Middleware

**1. Mis on Middleware?**

Middleware on tarkvara, mis võimaldab hajutatud võrgus kahe või enama rakenduse või rakenduskomponendi vahel ühte või mitut tüüpi suhtlust või ühenduvust. Lihtsustades rakenduste ühendamist, mis ei ole loodud üksteisega ühenduse loomiseks, ning pakkudes funktsioone nende arukaks ühendamiseks, lihtsustab rakenduste arendamist ja kiirendab turule jõudmise aega.

**2. Missuguseid extensione meetodeid kasutab?**

Kolm meetodit: Run, Map ja Use

Run - lisab rakenduse päringute konveierile terminali middleware delegaadi ehk pipeline’i.

Map - Branches the request pipeline based on matches of the given request path. If the request path starts with the given path, the branch is executed.

Use - Lisab rakenduse päringute kontuurile reas määratletud middleware ehk delegaadi.

**3. Iseloomusta Middleware käskude loogikat.**

Iga delegaat saab teha toiminguid enne ja pärast järgmist delegaati.Erandite käsitsemise delegaadid tuleks välja kutsuda varakult, et nad saaksid tabada erandeid, mis esinevad konveieri hilisemates etappides.

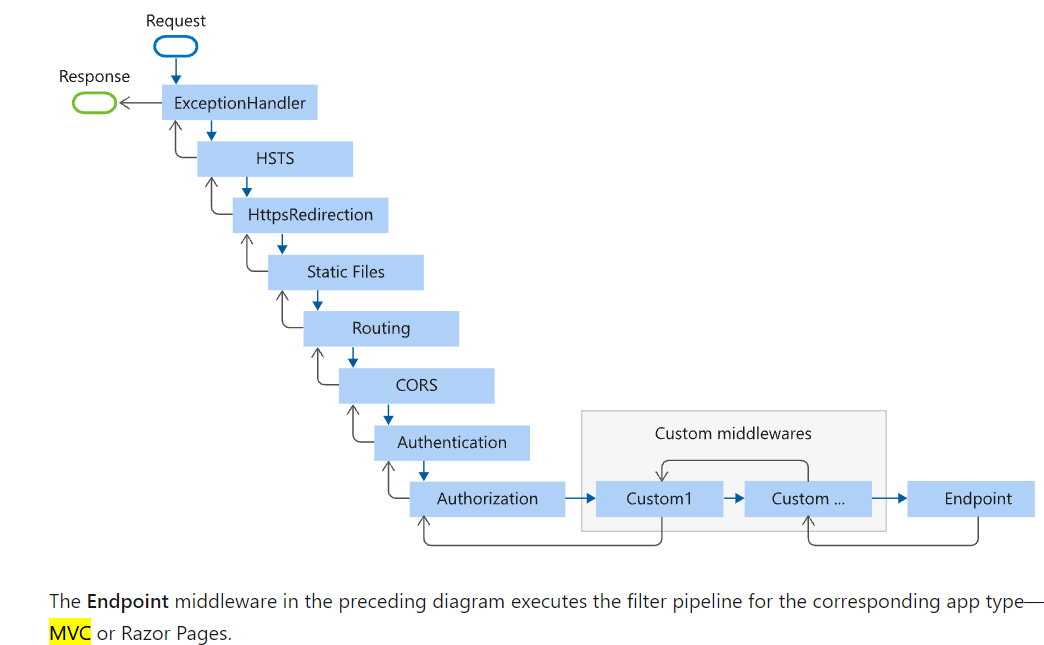
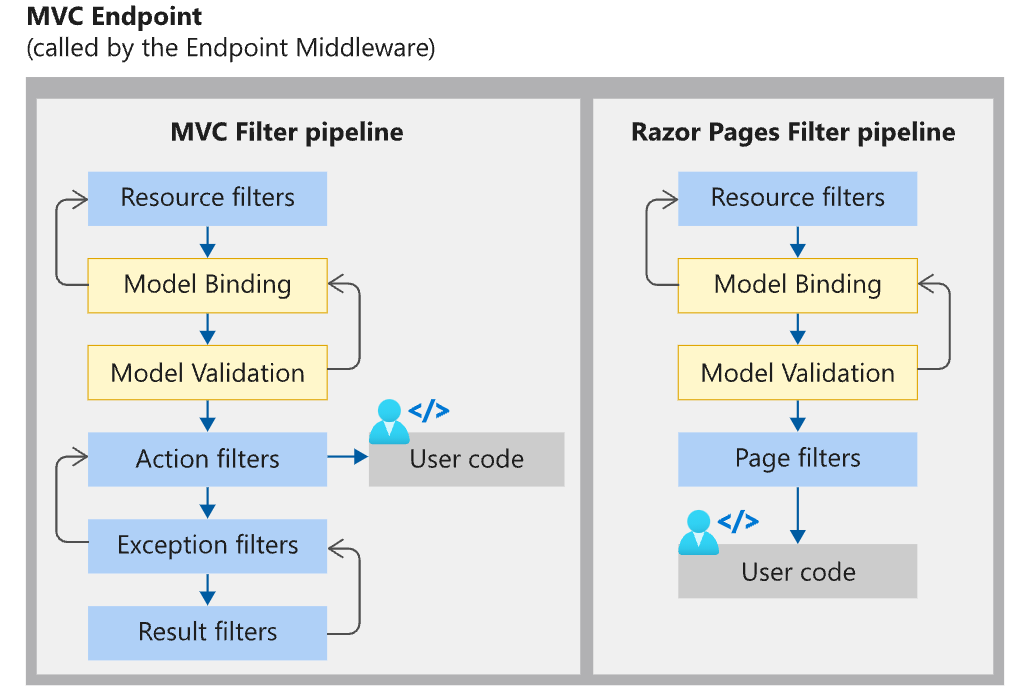
Lihtsaim võimalik rakendus ASP.NET Core seadistab ühe taotluste delegaadi, kes tegeleb kõigi päringutega. See juhtum ei hõlma tegelikku päringute kontuuri. Selle asemel kutsutakse vastuseks igale HTTP -päringule üks anonüümne funktsioon.

