Comprehensive TEX Archive Network Developer's Guide

Version 3.*

Gerd Neugebauer

May 9, 2025

The Comprehensive TEX Archive Network (CTAN) is the major repository for the TEX world. It offers TEX distributions, packages, and more. CTAN provides access via the Web. This site has needed a relaunch. This document describes the consideration on the relaunch of the Web site for CTAN and how to perform certain tasks for installation, operation, and maintenance.

The drawing of the TEX lion shown on the cover page has been provided by Duane Bibby.

Authors

Gerd Neugebauer

gene@ctan.org

Contents

1. Introduction

For long years the Comprehensive Archive Network (CTAN) has served the TEX world as the major repository for all kinds of material. The actual servers have changed over the years.

End of 2011 an activity has been started to relaunch the Web presence of CTAN. The pages used at this time where produced and hosted by Jim Heffron as a private project and not officially by the CTAN team since the time Jim has left the CTAN team. These pages focused on the presentation of the TEX Catalogue and not on CTAN as such.

In December 2012 the redesigned Web site had it's public launch. The home page can be seen in figure 1.1.

Now it has served its purpose for several years. The technology has advanced. New frameworks have shown up and the principles for a modern Web application with high usability standards have made some progress. The appearance of release 3 is shown in figure 1.2.

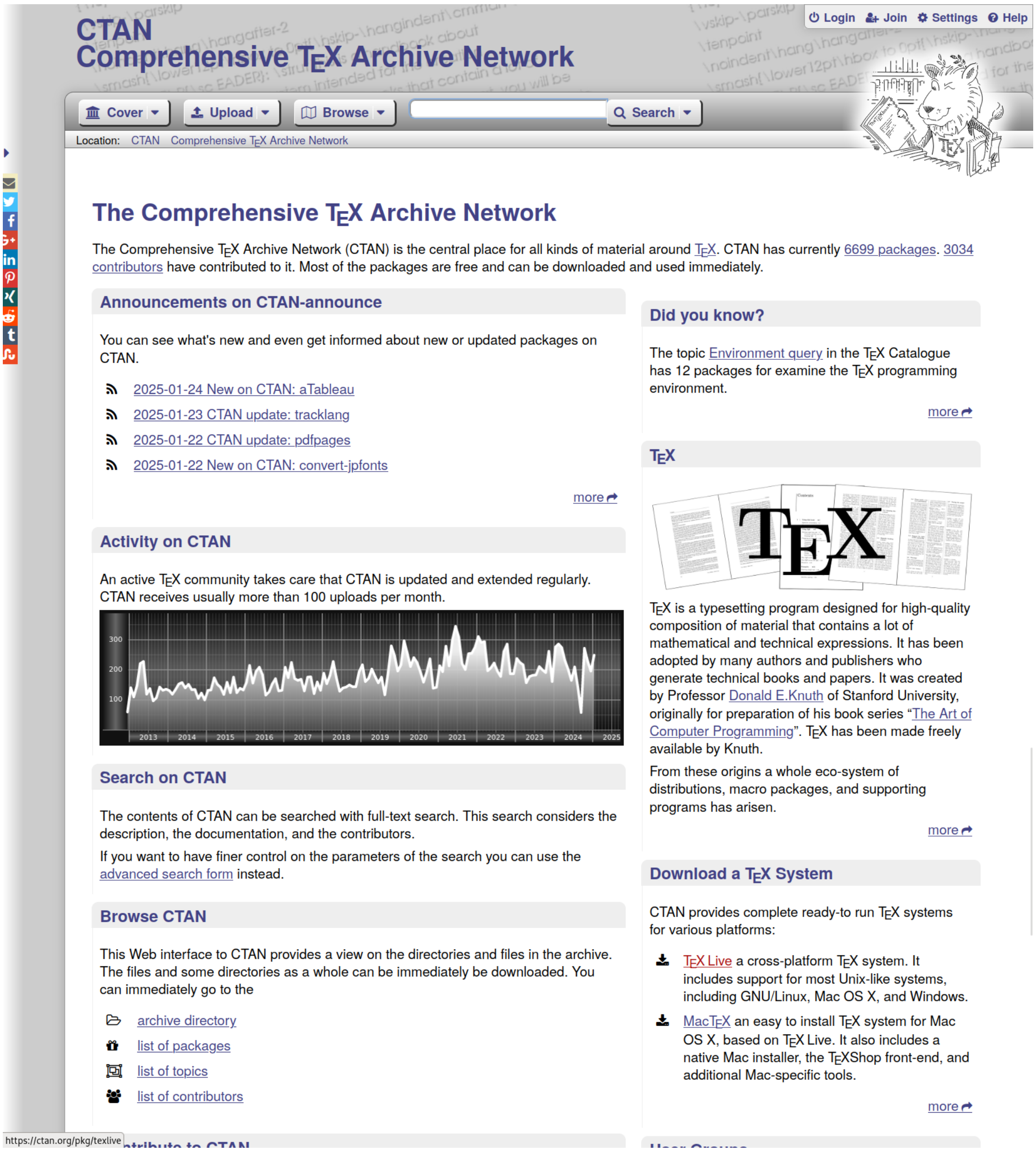
1.1. Goals

The Web site is accompanied with a few goals. Those goals are:

1. Presentation of a few static pages describing CTAN and such.
2. Browsing the CTAN archive by directory.
3. Browsing the CTAN archive by the Catalogue: packages, authors, topics.
4. Searching the CTAN.
5. Providing an uploading form for submitting packages.
6. Providing a registration form for CTAN mirrors.
7. Allow users to interact with the site by giving feedback, pointing out bugs and corrections, and possibly using additional community features.

1.2. Improvements

The relaunch might be accompanied by some community functions which goes into the direction of a communication platform and less into mere presentation of some information.



CTAN
Comprehensive TEX Archive Network
Login
Join
Settings
Help
Cover
Upload
Browse
Search
Location: CTAN Comprehensive TEX Archive Network

The Comprehensive TEX Archive Network

The Comprehensive TEX Archive Network (CTAN) is the central place for all kinds of material around TEX. CTAN has currently 6699 packages. 3034 contributors have contributed to it. Most of the packages are free and can be downloaded and used immediately.

Announcements on CTAN-announce

You can see what's new and even get informed about new or updated packages on CTAN.

- 2025-01-24 New on CTAN: aTableau
- 2025-01-23 CTAN update: tracklang
- 2025-01-22 CTAN update: pdfpages
- 2025-01-22 New on CTAN: convert-jpfonts

more

Activity on CTAN

An active TEX community takes care that CTAN is updated and extended regularly. CTAN receives usually more than 100 uploads per month.

