ДЗ#3: авторизация в Django

Основы веб-разработки (первый семестр)

Здесь про:
* авторизацию в django в целом
* регистрацию пользователей
* логин, логаут

Принципы работы авторизации в Django

Всё довольно просто - за вас всё делает приложение django.contrib.auth, подключенное к проекту по умолчанию.
То есть - в любом запросе в request.user лежит объект текущего пользователя. Если пользователь не зарегистрирован на сайте - он все равно будет лежать в request.user, просто это будет анонимный пользователь.
Также переменную {{ user }} можно использовать во всех шаблонах.

login, logout

Дефолтные контроллеры для входа на сайт и выхода с него опять же уже реализованы - это django.contrib.auth.views.login и django.contrib.auth.views.logout. Использовать их можно например так:

```
from django.contrib.auth.views import login, logout

urlpatterns = [
    url(r'^login/', login, {'template_name': 'core/login.html'}, name="login"),
    url(r'^logout/', logout, {'template_name': 'core/logout.html'}, name="logout"),
]
```

Обратите внимание - есть способ передачи аргументов внутрь контроллера не только из урла, но и программно - через словарь, указанный после собственно контроллера в функции url. Здесь мы передаем имя шаблона, с помощью которого хотим отрисовать результат работы функций login и logout.
В случае с логином в шаблон попадет форма логина в переменной form, в случае с логаутом - ничего особо полезного в шаблон не попадет, можно отрисовать просто страничку вроде "Вы успешно разлогинились, бла-бла-бла, спасибо-пожалуйста".
Здесь https://docs.djangoproject.com/en/1.9/topics/auth/default/#all-authentication-views можно почитать про поведение данных функций, и про параметры, которые они поддерживают. А также посмотреть прочие удобняшки (смена пароля, сброс

Прямой эфир

Марина Даньшина 18 минут назад
Расписание занятий → Расписание пересдач 0

Екатерина Рогушкова 8 часов назад
Получение значков достижений на портале (ачивки) 2

Ольга Августан Вчера в 18:41
Опросы → Получение значков достижений на портале (ачивки) 2

Ольга Августан Вчера в 13:19
Общие вопросы → Завершение семестра: последние итоговые встречи 0

Ольга Августан Вчера в 13:12
Завершение семестра: итоговые встречи по дисциплинам 10

Марина Даньшина 25 Мая 2016, 14:01
Разработка на Java (первый семестр) → Забыли зонтик 0

Ksenia Sternina 25 Мая 2016, 13:38
Домашнее задание - Юзабилити-тестирование 4

Летяго Владимир 25 Мая 2016, 01:40
Разработка приложений на Android - 1 (второй семестр) → Контрольный рубеж 0

Летяго Владимир 24 Мая 2016, 22:45
Домашнее задание №3 10

Вячеслав Ишутин 24 Мая 2016, 22:31
Разработка на C++ (открытый курс) → Итоговая встреча по дисциплине 0

Вячеслав Ишутин 24 Мая 2016, 22:22
Разработка на C++ (открытый курс) → Итоговое занятие. Презентация курсовых проектов. 0

Александр Агафонов 24 Мая 2016, 19:17
Мастер-класс Ильи Стыценко онлайн «Использование OAuth 2 в приложениях на Django» 1

Марина Даньшина 23 Мая 2016, 20:59
Мероприятия → Мастер-класс Ильи Стыценко онлайн «Использование OAuth 2 в приложениях на Django» 1

Ольга Августан 23 Мая 2016, 17:56
Разработка приложений на iOS - 1 (второй

пароля итп).

Некоторые атрибуты и методы дефолтного пользователя

https://docs.djangoproject.com/en/1.9/ref/contrib/auth/

username, email, first_name, last_name

Ну, понятно. имя пользователя, емейл, имя-фамилия и тому подобное. Можно посмотреть к примеру в админке

check_password, set_password

Пароли пользователей в открытом виде не хранятся. Поэтому нет способа выяснить текущий пароль пользователя, поле password у юзера зашифровано. Можно только проверить, подходит ли какой-либо пароль, либо установить новый. Соответственно, методы check_password и set_password. Единственный принимаемый методом аргумент - собственно пароль.

is_anonymous, is_authenticated

Два метода-антагониста :)
Первый вернет True если пользователь не зарегистрирован, второй - наоборот.

