

Асинхронные взаимодействия

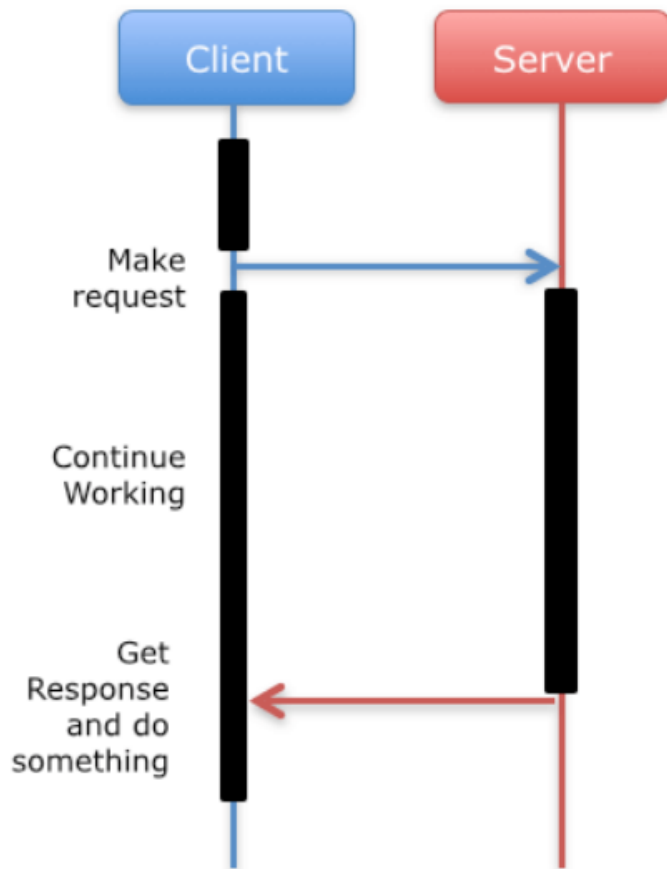
В асинхронном взаимодействии, клиент отправляет запрос и продолжает свою работу, не дожидаясь ответа от сервера. Это похоже на отправку письма по почте: вы отправляете письмо и продолжаете свою повседневную жизнь, не ожидая мгновенного ответа.

Почему их можно использовать:

- **Обработка ресурсоемких задач:** Асинхронные взаимодействия идеально подходят для задач, которые занимают много времени и не требуют мгновенного ответа. Например, если ваше приложение обрабатывает большие объемы данных и занимает значительное время, асинхронный подход позволит клиенту продолжить работу, пока сервер обрабатывает запрос.
- **Время отклика не важно:** Если время отклика не критически важно для приложения, асинхронные взаимодействия могут быть хорошим выбором. Это позволяет уменьшить нагрузку на систему, позволяя клиенту продолжить работу без ожидания ответа.
- **Неважность ответа:** Асинхронные вызовы также полезны, когда ответ от сервера не является критически важным для продолжения работы клиента. Например, если ваше приложение отправляет уведомления по электронной почте пользователям, оно может не ждать подтверждения об успешной отправке каждого уведомления, чтобы продолжить свою работу.
- **Масштабируемость:** Асинхронные системы часто обладают лучшей масштабируемостью, поскольку они способны обрабатывать запросы параллельно и не тратят время на ожидание ответа.
- **Отказоустойчивость:** В асинхронных системах, даже если один из компонентов временно недоступен, остальная часть системы может продолжать функционировать нормально. Это может сделать систему более устойчивой к сбоям.

Клиент может делать что угодно после отправки запроса:

Asynchronous



Важно отметить, что выбор между синхронным и асинхронным подходами зависит от конкретного сценария использования и требований к вашей системе. Оба подхода имеют свои преимущества и недостатки, и важно понимать их, чтобы принимать обоснованные архитектурные решения.