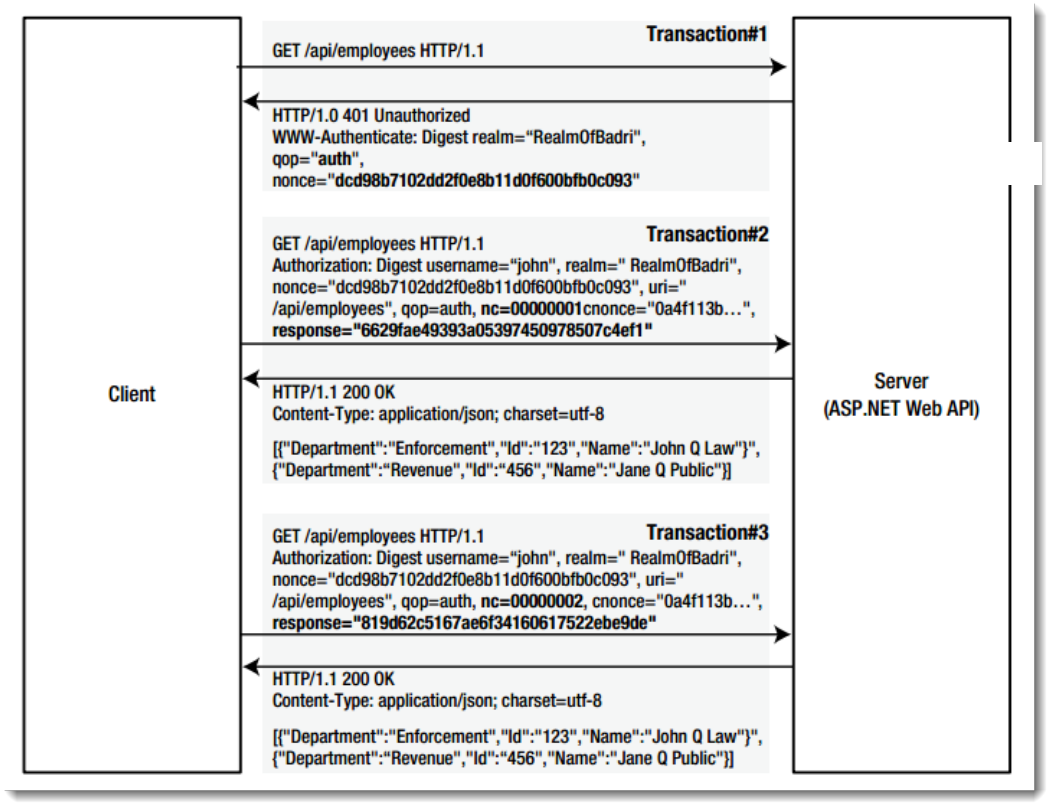
摘要认证原理

在基本认证的方式中，主要的安全问题来自于用户信息的明文传输，而在摘要认证中，主要通过一些手段避免了此问题，大大增加了安全性。



下面大致看一下这部分的验证流程：

1. 客户端请求 /api/employees；
2. 服务端返回401未验证的状态，并且在返回的信息中包含了验证方式Digest，realm的值，QOP(quality of protection)只设置成auth，nonce为一串随机值，在下面的请求中会一直使用到，当过了存活期后服务端将刷新生成一个新的nonce值；
3. 客户端接受到请求返回后，将username:realm:password进行HASH运算，假设运算后的值为HA1。又将请求的路径/api/employees进行HASH运算，假设运算后的值为HA2。再将HA1:nonce:nc:cnonce:qop:HA2进行HASH运算，得到的值放在response中。这里的cnonce为客户端生成的nonce值，而nc用于统计，假设开始时为00000001，下次请求后就变成了00000002，不一定每次都加1，但是后面请求中的nc值肯定大于前一次请求中的nc值。
4. 服务端收到请求后将验证nonce是否过期，如果过期，那么直接返回401，即第二步的状态。如果没有过期，那么比较nc值，如果比前一次nc值小或者前一次根本没有存储的nc值，那么也将直接返回401状态。如果前面的验证都通过，那么服务端也将按照步骤3中计算最终HASH值的步骤计算出HASH值与客户端的进行比较，然后比较客户端提交过来的HASH值与服务端计算出来的HASH进行比较，不匹配返回401，匹配获取请求的数据并返回状态200。

摘要验证主要就是通过上面的HASH比较的步骤避免掉了基本验证中的安全性问题。

摘要验证的优缺点

摘要验证很好地解决了使用基本验证所担心的安全性问题。

但是永远没有绝对的安全，当用户使用字典进行穷举破解时，还是会存在一些被破解的隐患。