e-cidadania Documentation

Release a

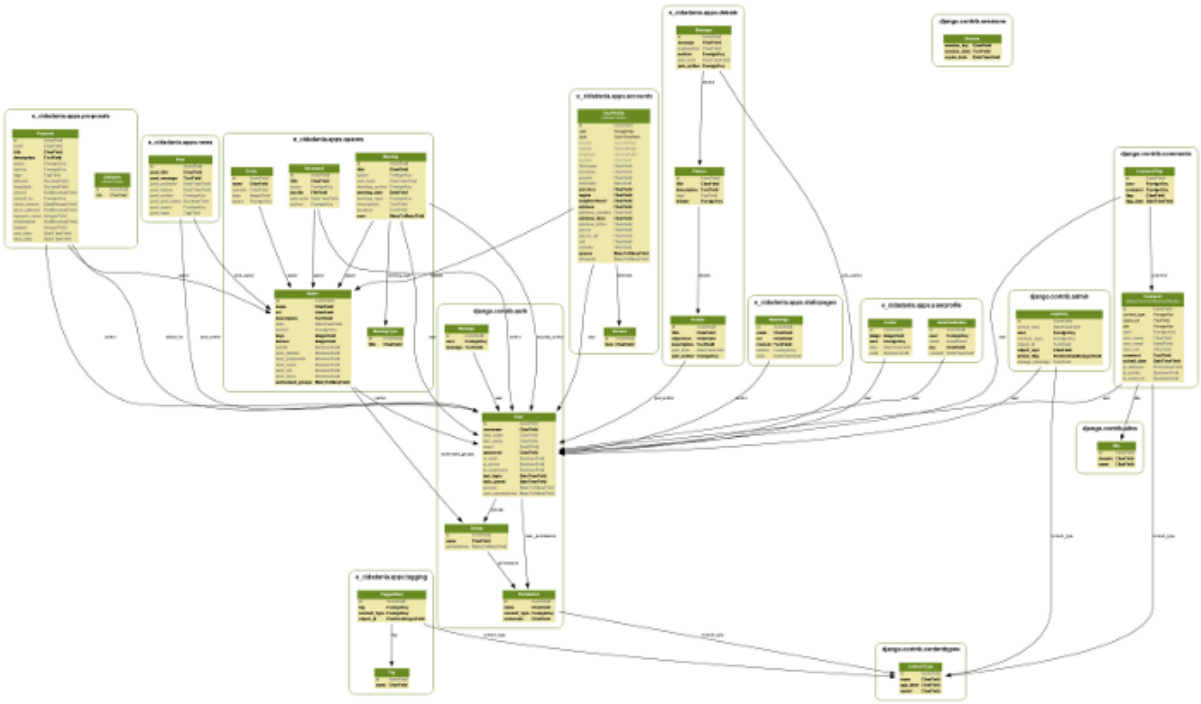
Cidadania S. Coop. Galega

October 26, 2011

Índice xeral

e-cidadania is an open-source e-democracy web platform for citizen participation. The key features are: a proposal system, a revolutionary ordered debate system, documentation repository, autogenerated reports or advanced user profiles with geolocation and messaging.

This software is based on the django framework and some external libraries.



Warning: e-cidadania is in heavy development, and because of that, some parts of it may suffer changes, especially the data models until some releases.

CAPÍTULO 1

Documentation

1.1 Installation

e-cidadania installation is very simple and is done the same as any other django platforms.

1.1.1 Requirements

- Apache, nginx, or any other web server with CGI suppport
- FastCGI, CGI, Passenger or other CGI.

Dependencies

- django 1.3
- PIL *(Python Imaging Library)*
- python-datetime *(versión 1.5)*
- django-tagging
- django-wysiwyg
- django-grappelli (para la administración)
- feedparser

You can install all the required dependencies automatically with this command:

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

1.1.2 Downloading platform

Note: The download section in the official website is not available yet.

