUNT : $EXPORT_FILE_NAME
    curl --silent -X GET -H "Cookie: auth_token=$AUTH_TOKEN"
"$BASE_URL/api/file/$FILE_ID/data" -o "$BACKUP_DIR"/flat_hirarchy/"$EXPORT_FILE_NAME"
done

#logout if finished
curl --silent -X POST -H "Cookie: auth_token=$AUTH_TOKEN" "$BASE_URL/api/user/logout" -k

```

# App for Android

Sismics Docs is an easy to use, modern and nice DMS (document management system). Sismics Docs can be used on Android. But at the moment there is no downloadable App on Google PlayStore. In the official git repository you can find the required sources to compile it for yourself. This short tutorial will show you how you can do this. Please feel free to leave comments about additions and corrections. As usual different configurations force different bugs at different users.

## Variant 1: Build with Android Studio

1. Install Android Studio
2. Run Android Studio
3. Open the project "docs-android" in Android Studio
4. Connect Smartphone by USB in "data transfer" mode
5. Run the build process "Run" → "Run 'app'" - this will install Teedy on Android
6. Finished

## Variant 2: Build without Android Studio

### Pull the repository

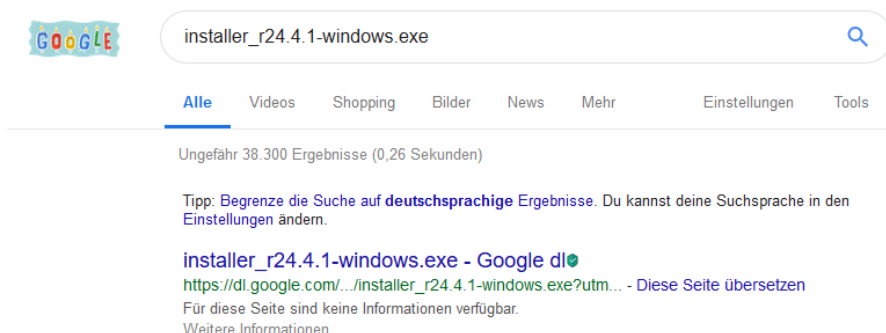
Go to <https://github.com/sismics/docs> and clone the repository to your local client.

### Install JDK 11

- <https://www.oracle.com/technetwork/java/javase/downloads/index.html>

### Install Android SDK Tools

The SDK Tools cannot be found directly on android homepage. Instead search them via web:



[https://filehippo.com/de/download\\_android\\_sdk](https://filehippo.com/de/download_android_sdk)

Download the setup and install it to the given standard directory. After that just update/upgrade the components. The installer will ask you if you like to do this.

## Configure Windows environment variables

- `JAVA_HOME` → `C:\Program Files\Java\jdk-11`
- `ANDROID_HOME` → `%userprofile%\AppData\Local\Android\android-sdk`
- `PATH` → `%userprofile%\AppData\Local\Android\android-sdk\tools` (add to existing)
- `PATH` → `%userprofile%\AppData\Local\Android\android-sdk\platform-tools` (add to existing)

## Create local.properties file in the repository

Locate the directory `<path to your repo>\docs-android\` and create a new file named `local.properties`. Open that file and insert:

```
C:\Users\mario\Git\teedy\docs-android
sdk.dir=C:\\Users\\mario\\AppData\\Local\\Android\\android-sdk
```

## Start the app building process

Open a new cmd shell window and insert the following commands

```
cmd
cd C:\Users\mario\Git\teedy\docs-android
#gradlew tasks
gradlew build
#gradlew --stacktrace --debug
```

## Copy the built .apk file to your phone

The compiled .apk file was generated in `<path to your repo>\docs-android\app\build\outputs\apk\release`. Just put it somewhere on your phone to access it from Android user interface.

## Sign the .apk file

Because the compiled file is not signed you cannot install it on your phone until you do a signing process. I found an application called ["apk-signer" in Google PlayStore](#) which did

the job for me.

After downloading and installing apk-signer, start apk-signer and select the Teedy apk file within the application. By selecting it will be signed. After this start the freshly signed apk file within apk-signer and you are allowed to install it.



## Sismics Docs

Möchtest du diese App installieren? Sie benötigt keinen besonderen Zugriff.

ABBRECHEN

INSTALLIEREN



Enter your configuration and start the app





To start, you must download and install  
Sismics Docs Server on [github.com/sismics/docs](https://github.com/sismics/docs) and enter its below

192.168.1.100:8080/dms/

Manual page

Username

LOGIN



