

18 апреля 2011 в 13:07

Django проект PR Hero: что внутри и полученный опыт

[Django](#)*

Я хочу продолжить хорошую традицию:

- [10 полезных модулей после Стиллион](#)
- [мультиселект с автокомплитом](#)
- [Улучшаем админку](#)

и рассказать об удачных решениях, примененных в Django проекте [PR-Hero](#).



Я призываю всех поступать так же и **делиться своим опытом** :) Объясню почему. В нашей команде каждое удачное решение или приложение из одного проекта обязательно использовалось в следующем. Проекты развивались и становились все лучше и лучше.

А что, если делиться опытом не внутри отдела, а **целым сообществом** российских джангистов?

Что внутри

Наш проект не назвать технически сложным. По сути, это интернет-магазин одной услуги и, внешне, самая сложная часть — оплата через Robokassa.

Внутри все немного интереснее: как в любом *хорошем* интернет-магазине, мы автоматизируем бизнес-процессы. Например, следим, чтобы каждый этап занимал не больше положенного ему времени, иначе руководитель проекта будет получать гневные робо-письма.

Небольшой пример — возможные примеры состояний заявки:

```
STATUSES_WORKFLOW = {
    'new': ['owned', 'cancelled'],
    'owned': ['bullshit', 'cancelled', 'wait_payment'],
    'bullshit': [],
    'wait_payment': ['unpaid', 'cancelled', 'write_release'],
    'unpaid': [],
    'write_release': ['wait_approval', 'cancelled'],
    'wait_approval': ['sending_release', 'cancelled'],
    'sending_release': ['feedback', 'cancelled'],
    'feedback': ['finished', 'cancelled'],
    'finished': [],
    'cancelled': []
}
```

К сожалению, мы еще только пробуем бизнес-модель и не все до конца автоматизировано. Поэтому сегодня я остановлюсь на внешней части. Немного требований к проекту:

- возможность быстро изменять тексты на сайте несколькими нетехническим людям;
- максимальное удобство для клиента;
- многоязычность.

Ну, и как обещал, приступим к положительному опыту.

