
GLPI Installation

Release 9.5

GLPI Project, Teclib'

Nov 18, 2019

1 Prerequisites	3
1.1 Web server	3
1.2 PHP	3
1.2.1 Mandatory extensions	3
1.2.2 Optional extensions	4
1.2.3 Configuration	4
1.3 Database	4
2 Install GLPI	7
2.1 Choose a version	7
2.2 Download	7
2.3 Installation	8
2.4 Files and directories locations	8
2.5 Post installation	9
3 Install wizard	11
3.1 Choose lang (Select your language)	11
3.2 License	11
3.3 Install / Update	12
3.3.1 Environment checks	13
3.3.2 Database connection	14
3.3.3 Database choice	14
3.3.4 Database initialization	15
3.3.5 Telemetry informations	16
3.3.6 End of installation	16
4 Timezones	19
4.1 Non windows users	19
4.2 Windows users	19
4.3 Grant access	20
5 Update	21
6 Command line tools	23
6.1 Console options	23
6.2 Install	24
6.3 Update	24

6.4	Database tools	24
6.4.1	Database schema check	24
6.5	Tasks tools	25
6.5.1	Task unlock	25
6.6	Migration tools	25
6.6.1	From MyISAM to InnoDB	25
6.6.2	Missing timestamps builder	25
7	Advanced configuration	27
7.1	SSL connection to database	27

This documentation presents **GLPI** installation instructions.

GLPI (Gestion Libre de Parc Informatique) is a free (as in “free speech” not as in “free beer”!) asset and helpdesk management solution accessible from a web browser built to manage all you asset management issues, from hardware components and software inventories management to user helpdesk management.

GLPI is a Web application that will need:

- a webservice;
- PHP;
- a database.

1.1 Web server

GLPI requires a web server that supports PHP, like:

- Apache 2 (or more recent);
- Nginx;
- Microsoft IIS.

1.2 PHP

As of 9.2 release, GLPI requires [PHP 5.6](#) or more recent. As of 9.5 release, GLPI requires [PHP 7.0.8](#) or more recent.

Note: We recommend to use the most recent stable PHP release for better performances.

1.2.1 Mandatory extensions

Following PHP extensions are required for the app to work properly:

- `curl`: for CAS authentication, GLPI version check, Telemetry, ...;

- `fileinfo`: to get extra informations on files;
- `gd`: to generate images;
- `json`: to get support for JSON data format;
- `mbstring`: to manage multi bytes characters;
- `mysqli`: to connect and query the database;
- `session`: to get user sessions support;
- `zlib`: to get backup and restore database functions;
- `simplexml`;
- `xml`.

1.2.2 Optional extensions

Note: Even if those extensions are not mandatory, we advise you to install them anyways.

Following PHP extensions are required for some extra features of GLPI:

- `cli`: to use PHP from command line (scripts, automatic actions, and so on);
- `domxml`: used for CAS authentication;
- `imap`: used for mail collector ou user authentication;
- `ldap`: use LDAP directory for authentication;
- `openssl`: secured communications;
- `xmlrpc`: used for XMLRPC API.
- `APCu`: may be used for cache; among others (see [caching configuration \(in french only\)](#)).

1.2.3 Configuration

PHP configuration file (`php.ini`) must be adapted to reflect following variables:

```
memory_limit = 64M ;           // max memory limit
file_uploads = on ;
max_execution_time = 600 ;     // not mandatory but recommended
register_globals = off ;       // not mandatory but recommended
magic_quotes_sybase = off ;
session.auto_start = off ;
session.use_trans_sid = 0 ;    // not mandatory but recommended
```

1.3 Database

Warning: Currently, only [MySQL](#) (5.6 minimum) and [MariaDB](#) (10.0 minimum) database servers are supported by GLPI.

In order to work, GLPI requires a database server.



Proceed as follow:

1. Choose a version,
2. Download the archive,
3. Install :)

2.1 Choose a version

Note: It is highly recommended you choose the latest stable release for a production usage.

GLPI follows a semantic versioning scheme, on 3 digits. The first one is the major release, the second the minor and the third the fix release.

Major releases may come with important incompatibilities as well as new features; minor versions may bring new features as well, but stay perfectly compatible inside a major version.

Fixes releases will only fix reported issues without adding anything new.

2.2 Download

Warning: On GitHub, there are always two archives named *Source code* which should not be used.

Go to the *download* section of the [GLPI website](#) (or get archive directly from [Github release](#)) and choose the `glpi-{version}.tgz` archive.

