

Teammer
Administrator's guide

Ivan Afonichkin	Nikita Kuts
ivanafonichkin@gmail.com	chaozik1337@gmail.com

May 22, 2016

Contents

1	License	2
2	Before you begin	3
	2.1 What is Teammer	3
	2.2 How to use this guide	3
3	Installing	4
4	Admin's web interface	5
5	Teammer's architecture	7
6	Modifying Teammer	8
	6.1 Templates	8
	6.2 Views	8
	6.3 Urls	8
	6.4 Models	9
7	Where to get help?	10

1 License

Teammer is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, version 3.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the [GNU General Public License](#) for more details.

2 Before you begin

2.1 What is Teammer

Teammer is an education platform that allows people join groups, interact with each other and, finally, learn new things!

2.2 How to use this guide

This guide is for Teammer administrators and webmasters responsible for installing, configuring, maintaining Teammer and it's realeated applications.

This guide is meant to help you set up Teammer quickly and easily.

We assume that you're familar with Django framework, if not you better learn primitives of it before starting the work.

3 Installing

You will need:

1. Django 1.9.4
2. PostgreSQL 9.4

During development all of us used UNIX-like systems, but if you're using Windows, you will still be able to get the work done, but we still recommend installing Linux (for example, it won't be that easy to set up virtualenv on Windows).

To start, get all the source code from [here](#).

Now we have something to work with, but we are still missing plugins. You will need one called 'psycopg2', it connects your python code with PostgreSQL-driven DB. You can get it from 'pip' very easily.

Now you should find a file called 'manage.py' in Terminal (cmd if you're on Windows). Run 'python manage.py runserver'. It will give you at least one error: "Can't find database XXX". Reasonable. There's two ways out:

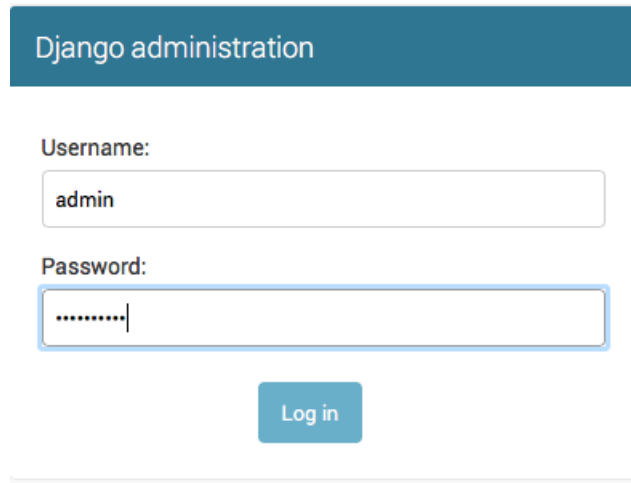
1. Create your own database (using psql) and set up your 'setting.py' file according to it.
2. Use database dump that comes from git and import it, but you will still need to change administrator username/password in 'settings.py'.

Now, you can try once again: 'python manage.py runserver'. If it still gives you errors it's because you made something wrong on previous step or you are still missing some needed packages. Just read your error-messages and 'pip install' them.

If you can run your server successfully now (just go to localhost:8000) - Congrats!