The are several ways to download e-cidadania. The most simple of them is going to the downloads page in the website and download the latest stable or development versions, ready to use.

Stable version

You can find the latest stable version in the download page in ecidadania.org:

```
http://ecidadania.org/downloads
```

Development version

Development version is available through various places. We use GIT as control version system, so you will have to install it in your computer.

Gitorious: *(official repository)*:

```
git clone git://gitorious.org/e-cidadania/mainline.git
```

GitHub *(secondary repository)*:

```
git clone git://github.com/oscarcp/e-cidadania.git
```

Repo.or.cz *(official mirror)*:

```
git clone git://repo.or.cz/e_cidadania.git
```

1.1.3 Installing with Apache 2

This section is in developent.

1.1.4 Installing with nginx + FastCGI

Note: The installtion through FastCGI should work for other web servers since FastCGI is the one who handles e-cidadania, and the server only serves the static content.

The installation with nginx and FastCGI...

This section is in development.

Now you can continue to "Configuration".

1.2 Configuration

The e-cidadania platform is almost ready-to-use adter unpacking, but you will have to edit the *settings.py* file.

1.2.1 Database

Configuring the database:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': 'db/sqlite.db',
        'USER': '',
        'PASSWORD': '',
        'HOST': '',
        'PORT': '',
    }
}
```

First of all will be to set up the database. By default e-cidadania is set up to use a local SQLite3 database, which will be useful if Lo primero de todo será configurar la base de datos. Por defecto, e-cidadania viene configurado para utilizar una base de datos local SQLite3, que puede servirte para hacer pruebas si lo necesitas pero no se debe utilizar bajo ningún concepto en producción.

Un ejemplo de base de datos en un servidor compartido de DreamHost es este:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'ecidadania_database',
        'USER': 'databaseadmin',
        'PASSWORD': 'somepassword',
        'HOST': 'mysql.ecidadania.org',
        'PORT': '',
    }
}
```

1.2.2 Modo Debug

El modo debug viene activado por defecto y se recomienda encarecidamente desactivarlo para comenzar a utilizar e-cidadania en producción. Para ello hay que desactivarlo en el fichero *settings.py*:

```
DEBUG = False
```

1.2.3 Perfiles de usuario

ACCOUNT_ACTIVATION_DAYS (número)

Esta variable especifica cuándos días tiene el usuario para activar su cuenta desde que recibe el correo de confirmación.

GOOGLE_MAPS_API_KEY (hash)

Llave de la API de Google para poder utilizar la interfaz de mapas. Debes crearte una propia a pesar de que e-cidadania venga con una configurada, ya que sólo funcionará en el dominio que hayas especificado.

1.2.4 Correo electrónico

ADMINS (lista)

Lista de adminsitradores y cuentas de correo para la notificación de errores del servidor. Sólo funciona si DEBUG = False

EMAIL_HOST (servidor)

Servidor de correo desde el cual se enviarán los correos a los usuarios.

DEFAULT_FROM_EMAIL

Dirección por defecto desde la que se enviarán los correos si no se especifica otra.

1.2.5 Idioma

LANGUAGE_CODE (código de idioma)

Idioma con el cual funcionará django por defecto.

1.2.6 Plugins

Note: Esta sección está sin redactar.

1.3 Manual de usuario

1.3.1 Cómo registrarse

El sistema de registro de usuarios dependerá de lo que el administrador quiera. Por defecto, e-cidadania permite el registro de usuarios para poder realizar algunas tareas básicas, y visitar los espacios públicos, pero sin poder participar.

Para registrarte debes hacer clic en "Entrar" y el la parte inferior del formulario de entrada te aparecerá la opción de registro. Basta con tu nombre, correo electrónico y clave.

1.3.2 Cómo enviar una propuesta

Enviar una propuesta es de lo más sencillo. Basta con acceder al espacio de participación correspondiente y hacer clic en "Añadir una propuesta" en la columna de la derecha.