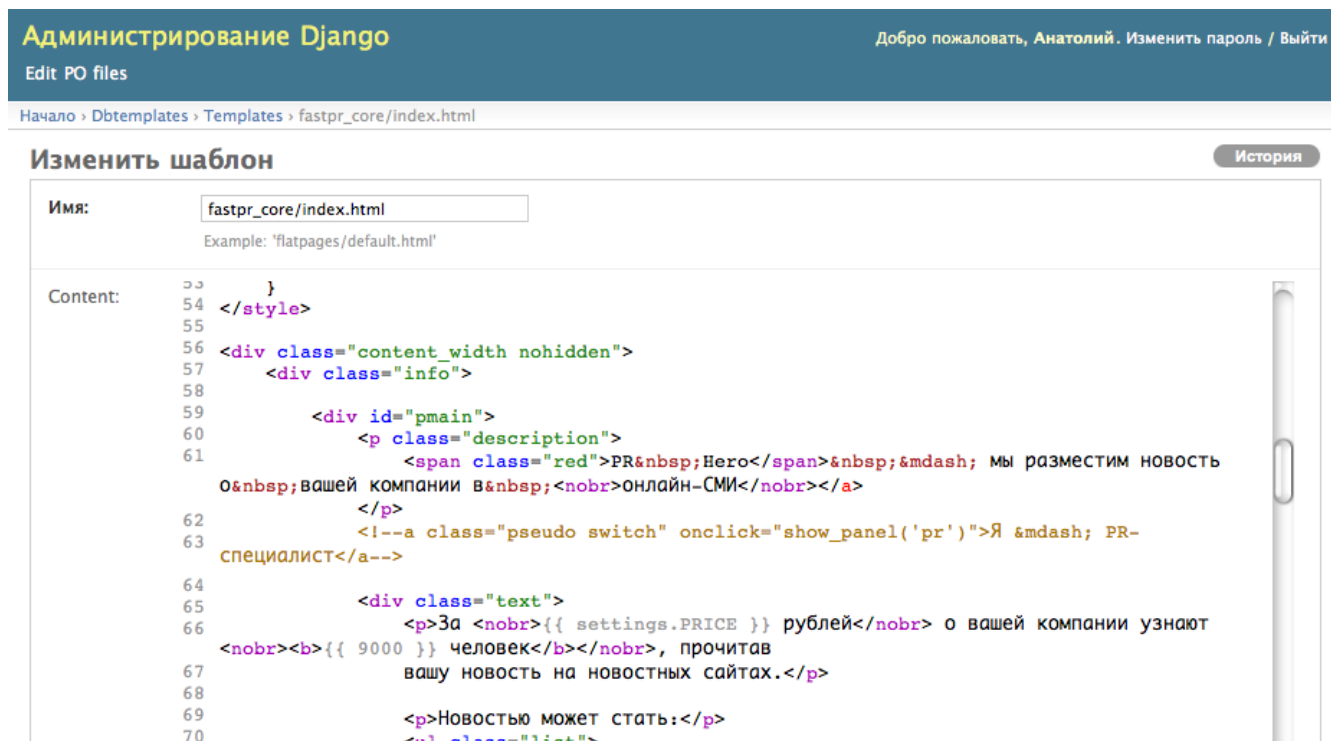
dbtemplates

Считаю это [приложение](#) жемчужиной среди находок этого проекта :)

DBTemplates предоставляет простейший функционал — **хранение и редактирование Django шаблонов в базе данных**.

При этом он поддерживает замечательный HTML редактор [CodeMirror](#) и хранит всю историю изменений при помощи [django-reversion](#).

Выглядит это примерно так:



Это простое решение полностью удовлетворило наши потребности в изменении текстов на сайте. На другом проекте мы попробовали DjangoCMS, но это просто ад. Скажете, что это не подойдет для пользователей? Может быть, но девочки PR-щики после краткого экскурса успешно правят и создают страницы сайта.

Не обошлось без небольших доработок:

- команда `sync_templates` была поправлена для работы с `unicode` и для автоматического создания файлов при синхронизации с БД;
- и был написан простенький скрипт, который делает автоматический коммит шаблонов раз в час.

И как ни странно, эта система отлично работает для нашего проекта. Скорее всего, мы выкинем DjangoCMS из e-Legion.ru в пользу dbtemplates.

Пока мы не доделали одну мелочь — редактирование меню и создание новых страниц, но думаю, [django-treemenus](#) вполне справится с этой задачей. После внедрения на e-legion расскажу подробнее.

autologin-token

Это приложение я создал в рамках проекта, оно маленькое и недописанное :) Но очень полезное.

Идея простая: разрешить пользователю входить на проект автоматически из электрописем.

Решение: создаем модельку и middleware и мы можем в любую ссылку добавить параметр `autotoken=ER2332FFF`, и пользователь аутентифицируется на сайте автоматически.

http://pr-hero.ru/manage_request/15/?autologin_token=ER2332FFF

Использовать приложение просто — генерируете токен для пользователя и вставляете в письмо:

```
data['autologin token'] = AutologinToken.objects.generate(target user)
```

Исходный код приложения (сильно не ругайте, не было времени причесать)

django-robokassa

Название приложения говорит о его назначении. Здесь мне сказать практически нечего, кроме **Большое спасибо** [kmiike](#). Я и подумать не мог, что будет все так просто, за полчаса я подключил сервис к [Robokassa](#).

Платежи с Яндекс Денег, WebMoney, Терминалов и банковских карт без проблем для нас и клиентов падают на расчетный счет.

Есть идея в будущем создать приложения для экрана оплаты, как у Эльбы. Может кто-нибудь подскажет готовое?

