[Django.fun](/)☰✕[Документация](/docs/)

## [Документация Django](/docs/django/ru/4.0/)

Django (Джанго) — свободный фреймворк для веб-приложений на языке Python, использующий шаблон проектирования MVC. Проект поддерживается организацией Django Software Foundation. Сайт на Django строится из одного или нескольких приложений, которые рекомендуется делать отчуждаемыми и подключаемыми. Это одно из существенных архитектурных отличий этого фреймворка от некоторых других (например, Ruby on Rails). Один из основных принципов фреймворка — DRY (англ. Don't repeat yourself). Также, в отличие от других фреймворков, обработчики URL в Django конфигурируются явно при помощи регулярных выражений. Для работы с базой данных Django использует собственный ORM, в котором модель данных описывается классами Python, и по ней генерируется схема базы данных.Выберите версию[dev](/docs/django/ru/dev/)[4.0](/docs/django/ru/4.0/)[3.2](/docs/django/ru/3.2/)[3.1](/docs/django/ru/3.1/)[2.2](/docs/django/ru/2.2/)

## [Документация Django библиотек](/docs/)

### [Рецепты Django ORM](/docs/django-orm-cookbook/ru/2.0/)

Рецепты Django ORM - это книга о работе с моделями Django ORM и Django. Django ORM является одним из ключевых столпов Django. Он предоставляет абстракции …

### [Django Rest Framework](/docs/django-rest-framework/ru/3.12/)

Django Rest Framework (DRF) — это библиотека, которая работает со стандартными моделями Django для создания гибкого и мощного API для проекта.

### [Django CMS](/docs/django-cms/ru/3.9/)

Django CMS - это современная платформа для веб-публикаций, построенная на Django, фреймворке веб-приложений «для перфекционистов с соблюдением сроков». Django CMS предлагает готовую поддержку общих функций, …

### [Channels](/docs/channels/ru/3/)

Channels - это проект, который использует Django и расширяет его возможности за пределы HTTP - для обработки WebSockets, протоколов чата, IoT-протоколов и многого другого. Он …

### [ASGI — спецификация и утилиты](/docs/asgiref/ru/3/)

ASGI (Asynchronous Server Gateway Interface) является духовным наследником WSGI, предназначенным для обеспечения стандартного интерфейса между асинхронными веб-серверами, платформами и приложениями Python. WSGI предоставил стандарт для …

### [Python Social Auth](/docs/social-docs/ru/0.1/)

Python Social Auth - это простой в настройке механизм социальной аутентификации/регистрации с поддержкой нескольких платформ и провайдеров аутентификации. Созданный с использованием базового кода из django-social-auth, …

[Перейти к полному списке библиотек →](/docs/)[Статьи](/articles/)

## [Статьи о Django](/articles/)

### [Новости](/news/)

Новости Django. Будьте всегда в курсе последних изменений и событий.

### [Руководства](/tutorials/)

Учебники по фреймворку Django, его функциям, вариантам использования и общим полезным вещам фреймворка.

### [Советы](/tips/)

Различные маленькие подсказки, подсказки, необычное использование Django - небольшие полезные вещи.

### [Видео](/videos/)

Видеоуроки по фреймворку Django, основам и использованию Python и Django.

### [Python](/python/)

Учебники по Python, его функциям, вариантам использования и общим полезным вещам.[Вопросы и ответы](/qa/)[Классы-представления Django](/ru/cbv/)window.yaContextCb.push(()=>{ Ya.Context.AdvManager.render({ renderTo: 'yandex\_rtb\_R-A-395615-13', blockId: 'R-A-395615-13' }) })

## [Документация по фреймворку Django](/docs/django/ru/4.0/)

**Версии:**[dev](/docs/django/ru/dev/)[4.0](/docs/django/ru/4.0/)[3.2](/docs/django/ru/3.2/)[3.1](/docs/django/ru/3.1/)[2.2](/docs/django/ru/2.2/)

Django (Джанго) — свободный фреймворк для веб-приложений на языке Python, использующий шаблон проектирования MVC. Проект поддерживается организацией Django Software Foundation. Сайт на Django строится из одного или нескольких приложений, которые рекомендуется делать отчуждаемыми и подключаемыми. Это одно из существенных архитектурных отличий этого фреймворка от некоторых других (например, Ruby on Rails). Один из основных принципов фреймворка — DRY (англ. Don't repeat yourself). Также, в отличие от других фреймворков, обработчики URL в Django конфигурируются явно при помощи регулярных выражений. Для работы с базой данных Django использует собственный ORM, в котором модель данных описывается классами Python, и по ней генерируется схема базы данных.