**4. Kuidas kutsutakse delegaati, kui see ei edasta päringut (request) järgmisele delegaadile?**

Kui middleware teeb katkestuse, nimetatakse seda **terminal middleware**, kuna see takistab edasisel middleware taotluse töötlemisel liikuda järgmisele.

**5. Kirjelda täieliku päringuprotsessi järjestust ASP.NET Core MVC näite puhul.**

Järgmine diagramm näitab rakenduste ASP.NET Core MVC ja Razor Pages täielikku päringutöötlusvoogu. Näete, kuidas tavalises rakenduses olemasolevad vahepealsed tooted tellitakse ja kuhu lisatakse kohandatud middleware. Teil on täielik kontroll olemasoleva middleware ümberkorraldamise või uute kohandatud middleware käivitamise üle, kui see on teie stsenaariumide jaoks vajalik.

**6. Kuidas mõjutab middleware koodi järjestus komponeneti tööd?**

Middleware komponentide järjekord määratakse käivitamisel. Konfigureerimismeetod määrab, millises järjekorras middleware komponente päringutel esile kutsutakse, ja vastupidine järjekord määratakse andmete tagasisaatmise puhul. Järjekord on turvalisuse, jõudluse ja funktsionaalsuse seisukohalt kriitilise tähtsusega.

Seadistamise meetod lisab turvalisusega seotud middleware komponendid tavapärases soovitatud järjekorras:  
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)

{

if (env.IsDevelopment())

{

app.UseDeveloperExceptionPage();

app.UseDatabaseErrorPage();

}

else

{

app.UseExceptionHandler("/Error");

app.UseHsts();

}

app.UseHttpsRedirection();

app.UseStaticFiles();

// app.UseCookiePolicy();

app.UseRouting();

// app.UseRequestLocalization();

// app.UseCors();

app.UseAuthentication();

app.UseAuthorization();

// app.UseSession();

// app.UseResponseCompression();

