FullStack.Cafe - Kill Your Tech Interview

Q1: What is Entity Framework? ☆

Topics: Entity Framework

Answer:

ADO.NET EF is an ORM (object-relational mapping) which creates a higher abstract object model over ADO.NET components. So rather than getting into dataset, datatables, command, and connection objects as shown in the below code, you work on higher level domain objects like customers, suppliers, etc.

Q2: What are the *benefits* of using EF? ☆

Topics: Entity Framework

Answer:

The main and the only benefit of EF is it auto-generates code for the Model (middle layer), Data Access Layer, and mapping code, thus reducing a lot of development time.

Q3: What are the different ways of creating these domain/entity objects? ☆☆

Topics: Entity Framework

Answer:

Entity objects can be created in two ways: from a database structure, or by starting from scratch by creating a model.

Q4: What is `pluralize` and `singularize` in the Entity Framework? ☆☆

Topics: Entity Framework

Answer:

“Pluralize” and “Singularize” give meaningful naming conventions to objects. In simple words it says do you want to represent your objects with the below naming convention:

- One Customer record means “Customer” (singular).
- Lot of customer records means “Customer’s” (plural, watch the “s”)

Q5: What is *Code First* approach in Entity Framework? ☆☆

Topics: Entity Framework

Answer:

In **Code First** approach we avoid working with the Visual Designer of Entity Framework. In other words the EDMX file is excluded from the solution. So you now have complete control over the context class as well as the entity classes.

Q6: What are `scalar` and `navigation` properties in Entity Framework? ☆☆

Topics: Entity Framework

Answer:

- *Scalar properties* are those where actual values are contained in the entities. Normally a scalar property will map to a database field.
- *Navigation properties* help to navigate from one entity to another entity directly in the code.

Q7: Mention in what all scenarios Entity Framework can be applicable? ☆☆

Topics: Entity Framework

Answer:

Entity Framework can be applicable in three scenarios

- If you have an existing database already or you want to build your database first than other parts of the application
- If your prime focus is your domain classes and then create the database from your domain classes
- If you want to design your database schema on the visual designer and create the classes and database

Q8: Mention what is *Code First Approach* and *Model First Approach* in Entity Framework? ☆☆

Topics: Entity Framework

Answer:

In Entity Framework,

- **Model First Approach:** In this approach we create entities, relationships directly on the design surface of EDMX.
- **Code Approach:** For code approach we avoid working with the visual designer or entity framework.

Q9: What is *Conceptual Model*? ☆☆

Topics: Entity Framework

Answer:

Conceptual Models are the model classes which contain the relationships. These are independent of the database design.

Q10: What is *Storage Model*? ☆☆

Topics: Entity Framework

Answer:

Storage Models are our database design models, which contains database tables, views, stored procs and keys with relationships.

Q11: What is *Mapping*? ☆☆

Topics: Entity Framework

Answer:

The Mapping will have the information on how the Conceptual Models are mapped to Storage Models.

Q12: What is the purpose of a `DBContext` class? ☆☆

Topics: Entity Framework

Answer:

You can think of `DbContext` as the database connection and a set of tables, and `DbSet` as a representation of the tables themselves. The `DbContext` allows you to link your model properties (presumably using the Entity Framework) to your database with a connection string.

Later, when you wish to refer to a database in your controller to handle data, you reference the `DbContext`.

Q13: What is *migration* in Entity Framework? ☆☆

Topics: Entity Framework

Answer:

Entity Framework introduced a migration tool that automatically updates the database schema when your model changes without losing any existing data or other database objects.

There are two kinds of Migration:

- Automated Migration
- Code-based Migration