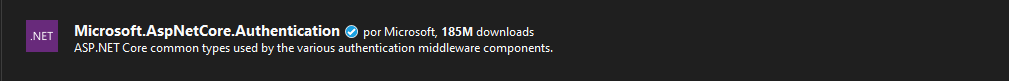
**JWT ASP.NET**

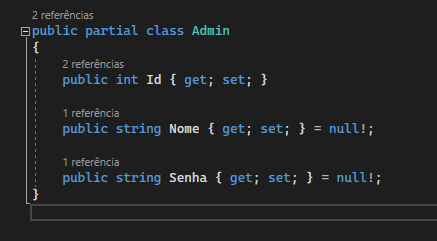
**Instalando pacotes:**

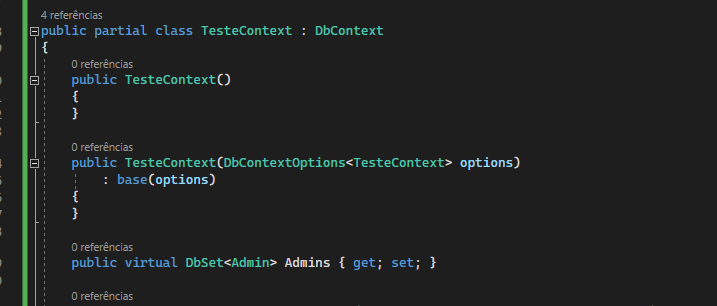
Antes de configurarmos o Auth devemos instalar os seguintes pacotes:





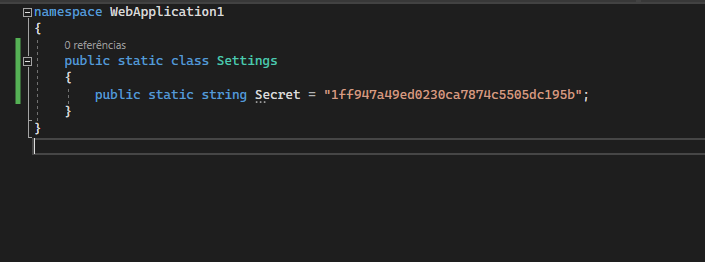
Após isso devemos configurar uma forma de “login” pode ser utilizando o .net Identity, alguma entidade no banco de dados ou por codigo mesmo:





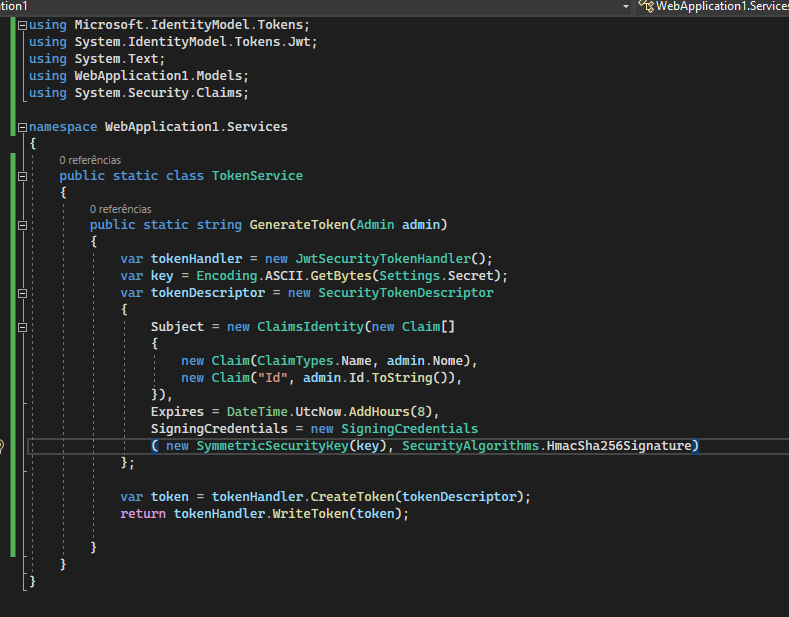
**Configurando a chave secreta:**

Bom para gerarmos os tokens devemos primeiro criar uma chave secreta para a aplicação, vamos fazer isso criando uma classe **settings.cs** na raiz do projeto e adicionando a variavel secreta:



**Gerando o Token:**

Após esses passos vamos configurar o nosso serviço de geração de token:



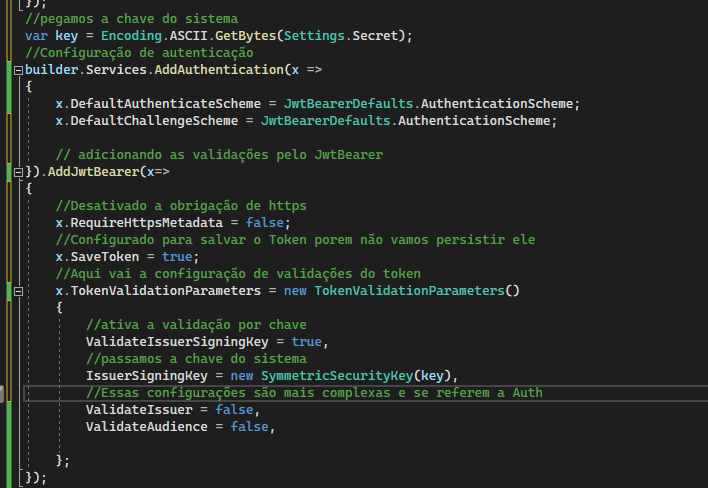
Basicamente o nosso serviço que ira gerar o token gera um token com base na JwtSecurity, basicamente um token é composto por “subject”, “Expires” e “SigningCredentials”, subject é entendido como informações do usuario e nos podemos escolher quais serão elas, o Expires determina o tempo que o token gerado sera valido e o SigningCredentials é basicamente a chave do sistema que geramos na classe anterior.

**Configurando a classe Program.cs:**

Devemos configurar a nossa classe do programa com UseAuthentication e UseAuthorization:

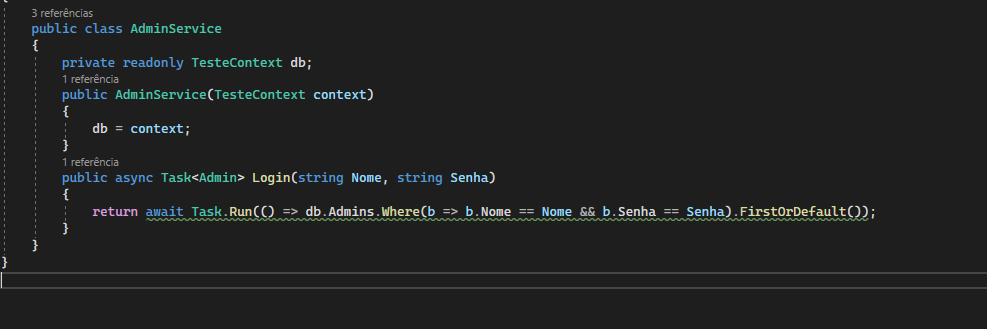


Após isso devemos configurar o processo de Authentication no Services:

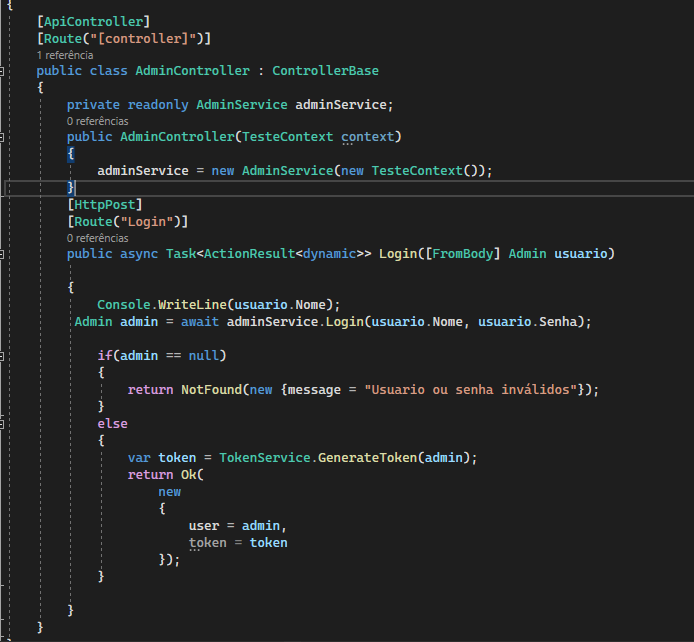


**Criando um método de login:**

Bom após termos configurado os tokens na aplicação podemos criar um método login que ira criar um token para um usuario, primeiro vamos criar o service de Admin e criar o método de login:

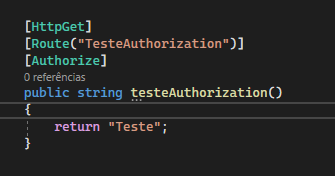


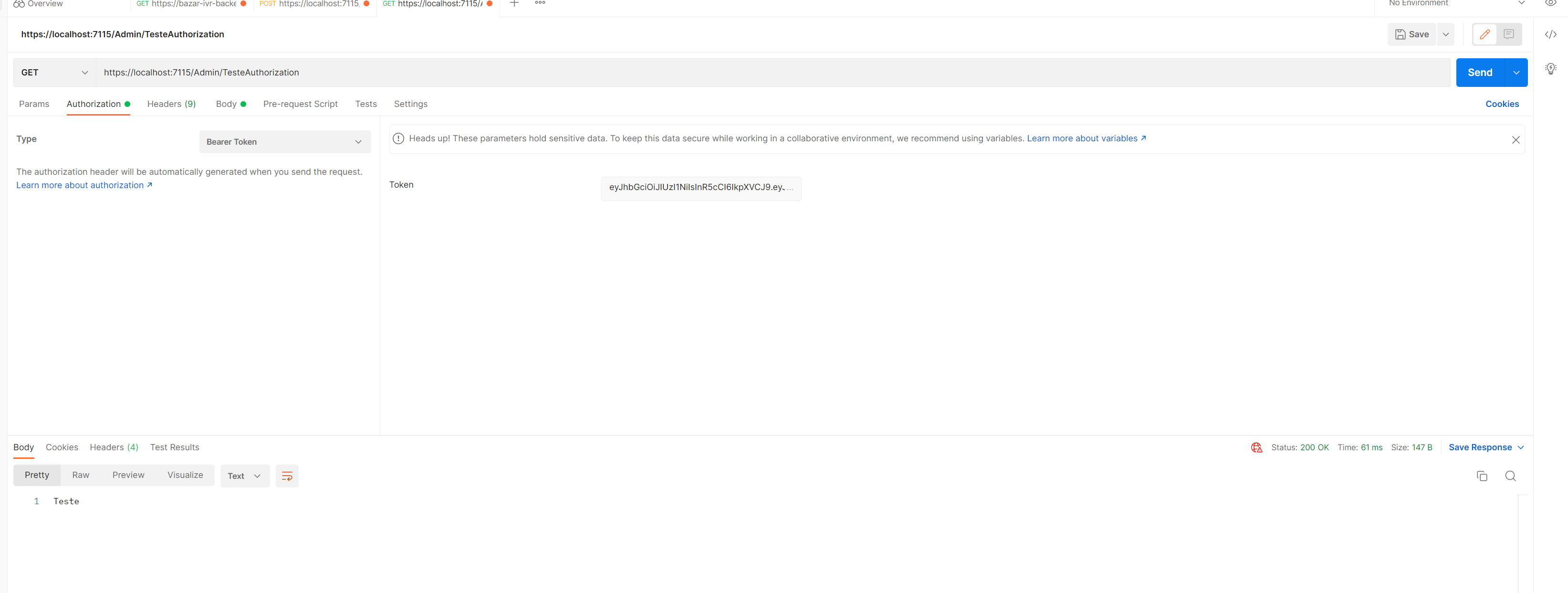
Em seguida vamos criar o seu controller que possui o método Login:



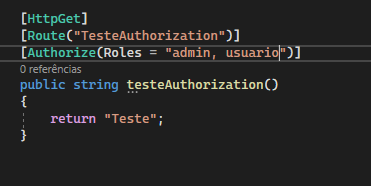
**Aplicando autorizações nos métodos:**

Após termos gerado um token podemos adicionar uma verificação nos métodos caso o token passado seja errado ele vai retornar o error 401:



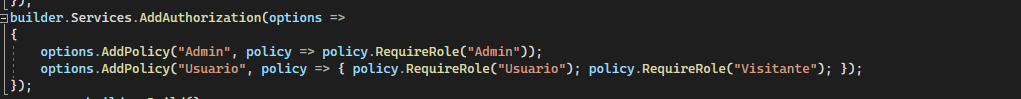


Podemos tambem adicionar alguma propriedade do token a ser verificada no Roles, basicamente esse Roles é uma propriedade do token e pode ser uma função do usuario ou estatus:



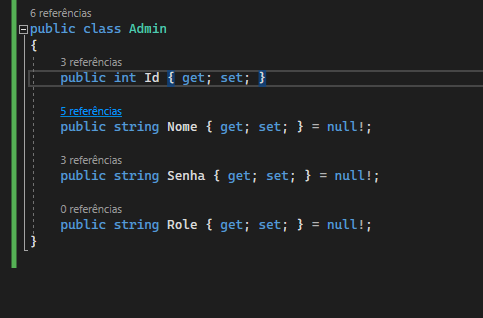
**Melhorando a autorização e adicionando as politicas no ASP.NET 6:**

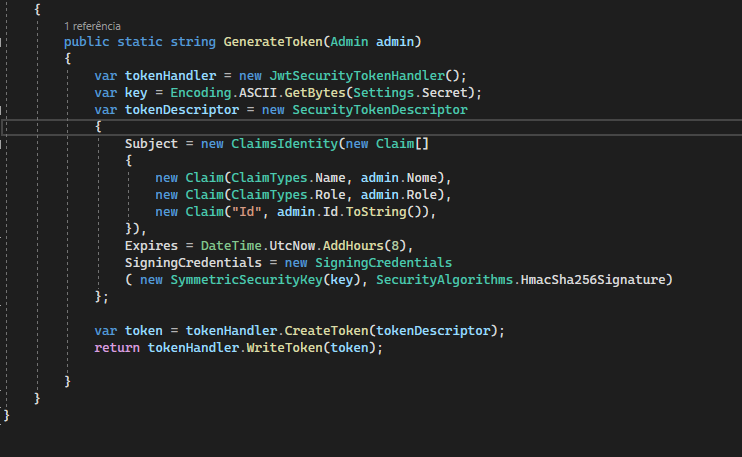
Apartir do .net 6 temos um conceito de “polices” que são politicas de usuarios já definidas:



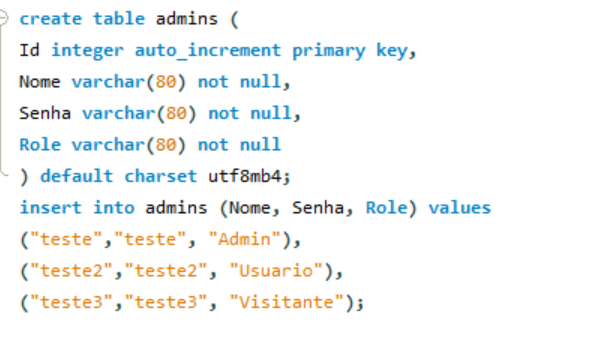
Basicamente podemos definir niveis de usuarios com Base no Name do Token ou na Role.

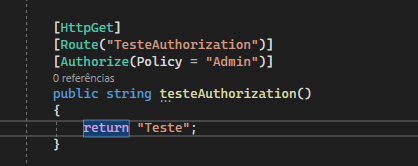
Para isso vamos adicionar um Role na nossa classe Admin e adicionar o Role no Token:





Claramente devemos mudar a estrutura dos dados no banco de dados e adicionar os usuarios de exemplo:



Após isso basta adicionarmos o atributo Policy as nossas anotações “Authorize”:  


Dessa forma os usuarios que possuem uma “Policy” diferente não poderão acessar o método mesmo que tenham um token, as “policies” basicamente são agregadores de permissões no sistema assim não precisamos todas as vezes declarar quais são os usuarios validos para acessar o método basta usarmos a “policy” adequada.