

ASP.NET Core API och Razor Pages

A.Bryngelsson fed19g

Inledning:

Active Server Page .NET Core (ASP.NET Core) är Microsofts ramverk för webbapplikationer och backend.

Backend koden skrivs i C# (C Sharp) som är ett Java liknande objektorienterat programmeringsspråk (OOP).

C# och Java det mest populära backend programmeringsspråk och används ofta av större företag.

För att visa resultaten av backend koden i en webbläsare kan Microsofts version av html användas som kallas cshtml.

Javascript ramverk kan också användas som React eller Angular för att visa information från API som skapats med ASP.NET ramverket.

Bakgrund:

Microsofts tidigare .NET Framework bestod av sluten källkod och kunde bara användas på Window operativsystem, men den nya .NET Core är plattformsoberoende och kan användas på Linux, Mac, Windows (Ibanez, 2004), källkoden är öppen och finns på GitHub.

C# är det största backend språket efter Java enligt Webbjobb undersökningen (Webbjobb, 2020), och nya uppdaterade versioner av .NET Core släpps ofta.

Servers som använder ASP.NET Core är snabba och kan leverera fler webb http förfrågningar per sekund än liknande Java Servlet eller Node.js servers enligt Ben Adams, från 2x till 8x förbättring jämfört med konkurrenterna (Adams 2019).

Razor Pages cshtml är Microsofts försök att minska komplexiteten av de vanligare Model View Controller (MVC) apparna, genom att kombinera mer av kontroll koden och det som visas på sidan (Watson 2017).

Användning:

Cshtml är en blandning av C# och html som omvandlas till läsbara filer på servern för standard webbläsare då dessa inte kan använda C# kod.

```
<ul class="list-group">
  @for (int i = 0; i < Model.ListOfUsers.Count; i++)
  {
    <li class="list-group-item">
      Name: @Model.ListOfUsers[i].Name <br>
      UserId: @Model.ListOfUsers[i].UserId <br>
      Last seen:<br>@Model.ListOfUsers[i].LastLogin
    </li>
  }
</ul>
```

(kod 1 cshtml: För att blanda C# koden med html används '@' framför kod delen och sedan kan vanlig html användas. Från /Pages/Rpage.cshtml)

```
//This is return when you visit the api page.
[HttpGet]
public string Get(){
    string userListString = JsonConvert.SerializeObject(
userList.getListOfUsers());
    return userListString;
}
```

(kod 2 API lämnar en JSON string från userlist object array) (bilden kortade rad 4. Från /Controllers/UserAPIController.cs)

```
services.AddTransient<UserList>();
//Loads in UserList once for all to use.
services.AddRazorPages(); //for razer pages cshtml.
```

(kod 3 UserList skickas till API och Razor pages via service. Från /Startup.cs)

Diskussion:

För att API och Razor Pages ska få tillgång till andra objekt bör dessa först laddas in med Startup.cs Configuration.

Det kan ses som ett extra onödigt steg för mindre projekt, men kan komma att bli mer användningsbart vid större projekt då man kan lägga till loggning och testning på samma plats.

Framtid:

Blazor är Microsofts nästa steg från server till webbläsare. Med Blazor kan man bygga webbapplikationer som kan köras i webbläsaren via WebAssembly istället för javascript. Blazor WebAssembly är för närvarande i preview, och det kan ta ett par år innan majoriteten av webbläsare stöder WebAssembly.

Ibanez, L. 2004

Microsoft gets on board with open source

<https://opensource.com/business/14/11/microsoft-dot-net-empower-open-source-communities>

Adams, B. 2019

ASP.NET Core: Saturating 10GbE at 7+ million requests/s

<https://www.ageofascent.com/2019/02/04/asp-net-core-saturating-10gbe-at-7-million-requests-per-second/>

Watson, M. 2017

ASP.NET Razor Pages vs MVC: How Do Razor Pages Fit in Your Toolbox.

<https://stackify.com/asp-net-razor-pages-vs-mvc/>

Webbjobb, 2020 (Databas för ranking av populära programmeringsspråk)

<https://webbjobb.io/programmeringssprak>

Angular API

Del 2. A.Bryngelsson

Inledning och bakgrund:

Angular är ett Frontend javascript ramverk för webbläsaren liknande andra ramverk som ReactJS och VueJS,

Ett par Google ingenjörer skapade första versionen av AngularJS 2010 som efter en stor uppdatering 2016 bara blev Angular efter version 2 (Gavigan 2018).

Efter att Google 2017 sade sig stå bakom Angular under en lång tid framöver kan vissa utvecklare känna att Angular är mer pålitligt, då ett stort företag uppdaterar och utvecklar ramverket enligt Liliia Harkushko (Harkushko 2020).

Användning:

Eftersom API server och Angular appen kommer från samma server har de också samma URL, som kallas BASE_URL och skickas in från src/main.ts.

Efter URL hittats, läggs userapi till för att nå JSON API.

JSON userlist består av Name, UserId, LastLogin.

```
constructor(http: HttpClient, @Inject('BASE_URL') baseUrl:
string) {
  http.get<User[]>(baseUrl + 'userapi').subscribe(result => {
    this.UserList = result;
    this.UserList.sort((a, b) => a.UserId - b.UserId);
  }, error => console.error(error));
```

(kod 1 Get API. Från /ClientApp/src/app/get-userlist/get-userlist.component.ts)

```

<h2 *ngIf="!UserList">Nothing yet...</h2>
<ul class="list-group" *ngIf="UserList">
  <li class="list-group-item" *ngFor="let User of UserList">
    Name: {{ User.Name }} <br>
    UserId: {{ User.UserId }} <br>
    Last seen: <br> {{ User.LastLogin.slice(0,19) }}
  </li>

```

(kod 2. Angular For loop med html. Från
/ClientApp/src/app/get-userlist/get-userlist.component.html)

```

{ path: 'get-userlist', component: GetUserListComponent },

```

(kod 3. Lägg till Path i app.module så den nya componenten kan nås. Från
/ClientApp/src/app/app.module.ts)

Diskussion:

När författaren försökte använda ng-href för att skapa länkar dynamisk istället för href, fungerade ej koden och det fanns väldigt få frågor/svar på nätet som kunde hjälpa, medans ReactJs ser ut att ha fler användare som skriver frågor/svar som kan vara oss noviser ytterst behjälpliga.

Gavigan, D. 2018

The History of Angular

<https://medium.com/the-startup-lab-blog/the-history-of-angular-3e36f7e828c7>

Harkushko, L. 2020

Angular: Best Use Cases and Reasons To Opt For This Tool

<https://yalantis.com/blog/when-to-use-angular/>