

Пройдёмся еще раз по **безопасности**, потому что это очень важно в контексте асинхронного API.

Самое первое — используйте асинхронные API (может, кроме WS и SSE) для передачи сигнала (триггера), а не информации. Пусть это будут просто «перехватчики» и «доносчики» событий, и не более. Информацию передавайте по защищённым каналам API HTTPS.

Как эффективно управлять этим?

- Используйте безопасные протоколы: защитите каналы связи с помощью стандартных протоколов, таких как HTTPS и WSS (WebSocket Secure), для защиты передачи данных между клиентом и сервером.

Что делать? Внедрите HTTPS для связи на основе HTTP и WSS для связи WebSocket в вашем API для обеспечения безопасной передачи данных.

- Аутентификация и авторизация доступа: внедрите надлежащие механизмы аутентификации и авторизации, чтобы гарантировать, что только авторизованные пользователи могут получить доступ к вашему API и выполнять действия, основанные на их правах доступа.

Что делать? Включите OAuth, JWT или другие широко используемые схемы аутентификации и авторизации в ваш API для проверки и контроля доступа пользователей.

- Проверяйте и валидируйте входные данные: убедитесь, что все входные данные, получаемые API, проверяются для предотвращения вредоносных атак, таких как SQL-инъекции или межсайтовый скриптинг (XSS).

Что делать? Внедрите механизмы проверки и валидации входных данных для всех данных, получаемых вашим API, проверяя типы данных, длину и допустимые символы, а также удаляя потенциально вредоносное содержимое.

- Шифруйте конфиденциальные данные: защитите конфиденциальные данные, такие как персонально идентифицируемая информация (PII), зашифровав их перед хранением или передачей.

Что делать? Используйте алгоритмы шифрования TLS для защиты конфиденциальных данных во время хранения или передачи в вашем API.

- Внедрите ограничение скорости: ограничьте количество запросов, которые могут быть сделаны к вашему API за определенный промежуток времени для защиты от атак типа "отказ в обслуживании" (DDoS) и предотвращения злоупотреблений.

Что делать? Интегрируйте механизмы ограничения скорости в ваш API, установив разумные ограничения на количество запросов от одного пользователя или IP-адреса в течение определенного периода времени.

- Отслеживайте и регистрируйте события безопасности: отслеживайте события и инциденты, связанные с безопасностью, чтобы выявлять потенциальные угрозы и своевременно устранять их.

Что делать? Внедрите системы мониторинга и регистрации, которые отслеживают события безопасности, такие как неудачные попытки аутентификации, необычные шаблоны запросов или обнаруженные атаки, и генерируют предупреждения для немедленного принятия мер.

- Используйте API-ключи и токены: используйте API-ключи или токены для предоставления доступа к вашему API, что позволит вам отслеживать и управлять использованием вашего API различными клиентами.

Что делать? Создайте систему выдачи API-ключей или токенов и включите их в механизмы аутентификации и авторизации вашего API для отслеживания использования клиентами.

Следуя этим лучшим практикам безопасности, вы сможете создать надежный и безопасный асинхронный API, который защитит данные ваших пользователей и вашу инфраструктуру от потенциальных угроз.