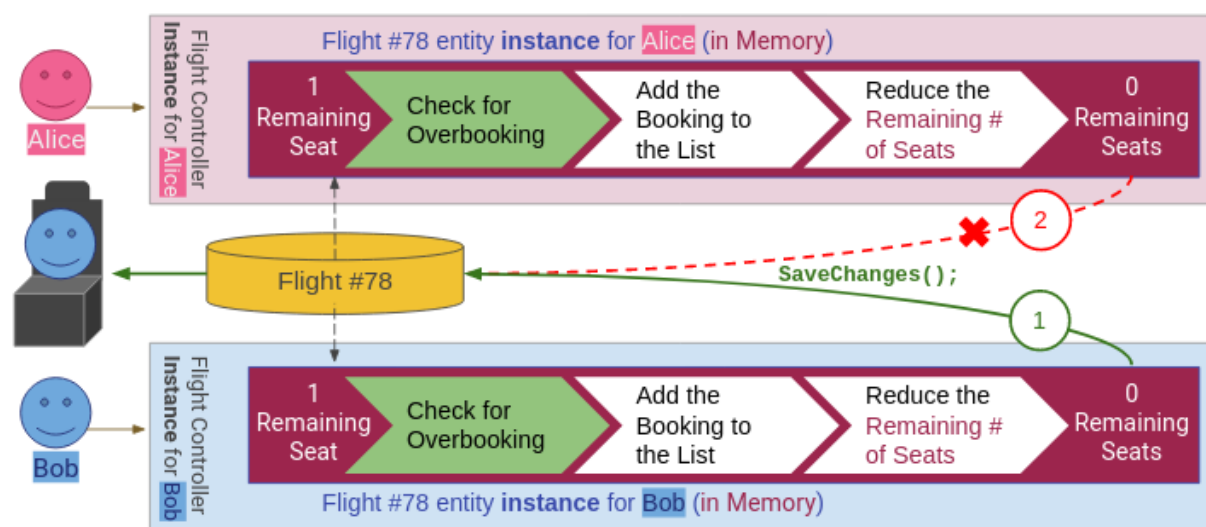


Day 17 - Angular & Asp in 21 days Handouts

Status

Day 17: Handling Concurrency Conflicts using Entity Framework



Race conditions are situations in which multiple users compete to allocate a limited resource. Like when two audiences want to book the only remaining seat of a theater at the same time.

You can define a concurrency token to handle race conditions.

```
public class Entities : DbContext
{
    ...
    protected override void OnModelCreating(ModelBuilder modelBuilder)
    {
        ...
        modelBuilder.Entity<Theater>().Property(e => e.RemainingNumberOfSeats).IsConcurrencyToken();
        ...
    }
}
```

So the transaction that delivers to the database first wins the seat.

While saving the changes, Entity Framework throws `DbUpdateConcurrencyException` if the value of the concurrency token field is changed in the database by another transaction during current transaction.

In general, you can ask user to retry when such an exception occurs.

```
try
{
    entities.SaveChanges();
}
catch (DbUpdateConcurrencyException e)
{
    return Conflict(new { message = "An error occurred while booking. Please try again." });
}
```