Необходимые настройки в settings.py

LOGIN_REDIRECT_URL

По умолчанию после успешного логина пользователя перенаправляет на ссылку, указанную в GET-параметре next запроса. Если же там ничего нет - используется LOGIN_REDIRECT_URL

LOGIN_URL

Существуют штатные средства отправить пользователя на логин, если он не залогинен, на определенных урлах (например, при попытке создать пост). Эти штатные средства (расскажу о них ниже) используют эту настройку.

Пример:

LOGIN_REDIRECT_URL = "core:base"
LOGIN_URL = "core:login"

Здесь предполагается, что приложение core подключено в наш корневой urls.py с namespace="core", и в нем есть два урла - для морды сайта (с name="base") и для логина (с name="login").

Разрешаем определенные разделы сайта только для залогиненых пользователей

семестр) → Завершение семестра 0

Ольга Августан 23 Мая 2016, 17:16
Итоговая встреча по дисциплине 2

Ольга Августан 23 Мая 2016, 17:07
Основы веб-разработки (первый семестр) → Итоговая встреча по дисциплине 0

Ольга Августан 23 Мая 2016, 17:06
Основы мобильной разработки (первый семестр) → Итоговая встреча по дисциплине 2

Ольга Августан 23 Мая 2016, 17:05
Проектирование интерфейсов мобильных приложений (второй семестр) → Итоговая встреча по дисциплине 0

Ольга Августан 23 Мая 2016, 17:02
Проектирование СУБД (второй семестр) → Итоговая встреча по дисциплине 0

Ольга Августан 23 Мая 2016, 16:59
Общие вопросы → Завершение семестра: итоговые встречи по дисциплинам 10

Весь эфир

Связь с разработчиками

Блоги

Основы веб-разработки (первый семестр) 26,04
Разработка на C++ (открытый курс) 16,97
Основы мобильной разработки (первый семестр) 14,70
Разработка на Java (первый семестр) 13,58
Общие вопросы 5,71
Разработка приложений на Android - 1 (второй семестр) 5,66
Стажировка 4,52
Проектирование интерфейсов мобильных приложений (второй семестр) 3,42
Разработка приложений на iOS - 1 (второй семестр) 2,29
Проектирование СУБД (второй семестр) 2,26

Есть две штатные возможности сделать это - либо через декоратор django.contrib.auth.decorators, либо через Mixin django.contrib.auth.mixins. LoginRequiredMixin

Пример для декоратора:

```
from django.contrib.auth.decorators import login_required

urlpatterns = [
    url(r'^create/$', login_required(views.QuestionCreate.as_view()), name='question_cre
]
```

Пример для Mixin-а:

```
from django.contrib.auth.mixins import LoginRequiredMixin

class QuestionCreate(LoginRequiredMixin, CreateView):
    ...
```

В обоих случаях мы разрешили создание нового вопроса только залогиненым пользователям. Остальные же будут перенаправлены на страницу логина (Django возьмет ее из settings.LOGIN_URL). Причем - после успешного логина пользователь попадет снова на страницу создания вопроса.

Если вы не знаете, что такое декоратор или mixin в python - гуглите :)
С другой стороны, для успешной работы рецепта эти знания не так уж обязательны.

Получение модели пользователя

Пользователь - такая же модель, как и прочие модели Django. Но есть нюанс! (с) Дело в том, что авторизацию в Django можно кастомизировать, в этом случае модель пользователя может лежать уже не в django.contrib.auth.models.User (она там по умолчанию), а в совершенно другом месте проекта. И настраивается это именно для всего проекта в целом. Поэтому текущую модель пользователя нельзя получать прямым импортом, а только так:

Привязка модели к пользователю

Делается через специальную переменную settings.AUTH_USER_MODEL (по умолчанию там лежит django.contrib.auth.models.User - дефолтная модель пользователя).

```
from django.conf import settings
class Post(models.Model):
    author = models.ForeignKey(settings.AUTH_USER_MODEL)
```

Получение модели пользователя в коде

django.contrib.auth.get_user_model(), и только так.

```
from django.contrib.auth import get_user_model

class CreateUser(CreateView):
    model=get_user_model()
    ...
```