django-public-project Documentation

Release 0.7-beta

Holger Drewes

Mar 16, 2017

Contents

Django Public Project (DPP) is an open source software (Website | GitHub) for **creating/maintaining an information website around a public project** (building of a new opera house or an airport, long-term preparations for a big sports event,...) bringing transparency to the roles of stakeholders, important events as well as the overall project progress.

DPP was initially funded by the pirate party in the german parliaments of Berlin and North Rhine-Westphalia for their projects BERwatch (german) and BLBwatch (german).

Note: There is now a **Live-Demo installation** to test the software: Website | Admin (User: `dpp` / `demo`)

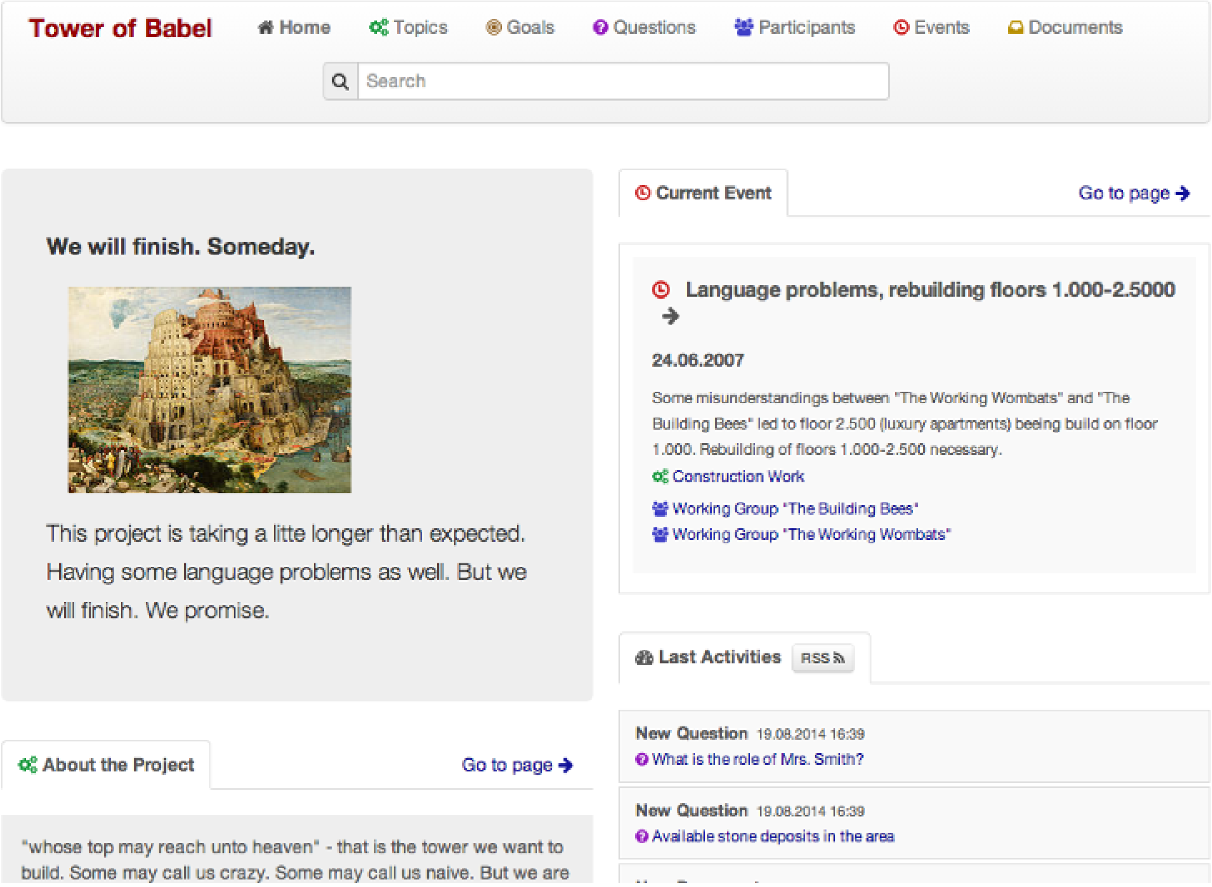
CHAPTER 1

Manual

Getting Started

Introduction

Django Public Project (DPP) can be best described as a **special kind of Content Management System (CMS)**, tailored to the peculiarities of big projects, which often tend to becoming fairely complex and intrasparent when they evolve. Below is a screenshot of the front-end view of the example project:



We will finish. Someday.

This project is taking a litte longer than expected. Having some language problems as well. But we will finish. We promise.

Language problems, rebuilding floors 1.000-2.5000

24.06.2007

Some misunderstandings between "The Working Wombats" and "The Building Bees" led to floor 2.500 (luxury apartments) beeing build on floor 1.000. Rebuilding of floors 1.000-2.500 necessary.

Construction Work

Working Group "The Building Bees"

Working Group "The Working Wombats"

New Question 19.08.2014 16:39

What is the role of Mrs. Smith?

New Question 19.08.2014 16:39

Available stone deposits in the area

"whose top may reach unto heaven" - that is the tower we want to build. Some may call us crazy. Some may call us naive. But we are

Use Cases

With DPP it is possible to build up an in-depth information website around a big project which can be used...

- ... **by administration** to inform the public about the progress of an ongoing project and so increase acceptance for it.
- ... **by civic groups** to monitor the progress of a project.
- ... **by parliament** for an enquiry around a project where certain problems occured.

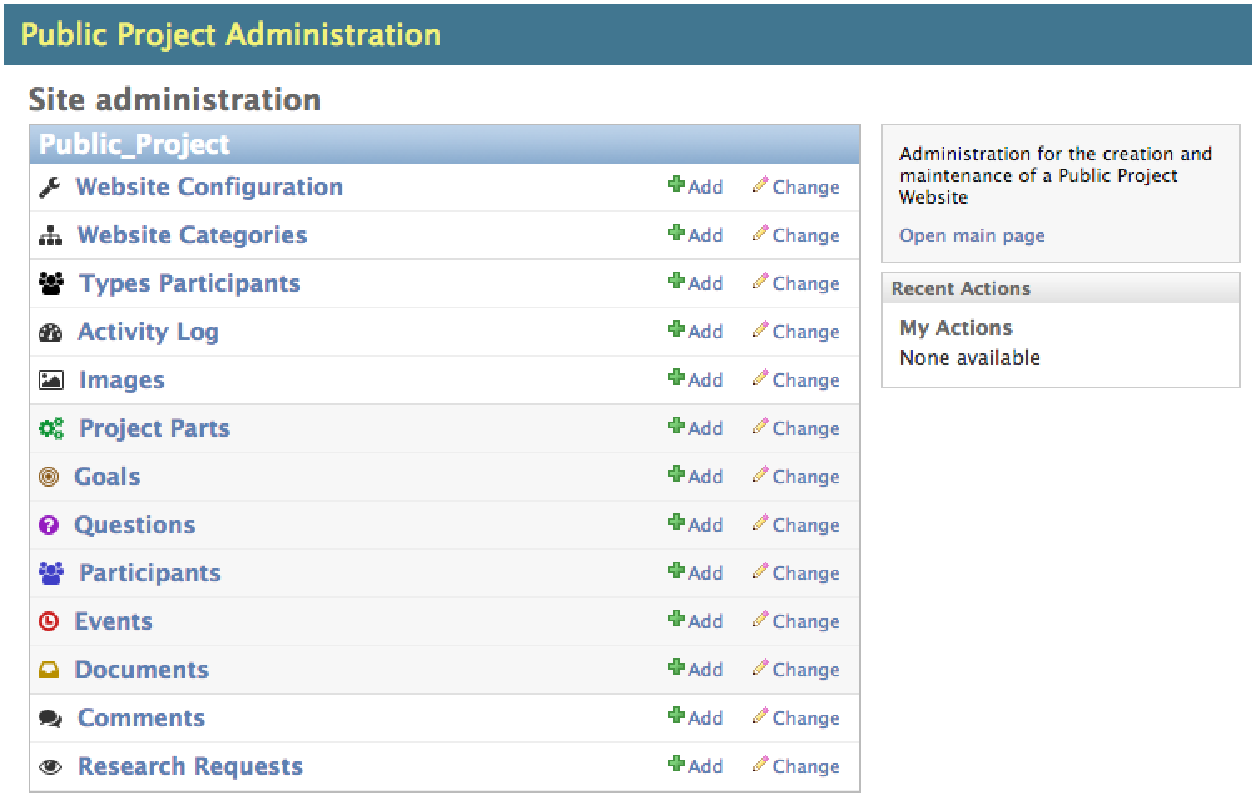
Features

DPP comes with an **administration interface**, where various high-level information around a project can be inserted and maintained. Which important *stakeholders* are involved in the project? What are *important events*? How is the *project structured*? All this information is then presented as a **front-end website** visible for end-users.

Notable features of the software are:

- Extensive admin interface
- Beautifully layouted front-end website
- Tags and external links for all project objects
- Full-text search
- Document manangement with integrated PDF viewer

- Advanced commenting system for end-users
- Languages available: EN/DE



Public Project Administration

Site administration

Public_Project		
Website Configuration	Add	Change
Website Categories	Add	Change
Types Participants	Add	Change
Activity Log	Add	Change
Images	Add	Change
Project Parts	Add	Change
Goals	Add	Change
Questions	Add	Change
Participants	Add	Change
Events	Add	Change
Documents	Add	Change
Comments	Add	Change
Research Requests	Add	Change

Administration for the creation and maintenance of a Public Project Website

Open main page

Recent Actions

My Actions

None available

Technology

DPP is build with `Python/Django` and comes as a `Django app` providing all the data models necessary together with the templates for the front end layout. This app can be integrated in a `Django` project representing the concrete project to be targeted.

> **Warning:** This software currently is in `beta` status. It can be used in a productive environment, but please follow closely the *Release Notes* if you want to make an update of the software and have an eye on changes in Django model declarations, software dependencies or config settings.

Installation

Requirements

For installing DPP you need the following `Python/Django` libraries, probably best installed in an own `virtualenv` environment:

- Python 2.7+ (Python 3.x not yet supported)
- Django 1.10 (1.11+ not yet supported)
- PDFMiner (Version 20110515 to avoid dependency errors!)

- Pillow 2.5.2+ (Replacing PIL, for Django ImageField type)
- Tastypie 0.13.3+ (for API access)

For PDF conversion to jpg files for having an IE compatible PDF viewer, you need to have the `ImageMagick` library with the `convert` command installed in your shell environment:

- ImageMagick (convert tool)

Note: There are some fabric tasks which can help you set up an environment for DPP located in an own GitHub repository which can be found here: https://github.com/holgerd77/django-public-project-fabric-tasks/

Installation with Pip

`DPP` is on the `Python Package Index` and you can install the software with all dependencies with:

```
pip install django-public-project
```

Manual Installation

If you want to have the latest version of `DPP`, you can install the sources manually with `PIP` (or directly clone the GitHub repository):

```
pip install -e git+https://github.com/holgerd77/django-public-project.git@master
↪#egg=django-public-project
```

Then install the requirements above. There is a `requirements.txt` file in the main directory of the repository you can use:

```
pip install -r requirements.txt
```

Project Creation

Create your `Django` project:

```
django-admin.py startproject myprojectwatch
```

Add the Django apps installed to your `settings.py` file (of course you also need the admin app which is essential for DPP):

```
INSTALLED_APPS = [
    ...
    'public_project', # Since DPP changes some admin templates, app has to be placed␣
↪before admin
    'django.contrib.admin',
    'tastypie',
]
```

Sync your database respectively use migrations for DPP:

```
python manage.py syncdb (due to database dependencies, don't create a superuser yet)
python manage.py migrate
python manage.py createsuperuser
```

Configuration

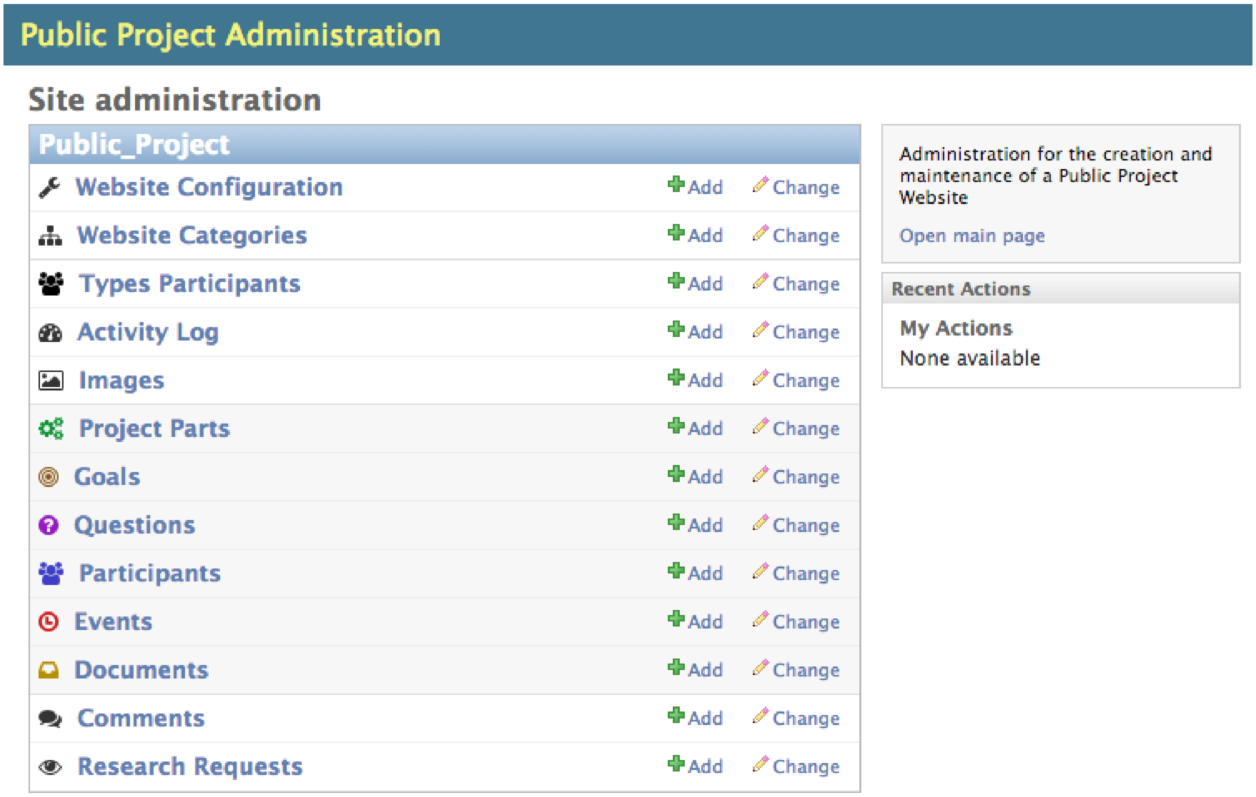
DPP is not really an app which you would install beside many other Django apps and integrate it in a more complex website. It is more a content management system already coming with an url structure and a given layout capsuled in a single app. So DPP takes control of more things than the normal Django app.

