**Lecture 7.1. Аутентификация в Django: полный пример входа, выхода и смены пароля**

В этом руководстве вы узнаете, как легко добавить полноценную систему аутентификации приложение Django с функциями входа в систему, выхода из системы, смены пароля и сброса.

Мы будем использовать django-crispy-forms и Bootstrap 4 [ <https://django.fun/tutorials/django-i-formy-bootstrap-4/>] для стилизации пользовательского интерфейса приложения.

**Что нужно знать**

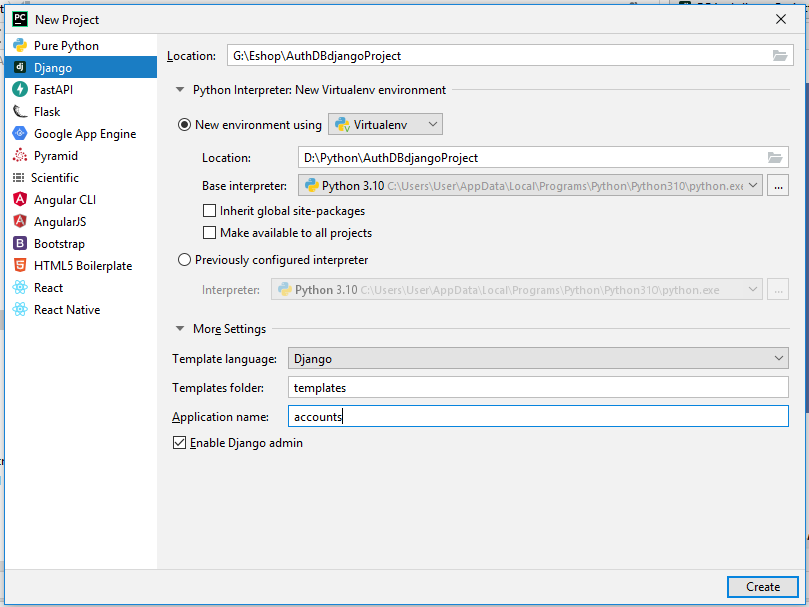
Давайте начнем с определением знаний для использования этого урока. Чтобы выполнить пошаговое руководство, вам необходимо выполнить несколько требований, таких как:

* базовые знания Python,
* знания Django (django-admin.py and manage.py),
* должна быть установлена последняя версия Python 3 (сейчас это 3.7),
* база данных SQLite установлена в вашей системе.

Мы будем использовать pip и venv, которые связаны в виде модулей в последних версиях Python, поэтому вам не нужно устанавливать их, если вы не работаете со старыми версиями.

Если вы готовы, давайте начнем!

**Создание проекта Django**



Здесь мы будем использовать Crispy-forms. Crispy-forms — это замечательное приложение, которое дает контроль над отображением форм **Django** не ломаю их обычную работу. Эта статья оринетирована на использование **Bootstrap** 4-ой версии, но также может быть применена и к старым версиям **Bootstrap**.

Основная причина использования **Crispy-forms** в своих проектах — это простое отображение форм **Django** без написания лишнего кода и с минимальными установками, что очень упрощает работу.

**Установка**

Устанавливаем с использованием pip:

|  |
| --- |
| pip install django-crispy-forms |

Добавляем приложение в INSTALLED\_APPS и выбираем, какой стиль будем использовать:

**settings.py**

|  |
| --- |
| INSTALLED\_APPS = [  ...  'crispy\_forms',  ]  CRISPY\_TEMPLATE\_PACK = **'bootstrap4'** |

В итоге получим:

|  |
| --- |
| INSTALLED\_APPS = [  **'django.contrib.admin'**,  **'django.contrib.auth'**,  **'django.contrib.contenttypes'**,  **'django.contrib.sessions'**,  **'django.contrib.messages'**,  **'django.contrib.staticfiles'**,  **'accounts.apps.AccountsConfig'**,  **'crispy\_forms'**, ] CRISPY\_TEMPLATE\_PACK = **'bootstrap4'** |

Теперь вы можете создать свою базу данных и запустить сервер разработки Django, используя следующие команды:

|  |
| --- |
| python manage.py migrate  python manage.py runserver |

Вы можете использовать свой браузер, чтобы перейти по адресу localhost:8000 и увидеть, как работает ваше веб-приложение.

**Встроенное приложение auth**

Приложение auth - это встроенная система аутентификации в Django, которая позволяет разработчикам добавлять аутентификацию в свои приложения, не изобретая велосипед, пытаясь реализовать базовую функциональность с нуля.

