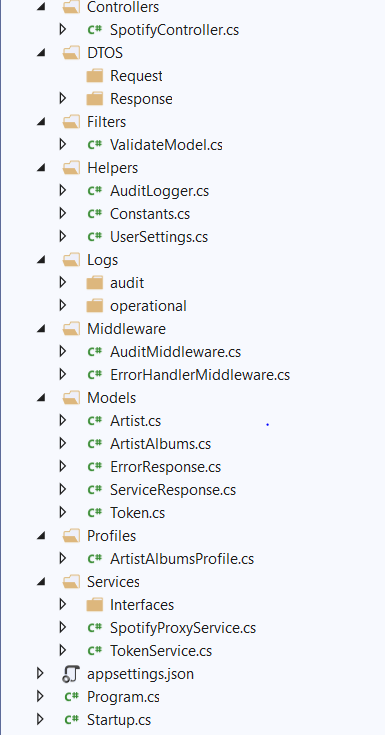
Architecture Followed to develop API’s on ASP.NET Core

Below Screenshot shows the project structure.



**Controllers** -> A controller contains the flow control logic for an ASP.NET application. A controller determines what response to send back to a user when a user makes a browser request

**DTO** ->DTO (Data Transfer objects) is a data container for moving data. It does not contain any business logic. We can define the models in DTOs for request or response such that it does not break the contract.

**Filters ->** Filters in ASP.NET Core allow code to be run before or after specific stages in the request processing pipeline.

**Validate Model** ->which extends ActionFilterAttribute to validate the request model. It will be helpful to through bad request and error message if request violates the model.

**Helpers** -> Helpers folder consists of constants, models, etc. which will be used throughout the application.

**Logs**-> using **Serilog** and **Sinks** in .Net we can log in Files or google cloud (stackdriver)

Maintaining two types of logging.

**Audit** -> Audit logs are logged in this folder which contains all the requests and responses of the API’s, Downstream API’s, any sensitive information can be logged.

**Operational** -> Application related logs are logged in this folder.

**Middlewares ->** Middleware in ASP.NET Core controls how our application responds to HTTP requests. Custom Middleware are maintained in this folder. We have been using two middleware as shown.

**AuditMiddleware ->** This Middleware is used to log audit information.

**ErrorHandlingMiddleware ->**This middleware is used to handle any exception that occurred throughout the application and sends response.

**Models ->** This Folder consists of all models (contains setters and getters) which are used within the application.

**Profiles ->** Since we are using Automapper to transform one object to other. We use Automapper profile class to create mapping between classes

**Services ->** The services (custom types or classes) which programmer create for application are maintained in this folder. It can contain any number of interfaces and its implementation.

**List of Services/ Library that have been used by adding as dependencies are:**

1. **MiniProfiler ->** MiniProfiler is a library and UI for profiling your application. By letting you see where your time is spent, which queries are run, and any other custom timings you want to add. It helps you debug issues and optimize performance.
2. **Swagger ->** Swagger is a specification for documenting REST API.
3. **ApiVersioning ->** Versioning APIs is done with this library.