URL structure

The urlpatterns for your project are completely coming from DPP, with an exception of the admin url, which should be adoptable for security reasons. So your minimal urls.py should look similar to this, importing the main url patterns from `public_project.urls`:

```
from django.conf import settings
from django.conf.urls import patterns, include, url

from django.contrib import admin
admin.autodiscover()

from public_project.urls import urlpatterns

urlpatterns += [
    url(r'^admin/', include(admin.site.urls)),
]

# Necessary for being able to use image upload in DEBUG mode
if settings.DEBUG:
    urlpatterns += [
        url(r'^media/(?P<path>.*)$', 'django.views.static.serve', {
        'document_root': settings.MEDIA_ROOT})
    ]
```

Now you should be able to enter both the admin view and an emtpy front-end dashboard site when you start a dev server. The site itself is not yet ready for prime time at this moment.



Public Project Administration

Site administration

Public_Project		
Website Configuration	Add	Change
Website Categories	Add	Change
Types Participants	Add	Change
Activity Log	Add	Change
Images	Add	Change
Project Parts	Add	Change
Goals	Add	Change
Questions	Add	Change
Participants	Add	Change
Events	Add	Change
Documents	Add	Change
Comments	Add	Change
Research Requests	Add	Change

Administration for the creation and maintenance of a Public Project Website

Open main page

Recent Actions

My Actions

None available

Basic settings

Since I'm not sure, if there are still some static references to static or media files somewhere in the code, you should use the following `STATIC_URL` and `MEDIA_URL` settings:

```
MEDIA_URL = '/media/'
STATIC_URL = '/static/'
```

For being able to get email notifications about comments and document relations, you need to configure the Django email settings properly:

```
EMAIL_FROM = 'admin@yourmailaccount.com'
EMAIL_HOST = 'smtp.yoursmtpserver.com'
EMAIL_HOST_USER =  'YOURUSERNAME'
EMAIL_HOST_PASSWORD = 'YOURSECUREPASSWORD'
```

DPP uses the request template context processor in its views and adds its own context processors, add them to the `TEMPLATES` setting in your `settings.py` file:

```
TEMPLATES = [
  {
      'BACKEND': 'django.template.backends.django.DjangoTemplates',
      'APP_DIRS': True,
      'OPTIONS': {
          'context_processors': [
              'public_project.context_processors.uploaded_images_list',
              'django.template.context_processors.request',
              'django.contrib.auth.context_processors.auth',
              'django.template.context_processors.debug',
              'django.template.context_processors.i18n',
              'django.template.context_processors.media',
              'django.template.context_processors.static',
              'django.template.context_processors.tz',
              'django.contrib.messages.context_processors.messages',
          ],
      },
  },
]
```

Language Selection

At the moment DPP supports the following languages:

- English (en) (experimental and not yet used in production, probably you have to correct some stuff)
- German (de)

The language is chosen depending on the `LANGUAGE_CODE` param in the `settings.py` module, e.g.:

```
LANGUAGE_CODE = 'de-de'
```

Document upload/viewer

The document viewer in DPP is based on the Mozilla pdf.js library (included in DPP) when using modern browsers like Google Chrome, Firefox or Safari.

For being able to view on site pdf documents with the Microsoft Internet Explorer there exists a basic alternative pdf viewer. For this viewer, single pages are converted to png files and are stored on disk and you need to have the `ImageMagick` library installed and make sure that the `convert` command from this library can be used from within your project path.

Since this approach can take a lot of disk space for large documents and root access to the server is needed, you have to activate IE compatible pdf viewer usage with the following setting in your `settings.py` file:

```
DPP_IE_COMPATIBLE_PDF_VIEWER = True
```

If this setting is set to false (default) a warning message will be shown on the document page for IE users, prompting them to use an alternative browser.

If this setting is set to true, documents are saved as the original pdf file and a corresponding document_x folder containing the pngs in your media folder. Please test-upload a pdf document and see if these files are generated. Then test the url with the pdf viewer for this document in both the MSIE and another browser.

Note: The conversion process of a pdf document takes place in the background and may take a while for large documents.

Custom JS/CSS Code

If you want to include custom Javascript code or CSS styles into your project - e.g. to add analytics to the site or customize the layout, you can use the following settings:

```
DPP_CUSTOM_JS = 'alert("This should show up on every page!")' #Example JS Code

DPP_CUSTOM_CSS = 'body { margin: 20px; }' #Example CSS Style
```

Site Domain

For urls in comment emails to work properly, you have to edit the `Site` object, which Django should have created in the `Sites` section in the Django admin.

Provide your fully qualified domain name there (e.g. ‘yourproject.yourdomain.com’), without trailing ‘http://‘.

JSON API

Since `v.0.4` DPP comes with a public API, which let developers access the public data of the system, leaving out internal comments and user comments. The API supports no authentication mechanism yet and will be accessible by everyone without limitation. To activate the API, add the following to your `settings.py` file:

```
DPP_PUBLIC_API = True
```

For the API to work you have got to have Tastypie 0.9.15+ installed:

```
pip install django-tastypie
```

And add `tastypie` to your `INSTALLED_APPS`.

When the API is working there will be an extra link in the footer leading to to API overview page:

```
http://yourproject.org/api/
```

Note: The API is still in an experimental/early stage, many features are missing and usage params will probably change in the future.

Where to go from here?

The main set-up process for your project website is now finished and the site is ready to be filled with some data.

Congratulations! :-)

Start by adding/changing some configuration parameters and introductory texts in the `SiteConfig` and `SiteCategory` menu.

In the next chapter you will learn how to use the admin interface and how to build up an information website around your project.

Tutorial

The Example Project

There is an **example project “Tower of Babel”** coming with the DPP sources as a ready-configured `Django project` containing some test data.

This tutorial is referencing the example project in the different sections. If you want you can take it as a basis when starting to learn, add some additional data, edit entries and play around a bit to get a feeling for the software. It is also possible though to directly start entering your own project data or build up an own first example.

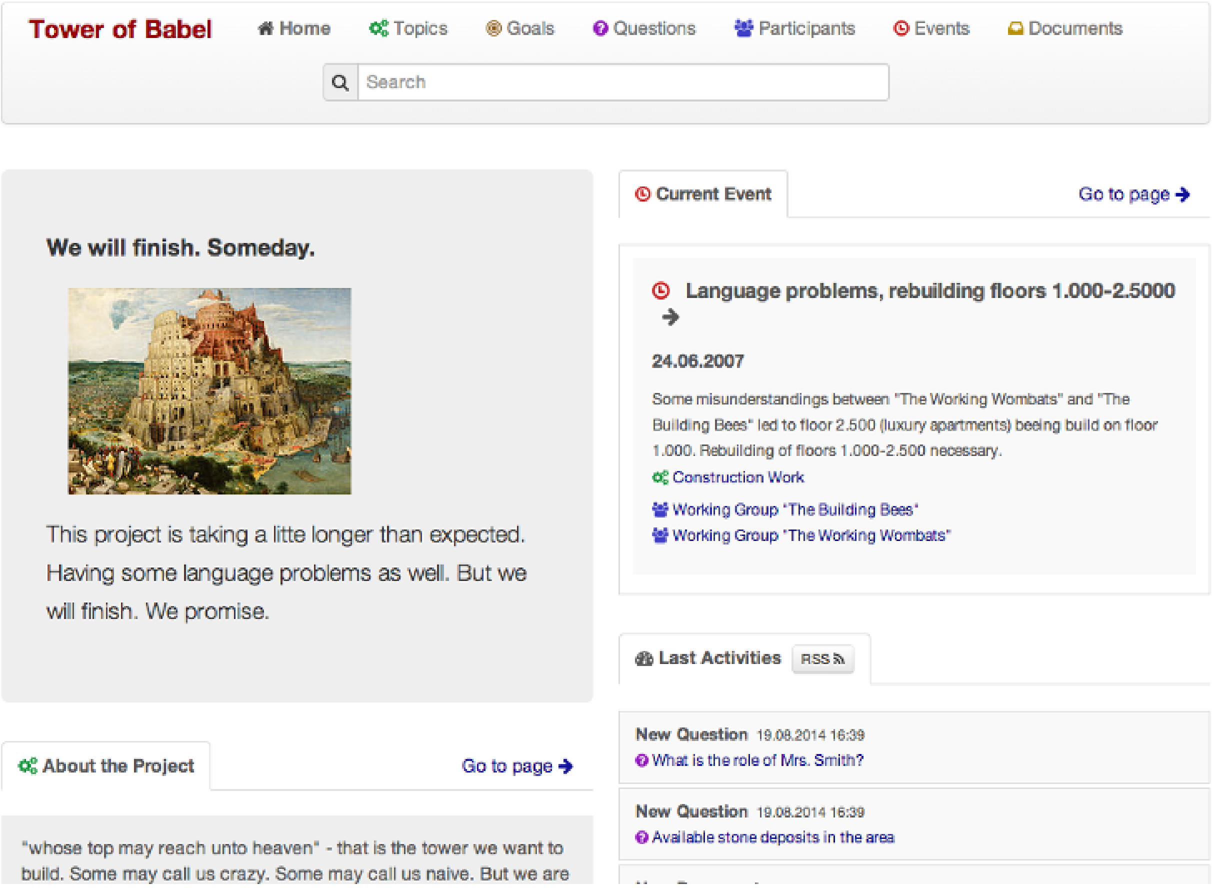
For using the example project find the `example_project` folder in your DPP installation directory and copy the folder to your own location.

Go to your folder copy and run the following scripts to init the project:

```
./init_example_project.sh
./create_example_data.sh
```

Note: If you are running Windows and you don't have configured a way to run shell scripts from the command line you have to look at both script files and run the commands manually (if you rewrite these scripts as Windows batch files you are very welcome to make a pull request on GitHub!).

You can then start the server with `./runserver.sh` and should be able to reach the `admin interface` and front-end website via the URLs provided.



We will finish. Someday.

This project is taking a litte longer than expected. Having some language problems as well. But we will finish. We promise.

Language problems, rebuilding floors 1.000-2.5000

24.06.2007

Some misunderstandings between "The Working Wombats" and "The Building Bees" led to floor 2.500 (luxury apartments) beeing build on floor 1.000. Rebuilding of floors 1.000-2.500 necessary.

Construction Work

Working Group "The Building Bees"
Working Group "The Working Wombats"

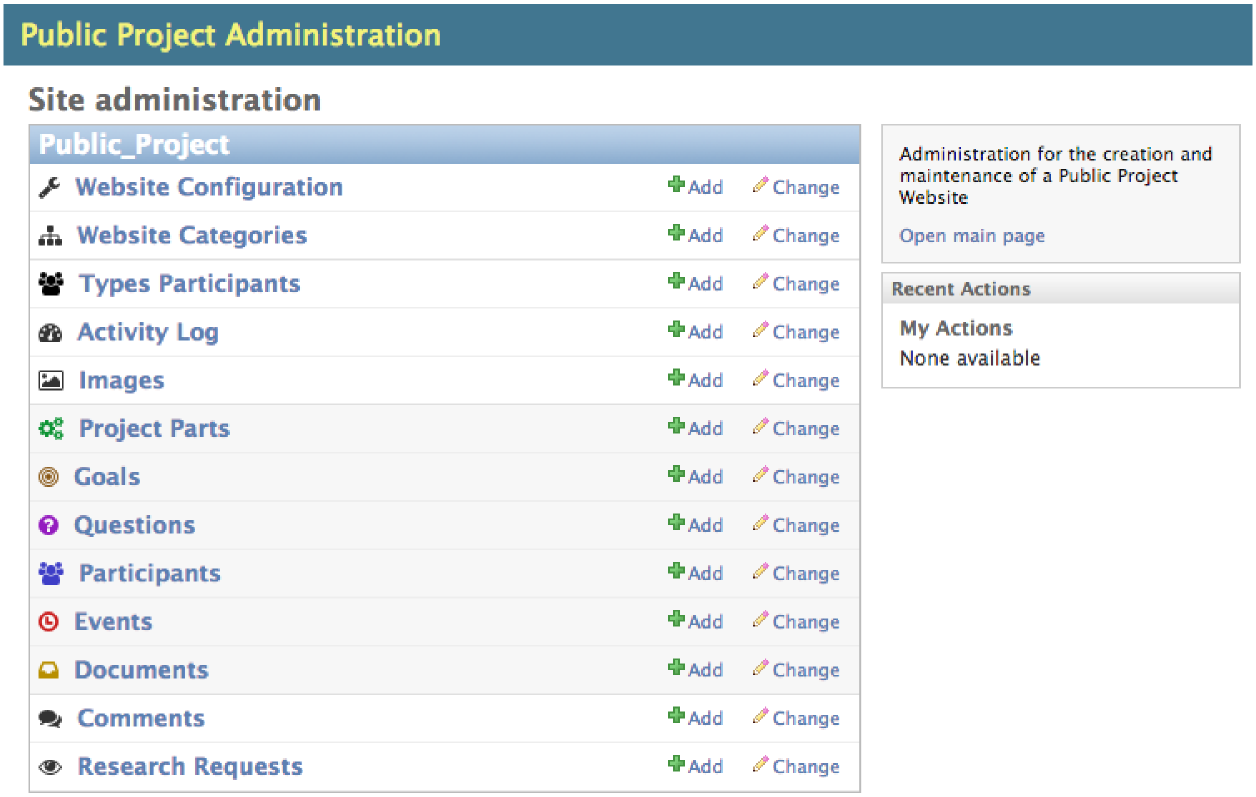
New Question 19.08.2014 16:39
What is the role of Mrs. Smith?

New Question 19.08.2014 16:39
Available stone deposits in the area

"whose top may reach unto heaven" - that is the tower we want to build. Some may call us crazy. Some may call us naive. But we are

Website Configuration and Categories

Before you start giving some structure to your project by providing topics and entering information about the different stakeholders and important events, you probably want to start with some general set-up of your site.



