

Taiga Documentation

Table of Contents

1. Introduction	1
2. How it Works	1
3. Requirements	1
4. Installation Guide	2
4.1. Setup Production Environment	2
4.2. Setup Development Environment	2
4.3. Other Ways to Setup an Initial Environment	2
4.4. Setup FAQs and common Bugs	2
4.5. Upgrades	3
5. How to Contribute	3
5.1. Reporting Bugs	3
6. API	3
7. Webhooks	4
8. Contrib Plugins	4
9. Themes	4
10. Community	4
11. License	5

1. Introduction

Free. Open Source. Powerful. *Taiga* is a project management platform for startups and agile developers & designers who want a simple, beautiful tool that makes work truly enjoyable.

2. How it Works

Taiga is made up of two components: [taiga-front](#) and [taiga-back](#)

- **taiga-front** has the frontend related code built with **angularjs** and **coffeescript**. It can run on any static file server like nginx.
- **taiga-back** has the backend related code consisting of an API, built using **django** and **python3**.

Taiga has some other modules, but they are optional and documented in other sections.

3. Requirements

This is a list of major *Taiga* dependencies:

Runtime:

- Python ≥ 3.4
- PostgreSQL ≥ 9.4
- RabbitMQ (optional as long as you don't want async notifications)

Compile time:

- GCC & Development Headers
- Ruby ≥ 2.1 (only for compiling sass)
- NodeJS ≥ 7.0 (with npm and gulp for downloading dependencies and compiling coffeescript)

Hardware

- The main Taiga hardware restriction is having **at least 0.75 GB of RAM**. When installing the python requirements, the package lxml will use gcc to build itself and fail if there is not enough memory.
- Taiga code doesn't need a lot of disc space. Only the database or the media directory might need more space with the passage of time.

4. Installation Guide

This section details everything you need to know to get *Taiga* up and running in both development and production environments.

4.1. Setup Production Environment

[This document contains the complete, step by step tutorial](#) of how to properly setup *Taiga* for a low traffic production environment.

4.2. Setup Development Environment

You can view the complete documentation on how to [setup a proper development environment](#).

4.3. Other Ways to Setup an Initial Environment

In addition to the traditional step-by-step manuals, *Taiga* comes with a collection of provisioning scripts and Vagrant support, and [complete documentation of how to use them](#).

4.4. Setup FAQs and common Bugs

You can view a list of common FAQs and bugs related to the setup process [FAQs and common bugs](#).

4.5. Upgrades

Instructions on [how to upgrade your Taiga instance](#).

5. How to Contribute

Because *Taiga* is an open source project, we gratefully accept contributions in various forms.

Examples of contributions include:

- [Bug reports](#)
- Code patches
- Improvements
- Patch reviews
- [Translate](#)
- [Create a theme](#)
- [Code contrib plugins](#)

Before starting to develop a big feature (intended to be included on the *Taiga* code base), it is strongly recommended that you reach out to us on the [mailing list](#).

5.1. Reporting Bugs

Before reporting a bug or requesting a new feature, please consider these general points:

- Check that someone hasn't already filed the bug or feature request by searching, or running custom queries in, the [issue tracker](#).
- Check the right repository to use for reporting an issue: bugs related to UI should go to **taiga-front**, but if something is broken on the backend, an issue should be opened on the **taiga-back** repository.
- Ask on the [mailing list](#) first if you're not sure if what you're seeing is a bug.
- Don't use the GitHub issues system to ask support questions. Use the [mailing list](#) for that.
- Don't use GitHub issues for lengthy discussions, because they're likely to get lost. If a particular ticket is controversial, please move the discussion to the [mailing list](#).

6. API

Taiga has a complete REST API (the one used by the web app), check the full API documentation [here](#).

7. Webhooks

Taiga has a webhooks notification system, check the full documentation [here](#).

8. Contrib Plugins

Taiga supports the inclusion of contrib plugins, each plugin has its own documentation and repository. The currently supported plugins are:

- [taiga-github-auth](#): GitHub Auth plugin
- [taiga-contrib-hipchat](#): HipChat integration plugin
- [taiga-contrib-letschat](#): Lets Chat integration plugin
- [taiga-contrib-slack](#): Slack integration plugin

9. Themes

You can create custom themes in Taiga.

To create a theme you have to create a new folder in `/app/themes/` with your theme name, or duplicate a previous one. If you create a new folder, then create two files inside.

- `variables.scss` - to override sass variables, fonts, mixins helpers..etc.
- `custom.scss` - to override any defined Taiga style for any module or layout.

Remember, the file `variables.scss` should only contain sass declarations that do not generate code by themselves (like variables or mixins).

If you need to override a style and generate some new CSS, you should do it in `custom.scss`.

Please, check the default Taiga theme to understand how these files work properly.

When the files are created you have to run gulp with the theme name. This is the same as `gulp default` but uses your theme instead of the default taiga theme.

```
gulp --theme themeName
```

If you want to compile all themes you should run.

```
gulp compile-themes
```

10. Community

[Taiga has a mailing list](#). Feel free to join it and ask any questions you may have.

To subscribe for announcements of releases, important changes and so on, please follow [@taigaio](#) on Twitter or read our [blog](#).

11. License

Every code patch accepted in the Taiga codebase is accepted under the AGPL v3.0 license. It is important that you do not include any code which cannot be licensed under AGPL v3.0.

You can see the complete license in the **LICENSE** file in the root of the repository.