## [Рецепты Django ORM](/docs/django-orm-cookbook/ru/2.0/)

**Версии:**[2.0](/docs/django-orm-cookbook/ru/2.0/)

Рецепты Django ORM - это книга о работе с моделями Django ORM и Django. Django ORM является одним из ключевых столпов Django. Он предоставляет абстракции для работы с базами данных, в основном не зависящие от баз данных. Django ORM сочетает в себе простоту использования с мощными абстракциями. Он поддерживает принцип «Простые вещи просты, а сложные вещи возможны». В этой книге мы будем изучать Django ORM, делая с ним разные вещи. Мы зададим около 50 вопросов о Django ORM и получим более глубокое понимание ORM.

## [Django Rest Framework](/docs/django-rest-framework/ru/3.12/)

**Версии:**[3.12](/docs/django-rest-framework/ru/3.12/)

Django Rest Framework (DRF) — это библиотека, которая работает со стандартными моделями Django для создания гибкого и мощного API для проекта.

## [Django CMS](/docs/django-cms/ru/3.9/)

**Версии:**[3.9](/docs/django-cms/ru/3.9/)

Django CMS - это современная платформа для веб-публикаций, построенная на Django, фреймворке веб-приложений «для перфекционистов с соблюдением сроков». Django CMS предлагает готовую поддержку общих функций, которые вы ожидаете от CMS, но также может быть легко настроена и расширена разработчиками для создания сайта, который точно соответствует их потребностям.

## [Channels — расширение для асинхронной работы с Django](/docs/channels/ru/3/)

**Версии:**[3](/docs/channels/ru/3/)

Channels - это проект, который использует Django и расширяет его возможности за пределы HTTP - для обработки WebSockets, протоколов чата, IoT-протоколов и многого другого. Он построен на спецификации Python под названием ASGI. Документация Channels на русском языке.

## [asgiref — спецификация и утилиты ASGI](/docs/asgiref/ru/3/)

**Версии:**[3](/docs/asgiref/ru/3/)

ASGI (Asynchronous Server Gateway Interface) является духовным наследником WSGI, предназначенным для обеспечения стандартного интерфейса между асинхронными веб-серверами, платформами и приложениями Python. WSGI предоставил стандарт для синхронных приложений Python, тогда как ASGI предоставляет стандарт как для асинхронных, так и для синхронных приложений с реализацией обратной совместимости WSGI и несколькими серверами и средами приложений.

## [Документация по Python Social Auth](/docs/social-docs/ru/0.1/)

**Версии:**[0.1](/docs/social-docs/ru/0.1/)

Python Social Auth - это простой в настройке механизм социальной аутентификации/регистрации с поддержкой нескольких платформ и провайдеров аутентификации. Созданный с использованием базового кода из django-social-auth, он реализует общий интерфейс для определения новых поставщиков аутентификации от третьих сторон и для обеспечения поддержки большего количества платформ и ORM. Python Social Auth - документация на русском языке.

## [Django Crispy Forms - лучшая работа с формами](/docs/crispy-forms/ru/1.12/)

**Версии:**[1.12](/docs/crispy-forms/ru/1.12/)[1.9](/docs/crispy-forms/ru/1.9/)

Лучший способ получить Django DRY формы. Создавайте программные макеты многократного использования из компонентов, имея полный контроль над отображаемым HTML без записи HTML в шаблонах. Все это без нарушения стандартного способа работы в Django, так что оно прекрасно работает с любым другим приложением для форм.

## [Виджет Select2 для Django](/docs/django_select2/ru/7.4.2/)

**Версии:**[7.4.2](/docs/django_select2/ru/7.4.2/)

Select2 предоставляет настраиваемое поле выбора с поддержкой поиска, тегов, удаленных наборов данных, бесконечной прокрутки и многих других часто используемых параметров.

## [Django Simple History](/docs/django-simple-history/ru/3.0/)

**Версии:**[3.0](/docs/django-simple-history/ru/3.0/)[2.12](/docs/django-simple-history/ru/2.12/)[2.11](/docs/django-simple-history/ru/2.11/)[2.10](/docs/django-simple-history/ru/2.10/)

Хранит историю моделей и позволяет просматривать/отменять изменения из админки.

