**Middleware**

1. Mis on Middleware?

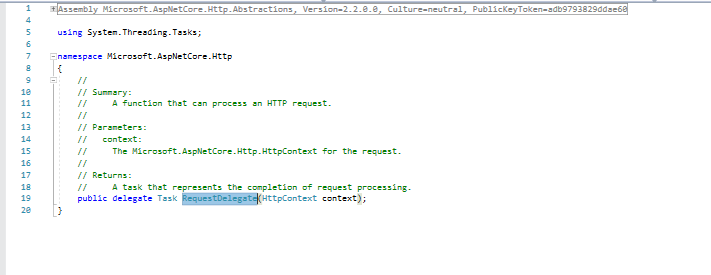
Middleware on tarkvara programm, mis võimaldab ühendada kahte või enamat erinevat rakendust läbi ühise võrgu. Lihtsustab protsessi tarkvara arendajatele, olles keskmine liide siis näiteks Front End ja Back End arendusel.

1. Missuguseid extensione meetodeid kasutab?

Build function, Invoke meetodina, configure meetod, run extension meetod ehk Terminal middleware, Map extension meetod, Use meetod

1. Iseloomusta Middleware käskude loogikat.

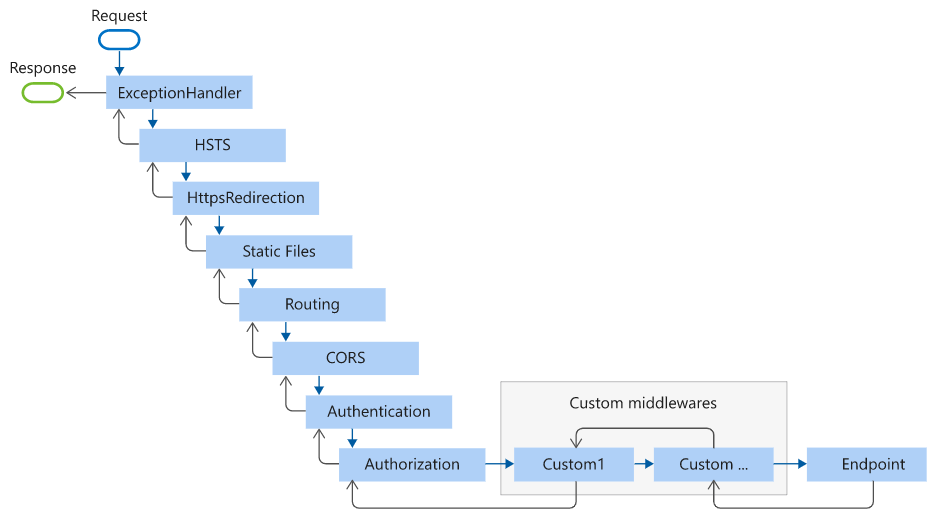
Konfigureerimis protsessiga seatakse taotluse delegaadid mille läbi Middleware hakkab päring läbi otseühenduse küsima teiselt rakenduselt, rakenduse käivitamisel.



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

Terminal Middleware

1. Kirjelda täieliku päringuprotsessi järjestust ASP.NET Core MVC näite puhul.



Luuakse delegaadid mille läbi päringut hakatakse looma. Iga delegaat saab töötada enne ja pärast teist delegaati. Erand delegaadid tuleks kohe algusesse kehtestada. Kui delegaat ei valideeri päringut ega saada seda järgmisele delegaadile siis seda kutsutakse Terminal Middleware. Run delegaadid ei saa järgmist parameetrit, (ei saada järgmisele delegaadiel) ja on alati terminal middleware, mis lõpetab “pipelinei”. Informatsioon esitatakse päringu esitajale kui middleware delegaadid on kõik informatsiooni läbi käinud. Delegaatide järjestust on võimalik muuta aga mõned delegaadid ei pruugi töötada. Optimaalne järjestus inbuilt delegaatidega:

Exception error handling raporteerib kui on rakenduse töötamise erroreid, püüab kinni erandid mis on kehtestatud.

HTTP päring suunatakse üle HSTS Strict Transport Security protokolliks

Staatilised failid tagastatakse, lõpetatakse edasine päringu protsess

Routing Middleware ruutib edasi päringu

Autentimis Middleware valideerib kasutaja enne kui jagab ligipääsu turvatud informatsioonile

Sessiooni Middleware loob ja hoiab sessiooni seisu või olukorda,

Endpoint Routing Middleware (UseEndpoints, MapRazorPages) lisab Razorr lehtede endpointi päritud pipelineilile.

1. Kuidas mõjutab middleware koodi järjestus komponeneti tööd?

Middleware süsteemi ülesehitus ja seadistus on väga oluline optimaalselt töötamiseks eriti kui on kasutusel palju mitmesuguseid seadmeid ja andmed tuleb edastada erinevatele osapooltele. Kui on kood õigesti ülesehitatud on võimalik analüüsida andmeid, komponentide tööd, ja troubleshootida kui vaja.

1. Mida teeb

Developer Exception - Raporteerib rakenduse töötamise vea

Page Middleware – Raporteerib andmebaasi töötamise vea

UseExceptionHandler – Arvestab erandid mis middlewarei sisestatud

UseHsts – lisab Strict Transpord Security päise Middlewarei

UseHttpsRedirection – suunab HTTPS päringu HTTPS protokolli

UseStaticFile – väljastab staatilise faili ja lõpetab päringu edasise protsessimise

UseCookiePolicy - kontrollib et rakendus vastaks EU General Data Protction Regulations regulatsioonidele

UseRouting – Ruutib päringud edasi

UseAuthentication – proovib autentida kasutajat enne kui faili juurdepääs antakse

UseAuthorization – autoriseerib kasutajale failide ligipääsu

UseSession – loob ja hoiab sessiooni seisu

UseEndpoints koos MapRazorPages – Lisab Razor lehtede Endpointi päritud Pipelinei

UseSpaStaticFiles - Seadistab rakenduse esitama üheleheküljelise rakenduse (SPA) staatilisi faile.

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

1. Cookie Policy – Jälgib kasutaja õigusi kasutaja personaalse informatsiooni salvestamiseks, ja kinnitab miinimum standardid küpsiste väljadele nagu secure ja Samesite

2. Health Check – kontrollib ASP.Net Core rakenduse staatus ja selle sõltuvust ning kontrollib andmebaasi saadavust

3. HTTP Method Override – Lubab sisse tuleva POST päringul üle kijutada meetodi

4. HTTP Strict Transport Security (HSTS) – lisa turvalisuse meede middlewareile lisades spetsiaalse vastava päise.

5. Static Files – pakub abi staatiliste failide ja kaustade sirvimisel