Armen Melkumyan • 1st
Technical / Solutions Architect
1yr • 🌐

From .Net C# interview questions

Question:

In Entity Framework, the DbContext can track changes to entities to facilitate updates to the database.

However, in some scenarios, you might want to disable this tracking for performance reasons.

Describe how you would achieve this and explain the implications of doing so.

Answer:

To disable change tracking in Entity Framework, you would use the AsNoTracking() method when querying the database. This method returns a new query where the result set will not be tracked by the context.

Implications of using AsNoTracking():

1) Performance Improvement: When entities are not being tracked, there's less overhead because the DbContext won't keep track of the changes. This can lead to faster query performance, especially when fetching large sets of data that you don't need to update.

2) Stateless Operations: Since the entities are not being tracked, any modifications to these entities will not be persisted back to the database unless you attach them manually and set their state. This can be both an advantage (for read-only operations) and a disadvantage (if updates are expected).

3) Memory Usage: Without tracking, there's potentially reduced memory consumption because the context doesn't hold onto the original values of the entities.

4)Short-lived Context: It pairs well with the best practice of using a short-lived DbContext just for the operation at hand.

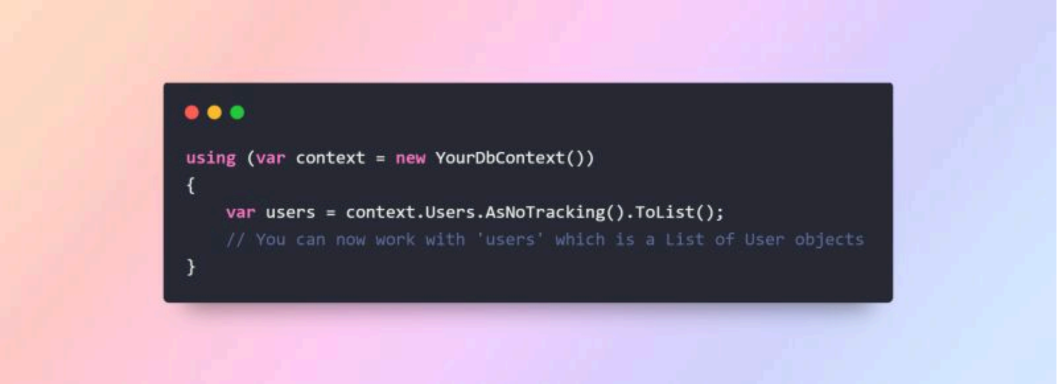
5) Manual State Management: If you need to save changes to entities fetched with AsNoTracking(), you'll need to attach them to a context and manage their state manually.

In summary, using AsNoTracking() can be beneficial for read-heavy operations where the data doesn't need to be updated. However, developers should be aware of the implications and use it judiciously.

#csharp #entityframework #ef #interview #interviewpreparation #interviewquestions #asknotracking

#performance

#performanceoptimization



```
using (var context = new YourDbContext())
{
    var users = context.Users.AsNoTracking().ToList();
    // You can now work with 'users' which is a List of User objects
}
```

7