

User Documentation

This documentation is for the administrator/chairperson.

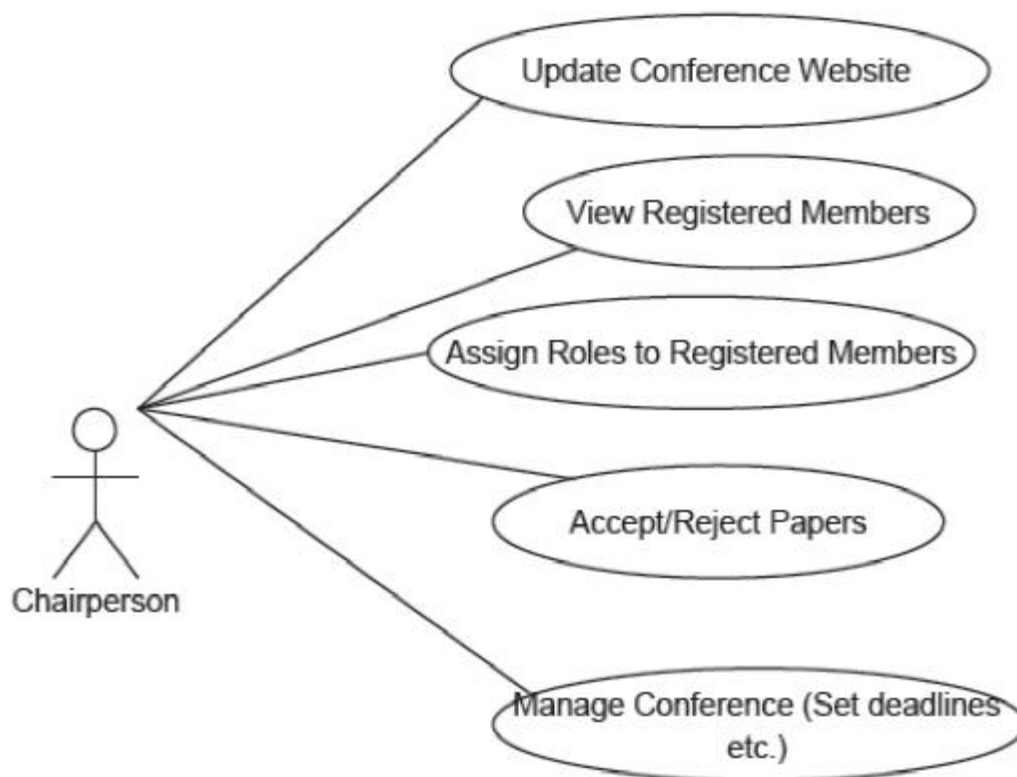
For a detailed user documentation for reviewer, author, delegate and committee member please go to the website. (Which can be found at /help.html).

For general help using Django server go to: <https://docs.djangoproject.com/en/1.8/>

For a complete breakdown of the server and all the functions packages used go to: /admin/doc

Contents:

1. Signing In
2. Navigating to the administration-backend
3. Understanding the tables
4. Editing the website
5. Assigning Permissions
6. Editing tables
7. Accepting/Rejecting papers
8. Signing out
9. Frequently Asked Questions



Signing In

A username and password has been pre-assigned for your convenience.

This can be changed at the settings.py file in the conference directory.

Username: admin

Password: password

Email: admin@conference.com

(Important! Do not share the above credentials with anyone, as it may lead to unauthorized and disastrous effects.)

There are two methods for signing in:

1. Via the “Log In” link on the main page.
2. Via the backend on /admin

Via the Log In:

Assuming the server is running (i.e. python manage.py runserver) you will be taken to the home page of your website. This is the first view that any user will be directed to upon visiting the site. On the right hand side there a navigation bar with links to various views (pages) of the site.

Click on “Log in” and you should be directed to a form prompting you for the above-mentioned credentials.

You should be logged in successfully.

