

# Öğretici: ASP.NET Core kullanmaya başlayın SignalR

21.11.2019 • Okumak için 11 dakika •  

## Bu makalede

[Önkoşullar](#)

[Web uygulaması projesi oluşturma](#)

[SignalRistemci kitaplığını ekleme](#)

[Merkez oluşturma SignalR](#)

[Yapılandırma SignalR](#)

[SignalRistemci kodu ekle](#)

[Uygulamayı çalıştırma](#)

Bu öğreticide, kullanarak gerçek zamanlı bir uygulama oluşturmanın temelleri öğretilir SignalR .  
Aşağıdakileri nasıl yapacağınızı öğrenirsiniz:

- ✓ Bir Web projesi oluşturun.
- ✓ SignalRistemci kitaplığını ekleyin.
- ✓ Bir SignalR hub oluşturun.
- ✓ Projeyi kullanmak üzere yapılandırın SignalR .
- ✓ Herhangi bir istemciden tüm bağlı istemcilere ileti gönderen kodu ekleyin.

Sonunda, çalışan bir sohbet uygulamanız olacaktır:



SignalRChat	SignalRChat
User: Hanna	User: Spencer
Message: Hi everyone!	Message: Got your message Hanna.
Send Message	Send Message
<ul style="list-style-type: none"><li>Hanna says Hi everyone!</li><li>Spencer says Got your message Hanna.</li></ul>	<ul style="list-style-type: none"><li>Hanna says Hi everyone!</li><li>Spencer says Got your message Hanna.</li></ul>
© 2019 - SignalRChat - Privacy	© 2019 - SignalRChat - Privacy

## Önkoşullar

Visual Studio

Visual Studio Code

Mac için Visual Studio

- ASP.net ve Web geliştirme iş yüküyle [Visual Studio 2019 16,4 veya üzeri](#)
- [.NET Core 3,1 SDK veya üzeri](#)

## Web uygulaması projesi oluşturma

Visual Studio

Visual Studio Code

Mac için Visual Studio

- Menüden **dosya > yeni proje'** yi seçin.
- Yeni proje oluştur** iletişim kutusunda **ASP.NET Core Web uygulaması'** nı seçin ve ardından **İleri'** yi seçin.
- Yeni projenizi yapılandırın** iletişim kutusunda, proje *SignalR sohbetini* adlandırın ve ardından **Oluştur'** u seçin.
- Yeni bir ASP.NET Core Web uygulaması oluştur** iletişim kutusunda **.net Core** ve **ASP.NET Core 3,1'** i seçin.
- Sayfaların kullanıldığı bir proje oluşturmak için **Web uygulaması Razor '** nı seçin ve ardından **Oluştur'** u seçin.

## Create a new ASP.NET Core web application

.NET Core ASP.NET Core 3.1



### Empty

An empty project template for creating an ASP.NET Core application. This template does not have any content in it.



### API

A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.



### Web Application

A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.



### Web Application (Model-View-Controller)

A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.



### Angular

A project template for creating an ASP.NET Core application with Angular



### React.js

A project template for creating an ASP.NET Core application with React.js

[Get additional project templates](#)

### Authentication

No Authentication

[Change](#)

### Advanced

☒ Configure for HTTPS

☐ Enable Docker Support

(Requires [Docker Desktop](#))