Заплатить за ▾

☒ Банковским переводом (платеж поступит в Эльбу в течение 2 — 5 банковских дней)

☐ Картой Visa

☐ Картой MasterCard

☐ WebMoney

☐ Яндекс.Деньги

мгновенные платежи

☐ Кодом активации

➔ Далее

Нажимая на кнопку «Далее» вы соглашаетесь с условиями [договора](#) на право использования Эльбы

django-rosetta

Это [приложение](#) оказывает неоценимую помощь в переводе проектов на иностранные языки. Мы уже успешно опробовали в двух проектах: Goozu и PR-Hero.

Перевод нам часто приходится делегировать, например, Goozu был переведен на украинский, хорватский и даже финский. Приходилось пересылать ро файлы по почте, давать ссылку на roedit, потом долго мержить и отсылать непереуведенные части обратно, попутно давая комментарии что значит та или иная фраза.

Rosetta предоставляет **удобный web-интерфейс для редактирования файлов переводов**. После того, как мы дописали простенький скрипт, который компилирует ро файлы и перезапускает веб-сервер.

Перевод на Английский

Отображать: ☒ Только без перевода ☐ Только с переводом ☐ Fuzzy only ☒ Все

Original	Английский	Fuzzy	Встречается в
Russian	<input type="text"/> suggest	<input type="checkbox"/>	settings_base.py:47
English	<input type="text"/> suggest	<input type="checkbox"/>	settings_base.py:48
Заказать услугу	Order now suggest	<input type="checkbox"/>	settings_base.py:167
Список СМИ	Media list suggest	<input type="checkbox"/>	settings_base.py:168
Пресс релизы	Press releases suggest	<input type="checkbox"/>	settings_base.py:169
О проекте	About suggest	<input type="checkbox"/>	settings_base.py:170
Контакты	Contacts suggest	<input type="checkbox"/>	settings_base.py:171
You are not allowed to attach executable files.	<input type="text"/> suggest	<input type="checkbox"/>	apps/core/fields.py:60
City code must be between 3 and 6 characters in length.	<input type="text"/>	<input type="checkbox"/>	apps/core/fields.py:67

Это решение работает идеально! Носители языков полностью **самостоятельно переводят** портал, сразу наблюдают результат и правят ошибки.

django-localeurl

И последнее на сегодня приложение [django-localeurl](#). Оно решает простую задачу — язык сайта определяется исключительно из ссылки. Например, [pr-hero.ru/en/medias](#) или [en.pr-hero.ru/medias](#) однозначно указывает на английский сайт.

В предыдущих проектах языки у нас определялись автоматом и хранились в куках, это плохое решение по двум причинам:

- поисковики не смогут добраться до контента в другом языке;
- когда даешь ссылку на какую-то статью, велика вероятность что пользователь с неправильными настройками языка получит 404.

Поэтому мы решили, что **URL однозначно определяет язык**. localeurl прекрасно в этом помогает, он отвечает как за смену языка из ссылки, так и за модификацию ссылок на странице (переопределяет `{% url %}`).

Вот и все

Вместо заключения дам ответ на свой вопрос, если делиться опытом после выполненных проектов внутри сообщества, то все наши Django-проекты и приложения станут намного лучше.

Очень надеюсь вскоре увидеть **ваши проекты** изнутри :)

[django](#), [dbtemplates](#), [localeurl](#), [rosetta](#), [autologin token](#), [pr-hero](#), [freshp](#), [robokassa](#)

+79

119

junk 101,5

Впечатляющие возможности
в стильном тонком корпусе из металла

комментарии (56) отслеживать новые: ☐ в почте ☐ в трежере



[danilissimus](#), 18 апреля 2011 в 13:36 #

-4

Да это же поне с говнокода ^_^



[junk](#), 18 апреля 2011 в 13:40 #



+2

Это полу-официальный символ Django.

[webcache.googleusercontent.com/search?hl=ru&q=cache%3Ahttp%3A%2F%2Fdjangoadvent.com%2F1.2%2Fdjango-pony-retrospective%2F&aq=f&aqi=&aql=&oq=](#)



[Crashus](#), 19 апреля 2011 в 15:22 #



0

срёт красным



[olebedev](#), 18 апреля 2011 в 13:42 #

+1

Все отлично сказал, но django-CMS в обиду не дадим!

