

# EF Core 并发冲突与令牌

2019年3月28日 11:44

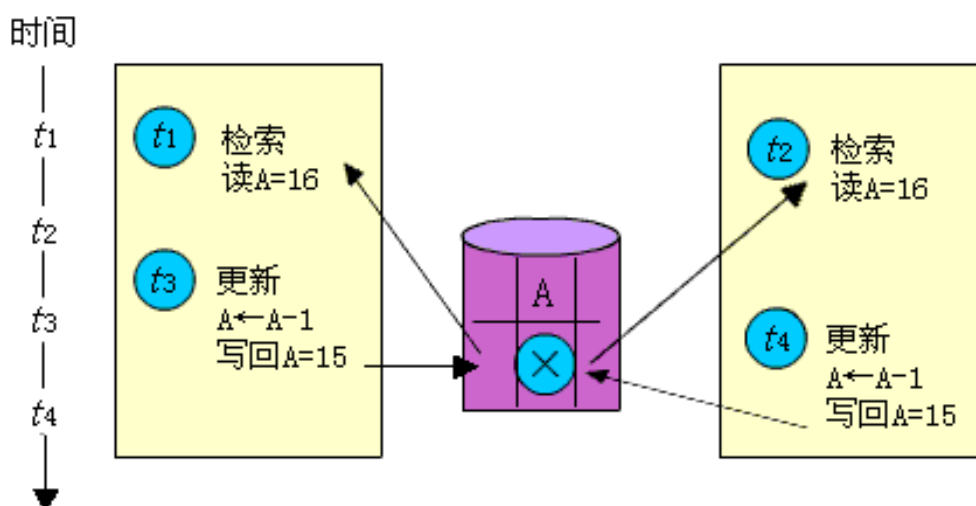
## 并发标记

并发分悲观并发和乐观并发。

悲观并发：比如有两个用户A,B，同时登录系统修改一个文档，如果A先进入修改，则系统会把该文档锁住，B就没办法打开了，只有等A修改完，完全退出的时候B才能进入修改。

乐观并发：同上面的例子，A,B两个用户同时登录，如果A先进入修改紧跟着B也进入了。A修改文档的同时B也在修改。如果在A保存之后B再保存他的修改，此时系统检测到数据库中文档记录与B刚进入时不一致，B保存时会抛出异常，修改失败。

Entity Framework Core 不支持悲观并发，只支持乐观并发。



## 约定

按照约定，属性永远不会配置为并发标记。

### Data Annotations

```
[ConcurrencyCheck]
public string LastName { get; set; }
```

### Fluent API

```
modelBuilder.Entity<Person>().Property(p => p.LastName).IsConcurrencyToken();
```

## 时间戳和行版本

每次插入或更新行时，数据库生成一个时间戳，该属性也被视为并发标记。

## 约定

按照约定，属性永远不会配置时间戳。

### Data Annotations

```
[Timestamp]
public byte[] Timestamp { get; set; }
```

### Fluent API

```
modelBuilder.Entity<Blog>().Property(p => p.Timestamp).IsRowVersion();
```

## 隐藏属性

隐藏属性是指：在实体类中未定义，但在数据库中有字段定义。

```
context.Entry(myBlog).Property("LastUpdated").CurrentValue = DateTime.Now;
```

```
var blogs = context.Blogs.OrderBy(b => EF.Property<DateTime>(b, "LastUpdated"));
```

### Data Annotations

不支持通过数据注解的方式配置隐藏属性。

### Fluent API

```
modelBuilder.Entity<Blog>().Property<DateTime>("LastUpdated");
```