

#### Proposta:

- ✓ Iremos utilizar VaotacaoApi como base de estudo.
- ✓ Nela iremos adicionar uma autenticação Jwt.

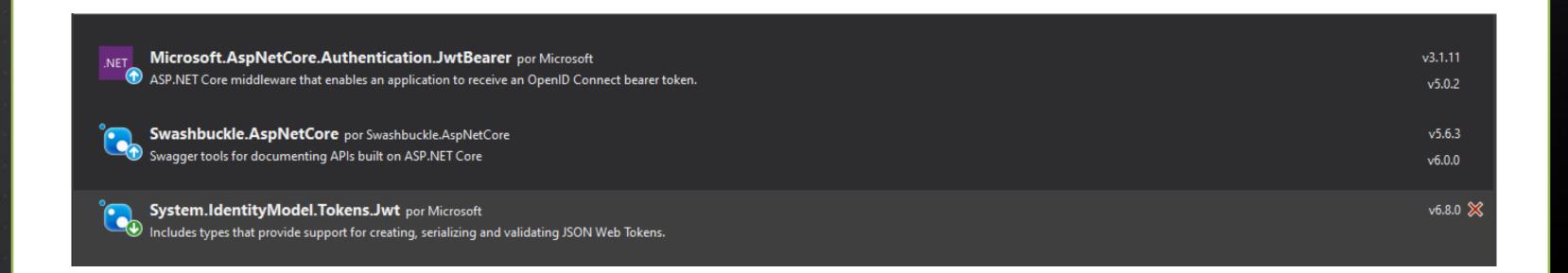


#### Passo a passo:

- ✓ Adicionar pacotes nuget.
- ✓ Configurar nossa chave.
- ✓ Criar nossa classe que irá gerar os tokens.
- ✓ Configurar a autenticação.
- ✓ Configurar o swagger.
- ✓ Adicionar anotações nas controllers.
- ✓ Configurar o retorno do enpoint de login.



### Adicionar pacotes nuget:





Atualizar nosso appSettings.json:



Criar nossa classe para geração de tokens:



Configurar a autenticação no arquivo Startup.cs:

```
region [+] Autenticação JWT
var keyString = Configuration.GetSection("SettingsAPI:ChaveJWT").Get<string>();
var key = Encoding.ASCII.GetBytes(keyString);
services.AddAuthentication(x =>
   x.DefaultAuthenticateScheme = JwtBearerDefaults.AuthenticationScheme;
   x.DefaultChallengeScheme = JwtBearerDefaults.AuthenticationScheme;
}).AddJwtBearer(x =>
   x.RequireHttpsMetadata = true;
   x.SaveToken = true;
    x.TokenValidationParameters = new TokenValidationParameters
       ValidateIssuerSigningKey = true,
       IssuerSigningKey = new SymmetricSecurityKey(key),
       ValidateIssuer = false,
       ValidateAudience = false
});
#endregion
```



### Atualizar a configuração do swagger no Startup.cs:

```
c.AddSecurityDefinition("Bearer", new OpenApiSecurityScheme
{
    Description = "Para autenticar use a palavra 'Bearer' + (um espaço entre a palavra Bearer e o Token) + 'Token'",
    Name = "Authorization",
    In = ParameterLocation.Header,
    Type = SecuritySchemeType.ApiKey
});

var scheme = new OpenApiSecurityScheme { Reference = new OpenApiReference { Type = ReferenceType.SecurityScheme, Id = "bearer" } };
c.AddSecurityRequirement(new OpenApiSecurityRequirement
{
    [scheme] = new List<string>()
});
```



#### Adicionar anotações nas controllers:

```
[Consumes("application/json")]
[Produces("application/json")]
[ApiController]
[Authorize]
1 referência
public class FilmeController : ControllerBase
```

```
[Consumes("application/json")]
[Produces("application/json")]
[ApiController]
[Authorize]
1 referência
public class VotoController : ControllerBase
```

```
[Consumes("application/json")]
[Produces("application/json")]
[ApiController]
[Authorize]
1 referência
public class UsuarioController : ControllerBase
```

```
[AllowAnonymous]
[HttpPost]
[Route("v1/usuarios/login")]
O referências
public ICommandResult UsuarioLogin([FromBody] LogarUsuarioCommand command)...
```



### Configurar o retorno do enpoint de login:

```
[AllowAnonymous]
[HttpPost]
[Route("v1/usuarios/login")]
O referências
public ICommandResult UsuarioLogin([FromBody] LogarUsuarioCommand command)
{
    try
    {
        var result = _handler.Handle(command);
        var tokenJWT = _tokenJWTService.GenerateToken((UsuarioQueryResult)result.Data);
        result.Data = tokenJWT;
        return result;
    }
    catch (Exception ex)
    {
        throw ex;
    }
}
```

