

## Важное заключение

В заключение проектирования синхронных API веб-сервисов (SOAP, gRPC, GraphQL, REST и др.) хотелось бы подчеркнуть, что разработка и использование API – это искусство, требующее гибкости, творчества и открытости к новым подходам.

В мире веб-технологий не существует строгих правил, которым следует безоговорочно подчиняться.

Хотя определенные стандарты и рекомендации, такие как RFC, обеспечивают основу и общую структуру, важно помнить, что они служат лишь ориентиром, а не догмой.

Веб-технологии постоянно развиваются и меняются, и мы должны адаптироваться к этим изменениям, а также искать инновационные решения.

- В качестве примера можно привести использование тела в GET-запросах. Хотя это не является общепринятой практикой и может вызвать некоторые проблемы совместимости, в определенных ситуациях это может быть полезным и оправданным решением (например, это используется в онлайн-системах ELK для поиска данных).
- Или использование глаголов в названии ресурса (как вы помните, это нежелательно). Но Twitter повсеместно использует это в своём API:
  - GET `mutes/users/id`
  - GET `mutes/users/list`
  - POST `mutes/users/create`
  - POST `mutes/users/destroy`

Помимо стандартов и рекомендаций, также стоит учитывать практические аспекты и потребности конкретных проектов. Не стесняйтесь экспериментировать и исследовать разные подходы к разработке API. Главное – оценивать эффективность, безопасность и масштабируемость ваших решений.

Ваш успех в разработке API во многом зависит от вашей способности комбинировать знания и опыт, находить компромиссы и быть готовым к переменам. Не бойтесь ошибаться, и помните: главное – это взаимодействие (ведь API - это интерфейс), понимание и удовлетворение потребностей пользователей и клиентов.

В нашем материале мы постарались донести до вас некие лучшие практики проектирования API. И надеемся, что этот курс станет для вас твердой основой и полезным инструментом на пути к созданию и использованию мощных, гибких и успешных API.