1 **Почему Spring MVC требует Tomcat, а WebFlux использует Netty?**

* **Spring MVC** использует Servlet API, которое работает на серверах приложений, таких как Tomcat. **WebFlux** использует реактивную модель и поддерживает серверы, такие как Netty, которые могут обрабатывать запросы асинхронно и эффективно управлять большим количеством соединений.

2 **Какие задачи лучше решать с помощью Spring WebFlux?**

* Задачи с высоким уровнем асинхронных операций, такие как обработка большого числа параллельных запросов, микросервисы с низкой задержкой и веб-приложения с долгими операциями, где важно не блокировать потоки.

3 **Что происходит, если увеличить количество запросов?**

* При увеличении количества запросов с синхронной моделью обработки (Spring MVC) возрастает нагрузка на сервер и время отклика, так как каждый запрос блокирует поток. В реактивной модели (WebFlux) нагрузка на сервер и задержка меньше, потому что запросы обрабатываются асинхронно.

4 **Почему WebFlux показывает меньшую задержку при высокой нагрузке?**

* WebFlux использует асинхронную обработку запросов и неблокирующие операции, что позволяет эффективно обрабатывать множество запросов без блокировки потоков, в отличие от традиционных моделей, где каждый запрос требует выделения потока.