2.3 Installation

Note: Packages may be available from your Linux distribution (Red Hat, CentOS, Fedora, Ubuntu, ...) that you should use with your standard packages manager as usual.

GLPI installation itself is composed of three steps:

1. Uncompress the archive in your website;
2. Give your webserver write access to the `files` and `config` directories;
3. *launch installation wizard* (or use the *command line installation script*).

Once these three steps have been completed the application is ready to be used.

If you need to set advanced configuration, like SSL connection parameters, please refer to *advanced configuration*.

2.4 Files and directories locations

Like many other web applications, GLPI can be installed by just copying the whole directory to any web server. However, this may be less secure.

Warning: Every file accessible directly from a web server must be considered unsafe!

GLPI stores some data in the `files` directory, the database access configuration is stored in the `config` directory, etc. Even if GLPI provides some ways to prevent files from being accessed by the webserver directly, best practise is to store data outside of the web root. That way, sensitive files cannot be accessed directly from the web server.

There are a few configuration directives you may use to achieve that (directives that are used in provided downstream packages):

- `GLPI_CONFIG_DIR`: set path to the configuration directory;
- `GLPI_VAR_DIR`: set path to the `files` directory;
- `GLPI_LOG_DIR`: set path to logs files.

Note: There are many other configuration directives available, the ones we talked about are the main to take into account for a more secure installation.

Directories choice is entirely up to you; the following example will follow the [FHS](#) recommendations.

Our GLPI instance will be installed in `/var/www/glpi`, a specific virtual host in the web server configuration will reflect this path.

GLPI configuration will be stored in `/etc/glpi`, just copy the `config` directory to this place. GLPI requires read rights on this directory to work; and write rights during the installation process.

GLPI data will be stored in `/var/lib/glpi`, just copy the `files` directory to this place. GLPI requires read and write rights on this directory.

GLPI logs files will be stored in `/var/log/glpi`, there is nothing to copy here, just create the directory. GLPI requires read and write access on this directory.

Following this instructions, we'll create a `inc/downstream.php` file into GLPI directory with the following contents:

```
<?php
define('GLPI_CONFIG_DIR', '/etc/glpi/');

if (file_exists(GLPI_CONFIG_DIR . '/local_define.php')) {
    require_once GLPI_CONFIG_DIR . '/local_define.php';
}
```

Warning: GLPI packages will certainly provide a `inc/downstream.php` file. This one must not be edited! GLPI looks for a `local_define.php` file in its own `config` directory. If you want to use one form new config directory, you have to load it.

Then, create a file in `/etc/glpi/local_define.php` with the following contents:

```
<?php
define('GLPI_VAR_DIR', '/var/lib/glpi');
define('GLPI_LOG_DIR', '/var/log/glpi');
```

Note: New in version 9.2.2.

For GLPI prior to 9.2.2, the `GLPI_VAR_DIR` constant did not exist and it was required to set all paths separately:

```
<?php
define('GLPI_VAR_DIR', '/var/lib/glpi');
define('GLPI_DOC_DIR', GLPI_VAR_DIR);
define('GLPI_CRON_DIR', GLPI_VAR_DIR . '/_cron');
define('GLPI_DUMP_DIR', GLPI_VAR_DIR . '/_dumps');
define('GLPI_GRAPH_DIR', GLPI_VAR_DIR . '/_graphs');
define('GLPI_LOCK_DIR', GLPI_VAR_DIR . '/_lock');
define('GLPI_PICTURE_DIR', GLPI_VAR_DIR . '/_pictures');
define('GLPI_PLUGIN_DOC_DIR', GLPI_VAR_DIR . '/_plugins');
define('GLPI_RSS_DIR', GLPI_VAR_DIR . '/_rss');
define('GLPI_SESSION_DIR', GLPI_VAR_DIR . '/_sessions');
define('GLPI_TMP_DIR', GLPI_VAR_DIR . '/_tmp');
define('GLPI_UPLOAD_DIR', GLPI_VAR_DIR . '/_uploads');
define('GLPI_CACHE_DIR', GLPI_VAR_DIR . '/_cache');

define('GLPI_LOG_DIR', '/var/log/glpi');
```

Of course, it is always possible to redefine any of those paths `if` needed.

2.5 Post installation

Once GLPI has been installed, you're almost done.