Search on CTAN

The contents of CTAN can be searched with full-text search. This search considers the description, the documentation, and the contributors.

If you want to have finer control on the parameters of the search you can use the advanced search form instead.

Browse CTAN

This Web interface to CTAN provides a view on the directories and files in the archive. The files and some directories as a whole can be immediately be downloaded. You can immediately go to the

- archive directory
- list of packages
- list of topics
- list of contributors

Did you know?

The topic Environment query in the TEX Catalogue has 12 packages for examine the TEX programming environment.

more

TEX

TEX is a typesetting program designed for high-quality composition of material that contains a lot of mathematical and technical expressions. It has been adopted by many authors and publishers who generate technical books and papers. It was created by Professor Donald E.Knuth of Stanford University, originally for preparation of his book series "The Art of Computer Programming". TEX has been made freely available by Knuth.

From these origins a whole eco-system of distributions, macro packages, and supporting programs has arisen.

more

Download a TEX System

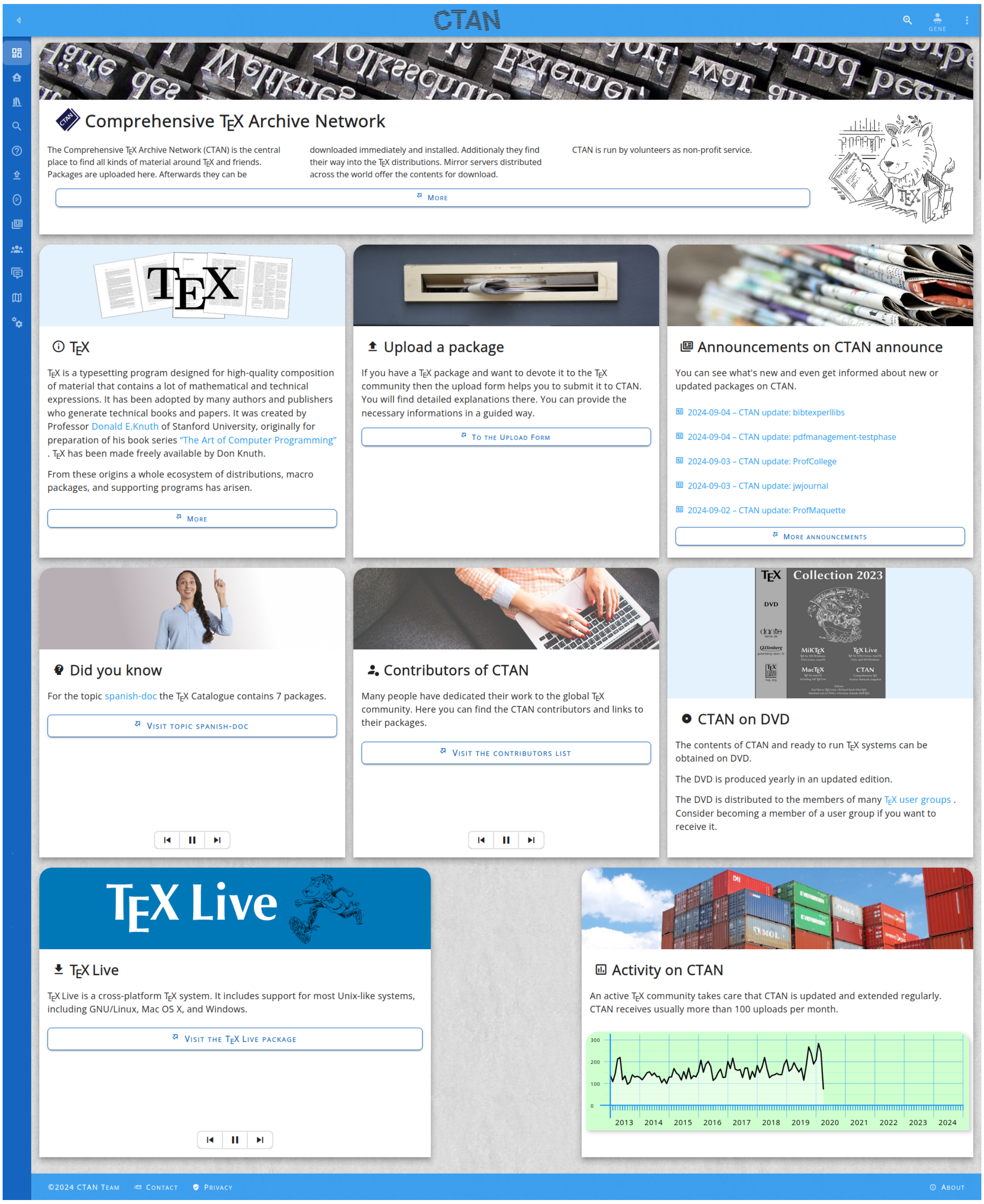
CTAN provides complete ready-to run TEX systems for various platforms:

- TEX Live a cross-platform TEX system. It includes support for most Unix-like systems, including GNU/Linux, Mac OS X, and Windows.
- MacTEX an easy to install TEX system for Mac OS X, based on TEX Live. It also includes a native Mac installer, the TEXShop front-end, and additional Mac-specific tools.

more

https://ctan.org/pkg/texlive

Figure 1.1.: The CTAN home page in version 2.7



Comprehensive TEX Archive Network

The Comprehensive TEX Archive Network (CTAN) is the central place to find all kinds of material around TEX and friends. Packages are uploaded here. Afterwards they can be downloaded immediately and installed. Additionaly they find their way into the TEX distributions. Mirror servers distributed across the world offer the contents for download.

CTAN is run by volunteers as non-profit service.

TEX

TEX is a typesetting program designed for high-quality composition of material that contains a lot of mathematical and technical expressions. It has been adopted by many authors and publishers who generate technical books and papers. It was created by Professor Donald E.Knuth of Stanford University, originally for preparation of his book series "The Art of Computer Programming" . TEX has been made freely available by Don Knuth.

From these origins a whole ecosystem of distributions, macro packages, and supporting programs has arisen.

Upload a package

If you have a TEX package and want to devote it to the TEX community then the upload form helps you to submit it to CTAN. You will find detailed explanations there. You can provide the necessary informations in a guided way.

Announcements on CTAN announce

You can see what's new and even get informed about new or updated packages on CTAN.

- 2024-09-04 – CTAN update: bibtexperllibs
- 2024-09-04 – CTAN update: pdfmanagement-testphase
- 2024-09-03 – CTAN update: ProfCollege
- 2024-09-03 – CTAN update: jwjournal
- 2024-09-02 – CTAN update: ProfMaquette

Did you know

For the topic spanish-doc the TEX Catalogue contains 7 packages.