Linux

☐ Enable Razor runtime compilation

Author: Microsoft

Source: Templates 3.1.7

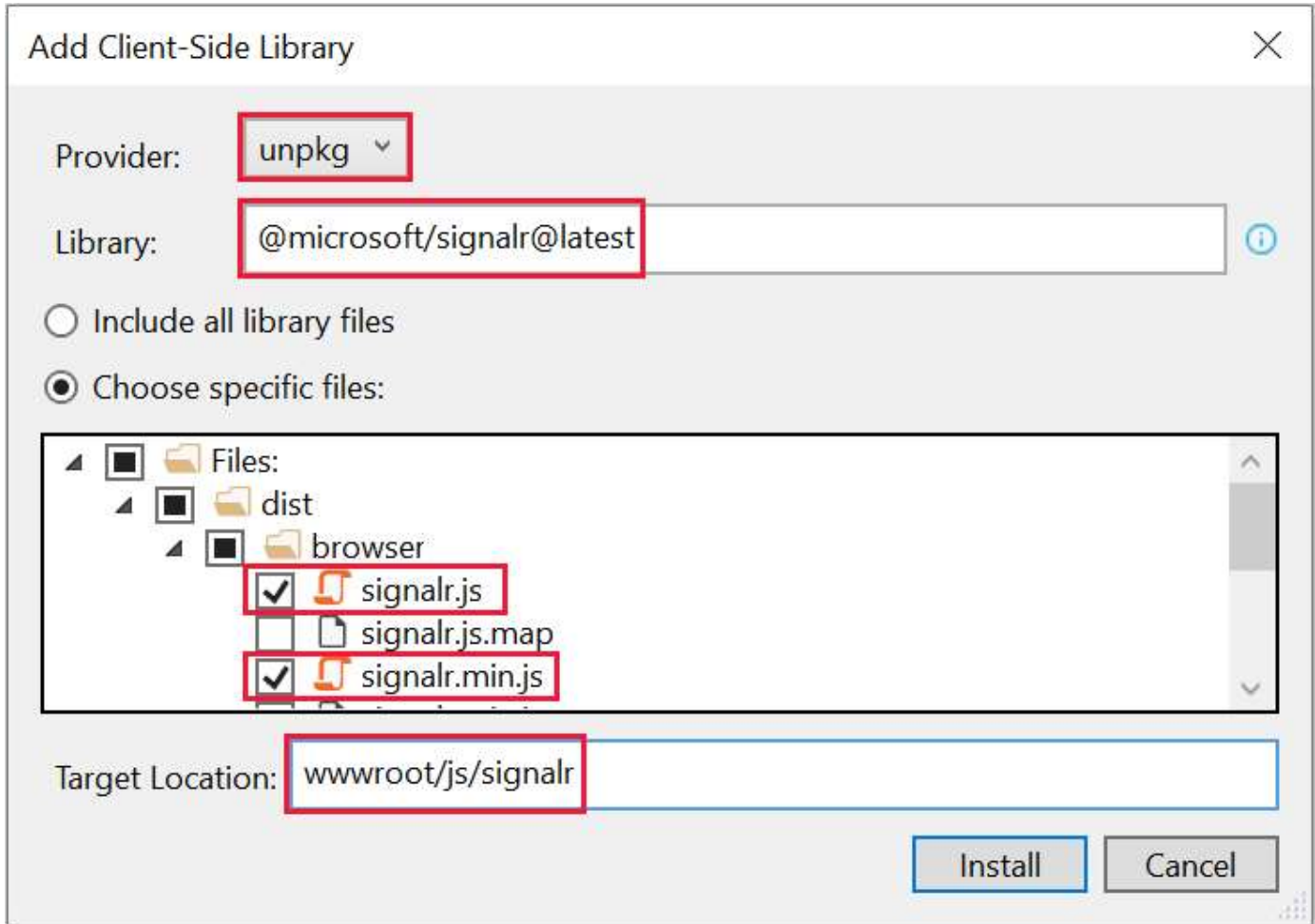
Back

Create

## SignalRistemci kitaplığını ekleme

SignalRSunucu kitaplığı ASP.NET Core 3,1 Paylaşılan çerçevesine dahildir. JavaScript istemci kitaplığı projeye otomatik olarak dahil değildir. Bu öğreticide, istemci kitaplığını *unpkg*'den almak için kitaplık Yöneticisi 'Ni (Libman) kullanacaksınız. unpkg, Node.js Paket Yöneticisi NPM 'de bulunan her şeyi teslim edebilen bir içerik teslim ağı (CDN).

