# 利用生存时间使 DocumentDB 集合中的数据自动过期

应用程序可以生成和存储大量数据。其中的某些数据（如计算机生成的事件数据、日志和用户会话信息）仅在有限的一段时间内才有用。当数据变得多余，应用程序不再需要时，可以安全地清除这些数据并减少应用程序的存储需求。

通过“生存时间”或 TTL 功能，Microsoft Azure DocumentDB 能够在一段时间后将文档自动从系统清除。可以在集合级别设置默认生存时间，并且可以在每个文档上覆盖该时间。TTL 设置后，无论作为集合默认或在文档级别，DocumentDB 都将自文档上次修改起的某段时间（以秒为单位）后，自动删除该文档。

DocumentDB 中的生存时间针对上次修改该文档的时间使用偏移量。为此，它会使用每个文档中存在的 \_ts 字段。\_ts 字段为 unix 型的时期时间戳，表示日期和时间。每次修改文档时都会更新 \_ts 字段。

## TTL 行为

TTL 功能在两个级别受 TTL 属性控制 - 集合级别和文档级别。设置这些值时以秒为单位，这些值被视为自上次修改文档所在的 \_ts 起的增量。

1. 集合的 DefaultTTL

* 如果缺失（或设置为 NULL），则文档不会自动删除。
* 如果存在且值为“-1”= 无限期 – 默认情况下，文档不过期
* 如果存在其值为某个数字（“n”）– 文档在上次修改“n”秒后过期

1. 文档的 TTL：

* 属性仅在对父集合设置 DefaultTTL 时适用。
* 替代父集合的 DefaultTTL 值。

只要文档已过期（ttl + \_ts >= 当前服务器时间），则文档就会标记为“已过期”。此时间过后，将不允许对这些文档执行任何操作，这些文档也将从执行的任何查询结果中排除。这些文档在系统中被物理删除，并且稍后找机会在后台删除。这不占用集合预算的任何[请求单元 (RU)](documentdb-request-units.md)。

上面的逻辑可显示在以下矩阵中：

|  |  |  |  |
| --- | --- | --- | --- |
|  | 集合上的 DefaultTTL 缺失/未设置 | 集合上的 DefaultTTL =-1 | 集合上的 DefaultTTL =“n” |
| 文档上的 TTL 缺失 | 在文档级别没有要替代的内容，因为文档和集合都没有 TTL 的概念。 | 在此集合中的文档不会过期。 | 在此集合中的文档将在 n 秒间隔后过期。 |
| 文档上的 TTL =-1 | 在文档级别没有要替代的内容，因为集合并未定义文档可以替代的 DefaultTTL 属性。系统未解释文档上的 TTL。 | 在此集合中的文档不会过期。 | 此集合中，TTL =-1 的文档将永不过期。所有其他文档将在“n”秒间隔后过期。 |
| 文档上的 TTL = n | 在文档级别没有要替代的内容。系统未解释文档上的 TTL。 | TTL = n 的文档将在时间间隔 n 秒后过期。其他文档将继承 -1 时间间隔，并且永不过期。 | TTL = n 的文档将在时间间隔 n 秒后过期。其他文档将从集合中继承“n”时间间隔。 |

## 配置 TTL

默认情况下，在所有 DocumentDB 集合和所有文档上禁用生存时间。

## 启用 TTL

若要在集合或集合内的文档上启用 TTL，需要将集合的 DefaultTTL 属性设置为 -1 或非零正数。将 DefaultTTL 设置为 -1 表示默认情况下，集合中的所有文档将永久生存，但 DocumentDB 服务应该监视此集合中已替代此默认值的文档。