Contributors of CTAN

Many people have dedicated their work to the global TEX community. Here you can find the CTAN contributors and links to their packages.

CTAN on DVD

The contents of CTAN and ready to run TEX systems can be obtained on DVD.

The DVD is produced yearly in an updated edition.

The DVD is distributed to the members of many TEX user groups . Consider becoming a member of a user group if you want to receive it.

TEX Live

TEX Live is a cross-platform TEX system. It includes support for most Unix-like systems, including GNU/Linux, Mac OS X, and Windows.

Activity on CTAN

An active TEX community takes care that CTAN is updated and extended regularly. CTAN receives usually more than 100 uploads per month.

Figure 1.2.: The CTAN home page

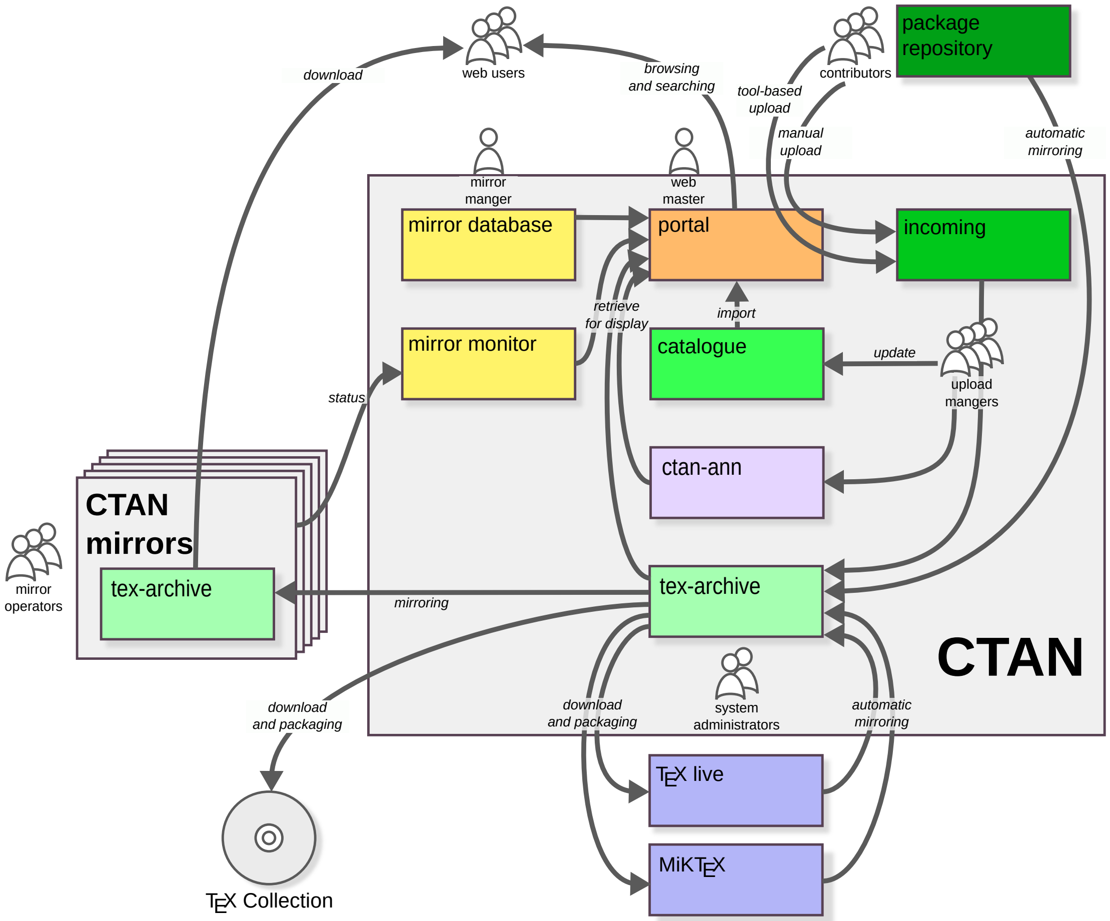


Figure 1.3.: Interaction flows overview

1.3. Interaction flows

missing

2. Rules

This chapter collects some of the rules to which the developers should obey.

1. All rules can be overruled
   All rules can be overruled. This requires a sufficient justification in written form.
2. Development language is English
   The language used throughout this application is English (in the British dialect).

   This applies to the documents describing the application as well the documentation of the sources, the description of the internal aspects, and the naming of files.

   An exception are documents for external audiences. Since the site is multilingual this means that the texts presented in th UI can also be considered to be an exception.
3. Sematic versioning
   Whenever versioning is required then we use the scheme of semantic versioning [SEM].
4. Documentation
   Documentation is required. For details see section 2.1.

2.1. Documentation

Documentation is not optional. Several goals can and should be achieved with appropriate documentation:

Preserving knowledge

New developers need to get used to the standards in an application. Asking the old fellows might not be possible. Oral tradition is time consuming and error prone.

The pure source code shows the how, but not the why. Thus documentation has to fill the gap.

Overall consistency

If rules are formalised and documented then it is easier to achieve overall consistency.

Written documentation helps to think about the rules and discuss them in a team.

Ensure quality

One principle in quality assurance is to do things several times. But sheding light on the same spot from different directions helps to avoid dark spots.

Thus the following activities support this idea:

- Write a specification of what should be achieved.
- Write source code to do what is required.
- Write test cases to check that the goals have been reached. (and test regularly)

2.1.1. General introduction

README.md

missing

2.1.2. Documenting Java code

License header

```
/*
 * Copyright ľ 2022-2025 The CTAN Team and individual authors
 *
 * This file is distributed under the 3-clause BSD license.
 * See file LICENSE for details.
 */
```

JavaDoc

missing

2.1.3. Documenting Vue code

License header

```
<!--
  -- Copyright ľ 2022-2025 The CTAN Team and individual items
  --
  -- This file is distributed under the 3-clause BSD license.
  -- See file LICENSE for details.
  -->
```

JSDoc

missing

2.1.4. Documenting Javascript

License header

```
/*
** Copyright ľ 2024-2025 The CTAN Team and individual authors
**
** This file is distributed under the 3-clause BSD license.
** See file LICENSE for details.
*/
```

JSDoc

missing

2.1.5. Document the licensing

Especially in an Open-Source context the licensing of the different parts should be made clear.

This application is distributed under the 3-clause BSD license.

The licensing information has to be put in several places. If parts of the software are reused somewhere else then the lisense should not be lost.