Public Project Administration

Site administration

Public_Project		
Website Configuration	Add	Change
Website Categories	Add	Change
Types Participants	Add	Change
Activity Log	Add	Change
Images	Add	Change
Project Parts	Add	Change
Goals	Add	Change
Questions	Add	Change
Participants	Add	Change
Events	Add	Change
Documents	Add	Change
Comments	Add	Change
Research Requests	Add	Change

Administration for the creation and maintenance of a Public Project Website

Open main page

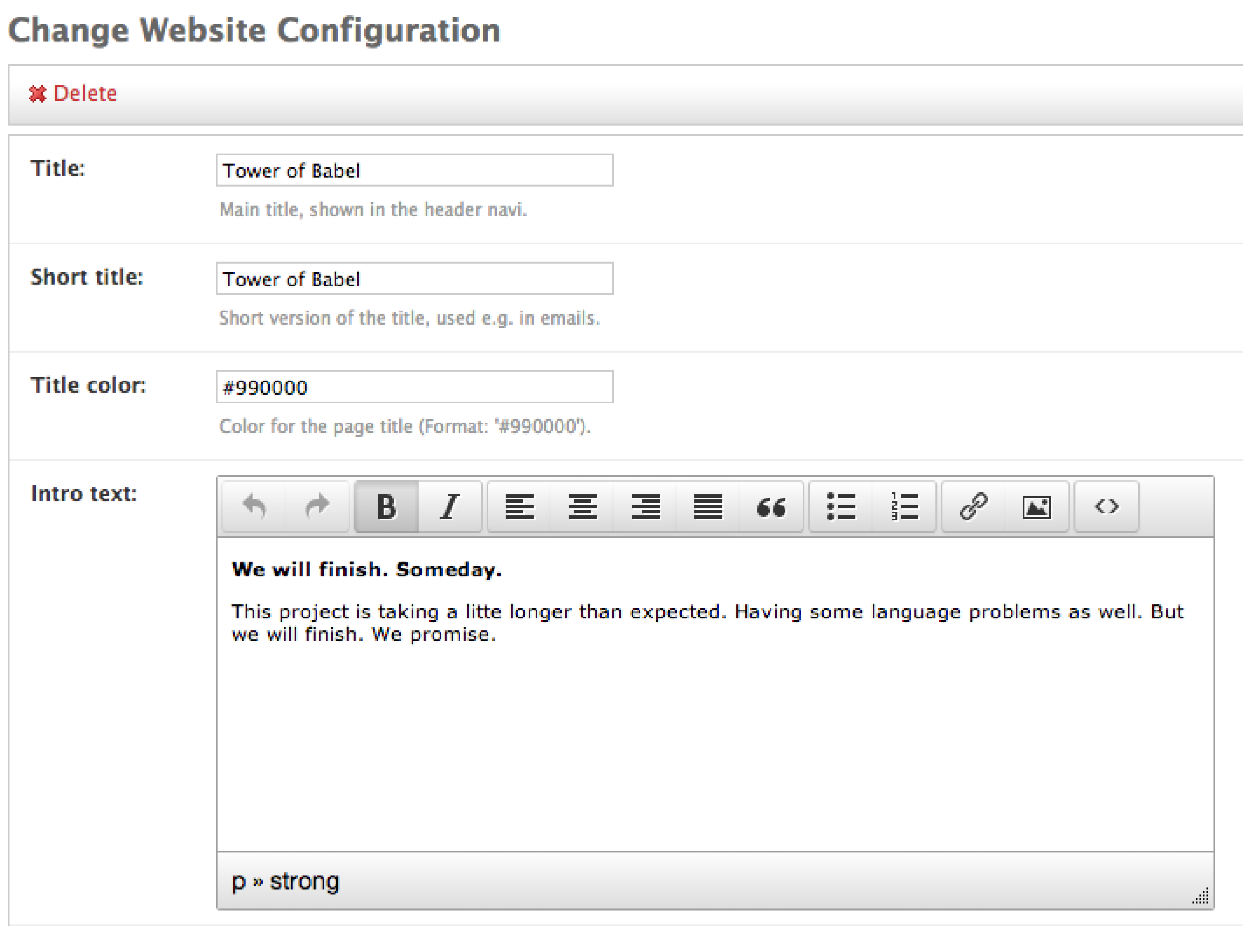
Recent Actions

My Actions

None available

Website Configuration

In the `Website Configuration` menu in your `project admin` you can enter a `title` for your project, a short `intro text` for the main page and some other generic texts like contact information for the "Contact" page or a short footer text.



Change Website Configuration

Title: Tower of Babel
Main title, shown in the header navi.

Short title: Tower of Babel
Short version of the title, used e.g. in emails.

Title color: #990000
Color for the page title (Format: '#990000').

Intro text:

We will finish. Someday.

This project is taking a litte longer than expected. Having some language problems as well. But we will finish. We promise.

Enter/edit some text and have a look how your text is formatted on the front-end website.

Website Categories

All the main categories on front-end website - Home, Topics, Goals,... - have their own item in the Website Category menu in the admin. Don’t add or delete items here, just edit the existing ones.

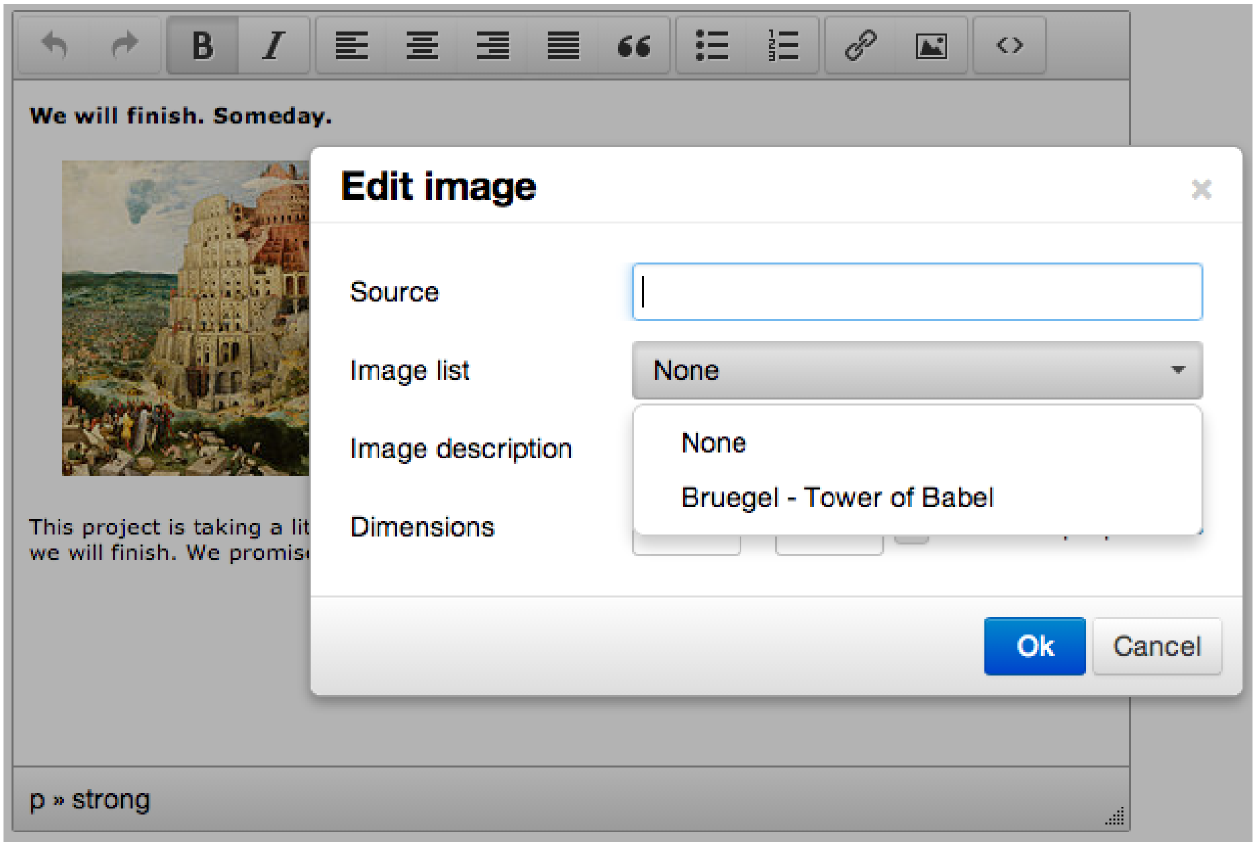
You can enter an introductory text for each category and add some documents or web sources (more on that later).

Side Quest: Adding Images

All longer description fields in the admin come with a simple WYSIWYG editor providing some basic formatting options like text formatting or text alignment. It is also possible to adding an image to your description.

For using an image in a description field you have to first upload it by going to the Images menu from the admin overview page and then add a new image. Upload the image, give it a meaningful title and provide attribution information if you took the image from an external source (this will be displayed publicly on the website).

You can then inserting the image by going back to your item to edit, click on the image icon and choose your image by title from the image list:


We will finish. Someday.

This project is taking a lit
we will finish. We promis

Edit image

Source

Image list: None

Image description

Dimensions

None

Bruegel - Tower of Babel

Topics (Project Parts)

Now that you have your basic website configuration in place you can focus on the project itself. Start by entering some topics around your project (for legacy reasons also sometimes referred to as "Project Parts").

Topics will be used to structure other project items such as documents or events, so you should think about how you want to structure your project. You can add/change topics later, but it will cause some extra work.

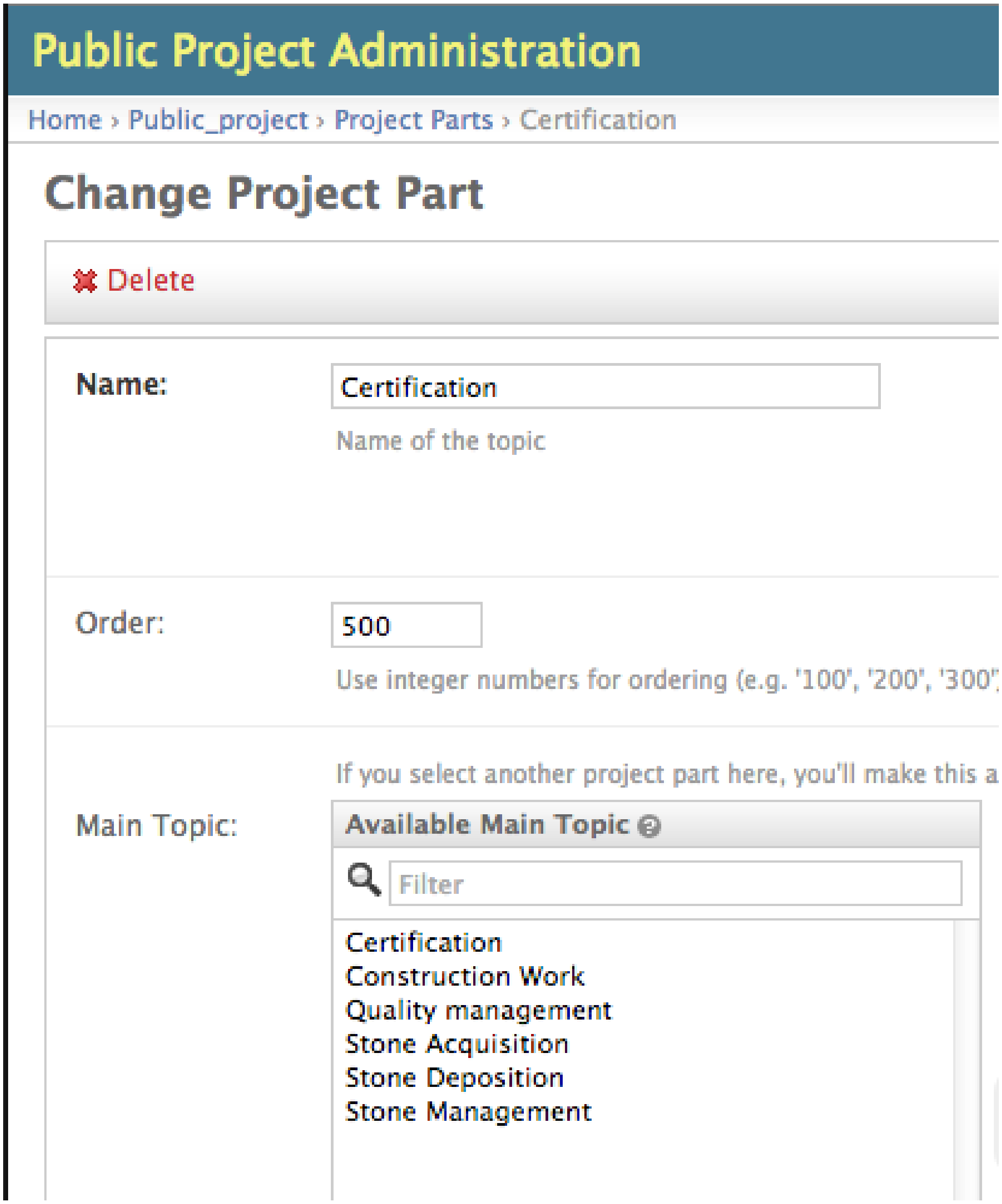
Some ideas for structuring a project would be:

- By **physical entities**, e.g. for a construction project (airport): "Terminals", "Runways", "Facilities", "Public Transport"
- By some **management categories**: "Planning", "Costs", "Organization", "Controlling",...
- **Geographical entities** if things are happening in different locations: "Berlin", "Hamburg", Stuttgart"

Often this is very much depending on the project you want to describe and the specifics you are most interested in. It is also possible to mix some of the topic dimensions from the examples above.

Adding a Topic

For adding a topic go to the `Topic (Project Part)` section in the `admin` and add a new entry.



Public Project Administration
Home › Public_project › Project Parts › Certification
Delete
Available Main Topic
Filter

Change Project Part

Name: Certification

Name of the topic

Order: 500

Use integer numbers for ordering (e.g. '100', '200', '300'

Main Topic:

If you select another project part here, you'll make this a

Certification
Construction Work
Quality management
Stone Acquisition
Stone Deposition
Stone Management

Start by providing a meaningful `name` and a `description` of your topic.

If you have a fairly complex project you want to present on your website it may be useful to **hierarchically structure** your project. If you choose one or more other topics you entered before as a `Main Topic`, your new topic will be listed as a `Sub Topic` under these main topics.

Side Quest: Ordering Items

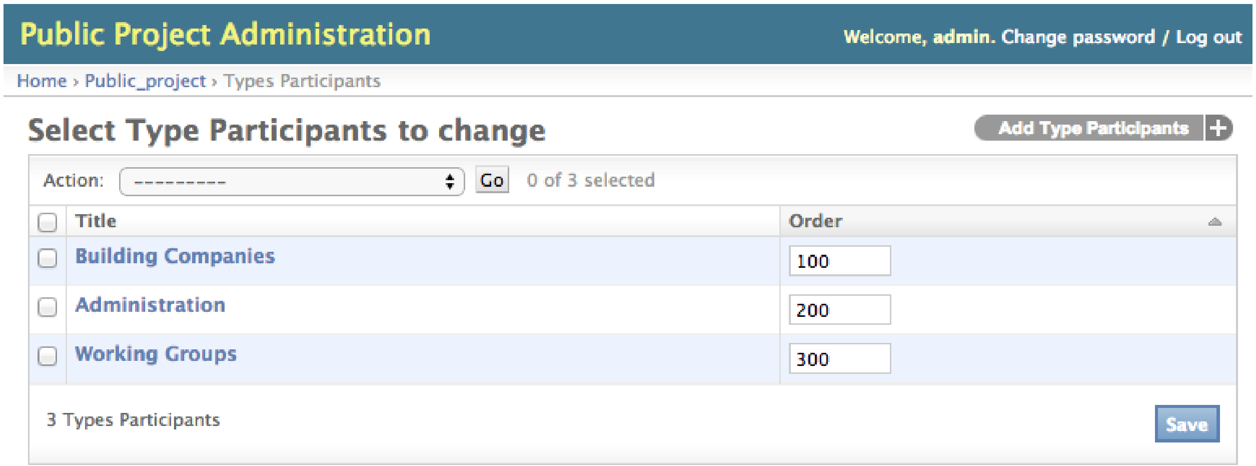
Many item types come with an `order` attribute you can use to determine the order the items are displayed on the website.

