

Anpassungen des Django-Admins

Autor: Arne Brodowski

Webseite: <http://www.arnebrodowski.de>

Datum: 12.03.2014

Dieses Dokument ist ein Handout zum Vortrag auf der Django User Group Hamburg am 12.03.2014.

Global & Index

Admin Titel

Änderung von Header, Seitentitel und Index-Titel.

<https://docs.djangoproject.com/en/dev/ref/contrib/admin/#adminsite-objects>

Django 1.7

z.B. urls.py

```
AdminSite.site_header
AdminSite.site_title
AdminSite.index_title
```

Django <= 1.6

admin/base_site.html

```
{% block title %}{{ title }} | {{ site_title }}{% endblock %}

{% block branding %}
<h1 id="site-name"><a href="{% url 'admin:index' %}">{{ site_header }}</a></h1>
{% endblock %}
```

Admin Usertools

Hinzufügen von Links zu den "Usertools".

admin/base.html block:userlinks

```
<a href="#">{% trans 'Custom action' %}</a> /
```

Sprachumschaltung

Sprachumschaltung im Django Admin hinzufügen.

<https://docs.djangoproject.com/en/1.6/topics/i18n/translation/#the-set-language-redirect-view>

<https://docs.djangoproject.com/en/dev/topics/i18n/translation/#how-django-discovers-language-preference>

admin/base.html block:userlinks

```
<form action="{% url 'set_language' %}" method="post" style="display:inline">
{% csrf_token %}
<select name="language" onchange="this.form.submit()">
{% get_language_info_list for LANGUAGES as languages %}
{% for language in languages %}
```

```
<option value="{{ language.code }}"
    {% if language.code == LANGUAGE_CODE %} selected="selected"{% endif %}>
    {{ language.name_local }} ({{ language.code }})
</option>
{% endfor %}
</select>
</form>
```

urls.py

```
url(r'^i18n/', include('django.conf.urls.i18n')),
```

settings.py

```
MIDDLEWARE_CLASSES = (
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.locale.LocaleMiddleware', # <-- eingefuegt
    'django.middleware.common.CommonMiddleware',
    ...
)
```

Admin Quicklinks

admin/index.html block:content

```
<div class="module">
<table style="width:498px">
<caption>
    <a href="#" class="section">{% trans "Shortcuts" %}</a>
</caption>
<tr>
    <th>
        <a href="">Seiten</a>
        <a href="">Kommentare</a>
        <a href="">Bewertungen</a>
    </th>
</tr>
</table>
</div>
```

Recent Actions aller User

admin/index.html block:sidebar

```
- {% get_admin_log 10 as admin_log for_user user %}
+ {% get_admin_log 20 as admin_log %}

+ {{ entry.user }}:
```

Admin Navigation

base_site.html block_nav-global

```
#site-name {display:none;}
```

Globale Views

urls.py

```
url(r'^admin/run_command/', staff_member_required(run_command)),
```

Hinweis: Vor dem `include(admin.site.urls)` Pattern einfügen!

Apps (Change-List & Change-Form)

App-Labels

Django 1.7

Labels in der Django-Admin Übersicht anpassen.

<https://docs.djangoproject.com/en/dev/ref/applications/#configuring-applications>

apps.py

```
from django.apps import AppConfig
from django.utils.translation import ugettext_lazy as _

class ExampleConfig(AppConfig):
    name = 'example'
    verbose_name = _("example management")
```

__init__.py

```
default_app_config = 'example.apps.ExampleConfig'
```

Admin Object Tools

admin/change_list.html oder
admin/app_label/change_list.html oder
admin/app_label/model_name/change_list.html

block:object-tools-items

```
<li><a href="" class="viewsitelink">{% trans "My action" %}</a></li>
```

Klassen: "viewsitelink", "addlink"

Filter als Drop-Down

admin/filter.html

```
<select onchange="window.location=\
    window.location.pathname+this.options[this.selectedIndex].value">
{% for choice in choices %}
    <option{% if choice.selected %} selected="selected"{% endif %}
        value="{{ choice.query_string|iriencode }}">
```

```
        {{ choice.display|truncatechars:20 }}</option>
    {% endfor %}
</select>
```

Filter: position fixed

CSS

```
#changelist-filter {
    position: fixed;
    top: 92px;
    right: 14px;
}
```

