Разница между Post и Put

Изображение выглядит как стол

Автоматически созданное описание

Спецификация HTTP 1.1 гласит, что PUT идемпотентен. Это значит, что клиент может выполнить множество PUT запросов по одному URI и это не приведет к созданию записей дубликатов. Операции присвоения — хороший пример идемпотентной операции

String userId = this.request["USER\_ID"];

Даже если эту операцию выполнить дважды или трижды, никакого вреда не будет (кроме лишних тактов процессора). POST же с другой стороны не идемпотентен. Это что-то вроде инкремента. Вам следует использовать POST или PUT с учетом того является ли выполняемое действие идемпотентным или нет. Говоря языком программистов, если клиент знает URL объекта, который нужно создать, используйте PUT. Если клиент знает URL метода/класса создающего нужный объект, используйте POST.

Разница между PUT и POST - это вопрос семантики. Коль скоро для операций используются разные глаголы, то и смысл у них должен быть разным.

Представьте, что ваш сервис оперирует понятиями блокнот (notebook) и запись (post). Один блокнот может содержать множество записей.

Для добавления новой записи в блокнот c идентификатором id вы будете использовать метод POST с URL mydomain/notebooks/id/. Ваш сервис, ориентируясь на метод POST, сам присвоит нужный идентификатор записи, добавит ее в блокнот и вернет вам URL созданной записи (для доступа к записи по GET или для удаления по DELETE). При этом хорошо бы вернуть клиенту URL созданной записи.

Допустим, запись с идентификатором post-id уже создана и доступна по URL mydomain/notebooks/id/posts/post-id. Но клиент (владелец записи) исправил в ней ошибку и хочет перезаписать ее. Для этого он использует метод PUT с URL mydomain/notebooks/id/posts/post-id и передает обновленную запись в теле запроса. Ваш сервис, ориентируясь на метод PUT удаляет старую запись и записывает новую, при этом она доступна по тому же URL.

Конечно, никто не мешает вам всегда использовать метод POST (например HTML 4 позволял использовать только методы GET и POST). Но все же стоит придерживаться рекомендаций в целях единообразной трактовки методов всеми разработчиками.

. Рекомендуется использоваться метод POST для создания подчиненного ресурса (дочернего по отношению к другому ресурсу; пример блокнота и записи как раз очень подходит).