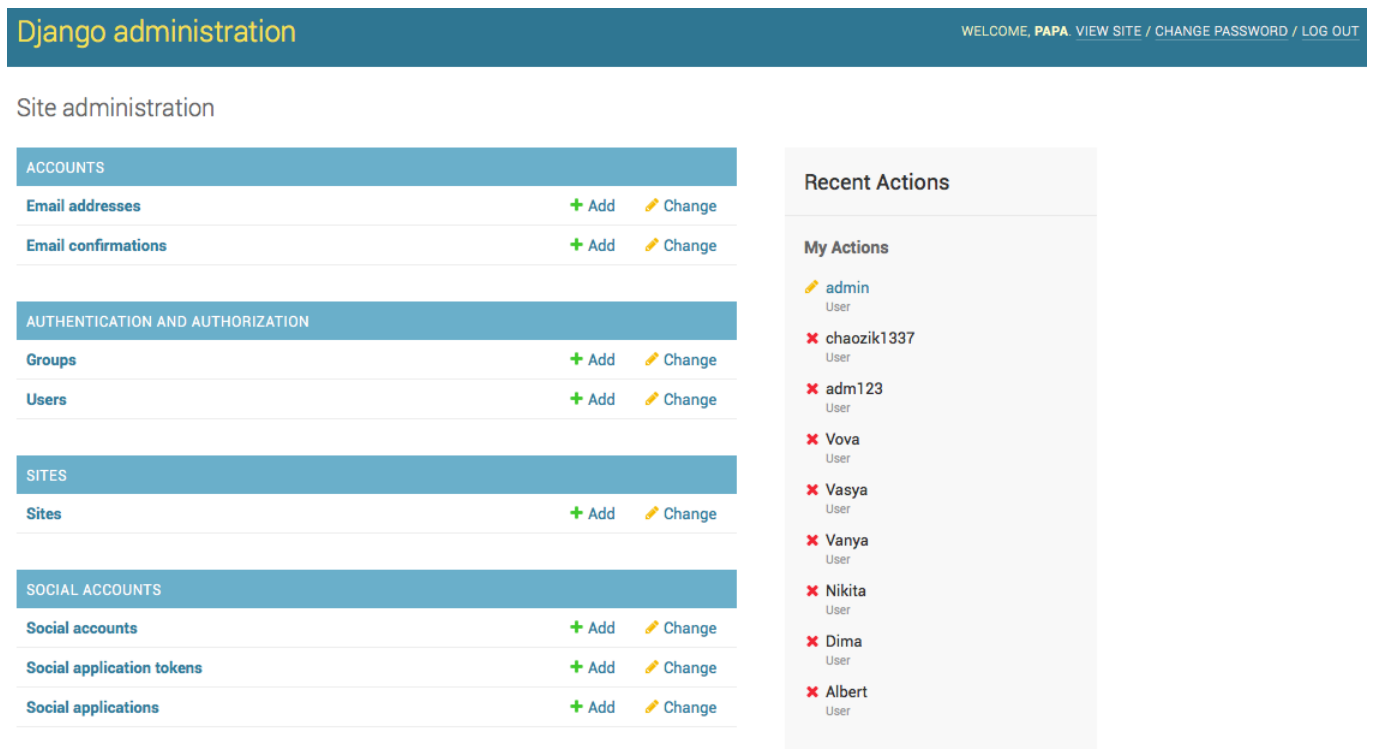
4 Admin's web interface

Go to the 'localhost:8000/admin' and you will see login dialogue:



The image shows the Django administration login page. It has a dark blue header with the text 'Django administration'. Below the header, there are two input fields: 'Username:' with the value 'admin' and 'Password:' with masked characters. A blue 'Log in' button is positioned below the password field.

After you login into it, you will be able to manage Teammer using graphical interface.



The image shows the Django administration dashboard after a successful login. The top header is dark blue with 'Django administration' on the left and 'WELCOME, PAPA VIEW SITE / CHANGE PASSWORD / LOG OUT' on the right. Below the header, the page is titled 'Site administration'. The main content area is divided into two columns. The left column contains four sections: 'ACCOUNTS' with 'Email addresses' and 'Email confirmations'; 'AUTHENTICATION AND AUTHORIZATION' with 'Groups' and 'Users'; 'SITES' with 'Sites'; and 'SOCIAL ACCOUNTS' with 'Social accounts', 'Social application tokens', and 'Social applications'. Each item in these sections has a green '+ Add' link and a yellow pencil 'Change' link. The right column is titled 'Recent Actions' and contains a list of actions under the heading 'My Actions'. The first action is 'admin User' with a yellow pencil icon. The following seven actions are marked with a red 'X' icon and are: 'chaozik1337 User', 'adm123 User', 'Vova User', 'Vasya User', 'Vanya User', 'Nikita User', and 'Dima User'. The last action is 'Albert User' without an icon.

