

## Как выяснили, есть два подхода - API First и Code First.

### Подход API First:

Подход API First направлен на проектирование API перед его реализацией. Этот процесс включает в себя создание спецификации API с использованием стандартного формата, такого как OpenAPI (ранее известный как Swagger). Эта спецификация служит контрактом между разработчиками и потребителями API, определяя функциональные возможности API, конечные точки и структуры данных.

### Преимущества подхода API First:

- Улучшенное сотрудничество: Спецификация API выступает в качестве центрального документа, позволяющего улучшить коммуникацию между заинтересованными сторонами, включая разработчиков, тестировщиков и владельцев продукта.
- Более простое обслуживание: Дизайн API остается отделенным от реализации, что упрощает обновление дизайна или рефакторинг кода, не затрагивая клиентов API.
- Более быстрая разработка: Используя инструменты проектирования API, разработчики могут генерировать кодовые заглушки, клиентские SDK и документацию на основе спецификации API, что упрощает процесс разработки.
- Последовательность: Определяя API в первую очередь, разработчики руководствуются последовательным набором правил, что обеспечивает согласованность и хорошую структуру результирующего API.

### Недостатки подхода API First:

- Затраты времени на начальном этапе: Процесс разработки может отнимать много времени, поскольку требует тщательного планирования и согласования между заинтересованными сторонами.
- Жесткие границы: Строгое следование спецификации API может препятствовать гибкости на этапе реализации, особенно при возникновении неожиданных технических проблем.

### Подход Code First:

При подходе Code First разработчики начинают с написания кода для реализации функциональных возможностей API. После реализации API разработчики создают документацию API и, возможно, спецификацию API на основе кода, используя такие инструменты, как Swagger или Spring Rest Docs.

### Преимущества подхода Code First:

- Более быстрая начальная разработка: Разработчики могут сразу же приступить к реализации API, без необходимости длительного предварительного проектирования.
- Гибкость: Подход Code First позволяет разработчикам легче адаптироваться к непредвиденным техническим проблемам и изменениям требований.

Недостатки подхода Code First:

- Меньше сотрудничества: Отсутствие первоначального документа по проектированию API может привести к недопониманию и несогласованности между членами команды.
- Проблемы с обслуживанием: Без отдельного документа по проектированию API внесение обновлений или рефакторинг кода могут быть более сложными и длительными.

Когда использовать тот или иной подход?

Используйте подход API First, когда сотрудничество, согласованность и простота сопровождения имеют решающее значение. Этот подход особенно подходит для крупных проектов, публичных API или проектов с участием нескольких команд.

Используйте подход Code First, когда вам нужно быстро создать прототип API, или когда команда разработчиков небольшая, и общение не так важно. Этот подход может быть более подходящим для внутренних API или небольших проектов.

В конечном итоге выбор между этими двумя подходами зависит от контекста проекта, предпочтений команды и конкретных требований API.