Приложение аутентификации Django предоставляет следующие функциональные возможности из «коробки»:

* Класс авторизации [LoginView](https://django.fun/cbv/Django/latest/django.contrib.auth.views/LoginView/),
* Класс выхода [LogoutView](https://django.fun/cbv/Django/latest/django.contrib.auth.views/LogoutView/),
* Сброс пароля [PasswordResetView](https://django.fun/cbv/Django/latest/django.contrib.auth.views/PasswordResetView/),
* Смена пароля [PasswordChangeView](https://django.fun/cbv/Django/latest/django.contrib.auth.views/PasswordChangeView/),

Вам нужно только предоставить шаблоны для реализации этих функций в вашем приложении.

Для регистрации пользователей необходимо создать свой вид и шаблон.

Вам нужно проверить, приложение django.contrib.auth указано в INSTALLED\_APPS файла settings.py, который используется по умолчанию.

Затем в accounts создайте файл urls.py в приложении учетных записей и добавьте следующий код:

|  |
| --- |
| **from** django.contrib.auth **import** views **from** django.urls **import** path  urlpatterns = [  ] |

**Авторизация пользователей с помощью LoginView**

Вы можете войти в систему в приложении Django, используя представление на основе классов [LoginView](https://django.fun/cbv/Django/latest/django.contrib.auth.views/LoginView/). В ваш accounts/urls.py добавьте следующий путь:

|  |
| --- |
| **from** django.contrib.auth **import** views **from** django.urls **import** path  urlpatterns = [  path(**'login/'**, views.LoginView.as\_view(), name=**'login'**),  ] |

Вы просто используете метод as\_view() класса LoginView, чтобы вернуть объект обратного вызова, который может быть назначен в качестве функции представления функции path().

Далее вам нужно предоставить шаблон для вашего входа в систему. Создайте папку с шаблонами в корне приложения ваших учетных записей и добавьте файл base.html со следующим кодом:

|  |
| --- |
| <!doctype **html**> <**html lang="en"**> <**head**>  <**link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"  integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous"**>  <**title**>Django Authentication Example</**title**> </**head**> <**body**> <**div class="container"**>  <**div class="row justify-content-center"**>  <**div class="col-4"**>  <**h1 class="text-center"**>Django Authentication Example</**h1**>  {% **block main** %}  {% **endblock** %}  </**div**>  </**div**> </**div**> </**body**> </**html**> |

Переходим к файлу urls.py приложений и изменим его содержание:

|  |
| --- |
| **from** django.urls **import** path, include urlpatterns = [  path(**'accounts/'**, include(**'accounts.urls'**)) ] |

Сначала мы импортируем Bootstrap 4 в наш базовый HTML-шаблон. Затем мы создаем контейнер <div> заголовок и основной блок main, где Django отображает другие части наших шаблонов.

Затем создайте папку templates/registration и шаблон login.html со следующим кодом:

|  |
| --- |
| {% **extends 'base.html'** %} {% **load crispy\_forms\_tags** %}  {% **block main** %}  <**div class="card"**>  <**div class="card-body"**>  <**h4 class="card-title"**>Log in to your account</**h4**>   <**form method="post"**>  {% **csrf\_token** %}  <**input type="hidden" name="next" value="**{{ **next** }}**"**>   {{ **form**|**crispy** }}   <**button type="submit" class="btn btn-primary btn-block"**>Log in</**button**>  </**form**>  </**div**>  </**div**> {% **endblock** %} |

Мы расширяем предыдущий базовый шаблон, загружаем crispy\_forms\_tags и затем перезаписываем основной блок main, чтобы добавить нашу форму входа.

Затем мы создаем HTML-форму с помощью метода POST и визуализируем поля формы с помощью фильтра crispy, который применяет стили Bootstrap к отдельным полям.

csrf\_token добавляет поле для защиты CSRF в форму входа в систему.

Мы также добавляем скрытое поле формы, содержащее следующий URL-адрес, который будет использоваться Django для перенаправления пользователя на следующую страницу после успешного входа в систему. По умолчанию он перенаправляет на URL-адрес accounts/profile.