Via the Admin page:

Alternatively, you may decide to browse straight to the backend log in by adding /admin to the URL link. (URL link is the http:// address at the top of your browser).

There you shall be prompted for a similar form asking for your username and password.

Navigating to the Admin-Backend.

To get to the admin backend where all of the permissions and editing will be done you will have to manually add the /admin to the URL address of the homepage. So if your server host is www.example.com, then adding www.example.com/admin will take you to the backend. You will be prompted for username and password if you are not logged in.

Understanding the tables

At the backend the admin interface will be comprised of the databases which contain all the relevant information stored. Clicking on an item will display its table.

Select user to change

Action: 1 of 3 selected

<input type="checkbox"/>	Username	E-mail address	First name	Last name	Staff status
<input type="checkbox"/>	adrian	adrian@example.com	Adrian	Holovaty	⊖
<input checked="" type="checkbox"/>	jacob	jacob@example.com	Jacob	Kaplan-Moss	✓
<input type="checkbox"/>	simon	simon@example.com	Simon	Willison	⊖

3 users

Above is an example of how the table will look once you click on one of them. There is a box on the right hand side that will show you any changes you've made. Remember to be certain about the change you're going to make, because this data cannot be returned or undone.

Select article to change

Action: ☒ 2 of 5 selected

☒ Delete selected articles
☒ Mark selected stories as published

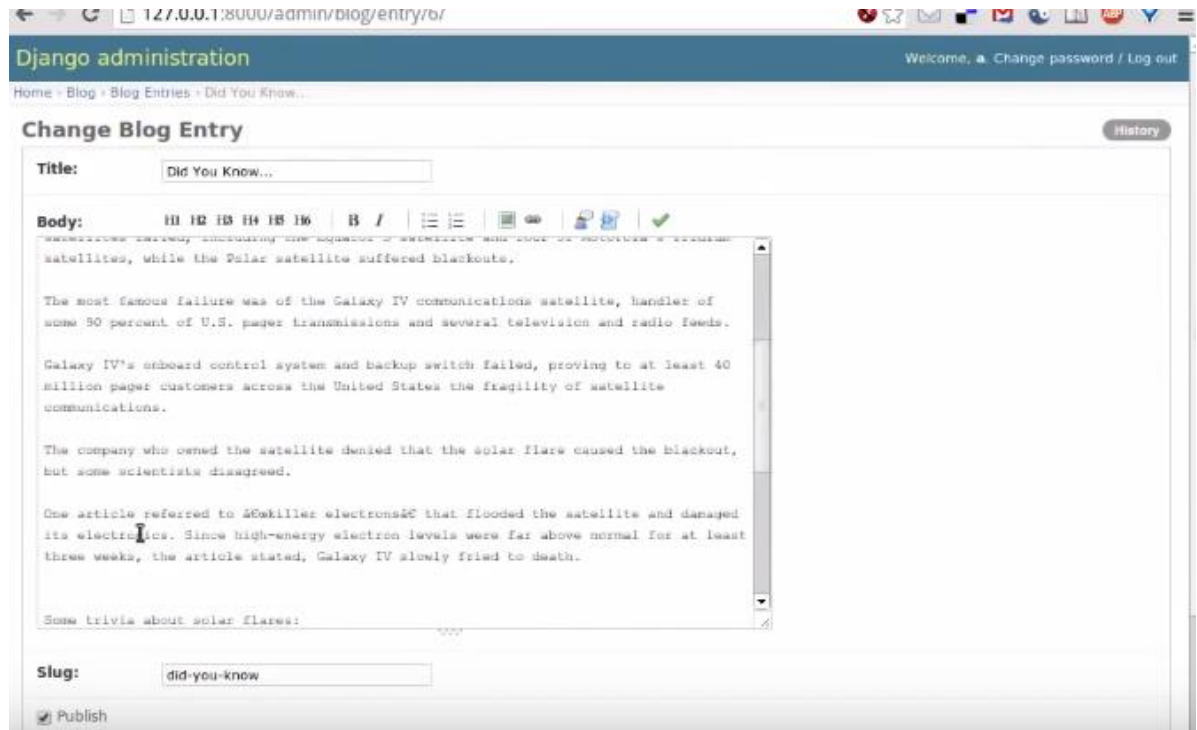
<input type="checkbox"/>	Title	Status
<input type="checkbox"/>	An Exercise in Species Barcoding	Published
<input checked="" type="checkbox"/>	Django 1.4 released	Draft
<input type="checkbox"/>	Example Headlines Considered Harmful	Published
<input checked="" type="checkbox"/>	Global is the new private	Draft
<input type="checkbox"/>	Man lands on Mars	Withdrawn

