Search Topic: Fluent API Relation in Entity Frameworks

The Fluent API in Entity Framework is a way to configure entity relationships, mappings, and other model configurations using a fluent-style interface rather than data annotations or conventions.

**Advantages of Fluent API Relationships:**

* **Explicitness and Readability:** Provides a clear and explicit way to define entity relationships, making it easier to understand the model configuration.
* **Complex Configurations**: Offers capabilities for complex mappings and configurations that might not be achievable through data annotations.
* **Separation of Concerns**: Separates the configuration from the entity classes, promoting cleaner code and better maintainability.

Using Fluent API for configuring entity relationships in Entity Framework is particularly beneficial when dealing with complex models or when needing more explicit control over the configuration aspects of the database schema.

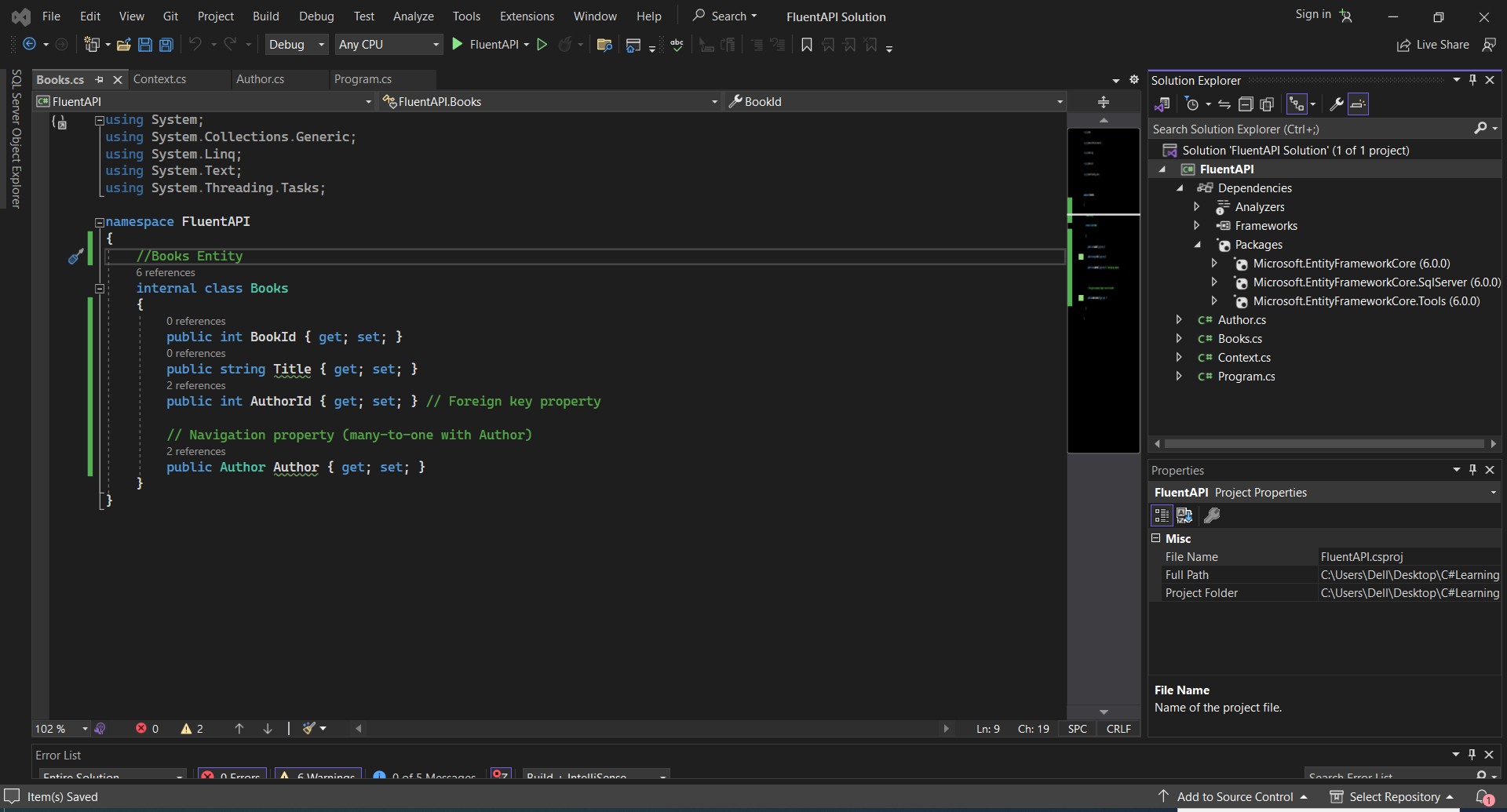
**Key Points about Fluent API Relationships:**