- Provide a file `LICENSE` in the top-level directory of the project containing the licensing conditions.
- Each source file containing program code should have a head which names the license under which is distributed.

missing something else?

2.1.6. Documenting LaTeX

License header

```
%%
    *****************************************************************************

%%
%% Copyright Ï 2024-2025 The CTAN Team and individual authors
%%
%% This file is distributed under the 3-clause BSD license.
%% See file LICENSE for details.
%%
%% Author: <a href="mailto:gene@ctan.org">Gerd Neugebauer</a>
%
    %-----------------------------------------------------------------------------
```

missing

2.2. Definition of done

The term "definition of done" originated in the Scrum methodology [SCR] for software development.

According to the Scrum Guide

> The Definition of Done is a formal description of the state of the Increment when it meets the quality measures required for the product.

1. Unit tests written and passing.
   - The unit tests for Java should cover more than 70% of the code.
   - The unit tests for Javascript are optional.

   - The active unit tests all passed.

2. The build passed.
3. The user interface has been tested.
4. The documentation is up-to-date.
5. The generation of Javadoc passed.

3. Architecture

missing

4. Architecture Decision Records

more

4.1. Back-end framework: Dropwizard

Issue

The overall architecture of a Web application like the CTAN site consists of back-end server and a client part. The back-end server should be based on an appropriate framework to ease the task of programming.

Decision

The back-end framework Dropwizard is used.

Status

Decided

Constraints

1. The framework should be Open-Source.
2. The programming language for the back-end server is Java – or at least it runs in a JVM.

Positions

1. Dropwizard
2. Spring Boot
3. Grails

Argument

- Dropwizard has good support as one of the major frameworks.
- Dropwizard has less features preconfigured. Thus you can freely decide what to add.

- Spring Boot has more features enabed by default. Thus the startup appears to be slower.
- Grails is based on Spring boot. It is tailored towards server-side rendering.

4.2. Database: Postgres

Issue

The various data items need to be accessed efficiently. This includes the catalogue as well as other entities like users and parameters.

Decision

The data is primarily stored in a SQL database with PostgreSQL as DBMS.

Status

Decided

Constraints

1. The system is self-hosted.
2. The underlying operating system is Linux – most probably Debian.
3. The database should be Open-Source.

Positions

1. Keep the data in files. Possible formats include XML, JSON, or YAML.
2. Use a SQL database. Possible DBMS' are PostgreSQL or MariaDB.
3. Use a non-SQL database.

Argument

- The use of an SQL database allows efficient and fast access to the data. This supports the usability and acceptance by the end user.
- The DBMS should be an Open-Source product to avoid any costs for development and production.
- PostgreSQL DBMS was already in use for the mirror database.
- PostgreSQL was well-known to the persons involved.

Implications

- The workflow of the upload managers is still based on the XML files of the Catalogue. Thus a regular update of the database is required.

4.3. Front-end framework: Vuetify

Issue

The overall architecture of a Web application like the CTAN site consists of back-end server and a client part. The front-end client should be based on an appropriate framework to ease the task of programming.

Decision

The front-end framework Vuetify is used.

Status

Decided

Constraints

1. The framework should be Open-Source.
2. The programming language for the front-end server is JavaScript.

Positions

1. Vuetify is a framework based on Nuxt and thus Vue. It provides components to be compatible to Google's Material Design
2. Angular is one of the older frameworks. It has a large user base and many components. Nevertheless it suffers from some ancient design decisions. Especially the separation of structure, design, and functionality can be distracting.
3. React is one of the major frameworks. It is gaining momentum and is a real alternative.
4. Svelte is a newcomer. It has the advantage to be fast. Nevertheless as a newcomer the stability and richness of functions was not given at the time when this decision has to be made.
5. JQuery is a well established framework. It provides a small set of low-level functions to make the same code running under all supported browser versions.

Argument

- Vuetify provides useful components.
- The components are prepared to buils a modern looking UI.
- The bundling of a component in one file helps to keep things organised and to have an easier overview.

5. Environments

5.1. Data flows

Data from various sources play a role in the context of the CTAN site. An overview is shown in figure 5.1. The components and data flows are described in the following section.

5.1.1. TEX-Archive Directory

The TEX-archive directory (`/home/ftp/pub/tex`) is a directory which contains the raw files of CTAN. The TEX-archive directory is manually filled with the contributions by the upload managers. In addition some external sources are mirrored into this directory structure.

The TEX-archive directory is traversed by the Archive Indexer. The search index is updated accordingly.

The upload managers place the contributions in the TEX-archive directory. From there the files are accessible through the CTAN site or the mirror servers.

5.1.2. Site Database

The site database contains the data for the CTAN site. The CTAN catalogue is copied into the site database (database: `portalctan; host: localhost; port: 5432`) contains the These data has the master in the XML files of the catalogue. It is located on an instance of PostgreSQL.

Other data is primarily contained in the site database. This is especially the attributes of the users and other data not originated in the catalogue XML files.

5.1.3. Incoming Directory

The incoming directory (`/serv/www/www.ctan.org/incoming`) is used by the CTAN site to store incoming submissions. The upload managers prepare the submission and finally place them in the TEX-archive directory and update the Catalogue in the Catalogue Repository.