Items are ordered by entering integer numbers, starting from low to high. So `100`, `200`, `300`, `400`, `500` would order items in the desired manner. It is recommended to user numbers in steps by `100` (`1`, `2`, `3`, `4`, `5` would work as well). This makes it easier to change ordering afterwards. For getting the fifth element as first, you just have to change `500` to `50`, leading to a `50`, `100`, `200`, `300`, `400` order. Otherwise you would have to go to every single element and change every `order` entry to have the same effect.

Participants

You can enter information on the different stakeholders of your project in the `Participants` menu in the `admin`.

Participants are structured in their category on the front-end website by type. You have to first enter the `Participant Type` menu in the `admin` and think about what type of participants you will have. This could be something generic like "Administration", "Politics", "Companies", but also something more tailored to your project.



Public Project Administration
Welcome, admin. Change password / Log out
Home › Public_project › Types Participants
Add Type Participants +
Go
Save

Select Type Participants to change

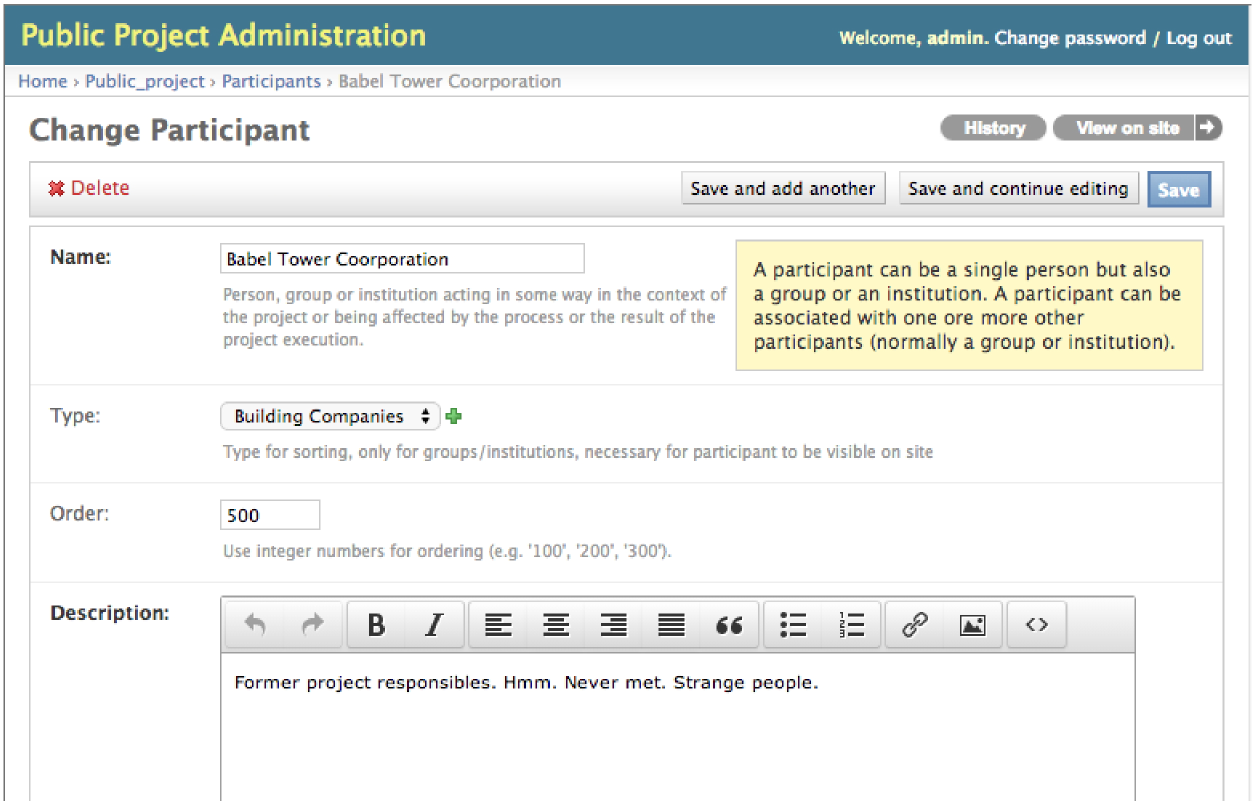
Action: ---------

0 of 3 selected

□	Title	Order
□	Building Companies	100
□	Administration	200
□	Working Groups	300

3 Types Participants

Once you have your participant types in place you can add/edit a concrete participant. Start by providing a `name`, choosing one of the `participant types` you entered before and provide a brief description.


Public Project Administration
Welcome, admin. Change password / Log out
Home › Public_project › Participants › Babel Tower Coorporation
History
View on site
Delete
Save and add another
Save and continue editing
Save

Change Participant

Name: Babel Tower Coorporation

Person, group or institution acting in some way in the context of the project or being affected by the process or the result of the project execution.

A participant can be a single person but also a group or an institution. A participant can be associated with one ore more other participants (normally a group or institution).

Type: Building Companies

Type for sorting, only for groups/institutions, necessary for participant to be visible on site

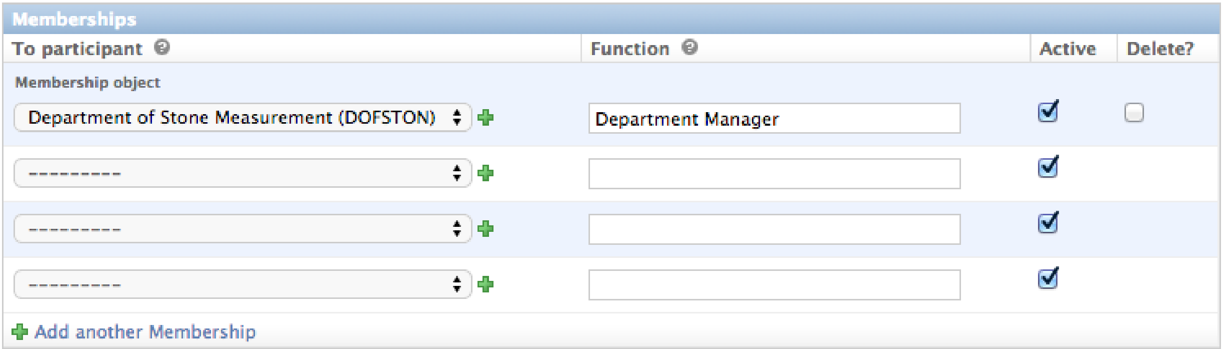
Order: 500

Use integer numbers for ordering (e.g. '100', '200', '300').

Description:

Former project responsibles. Hmm. Never met. Strange people.

Participants can be organized very flexibly and can both be institutions (companies, organizations and so on) or human beings. A particiant can have a `membership` relation to another participant. Normally this will mean a person beeing a `Member` of a company or organization beeing deployed there, but this can also be a ministry being part of a government, depending on what you want to describe.

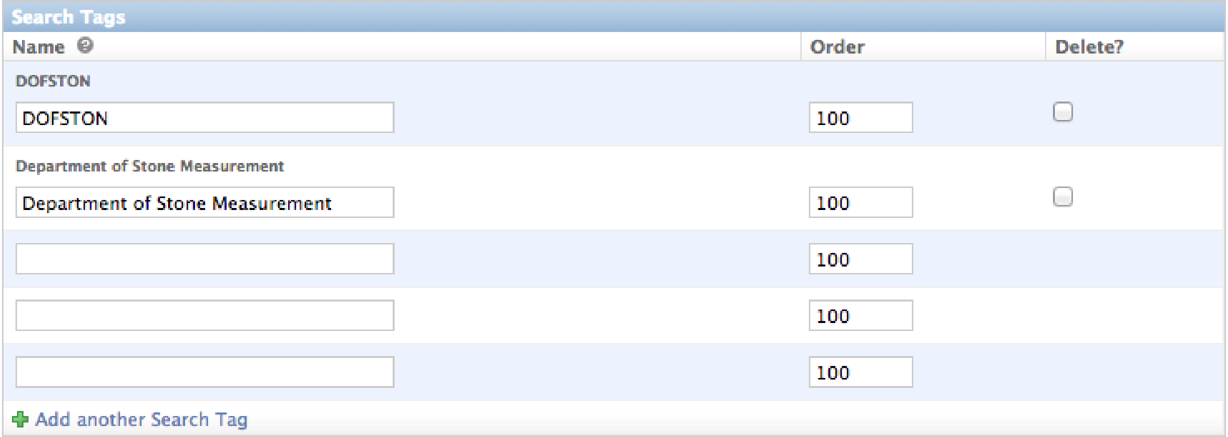


To participant	Function	Active	Delete?
Membership object			
Department of Stone Measurement (DOFSTON)	Department Manager	☑	□
---------		☑	
---------		☑	
---------		☑	

Side Quest: Adding Search Tags

For most types of project items you are able to add `Search Tags` to enrich the item data. These are mainly used for automatically finding fitting document content associated with this item.

The following `Search Tags` for the "Department of Stone Measurement" participant in the example...

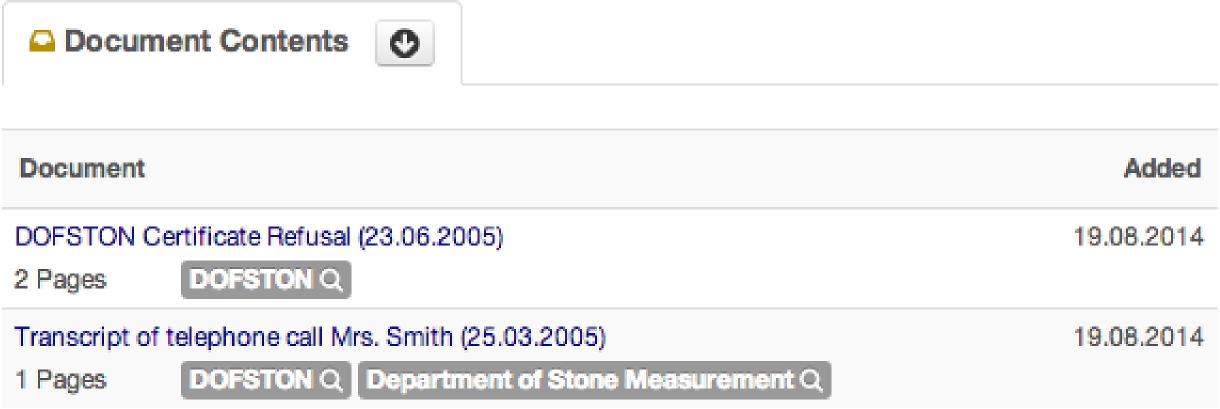


Search Tags

Name	Order	Delete?
DOFSTON DOFSTON	100	□
Department of Stone Measurement Department of Stone Measurement	100	□
	100	
	100	
	100	

Add another Search Tag

...leads to documents containing these tags being listed on the participant page:



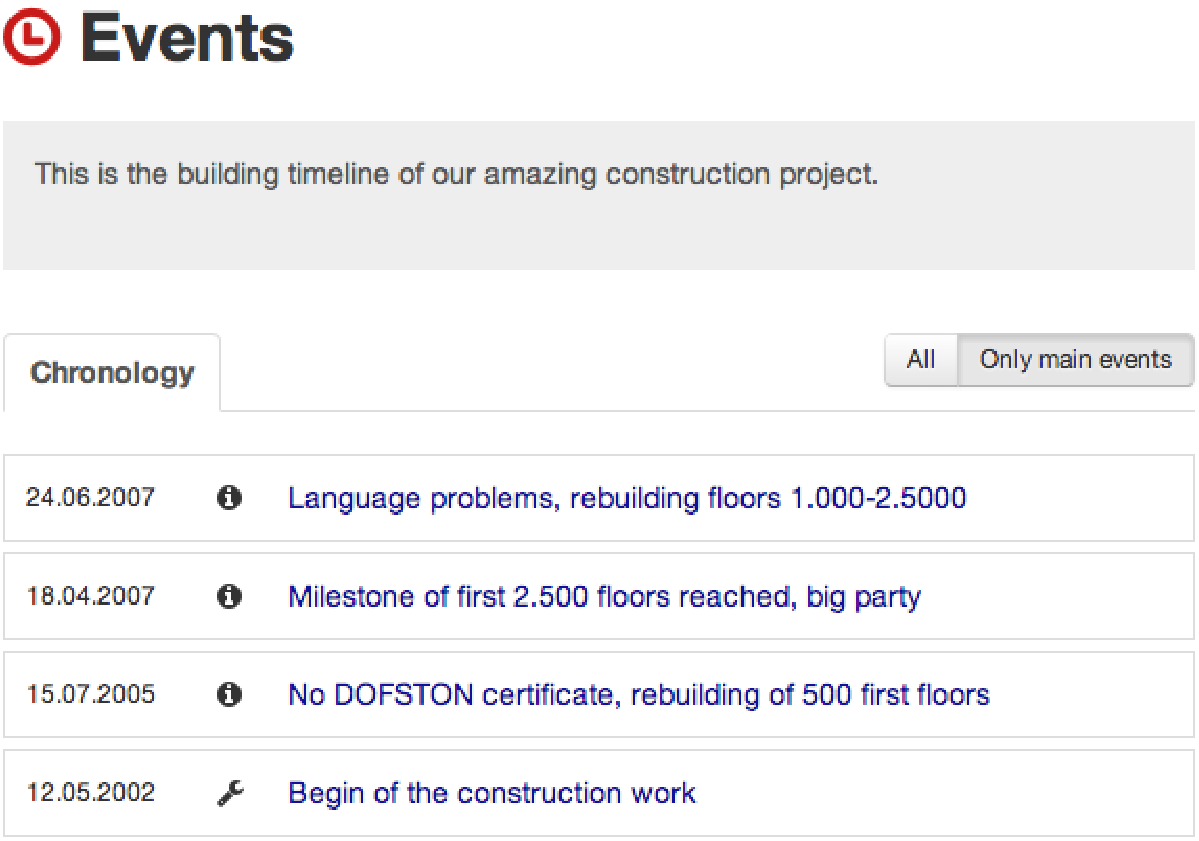
Document Contents

Document	Added
DOFSTON Certificate Refusal (23.06.2005) 2 Pages DOFSTON	19.08.2014
Transcript of telephone call Mrs. Smith (25.03.2005) 1 Pages DOFSTON Department of Stone Measurement	19.08.2014

For particpants it makes sense use synonyms or abbreviations of a participants name as a `Search Tag`, for a topic tags describing the content of the topic might be useful. Just experiment with this a bit. Note that you can't see the effects of a search tag before you added some documents.