Вообще то для веб приложений(тем более с i18n) она не подходит. Да.



[junk](#), 18 апреля 2011 в 13:43 #



+1

Приходите к нам копирайтером работать, посмотрю что вы скажете через месяц Django-CMS :)



[marazmiki](#), 18 апреля 2011 в 14:05 #



0

А что не так с django-cms? И, самое главное, почему это должно заботить копирайтеров?



[junk](#), 18 апреля 2011 в 14:15 #



0

нам абсолютно не подошел в работе. Крайне неудобен и сильно ограничивает гибкость, например, `iframe/javascript` не добавив на страницу.

Личное мнение, мы попробовали dbtemplates и Django CMS. Первый победил *для нас* по удобству.



[vaxXxa](#), 18 апреля 2011 в 14:21 #



+1

Что мешает добавить свой виджет в Django CMS, который хоть `iframe`, хоть `google maps` выводит?

Лично мне Django CMS не понравился по скорости и огромному количеству запросов к базе.



[junk](#), 18 апреля 2011 в 14:23 # ↑

0

Да это решение, но мне больше понравилось редактировать шаблоны на прямую. Супер-гибко, быстро и понятно.



[olebedev](#), 18 апреля 2011 в 14:31 # ↑

+2

Обычно это бывает, когда забываешь настроить локаль.



[olebedev](#), 18 апреля 2011 в 14:29 # ↑

0

> Личное мнение, мы попробовали dbtemplates и Django CMS. Первый победил для нас по удобству.

Вот теперь вполне корректно :)

dbtemplates для профессионалов(копирайтеров и т.д.),

Django CMS для заказчиков.



[marazmiki](#), 18 апреля 2011 в 14:46 # ↑

0

Ну что вы! Для истинных профессионалов только notepad.exe и ручное (читай — качественное) редактирование страниц!



[marazmiki](#), 18 апреля 2011 в 14:44 # ↑

+2

Я думаю, вы его неправильно готовили. В первую очередь это моё мнение обусловлено тем, что Вы рядом упоминаете dbtemplates и cms. Мало того, что эти приложения совершенно разного уровня, так они ещё и ничего общего по функциональному назначению не имеют. Конечно, если использовать cms как хранилище шаблонов, ничего путного не выйдет.

Не буду разоряться на тему dbtemplates, оно простое, как две копейки и особого упоминания не заслуживает, по-моему, хотя у меня оно используется довольно часто. Лучше поразоряюсь на тему cms.

Соглашусь с тем, что редактирование страницы (не дерева, а именно страницы) в cms неочевидно и по первости может вызывать ступор. Она (страница редактирования) больше подходит для разработчиков, даже не просто подходит, она для них и предназначена. Пожалуй, вы сами виноваты: вы начали использование cms прежде чем попытались вникнуть в его философию.

Просто скажу, что:

- При правильном подходе ситуация, когда контент редактируется в админке страницы, не обязательна. В неё можно вообще не заходить. Если, конечно, об этом позаботится
- А копирайтерам для редактирования контента, можно в админку даже не заходить, а править всё прямо на сайте.

Про гибкость — я даже и не знаю, что на такое отвечать. Конечно, django-cms не идеален, но если оно не гибкое, то что же тогда гибкое?

Про ифреймы и js не понял. Почему не вставить? Кто запретит?



[junk](#), 18 апреля 2011 в 16:07 # ↑

+1

Я не хочу развивать флейм и доказывать свою точку зрения.

Мы в течении длительного количества времени (1 год/3 месяца) пользовались обоими решениями. *Лично для нас* dbtemplates оказались удобнее.

Если совсем абстрагироваться, решают они одну и ту же задачу — быстрое создание админки для контента.

iframe и js режется в Text-plugin DjangoCMS



[marazmiki](#), 18 апреля 2011 в 16:48 # ↑

0

Вы что, целых 15 месяцев пользовались django-cms в качестве редактора простеньких страничек? Если это так, то я вашу неприязнь понимаю и целиком разделяю: есть и более достойные решения редактирования контента.