Prepoulated fields: Umlaute

admin/js/urlify.js

```
/* better handling of umlauts */
var LATIN_MAP = {
    'À': 'A', 'Á': 'A', 'Â': 'A', 'Ã': 'A', 'Ä': 'Ae', 'Å': 'A', 'Æ': 'AE', 'Ç': 'C',
    'È': 'E', 'É': 'E', 'Ê': 'E', 'Ë': 'E', 'Ì': 'I', 'Í': 'I', 'Î': 'I',
    'Ï': 'I', 'Ð': 'D', 'Ñ': 'N', 'Ò': 'O', 'Ó': 'O', 'Ô': 'O', 'Õ': 'O', 'Ö': 'Oe',
    'Ø': 'O', 'Ù': 'U', 'Ú': 'U', 'Û': 'U', 'Ü': 'Ue', 'Û': 'U',
    'Ý': 'Y', 'ß': 'TH', 'ß': 'ss', 'à': 'a', 'á': 'a', 'â': 'a', 'ã': 'a', 'ä': 'ae',
    'å': 'a', 'æ': 'ae', 'ç': 'c', 'è': 'e', 'é': 'e', 'ê': 'e', 'ë': 'e',
    'ì': 'i', 'í': 'i', 'î': 'i', 'ï': 'i', 'ð': 'd', 'ñ': 'n', 'ò': 'o', 'ó': 'o',
    'ô': 'o', 'õ': 'o', 'ö': 'oe', 'ø': 'o', 'ù': 'u', 'ú': 'u',
    'û': 'u', 'ü': 'ue', 'Û': 'u', 'ý': 'y', 'þ': 'th', 'ÿ': 'y'
}
...

```

Eine Kopie der angepassten Datei einfach in einen STATICFILES_DIRS Ordner legen.

Tabs für Fieldsets

admin.py

```
fieldsets = (
    ('Core', {'fields': ('name', 'status')}),
    ('Meta', {'fields': ('feld1', 'feld2')}),
    ('Debug', {'fields': ('feld3', 'feld4')}),
)
class Media:
    js = ('jquery.migrate.js',
          'ui.core.js',
          'ui.tabs.js')
    css = {'screen': ('ui.tabs.css',)}
```

admin/app_label/model_name/change_form.html block:content

```

tabnavigation = document.createElement('ul')
tabnavigation.id = "tabnavigation"
$('fieldset').each(function(i,el){
    // add <li> elements to tabnavigation
})
$('#content-main form').prepend(tabnavigation)
$('#content-main form > ul#tabnavigation').tabs()

```

Custom Admin über Proxy-Models

models.py

```

class ExampleProxy(Example):
    class Meta:
        proxy = True

```

admin.py

```

class ExampleProxyAdmin(ExampleAdmin):
    # Raum fuer eigene Anpassungen
    pass
admin.site.register(ExampleProxy, ExampleProxyAdmin)

```

Custom Admin mit get_queryset

admin.py

```

def get_queryset(self, request):
    qs = super(ExampleAdmin, self).get_queryset(request)
    qs = qs.filter(status='ok')
    return qs

```

Beispiel: Getrennte Admins für Redakteure und reguläre Benutzer.

Ordering für Method-Columns

admin.py admin_order_field

```

MyMethod.admin_order_field = "name"

```

Spalte zum Filtern

admin.py

```

def Status(self, instance):
    return u'<a href="?status__exact=%s">%s</a>' % (instance.status,
        instance.get_status_display())
Status.allow_tags = True
Status.short_description = u"Status Filter"
Status.admin_order_field = "status"

```

Hinweis: Funktioniert evtl. nicht optimal mit den neuen Preserved-Filters. Tipp: Ein Icon eignet sich auch gut dafür.

Views für Objekte

admin.py

```
def get_urls(self):
    from django.conf.urls import patterns, url

    def wrap(view):
        def wrapper(*args, **kwargs):
            return self.admin_site.admin_view(view)(*args, **kwargs)
        return update_wrapper(wrapper, view)

    info = self.admin_site.name, self.model._meta.app_label, self.model._meta.module_name

    super_urlpatterns = super(ExampleProxyAdmin, self).get_urls()
    urlpatterns = patterns('',
        url(r'^(.+)/example/$',
            wrap(self.example_view),
            name='%sadmin_%s_%s_example' % info),
    )
    urlpatterns += super_urlpatterns

    return urlpatterns

def example_view(self, request, object_id, extra_context=None):
    ...
```

admin/change_form.html oder
admin/app_label/change_form.html oder
admin/app_label/model_name/change_form.html

block:object-tools-items

```
<li><a href="example/" class="viewsitelink">{% trans "Example" %}</a></li>
```

Google Maps Change List

admin.py

```
class LocationAdmin(admin.ModelAdmin):
    class Media:
        js = ( '//maps.googleapis.com/maps/api/js?sensor=false', )
```

admin/app_label/model_name/change_list.html block:content

```
$('#changelist > form table').before(map_div)
$('#map').css('height', '500px')
$('#changelist > form table').hide()

...

{% for el in cl.result_list %}
    var html = '<p><a href="{ el.pk }"/">{{ el }}</a></p>';
```

```
    add_marker(map, {{ el.lat }}, {{ el.lon }}, html, "{{ el.name }}", {{ el.pk }});  
{% endfor %}
```

Ajax-Inline Changelist

admin.py

```
class Media:  
    js = ('ajaxinline.js',)  
    css = {'all': ('ajaxinline.css', 'admin/css/forms.css')}
```

admin/app_label/model_name/change_list.html block:extrahead

```
<script type="text/javascript">  
(function($) {  
    $(document).ready(function($) {  
        $('#result_list a').ajaxEditable({});  
    });  
})(django.jQuery);  
</script>
```