

# POST-PUT

- POST — создание ресурса
- PUT — обновление ресурса

**Метод запроса HTTP POST** — это отправка новых данных на сервер. Метод POST подразумевает, что данные передаются в теле запроса. Он может использоваться для различных действий, включая создание нового ресурса, загрузку файла или отправку веб-формы. Метод POST является неидемпотентным, то есть, при повторном выполнении результаты будут отличаться, данные на сервере будут меняться каждый раз даже при отправке идентичных данных. Однако ответы POST кэшируются до тех пор, пока сервер устанавливает соответствующие заголовки Cache-Control и Expires.

Метод POST используется для запроса, чтобы исходный сервер принял объект, включенный в запрос, в качестве нового подчиненного ресурса, идентифицируемого Request-URI в строке запроса. Действие, выполняемое методом POST, может не привести к ресурсу, который может быть идентифицирован с помощью URI. В этом случае либо 200 (ОК), либо 204 (Нет содержимого) — это соответствующий статус ответа, в зависимости от того, включает ли ответ объект, который описывает результат.

**Метод запроса HTTP PUT** - заменяет текущие представления ресурса данными запроса, подразумевает, что Вы передаёте всё, что нужно в URL, т.к. тела запроса нет. Метод PUT можно использовать для обновления или создания нового ресурса. Метод PUT должен быть идемпотентным. Идемпотентный метод при повторном выполнении дает ожидаемо одинаковые результаты, то есть несколько одинаковых PUT на один endpoint не меняют состояния API. Ответы PUT не кэшируются. Для запросов PUT клиент должен знать точный URI ресурса. Если новый ресурс создан, сервер происхождения ДОЛЖЕН проинформировать пользовательский агент через ответ 201 (Создано). Если существующий ресурс модифицируется, то должны быть отправлены коды ответа 200 (ОК) или 204 (Нет содержимого), чтобы указать успешное завершение запроса.

## Разница между POST и PUT:

Фундаментальное различие между запросами POST и PUT отражается в различном значении URI запроса. URI в запросе POST идентифицирует ресурс, который будет обрабатывать вложенный объект. Этот ресурс может быть процессом, принимающим данные, шлюзом для какого-либо другого протокола или отдельным объектом, который принимает аннотации. Напротив, URI в запросе PUT идентифицирует объект, вложенный в запрос - агент пользователя знает, какой URI предназначен, и сервер не должен пытаться применить запрос к какому-либо другому ресурсу.

## ПРИМЕР:

- При оформлении заказа через запрос POST с одинаковым набором данных создается столько одинаковых запросов, сколько раз был отправлен запрос POST.
- При оформлении заказа через запрос PUT заказ оформляется один раз, даже если этот запрос будет отправлен несколько раз.