Events

You can add important events around your project in the `Events` menu of the administration interface, which are then displayed chronologically as well as sorted by the topics you provided on the front-end website:



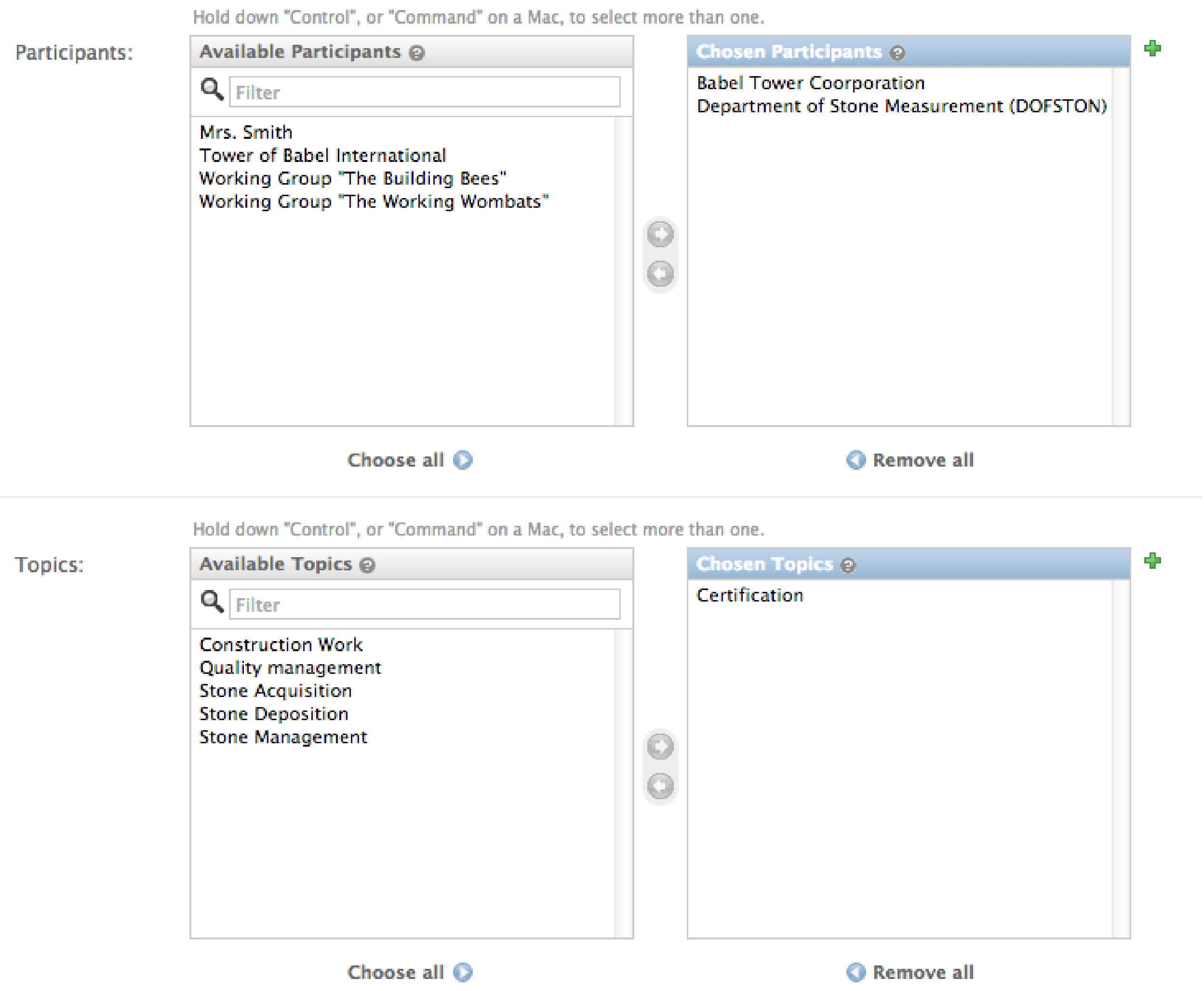
Events

This is the building timeline of our amazing construction project.

24.06.2007	ℹ	Language problems, rebuilding floors 1.000-2.5000
18.04.2007	ℹ	Milestone of first 2.500 floors reached, big party
15.07.2005	ℹ	No DOFSTON certificate, rebuilding of 500 first floors
12.05.2002	🔧	Begin of the construction work

One of the central concepts of `DPP` is, that you can provide connections between different items/aspects of your project whereever possible.

Events e.g. can be connected with several participants and project topics:



Hold down "Control", or "Command" on a Mac, to select more than one.

Participants:

Available Participants

Mrs. Smith
Tower of Babel International
Working Group "The Building Bees"
Working Group "The Working Wombats"

Chosen Participants

Babel Tower Coorporation
Department of Stone Measurement (DOFSTON)

Hold down "Control", or "Command" on a Mac, to select more than one.

Topics:

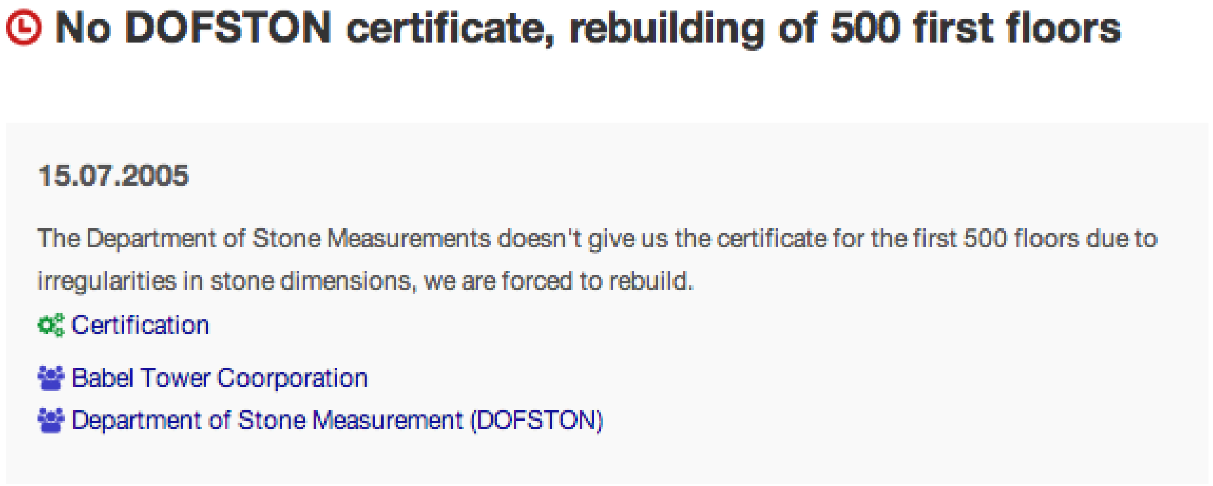
Available Topics

Construction Work
Quality management
Stone Acquisition
Stone Deposition
Stone Management

Chosen Topics

Certification

These connections are then presented on the front-end website together with some identifying icons:



No DOFSTON certificate, rebuilding of 500 first floors

15.07.2005

The Department of Stone Measurements doesn't give us the certificate for the first 500 floors due to irregularities in stone dimensions, we are forced to rebuild.

Certification

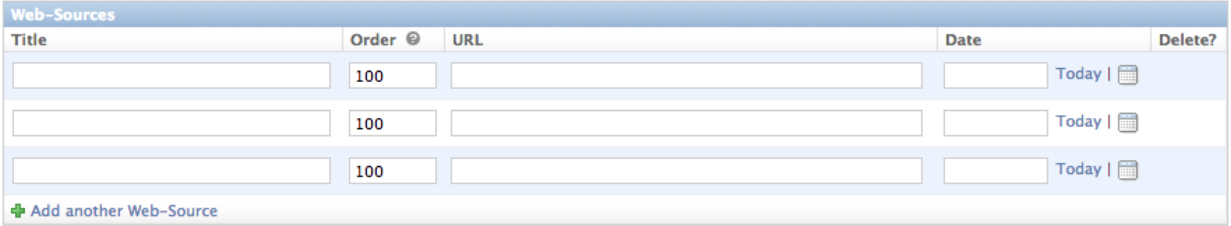
Babel Tower Coorporation

Department of Stone Measurement (DOFSTON)

This makes it very easy for end-users of the site to jump from item to item and explore the different aspects of the project.

Side Quest: Providing Web Sources

If you are describing a project of public interest it is very likely that important events are accompanied by media coverage and web articles.



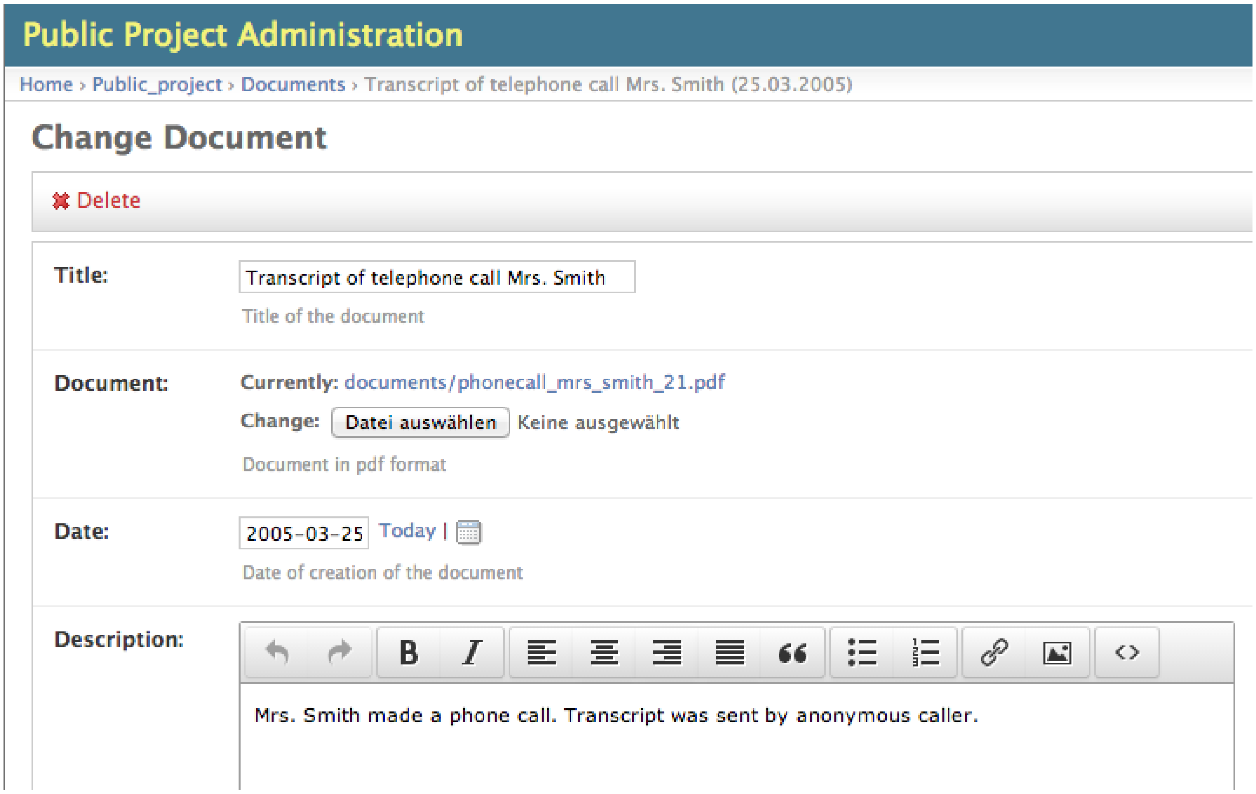
Web-Sources

Title	Order	URL	Date	Delete?
	100		Today |	
	100		Today |	
	100		Today |	

Adding `web sources` to project items is a very integral part of `DPP` you should make heavy use of and is possible for nearly every item type. Often many things have already been written on the web - there are Wikipedia articles on participants and news coverage of events - and it is of no necessity to write all these things again. Instead it often makes more sense - and is also less work - to link to the original source.

Documents

In `DPP` it is possible to provide documents about and from a project and then interlink this document with other participants, topics or events. For adding a document go to the `Document` menu in the `admin`.



Public Project Administration
Home › Public_project › Documents › Transcript of telephone call Mrs. Smith (25.03.2005)

Change Document

Title: Transcript of telephone call Mrs. Smith
Title of the document

Document: Currently: documents/phonecall_mrs_smith_21.pdf
Change: Datei auswählen Keine ausgewählt
Document in pdf format

Date: 2005-03-25 Today |
Date of creation of the document

Description:

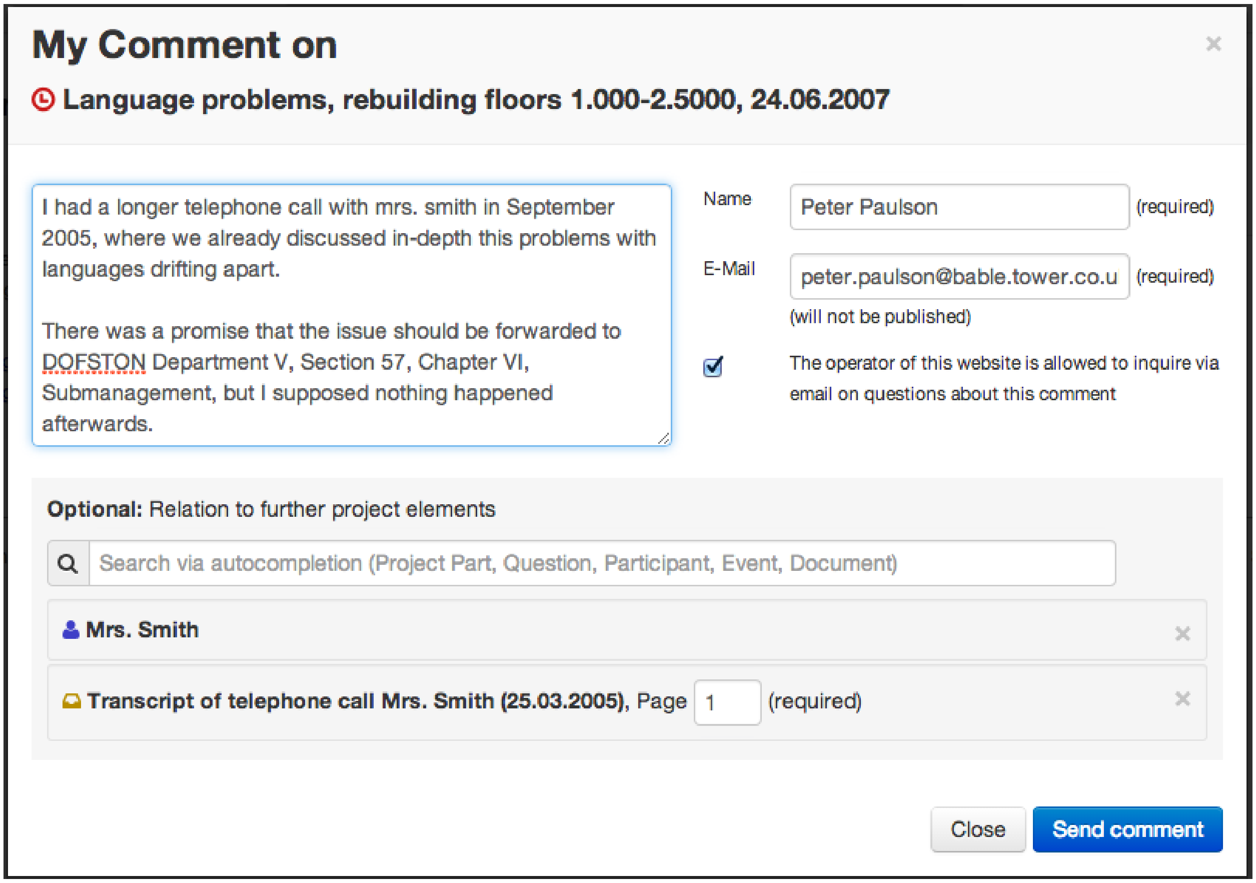
Mrs. Smith made a phone call. Transcript was sent by anonymous caller.

