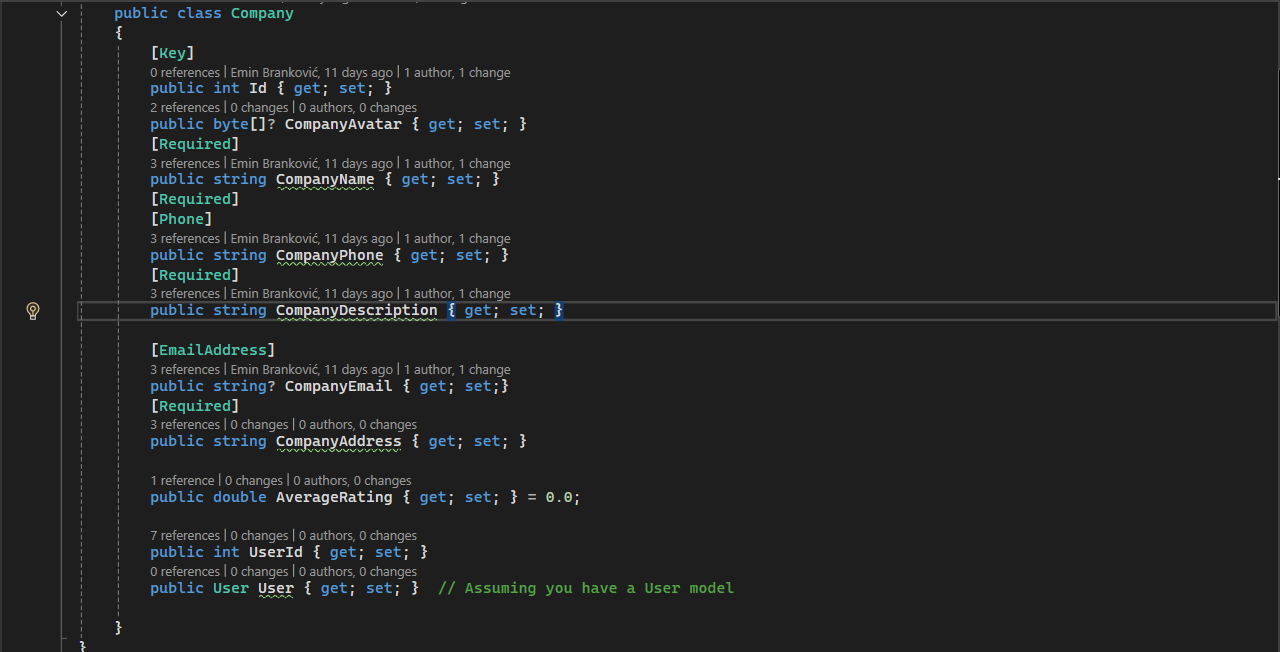
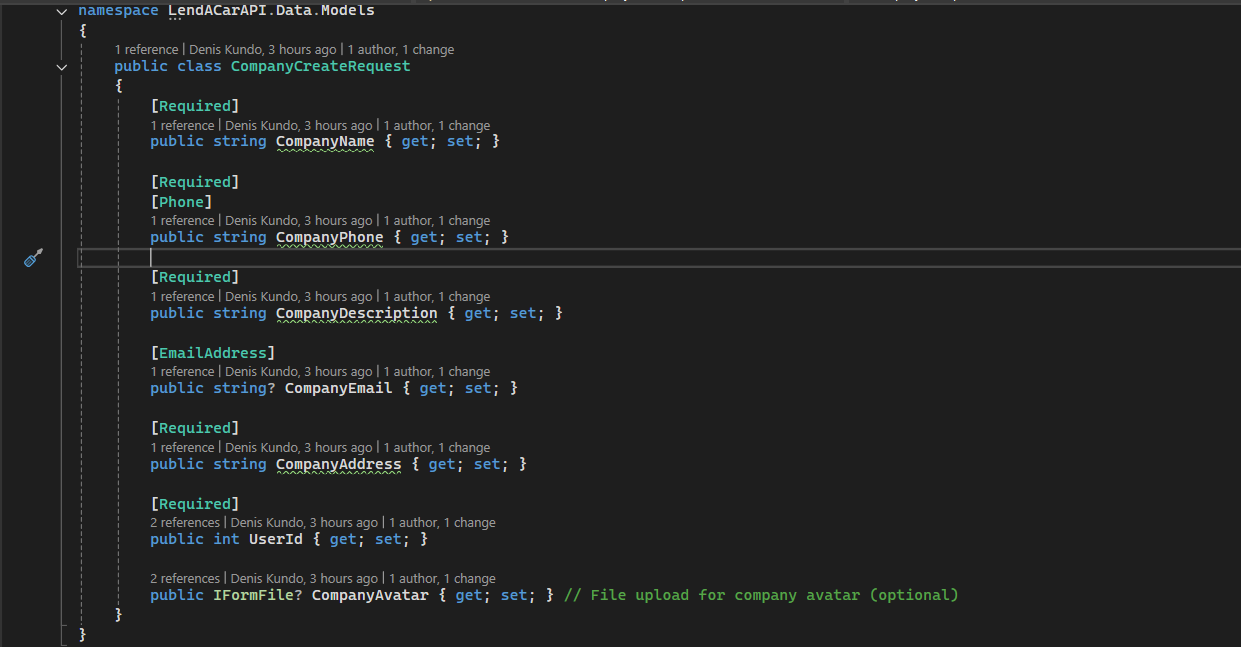
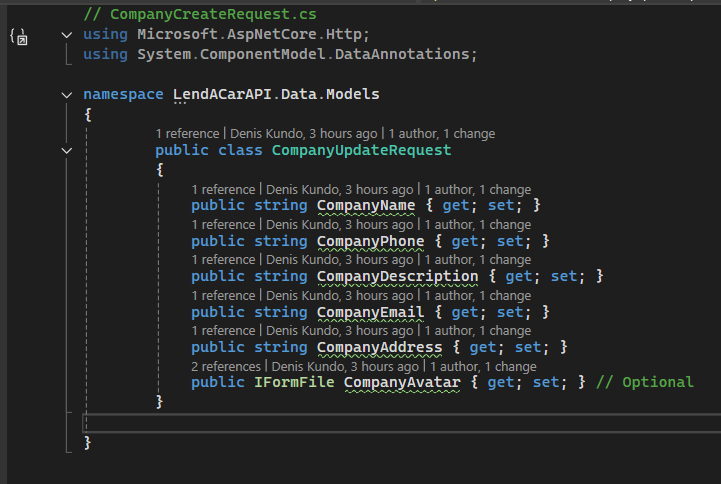
**Core Features**

