**🔐 1. tokenHandler**

This is typically an instance of the class:

var tokenHandler = new JwtSecurityTokenHandler();

**✅ Purpose:**

* It is responsible for **creating**, **reading**, **validating**, and **writing** JWT tokens.
* It provides methods like:
  + CreateToken(tokenDescriptor)
  + WriteToken(token)
  + ValidateToken(token, validationParameters, out validatedToken)

**🧾 2. Claims**

Claims are **key-value pairs** that represent information about the user or entity.

**✅ Purpose:**

* They are embedded inside the JWT payload.
* Common claims include:
  + sub (subject or user ID)
  + name
  + email
  + role
  + Custom claims like department, region, etc.

var claims = new[]

{

    new Claim(ClaimTypes.Name, "Siddharth"),

    new Claim(ClaimTypes.Role, "Admin")

};

**🧱 3. tokenDescriptor**

This is an instance of SecurityTokenDescriptor, which defines the **structure and content** of the token to be created.

**✅ Purpose:**

* It includes:
  + **Subject** (claims identity)
  + **Expires** (token expiration time)
  + **SigningCredentials** (used to sign the token)
  + **Issuer** and **Audience** (optional)

var tokenDescriptor = new SecurityTokenDescriptor

{

    Subject = new ClaimsIdentity(claims),

    Expires = DateTime.UtcNow.AddHours(1),

    SigningCredentials = new SigningCredentials(new SymmetricSecurityKey(key), SecurityAlgorithms.HmacSha256Signature)

};