The document is then presented on the front-end website by the **integrated PDF viewer**. It also possible to **full-text search** through the document.

Side Quest: Managing User Comments

Beside from working out the structures and interdependencies of a project yourself you can rely on interested users visiting your site and encourage them to participate with their knowledge. For that DPP comes with an advanced commenting system.

Users are able to comment on every item of the project and connect their comment with additional participants, documents, etc.:



My Comment on

Language problems, rebuilding floors 1.000-2.5000, 24.06.2007

I had a longer telephone call with mrs. smith in September 2005, where we already discussed in-depth this problems with languages drifting apart.

There was a promise that the issue should be forwarded to DOFSTON Department V, Section 57, Chapter VI, Submanagement, but I supposed nothing happened afterwards.

Name: Peter Paulson (required)

E-Mail: peter.paulson@bable.tower.co.u (required)
(will not be published)

☑ The operator of this website is allowed to inquire via email on questions about this comment

Optional: Relation to further project elements

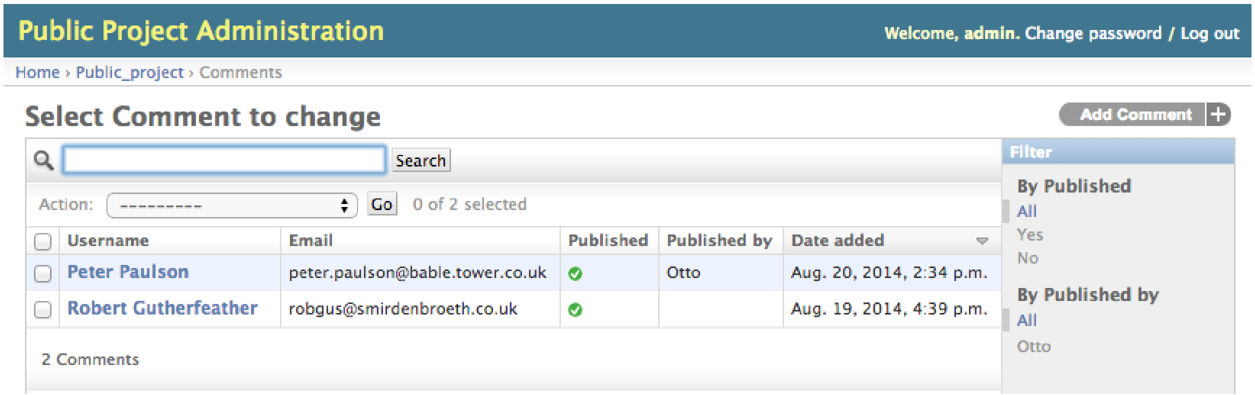
Search via autocompletion (Project Part, Question, Participant, Event, Document)

Mrs. Smith

Transcript of telephone call Mrs. Smith (25.03.2005), Page 1 (required)

`Comments` are displayed directly on the front-end website but have to be reviewed and published by an `admin user`. Every `admin user`, where `Receive new comment emails` is set, gets an email when a user is commenting something and can publish the comment via a link in the mail.

A comment can also be published directly in the `admin` by going to the specific `Comment` in the `Comments` menu:

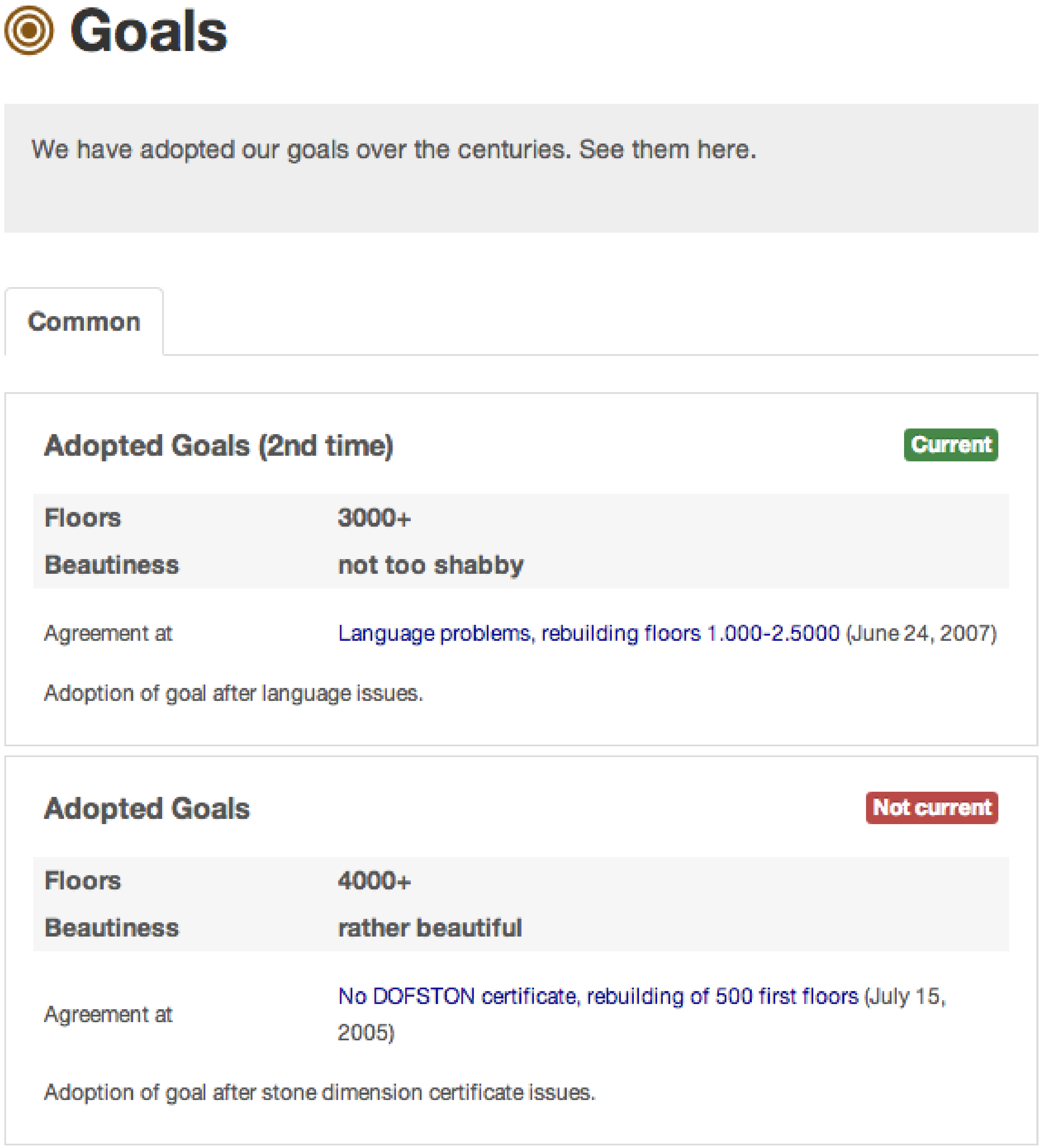


□	Username	Email	Published	Published by	Date added
□	Peter Paulson	peter.paulson@bable.tower.co.uk	✓	Otto	Aug. 20, 2014, 2:34 p.m.
□	Robert Gutherfeather	robgus@smirdenbroeth.co.uk	✓		Aug. 19, 2014, 4:39 p.m.

2 Comments

Goals

DPP lets you retrace and publish the development of project goals.



Goals

We have adopted our goals over the centuries. See them here.

Common

Adopted Goals (2nd time) Current

Floors	3000+
Beautiness	not too shabby
Agreement at	Language problems, rebuilding floors 1.000-2.5000 (June 24, 2007)

Adoption of goal after language issues.

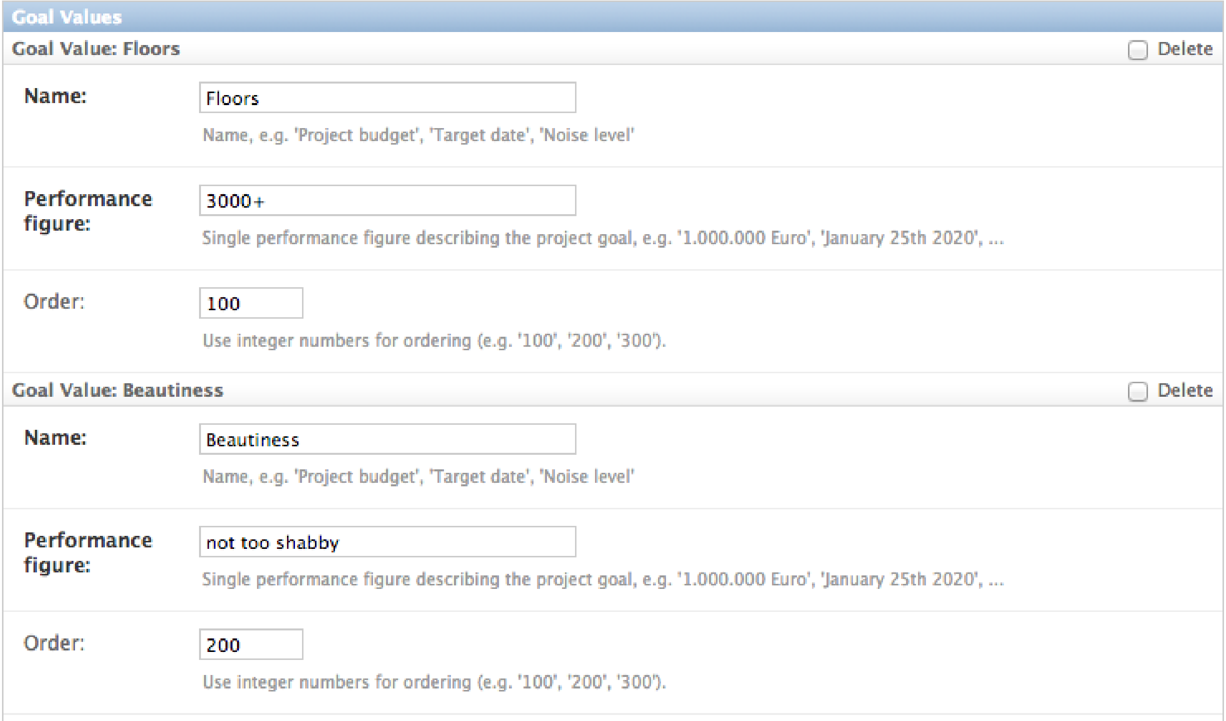
Adopted Goals Not current

Floors	4000+
Beautiness	rather beautiful
Agreement at	No DOFSTON certificate, rebuilding of 500 first floors (July 15, 2005)

Adoption of goal after stone dimension certificate issues.

For entering/editing a goal go to the `Goals` menu in the `admin`. Each `Goal` is a set of `performance figures` which were agreed on in some point in time and may be still current or not.

In our example `performance figures` are “floors” and “beautiness”, but this can be anything you like and you want to focus on depending on your project, like “costs”, “capacity” and so on.



Goal Values

Goal Value: Floors □ Delete

Name: Floors

Name, e.g. 'Project budget', 'Target date', 'Noise level'

Performance figure: 3000+

Single performance figure describing the project goal, e.g. '1.000.000 Euro', 'January 25th 2020', ...

Order: 100

Use integer numbers for ordering (e.g. '100', '200', '300').

Goal Value: Beautiness □ Delete

Name: Beautiness

Name, e.g. 'Project budget', 'Target date', 'Noise level'

Performance figure: not too shabby

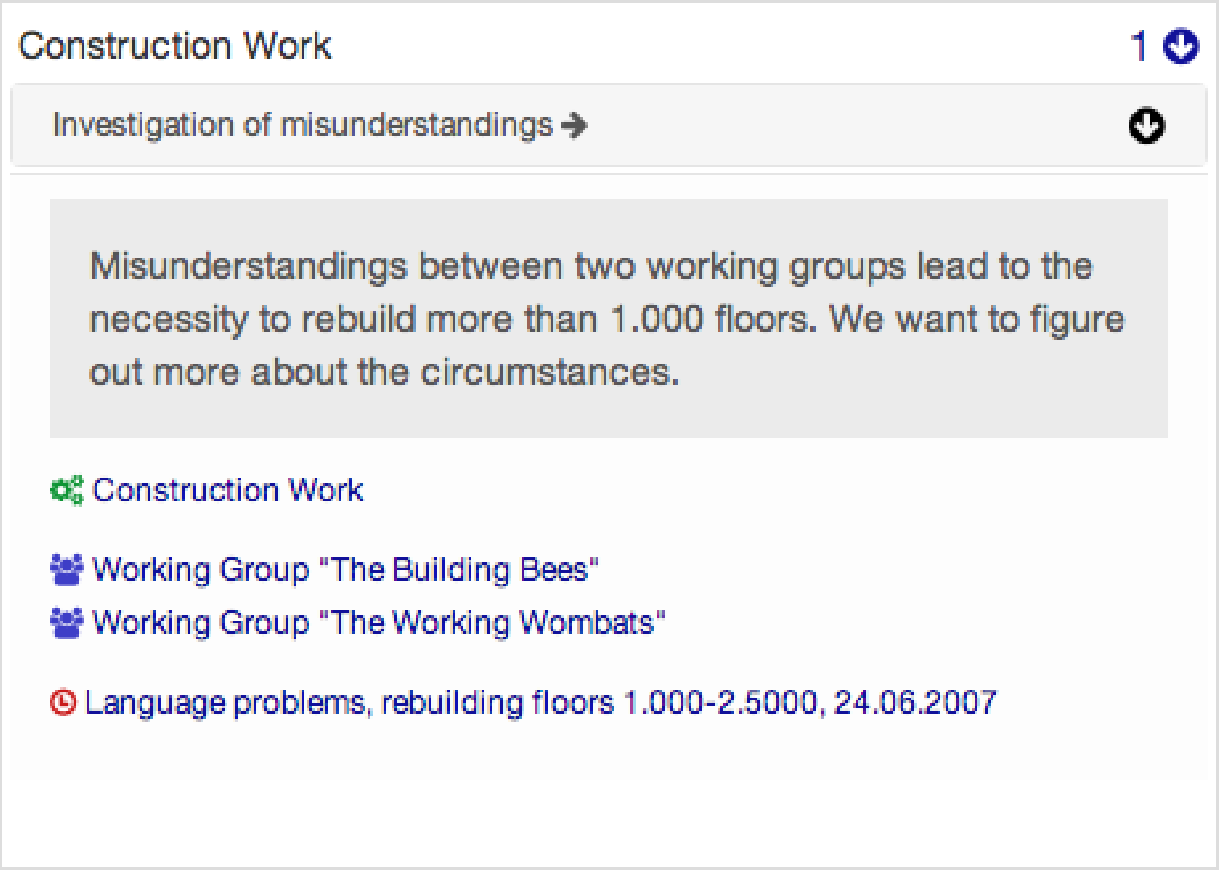
Single performance figure describing the project goal, e.g. '1.000.000 Euro', 'January 25th 2020', ...