С другой стороны, редактировать с помощью django-cms контент и при этом не пользоваться идущими из коробки мультязычностью (привет, localurl!), поддержку меню и хлебных крошек (привет, костыльный treemenus!) и системой плагинов (привет любителям редактировать контент в свойствах страницы) — это дикость, я считаю.

Извините.



[junk](#), 18 апреля 2011 в 16:58 # ↑

0

12 месяцев и до сих пор используем, нет ресурсов на переписывание.

Еще раз повторюсь, что **лично мне** DjangoCMS не подошел. Вам подходит — используйте :) UMI.CMS тоже, говорят, хорошая штука.



[marazmiki](#), 18 апреля 2011 в 17:09 # ↑

+1

Да понятно, что буду использовать :)

За державу обидно просто: прочтёт кто-нибудь статью, отложится у него в голове, что django-cms ад, ну и наговнокодит велосипедов. Или, что тоже возможно, начнёт эту же точку зрения в массы продвигать по старому доброму принципу: «я Пастернака не читал, но осуждаю».

И да, если не секрет, какую версию cmsки используете, не 2.0.x, случайно?



[kmike](#), 18 апреля 2011 в 17:22 #



+2

Использование небольших, простых, независимых приложений — это не велосипедизм и не дикость, а очень разумный подход к архитектуре. Это ж замечательно, когда приложение настолько простое, что о нем даже упоминать не стоит.



[marazmiki](#), 18 апреля 2011 в 18:05 #



+2

Вы мой кумир и спорить сочту богохульством :)

Ни в коем случае не против независимых приложений. Велосипедизм тоже люблю, если в пределах разумного. Но я про другое немного. В посте описано три задачи, на которые хотелось бы обратить внимание: редактирование контента страницы, редактирование структуры сайта и меню, выбор языка.

DBTemplates предоставляет простейший функционал — хранение и редактирование Django шаблонов в базе данных... Это простое решение полностью удовлетворило наши потребности в изменении текстов на сайте

С первой задачей вроде справились: контент вбивается в модель dbtemplate (*непонятно, правда, откуда django знает, какой шаблон рендерить. Имя шаблона строится по соглашению? Прошу пояснений* [junk](#)).

Пока мы не доделали одну мелочь — редактирование меню и создание новых страниц, но думаю, django-treemenus вполне справится с этой задачей

Вторую реализовывать пока не стали, обозвав мелочью. Я с этим определением не согласен: чисто субъективно, задача навигации и всякого рода меню (тех же бредкрамбсов) — одна из самых мутных и труднореализуемых в джанге. Красивое решение видел только одно, о котором ниже. *Если есть что-то достойное против костылей а-ля treemenu, буду очень рад узнать.*

Да и вообще, чего получается-то: девочкам, которые будут вбивать контент, придётся руками рисовать меню? HTML-тегами или тегами django, но руками? Если да, то опять же, чисто субъективно, решение не представляет никакого интереса.

И последнее на сегодня приложение django-localeurl. Оно решает простую задачу — язык сайта определяется исключительно из ссылки.

Тоже с успехом сделали.

А теперь посмотрим на django-cms. Она изначально задумана для решения такого рода задач. Там есть и встроенное редактирование контента страницы, и управление структурой отличное. И localeurl из коробки идёт. То есть один комплекс решает все мелкие задачи, при этом избавляя от рутины и непоняток (см. примечание про редактирование меню девочками).

Неужели более логично заюзать кучу мелких несвязанных приложений, чем несколько приложений покрупнее, но тесно интегрированных друг с другом?



[nimnull](#), 18 апреля 2011 в 18:17 #



0

миш, мне изменяет память или ты по-первости сам плевался на django-cms, когда мы только начинали с ней работать? .)



[marazmiki](#), 18 апреля 2011 в 18:21 #



+2

Не изменяет. Что поделаешь, молод был, глуп.



