JSON Web Token Kullanımı

14 Mayıs 2020 Perşembe 13:23

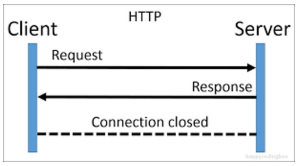
- Stateless Protokol (Bağlantısız Ortam)
  - Bağlantısız Ortamda İstemciyi Yetkilendirme?
- Token Nedir?
- JSON Web Token Nedir?
  - Avantajları
  - JWT Yapısı
  - JWT'lerin Doğrulanması (JWT Verify)
    - Secret Key
      - IConfiguration Servisi
    - base64 Encoding
    - HS256 Algorithm
- ASP.NET Core ile JWT
  - Servis Ayarları
  - JWT Üretmek
  - jwt.io sitesinde jwt'yi doğrulamak
- Identity Framework ile JWT Kullanımı
  - Kimliği Doğrulanmış Kullanıcı İçin JWT Üretmek
  - JWT ile Yetkilendirme
  - Postman ile JWT Test İşlemi
  - JWT ile Rol Tabanlı Yetkilendirme
    - Claim Yapısı

Kaynaklar

- https://github.com/cornflourblue/aspnet-core-3-jwt-authentication-api
- http://www.minepla.net/2018/10/asp-net-core-jwt-kimlik-dogrulama/
- https://cryptii.com/
- https://www.gencayyildiz.com/blog/asp-net-mvc-web-api-token-authentication/
- https://medium.com/kodcular/jwt-nedir-nasil-calisir-1c4a133ff98b

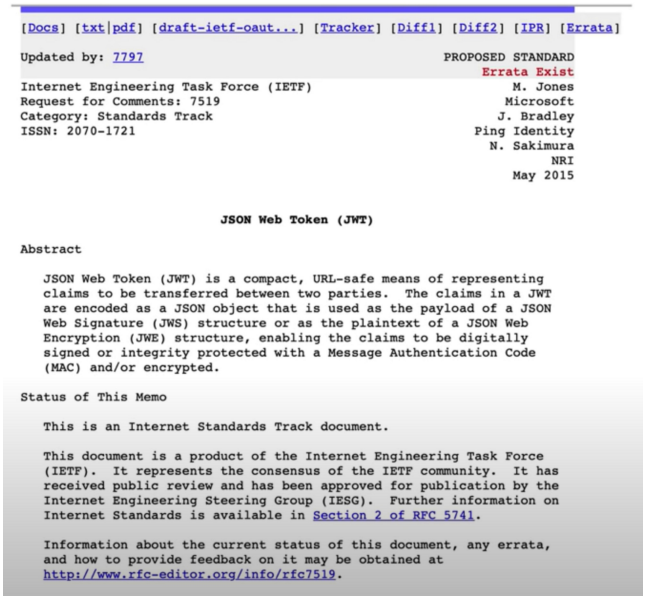
Stateless protocol

İngilizceden çevrilmiştir - Hesaplamada durum bilgisi olmayan bir protokol, alıcı tarafından, genellikle bir sunucu tarafından hiçbir oturum bilgisinin tutulmadığı bir iletişim protokolüdür.
Wikipedia (İngilizce)



JSON Web Token Nedir?

JSON Web Token, tarafların birbirleri arasındaki veri alışverişini ve bunun doğrulamasını sağlayan JSON tabanlı RFC 7519'de tanımlanmış açık bir standarttır. Örneğin bir sunucu, kullanıcının yönetici ayrıcalıklarına sahip olduğunu belirten bir anahtar oluşturabilir ve bunu kullanıcıya gönderebilir. Vikipedi



[Docs] [txt|pdf] [draft-ietf-oaut...] [Tracker] [Diff1] [Diff2] [IPR] [Errata]

Updated by: 7797 — PROPOSED STANDARD, Errata Exist

Internet Engineering Task Force (IETF) — M. Jones, Microsoft
Request for Comments: 7519 — J. Bradley, Ping Identity
Category: Standards Track — N. Sakimura, NRI
ISSN: 2070-1721 — May 2015

JSON Web Token (JWT)

Abstract

JSON Web Token (JWT) is a compact, URL-safe means of representing claims to be transferred between two parties. The claims in a JWT are encoded as a JSON object that is used as the payload of a JSON Web Signature (JWS) structure or as the plaintext of a JSON Web Encryption (JWE) structure, enabling the claims to be digitally signed or integrity protected with a Message Authentication Code (MAC) and/or encrypted.