// app.UseResponseCaching();

app.UseEndpoints(endpoints =>

{

endpoints.MapRazorPages();

endpoints.MapControllerRoute(

name: "default",

pattern: "{controller=Home}/{action=Index}/{id?}");

});

}

**7. Mida teeb Developer Exception Page Middleware, UseExceptionHandler, UseHsts, UseHttpsRedirection, UseStaticFile, UseCookiePolicy, UseRouting, UseAuthentication, UseAuthorization, UseSession ja UseEndpoints koos MapRazorPages, UseSpaStaticFiles?**

*Developer Exception Page Middleware* kuvab üksikasjalikku teavet töötlemata päringuerandite kohta.  
*UseExceptionHandler* Lisab pipeline middleware, mis tabab erandid, logib need ja täidab taotluse alternatiivsel konveieril uuesti. Taotlust ei täideta uuesti, kui vastus on juba alanud.

*UseHsts* Lisab HSTS-i kasutamiseks middleware, mis lisab päise Strict-Transport-Security.

*UseHttpsRedirection* Lisab vahevara HTTP -päringute HTTPS -i ümbersuunamiseks.

*UseStaticFile* Staatilised failid, nagu HTML, CSS, pildid ja JavaScript, on vara, mida ASP.NET Core rakendus vaikimisi otse klientidele teenindab.

*UseCookiePolicy* lisab CookiePolicyMiddleware käitleja määratud IApplicationBuilderile, mis võimaldab küpsisepoliitika võimalusi.  
*UseRouting* lisab middleware pipeline marsruudi sobitamise. See middleware vaatab rakenduses määratletud lõpp -punktide kogumit ja valib päringu põhjal parima vaste. *UseAuthentication* lisab AuthenticationMiddleware määratud IApplicationBuilderile, mis võimaldab autentimisvõimalusi. *UseAuthorization* lisab määratud IApplicationBuilderile AuthorizationMiddleware, mis võimaldab autoriseerimise võimalusi. *UseSession* seansi olek on ASP.NET põhistsenaarium kasutajaandmete salvestamiseks, kui kasutaja veebirakendust sirvib. Seansi olek kasutab kliendi taotluste andmete säilitamiseks rakenduse hallatavat ruumi. *UseEndPoints* lisab lõppvara täitmise middleware konveierile. See käivitab valitud lõpp-punktiga seotud delegaadi. *UseSpaStaticFiles* seadistab rakenduse esitama üheleheküljelise rakenduse (SPA) staatilisi faile. Failide leidmiseks kasutatakse registreeritud teenust ISpaStaticFileProvider.

**8. Valige välja viis Built-in Middleware ja iseloomustage neid**.

*Application-level middleware*

Selline middleware kehtib kõigi rakendusele saabuvate taotluste kohta, st autentimine ja volitamine või lisab igale päringule algusaja. Enne mis tahes marsruudi kasutamist kasutage kindlasti sellist middleware.  
*Router-level middleware*

Ruuteritaseme middleware töötab samamoodi nagu rakenduse tase, välja arvatud see, et see on rakendatav ainult marsruutidele, kus me selle vahevara läbime.  
*Error-handling middleware*

Vigade käsitlemisel vahevara korral on meil üks täiendav argument, seejärel rakenduse ja ruuteri taseme vahevara (st req, res, next), mis on viga.

*Built-in middleware*

Expressil on järgmised sisseehitatud middlewarefunktsioonid:

* express.static teenindab staatilisi varasid, näiteks HTML -faile, pilte jne.
* express.json analüüsib sissetulevaid päringuid JSONi koormustega. MÄRKUS. Saadaval Express 4.16.0+ versiooniga
* express.urlencoded parsib sissetulevaid päringuid URL-i kodeeritud kasulike koormustega. MÄRKUS. Saadaval Express 4.16.0+ versiooniga

*Third-party middleware*

On mitmeid kolmanda osapoole vahevara, näiteks body-parser, mongoose, morgan jne. Neid saab installida käsuga - npm install <mooduli nimi>