**Настройка URL для редиректа после авторизации**

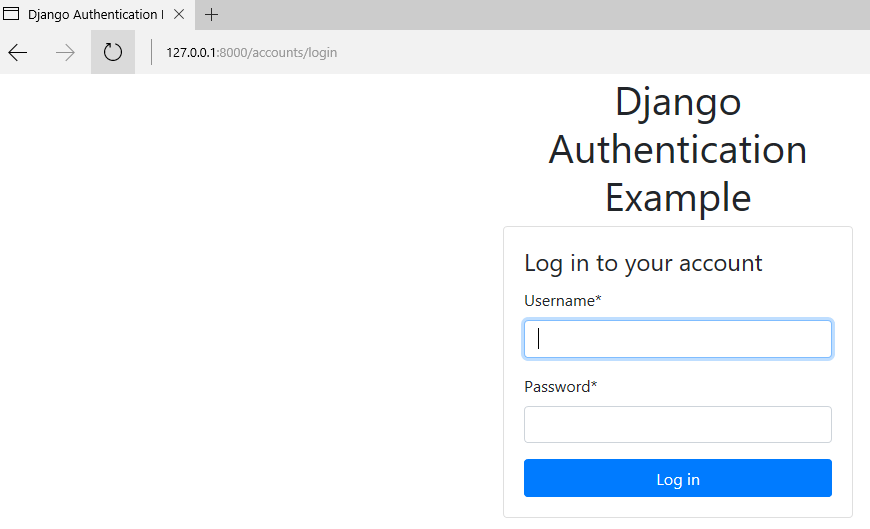
Вы можете установить URL перенаправления с помощью параметра LOGIN\_REDIRECT\_URL. Откройте файл settings.py и в конца его добавьте:

|  |
| --- |
| LOGIN\_REDIRECT\_URL = **'/'** |

Для проверки представления входа в систему вы можете создать пользователя с помощью команды manage.py createsuperuser из своего терминала.

**Примечание.** После входа в систему вы будете перенаправлены на URL-адрес /accounts/profile.

Это скриншот формы входа в систему с Bootstrap 4:



Переходим к папку template и создаём html файл с названием Index.html

|  |
| --- |
| <!DOCTYPE **html**> <**html lang="en"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>Title</**title**> </**head**> <**body**> {% **block main** %}  <**ul class="sidebar-nav"**>  {% **if user**.**is\_authenticated** %}  <**li**>Пользователь: {{ **user**.**get\_username** }}</**li**>  <**li**><**a href="**{% **url 'logout'** %}**?next=**{{ **request**.**path** }}**"**>Logout</**a**></**li**>  {% **else** %}  <**li**><**a href="**{% **url 'login'** %}**?next=**{{ **request**.**path** }}**"**>Login</**a**></**li**>  {% **endif** %}  </**ul**> {% **endblock** %} </**body**> </**html**> |

Переходим к файлу urls.py приложений и изменим его содержание:

|  |
| --- |
| **from** django.urls **import** path, include **from** django.views.generic **import** TemplateView  urlpatterns = [  path(**'accounts/'**, include(**'accounts.urls'**)),  path(**''**, TemplateView.as\_view(template\_name=**"index.html"**)), ] |

**Выход пользователя с помощью LogoutView**

Вы можете выйти из системы в вашем приложении, используя представление на основе классов [LogoutView](https://django.fun/cbv/Django/latest/django.contrib.auth.views/LogoutView/). В своем файле accounts.py добавьте путь logout/ и свяжите его с вызываемым представлением LogoutView:

|  |
| --- |
| path(**'logout/'**, views.LogoutView.as\_view(), name=**'logout'**), |

В результате получим

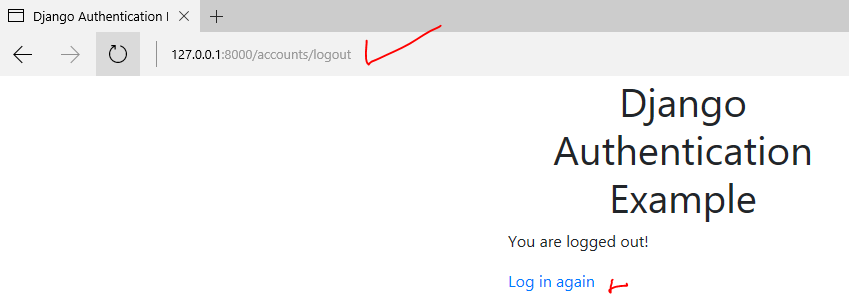
|  |
| --- |
| **from** django.contrib.auth **import** views **from** django.urls **import** path  urlpatterns = [  path(**'login/'**, views.LoginView.as\_view(), name=**'login'**),  path(**'logout/'**, views.LogoutView.as\_view(), name=**'logout'**),  ] |

Мы снова используем метод as\_view() для возврата вызываемого объекта из класса LogoutView.

