

Taiga: Setup FAQs and common bugs

Table of Contents

1. FAQs.....	1
1.1. How can I change the domain name?	1
1.2. How can I disable registration?.....	1
1.3. How can I restrict registration to a concrete domain?	2
1.4. How can I restart the backend application?	2
1.5. How I can keep my app up to date with the most recent Taiga version?	2
1.6. How I can add extra functionality to Taiga?	2
1.7. How I can make a backup of the data? And then restore them?	3
2. Common Bugs	3
2.1. UnicodeEncodeError	3
3. Importers	4
3.1. How can I configure Trello importers in my own instance?	4
3.2. How can I configure Github importers in my own instance?	5
3.3. How can I configure Asana importers in my own instance?.....	5
3.4. How can I configure Jira importers in my own instance?	6

1. FAQs

1.1. How can I change the domain name?

The domain name mainly affects the frontend application, because it needs to communicate with the backend through the domain/public-ip.

To do this you should update the url value `api` on frontend config file `dist/conf/conf.json`. Also you should update the domain related configuration on the backend settings file: `settings/local.py`.

And finally reload the backend config with: `circusctl reload taiga`

1.2. How can I disable registration?

You can disable public registration for external user. To do this you should set `to False` in `settings/local.py` and `disable` too in front, in `dist/conf.json` set `ro` to `false`.

```
PUBLIC_REGISTER_ENABLED = False
```

```
{
  (...)
  "publicRegisterEnabled": false,
  (...)
}
```

If you disable it, only invited users can create accounts in the instance.

1.3. How can I restrict registration to a concrete domain?

You can restrict the usage of the platform to a set of valid email domains, so this allows you to restrict the registration to users that have a valid company email account.

To do this you should update the `USER_EMAIL_ALLOWED_DOMAINS` on your `settings/local.py` and add the list of valid domains, for example, if you want to allow registrations only from `mycompany.com` and `mycustomer.com`, you can set it as:

```
USER_EMAIL_ALLOWED_DOMAINS = ["mycompany.com", "mycustomer.com"]
```

And finally reload the backend config with: `circusctl reload taiga`

1.4. How can I restart the backend application?

The Backend application is running under circus. To restart any application running with circus use the `circusctl` command:

```
circusctl restart taiga
```

In case taiga-celery is configured restart it too:

```
circusctl restart taiga-celery
```

1.5. How I can keep my app up to date with the most recent Taiga version?

[Check the upgrades instructions.](#)

1.6. How I can add extra functionality to Taiga?

Taiga allows adding functionality through contrib plugins. The installation of each plugin depends on the plugin itself and must be documented in the plugin repository.

The current supported plugins are:

- [taiga-contrib-cookie-warning](#) A funny cookie warning message
- [taiga-contrib-github-auth](#): GitHub Authentication
- [taiga-contrib-gogs](#): Gogs integration
- [taiga-contrib-hipchat](#): HipChat integration
- [taiga-contrib-letschat](#): Let's Chat integration
- [taiga-contrib-slack](#): Slack integration

1.7. How I can make a backup of the data? And then restore them?

To backup data from your server you just need to store a copy of the database and attachments files. That is, You just need to make a dump of the database and a copy of `taiga-back/media` directory.

1. Create dump:

```
pg_dump taiga > taiga.sql
```

2. Copy media to a `<dest>`:

```
cp -r taiga-back/media <dest>
```

Then You can restore the backup creating a new database to load the dump and copying the media directory:

```
createdb taiga  
psql taiga < taiga.sql  
cp -r <dest> taiga-back/media
```

2. Common Bugs

2.1. UnicodeEncodeError

If you have some error like this:

```
File "/home/taiga/taiga-back/taiga/projects/services/invitations.py", line 17, in
send_invitation
    email.send()
File "/home/taiga/.virtualenvs/taiga/lib/python3.4/site-
packages/django/core/mail/message.py", line 286, in send
    return self.get_connection(fail_silently).send_messages([self])
File "/home/taiga/.virtualenvs/taiga/lib/python3.4/site-
packages/django/core/mail/backends/console.py", line 36, in send_messages
    self.write_message(message)
File "/home/taiga/.virtualenvs/taiga/lib/python3.4/site-
packages/django/core/mail/backends/console.py", line 23, in write_message
    self.stream.write('%s\n' % msg_data)
UnicodeEncodeError: 'ascii' codec can't encode character '\xfc' in position 487:
ordinal not in range(128)
```