[junk](#), 18 апреля 2011 в 23:05 #



0

пока не прикрутили treemenus все страницы забиваются через genericviews

Как прикрутим, будет ссылка из treemenus на редактирование шаблона.



[tipugin](#), 18 апреля 2011 в 13:45 #

0

спасибо, не знал о rosetta и dbtemplates!



[Amper](#), 18 апреля 2011 в 13:48 #

+1

dbtemplates — отличная идея, спасибо!



[helm2004](#), 18 апреля 2011 в 13:54 #

-1

Вообще то самый большой минус Django — интернализация (надеюсь правильно написал). Не видел нормальных модулей для сабжа, что б был доступ для модели только для переводчика, копирайтера. Подскажите, хабровчане, пожалуйста.



[bobry](#), 18 апреля 2011 в 14:04 #



+3

Если я правильно понял, то для ваших целей должно хватить `django.contrib.auth`, который вполне себе умеет model-level permissions. Сделайте группу «translator» и дайте ей нужные права, больше вот [TVT](#).



[Never](#), 18 апреля 2011 в 14:15 #



+2

Особенности интернализации Django — это не минус, а специализация. Это не CMS, а фреймворк, который хорошо подходит под нестандартные архитектурные решения и высокую нагрузку.

А если говорить про права, то у него очень гибкая система их настройки. Почитайте по ссылке выше от bobry



[verwolfdotss](#), 18 апреля 2011 в 14:25 #



+1

интернационализация



[Prophet](#), 18 апреля 2011 в 14:55 #

0

Молодцы!

Вообще, имеет ли смысл пытаться собрать каталог удачных модулей для джанго с рейтингом и комментариями, как думаете? Может уже есть что-то подобное.



[laQie](#), 18 апреля 2011 в 15:16 #



+1

Есть же [djangopackages](#)



[mktums](#), 18 апреля 2011 в 15:28 #



0

[djangopackages.com](#)



[mktums](#), 18 апреля 2011 в 15:29 #



+5

Я всегда буду обновлять страницу перед отправкой комментария.



[StopDesign](#), 19 апреля 2011 в 08:59 #



-1

А я не буду. У меня интернет медленный.



[junk](#), 18 апреля 2011 в 16:04 #



+1

Мне, кажется, формат постов на Хабре более подходящий для обмена опытом.

Он не ограничен конкретно приложениями, можно делиться другими best-practice, ну и самое важное личные впечатления.



[Prophet](#), 18 апреля 2011 в 16:06 #



0

Можно тогда какой-нить тег для постов тогда придумать, чтобы их легче было искать среди джанго-постов ;)



[mktums](#), 19 апреля 2011 в 16:21 #



0

py-recipe =)



[vostryakov](#), 18 апреля 2011 в 15:04 #

0

Отличная статья! Не знал некоторых приложений. Буду смотреть.



[barbuza](#), 18 апреля 2011 в 18:05 #

+2

Онотолый, тебя не тошнит от запросов, которые делает reversion? нравится видеть по 3 — 4 запроса на темплейты там, где можно в базу не вообще не лазить? может еще и сессии в базе держите?



[junk](#), 18 апреля 2011 в 23:08 #



0

Витя, я если честно даже не смотрел что-там reversion делает, ни разу им не пользовался. Мне понравилось что он есть из-коробки, авось пригодится.

Спасибо, за совет, вырублю его. Как писал выше, темплейты раз в час сбрасываются на диск.

Сессии держим в базе, да. На ресурсах с <500 уников в день, я об этом даже не задумываюсь. Для ресурсов которые предполагают трафик, нужен совершенно другой подход и это не тот случай.



[el777](#), 21 апреля 2011 в 02:24 #

0

А зачем хранить шаблоны в базе?

Вот даже по теоретическому смыслу: база — это данные (модель), шаблоны — это представление (вьюха).

Чисто практически шаблоны ближе к программному коду, поэтому их удобнее держать в системе контроля версий. Вы их так часто меняете?