Change user

Username:	<input type="text" value="Bodyan"/>
<small>Required. 30 characters or fewer. Letters, digits and @/./+/-/_ only.</small>	
Password:	algorithm: pbkdf2_sha256 iterations: 24000 salt: xxulzT***** hash: 5IQ16R*****
<small>Raw passwords are not stored, so there is no way to see this user's password, but you can change the password using this form.</small>	

Personal info

First name:	<input type="text"/>
Last name:	<input type="text"/>
Email address:	<input type="text" value="bodyan@mail.ru"/>

Permissions

☒ **Active**
Designates whether this user should be treated as active. Unselect this instead of deleting accounts.

☐ **Staff status**
Designates whether the user can log into this admin site.

☐ **Superuser status**
Designates that this user has all permissions without explicitly assigning them.

This interface gives you power in managing users (you can block them, manage their data, etc). Have to say, it's pretty intuitive.

5 Teammer's architecture

Teammer is built using Django framework, thus it has 'typical' structure for any Django built application. MVC pattern and so on.

If you see your project folder contains another folder called 'templates_dir/'. Inside of it you can found all the templates, that are rendered while you surfing Teammer pages. We do not use template inheritance.

Teammer is an application, that contains another application called 'backend'. You can work with it directly using Terminal (django feature). And 'backend' application is the brain of whole 'Teammer' it handles a lot and there's no need for any other applications inside Teammer at the moment.

Our user structure is pretty clear: we use basic Django class called 'User' and we extend it with one of our models ('UserInfo') to store additional info. We will touch other models later.

Basically, that's all about architecture. Specific things going to be discussed next.

6 Modifying Teammer

6.1 Templates

If you want to change structure of a template you go to ‘template/dir’ and do it! But you have to remember just a few things:

1. Don’t forget that static imports for HTML-template are stored inside ‘static/’ folder
2. All the imports have ‘Django-syntax’. Read django docs if you are not familiar with it.
3. Our application uses ‘csrf tokens’ for every page, if you delete it, something will go wrong. Tokens are needed to make Teammer safer.
4. You will find views names on HTML pages, so if you are going to edit, you better do the same to view first.

6.2 Views

Go to ‘views.py’ inside of ‘backend’ application. Views are the places where all the database interaction takes place and templates are getting filled with specific info before user will see them rendered.

1. Data that is going to be passed to template is stored in dictionary. We call it ‘args’.
2. Before trying to ‘read’ users input from some form do not forget to validate ‘csrf-token’.
3. We use ‘render_to_response’ function to render the template.
4. We stick to ORM instead of writing down queries and you should do the same.

6.3 Urls

Go to ‘urls.py’ inside of ‘backend’ application. Urls file defines the way user can access any view. Nothing special, just Django protocol.

We prefer sending things such as project ids right in the address, but not as additional argument.

Example:

```
url(r'^manage_project/(?P<project_id>[0-9]+)/$', manage_project)
```

It makes things clear for everyone. If you want to map view to address - add urlpattern and connect it to view, you just wrote.

6.4 Models

Django gives us the way to interact with DBMS not by writing queries, but by using ORM. And we use that. Every model maps to unique table in PostgreSQL.

Every model change requires server restart, so don't do it in case you are not sure if you need it. All the changes (after they're saved) are mapped to your database. You will need 'makemigrations' to proceed.

Be careful with this part, if you delete one line, it will surely crash the whole thing.

7 Where to get help?

Now you're ready to configurate and administrate Teammer! If you still have some issues or you just want to contact the creators, you can always write us a message (Contact information in the title of this document).