An extra step would be to secure (or remove) installation directory. As an example, you can consider adding the following to your Apache virtual host configuration (or in the `glpi/install/.htaccess` file):

```
<IfModule mod_authz_core.c>
    Require local
</IfModule>
<IfModule !mod_authz_core.c>
    order deny, allow
    deny from all
    allow from 127.0.0.1
    allow from ::1
</IfModule>
ErrorDocument 403 "<p><b>Restricted area.</b><br />Only local access allowed.<br />
↪Check your configuration or contact your administrator.</p>"
```

With this example, the *install* directory access will be limited to localhost only and will display an error message otherwise. Of course, you may have to adapt this to your needs; refer to your web server's documentation.



To begin installation process, point your browser to the GLPI main address: https://\protect\T1\textbraceleftadresse_glpi\protect\T1\textbraceright/

When GLPI is not installed; a step-by-step installation process begins.

3.1 Choose lang (Select your language)

The first step will let you choose the installation language. Select your lang, and click validate.



3.2 License

Usage of GLPI is subject to GNU license approval. Once licensing terms read and accepted, just validate the form.



If you do not agree with licensing terms, it is not possible to continue installation process.

3.3 Install / Update

This screen allows to choose between a fresh GLPI installation or an update.



Click on install.

3.3.1 Environment checks

This step will check if prerequisites are met. If they're not, it is not possible to continue and an explicit error message will tell you about what is wrong and what to do before trying again.

GLPI

GLPI SETUP

Step 0

Checking of the compatibility of your environment with the execution of GLPI

Test done	Results
Testing PHP Parser	✓
Sessions test	✓
Test if Session_use_trans_sid is used	✓
mysqli extension test	✓
ctype extension test	✓
fileinfo extension test	✓
json extension test	✓
mbstring extension test	✓
zlib extension test	✓
curl extension test	✓
gd extension test	✓
simplexml extension test	✓
xml extension test	✓
imap extension test	✓
APCu extension test	✓
xmldrpc extension test	✓
ldap extension test	❗ ldap extension is not present
Zend OPcache extension test	❗ Zend OPcache extension is not present
Allocated memory test	✓
Checking write permissions for setting files	✓
Checking write permissions for document files	✓
Checking write permissions for dump files	✓
Checking write permissions for session files	✓
Checking write permissions for automatic actions files	✓
Checking write permissions for graphic files	✓
Checking write permissions for lock files	✓
Checking write permissions for plugins document files	✓
Checking write permissions for temporary files	✓
Checking write permissions for cache files	✓
Checking write permissions for rss files	✓
Checking write permissions for upload files	✓
Checking write permissions for pictures files	✓
Checking write permissions for log files	✓
SELinux mode is Enforcing	✓
SELinux boolean configuration for httpd_can_network_connect --> on	✓
SELinux boolean configuration for httpd_can_network_connect_db --> on	✓
SELinux boolean configuration for httpd_can_sendmail --> on	✓

Do you want to continue?

Continue Try again

Some prerequisites are optional, it will be possible to continue installation event if they're not met.

3.3.2 Database connection

Database connection parameters are asked.



The screenshot shows the GLPI Setup interface. At the top left is the GLPI logo. The main heading is "GLPI SETUP" in yellow. Below it, "Step 1" and "Database connection setup" are displayed in white. A white form box contains the following fields: "SQL server (MariaDB or MySQL)", "SQL user", and "SQL password", each with an adjacent input field. A yellow "Continue" button is located at the bottom center of the form.

- *MySQL server*: enter the path to your MySQL server, *localhost* or *mysql.domaine.tld* as example;
- *MySQL user*: enter user name that is allowed to connect to the Database;
- *MySQL password*: enter user's password.

Once all fields are properly filled, validate the form.

A first database connection is then established. If parameters are invalid, an error message will be displayed, and you'll have to fix parameters and try again.

3.3.3 Database choice

Once connection to the database server is OK, you have to create or choose the database you want for your GLPI and init it.



There are 2 ways to go:

- use an existing database

Select this database in the displayed list. Validate to use.

Warning: Selected database contents will be destroyed on installation.

- Create a new database

Choose *Create a new database*, enter the database name in the relevant field and then validate to create the base.

Warning: SQL user must be able to create new database for this option to work.

3.3.4 Database initialization

This step initializes the database with default values.



If there is any error; pay attention to the displayed informations.