Поправили на тестовой версии, проецировали, перенесли в бой.

Убила идея инхронизации шаблонов на диск из базы раз в час. Наверное, навеяно идеями Continuous Integration? Это по-моему за гранью добра и зла. А если у вас в этот момент у вас кто-то правит шаблон и допустил ошибку? Сайт сломается на час?

Извините, но вы из прекрасной Django делаете какую-то мерзкую Zori :)



[junk](#), 21 апреля 2011 в 13:55 #



0

Тексты у нас на проекте может менять любой сотрудник. Причем с моментальной обратной связью (шаблоны читаются с БД). За два месяца мы переписали все тексты два раза. Причем, кто-то пишет основу, другие правят орфографию и пунктуацию.

Ваше предложение с тестом и эсвэзном НАМ абсолютно не подходит. Дергать девелоперов каждый раз, чтобы поправить запятую это бред.

И повторю еще раз. Не зависимо от технической красоты решения. Это удобно для НАС при эксплуатации. Мы решаем бизнес-задачи.



[el777](#), 21 апреля 2011 в 19:41 #



0

Мы с вами под словами «шаблоны» и «контент» понимаем одно и тоже?



[junk](#), 24 апреля 2011 в 11:15 #



0

не знаю :)

Для меня шаблоны нижнего уровня (которые наследуют но не наследуются) = контент.

И очень удобно его править именно в шаблонах, я могу использовать все template filters. И как, мне кажется, создатели Django со мной солидарны, в шаблонах минимум логики и вся она про представление информации.

Какие фильтры и теги мы используем сейчас:

- url
- rurl вставляет знак рубля
- include
- форматирование чисел через i18n {{ 10000 }}
- различные переменные из settings и других ContextProcessors



[MVR](#), 21 апреля 2011 в 09:38 #

0

Тоже используем Rosetta в своем проекте. Отличная штука!



[itforge](#), 22 апреля 2011 в 10:39 #

+1

> autologin-token

Гляньте это приложение bitbucket.org/lorien/django-urlauth/src/94c3fa013950/urlauth/tests.py



[junk](#), 24 апреля 2011 в 11:04 #



0

Класс, спасибо! Замену свое самопальное приложение.



[rhangelxs](#), 28 апреля 2011 в 00:19 #

0

Как у вас работает словарь «состояние заявки», где статусы дублируются? Можете пояснить как вы с этим работаете?



[junk](#), 28 апреля 2011 в 10:05 #



+1

неправильно описал этот словарь. Это граф возможных переходов состояний заявки.

По нему кнопки в админке рисуются и проверяется возможность перехода.

 **rhangelxs**, 28 апреля 2011 в 14:17 # ↑ 0

Если это граф, то он получается не направленный?

 **junk**, 28 апреля 2011 в 14:52 # ↑ 0

почему? Вполне себе направленный.

```
'new': ['owned', 'cancelled'],
```

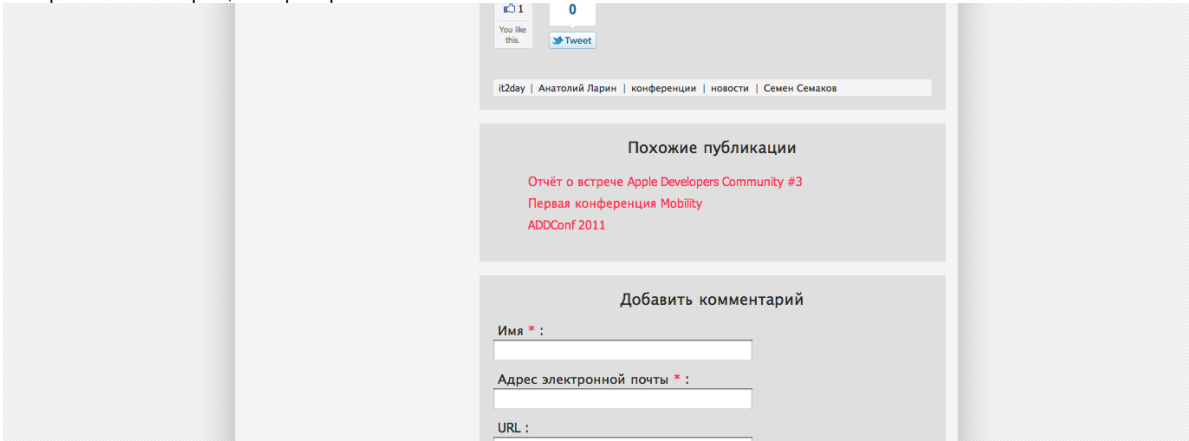
Это значит что из узла «new» можно перейти в узлы «owned», «cancelled».

