Getting Acquainted with Entity Framework Core

Kevin Dockx
Architect

@KevinDockx | www.kevindockx.com

Coming Up

Introducing Entity Framework Core

Working with migrations

Seeding the database

Safely storing sensitive data

What about SQL injection?

Object-Relational Mapping

Object-Relational Mapping (ORM) is a technique that lets you query and manipulate data from a database using an object-oriented paradigm

Introducing Entity Framework Core

Problem	Solution
Relational models and object models do not work very well together	An ORM: the library that implements the object-relational mapping technique
Tabular database format versus an interconnected object graph	It takes care of mapping between that tabular format and the object graph

Introducing Entity Framework Core

Supports a variety of databases, even non-relational ones

Code-first or database-first approach

Demo

Creating entity classes

Demo

Creating a `DbContext`

Demo

Working with migrations

Demo

Seeding the database with data

Demo

Safely storing sensitive configuration data

SQL injection

An attack in which malicious code is inserted into strings that are later passed to an instance of a database server for parsing and execution

```
// inputted string, "cityName" has value Antwerp
var sql = "select * from Cities where Name = '" + cityName + "'";
```

What About SQL Injection?

A consumer can input a city name to filter on

```
// inputted string, "cityName" has value Antwerp
var sql = "select * from Cities where Name = "'" + cityName + “’”;



SELECT * from Cities WHERE Name = 'Antwerp'
```

What About SQL Injection?

A consumer can input a city name to filter on

```
// inputted string, "cityName" has value Antwerp'; drop table Cities--
var sql = "select * from Cities where Name = "'" + cityName + “’”;
```

What About SQL Injection?

A consumer can input a city name to filter on

```
// inputted string, "cityName" has value Antwerp'; drop table Cities--
var sql = "select * from Cities where Name = "'" + cityName + “’”;
```

```
SELECT* FROM Cities WHERE Name = 'Antwerp'; drop table Cities--'
```

What About SQL Injection?

A consumer can input a city name to filter on

Code that constructs SQL statements should be reviewed for injection vulnerabilities

```
// result without parameterizing
SELECT* FROM Cities WHERE Name = 'Antwerp'; drop table Cities--’
```

What About SQL Injection?

Can be mitigated by encapsulating and parameterizing SQL commands
Pass user input as a `DbParameter` when constructing the query

```
// result without parameterizing
SELECT* FROM Cities WHERE Name = 'Antwerp'; drop table Cities--’


// result with parameterizing
SELECT* FROM Cities WHERE Name = '''Antwerp''; drop table Cities--'''
```

What About SQL Injection?

Can be mitigated by encapsulating and parameterizing SQL commands
Pass user input as a `DbParameter` when constructing the query

What About SQL Injection?

Safe approaches	Vs	Potentially unsafe approaches
LINQ queries `.FromSql()` (when passing user input as parameter data) `.FromSqlInterpolated()` (when passing user input as parameter data)		`.FromSqlRaw()` Manually sanitizing the inputted values is required

Summary

Object-Relational Mapping (ORM) is a technique that lets you query and manipulate data from a database using an object-oriented paradigm

Summary

Separate entity model from the outer facing model (DTOs)

- Use conventions or annotations for keys, required fields, ...

Summary

The `DbContext` represents a session with the database and can be used to query and save instances of entities

Migrations allow you to provide code to change the database from one version to another

Summary

Seeding the database means providing it with data to start with

- `HasData()` method when configuring the model

Use environment variables for safer storage of sensitive data

Summary

EF Core mostly protects against SQL injection out of the box but be careful with the `.FromSqlRaw` method

Up Next:

Using Entity Framework Core in Your Controllers