幂等Idempotency

定义

一个操作如果多次任意执行所产生的影响，均与一次执行的影响相同。

例子

顺利的情况

电商网站，当发送付款请求后，一个顺利的场景是不会有任何错误发生的，支付端收到付款请求，处理所有转账，然后返回一个 HTTP 200 消息表示交易完成。

请求超时的情况

- 这个请求在到达支付端前就已 经发生超时，支付从来没有收到 这样的请求。
- 这个请求到达支付端，但是支付交易失 败，这时发生超时，支付端收到 这样的请求，但没有 处理成功。
- 这个请求到达支付端，并且支付交易成功， 这时发生超时，支付端收到 这样的请求，处理成功，但是没有回 执。
- 这个请求到达支付端，并且支付交易成功，并且 发回回执，然而因为网络原因回执丢失，客户端超时，支付端收到 这样的请求，处理成功，发出回执，但是客户没有收到。

去重Deduplication

HTTP方法 - GET/POST/PUT/PATCH/DELETE/HEAD

DELETE 应该删除某个资源或返回错误。

假设系统中没有任何变化，则 GET 应该始终返回相同的数据。

如果有必要，PUT和PATCH 应该继续进行相同的更改。如果你连续两次或三次发出 PUT 或 PATCH，则不应该因为没有更改而使第二或第三次失败，它应该可以正常工作。

实现

- 幂等令牌（Idempotency Key）
- 确保唯一性（Uniqueness Guarantee）

最常用的做法是利用数据库。比如把幂等令牌所在的数据库表的列做唯一性索引。这样，当试图存储两个含有同样令牌的请求时，必定有一个会报错。

现实中的问题

- 幂等令牌什么时候产生，怎样产生
- 令牌有没有被误删的可能
- 各种竞争条件
- 对请求重试的处理
- 多层幂等