1. **Model Definitions**:
   * Company: Represents a company's data in the database, with properties like CompanyAvatar, CompanyName, CompanyPhone, CompanyDescription, CompanyEmail, CompanyAddress, AverageRating, and UserId (foreign key to a User).



* + CompanyCreateRequest and CompanyUpdateRequest: Define data transfer objects (DTOs) for creating and updating a Company, respectively.

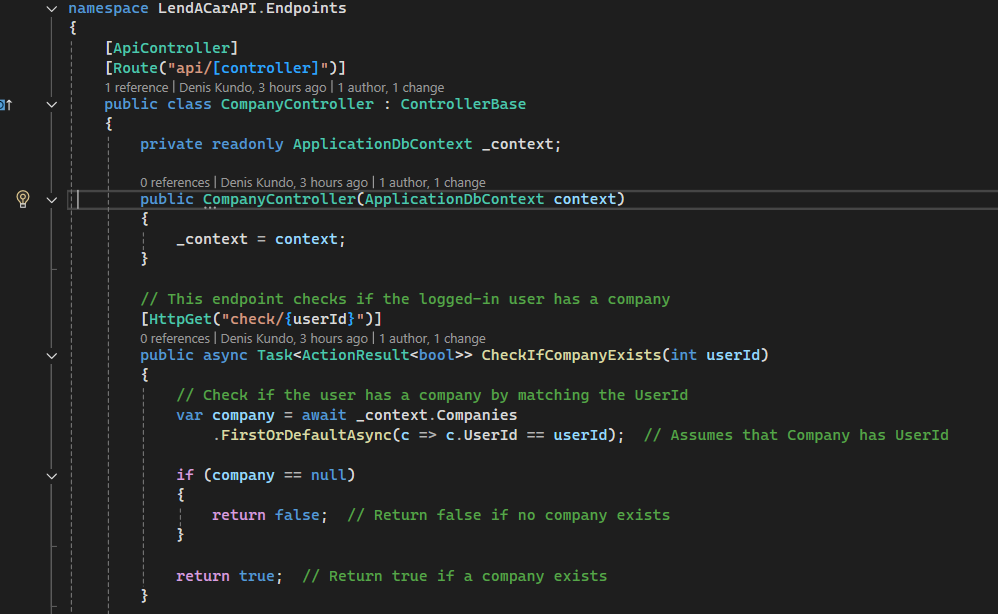




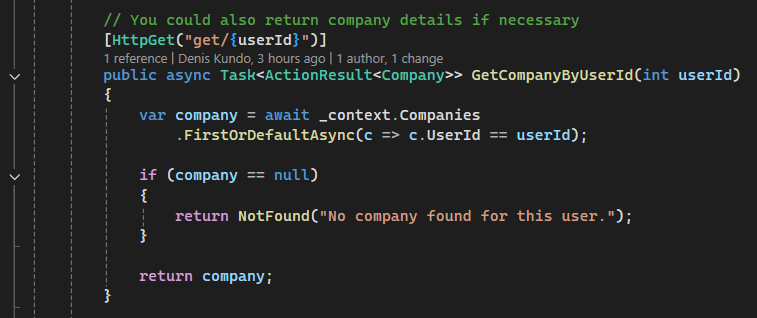
1. **Controller (CompanyController)**:
   * Endpoints manage company operations for users, including checking existence, creating, retrieving, updating, and deleting a company.

**Endpoints Overview**

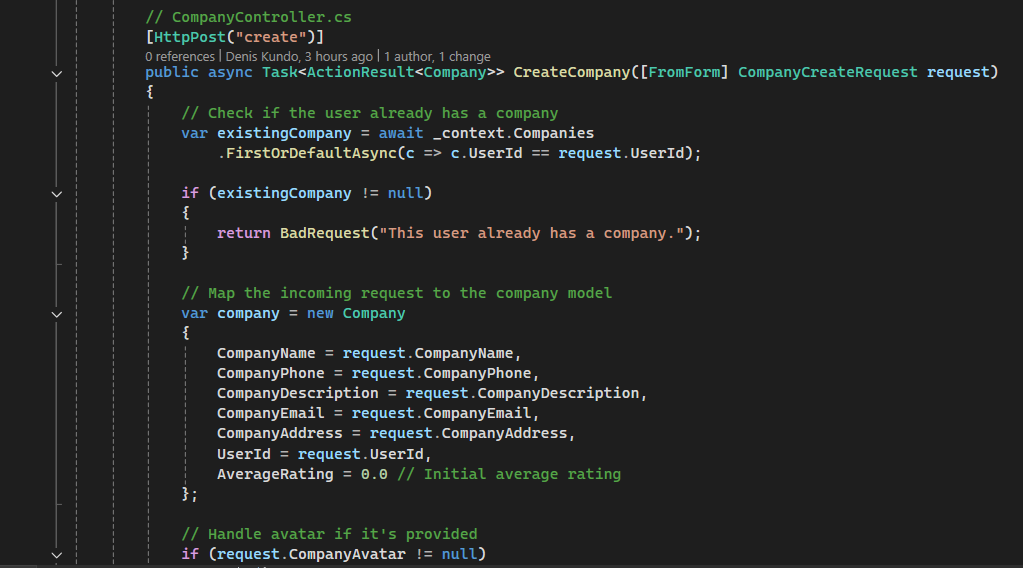
1. **Check if Company Exists**:
   * GET api/company/check/{userId}
   * Returns true if a company exists for the given user; otherwise, false.

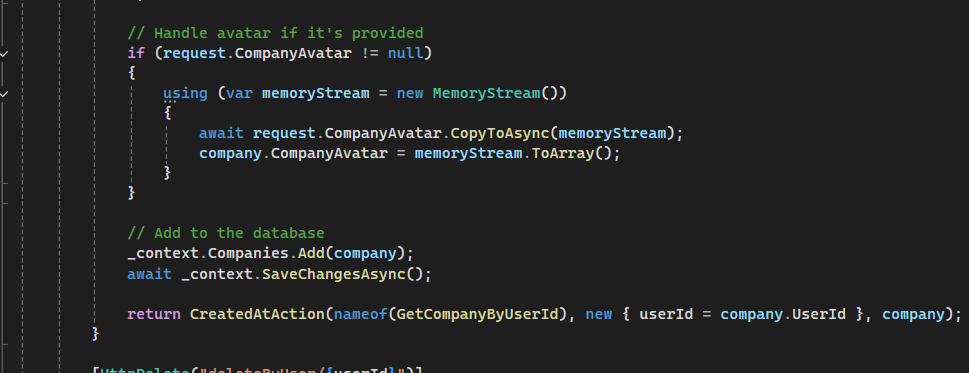


1. **Get Company Details by User**:
   * GET api/company/get/{userId}
   * Retrieves the Company associated with a specific userId.