## [Celery](/docs/celery/ru/5.1/)

**Версии:**[5.1](/docs/celery/ru/5.1/)

Очереди задач используются как механизм распределения работы между потоками или машинами. Входом очереди задач является единица работы, называемая задачей. Выделенные рабочие процессы постоянно следят за очередями задач в поисках новой работы.

## [Представления-классы(Class-Based Views)](/ru/cbv/)

Представления-классы Django — это базовые абстрактные классы, реализующие общие задачи веб-разработки на Django. Они достаточно "мощные" в плане использования и полностью используют возможности объектно-оринтированного программирования и множественного наследования Python для расширения своей функциональности. Они больше, чем просто общие базовые упрощения использования Django — они предоставляют утилиты, которые можно смешивать со сложными представлениями в своих задачах.

###### **Auth mixins**

* [AccessMixin](/ru/cbv/Django/3.2/django.contrib.auth.mixins/AccessMixin/)
* [LoginRequiredMixin](/ru/cbv/Django/3.2/django.contrib.auth.mixins/LoginRequiredMixin/)
* [PermissionRequiredMixin](/ru/cbv/Django/3.2/django.contrib.auth.mixins/PermissionRequiredMixin/)
* [UserPassesTestMixin](/ru/cbv/Django/3.2/django.contrib.auth.mixins/UserPassesTestMixin/)

###### **Auth views**

* [LoginView](/ru/cbv/Django/3.2/django.contrib.auth.views/LoginView/)
* [LogoutView](/ru/cbv/Django/3.2/django.contrib.auth.views/LogoutView/)
* [PasswordChangeDoneView](/ru/cbv/Django/3.2/django.contrib.auth.views/PasswordChangeDoneView/)
* [PasswordChangeView](/ru/cbv/Django/3.2/django.contrib.auth.views/PasswordChangeView/)
* [PasswordContextMixin](/ru/cbv/Django/3.2/django.contrib.auth.views/PasswordContextMixin/)
* [PasswordResetCompleteView](/ru/cbv/Django/3.2/django.contrib.auth.views/PasswordResetCompleteView/)
* [PasswordResetConfirmView](/ru/cbv/Django/3.2/django.contrib.auth.views/PasswordResetConfirmView/)
* [PasswordResetDoneView](/ru/cbv/Django/3.2/django.contrib.auth.views/PasswordResetDoneView/)
* [PasswordResetView](/ru/cbv/Django/3.2/django.contrib.auth.views/PasswordResetView/)
* [SuccessURLAllowedHostsMixin](/ru/cbv/Django/3.2/django.contrib.auth.views/SuccessURLAllowedHostsMixin/)

###### **generic**

* [GenericViewError](/ru/cbv/Django/3.2/django.views.generic/GenericViewError/)

###### **Generic base**

* [ContextMixin](/ru/cbv/Django/3.2/django.views.generic.base/ContextMixin/)
* [RedirectView](/ru/cbv/Django/3.2/django.views.generic.base/RedirectView/)
* [TemplateResponseMixin](/ru/cbv/Django/3.2/django.views.generic.base/TemplateResponseMixin/)
* [TemplateView](/ru/cbv/Django/3.2/django.views.generic.base/TemplateView/)
* [View](/ru/cbv/Django/3.2/django.views.generic.base/View/)

