

Zend Django App

Rubén Ramírez

24 de marzo de 2020

Índice general

1. Modelos y Autenticación	1
1.1. django.contrib.auth.backends	1
1.2. django.contrib.auth.models	1
1.3. django.contrib.auth.validators	1
2. Base de Datos	3
2.1. auth_group	3
2.2. auth_group_permissions	3
2.3. auth_permission	3
2.4. auth_user	4
2.5. auth_user_groups	4
2.6. auth_user_user_permissions	4
2.7. django_admin_log	4
2.8. django_content_type	4
2.9. django_migrations	5
2.10. django_session	5
3. Diagrama de Clases	7
3.1. django.contrib.auth.models	7
3.2. django.contrib.contenttypes.models	8
3.3. django.contrib.auth.base_user	8
3.4. Otros	8
3.5. zend_django.functional_tests.utils_test	9
4. Utilerias HTML	11
4.1. zend_django.templatetags.html_helpers	11
4.1.1. Funciones	11
4.1.2. Inclusion Tags	11
4.2. zend_django.templatetags.op_helpers	11
4.2.1. CRUDs	11
4.2.2. Actions	11
4.2.3. Simple Tags	12
4.3. html_struct.html	12
4.4. toolbar.html	13
4.5. form.html	13
5. Utilerías JS	15
5.1. zend_django.js	15
5.1.1. Date	15
5.1.2. Number	15
5.1.3. clsApp : App	15

Índice de figuras

4.1. html_struct.html	12
4.2. forms.html	14

Capítulo 1

Modelos y Autenticación

1.1. `django.contrib.auth.backends`

- `BaseBackend`
- `ModelBackend`
- `AllowAllUsersModelBackend`
- `RemoteUserBackend`
- `AllowAllUsersRemoteUserBackend`

1.2. `django.contrib.auth.models`

- `User`
- `UserManager`
- `AnonymousUser`
- `Permission`
- `Group`

1.3. `django.contrib.auth.validators`

- `ASCIIUsernameValidator`
- `UnicodeUsernameValidator`

Capítulo 2

Base de Datos

2.1. auth_group

Campo	Tipo	Pk	NN	U	Referencia
id	integer	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
name	varchar(150)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

2.2. auth_group_permissions

Campo	Tipo	Pk	NN	U	Referencia
id	integer	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
group_id	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	auth_group.id
permission_id	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	auth_permission.id

Grupos únicos:

- (group_id, permission_id)

2.3. auth_permission

Campo	Tipo	Pk	NN	U	Referencia
id	integer	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
content_type_id	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	django.content_type.id
codename	varchar(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
name	varchar(255)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Grupos únicos:

- (content_type_id, codename)

2.4. auth_user

Campo	Tipo	Pk	NN	U	Referencia
id	integer	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
password	varchar(128)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
last_login	datetime	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
is_superuser	bool	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
username	varchar(150)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
first_name	varchar(30)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
email	varchar(254)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
is_staff	bool	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
is_active	bool	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
date_joined	datetime	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
last_name	varchar(150)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

2.5. auth_user_groups

Campo	Tipo	Pk	NN	U	Referencia
id	integer	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
user_id	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	auth_user.id
group_id	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	auth_group.id

Grupos únicos:

- (user_id, group_id)

2.6. auth_user_user_permissions

Campo	Tipo	Pk	NN	U	Referencia
id	integer	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
user_id	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	auth_user.id
permission_id	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	auth_permission.id

Grupos únicos:

- (user_id, permission_id)

2.7. django_admin_log

Campo	Tipo	Pk	NN	U	Referencia
id	integer	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
action_time	datetime	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
object_id	text	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
object_repr	varchar(200)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
change_message	text	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
content_type_id	integer	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	django_content_type.id
user_id	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	auth_user.id
action_flag	smallint unsigned	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

2.8. django_content_type

Campo	Tipo	Pk	NN	U	Referencia
id	integer	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
app_label	varchar(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
model	varchar(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Grupos únicos:

- (app_label, model)

2.9. django_migrations

Campo	Tipo	Pk	NN	U	Referencia
id	integer	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
app	varchar(255)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
name	varchar(255)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
applied	datetime	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

2.10. django_session

Campo	Tipo	Pk	NN	U	Referencia
session_key	varchar(40)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
session_data	text	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
expire_date	datetime	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Capítulo 3

Diagrama de Clases

3.1. django.contrib.auth.models

User
username: string(150) first_name: string(30) last_name: string(30) email: email password: hash groups: Group[] user_permissions: Permission[] is_staff: boolean is_active: boolean is_superuser: boolean last_login: datetime date_joined: datetime
is_authenticated: boolean is_anonymous: boolean
get_username(): string get_full_name(): string get_short_name(): string set_password(raw_password) check_password(raw_password): boolean set_unusable_password() has_usable_password(): boolean get_user_permissions(obj=None): string[] get_group_permissions(obj=None): string[] get_all_permissions(obj=None): string[] has_perm(perm, obj=None): boolean has_perms(perm[], obj=None): boolean has_module_perms(app_label): boolean email_user(subject, message, from_email=None, **kwargs)
UserManager: BaseUserManager
create_user(username, email=None, password=None, **extra_fields): User create_superuser(username, email=None, password=None, **extra_fields) with_perm(perm, is_active=True, include_superusers=True, backend=None, obj=None): User[]
Permission
name: string content_type: +django_content_type codename: string

