

Давайте поймём разницу между Stateful и Stateless API

Stateful API:

- Сервер запоминает состояние клиента при нескольких запросах. Он хранит информацию о клиенте, такую как предпочтения пользователя, прошлые действия или другие данные, относящиеся к взаимодействию клиента с веб-службой.
- Преимущества:
 - Персонализация. Поскольку сервер хранит состояние клиента, он может обеспечить персонализированный пользовательский опыт, основанный на прошлых взаимодействиях клиента.
 - Более простая навигация. Пользователи могут легко перемещаться между различными частями веб-приложения без необходимости каждый раз заново вводить данные или проходить аутентификацию.
- Недостатки:
 - Проблемы масштабируемости. Хранение состояния клиента на сервере потребляет ресурсы и может стать узким местом при увеличении числа клиентов.
 - Сложность. Управление состоянием клиента на сервере усложняет приложение и делает его более сложным в обслуживании и устранении неполадок.
- Пример из реального мира: работа с файлами по FTP протоколу — каждое действие клиента регистрируется сервером в состоянии и сохраняется на сервере.

Stateless API

- В API без состояния сервер не хранит никакой информации о состоянии клиента между запросами. Каждый запрос должен содержать всю необходимую информацию, чтобы сервер мог его обработать.
- Преимущества
 - Масштабируемость. API без статических данных более масштабируемы, так как серверу не нужно хранить и управлять состоянием клиента, освобождая ресурсы.

- Простота. Отсутствие управления состоянием на сервере упрощает приложение и облегчает его обслуживание и устранение неполадок.
- Недостатки:
 - Меньшая персонализация. API без состояния не обеспечивают такого же уровня персонализации, как API с состоянием, поскольку сервер не запоминает прошлые взаимодействия клиента.
 - Повышенная сложность на стороне клиента. Клиент должен управлять своим состоянием и включать всю необходимую информацию в каждый запрос, что может увеличить сложность на стороне клиента.
- Пример из реального мира: API REST, широко используемые в веб-сервисах, как правило, не имеют состояния по своей конструкции. Они полагаются на то, что клиент предоставляет всю необходимую информацию в каждом запросе, позволяя серверу сосредоточиться на обработке запроса без сохранения состояния клиента.