5 articles

By selecting a row and going to action you can select whether to edit or delete etc.

Editing the website

Using the markdown to add new article entries and remove old ones. This is done on the backend under blog (posts).



The screenshot shows the Django administration interface for editing a blog entry. The browser address bar shows the URL `127.0.0.1:8000/admin/blog/entry/6/`. The page title is "Django administration" with a "Welcome, [user] Change password / Log out" link. The breadcrumb trail is "Home > Blog > Blog Entries > Did You Know...". The main heading is "Change Blog Entry" with a "History" button. The form includes a "Title:" field with the value "Did You Know...", a "Body:" text area with a rich text editor toolbar, and a "Slug:" field with the value "did-you-know". A "Publish" checkbox is at the bottom left.

Fill in the new title. Then proceed to edit the body with the new content.

Slug is the blog term for keyword (tag). Please add a tag, which is a category.

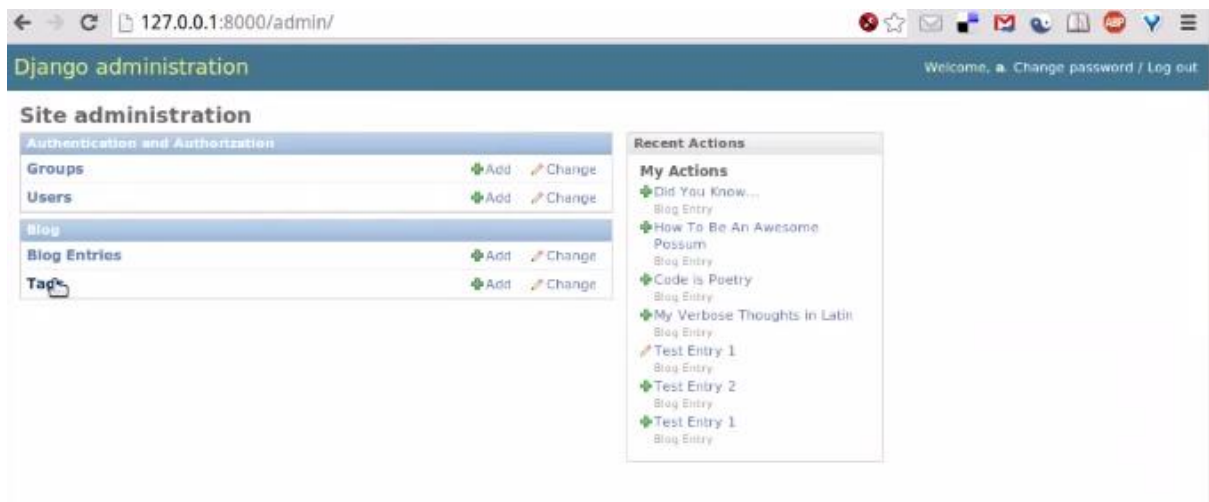
Save changes. Select publish checkbox will add to the website homepage.



The screenshot shows the Django administration interface with a success message: "2 stories were successfully marked as published." Below this is the heading "Select article to change". There is an "Action:" dropdown menu with a "Go" button and a "0 of 5 selected" indicator. A table lists three articles:

<input type="checkbox"/>	Title
<input type="checkbox"/>	An Exercise in Species Barcoding
<input type="checkbox"/>	Django 1.4 released

The green checkbox shows success.