## 在集合上配置默认 TTL

你可以在集合级别配置默认生存时间。

若要在集合上设置 TTL，则需要提供一个非零正数，该数字表示自文档上次的修改时间戳 (\_ts) 之后，集合中的所有文档过期所经过的时间段（以秒为单位）。

或者，可以将默认值设置为 -1，这意味着插入到集合中的所有文档在默认情况下将无限期地生存。

## 在文档上设置 TTL

除了在集合上设置默认的 TTL 外，还可以在文档级别设置特定的 TTL。执行此操作将替代该集合的默认设置。

若要在文档上设置 TTL，则需要提供一个非零正数，该数字表示自文档上次的修改时间戳 (\_ts) 之后，文档过期所经过的时间段（以秒为单位）。

若要设置此到期偏移量，请在文档上设置 TTL 字段。

如果文档没有 TTL 字段，将应用集合的默认值。

如果在集合级别禁用了 TTL，在文档上的 TTL 字段将被忽略，直到在集合上再次启用 TTL。

## 在现有文档上扩展 TTL

通过对文档执行任何写入操作，可以在文档上重置 TTL。执行此操作会将 \_ts 设置为当前时间，并将再次开始 ttl 所设置的对文档到期的倒计时。

如果想要更改文档的 ttl，则可以像使用任何其他可设置的字段那样更新字段。

## 从文档中移除 TTL

如果已在文档上设置 TTL，并且不再想要该文档过期，则可以检索文档，移除 TTL 字段并替换服务器上的文档。

当从文档中移除 TTL 字段时，将应用集合的默认值。

若要阻止文档过期并且不从集合继承，则需要将 TTL 值设置为 -1。

## 禁用 TTL

若要在集合上完全禁用 TTL 并阻止后台进程查找过期文档，应删除集合上的 DefaultTTL 属性。

删除此属性不同于将其设置为 -1。设置为 -1 表示添加到集合中的新文档将永久生存，但你可以替代此集合中的特定文档。

完全从集合中移除该属性意味着文档不会过期，即使有的文档已显示替代以前的默认值。

## 常见问题

**TTL 将收取我多少费用？**

在文档上设置 TTL 不会产生额外费用。

**TTL 运行后删除我的文档需要多长时间？**

只要文档已过期（ttl + \_ts >= 当前服务器时间），则文档就会标记为不可用。此时间过后，将不允许对这些文档执行任何操作，这些文档也将从执行的任何查询结果中排除。系统在后台在物理上删除文档。这不会占用集合预算的任何 RU。

**如果删除文档花费了一段时间，那么文档在删除前，会计入我的配额（和帐单）吗？**

不会，文档到期后将不会对这些文档的存储计费，并且文档的大小将不计入集合的存储配额。

**文档上的 TTL 对 RU 费用是否会产生影响？**

不会，不会影响对 DocumentDB 内的任何文档所执行操作的 RU 费用。

**删除文档是否会影响我已对集合预配的吞吐量？**

不会，针对集合的处理请求将优先于运行后台进程删除文档。将 TTL 添加到任一文档对此不会产生影响。

**文档过期时，在删除前它会在集合中保留多长时间？**

只要文档过期，它将不再可访问。文档在实际删除前在集合中保留的确切时间并非确定的，并且基于后台进程何时能够删除该文档。

**过期文档是否跨所有节点被删除或者它是否“最终一致”？**

该文档将在跨所有节点和在所有区域的同时变得不可用。

**TTL 监控的后台任务是否存在 RU 费用？**

否，没有任何相关 RU 费用。

**TTL 过期检查的频率？**

检查 TTL 过期不作为后台进程进行。在响应请求时，后台服务将执行内联检查并排除任何已过期的文档。删除物理文档是在后台异步运行的唯一进程。此进程的频率由集合上可用的 RU 决定。

**TTL 功能是否仅应用于整个文档，或者是否可以使单个文档的属性值过期？**

TTL 应用于整个文档。如果只是想要使文档的一部分过期，则建议将该部分从主文档中提取到一个单独的“链接”文档，然后对被提取的文档使用 TTL。

**TTL 功能是否具有特定的索引编制要求？**

是的。该集合必须将[索引策略设置](documentdb-indexing-policies.md)为“迟缓”或或“一致”。尝试在索引设置为“无”的集合上设置 DefaultTTL 将导致错误，尝试关闭已设置 DefaultTTL 的集合上的索引也是如此。

## 后续步骤

若要了解有关 Azure DocumentDB 的详细信息，请参阅该服务的[文档](https://azure.microsoft.com/documentation/services/documentdb/)页。