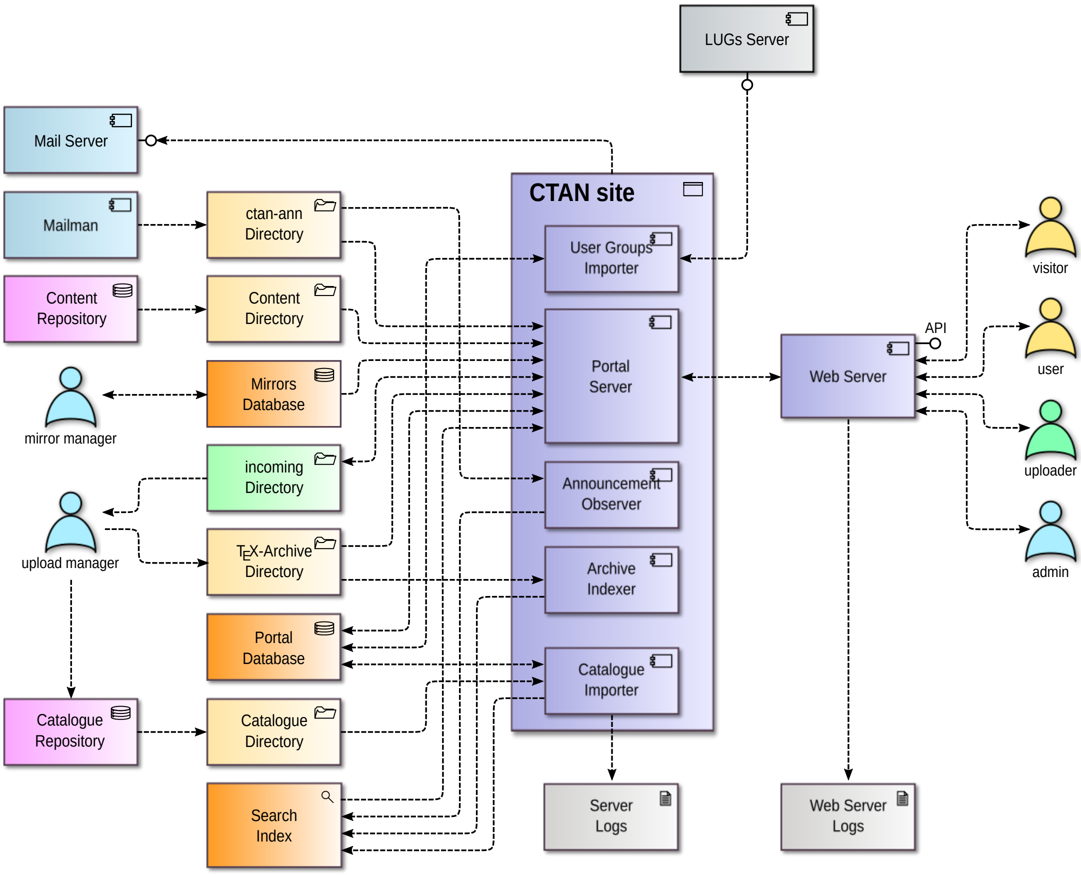


Figure 5.1.: Data flows in the context of the CTAN site

5.1.4. Content Repository

The content repository contains the pages and fragments in the supported languages. This allows an update without rebooting the web application.

The content repository is checked out into the content directory regularly.

The content repository is a Git repository. It is located at `https://gitlab.com/comprehensive-tex-archive-network/ctan-content`.

5.1.5. Content Directory

The CTAN site is designed to be multi-lingual. Currently the languages English and German are present. The pages and fragments for the supported languages are stored in the content repository and checked out into a working directory, the content directory (`/serv/www/www.ctan.org/ctan-content`). From there the CTAN site read the texts and provides them for the Browser UI.

The content directory is updated on a regular schedule. This means a checkout is scheduled.

5.1.6. Catalogue Repository

The catalogue is kept in the catalogue repository. The catalogue repository is am Apache Subversion repository. The upload managers maintain the package metadata in the catalogue. The catalogue is checked out regularly into a workspace in the catalogue directory.

5.1.7. Catalogue Directory

The catalogue directory (`/serv/www/www.ctan.org/texcatalogue/entries`) is a workspace of the catalogue repository. From here the site database and the search index are updated.

5.1.8. Search Index

The CTAN site uses Apache Lucene as search engine. This search engine maintains a search index in the directory `/serv/www/www.ctan.org/index`.

5.1.9. Server Logs

The back-end server is an instance of Apache Tomcat. It writes log files.

5.1.10. Web-server Logs

The Web server acts primarily as proxy server. The Web server is an instance of the Apache HTTPD. It writes log files.

5.1.11. Mail Server

The mail server is an SMTP server to deliver outgoing mails from the CTAN site.

5.1.12. Mailman

Mailman 3 is used to manage mailing lists. Especially the mailing list ctan-ann is processed this way.

Regularly the mails sent out is stored in the ctan-ann directory.

5.1.13. ctan-ann Directory

The ctan-ann directory (`/serv/www/www.ctan.org/ctan-ann`) contains copies of the announcements of package uploads on the mailing list ctan-ann.

5.1.14. Mirrors Database

The mirrors database (database: `ctanctan; host: localhost; port: 5432`) contains the mirrors servers. It is located on an instance of PostgreSQL.

5.1.15. Lugs Server

The local TEX user groups (LUGs) are maintained by the NTG. The CTAN site retrieves this data from `https://www.ntg.nl/lug/lugs/` and offers a page with it.

5.1.16. Visitor

Visitor is one of the roles in the context of CTAN. A visitor uses the CTAN site. This can be done anonymously. Then the visitor has not further special rights.

5.1.17. User

User is one of the roles in the context of CTAN. A user is authenticated. Thus the additional rights and functions of an authenticated user are available. A user is a visitor with additional permissions.

A user has to go through a registration process and a login.

5.1.18. Uploader

An uploader is a visitor who submits a package to CTAN. This can be performed anonymously or authenticated. When the uploader is authenticated then some more functionality is available.

5.1.19. Admin

Admin usually perform their tasks in the background behind the CTAN site. Nevertheless some functionality is also provided via the Web site.

5.1.20. Upload Manager

The upload manager takes care of upload. When a contributor has uploaded a new package or an update of a package. The upload manager receives am email notification about an upload. Then the uploaded package can be retrieved from the incoming directory and move the package files to the TeX-archive directory and the catalogue repository.

5.1.21. Mirrors Manager

The submission of new mirrors can be performed with the CTAN site. The mirror manager updates the active mirrors in the mirrors database.

5.2. Development environment

The development environment should be located on the personal server of the developers.

missing

5.3. Production environment

The production environment is mainly located on the server `irony`.