Aparecerá un formulario sencillo que tiene varios campos:

Título El título de la propuesta. Debe ser una síntesis concisa de la propuesta.

Descripción Una descripción más extensa de la propuesta. Se pueden vincular elementos externos como páginas web, imágenes, vídeos, etc.

Latitud/Longitud Ahora mismo e-cidadania cuenta con un sistema básico de geoposicionamiento. Si quieres situar tu propuesta en un mapa tendrás que poner las coordenadas.

Etiquetas Las etiquetas identifican de forma sencilla tu propuesta, por ejemplo:

```
Titulo: La acera de Cobián Roffignac está rota
Etiquetas: acera, cobian roffignac, pontevedra, roto
```

1.3.3 Cómo participar en un debate

El sistema de debates es nuevo, así que procura prestar atención para no perderte, esto no es un foro de internet.

Se han trasladado diversos modelos de debate presencial hasta esta plataforma, y uno de ellos es el que te vamos a enseñar ahora. ¡Ya verás qué rápido te acostumbras!

1.3.4 Cómo consultar documentos

Basta con hacer clic en el documento que quieras leer para que comience a descargarse. Si el documento que quieres no figura en la columna de la derecha puedes darle al botón "Ver todos los documentos" y se cargará una página nueva que te mostrará todos los documentos que están almacenados en ese espacio.

1.3.5 Preguntas frecuentes

Si tienes alguna pregunta frecuente que no está aquí por favor dínoslo!

1.4 Administration manual

This is a small introoductory manual to teach you about how to use e-cidadania without messing everything up :).

1.4.1 User registration

Note: This section is obsolete.

User registration in version *0.1 alpha* is done manually due to the lack of a secure authentication mechanism, except the electronic certificate, that is not too much extended.

Anyway, e-cidadania has from the start a basic automated registration system, that the administrator will have to activate when he considers removing the comment symbol (sharp).

apps/userprofile/urls/en.py:107:

```
# url(r'^register/$', register, name='signup'),
```

If the platform is well set up, the registration system should take care of everything.

1.4.2 Permissions

Permissions in e-cidadania are inherited directly from the django auth system. This way we have group based permissions and user based permissions. For the first release is enough, but **it would be extremely recommended to not use the platform yet, if security if your priority**.

e-cidadania 0.2 will have a row-level permission system, more detailed and secure than the current ones.

1.4.3 Groups

Groups are a massive way to give permissions to people. In this version groups will be a way to group people inside the spaces, except that for any reason you'll have to give them another permission for a specific task.

1.4.4 Spaces

Spaces are where partivipative processes take place.

1.4.5 Modules

e-cidadania is a modular platform. Even its basic features (news, documents, spaces, proposals) are modules that can be replaced at any moment without affecting the general application structure.

Moderation

Moderation tasks inside the platform are very simple. Every module has three basic tasks which are: creation, editing and deletion.

Creation Depending on the moderation grade that you have been given, you can add simple content or more complex. The highest moderation leves have a very high detail grade when adding content.

Editing The editing task is similar to creation, it will return a formulary with the current data based on your permission level.

Deletion Usually in forums a moderator can delete user entries. In e-cidadania that is not the objective. User generated content must be preserved, it only can be deleted by platform administrators.

1.4.6 Frequent errors

The most frequest errors are due to the server or a bad administrator management in the groups/permission case.

CAPÍTULO 2

Development

2.1 Style guide

The style guide stablish a series of rules to follow when coding in e-cidadania. This rules are unbreakable. The style guide follows closely the PEP8 document, with some exceptions provided from the internal style guide at Pocoo.

2.1.1 Python

Imports Every import must be situated in the file header, below to the comment header. The python imports must precede any others, and the external librearies or third party modules must precede the application ones.

Ejemplo:

```
import os
import sys

from extlib import function

from myapp.module import function
```

Line width (columns) The code must be always 80 columns wide except on templates.