Order: 200

Use integer numbers for ordering (e.g. '100', '200', '300').

Questions

If you have open questions or investigate some issues around your project, you can publish these questions in the `Questions` section.



Construction Work 1

Investigation of misunderstandings

Misunderstandings between two working groups lead to the necessity to rebuild more than 1.000 floors. We want to figure out more about the circumstances.

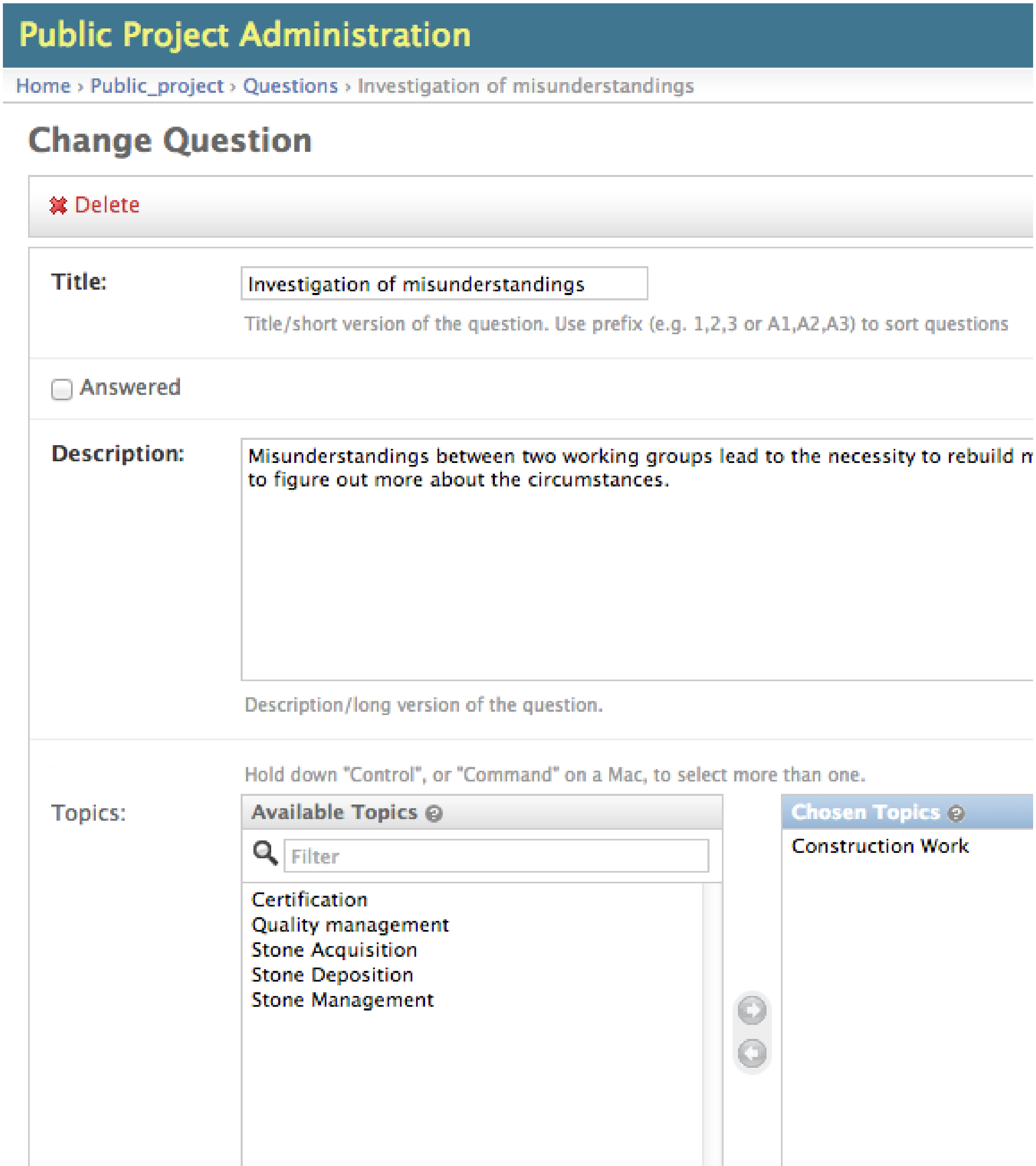
Construction Work

Working Group "The Building Bees"

Working Group "The Working Wombats"

Language problems, rebuilding floors 1.000-2.5000, 24.06.2007

Go to the `Questions` menu in the `admin` and enter a `title`, a `description` and one or several `topics` your question belongs to. You can also connect other items of your project with the question.


Public Project Administration
Home › Public_project › Questions › Investigation of misunderstandings
Change Question
Delete

Title: Investigation of misunderstandings

Title/short version of the question. Use prefix (e.g. 1,2,3 or A1,A2,A3) to sort questions

□ Answered

Description: Misunderstandings between two working groups lead to the necessity to rebuild
to figure out more about the circumstances.

Description/long version of the question.

Hold down "Control", or "Command" on a Mac, to select more than one.

Topics:

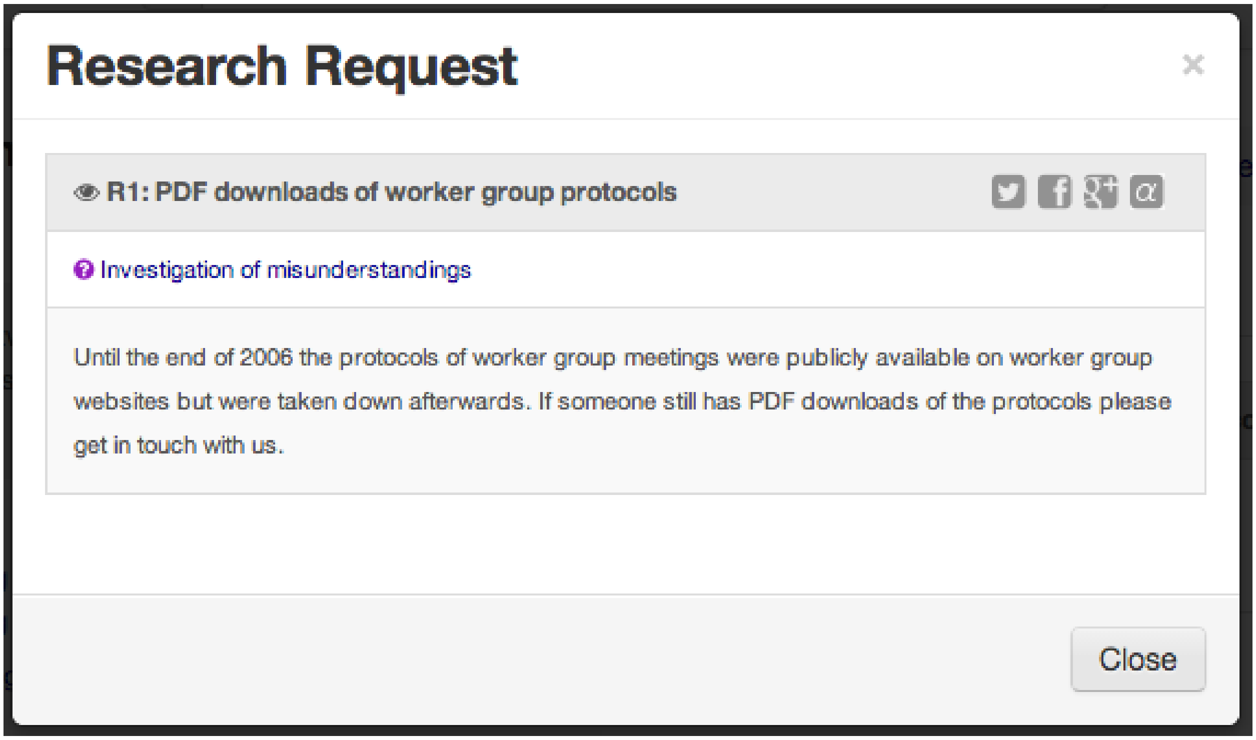
Available Topics: Certification, Quality management, Stone Acquisition, Stone Deposition, Stone Management

Chosen Topics: Construction Work

If a question is later answered you can check the `answered` checkbox and optionally provide a description of the answer in the additional text box.

Side Quest: Assigning Research Requests

If you want to go one step further including your users into the investigation of your project you can publish research requests associated with questions on your site.



Research Request

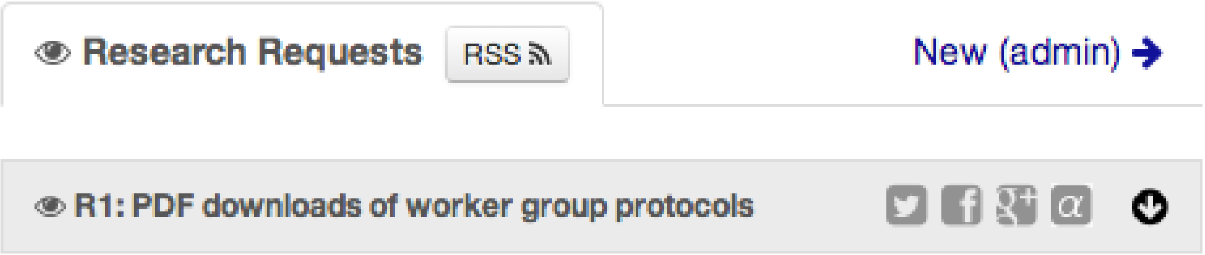
R1: PDF downloads of worker group protocols

Investigation of misunderstandings

Until the end of 2006 the protocols of worker group meetings were publicly available on worker group websites but were taken down afterwards. If someone still has PDF downloads of the protocols please get in touch with us.

In a research request you can describe a concrete task - researching some information, providing some special knowledge, helping in analyzing a document,... - and make it public for everyone capable to take part.

Adding a research request is not working (comfortably) on the admin interface. Instead log yourself in the system (on the admin). Then go to the question you want associate a research request with on the front-end website. There you find a button `New (YOUR_USERNAME) ->` on top of the `Research Requests Box`.

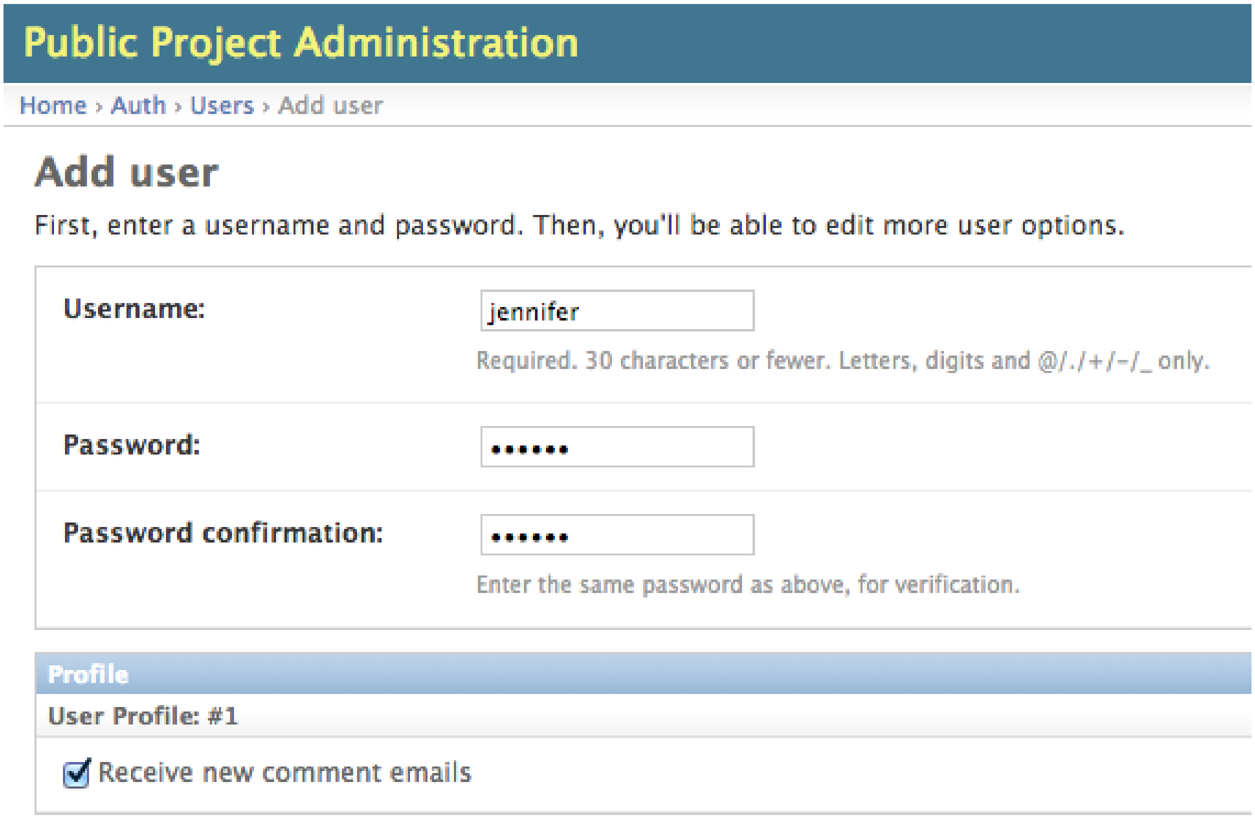


Clicking this button opens a form to enter a new research request.

Administration

User Management

Often it makes sense to work with several people with several accounts on one project. You can add new users in the `User` menu in the admin:



Add user

First, enter a username and password. Then, you'll be able to edit more user options.