missing

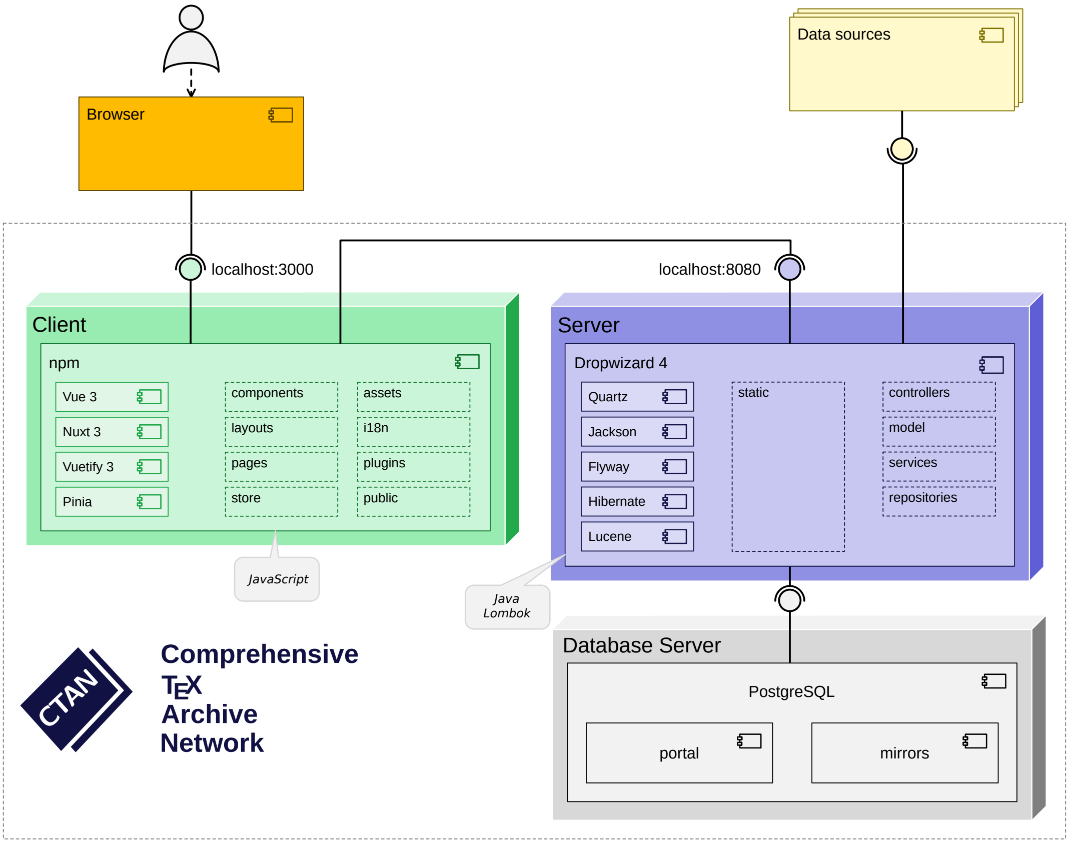


Figure 5.2.: The development environment

6. Setup a development environment

6.1. Git

6.2. Workspaces

6.3. Directories

6.4. PostgreSQL

6.5. Java

6.6. Gradle

6.7. Node and Yarn

6.8. Perl

6.9. IDE: Eclipse

7. Installation of Production

This chapter describes all steps necessary to bring up the CTAN site.

7.1. Prerequisites

In the following description we assume a Debian-based system to be present.

Apache HTTPD should be installed with a package manager in Debian.

Apache Tomcat should be installed with a package manager in Debian.

PostgreSQL should be installed with a package manager in Debian.

Java 21 should be installed with a package manager in Debian.

7.2. Access to the Sources

For simplicity we assume that the sources for the CTAN site have been checked out from the Git repository. See section ?? for details.

In the following we assume that the location of the directory is stored in the environment variable SITE_SRC:

```
-> export SITE_SRC=‘pwd‘
```

This is only for simplifying the description of the installation and not required for building or running the site.

7.3. Cron Jobs

Usually the time-based services are run from within the CTAN site as Quartz jobs. The are distributed within the war archive of the CTAN site and need no special installation.

Nevertheless one task is performed by a traditional cron job. This is the checkout of the Catalogue from the Subversion repository. This cron job has to be run by an authorized user. It runs unattended.

7.4. Preparing the Databases

The CTAN site needs a PostgreSQL database installed. The site needs access to two databases site and ctan.

7.4.1. Creating the Database `site`

The database site is used to store the sites own data. Thus the CTAN site needs reading and writing access. For the access to the database the technical user `www` is used.

7.4.2. Creating the Database `ctan`

The database ctan contains the information on the CTAN mirrors. The CTAN site just needs reading access to this database. For the access to the database the technical user `www` is used.

If this has not done before use psql:

```
-> psql ctan
```

and grant the required permissions:

```
grant select on all tables in schema ctan to www;
```

Since there is no distinction for the environments only one database is used for all environments.

7.5. Preparing the Working Directory

The CTAN site is mainly run as Web application in Tomcat. This instance of Tomcat is run under the user `tomcat` and the group `tomcat`. We have to make sure that this user or this group have the appropriate reading and writing permissions.

7.5.1. The Index Directory

The search stores the index information in the file system. For this purpose several directories are required. These directories need to be readable by the user under which the servlet container is running. This user is named `tomcat`. The setup is performed with the following commands:

```
-> mkdir -p
-> chgrp tomcat
-> chmod g+rw
```

To set up the server for the test environment the following commands are required:

```
-> mkdir -p -test
-> chgrp tomcat -test
-> chmod g+rw -test
```

And finally in the development environment another set of similar commands does the same:

```
-> mkdir -p -dev
-> chgrp tomcat -dev
-> chmod g+rw -dev
```

7.5.2. The Static Index

The search index contains entries for the static pages as well as for the data from the databases and possibly the TEX archive. Thus it is necessary to populate the index for the static pages.

This command creates and fills the directory . If you run it on the target machine then everything is prepared for the production environment.

If you run it on another machine then you can simply move it to the target machine.

If you are about to prepare the dev of test environment then simply copy or move this directory to the environment's sub-directory.

7.5.3. The Logs Directory

The CTAN site is run in an Apache Tomcat as servlet container. Usually the user running it is named `tomcat`. He belongs to the group `tomcat`. Thus this user needs write access for the places where the log files are located.

```
-> mkdir -p /logs
-> chgrp tomcat /logs
-> chmod g+rwx /logs
```

7.6. Preparing the Web Server

It is assumed that the Apache Web server is used on the frontend. This Web server has to be configured. The configuration must be places in the file

```
/etc/apache2/sites-available/www.ctan.org
```

This file can be found in the sources under

```
$SITE_SRC/src/apache2/www.ctan.org
```

Thus

```
-> cp $SITE_SRC/src/apache2/www.ctan.org /etc/apache2/sites-available/www.
   ctan.org
```

Some artifacts are served from the Web server. Those are required for producing an error page when the servlet container Tomcat is not reachable. Those artifacts are located in the directory `/htdocs` and provided in the sources on the directory `$SITE_SRC/src/apache2/htdocs`. Thus

```
-> cp -r $SITE_SRC/src/apache2/htdocs /serv/www/www.ctan.org/
```

7.7. Building the Web Application

To build the war file of the CTAN site follow the steps described in chapter 8. Finally you should have a file named `target/ctan-site-3.*.war` to continue with.

7.8. Deploying the Web Application

We assume that the war for the CTAN site has been created (see section 8). The war should be named `ctan-site-`*x.y.z*`.war` where x.y.z is the version number of the current build.