Long declarations If a code line does not fit in 80 columns, try to reduce it declaring variables previously. If it still can not fit, you can divide the lines this way:

Parentheses:

```
website = models.URLField(_('Website'), verify_exists=True,
                          max_length=200, null=True, blank=True,
                          help_text=_('The URL will be checked'))
```

Declararations:

```
this_is_a_very_long(function_call, 'with many parameters') \
    .that_returns_an_object_with_an_attribute

MyModel.query.filter(MyModel.scalar > 120) \
             .order_by(MyModel.name.desc()) \
             .limit(10)
```

Lists, tuples and dictionariess:

```
items = [
    'this is the first', 'set of items', 'with more items',
    'to come in this line', 'like this'
]

dict = {
    ('mobile': phone),
    ('car': key),
    ('another': thing),
}
```

Indentation Indentation must be *always* 4 spaces per level, no exceptions. You can not use tabs for indentig.

Blank lines Every function and classes must be separated by two blank lines. The code inside a class or method by one blank line.

Example:

```
class ListDocs(ListView):
    ----blank line----
    """
    List all documents stored whithin a space.
    """
    paginate_by = 25
    context_object_name = 'document_list'
    ----blank line----
    def get_queryset(self):
        place = get_object_or_404(Space, url=self.kwargs['space_name'])
        objects = Document.objects.all().filter(space=place.id).order_by('pub_date')
        return objects
    ----blank line----
    def get_context_data(self, **kwargs):
        context = super(ListDocs, self).get_context_data(**kwargs)
        context['get_place'] = get_object_or_404(Space, url=self.kwargs['space_name'])
        return context
----blank line----
----blank line----
def whatever(args):
    ----blank line----
    """
    A comment.
    """
    this_is_something = 0
```

2.1.2 HTML

Columns HTML code does not have a column limit, but it must be indented in a way we can locate easily every element inside the document. The indentation preceeds rendered results in application.

Indentation The X/HTML code must be indented with two spaces, no exception.

2.1.3 CSS

Indentation Indentation will be 4 spaces, always, like Python code.

Example:

```
body {
    background: #FAFAFA;
    padding: 0;
    margin: 0;
    font-family: Verdana, "Lucida Sans", Arial;
```

```
        font-size: 1em;
        color: #000;
        cursor: default;
    }
```

Colors Colors must be always wrote in hexadecimal. You are allowed to use three digits abbreviations.

Font size Font size must be declared in **em's** except a presentation requirement.

2.1.4 JavaScript

Estilo de código JavaScript.

2.2 User accounts

Th user account system in e-cidadania is abased on the django *auth* module and in django-userprofile, created by James Bennet.

2.2.1 django-userprofile

django-userprofile provides the views and functions to extend the user data model in django. Together with a module created for extending the data model everything goes fine.

2.2.2 accounts

The accounts module is our extended user data model. In it you can find all the extra fields that are needed and will be added to *django-userprofile*.

2.3 Creating modules

e-cidadania is extensible through modules, even if the installation procesdures are not working yet.

2.3.1 Structure

A module is basically a django application which we integrate in e-cidadania. At this time we advocate for the django default structure in the distribution and file names.

2.4 Generating the documentation

e-cidadania documentation is generated through sphinx (1.0.7) in three languages by default, which are:

- English
- Spanish
- Galician

2.4.1 Linguistic rules

This are the linguistic rules for e-cidadania documentation.

2.4.2 Fuzzy words

Fuzzy words dictionary.

2.5 Translations

For the translation we use two tools:

- django-rosetta
- gettext

Both ways of translating are simple thaks to the Django *middleware*.

2.5.1 Translating with rosetta

To make translations with rosetta, you will need to have an account on the system and belong to the **'translators'** group. Once you have done that, the rest is simple.

Just access to the translation URL and the first you will see is a list with the languages available to translate.