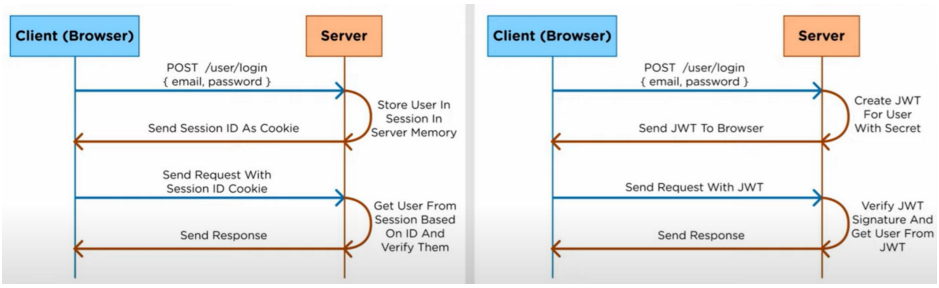
Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7519.

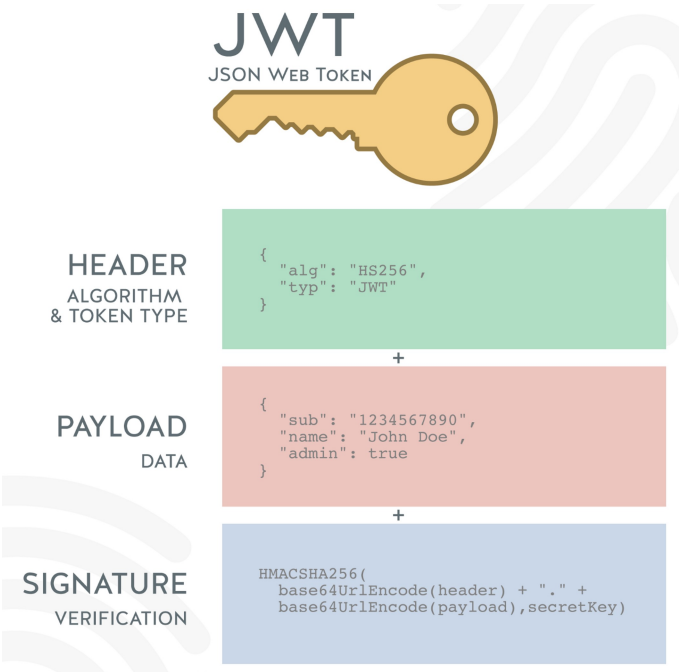
Token - JSON Web Token

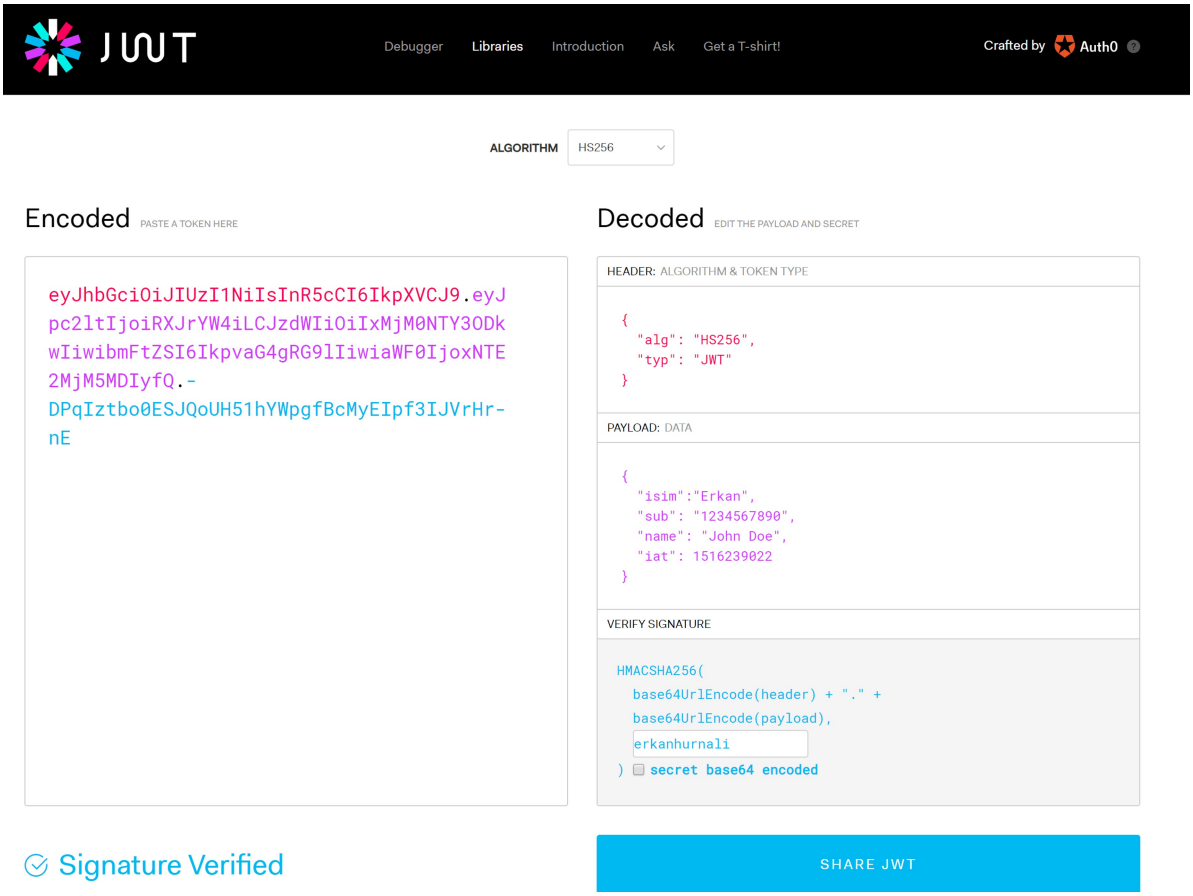


Avantajları Nelerdir ?

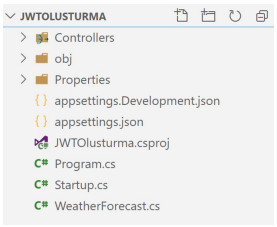
- JSON kullanması
- URL üzerinde taşınabilmesi
- Web çerezleri kullanma zorunluluğu olmaması
- Yöntem ve mantık basit olduğu için hızlı doğrulama yapılabilmesi
- Oluşturulan tokenlar(anahtarlar) ile HTTP request atılabilen her istemcide kullanılabilmesi
- Web uygulamaları açısından HTTP *session* gerekmemesi, *stateless* kullanıma uygun olması
- Veri bütünlüğünü sağlaması

JSON Web Token Yapısı





dotnet new webapi



dotnet add package Microsoft.AspNetCore.Authentication.JwtBearer

Startup.cs

```
29          public void ConfigureServices(IServiceCollection services)
30          {
31              services.AddControllers();
32              var key = Encoding.ASCII.GetBytes(Configuration["Secret"]);
33              services.AddAuthentication(x =>
34              {
35                  x.DefaultAuthenticateScheme = JwtBearerDefaults.AuthenticationScheme;
36                  x.DefaultChallengeScheme = JwtBearerDefaults.AuthenticationScheme;
37              })
38              .AddJwtBearer(x =>
39              {
40                  x.RequireHttpsMetadata = false;
41                  x.SaveToken = true;
42                  x.TokenValidationParameters = new TokenValidationParameters
43                  {
44                      ValidateIssuerSigningKey = true,
45                      IssuerSigningKey = new SymmetricSecurityKey(key),
46                      ValidateIssuer = false,
47                      ValidateAudience = false
48                  };
49              });
50          }
```

appsettings.json

```
1   {
2     "Logging": {
3       "LogLevel": {
4         "Default": "Information",
5         "Microsoft": "Warning",
6         "Microsoft.Hosting.Lifetime": "Information"
7       }
8     },
9     "AllowedHosts": "*",
10    "Secret":"buerkanhurnalinincokgizlianahtaridir"
11  }
```

