Micsa Ciprian
Anul IV, IS
semigrupa 4.1

TEMA 2
Background Jobs

Ce sunt background jobs?

Background Jobs sunt procese non-iterative, care se ruleaza in background, in spatele operatiunilor interactionarii normale. Ele pot rula in paralel si nu incurca procesele interactive. Procesele interactive sunt procese asa numite foreground. Aceste joburi pot fi executate fara a fi necesara interactiunea cu utilizatorul. Pe scurt, joburile pot fi lansate la pornirea programului si executarea acestora se desfasoara in background

Joburile background pot fi initiate in mai multe moduri. Acestea se incadreaza in una din urmatoarele categorii:
- Triggere declansate de eveniment
- Triggere planificate

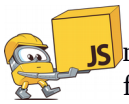
Sunt mai multe tipuri de joburi background:
- fluxuri de lucru pe termen lung(furnizarea de servicii si sisteme)
- prelucrarea datelor sensitive(transferarea taskurilor si procesarea lor intr-un loc mai sigur)
- CPU intensive(se bazeaza pe calcule matematice, analiza modelelor structurale)
- I/O intensive(input-output → stocarea si indexarea unui numar mare de fisiere)
- joburi batch(update-uri si procesarea planificata din timp a unor operatii)

Joburile background sunt executate asincron. Ele pot fi executate chiar si dintr-o locatie separata, din interfata de utilizare sau din procesul de unde este invocat task-ul in fundal. Procesul de apelare nu asteapta incheierea sarcinilor, acesta neputand detecta automat cand se incheie sarcina de lucru. In mod iteal, sarcinile de fundal sunt operatii care se pornesc in fundal, iar progresul executiei acestora nu impacteaza interfata utilizatorului (UI), sau a procesului de apelare.

Daca e nevoie ca un background job sa indice progresul sau finalizarea sa, trebuie implementat un mecanism pentru acesta.
Cateva exemple de mecanisme:
- Adaugarea unui indicator de status intr-o variabila care poate fi accesata de UI sau de un task foreground
- Stabilirea unei cozi de raspuns pe care o asculta UI sau un task(ReplyTo, CorrelationId din Azure Service Bus)
- Expunerea unui API sau a unui endpoint din taskul background asupra caruia UI are acces.

Hosting-ul joburilor backround se poate efectua utilizand serviciile platformei **Azure**:

- Azure Web Apps, WebJobs
- Azure Virtual Machines
- Azure Batch
- Azure Kubernetes Service(AKS)

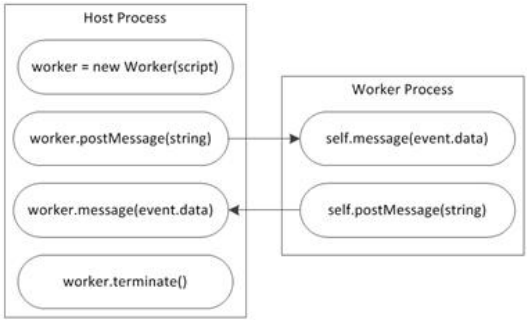


Un job backround poate fi, ca exemplu, un **web worker**. Un web worker este un JavaScript care rulează în fundal, independent de alte scripturi, fără a afecta performanța paginii. Puteți continua să faceți tot ce doriți: clic, selectarea obiectelor etc, în timp ce web worker-ul rulează în background.

Exemplu de web worker care numara in background:

```
<!DOCTYPE html>
<html>
<body>

<p>Count numbers: <output id="result"></output></p>
<button onclick="startWorker()">Start Worker</button>
<button onclick="stopWorker()">Stop Worker</button>

<script>
var w;

function startWorker() {
    if(typeof(Worker) !== "undefined") {
        if(typeof(w) == "undefined") {
            w = new Worker("demo_workers.js");
        }
        w.onmessage = function(event) {
            document.getElementById("result").innerHTML = event.data;
        };
    } else {
        document.getElementById("result").innerHTML = "Sorry, your browser does not support Web
Workers...";
    }
}

function stopWorker() {
    w.terminate();
    w = undefined;
}
</script>

</body>
</html>
```

API al unui web worker consta din urmatoarele concepte:
- Initializarea workerului (utilizand clasa Worker(url))
- Executia web workerului (utilizand metoda postMessage(data))
- Intershimbarea mesajelor intre worker si thread-ul principal
- Inchiderea workerului (utilizand metoda terminate())



Bibliografie:
https://www.w3schools.com/html/html5_webworkers.asp
https://www.html5rocks.com/en/tutorials/workers/basics/
https://docs.microsoft.com/EN-US/AZURE/ARCHITECTURE/BEST-PRACTICES/BACKGROUND-JOBS
https://www.sitepoint.com/how-to-schedule-background-tasks-in-javascript/