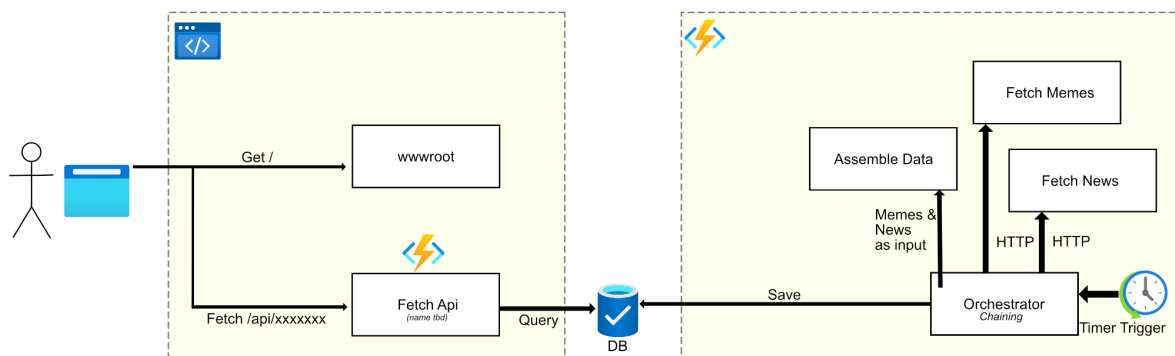


Projekt Plan

Alexander Tranehed, Alexander Dahlen, Felix Emt
Siktar på VG

Världen är deprimerande; krig i vår närhet, ökad inflation, Tottenham förlorar 2-0 mot United, antisemiter i riksdagen.. För att lätta upp stämningen lite så aggregerar vi nyheter men vi byter ut bilderna som artiklarna kommer till memes. På så sätt kan man hålla sig uppdaterad men samtidigt få underhållning.

En SWA frontend, en Fetch API som hämtar från en Table Databas. En chaining durable function som triggas på en timer. Den hämtar data från en nyhetskälla, preliminärt NYTimes, samt data från en memekälla, preliminärt imgflip. Outputsen från dessa två fetches används som input till en assemble data function. Denna funktion sätter ihop datan så som vi vill spara den i DB. Orkestratorn sparar sedan ned output från assemble data funktionen till databasen.



Databas

Sikta på att köra cosmos db, men ha table databas som backup ifall det inte funkar.

Frontend

Personen som blir ansvarig för frontend kan välja själv om de vill jobba med Blazor/React/Vanilla eller något annat.

Tidsplan

V43 - barebones, så att allt funkar.

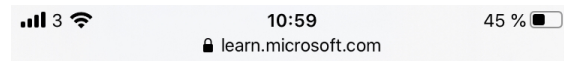
V44 : slipa, deploya, roligare databas lösning. Lösa buggar. Fixa så saker ser snyggt ut

v45 : Måndag - Onsdag Planera redovisning.

V45: Torsdag 10: Redovisning

Frågor

- Queues eller inte mellan alla steg i orkestrator?



Pattern #1: Function chaining

In the function chaining pattern, a sequence of functions executes in a specific order. In this pattern, the output of one function is applied to the input of another function. The use of queues between each function ensures that the system stays durable and scalable, even though there is a flow of control from one function to the next.



<https://learn.microsoft.com/en-us/azure/azure-functions/durable/durable-functions-sequence?tabs=csharp>