Chirpii Alexei
AC IS Anul IV
semigrupa 1.2

TEMA 2
Background Jobs

Background jobs - procesele non-interactive care se desfășoară în spatele operațiunilor interactionarii normale. Ele rulează în paralel și nu deranjează procesele și operațiunile interactive (procesele foreground). Aceste joburi pot fi executate fără necesitatea de a interacționa cu utilizatorul, adica pot fi lansate la pornirea programului si se executa in background.

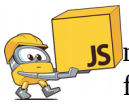
Sunt mai multe tipuri de joburi backround:
- CPU intensive(calcule matematice, analiza modelelor structurale)
- I/O intensive(stocarea si indexarea unui numar mare de fisiere)
- joburi batch(update-uri si procesarea planificata din timp a unor operatii)
- fluxuri de lucru pe termen lung(furnizarea de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

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

Joburile background se execută asincron într-un proces separat, sau chiar într-o locație separată, din interfața de utilizare sau procesul care a invocat taskul in fundal. În mod ideal, sarcinile de fundal sunt operații care se pornesc si se uita de ele, iar progresul execuției acestora nu are nici un impact asupra UI sau a procesului de apelare. Aceasta înseamnă că procesul de apelare nu așteaptă încheierea sarcinilor. Prin urmare, nu poate detecta automat când se încheie sarcina.
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)

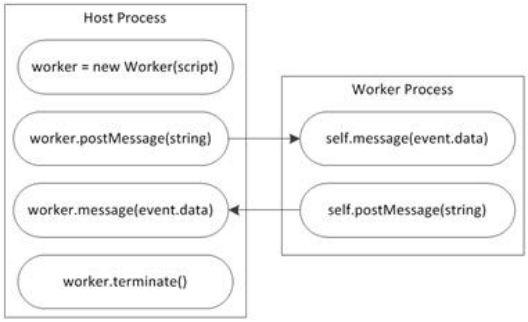


Un job backround poate fi, ca exemplu, un **web worker**. Un web worker este un JavaScript care rulează în fundal, independent de alte scripturi, fără a afecta performanța paginii. Puteți continua să faceți tot ce doriți: clic, selectarea obiectelor etc, în timp ce web worker-ul rulează în background.

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