###### **Generic dates**

* [ArchiveIndexView](/ru/cbv/Django/3.2/django.views.generic.dates/ArchiveIndexView/)
* [BaseArchiveIndexView](/ru/cbv/Django/3.2/django.views.generic.dates/BaseArchiveIndexView/)
* [BaseDateDetailView](/ru/cbv/Django/3.2/django.views.generic.dates/BaseDateDetailView/)
* [BaseDateListView](/ru/cbv/Django/3.2/django.views.generic.dates/BaseDateListView/)
* [BaseDayArchiveView](/ru/cbv/Django/3.2/django.views.generic.dates/BaseDayArchiveView/)
* [BaseMonthArchiveView](/ru/cbv/Django/3.2/django.views.generic.dates/BaseMonthArchiveView/)
* [BaseTodayArchiveView](/ru/cbv/Django/3.2/django.views.generic.dates/BaseTodayArchiveView/)
* [BaseWeekArchiveView](/ru/cbv/Django/3.2/django.views.generic.dates/BaseWeekArchiveView/)
* [BaseYearArchiveView](/ru/cbv/Django/3.2/django.views.generic.dates/BaseYearArchiveView/)
* [DateDetailView](/ru/cbv/Django/3.2/django.views.generic.dates/DateDetailView/)
* [DateMixin](/ru/cbv/Django/3.2/django.views.generic.dates/DateMixin/)
* [DayArchiveView](/ru/cbv/Django/3.2/django.views.generic.dates/DayArchiveView/)
* [DayMixin](/ru/cbv/Django/3.2/django.views.generic.dates/DayMixin/)
* [MonthArchiveView](/ru/cbv/Django/3.2/django.views.generic.dates/MonthArchiveView/)
* [MonthMixin](/ru/cbv/Django/3.2/django.views.generic.dates/MonthMixin/)
* [TodayArchiveView](/ru/cbv/Django/3.2/django.views.generic.dates/TodayArchiveView/)
* [WeekArchiveView](/ru/cbv/Django/3.2/django.views.generic.dates/WeekArchiveView/)
* [WeekMixin](/ru/cbv/Django/3.2/django.views.generic.dates/WeekMixin/)
* [YearArchiveView](/ru/cbv/Django/3.2/django.views.generic.dates/YearArchiveView/)
* [YearMixin](/ru/cbv/Django/3.2/django.views.generic.dates/YearMixin/)

###### **Generic detail**

* [BaseDetailView](/ru/cbv/Django/3.2/django.views.generic.detail/BaseDetailView/)
* [DetailView](/ru/cbv/Django/3.2/django.views.generic.detail/DetailView/)
* [SingleObjectMixin](/ru/cbv/Django/3.2/django.views.generic.detail/SingleObjectMixin/)
* [SingleObjectTemplateResponseMixin](/ru/cbv/Django/3.2/django.views.generic.detail/SingleObjectTemplateResponseMixin/)

###### **Generic edit**

* [BaseCreateView](/ru/cbv/Django/3.2/django.views.generic.edit/BaseCreateView/)
* [BaseDeleteView](/ru/cbv/Django/3.2/django.views.generic.edit/BaseDeleteView/)
* [BaseFormView](/ru/cbv/Django/3.2/django.views.generic.edit/BaseFormView/)
* [BaseUpdateView](/ru/cbv/Django/3.2/django.views.generic.edit/BaseUpdateView/)
* [CreateView](/ru/cbv/Django/3.2/django.views.generic.edit/CreateView/)
* [DeleteView](/ru/cbv/Django/3.2/django.views.generic.edit/DeleteView/)
* [DeletionMixin](/ru/cbv/Django/3.2/django.views.generic.edit/DeletionMixin/)
* [FormMixin](/ru/cbv/Django/3.2/django.views.generic.edit/FormMixin/)
* [FormView](/ru/cbv/Django/3.2/django.views.generic.edit/FormView/)
* [ModelFormMixin](/ru/cbv/Django/3.2/django.views.generic.edit/ModelFormMixin/)
* [ProcessFormView](/ru/cbv/Django/3.2/django.views.generic.edit/ProcessFormView/)
* [UpdateView](/ru/cbv/Django/3.2/django.views.generic.edit/UpdateView/)

###### **Generic list**

* [BaseListView](/ru/cbv/Django/3.2/django.views.generic.list/BaseListView/)
* [ListView](/ru/cbv/Django/3.2/django.views.generic.list/ListView/)
* [MultipleObjectMixin](/ru/cbv/Django/3.2/django.views.generic.list/MultipleObjectMixin/)
* [MultipleObjectTemplateResponseMixin](/ru/cbv/Django/3.2/django.views.generic.list/MultipleObjectTemplateResponseMixin/)© Django.Fun 2017-2021 | Django.Fun не связан с Django Software Foundation. Django - зарегистрированная торговая марка Django Software Foundation.(function (d, w, c) { (w[c] = w[c] || []).push(function() { try { w.yaCounter46974723 = new Ya.Metrika({ id:46974723, clickmap:true, trackLinks:true, accurateTrackBounce:true, webvisor:true }); } catch(e) { } }); var n = d.getElementsByTagName("script")[0], s = d.createElement("script"), f = function () { n.parentNode.insertBefore(s, n); }; s.type = "text/javascript"; s.async = true; s.src = "https://mc.yandex.ru/metrika/watch.js"; if (w.opera == "[object Opera]") { d.addEventListener("DOMContentLoaded", f, false); } else { f(); } })(document, window, "yandex\_metrika\_callbacks");