First, we need a ssh tunnel to the production host:

```
-> ssh irony -f -L 9999:localhost:443 -N
```

If Tomcat has the management Web application installed then you can navigate to `http://localhost.ctan.org:9999/manager/html/`. Here you find a list of deployed web applications (see figure 7.1)

1. Create a lock file for the maintenance. The creation of this lock file redirects users to the maintenance page by the Apache. No traffic is passed to Tomcat.

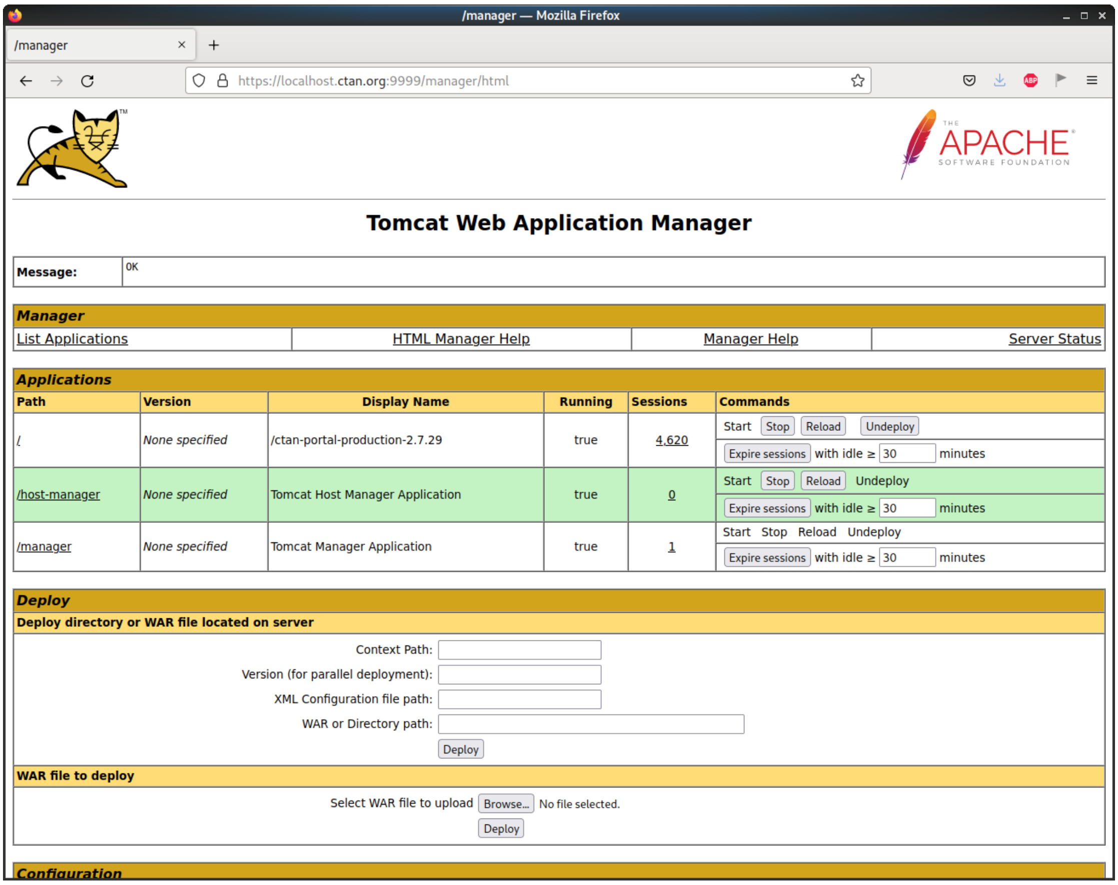
   ```
   -> touch /serv/www/www.ctan.org/maint.lock
   ```

2. Next undeploy any instance already running.
3. Then you can deploy the war file under the context `/ctan-site` (see figure 7.2).
4. Restart Tomcat to prevent problems with the memory management:

   ```
   -> sudo /etc/init.d/tomcat-ctan restart
   ```

5. Release the lock to reactivate the site:

   ```
   -> rm /serv/www/www.ctan.org/maint.lock
   ```



/manager — Mozilla Firefox

/manager

https://localhost.ctan.org:9999/manager/html

Tomcat Web Application Manager

Message:	OK

Manager			
List Applications	HTML Manager Help	Manager Help	Server Status

Applications

Path	Version	Display Name	Running	Sessions	Commands
/	*None specified*	/ctan-portal-production-2.7.29	true	4,620	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	*None specified*	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	*None specified*	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Deploy

Deploy directory or WAR file located on server

Context Path:

Version (for parallel deployment):

XML Configuration file path:

WAR or Directory path:

Deploy

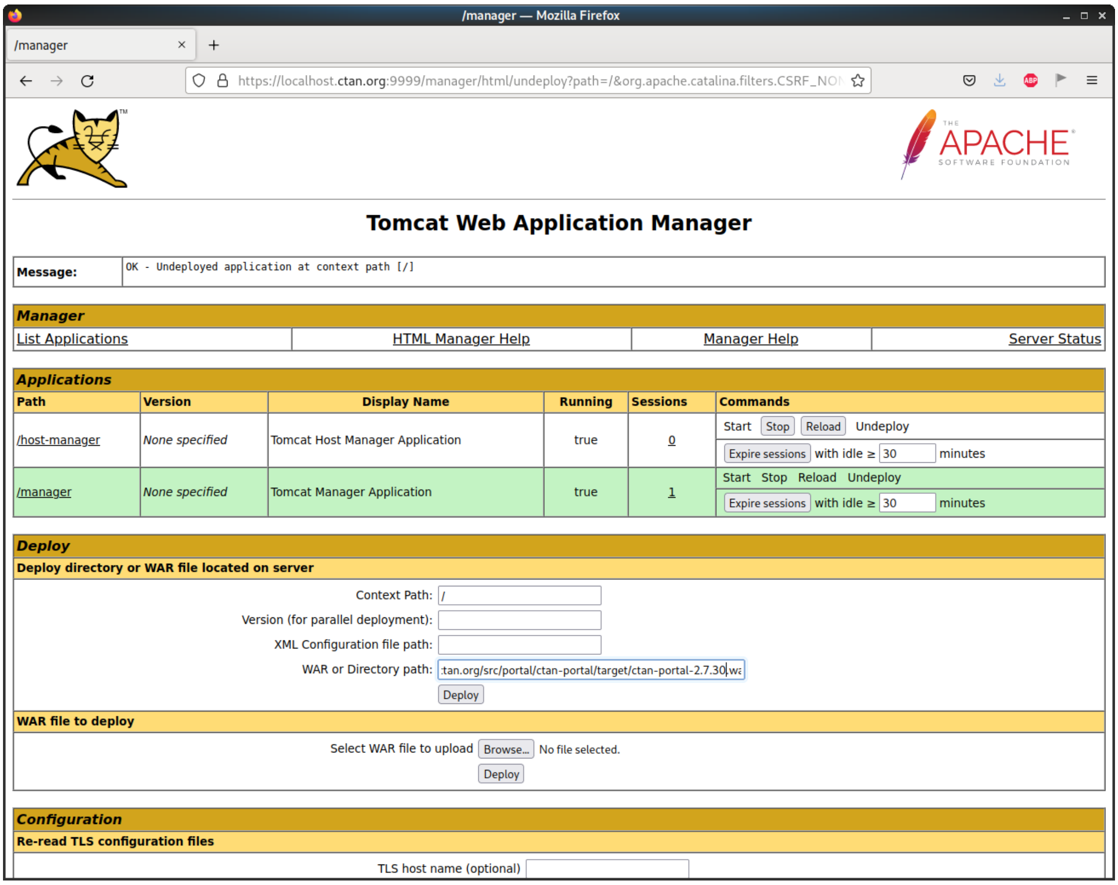
WAR file to deploy

Select WAR file to upload Browse... No file selected.

Deploy

Configuration

Figure 7.1.: Tomcat manager



/manager — Mozilla Firefox

https://localhost.ctan.org:9999/manager/html/undeploy?path=/&org.apache.catalina.filters.CSRF_NON

Tomcat Web Application Manager

Message:	`OK - Undeployed application at context path [/]`

Manager

List Applications	HTML Manager Help	Manager Help	Server Status

Applications

Path	Version	Display Name	Running	Sessions	Commands
/host-manager	*None specified*	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	*None specified*	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Deploy

Deploy directory or WAR file located on server

Context Path: /

Version (for parallel deployment):

XML Configuration file path:

WAR or Directory path: :tan.org/src/portal/ctan-portal/target/ctan-portal-2.7.30.wa

Deploy

WAR file to deploy

Select WAR file to upload Browse... No file selected.

Deploy

Configuration

Re-read TLS configuration files

TLS host name (optional)

Figure 7.2.: Deploying a Web application

7.9. Supporting Utilities

The files contained in the sources under `$SITE_SRC/bin` contains a supporting utility. It should be copied to `/bin`.

Beware to preserve the permission bits for the executable.

8. Build

9. Quality assurance

9.1. Java: unit tests

9.1.1. Definition of the unit tests

The unit tests are written utilizing JUnit.

The unit tests can be found in the directory `src/test/java`.

9.1.2. Running the unit tests

To run the unit tests issue the following command:

```
-> ./grailsw test
```

The results can be found in the directory `build/reports/tests/test`.

9.2. Java: checkstyle

Checkstyle is a tool for static code analysis focusing on uniform appearance of the code. It has a larger set of rules which are checked and might lead to errors or warnings.

9.2.1. Definition of the checkstyle rules

The rules applied for this project can be found in the file `config/checkstyle/checkstyle.xml`.

9.2.2. Running Checkstyle

To run Checkstyle on the productive Java code issue the following command:

```
-> ./grailsw checkstyleMain
```

The results can be found in the directory `build/reports/checkstyle/main.html`.

To run Checkstyle on the Java test code issue the following command:

```
-> ./grailsw checkstyleTest
```

The results can be found in the directory `build/reports/checkstyle/test.html`.

9.3. Java: SpotBugs

SpotBugs is a tool for static code analysis. It has a larger set of rules which are checked and might lead to errors or warnings.

9.3.1. Running SpotBugs

To run SpotBugs on the productive Java code issue the following command:

```
-> ./grailsw spotbugsMain
```

The results can be found in the directory `build/reports/spotbugs/main`.

To run SpotBugs on the Java test code issue the following command:

```
-> ./grailsw spotbugsTest
```

The results can be found in the directory `build/reports/spotbugs/test`.

9.4. Vue: eslint

9.4.1. Definition of the eslint rules

The rules applied for this project can be found in the file `src/client/eslint.config.js`.

9.4.2. Running eslint

To run eslint on the client code issue the following command while you are in the directory `src/client`:

```
-> yarn lint
```

Appendix

A. Glossary

Author

See Contributor.

Catalogue

The Catalogue is the source of all information about packages, authors, and topics on CTAN. It has been created initially by Graham J. Williams and is now maintained by the CTAN team.

The Catalogue is stored in a set of XML files. Initially static Web pages have been generated from these sources. Nowadays the XML file are regularly imported into a database. From there the data is presented on the Web.

Contributor

A contributor is a person or group who are related to a package. This can be as active or former maintainer or as uploader.

Formerly the contributor has been called author.

CTAN

CTAN is the name of the Comprehensive TEX Archive Network. Initially it consisted of three primary servers in Germany, the US, and the UK and a lot of mirrors. Nowadays the internet is more performant and more reliable. Thus only one primary server is left behind.

CTAN-ann

CTAN-ann is a mailing list to publish the announcements of packages on CTAN. The email address is ctan-ann@ctan.org.

Incoming

The uploaded packages are stored in the incoming directory. The upload managers take care of updating the catalogue and move the content to the TEX-archive.

License

The packages on CTAN are provided with the licenses as defined by the contributors. These can be Open-Source licenses but are not restricted to those. The TEX distributions which take the packages from CTAN may be more restrictive and offer only free packages.

The software of the CTAN site itself is distributed under the 3-clause BSD license (see page 39).

MiKTEX

MiKTEX is one of the major TEX distributions. It gets its contents directly from CTAN.

Mirror

Mirrors are secondary servers around the world which provide access to the TEX archive. For this purpose they copy the content of the primary server and offer it via HTTPS mand maybe HTTP and FTP.

In order to be an official mirror the server has to be registered at CTAN.

Mirror database

The mirrors are kept in a separate database. This mirrors database is read by the CTAN site. The update is external.

Mirror monitor

The mirrors are monitored by CTAN. This allows us to redirect the users only to the reachable and up-to-date mirrors.

Package

A package on CTAN denotes a set of files to achieve a certain goal when typesetting with TEX and friends.

TEX archive

The TEX archive is a directory structure which contains the sources of the contributed packages on CTAN.

TEX Live

TEX Live is one of the major TEX distributions. It gets its contents directly from CTAN.

Topic

Topics provide a means to classify the packages on CTAN. Initially the directory structure of the TEX archive was the only way to express a classification. The topics allow more expressiveness since a package may have more than one topic.

B. BSD 3-Clause License

Bibliography

[SCR] The 2020 scrum guide. https://scrumguides.org/scrum-guide.html.

[SEM] Semantic versioning 2.0.0. https://semver.org/.

Index