You have to change the locale of your system to someone that support UTF8 characters. To fix it add to the taiga user `.bashrc` file this lines:

if you have a German Linux machine, you can fix the problem by exporting the locale to `de_DE.UTF-8`:

```
export LANG=de_DE.UTF-8
export LC_ALL=de_DE.UTF-8
export LC_TYPE=de_DE.UTF-8
```

If you are on a US machine, `en_US.utf-8` is the encoding of choice. On some newer Linux systems, you could also try `en_US.UTF-8` as the locale:

```
export LANG=en_US.UTF-8
export LC_ALL=en_US.UTF-8
export LC_TYPE=en_US.UTF-8
```

Restart the shell, or the machine, and try again.

3. Importers

You can configure importers for Jira, Github, Asana and Trello in your own instance.

3.1. How can I configure Trello importers in my own instance?

For configure Trello, you have to go to <https://trello.com/app-key> and obtaining your API key and your oauth secret, and then, configure it in your settings/local.py:

```
IMPORTERS["trello"] = {
    "active": True,
    "api_key": "your-trello-api-key",
    "secret_key": "your-trello-oauth-secret"
}
```

And you have to enable in your dist/conf.json in taiga-front adding it to importers list:

```
{
    ...
    importers: ["trello", ...]
    ...
}
```

3.2. How can I configure Github importers in my own instance?

To configure Github, you have to go to <https://github.com/settings/developers>, register a new application and obtain a client id and a client secret. When creating the GitHub OAuth app the "Authorization callback URL" should be set to the base url for your Taiga instance. After you have created the app you have to configure Taiga for it in your settings/local.py:

```
IMPORTERS["github"] = {
    "active": True,
    "client_id": "your-github-client-id",
    "client_secret": "your-github-client-secret"
}
```

And you have to enable in your dist/conf.json in taiga-front adding it to importers list:

```
{
    ...
    importers: ["github", ...]
    ...
}
```

3.3. How can I configure Asana importers in my own instance?

For configure Asana, you have to go to your asana account settings, in Apps section, and click on "Manage Developer Apps", then you have to "Add new application", ensure that the callback url match with your settings, and the url of your service with the path /project/new/import/asana. After that, you can get your client id and your client secret and configure it in your settings/local.py:

```
IMPORTERS["asana"] = {
    "active": True,
    "callback_url": "{}://{}project/new/import/asana".format(SITES["front"]
"scheme"],
                                                    SITES["front"]
"domain")),
    "app_id": "your-asana-client-id",
    "app_secret": "your-asana-client-secret"
}
```

And you have to enable in your dist/conf.json in taiga-front adding it to importers list:

```
{
    ...
    importers: ["asana", ...]
    ...
}
```

3.4. How can I configure Jira importers in my own instance?

For configure Jira, you have to go to generate a public of private/public keys pair. To do this, you can run the next command:

```
openssl genpkey -algorithm RSA -out private_key.pem -pkeyopt rsa_keygen_bits:2048
openssl rsa -pubout -in private_key.pem -out public_key.pem
```

Once you have the public and private key, you can configure it in your settings/local.py:

```
IMPORTERS["jira"] = {
    "active": True,
    "consumer_key": "one-key-at-your-election",
    "cert": file('private_key.pem').read(), # You can directly copy and paste the
content here
    "pub_cert": file('public_key.pem').read(), # You can directly copy and paste the
content here
}
```

And you have to enable in your dist/conf.json in taiga-front adding it to importers list:

```
{  
  ...  
  importers: ["jira", ...]  
  ...  
}
```

After that, you have to configure your Jira instance to allow to connect using this certificates, you can get more information about it in <https://tree.taiga.io/support/>.