Technologies:  
.Net Core, EF Core (code first), Authentication (depends on main applications auth), MSSql Server.  
  
News API probably going to be a part of the AvaTrade microservices.  
  
Project separates to 2 functionalities: Getting news, and Showing news. Showing – it may have different implementation of specific components, depends on developer. Basic suggestions – Use Dependency injection, make scalable system ( for example use httpClientFactory for different sources of news, same for Models of that news from API’s). Avoid Repositories, use dbContext in services.  
  
The main difference might be in the implementation of Ticker for Getting news.  
  
Components:  
1. Consume trading news every 1 hour from any provider available i.e. https://polygon.io/   
Here could be a lot of different solutions, as in task description asked 2, here:

* BackgroundService – register a custom NewsGetterBackgroundService : BackgroundService which will get news every hour while we didn’t send cancelation token.
* Add middleware (for example with static field LastTickDate) – and with every request we will check if current DateTime - LastTickDate > 1 hour – we will send request to get new news

2. For each news items a. Enrich the data with any information you can think of i.e. ticker chart b. Save the data in any storage solution.   
 - Service for saving news with mapping.  
 - Validation class ( for example Unique new ID for specific source )  
 - Models that come from API’s, entities for saving in DB.  
 - Test cases (unique, saving, connection error, wrong URL etc.)

3. Create webapi that any client can reach and implement the following.  
 - Controllers : AuthController – suggest 2 controller, for Subscription and for Getting news  
 - Service with interfaces.   
 - ViewModels and Entities. (News, Instrument, Source of the news could be enum)  
 - Additional classes, like Mapper, DIExtentions, LoggingExtentions, QueryExtentions (skip, take for paging)  
 - Exception and Configuration classes.