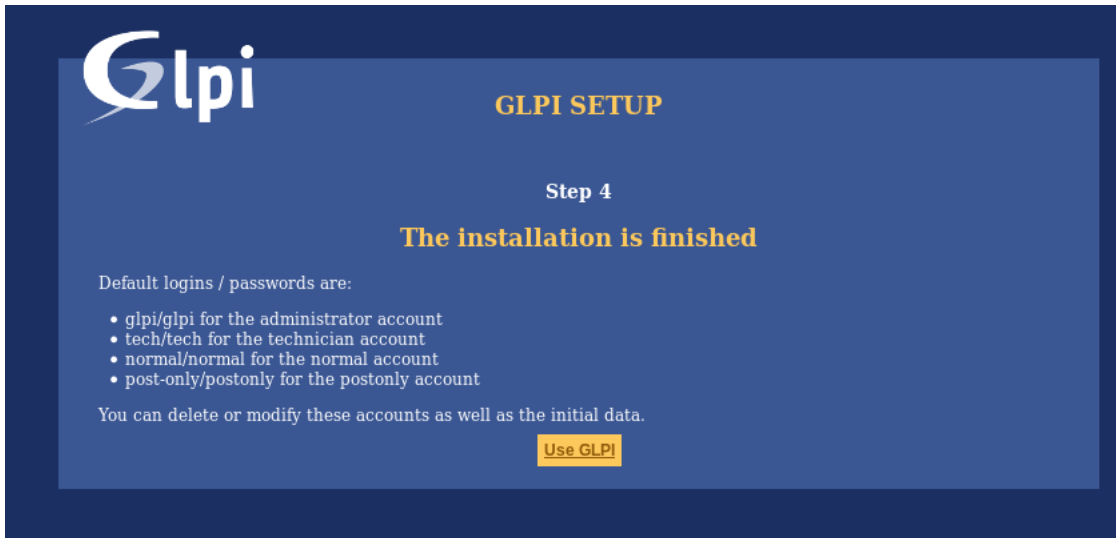
3.3.5 Telemetry informations

GLPI will ask you to share some Telemetry informations and to register. This is not mandatory.



3.3.6 End of installation

This step presents a summary of the installation and give created users list. Please pay attention to those informations and validate to go to the app.



Note: Default user accounts are:

- `glpi/glpi` admin account,
- `tech/tech` technical account,
- `normal/normal` “normal” account,
- `post-only/postonly` post-only account.

Warning: For obvious security concerns, you’ll have to delete or edit those accounts.

Before removing the `glpi` account, please make sure you have created another user with `super-admin` profile.



In order to get timezones working on a MariaDB/MySQL instance, you will have to initialize Timezones data and grant GLPI database user read ACL on their table.

Warning: Enabling timezone support on your MySQL instance may affect other database in the same instance; be carefull!

4.1 Non windows users

On most systems, you'll have to initialize timezones data from your system's timezones:

```
mysql_tzinfo_to_sql /usr/share/zoneinfo | mysql -p -u root mysql
```

You may want to check [MariaDB documentation about mysql_tzinfo_to_sql](#) and your system documentation do know where data are stored (if not in `/usr/share/zoneinfo`).

Do not forget to restart the database server once command is successfull.

4.2 Windows users

Windows does not provide timezones informations, you'll have to download and intialize data yourself.

See [MariaDB documentation about timezones](#).

4.3 Grant access

Warning: Be carefull not to give your GLPI database user too large access. System tables should **never** grant access to app users.

In order to list possible timezones, your GLPI database user must have read access on `mysql.time_zone_name` table. Assuming your user is `glpi@localhost`, you should run something like:

```
GRANT SELECT ON `mysql`.`time_zone_name` TO 'glpi'@'localhost';  
FLUSH PRIVILEGES;
```



Note: As for every update process, you have to backup some data before processing any upgrade:

- **backup your database;**
- backup your files directory;
- backup your configuration.

First, download latest GLPI version and extract files. GLPI update process is then automated. To start it, just go to your GLPI instance URI, or (recommended) use the *command line tools*.

Once a new version will be installed; you will not be able to use the application until a migration has been done.

Please also note the update process will automatically disable your plugins.

Warning: You should not try to restore a database backup on a non empty database (say, a database that has been partially migrated for any reason).

Make sure your database is empty before restoring your backup and try to update, and repeat on fail.

Warning: From the beginning, it was possible to update to any GLPI version from any older version. But this requires a lot of work to maintain such an amount of very, very old code (that can't even be tested!). . .

We've decided to support migrations from GLPI 0.80 only. If you want to update from an older version, you'll need to proceed in two steps: first updating to 9.4.x (even if any version ≥ 0.80 would work, it's maybe a good step to go to latest stable before), then to $\geq 10.x$.



Command line tools

Since GLPI 9.2.2, command line tools are provided as supported scripts and are available from the `scripts` directory of the archive. On previous versions, those scripts were present in the `tools` directory that is not official and therefore not in the release archive.