Username:	jennifer Required. 30 characters or fewer. Letters, digits and @/./+/-/_ only.
Password:	••••••
Password confirmation:	•••••• Enter the same password as above, for verification.

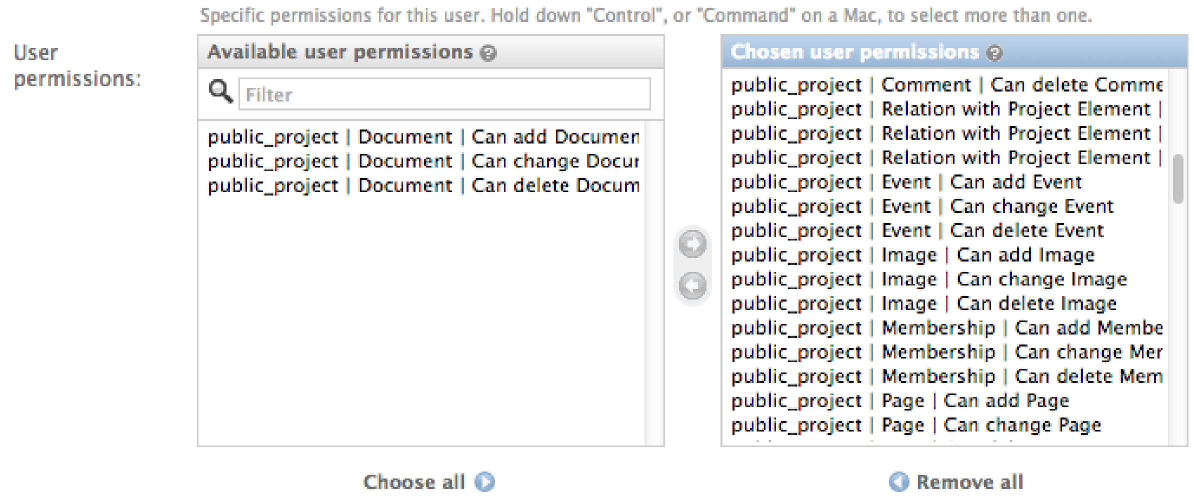
Profile

User Profile: #1

☑ Receive new comment emails

If you leave the Receive new comment emails checkbox checked the user will get emails on new comments and will be able to publish them. After clicking save be sure to activate Staff status on the following Change User form so that the user is able to log into the admin.

Management of user rights is a bit tricky. Since everything in DPP is so interconnected if you just grant some single user permissions to a user it is easy to miss some dependent item prevent the user from finishing a desired task because he/she has not the necessary user rights. For that reason it may be easier choosing all user rights in the first place and then explicitly removing the ones you don't want to grant, e.g. the right to add/edit/delete documents:



Specific permissions for this user. Hold down "Control", or "Command" on a Mac, to select more than one.

User permissions:

Available user permissions

public_project | Document | Can add Documen
public_project | Document | Can change Docur
public_project | Document | Can delete Docum

Chosen user permissions

public_project | Comment | Can delete Comme
public_project | Relation with Project Element |
public_project | Relation with Project Element |
public_project | Relation with Project Element |
public_project | Event | Can add Event
public_project | Event | Can change Event
public_project | Event | Can delete Event
public_project | Image | Can add Image
public_project | Image | Can change Image
public_project | Image | Can delete Image
public_project | Membership | Can add Membe
public_project | Membership | Can change Mer
public_project | Membership | Can delete Mem
public_project | Page | Can add Page
public_project | Page | Can change Page

Backing up the Database

When backing up the database of a DPP installation, it works best to use the `-n` option for saving content type and some ohter references as natural keys and at the same time ommit backing up the contenttypes app and the auth.Permission model. This makes it easier to recover an installation after DB data loss, since Django is automatically generating the content type objects (used in DPP for comments) which could lead to problems with IDs if not using natural keys:

```
python manage.py dumpdata -n -e contenttypes -e auth.Permission > dpp_dump.json
```

When loading the data from a generated dump it is important to comment out the `post_save` signals from the `models.py` file, otherwise an error will occur:

```
python manage.py loaddata dpp_dump.json
```

Developers

Running the Test Suite

Tests for `DPP` are organized in an own django project called `bpw_tests` located under the folder `tests`. For running the test suite the following libraries are required:

- Selenium 2.45+
- django-dynamic-fixture 1.8+

The following types of tests are implemented:

Test Server

A test server is necessary to run some of the tests (e.g. testing RSS feeds). The data for the test server configuration can be created with:

```
./init_test_project.sh
```

This doesn't has to be done every time before running the tests, but at least once and every time after updating the `DPP` library.

The test server can than be started with:

```
./testserver.sh
```

The test server serves a normal `DPP` instance and both the frontend site and the admin should be normally accessible through the browser:

- http://127.0.0.1:8010/
- http://127.0.0.1:8010/admin/

Browser/Selenium Tests

The purpose of Selenium tests is to test the front-end functionality of the site. Tests are organized in the app `browser` and can be run from within the `tests` directory with:

```
python manage.py test browser #whole test suite
python manage.py test browser.GenericTest #one test case
python manage.py test browser.GenericTest.test_main_page #a single test method
```

Testing the 404 Template

When `DEBUG` is set to `True` in `settings.py`, `404 template` can be tested via the following url:

```
http://yourdevelopmenturl/404test/
```

How to contribute: Translation

General How-To

The main area for contribution for this project is translation, since the scope of the software is relatively wide. So if you have got some time, speak English as a base language and another language like Spanish, Russian, French,... you are very welcome to help out (you don't need to be a developer for this task)!

You find the basic english language file called `django.po` on the DPP GitHub Page in the following folder:

```
public_project/locale/en/LC_MESSAGES/
```

Open this file and copy its contents. Then write the translation of the `msg` id strings between the double quotes after the `msstr` attribute. For longer strings you can use a format like this:

```
#: models.py:123
msgid "Structural parts of the project being stable over time."
msgstr ""
"Structural parts of the project being stable over time, e.g. 'Terminals', "
"'Gates', 'Traffic Control', 'Integration of Public Transportation', not too "
"much (<10), often useful as well: one entry for the project as a whole."
```

Just replace the `msgstr` with the translation in your language. If there is already a `msgstr` in english in the `django.po` file, use this string as a translation basis instead of `msgid` and replace the english string with your language translation.

When you are ready with your translation open an issue on GitHub and past your text there or (advanced developer version) make a pull request.

Note: If you have got limited time: please choose accuracy over speed, it's more helpful if you translate 20 strings in an appropriate manner and take some time to think about the translation than translating 50 strings and often missing the context or have spelling errors!

Generating/compiling message files

For generating the message files for a specific locale from the source identifiers, change to the `public_project` app directory and generate the message file for the desired locale with:

```
django-admin.py makemessages -l de
```

Then translate the missing identifier strings and compile the message files with:

```
django-admin.py compilemessages
```

Release Notes

Changes in version 0.7.5-beta (2017-03-16)

- Support for `Django 1.10` (support for older versions dropped)

Changes in version 0.7.4-beta (2017-02-18)

- Bugfixes

Changes in version 0.7.3-beta (2017-02-18)

- Fix tag cloud links on start page for english version
- Removed language-specific (german) date formats for better i18n compatibility

Changes in version 0.7.2-beta (2017-02-11)

- Slugified/more meaningful URLs for topics, questions, events and participants
- New `DPP_CUSTOM_JS` and `DPP_CUSTOM_CSS` settings to include custom Javscript code or CSS styles in the project (see: *Custom JS/CSS Code*)
- Added admin website configuration settings for optionally not showing goals and questions categories and deactivating user comments, new migration `0004_activation_flags_for_goals_questions_comments`

Changes in version 0.7.1-beta (2015-08-31)

- Updated `Bootstrap` from `2.3` to `3.3`
- Improved menu navigation on mobile devices

Changes in version 0.7.0-beta (2015-08-27)

- Support for `Django 1.8` (support for older versions dropped)
- Switched to `Django` internal migrations. South dependencies are removed, but you can still find the old `South` migration files in the `south_migrations` folder. To make sure the update runs smoothly make sure you have applied all `South` migrations from the previous releases. In doubt update to the latest `0.6` release first and run the `migrate` command within `South` context before switching to this release. Then from `0.7` run the `migrate` command with the `--fake-initial` flag: `python manage.py migrate --fake-initial`.
- Updated requirements of various library dependencies
- Fixed a bug for document comments

Changes in version 0.6.3-beta (2014-12-08)

- Fixed some unnecessary error messages caused by crawlers

Changes in version 0.6.2-beta (2014-10-18)

- Minor layout and admin improvements

Changes in version 0.6.1-beta (2014-10-18)

- Layout improvements for sites not using all customizations from new DPP version

Changes in version 0.6-beta (2014-08-21)

- Replaced structuring of participants by participant type with a more flexible concept allowing the **grouping participants to other participants (groups)** by a new attribute `belongs_to` in admin and a new many-to-many model `Membership`. A membership is described by a `function` and a boolean field `active`, connecting two participants. This is replacing the former concept `responsible_participants` and `former_responsible_participants`, which could be found in `Project` tabe. Both fields were removed. DB changes: migrations `0002_auto__del_field_participant_type.py`, `0003_auto.py`, `0008_auto_add_membership.py`.
- **Project Parts (Topics) can now also be hierarchically structured**, every project part object now has a new attribute `main_project_part` allowing to connect project parts to a main topic. This new structure (as well as the participant grouping) will be visible in the frontend as well. DB changes: migration `0004_auto_add_field_projectpart_main_project_part.py`
- **New SiteCategory model** for providing intro texts to the website categories ("Home", "Questions", ...) and connecting documents and websites with categories, **replacing the old model Project (deleted)**. DB changes: migrations `0005_auto_add_sitecategory.py`, `0006_intro_texts_to_site_category.py` (for automatic data transfer from `Project` instance) and `0007_auto_del_project.py`.
- **Direct integration of TinyMCE as HTML editor** for descriptive admin fields by overwriting Django admin templates. `public_project` app in `INSTALLED_APPS` in `settings.py` now has to be placed before (!) Django admin app, new `TEMPLATE_CONTEXT_PROCESSOR` `public_project.context_processors.uploaded_images_list` (also has to be added to `settings.py`) for loading images in Admin to be selectable by TinyMCE editor
- Introduction of new **main category for goals**
- Restructuring, icons and help text for admin, more information on overview pages
- Translation of admin interface
- Many **layout improvements**, overhaul of overview all overview pages with expand/collapse boxes and displaying number of sub elements
- New **universal search box**
- Completely revamped documentation

Changes in version 0.5-alpha (Renaming Release) (2013-05-27)

This release is just for renaming the Django app. Due to the development of the software it came up, that the focus of the software is broader than actually thought, so the name `django-big-projects-watch` (BPW) is misleading and the software was renamed to `django-public-project` (DPP). This comes with a lot of hassle and won't happen again in the lifecyle of this software, but I felt, that in this early stage of the software, it is the only chance to make such a step.

If you already have a deployment of the software installed and have problems upgrading please contact me (@HolgerD77).

On `GitHub` the software moved to a new repository https://github.com/holgerd77/django-public-project with a new commit history. The `South` history has been restarted as well.

Steps to manually upgrade:

1. BACKUP YOUR DATABASE! BACKUP YOUR PROJECT FOLDER!
2. Create a JSON dump of your project with the `-n` option for preserving natural keys, leave out the `South` tables: `python manage.py dumpdata -n -e contenttypes -e auth.Permission -e south > bpw_dpp_dump.json`
3. Rename the suffix of `django-public-project` specific settings in `settings.py` from `BPW` to `DPP`
4. Remove `big_projects_watch` from `INSTALLED_APPS` in your `settings.py` file and add `public_project`.

5. Enter a new database name (for security reasons, leave old DB untouched) in your `settings.py`.
6. Run `python manage.py syncdb`, `python manage.py migrate`, don't create a superuser
7. Search and replace all occurrences of `big_projects_watch` in your JSON DB dump with `public_project` (e.g. in vi use ":%s/big_projects_watch/public_project/g"), keep a copy of the unmodified file!
8. Load your JSON dump in the new DB with `python manage.py loaddata yourjsonfile.json`.
9. Test your application. Sorry for the inconvenience.

Changes in version 0.4-alpha (2013-05-04)

- New **activity feed on main page**, integrating different activities in the system like an admin user adding a new object (e.g. a new event, participant, ...) or an visitor on the website commenting on an object. New model `ActivityLog` (see Migration 0016), activities are always bound to objects in the system, concept is flexible and expandable so that new activities around system objects can be added in the future
- **RSS feeds** for various pages of the system, closely connected to the activity concept. Feeds for the different new system objects, new comments on certain objects, a general activity feed, a general comment feed and a feed for new research requests (see further down)
- **Own pages/urls for questions, expanded editorial possibilities**: every question now has an own url and expanded possibilities to be described, new model fields for `Question` model class (see Migration 0018)
- **Integration of questions in system comments**: questions can now be referenced by site visitors in there comments and questions can be commented itself as well
- **New research requests associated with questions**: site owners can now give research requests to the crowd, describing tasks to be done or information to be found in documents. A research request is always associated with a question and can further - similar to comments - be associated with different system objects. Site admins can directly enter new requests on the associated question page.
- **Experimental version of a public API** Various objects in the system can now be accessed via a public JSON API if desired

Changes in version 0.3-alpha (2013-04-08)

- Layout overhall (category colors, bigger headlines, breadcrumb navigation, UI tweaks)
- `WITH_PUBLIC_DOCS` setting in `settings.py` replaced with `BPW_IE_COMPATIBLE_PDF_VIEWER` (see: *Installation*)
- New detail info boxes for events, documents, used on main page to highlight newest events, documents
- Introduced search tags as new information concept (new DB models `SearchTag`, `SearchTagCacheEntry`, use `South` when upgrading): provided in Django admin for Events, Participants, ProjectParts, used for tag cloud generation and displaying documents containing these search tags on detail pages for Events, Participants, ProjectParts
- Search tag clouds (click induces search) on main page, document pages
- One unified crowdsource concept, merging the former concepts `DocumentRelations` into a broader `Comments` concept. ATTENTION! THESE CHANGES COME ALONG WITH HEAVY DB CHANGES AND NEED MANUAL WORK TO GET THINGS WORKING AGAIN!
  - When upgrading create a dump from your `DocumentRelation`, `Comment` table entries first
  - `DocumentRelation` model is completely removed, entries have to be manually copied into `Comment` table

Changes in version 0.2-alpha (2013-01-22)

- Layout based on Twitter Bootstrap

- Participants, ProjectParts, ProjectGoals, Events as basic project entities
- Modeling of questions around the project
- Document upload / PDF viewer based on pdf.js
- Crowdsourcing of comments / document relations

Changes in version 0.1-pre-alpha (2012-08-08)

- Initial verion

CHAPTER 2

Indices and tables

- genindex
- modindex
- search