Django 3.0

В 3-й версии Django добавилась поддержка асинхронности.

https://docs.djangoproject.com/en/3.2/releases/3.0/

> “Django 3.0 begins our journey to making Django fully async-capable by providing support for running as an ASGI application.”

3-я версия — это начало поддержки асинхронности. Для полной совместимости еще многое нужно сделать в самой Django и в ее экосистеме.

Как видно из экспериментов, если в асинхронном фреймворке использовать синхронную операцию, производительность драматически падает. Даже если в одном месте использовать быструю синхронную операцию, последствия все равно будут заметны.

Что нужно сделать в Django для поддержки полной асинхронности?

Внутренние компоненты. Работа с ORM, с cache и прочим остается синхронной. Эти компоненты пока не переписали, но сделали костыль sync_to_async. Он нем можно почитать тут: https://docs.djangoproject.com/en/3.2/topics/async/

В следующих версиях Django в них, скорее всего, добавят нативную поддержку async await интерфейса без костылей.

Внутренние библиотеки contrib. Внутри Django есть набор готовых компонент, которые позволяют быстрее писать код. Например, django.contrib.auth. Для поддержки асинхронности нужно переписать все эти компоненты.

Внешние библиотеки. Если предыдущие шаги зависели от разработчиков библиотеки, и это централизованно можно было бы изменить, то огромная база написанных библиотек сторонними разработчиками тоже должна поддержать асинхронность и переписать свои библиотеки под asyn await интерфейс. При этом нужно сохранить обратную совместимость, что является нетривиальной задачей.

Исходя из сказанного, путь полной поддержки в django будет достаточно долгий, но с большой вероятностью в будущем мы придем в асинхронности в Django.