Group
name: string
permissions: Permission[]

Formatos:

- perm: “app_label.permission_codename”

3.2. django.contrib.contenttypes.models

ContentType
app_label: string
model: string
name: string
get_object_for_this_type(**kwargs): <i>ModelClassObjects</i>
model_class(): <i>ModelClass</i>

ContentTypeManager
clear_cache()
get_for_id(id): ContentType
get_for_model(model, for_concrete_model=True): ContentType
get_for_models(*models, for_concrete_model=True): dict
get_by_natural_key(app_label, model): ContentType

3.3. django.contrib.auth.base_user

AbstractBaseUser

UserManager

3.4. Otros

BaseUserManager
<i>static</i> normalize_email(email): string
get_by_natural_key(username): User
make_random_password(length=10, allowed_chars= 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ23456789'): string

3.5. zend_django.functional_tests.utils_test

FunctionalTest: StaticLiveServerTestCase
main_model_name: string = "" base_data_model: model = None duplicar: string = "alfa" actualizar1: string = "beta" actualizar2: string = "beta_002"
setUp() tearDown() XPathFind(xpath, multiple=False, base_object = None) Wait2PresenceOf(xpath, seconds=30) t_list(search_text="alfa") t_list_to_crud_pages() t_read_to_crud_pages() t_update_right()

URLsTests: SimpleTestCase
model_name: string = "" main_views: views_modelule = None
t_url_resolves(url, view) t_list_url_resolves() t_crerate_url_resolves() t_update_url_resolves() t_delete_url_resolves() t_read_url_resolves()

ViewsTests: TestCase
<pre> model_name: string = "" duplicar: string = "alfa" actualizar1: string = "beta" actualizar2: string = "beta.002" inexistente: string = "inexistente" idinexistente: int = 99999 main_views: views_module = None campo_base: string = "" base_data_model: model = None objs: data_object = [] </pre>
<pre> getData(obj) preSetUp() t_list_get_post(method="get") t_list_post_searching() t_list_post_no_searching() t_list_post_searching_inexistent() t_list_post_no_searching_inexistent() t_read_get_existente(template_file="zend_django/html/form.html") t_read_get_inexistente() t_read_post(id) t_create_get_post(looking_for, method="get", template_file="zend_django/html/form.html") t_create_post_well(data=None) t_create_post_duplicating(looking_for, data=None, template_file="zend_django/html/form.html") t_update_get_post(id, looking_for, method="get", template_file="zend_django/html/form.html") t_update_get_inexistente() t_update_post_well(data=None) t_update_post_duplicating(looking_for, data=None, template_file="zend_django/html/form.html") t_update_post_inexistente_empty() t_update_post_inexistente(data) t_update_post_inexistente_well() t_update_post_inexistente_duplicating() t_delete_get_existente(obj) t_delete_get_inexistente() t_delete_post(id) </pre>

Capítulo 4

Utilerias HTML

4.1. zend_django.templatetags.html_helpers

4.1.1. Funciones

`get_apps()` : `list`

4.1.2. Inclusion Tags

```
{% tag %}

generate_get_css_apps() : {'apps': list_obj}
generate_get_js_apps() : {'apps': list_obj}
requiere_ui_css(context) : {'apps': ['jquery-ui']} | {}
requiere_ui_js(context) : {'apps': ['jquery-ui.min', 'datepicker-es'], 'req-ui': True} | {}

context.request.META["HTTP_USER_AGENT"]
```

4.2. zend_django.templatetags.op_helpers

4.2.1. CRUDs

- `create`
- `create_f`
- `read`
- `update`
- `delete`
- `list`

4.2.2. Actions

- `save`
- `ok`
- `cancel`
- `send_mail`
- `call`
- `send_whatsapp`
- `reset_password`

4.2.3. Simple Tags

```
{% tag %}
```

```
crud_icon(operation): string
crud_label(operation): string
crud_smart_button(operation): string
```

```
action_icon(action): string
action_label(action): string
action_smart_button(operation): string
```