- Çözüm Gezini' de projeye sağ tıklayın ve > **istemci tarafı kitaplığı Ekle** ' yi seçin.
- **Client-Side kitaplığı Ekle** iletişim kutusunda **sağlayıcı** için **unpkg** ögesini seçin.
- **Kitaplık için girin** `@microsoft/signalr@latest` .
- **Belirli dosyaları seç** ' i seçin, *dağ/Browser* klasörünü genişletin ve *signalr.js* ve *signalr.min.js* ' ı seçin.
- **Hedef konumu** *Wwwroot/js/SignalR/* olarak ayarlayın ve **yüklemeyi** seçin.



LibMan, bir *Wwwroot/js/SignalR* klasörü oluşturur ve seçilen dosyaları buna kopyalar.

## Merkez oluşturma SignalR

*Hub* , istemci-sunucu iletişimini işleyen yüksek düzeyli bir işlem hattı görevi gören bir sınıftır.

- SignalRSohbet projesi klasöründe bir *hub* klasörü oluşturun.

- *Hub* 'lar klasöründe, aşağıdaki kodla bir *ChatHub.cs* dosyası oluşturun:

C#

Kopyala

```
using Microsoft.AspNetCore.SignalR;
using System.Threading.Tasks;

namespace SignalRChat.Hubs
{
    public class ChatHub : Hub
    {
        public async Task SendMessage(string user, string message)
        {
            await Clients.All.SendAsync("ReceiveMessage", user, message);
        }
    }
}
```

*ChatHub* Sınıf sınıfından devralır *SignalR Hub* . *Hub* Sınıfı bağlantıları, grupları ve mesajlaşmayı yönetir.

*SendMessage* Yöntemi, tüm istemcilere ileti göndermek için bağlı bir istemci tarafından çağrılabilir.

Yöntemi çağırان JavaScript istemci kodu Öğreticinin ilerleyen kısımlarında gösterilmektedir. *SignalR* maksimum ölçeklenebilirlik sağlamak için kod zaman uyumsuzdur.

## Yapılandırma *SignalR*

*SignalR* Sunucunun istekleri geçirilecek şekilde yapılandırılması gerekir *SignalR* .

- Aşağıdaki Vurgulanan kodu *Startup.cs* dosyasına ekleyin.

C#

Kopyala

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Hosting;
using Microsoft.AspNetCore.HttpsPolicy;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;
using SignalRChat.Hubs;

namespace SignalRChat
{
    public class Startup
    {
        public Startup(IConfiguration configuration)
        {
            Configuration = configuration;
        }
    }
}
```

```

public IConfiguration Configuration { get; }

// This method gets called by the runtime. Use this method to add services
to the container.
public void ConfigureServices(IServiceCollection services)
{
    services.AddRazorPages();
    services.AddSignalR();
}

// This method gets called by the runtime. Use this method to configure the
HTTP request pipeline.
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
    else
    {
        app.UseExceptionHandler("/Error");
        // The default HSTS value is 30 days. You may want to change this
        for production scenarios, see https://aka.ms/aspnetcore-hsts.
        app.UseHsts();
    }

    app.UseHttpsRedirection();
    app.UseStaticFiles();

    app.UseRouting();

    app.UseAuthorization();

    app.UseEndpoints(endpoints =>
    {
        endpoints.MapRazorPages();
        endpoints.MapHub<ChatHub>("/chatHub");
    });
}
}
}

```

Bu değişiklikler SignalR ASP.NET Core bağımlılığı ekleme ve yönlendirme sistemlerine ekler.

## SignalRistemci kodu ekle

- *Pages\index.cshtml* içindeki içeriği şu kodla değiştirin:

CSHTML

Kopyala