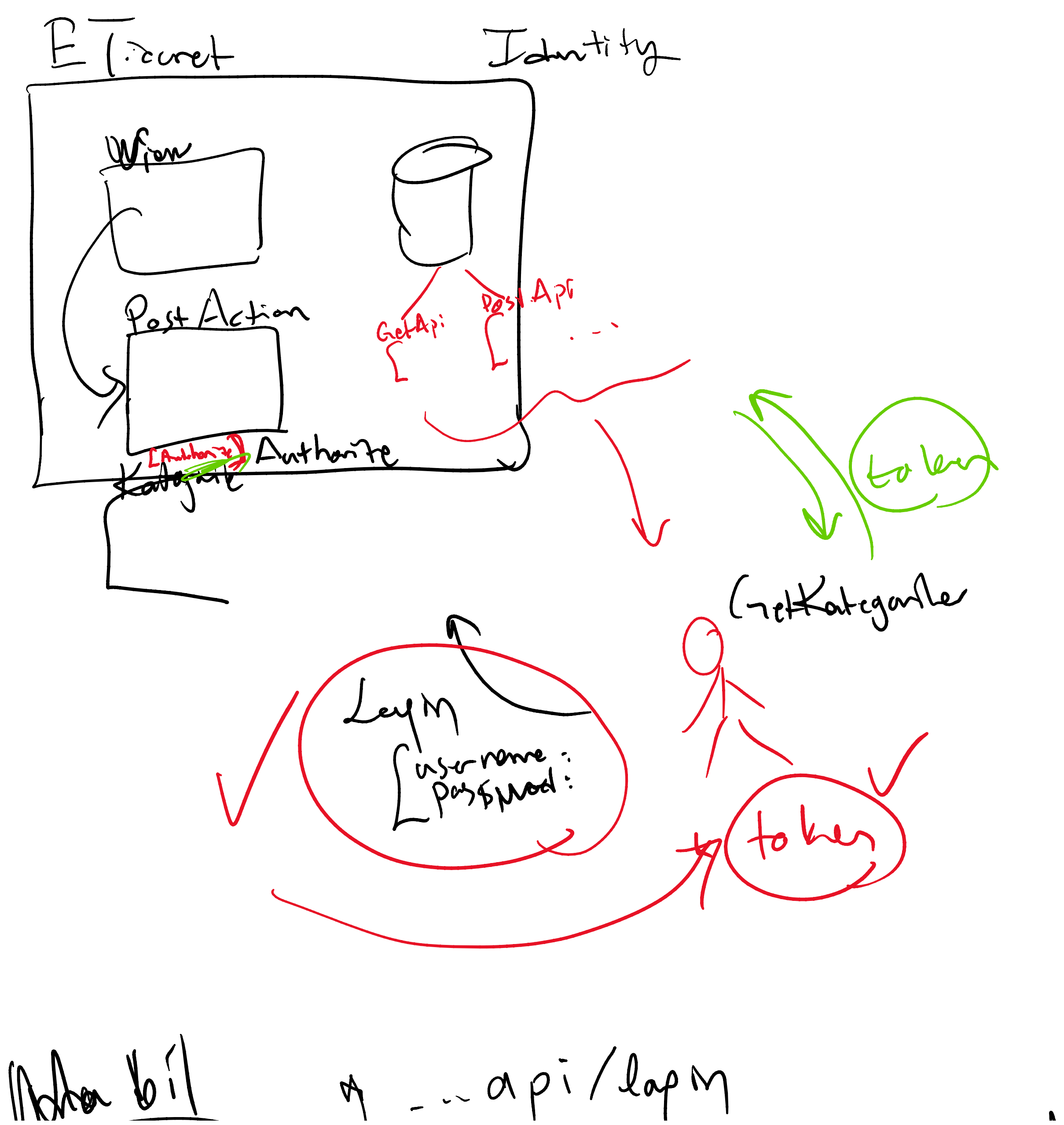
TokenController.cs

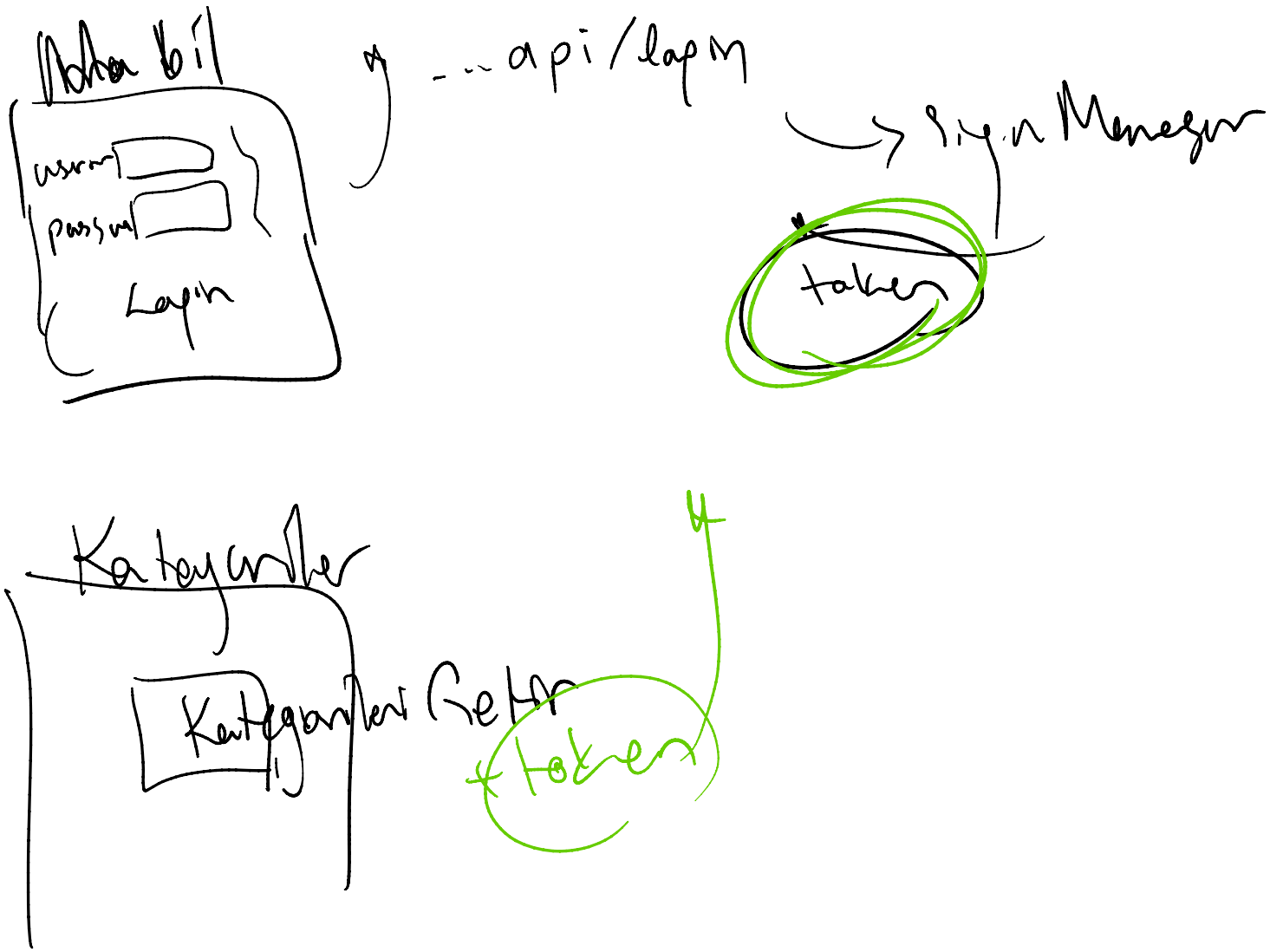
Controllers > TokenController.cs > { } JWTOlusturma.Controllers > JWTOlusturma.Controllers.TokenController > GetToken()

```csharp
0 references
public class TokenController : ControllerBase
{
    2 references
    private readonly IConfiguration configuration;

    0 references
    public TokenController(IConfiguration configuration)
    {
        this.configuration = configuration;
    }

    // GET api/token
    [HttpGet("")]
    0 references
    public ActionResult GetToken()
    {
        var tokenHandler = new JwtSecurityTokenHandler();
        var key = Encoding.ASCII.GetBytes(configuration["Secret"]);
        var tokenDescriptor = new SecurityTokenDescriptor
        {
            Subject = new ClaimsIdentity(new Claim[]
            {
                new Claim(ClaimTypes.Name, "Erkan")
            }),
            Expires = DateTime.UtcNow.AddDays(7),
            SigningCredentials = new SigningCredentials(new SymmetricSecurityKey(key),
                                        SecurityAlgorithms.HmacSha256Signature)
        };
        var token = tokenHandler.CreateToken(tokenDescriptor);
        var tokenString = tokenHandler.WriteToken(token);
        return Ok(tokenString);
    }
```




Kontrol
username
password
Login
...api/login
Sign Manager
token
Kategoriler
Kategorileri Getir
token