<https://djbook.ru/rel1.9/topics/signals.html>

* [django.db.models.signals.pre\_save](https://djbook.ru/rel1.9/ref/signals.html#django.db.models.signals.pre_save) & [django.db.models.signals.post\_save](https://djbook.ru/rel1.9/ref/signals.html#django.db.models.signals.post_save)

Отправляются до или после вызова метода [save()](https://djbook.ru/rel1.9/ref/models/instances.html" \l "django.db.models.Model.save" \o "django.db.models.Model.save) модели.

* [django.db.models.signals.pre\_delete](https://djbook.ru/rel1.9/ref/signals.html#django.db.models.signals.pre_delete) & [django.db.models.signals.post\_delete](https://djbook.ru/rel1.9/ref/signals.html#django.db.models.signals.post_delete)

Отправляются до или после вызова метода [delete()](https://djbook.ru/rel1.9/ref/models/instances.html" \l "django.db.models.Model.delete" \o "django.db.models.Model.delete) модели или [delete()](https://djbook.ru/rel1.9/ref/models/querysets.html" \l "django.db.models.query.QuerySet.delete" \o "django.db.models.query.QuerySet.delete) класса QuerySet.

* [django.db.models.signals.m2m\_changed](https://djbook.ru/rel1.9/ref/signals.html#django.db.models.signals.m2m_changed)

Отправляются после изменения [ManyToManyField](https://djbook.ru/rel1.9/ref/models/fields.html" \l "django.db.models.ManyToManyField" \o "django.db.models.ManyToManyField) в модели.

* [django.core.signals.request\_started](https://djbook.ru/rel1.9/ref/signals.html#django.core.signals.request_started) & [django.core.signals.request\_finished](https://djbook.ru/rel1.9/ref/signals.html#django.core.signals.request_finished)

Отправляются, когда Django начинает или заканчивает HTTP запрос.

Signal.connect**(***receiver***,***sender=None***,***weak=True***,***dispatch\_uid=None***)**[**¶**](https://djbook.ru/rel1.9/topics/signals.html#django.dispatch.Signal.connect)

**Функции-получатели**

**def** **my\_callback**(sender, **\*\***kwargs):

**print**("Request finished!")

**from** django.core.signals **import** request\_finished

request\_finished**.**connect(my\_callback)

or

**from** django.core.signals **import** request\_finished

**from** django.dispatch **import** receiver

@receiver(request\_finished)

**def** **my\_callback**(sender, **\*\***kwargs):

**print**("Request finished!")

обработчики сигналов лежат в модуле signals приложения, к которому они относятся

sender:

@receiver(pre\_save, sender**=**MyModel)

**def** **my\_handler**(sender, **\*\***kwargs):

**...**

**Предотвращение дублирования сигналов**

request\_finished**.**connect(my\_callback, dispatch\_uid**=**"my\_unique\_identifier")

Обычно, этот идентификатор является строкой, хотя подойдёт любой хешируемый объект

Создание и посылка сигналов

**import** django.dispatch

pizza\_done **=** django**.**dispatch**.**Signal(providing\_args**=**["toppings", "size"])

Signal.send**(***sender***,***\*\*kwargs***)**

Signal.send\_robust**(***sender***,***\*\*kwargs***)**

**def** **send\_pizza**(self, toppings, size):

pizza\_done**.**send(sender**=**self**.**\_\_class\_\_, toppings**=**toppings, size**=**size)

**...**

Signal.disconnect**(**receiver=None**,**sender=None**,**dispatch\_uid=None**)**

<https://djbook.ru/rel3.0/topics/cache.html>

Memcached

Кэширование в базу данных

Кэширование на файловую систему

Кэширование в оперативной памяти

Псевдокэширование (для разработки)

Кэш для всего сайта

Кэширование на уровне представлений

Определение кэша уровня представления в URLconf

Кэширование фрагментов шаблона

API низкого уровня для кэширования