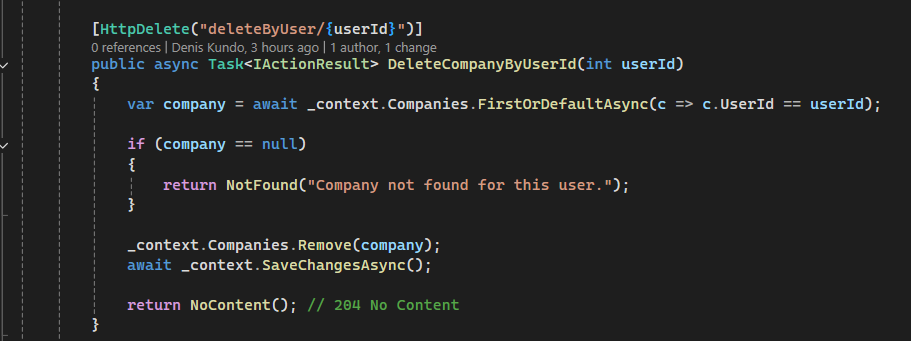


1. **Create a Company**:
   * POST api/company/create
   * Accepts CompanyCreateRequest via FromForm. Creates a new company for the user, checking if one already exists. Supports optional avatar upload.

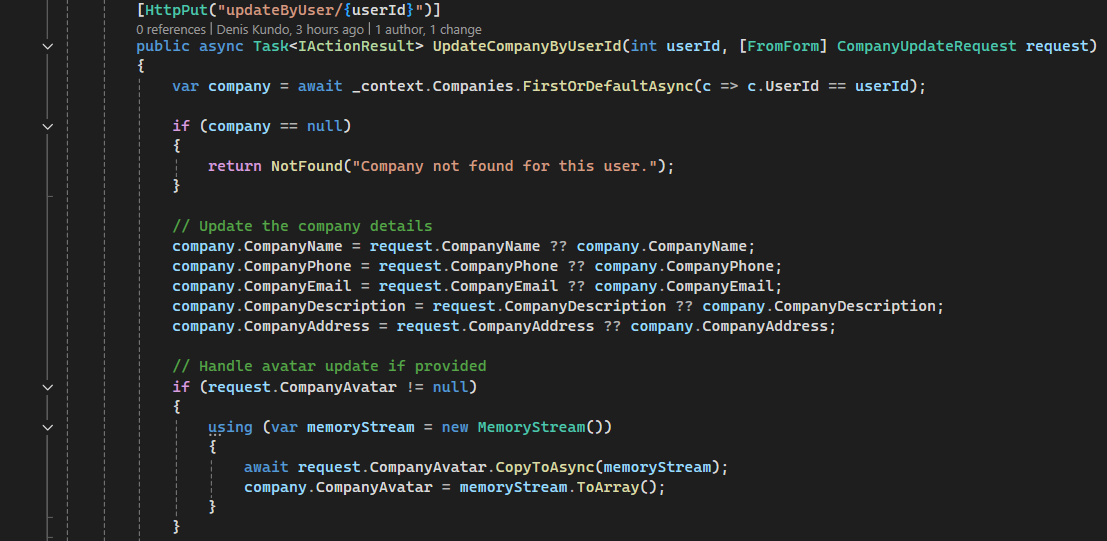


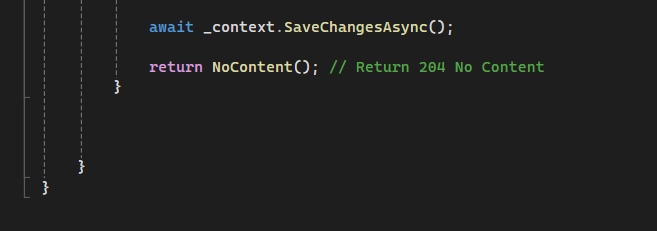


1. **Delete a Company**:
   * DELETE api/company/deleteByUser/{userId}
   * Deletes the Company associated with the provided userId.



1. **Update a Company**:
   * PUT api/company/updateByUser/{userId}
   * Updates the Company details for a given userId. Supports partial updates, including avatar replacement.





**Key Functionalities**

1. **Avatar Handling**:
   * Avatar is managed as a byte[] in the database, allowing image uploads.
   * Memory streams are used to read and save uploaded files.
2. **Validation**:
   * DTOs enforce data integrity using annotations ([Required], [Phone], [EmailAddress]).
3. **Database Operations**:
   * Entity Framework Core handles CRUD operations using DbContext.
4. **Default Values**:
   * AverageRating defaults to 0.0.
5. **Error Handling**:
   * Returns appropriate HTTP status codes (e.g., 404 Not Found, 400 Bad Request, 204 No Content).