MMWP2024 - Übungsserie 06 - Webworker

Lehrveranstaltung Multimediale Webprogrammierung
Wintersemester 2024/25

Link zum Kurs: https://bildungsportal.sachsen.de/opal/auth/RepositoryEntry/46557921282
Zum Vertiefen der Kenntnisse: https://www.w3schools.com/html/html5_webworkers.asp

1. Experimentieren Sie mit der Berechnung von großen Zahlen mithilfe von JS. Welches Verhalten stellen Sie fest, wenn Sie während der Berechnung mit der Webseite interagieren?

Ein Beispiel kann hier das SELFHTML-Wiki bieten:

https://wiki.selfhtml.org/wiki/JavaScript/Web_Worker

2. Eine Möglichkeit um die Interaktivität der Webseite während starker Belastung und/oder blockierender Berechnungen zu erhalten ist die Verwendung von Webworkern.

a) Legen Sie einen Worker an und lassen Sie das Ergebnis der Berechnung aus 1. berechnen. Stellen Sie das Ergebnis auf der Webseite dar. Stellen Sie die Berechnungszeit als einfachen Countdown dar.

https://wiki.selfhtml.org/wiki/JavaScript/Web_Worker

https://developer.mozilla.org/en-US/docs/Web/API/Web_Workers_API

https://www.w3schools.com/html/html5_webworkers.asp

b) Ein Paradigma, was hier erkennbar wird, ist die Message-basierte Kommunikation. Lesen Sie dazu in Auszügen folgende Artikel und verschaffen Sie sich einen Überblick:

https://learn.microsoft.com/en-us/dotnet/architecture/microservices/architect-microservice-container-applications/asynchronous-message-based-communication

c) Verschaffen Sie sich einen Überblick über die Möglichkeiten und Limitierungen der Web Worker API, wenden Sie eine Vorgehensweise wie bei der Betrachtung der Canvas API an.

https://developer.mozilla.org/en-US/docs/Web/API/Web_Workers_API

Für weitere Informationen betrachten Sie die folgenden Beispiele:

https://github.com/mdn/dom-examples/tree/main/web-workers/offscreen-canvas-worker

3. Betrachten Sie das folgende Beispiel zu Bubble Sort:

https://github.com/afshinm/50k

Implementieren Sie auf die gleiche Art und Weise Selection Sort

https://www.geeksforgeeks.org/selection-sort/

4. Betrachten Sie das folgende Beispiel:

https://dobsondev.com/2015/05/29/web-workers-demo/

Übernehmen Sie die Codeabschnitte und betrachten Sie den Programmablauf.

5. Webworker erlauben den Erhalt der Interaktivität. Eine mögliche Betrachtungsweise ist es dabei Events und UI-Rendering im Haupt-Thread zu belassen und komplexe Berechnungen und Interaktionen in Webworker zu verschieben.

Zusätzlich stehen weitere Möglichkeiten zur Verwendung von asyncroner Berechnungen in Javaskript zur Verfügung.

a) Machen Sie sich mit dem Konzept des Critical Rendering Path vertraut:

https://developer.mozilla.org/en-US/docs/Web/Performance/Critical_rendering_path

Betrachten Sie insbesondere den Aspekt der Veränderung des DOM und CSSOM durch JS und die Auswirkungen auf die Berechnungszeiten der Webseite.

b) Machen Sie sich mit weiteren Methoden zum asynchronen Laden von Elementen vertraut.

https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes/rel/preload

https://developer.mozilla.org/en-US/docs/Web/HTML/Element/script#async

Selbststudienzeit/Weitere Schritte:

Nutzen Sie die Tutorials für eine weitere Vertiefung: https://developer.mozilla.org/en-US/docs/Web/API/Web_Workers_API/Using_web_workers oder https://web.dev/learn/performance/web-worker-overview