**JWT Servisi**

IJwtService ara yüzü, Music Tracker uygulamasında kullanıcı kimlik doğrulama işlemleri için JSON Web Token (JWT) üretimini sağlayan bir servis ara yüzüdür. JWT, kullanıcının kimliğini doğrulayan ve dijital olarak imzalanmış bir token yapısıdır. Uygulama genelinde, oturum yönetimi için kullanılır ve Authorize gerektiren her API isteği sırasında kullanıcının doğruluğunu doğrulamak amacıyla kontrol edilir.

Bu ara yüzü projedeki diğer bileşenlerin, kullanıcı kimliğine özel JWT oluşturulabilmesini sağlar. Token içinde, kullanıcının benzersiz kimliği (userId) gibi talepler (claims) yer alır. Token belirli bir süre geçerli olacak şekilde üretilir ve HMAC-SHA256 algoritması ile imzalanır.

**Servis Metodu:**

**GenerateToken Metodu**

GenerateToken metodu, bir kullanıcı kimliğine (userId) karşılık gelen JWT (Json Web Token) değerini üretir. Bu token, uygulama içinde kullanıcıyı doğrulamak ve güvenli API erişimi sağlamak amacıyla kullanılır.

Kodun Genel Akışı:

1. Gizli anahtar Secrets.JwtSecretKey sabitinden alınır ve UTF-8 formatında byte dizisine (byte[]) dönüştürülür.
2. Token’e eklenecek talepler (claims) tanımlanır.
   * ClaimTypes.NameIdentifier: kullanıcının benzersiz kimliği (userId) eklenir.
   * Jti: Token’e ait benzersiz bir kimlik (UUID) oluşturulur.
3. JwtSecurityToken sınıfı ile bir token oluşturulur.
   * issuer: Token’in kimin tarafından oluşturulduğunu belirler.
   * audience: Token’in hangi istemci(ler) için geçerli olduğunu belirler.
   * claims: Talepler eklenir.
   * expires: Token’in ne kadar süreyle geçerli olacağı belirlenir.
   * singingCredentials: Token’in hangi algoritma ile ve hangi anahtarlarla imzalanacağı tanımlanır. (HMAC-SHA256 ve secret key).
4. JwtSecurityTokenHandler sınıfı ile oluşturulan token, string formatına çevrilir ve metodun çıktısı olarak döndürülür.

Kod Bloğu:

|  |
| --- |
| public string GenerateToken(string userId)  {  var key = Encoding.UTF8.GetBytes(Secrets.JwtSecretKey);  var claims = new[]  {  new Claim(ClaimTypes.NameIdentifier, userId),  new Claim(JwtRegisteredClaimNames.Jti, Guid.NewGuid().ToString())  };  var token = new JwtSecurityToken(  issuer: \_config["JwtSettings:Issuer"],  audience: \_config["JwtSettings:Audience"],  claims: claims,  expires: DateTime.UtcNow.AddMinutes(double.Parse(\_config["JwtSettings:ExpiryMinutes"])),  signingCredentials: new SigningCredentials(new SymmetricSecurityKey(key), SecurityAlgorithms.HmacSha256)  );  return new JwtSecurityTokenHandler().WriteToken(token);  } |