1. **Configuration Flexibility**:
   * Fluent API provides more flexibility and explicit control over configuring entity relationships compared to data annotations. It allows for complex mappings and configurations.
2. **Separation of Concerns**:
   * Using Fluent API separates the configuration from the domain classes themselves, making the codebase cleaner and more maintainable, especially when dealing with complex relationships.
3. **Configuration in DbContext**:
   * The Fluent API configurations are generally done in the **OnModelCreating** method of the **DbContext** class.
4. **Configuring Relationships**:
   * Fluent API allows configuring various types of relationships between entities, such as one-to-one, one-to-many, many-to-one, and many-to-many relationships.
   * You can specify foreign keys, navigation properties, cascade delete behavior, and other relationship-specific settings using Fluent API.

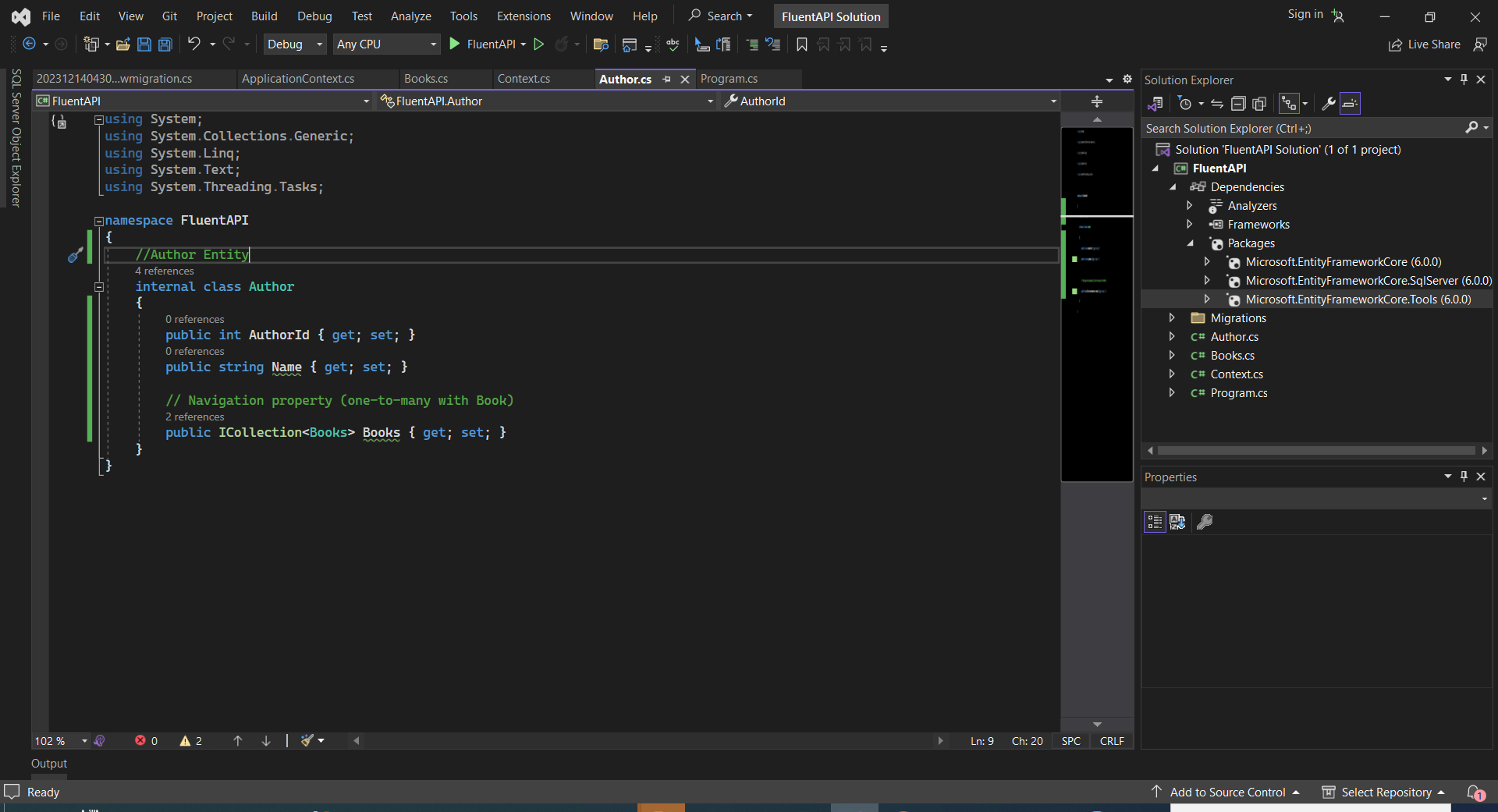
Example implementation: one-to-many relationship using the Fluent API.