 **rhangelxs**, 28 апреля 2011 в 14:54 # ↑ 0

А «cancelled» например из «feedback» и из вашего примера это одна и та же точка или вы еще храните историю переходов?

 **junk**, 28 апреля 2011 в 15:08 # ↑ 0

Мы храним всю историю, вот пример.



 **junk**, 28 апреля 2011 в 15:08 # ↑ +1

Гхм, не так картинка

wait_payment		Дата: 29.03.2011	Дата: <input type="text"/>	<input type="checkbox"/>
<div>ждем оплаты</div>		Сегодня <input type="text"/>	Сегодня <input type="text"/>	
<input type="text"/>		Время: 18:49:40	Время: <input type="text"/>	
		Сейчас <input type="text"/>	Сейчас <input type="text"/>	
write_release		Дата: 30.03.2011	Дата: <input type="text"/>	<input type="checkbox"/>
<div>пишем пресс-релиз</div>		Сегодня <input type="text"/>	Сегодня <input type="text"/>	
<input type="text" value="4999.000000"/>		Время: 11:07:04	Время: <input type="text"/>	
		Сейчас <input type="text"/>	Сейчас <input type="text"/>	
wait_approval		Дата: 30.03.2011	Дата: <input type="text"/>	<input type="checkbox"/>
<div>ждем одобрения пресс-релиза</div>		Сегодня <input type="text"/>	Сегодня <input type="text"/>	
<input type="text"/>		Время: 20:10:35	Время: <input type="text"/>	
		Сейчас <input type="text"/>	Сейчас <input type="text"/>	
sending_release		Дата: 27.04.2011	Дата: <input type="text"/>	<input type="checkbox"/>
<div>рассылка пресс-релиза</div>		Сегодня <input type="text"/>	Сегодня <input type="text"/>	
<input type="text"/>		Время: 16:57:37	Время: <input type="text"/>	
		Сейчас <input type="text"/>	Сейчас <input type="text"/>	
feedback		Дата: 27.04.2011	Дата: <input type="text"/>	<input type="checkbox"/>
<div>ждем оценки клиентов</div>		Сегодня <input type="text"/>	Сегодня <input type="text"/>	
<input type="text"/>		Время: 16:57:39	Время: <input type="text"/>	
		Сейчас <input type="text"/>	Сейчас <input type="text"/>	

 **rhangelxs**, 28 апреля 2011 в 22:20 # ↑ 0

И допустим после оценки если происходит «отмена» все в обратном порядке разворачивается?



[junk](#), 28 апреля 2011 в 23:13 # ↑

0

Нет, смотрим на словарь

```
'cancelled': []
```

из состояния отмена деться никуда нельзя.



[rhangelxs](#), 28 апреля 2011 в 23:15 # ↑

0

Спасибо за ваши пояснения, интересный подход!

[Яндекс 23 минуты был
заблокирован Ростелекомом](#)

[Роскомнадзор планирует
блокировать ресурсы с
информацией о
заблокированных ресурсах](#)

[Опубликованы технические
спецификации проекта
Hyperloop](#)

[Все мозги в одном месте](#)

[Заказы для фрилансеров](#)

[Вакансии для айтишников](#)

[Уютная и дружелюбная](#)