Rosetta

Inicio › Language selection

Espanol

Application	Progress	Mensaxes	Translated	Fuzzy	Obsolete	File
Debate	100,00%	5	5	0	0	/home/oscar.carballal/devel/e
Spaces	100,00%	63	63	0	0	/home/oscar.carballal/devel/e
E_Cidadania	100,00%	65	65	0	0	/home/oscar.carballal/devel/e
Accounts	100,00%	24	24	0	0	/home/oscar.carballal/devel/e
News	100,00%	23	23	0	0	/home/oscar.carballal/devel/e
Proposals	100,00%	20	20	0	0	/home/oscar.carballal/devel/e
Userprofile	38,00%	375	145	0	0	/home/oscar.carballal/devel/e

inglés

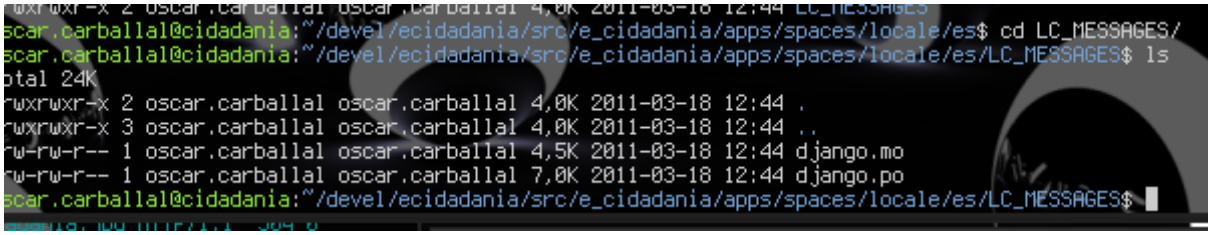
Application	Progress	Mensaxes	Translated	Fuzzy	Obsolete	File
Spaces	0,00%	27	0	0	0	/home/oscar.carballal/devel/e
News	0,00%	9	0	0	0	/home/oscar.carballal/devel/e
Userprofile	0,00%	375	0	0	0	/home/oscar.carballal/devel/e

Click on the component you want to translate and start translating (the translation is done from english to other languages). If you find yourself stuck you can use the option "Suggest" which will make a query to the Google Translate database and write for you the translation.

> **Warning:** Never trust the results of the "suggest" button. In most occasions will be incorrect.

2.5.2 Translating with gettext

Gettext is a well known tool by all the translators around the world. Its a standard. Thanks to the django *middleware* out work will be minimum, we only have to edit the .po files in the source files.



Instead of making one global translation, we decided to keep a translation file for every module, that way the translations will keep even if the modules are moved.

The location of the strings is usually a directory called **locale** inside the module. Inside it, you can find directories with the country code (en, es, us, gl, fr, etc.) and inside this one, the PO and MO files.

To translate, you must edit the PO file, which is a plain text file.

The MO file is the compiled version of the translation so the machine can read it to use it.

Warning: Stablish a workflow for translators and explain it here.

CAPÍTULO 3

Appearance / Themes

3.1 Apariencias de e-cidadania

3.1.1 Otras apariencias

Este es un pequeño índice de apariencias de e-cidadania.

3.1.2 Crea tu propia apariencia para e-cidadania

Crear una nueva apariencia para e-cidadania puede llegar a ser algo difícil, pero no por ello imposible. Aquí te daremos las instrucciones precisas.

3.1.3 Referencia

Los temas de e-cidadania se localizan mayoritariamente en los propios módulos. Cada aplicación gestiona por separado sus apariencias, no en cuanto a estilo pero sí en cuanto a distribución.

Además de esto, existen algunas platillas generales que están situadas en el directorio *templates*.

General

asdasd

Perfil de usuario

asdasdas

Propuestas

sdfsdfs

Debates

sdfsdfsd

Noticias

sdfsdf

Documentos

CAPÍTULO 4

Reference

4.1 spaces.admin — Spaces administration models

4.2 spaces.forms — Space-related forms

This module contains all the space related forms, including the forms for documents, meetings and entities. Most of the forms are directly generated from the data models.

class e_cidadania.apps.spaces.forms.**SpaceForm**(*ModelForm*)
Returns a form to create or edit a space. SpaceForm inherits all the fields from the Space data model.