Далее вам нужно создать файл registration/logged\_out.html со следующим кодом:

|  |
| --- |
| {% **extends 'base.html'** %} {% **block main** %}  <**p**>You are logged out!</**p**>  <**a href="**{% **url 'login'** %}**"**>Log in again</**a**> {% **endblock** %} |

Это снимок экрана с завершенным выходом:



PasswordResetView, PasswordResetDoneView, PasswordResetConfirmView and PasswordResetCompleteView

Вы можете разрешить своим пользователям сбрасывать свои пароли, используя множество представлений:

* [PasswordResetView](https://django.fun/cbv/Django/latest/django.contrib.auth.views/PasswordResetView/),
* [PasswordResetDoneView](https://django.fun/cbv/Django/latest/django.contrib.auth.views/PasswordResetDoneView/),
* [PasswordResetConfirmView](https://django.fun/cbv/Django/latest/django.contrib.auth.views/PasswordResetConfirmView/)
* [PasswordResetCompleteView](https://django.fun/cbv/Django/latest/django.contrib.auth.views/PasswordResetCompleteView/)

В своем файле accounts/urls.py добавьте следующие пути:

|  |
| --- |
| **from** django.contrib.auth **import** views **from** django.urls **import** path  urlpatterns = [  path(**'login/'**, views.LoginView.as\_view(), name=**'login'**),  path(**'logout/'**, views.LogoutView.as\_view(), name=**'logout'**),  path(**'password-change/'**, views.PasswordChangeView.as\_view(), name=**'password\_change'**),  path(**'password-change/done/'**, views.PasswordChangeDoneView.as\_view(), name=**'password\_change\_done'**),  ] |

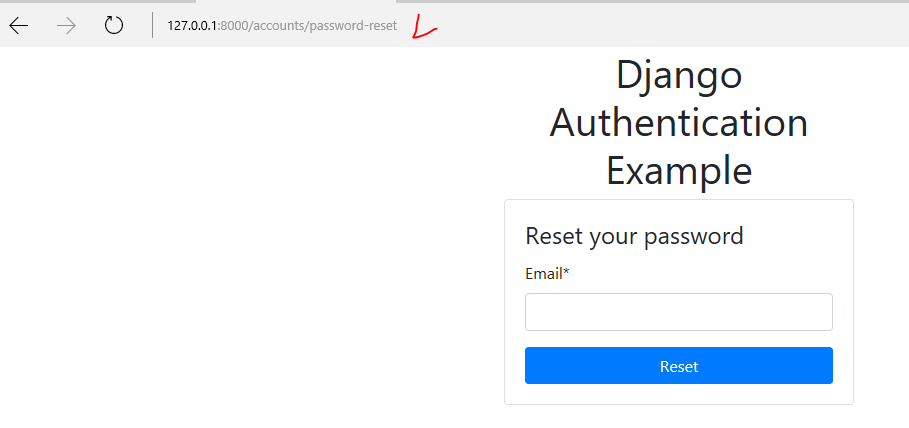
Затем вам нужно добавить шаблон registration/password\_reset\_form.html со следующим кодом:

|  |
| --- |
| {% **extends 'base.html'** %} {% **load crispy\_forms\_tags** %}  {% **block main** %}  <**div class="card"**>  <**div class="card-body"**>  <**h4 class="card-title"**>Reset your password</**h4**>  <**form method="post"**>  {% **csrf\_token** %}  <**input type="hidden" name="next" value="**{{ **next** }}**"**>  {{ **form**|**crispy** }}   <**button type="submit" class="btn btn-primary btn-block"**>Reset</**button**>  </**form**>  </**div**>  </**div**>  </**div**> {% **endblock** %} |

Таким же образом, вам нужно добавить шаблоны password\_reset\_confirm.html, password\_reset\_done.html, password\_reset\_email.html и password\_reset\_complete.html.

Это скриншот формы сброса пароля со стилями Bootstrap 4:

<http://127.0.0.1:8000/accounts/password-reset/>



**Смена пароля: PasswordChangeView и PasswordChangeDoneView**

Вы можете разрешить своим пользователям изменять свои пароли с помощью представлений на основе классов PasswordChangeView и PasswordChangeDoneView.

В своем файле accounts/urls.py добавьте следующие пути:

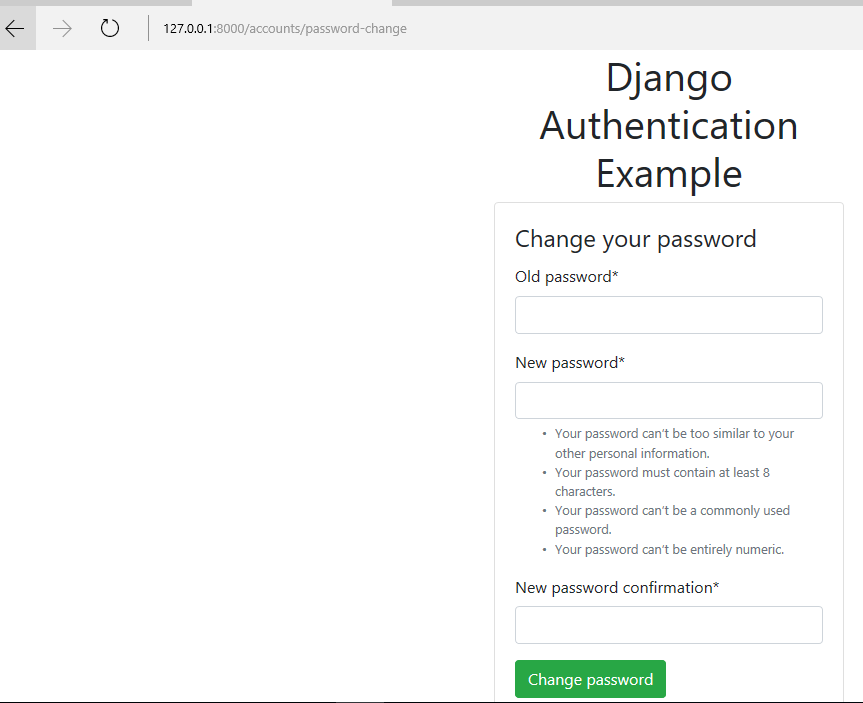
|  |
| --- |
| path(**'password-reset/'**, views.PasswordResetView.as\_view(), name=**'password\_reset'**), path(**'password-reset/done/'**, views.PasswordResetDoneView.as\_view(), name=**'password\_reset\_done'**), |

Затем создайте шаблон registration/password\_change\_form.html и добавьте следующий код:

|  |
| --- |
| {% **extends 'base.html'** %} {% **load crispy\_forms\_tags** %}  {% **block main** %}  <**div class="card"**>  <**div class="card-body"**>  <**h4 class="card-title"**> Change your password</**h4**>   <**form method="post"**>  {% **csrf\_token** %}  {{ **form**|**crispy** }}   <**button type="submit" class="btn btn-success"**>Change password</**button**>  </**form**>  </**div**>  </**div**> {% **endblock** %} |

Вам также необходимо добавить шаблон password\_change\_done.html.

Это скриншот формы смены пароля:



**Регистрация пользователей**

Для регистрации пользователей встроенное приложение аутентификации Django не предоставляет готовую функцию или представление на основе классов, поэтому вам необходимо создать собственную реализацию регистрации.

Импорт URL в основой urls.py вашего проекта

Вы добавили различные URL-адреса для реализации аутентификации в вашем приложении учетных записей, но их нельзя использовать, пока вы не добавите их в файл уровня проекта urls.py.

Во-первых, вот полный файл accounts/urls.py:

|  |
| --- |
| **from** django.contrib.auth **import** views **from** django.urls **import** path  urlpatterns = [  path(**'login/'**, views.LoginView.as\_view(), name=**'login'**),  path(**'logout/'**, views.LogoutView.as\_view(), name=**'logout'**),  path(**'password-change/'**, views.PasswordChangeView.as\_view(), name=**'password\_change'**),  path(**'password-change/done/'**, views.PasswordChangeDoneView.as\_view(), name=**'password\_change\_done'**),  path(**'password-reset/'**, views.PasswordResetView.as\_view(), name=**'password\_reset'**),  path(**'password-reset/done/'**, views.PasswordResetDoneView.as\_view(), name=**'password\_reset\_done'**),  path(**'reset/<uidb64>/<token>/'**, views.PasswordResetConfirmView.as\_view(), name=**'password\_reset\_confirm'**),  path(**'reset/done/'**, views.PasswordResetCompleteView.as\_view(), name=**'password\_reset\_complete'**), ] |

Затем откройте файл urls.py, который находится в базовом каталоге вашего проекта, и используйте функцию include() для импорта URL-адресов учетных записей:

|  |
| --- |
| **from** django.urls **import** path, include **from** django.views.generic **import** TemplateView  urlpatterns = [  path(**'accounts/'**, include(**'accounts.urls'**)),  path(**''**, TemplateView.as\_view(template\_name=**"index.html"**)), ] |

Теперь запустите сервер разработки. Вы можете использовать аутентификационные URL-адреса по адресу /accounts:

<http://127.0.0.1:8000/accounts/login/> для входа,

<http://127.0.0.1:8000/accounts/logout/> для выхода,

<http://127.0.0.1:8000/accounts/password-change/> для смены пароля,

<http://127.0.0.1:8000/accounts/password-reset/> для сброс пароля.