4.3. html_struct.html

Llamada como	zend_django/html/html_struct.html
Carga	html_helpers op_helpers parametros_helpers

```
<!doctype html>

<head>

  <title>{% parametro_de_sistema seccion='SitioGeneral' nombre='site_title' %}</title>

  <link rel="icon" href="{{ MEDIA_URL }}{% parametro_de_sistema seccion='SitioGeneral' nombre='favicon' %}" type="image/png" />

  {% generate_get_css_apps %}
  {% requiere_ui_css %}

  <script type="text/javascript">
    let BASE_URL = '{{ request.get_host }}'
  </script>

</head>

<body>

  {% include "zend_django/menupopo/main_menu.html" %}

  {% block extraaftermain %}
  {% endblock %}

  .container

    {% include "zend_django/html/toolbar.html" %}

    <h1>
      {{ titulo | safe }}
      <small class="text-muted">
        {{ titulo_descripcion | safe }}
      </small>
    </h1>

    {% block extrabefore %}
    {% endblock %}

    {% block content %}
    {% endblock %}

    {% block extraafter %}
    {% endblock %}

  {% include "zend_django/html/footer.html" %}

  {% generate_get_js_apps %}
  {% requiere_ui_js %}

  {% if req_chart %}
    <script type="text/javascript" src="{% static 'api/js/chart-272.min.js' %}"></script>
  {% endif %}

  {% if read_only %}
    App.setReadOnlyForm();
  {% endif %}
```

Figura 4.1: html_struct.html

4.4. toolbar.html

```

toolbar = [
{'type': 'rlink',      'label': 'Etiqueta',      'url': 'http://...'},
{'type': 'link',      'label': 'Etiqueta',      'view': 'NombreVista'},
{'type': 'link_pk',   'label': 'Etiqueta',      'view': 'NombreVista', pk: 'Valor'},
{'type': 'link_pk_del', 'label': 'Etiqueta',      'view': 'NombreVista', pk: 'Valor'},
{'type': 'button',    'label': 'Etiqueta',      'onclick': 'CodigoJS'},
{'type': 'search'},
]

```

4.5. form.html

Llamada como	zend_django/html/form.html
Extiende	zend_django/html/html_struct.html
Carga	crispy_forms_tags op_helpers

```

forms = {
    'top': [{ 'title': 'XXXX', 'form': form_object }, ... ],
    'left': [...],
    'right': [...],
    'bottom': [...],
}

```

#main-form[enctype="multipart/form-data"][method="post"]

forms.top

form_obj.title

☐ Select fields to update status

Position
☐ Senior Engineer ok

Company
☐ Tesla Motors ok

Working from
☐ October ▼ 2011 ▼ — Working to Current time
 December ▼ 2012 ▼ ok

forms.left

form_obj.title

☐ Select fields to update status Mark all as OK

☐ Engineering ok

☐ Thinking ok

☐ Working ok

☐ Sleeping ok

☐ Eating ok

☐ Walking ok

forms.right

form_obj.title

☐ Select fields to update status Mark all as OK

☐ Engineering ok

☐ Thinking ok

☐ Working ok

☐ Sleeping ok

☐ Eating ok

☐ Walking ok

forms.bottom

form_obj.title

☐ Select fields to update status

Position
☐ Senior Engineer ok

Company
☐ Tesla Motors ok

Working from
☐ October ▼ 2011 ▼ — Working to Current time
 December ▼ 2012 ▼ ok

{% action_icon 'save' %} {% action_label 'save' %}

Figura 4.2: forms.html

Capítulo 5

Utilerías JS

5.1. zend_django.js

5.1.1. Date

String asMySQL() Devuelve la fecha en formato MySQL YYYY-MM-DD

String asMx() Devuelve la fecha en formato europeo DD-MM-YYYY

String theTime() Devuelve la hora en formato hh:mm

Date fromMX(String date) Recibe una cadena en formato DD-MM-YYYY y devuelve el objeto tipo Date correspondiente

Date addDays(int dias) Suma un numero de días a la fecha

5.1.2. Number

String asMoney() Devuelve el número como cadena en formato de moneda (con dos decimales)

5.1.3. clsApp : App

void checkInputIn(String idcontainer) Activa todas las casillas de verificación contenidas en el contenedor con id = idcontainer

void uncheckInputIn(String idcontainer) Desactiva todas las casillas de verificación contenidas en el contenedor con id = idcontainer

void openPanel(String body, String title, boolean close=true, String footer=null,

String idmodal="modal-panel-message") Genera un nuevo panel bootstrap con el contenido de body y el identificador indicado en idmodal

void closePanel(String idmodal="modal-panel-message") Cierra el panel bootstrap identificado por idmodal

void setUIControls Genera los controles UI para navegadores que no soportan controles HTML5 para controles input tipo date

void setReadOnlyForm(String container_selector="#main-form") Aplica readonly a los controles de formulario contenidos en el contenedor indicado por container_selector

void showPrivacyPolicy() Abre en un panel bootstrap el contenido de politica de privacidad, establecido en el elemento HTML con el selector *#privacy-policy-template*

void showDeletingConfirmation(String url, String elemento="elemento", String pre_elemento="el") Muestra el panel bootstrap de confirmación de eliminación de 'el' 'elemento', el boton de *Aceptar* es un hipervínculo a url

boolean isEmpty(variant valor) Indica si es una cadena vacia o vale 0 (cero)

boolean validate_required_fields(String container) Verifica si todos los controles dentro del container (selector) tienen un valor establecido