To edit the website first go to Blog Entries. Pictured above. There you can fill in the article title and body and publish it.

Assigning Permissions

Permissions can be found under the myuser table in the backend database. Go into the table.

Action: <input type="text"/> Go 0 of 4 selected						
<input type="checkbox"/>	Username	Email address	Is admin	Is an Author	Is a Reviewer	Is a Committee Member
<input type="checkbox"/>	admin	admin@conference.com	✓	✓	✓	✓
<input type="checkbox"/>	Aron	aron@sun.ac.za	✗	✓	✓	✗
<input type="checkbox"/>	Richyrich	gates@msoft.com	✗	✓	✓	✗
<input type="checkbox"/>	Tiaan	tiaan@gmail.com	✗	✓	✗	✗
4 my users						

Select the user whose permissions you would like to edit. You will be taken to their profile. Once here go to the roles attribute, and select the appropriate one.

As so:

Permissions
<input type="checkbox"/> Is admin
<input checked="" type="checkbox"/> Is an Author
<input checked="" type="checkbox"/> Is a Reviewer
<input type="checkbox"/> Is a Committee Member
<input type="button" value="Delete"/> <input type="button" value="Save and add another"/> <input type="button" value="Save and continue editing"/> <input type="button" value="Save"/>

Editing tables

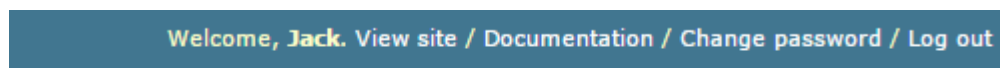
It is not recommended to add or change table content at the backend. However, should you need to manually edit a user or a delegate you may do so by clicking the Add or change button on the site admin page.



Accepting/Rejecting papers

As chairperson you will have to accept or reject a paper once the committee has reviewed the scores. Go to Papers under the papers box in admin. There you can edit the specific paper by selecting accept or reject.

Signing out



Above is the navigation on the top of admin page. Here you may decide to log out. There are other options available such as viewing the documentation (Similar to this PDF version but specific to content on the page.) or change your password.

You can also log out on the homepage of the website, like a regular user.

Frequently Asked Questions?

I can't log in. When I enter a valid username and password, it just brings up the login page again, with no error messages.

Try the following two things? It is likely the cookie is not set up correctly.

Set the `SESSION_COOKIE_DOMAIN` setting in your admin config file to match your domain. For example, if you're going to "<http://www.example.com/admin/>" in your browser, in "ProjectX.settings" you should set `SESSION_COOKIE_DOMAIN = 'www.example.com'`.

I can't log in. When I enter a valid username and password, it brings up the login page again, with a "Please enter a correct username and password" error.

The admin site only allows access to users with `is_active` and `is_staff` set to `True`. You may have changed it accidentally.

.How can I customize the functionality of the admin interface?

You've got several options. If you want to piggyback on top of an add/change form that Django automatically generates, you can attach arbitrary Javascript modules to the page via the model's class `Admin` `is_parameter`. That parameter is a list of URLs, as strings, pointing to Javascript modules that will be included within the admin form via a `<script>` tag.

If you want more flexibility than simply tweaking the auto-generated forms, feel free to write custom views for the admin. The admin is powered by Django itself, and you can write custom views that hook into the authentication system, check permissions and do whatever else they need to do.

If you want to customize the look-and-feel of the admin interface, read the next question.

You can modify the admin site's presentation by editing the CSS stylesheet and/or associated image files. The site is built using semantic HTML and plenty of CSS hooks, so any changes you'd like to make should be possible by editing the stylesheet.

What browsers are supported for using the admin?

The admin provides a fully-functional experience to all browsers, with the notable exception of IE6, which is not supported.

There may be minor stylistic differences between supported browsers—for example, some browsers may not support rounded corners. These are considered acceptable variations in rendering.

Regards, Dev team. (Group 1D?E) Tiaan, Aron, Lujane, Marco