```

@page
<div class="container">
    <div class="row">&nbsp;  </div>

```

```

<div class="row">
  <div class="col-2">User</div>
  <div class="col-4"><input type="text" id="userInput" /></div>
</div>
<div class="row">
  <div class="col-2">Message</div>
  <div class="col-4"><input type="text" id="messageInput" /></div>
</div>
<div class="row">&nbsp;</div>
<div class="row">
  <div class="col-6">
    <input type="button" id="sendButton" value="Send Message" />
  </div>
</div>
<div class="row">
  <div class="col-12">
    <hr />
  </div>
</div>
<div class="row">
  <div class="col-6">
    <ul id="messagesList"></ul>
  </div>
</div>
<script src="~/js/signalr/dist/browser/signalr.js"></script>
<script src="~/js/chat.js"></script>

```

Yukarıdaki kod:

- Ad ve ileti metni ve Gönder düğmesi için metin kutuları oluşturur.
- `id="messagesList"` Hub 'dan alınan iletileri görüntülemek için içeren bir liste oluşturur SignalR .
- , SignalR Bir sonraki adımda oluşturduğunuz *chat.js* uygulama koduna ve betik başvurularını içerir.
- *Wwwroot/js* klasöründe, aşağıdaki kodla bir *chat.js* dosyası oluşturun:

JavaScript

Kopyala

```

"use strict";

var connection = new signalR.HubConnectionBuilder().withUrl("/chatHub").build();

//Disable send button until connection is established
document.getElementById("sendButton").disabled = true;

connection.on("ReceiveMessage", function (user, message) {
  var msg = message.replace(/&/g, "&").replace(/</g, "<").replace(/>/g, ">");
  var encodedMsg = user + " says " + msg;
  var li = document.createElement("li");
  li.textContent = encodedMsg;
  document.getElementById("messagesList").appendChild(li);
});

```

```

    connection.start().then(function () {
        document.getElementById("sendButton").disabled = false;
    }).catch(function (err) {
        return console.error(err.toString());
    });