Return type HTML Form

New in version 0.1.

class e_cidadania.apps.spaces.forms.**EntityForm**(*ModelForm*)
Returns a form to create or edit an entity, based on the :class:Entity data model.

Return type HTML Form

New in version 0.1.

class e_cidadania.apps.spaces.forms.**DocForm**(*ModelForm*)
Returns a form to create or edit a space related document, based on the spaces.Document data model.

Return type HTML Form

New in version 0.1.

class e_cidadania.apps.spaces.forms.**MeetingForm**(*ModelForm*)
Returns a form to create or edit a space related meeting, based on the spaces.Meeting data model.

Return type HTML Form

New in version 0.1.

4.3 spaces.models — Spaces data models

class e_cidadania.apps.spaces.models.**Space**(*models.Model*)
Spaces model. This model stores a "space" or "place". Every place has a minimum set of settings for customization.

class e_cidadania.apps.spaces.models.**Entity**(*models.Model*)
This models stores the name of the entities responsible for the creation of the space.

class `e_cidadania.apps.spaces.models.`**`Document`** (*models.Model*)
Document model

class `e_cidadania.apps.spaces.models.`**`MeetingType`** (*models.Model*)
Meeting type model. This will give enough flexibility to add any type of meeting in any space.

class `e_cidadania.apps.spaces.models.`**`Meeting`** (*models.Model*)
Meeting data model. Every space (process) has N meetings. This will keep record of the assistants, meeting name, etc.

4.4 `spaces.views` — Space-related views

These are the views that control the spaces, meetings and documents.

4.4.1 General spaces views

class `e_cidadania.apps.spaces.views.`**`ListPosts`** (*ListView*)
Returns a list with all the posts attached to that space. It's similar to an archive, but without classification or filtering.

Return type Object list

Context post_list

4.4.2 Spaces views

class `e_cidadania.apps.spaces.views.`**`GoToSpace`** (*RedirectView*)
Sends the user to the selected space. This view only accepts GET petitions. GoToSpace is a django generic `RedirectView`.

Attributes **self.place** - Selected space object

Return type Redirect

class `e_cidadania.apps.spaces.views.`**`ListSpaces`** (*ListView*)
Return a list of spaces in the system (except private ones) using a generic view. The users associated to a private spaces will see it, but not the other private spaces. ListSpaces is a django generic `ListView`.

Return type Object list

Contexts object_list

class `e_cidadania.apps.spaces.views.`**`ViewSpaceIndex`** (*DetailView*)
Returns the index page for a space. The access to spaces is restricted and filtered in the get_object method. This view gathers information from all the configured modules in the space.

Attributes space_object, place

Return type Object

Context get_place, entities, documents, proposals, publication

class `e_cidadania.apps.spaces.views.`**`DeleteSpace`** (*DeleteView*)
Returns a confirmation page before deleting the space object completely. This does not delete the space related content. Only the site administrators can delete a space.

Return type Confirmation

`e_cidadania.apps.spaces.views.`**`edit_space`** (*request*, *space_name*)
Returns a form filled with the current space data to edit. Access to this view is restricted only to site and space administrators. The filter for space administrators is given by the edit_space permission and their belonging to that space.

Attributes

- place: current space intance.
- form: SpaceForm instance.
- form_uncommited: form instance before commiting to the DB, so we can modify the data.

Parameters **space_name** – Space URL

Return type HTML Form

Context form, get_place

`e_cidadania.apps.spaces.views.`**`create_space`**(*request*)

Returns a SpaceForm form to fill with data to create a new space. There is an attached EntityFormset to save the entities related to the space. Only site administrators are allowed to create spaces.

Attributes

- space_form: empty SpaceForm instance
- entity_forms: empty EntityFormSet

Return type Space object, multiple entity objects.