Since GLPI 9.4.0, command line tools are being centralized in a console application (`bin/console`). Calling `php bin/console` from GLPI directory displays the list of available commands.

Note: If APCu is installed on your system, it may fail from command line since default configuration disables it from command-line. To change that, set `apc.enable_cli` to `on` in your APCu configuration file.

6.1 Console options

For every console command, following options are available:

- `--config-dir=CONFIG-DIR` path of configuration directory to use, relative to current working directory (required only if a custom path is used)
- `-h, --help` displays command help
- `--lang=LANG` output language code (default value is existing GLPI “language” configuration or “en_GB”)
- `-n, --no-interaction` disable command interactive questions
- `--no-plugins` disable GLPI plugins during command execution
- `-q, --quiet` disable command output
- `-v|vv|vvv, --verbose=VERBOSE` verbosity level: 1 for normal output, 2 for more verbose output and 3 for debug

6.2 Install

The `php bin/console db:install` has been made to install GLPI database in CLI mode.

Possible options for this command are:

- `-H, --db-host host name` (*localhost* per default)
- `-P, --db-port database port` (default MySQL port if option is not defined)
- `-d, --db-name database name`
- `-u, --db-user database user name`
- `-p, --db-password database user's password`
- `-L, --default-language default language of GLPI` (*en_GB* per default)
- `-f, --force do not check if GLPI is already installed and drop what would exists`

If mandatory options are not specified in the command call, the console will ask for them.

See also *console options*.

6.3 Update

The `php bin/console db:update` has been made to update GLPI database in CLI mode from a previously installed version.

There is no required arguments, just run the command so it updates your database automatically.

Warning: Do not forget to backup your database before any update try!

Possible options for this command are:

- `-u, --allow-unstable allow update to an unstable version` (use it with cautions)
- `-f, --force force execution of update from v-1 version of GLPI even if schema did not changed`

See also *console options*.

6.4 Database tools

6.4.1 Database schema check

The `php bin/console db:check` command can be used to check if your database schema differs from expected one.

If you have any diff, output will look like :

```
$ php bin/console glpi:database:check
Table schema differs for table "glpi_rulecriterias".
--- Original
+++ New
@@ @@
 create table `glpi_rulecriterias` (
```

(continues on next page)

(continued from previous page)

```

`id` int(11) not null auto_increment
`rules_id` int(11) not null default '0'
`criteria` varchar(255) default null
`condition` int(11) not null default '0'
- `pattern` text default null
+ `pattern` text
  primary key (`id`)

```

6.5 Tasks tools

6.5.1 Task unlock

The `php bin/console task:unlock` command can be used to unlock stucked cron tasks.

Warning: Keep in mind that no task should be stucked except in case of a bug or a system failure (database failure during cron execution for example).

Possible options for this command are:

- `-a, --all` unlock all tasks
- `-c, --cycle[=CYCLE]` execution time (in cycles) from which the task is considered as stuck (delay = task frequency * cycle)
- `-d, --delay[=DELAY]` execution time (in seconds) from which the task is considered as stuck (default: 1800)
- `-t, --task[=TASK]` itemtype::name of task to unlock (e.g: MailCollector::mailgate)

See also *console options*.

6.6 Migration tools

6.6.1 From MyISAM to InnoDB

New in version 9.3.0.

Since version 9.3.0, GLPI uses the InnoDB engine instead of previously used MyISAM engine.

The `php bin/console glpi:migration:myisam_to_innodb` command can be used to migrate exiting tables to InnoDB engine.

6.6.2 Missing timestamps builder

New in version 9.1.0.

Prior to GLPI 9.1.0, fields corresponding to creation and modification dates were not existing.

The `php bin/console glpi:migration:build_missing_timestamps` command can be used to rebuild missing values using available logs.



7.1 SSL connection to database

New in version 9.5.0.

Once installation is done, you can update the `config/config_db.php` to define SSL connection parameters. Available parameters corresponds to parameters used by `mysqli::ssl_set()` <<https://www.php.net/manual/en/mysqli.ssl-set.php>>:

- `$dbssl` defines if connection should use SSL (*false* per default)
- `$dbsslkey` path name to the key file (*null* per default)
- `$dbsslcert` path name to the certificate file (*null* per default)
- `$dbsslca` path name to the certificate authority file (*null* per default)
- `$dbsslcapath` pathname to a directory that contains trusted SSL CA certificates in PEM format (*null* per default)
- `$dbsslcipher` list of allowable ciphers to use for SSL encryption (*null* per default)

Warning: For now it is not possible to define SSL connection parameters prior or during the installation process. It has to be done once installation has been done.