If we have two entities: **Author** and **Book**, and each book is associated with an author through a foreign key relationship.

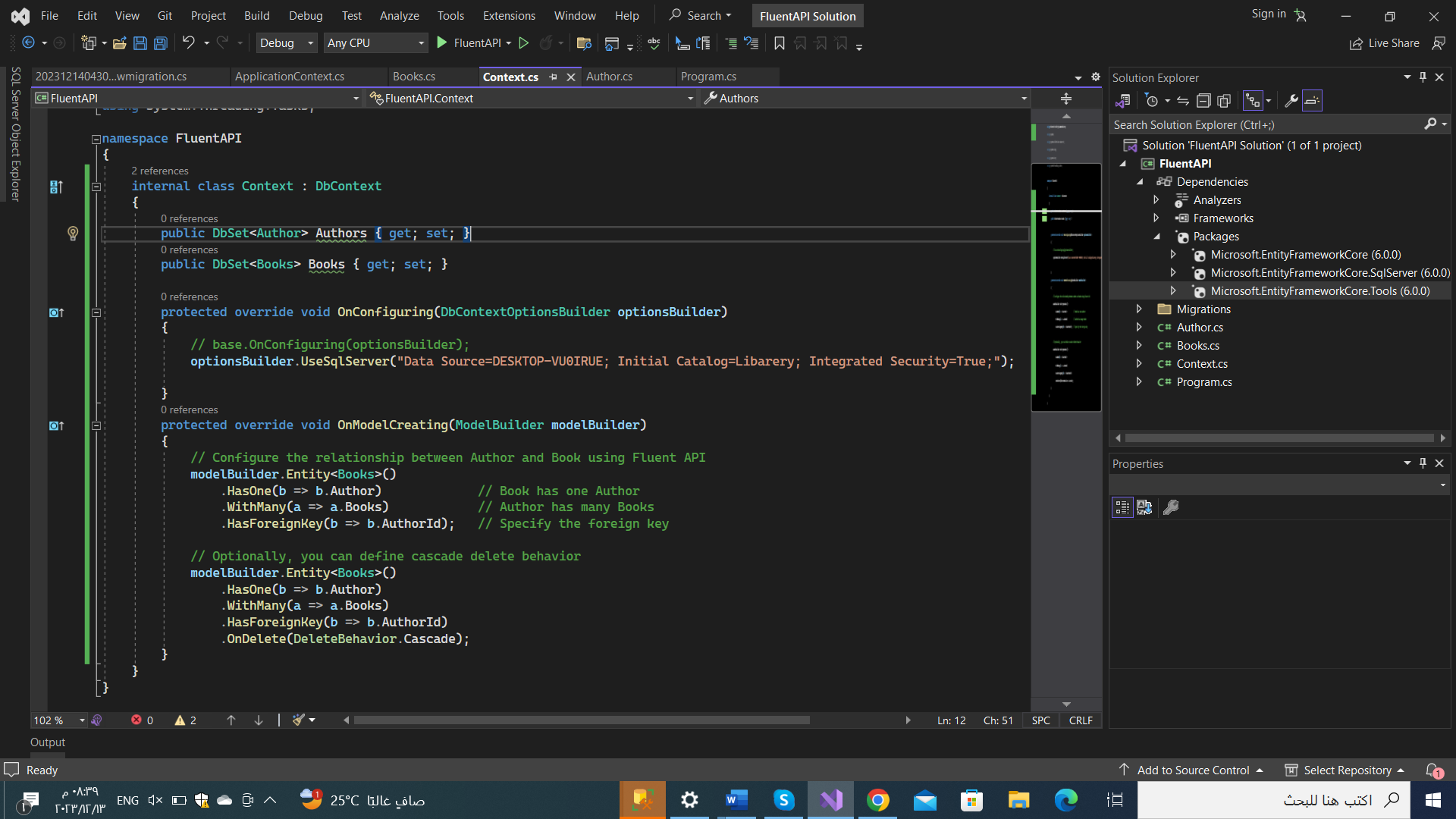
**Books Entity:**



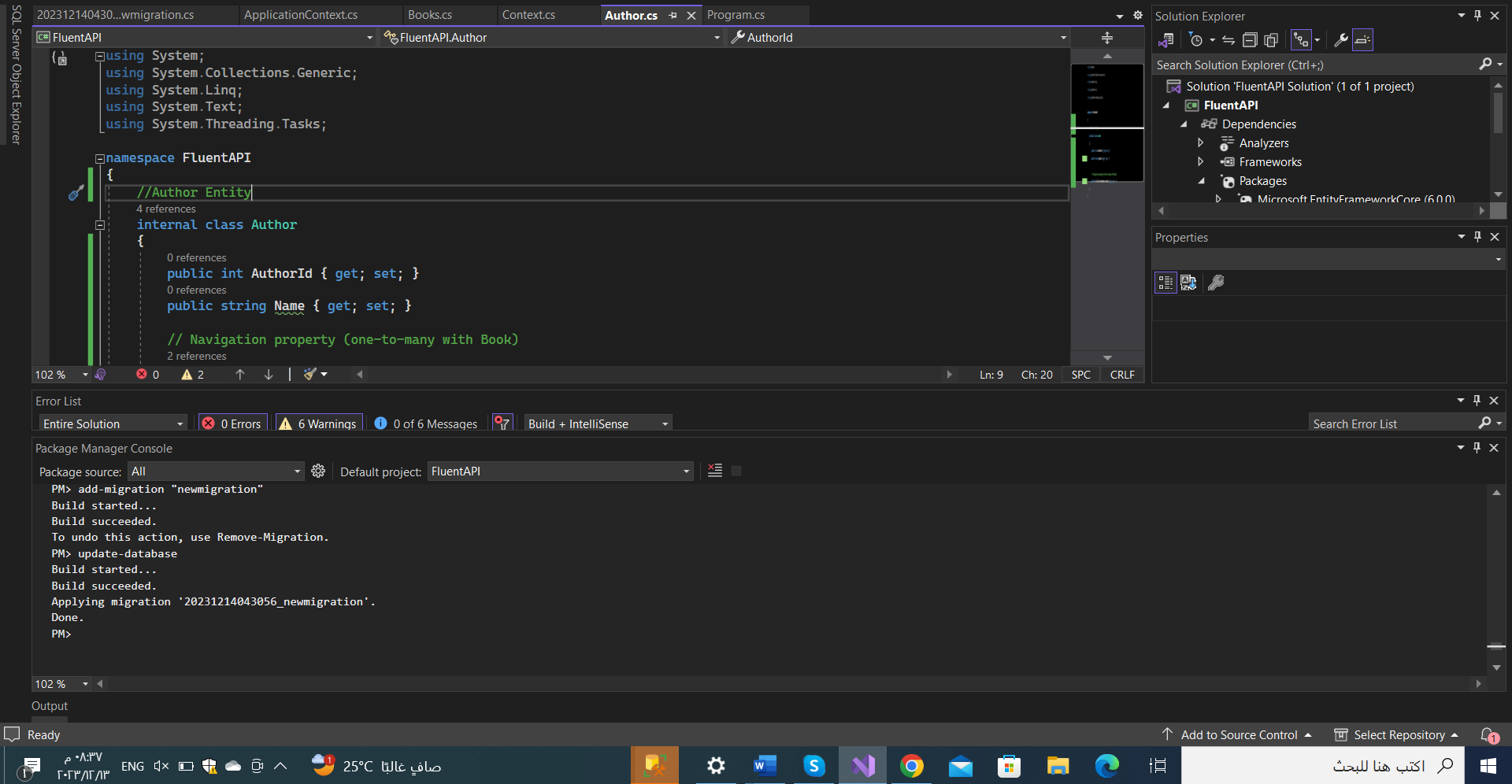
**Author Entity**:



**DbContext Configuration using Fluent API:**



Using add-migration and update-database commands 🡺 to made changes in database(creating connection with database, create tables(Entity).



You can see here in database , two tables created with specified data.