    document.getElementById("sendButton").addEventListener("click", function (event) {
        var user = document.getElementById("userInput").value;
        var message = document.getElementById("messageInput").value;
        connection.invoke("SendMessage", user, message).catch(function (err) {
            return console.error(err.toString());
        });
        event.preventDefault();
    });
});

```

Yukarıdaki kod:

- Bir bağlantı oluşturur ve başlatır.
- , Hub 'a ileti gönderen bir işleyiciye Gönder düğmesine ekler.
- , Hub 'dan iletileri alan ve bunları listeye ekleyen bir işleyici olan bağlantı nesnesine ekler.

## Uygulamayı çalıştırma

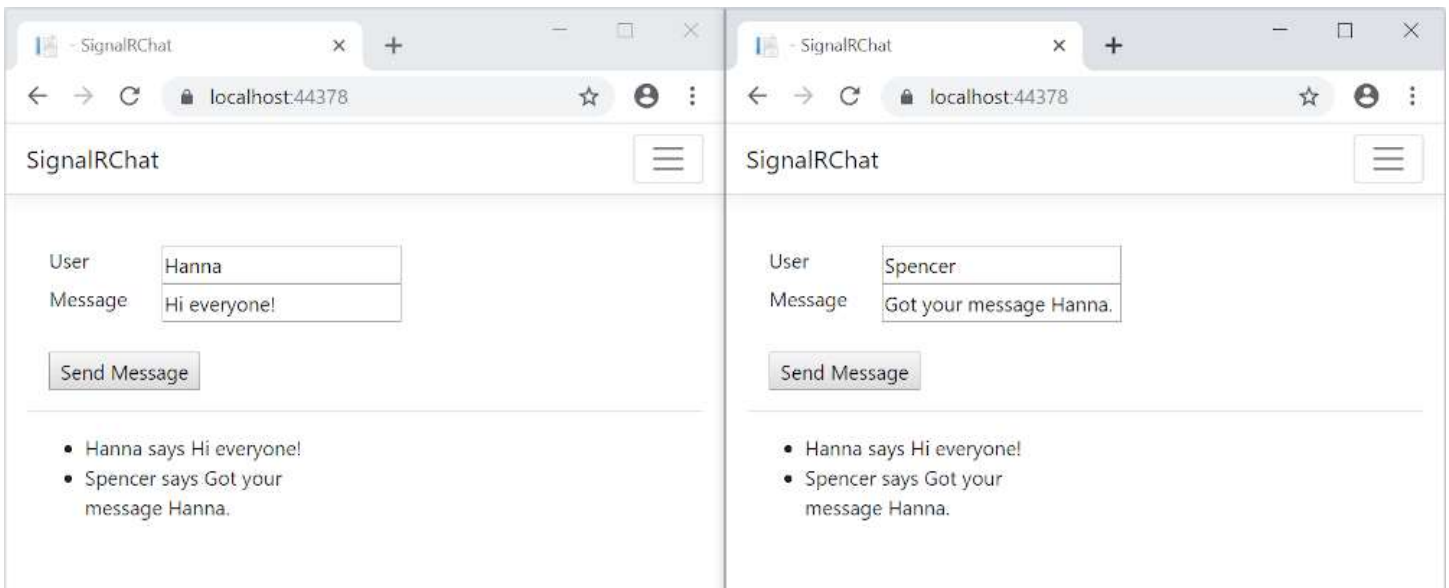
Visual Studio

Visual Studio Code

Mac için Visual Studio

- Uygulamayı hata ayıklamadan çalıştırmak için **CTRL + F5** tuşlarına basın.

- Adres çubuğundan URL 'yi kopyalayın, başka bir tarayıcı örneği veya sekme açın ve adres çubuğuna URL 'YI yapıştırın.
- Tarayıcı ' yı seçin, bir ad ve ileti girin ve **İleti gönder** düğmesini seçin. Ad ve ileti anında her iki sayfada da görüntülenir.

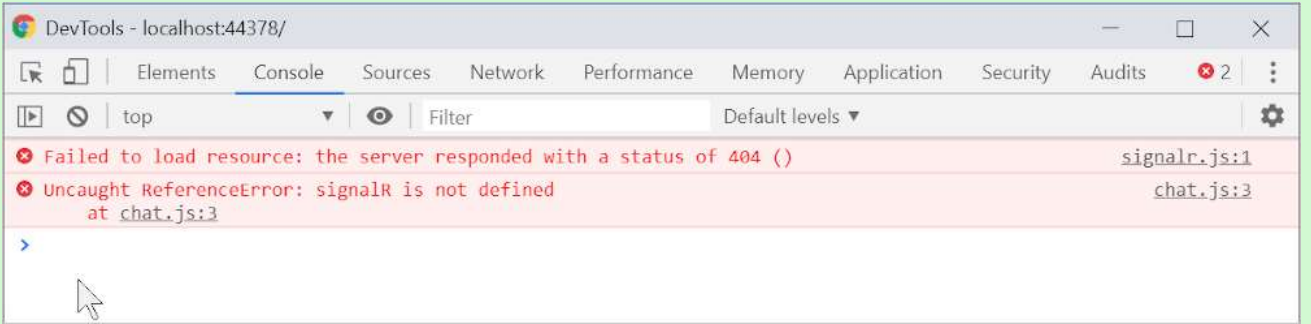




## İpucu

- Uygulama işe yaramazsa, tarayıcı geliştirici araçlarınızı (F12) açın ve konsola gidin. HTML ve JavaScript kodunuzla ilgili hatalarla karşılaşabilirsiniz. Örneğin, *signalr.js* yönlendirenden farklı bir klasöre yerleştirdiğinizi varsayalım. Bu durumda, bu dosyaya başvuru çalışmaz ve

konsolunda 404 hatası görürsünüz.



- Chrome 'da hata ERR\_SPDY\_INADEQUATE\_TRANSPORT\_SECURITY alırsanız, geliştirme sertifikanızı güncelleştirmek için şu komutları çalıştırın:

.NET CLI

Kopyala

```
dotnet dev-certs https --clean  
dotnet dev-certs https --trust
```

## Bu sayfayı yararlı buluyor musunuz?

👍 Yes    👎 No