Context form, entityformset

4.4.3 Document views

class `e_cidadania.apps.spaces.views.`**`ListDocs`**(*ListView*)

Returns a list of documents attached to the current space.

Return type Object list

Context object_list, get_place

class `e_cidadania.apps.spaces.views.`**`DeleteDocument`**(*DeleteView*)

Returns a confirmation page before deleting the current document.

Return type Confirmation

Context get_place

`e_cidadania.apps.spaces.views.`**`add_doc`**(*request*, *space_name*)

Upload a new document and attach it to the current space.

Return type Object

Context form, get_place

`e_cidadania.apps.spaces.views.`**`edit_doc`**(*request*, *space_name*, *doc_id*)

Returns a DocForm filled with the current document data.

Return type HTML Form

Context doc, get_place

4.4.4 Meeting views

class `e_cidadania.apps.spaces.views.`**`ListMeetings`**(*ListView*)

List all the meetings attached to a space.

Return type Object list

Context meeting_list, get_place

class `e_cidadania.apps.spaces.views.`**`ViewMeeting`**(*DetailView*)
View the content of a Meeting.

Return type Object

Context meeting, get_place

class `e_cidadania.apps.spaces.views.`**`DeleteMeeting`**(*DeleteView*)
Returns a confirmation page before deleting the Meeting object.

Return type Confirmation

Context get_place

`e_cidadania.apps.spaces.views.`**`add_meeting`**(*request*, *space_name*)
Returns an empty MeetingForm to create a new Meeting. Space and author fields are automatically filled with the request data.

Return type HTML Form

Context form, get_place

`e_cidadania.apps.spaces.views.`**`edit_meeting`**(*request*, *space_name*, *meeting_id*)
Returns a MeetingForm filled with the current Meeting data to be edited.

Return type HTML Form

Context meeting, get_place

4.5 `proposals.admin` — Proposal administration models

4.6 `proposals.forms` — Proposal forms

4.7 `proposals.models` — Proposal data models

Proposal data models are the ones to store the data inside the DB.

class `e_cidadania.apps.proposals.models.`**`BaseClass`**(*models.Model*)
Abstract base class for titles and descriptions (dummy models)

class `e_cidadania.apps.proposals.models.`**`Category`**(*BaseClass*)
Dummy class for proposal categories. Inherits directly from `BaseClass` without adding any fields.

class `e_cidadania.apps.proposals.models.`**`Proposal`**(*models.Model*)
Proposal data model. This will store the user proposal in a similar way that Stackoverflow does. Take in mind that this data model is very exhaustive because it covers the administrator and the user.

Automatically filled fields Space, Author, Pub_date, mod_date.

User filled fields Title, Description, Tags, Latitude, Longitude.

Admin fields (manual) Code, Closed, Close_reason, Anon_allowed, Refurbished, Budget.

Admin fields (auto) Closed_by

Extra permissions proposal_view

`CLOSE_REASONS` for :class:Proposal data model is hardcoded with four values, which will fit most of the requirements.

4.8 `proposals.views` — Proposal views

CAPÍTULO 5

Finale

5.1 Getting help

- Try looking the FAQ.
- e-cidadania mailing lists
  - ecidadania-users@freelists.org
  - ecidadania-dev@freelists.org
  - ecidadania-es@freelists.org
- Bug reporting in the e-cidadania bug tracker.

5.2 About this document

Authors Oscar Carballal <oscar.carballal@cidadania.coop> <info@oscarcp.com>

Galician translation Armando Broz Fidalgo <armando.broz@cidadania.coop>

Rosa Muñiz Castro <rosa.munhiz@cidadania.coop>

English translation Oscar Carballal Prego <oscar.carballal@cidadania.coop> <info@oscarcp.com>

5.3 Thanks

- Algueirada
- Xunta de Galicia
- CESGA (Centro de Supercomputación de Galicia)
- Concello de Ferrol
- NovaCaixaGalicia

Python Module Index

e