

0. A Docker Desktop telepítése

Windows 10/11:

Töltse le a Docker Desktop szoftvert innen: <https://www.docker.com/products/docker-desktop/> (a legtöbb gépen a Download for Windows AMD64 letöltési linken található verzió lesz a megfelelő).

Futtassa a letöltött telepítőfájlt (rendszerigazgatói jogosultság szükséges). Telepítés után indítsa el, és ellenőrizze, hogy működik-e: nyisson egy Powershell-t vagy parancssort (Win + X, Parancssor vagy Parancssor (rendszerigazgatói)), és írja be, majd nyomjon Entert:

```
docker version
```

Ha a telepítés sikeres volt, akkor a képernyőn a Docker Desktop szoftverre vonatkozó információkat fog látni.

MacOS Monterey (12)/Ventura (13)/Sonoma (14):

Töltse le a Docker Desktop szoftvert innen: <https://www.docker.com/products/docker-desktop/> (válassza a processzorának megfelelő verziót: Mac with Apple silicon/Mac with Intel chip).

Nyissa meg a letöltött .dmg fájlt. Húzza a Docker ikont az Applications mappába, indítsa el a Docker.appot az Applicationsból. Az első indításkor kérhet rendszerigazgatói jelszót, kérheti, hogy engedélyezze a „privileged access”-t (ez kell a konténerek futtatásához), illetőleg előfordulhat, hogy újra kell indítani a Docker Desktopot.

Az Applications (Alkalmazások) mappából indítsa el a Terminált. Írja be, majd üssön Entert:

```
docker version
```

Ha a telepítés sikeres volt, akkor a képernyőn a Docker Desktop szoftverre vonatkozó információkat fog látni.

1. A Browsertrix crawler telepítése

Win + x (Parancssor) (Windows)

Alkalmazások → Segédprogramok → Terminál (macOS)

Letöltjük a képfájlt:

```
docker pull webrecorder/browsertrix-crawler:latest
```

A Docker Desktopban látnunk kell az Images menüpontban a webrecorder/browsertrix-crawlert.

2. Az első futtatás előkészítése

A Browsertrix felhasználói útmutatója itt található:

<https://crawler.docs.browsertrix.com/user-guide/>

Kétféleképpen futtatható:

- parancssoros konfigurációval
- YAML-konfigurációval

3. Konfigurációs fájl létrehozása

Hozzunk létre egy crawls nevű könyvtárat! Ebben helyezzünk el egy másikat collections néven! (Itt lesznek a gyűjteményeink.) Ebben készítsünk egy újabb könyvtárat az első kollekciónknak (a példában oszk)! Ebben hozzunk létre egy szövegfájlt crawl-config néven, majd mentjük el crawl-config.yaml-ként! Ez lesz az első crawlunk konfigurációs fájlja. Ha nem akarjuk parancssorban konfigurálni a crawlakat, akkor minden gyűjtemény mappájában létre kell hoznunk egy ilyen konfigurációs fájlt.

A Browsertrix YAML-lel konfigurálásáról itt lehet olvasni:

<https://crawler.docs.browsertrix.com/user-guide/yaml-config/>

4. Konfiguráció

A crawler futtatásához minimálisan két beállítás megadására van szükség: url (a kiinduló v. ún. seed URL), collection (a gyűjtemény neve). A YAML-ben az URL-eket (és reguláris kifejezéseiket) nem kell idézőjelbe tenni, de a kollekció nevét igen! (Amennyiben numerikus valamilyen beállítás, azt sem kell.)

```
url: https://oszk.hu  
collection: "oszk"
```

A konfigurációs lehetőségekről itt lehet olvasni:

<https://crawler.docs.browsertrix.com/user-guide/cli-options/>

Ha azonban így indítanánk el, a teljes webhelyet lementené, ami sokáig tart, ezért kiegészítjük még egy beállítással:

scopeType: "page"

Ez azt jelenti, hogy csak a nyitóoldalt mentjük. (A scope-ról, azaz a crawl kiterjedéséről alább lesz szó!)

5. Crawl

Indítsuk el az első crawl-t ezekkel a beállításokkal! A Browsertrix-nek meg kell adnunk, hogy melyik könyvtárat használja a crawl-ok könyvtáraként, valamint azt, hogy hol találja a konfigurációs fájlt:

```
docker run --rm -d -v "C:\Users\admin01\crawls:/crawls" -v  
"C:\Users\admin01\crawls\collections\oszk\crawl-  
config.yaml:/app/crawl-config.yaml" webrecorder/browsertrix-  
crawler crawl --config /app/crawl-config.yaml
```

6. Visszajátszás

A ReplayWeb.page szoftverrel tudjuk megnézni a mentést. Ha nincs kliens telepítve, akkor a webes verzióban is megnézhető, de nagy fájlok feltöltése sokáig tarthat.

<https://replayweb.page/>

7. Scope

A crawl kiterjedésének a beállítása alapvető fontosságú. Itt lehet erről olvasni:

<https://crawler.docs.browsertrix.com/user-guide/crawl-scope/>

A legfontosabb beállítási lehetőségek:

- scopeType: page, page-spa, host, domain, prefix (default), any, custom.
- limit: hány oldalt mentsen
- depth: hány ugrásnyira kövesse a linkeket a scope-on belül
- include: reguláris kifejezés, amely megmondja a crawlernek, hogy milyen felépítésű linkeket kövessen
- exclude: reguláris kifejezés, amely megmondja a crawlernek, hogy milyen felépítésű linkeket NE kövessen

Ezeket a beállításokat globálisan (az egész crawlra vonatkozóan) és URL-enként is meg lehet adni.

Ha pl. az OSZK oldaláról csak a híreket szeretnénk menteni, akkor azt így lehet konfigurálni:

```
scopeType: "custom"  
include: https://oszk.hu/(hirek|rendezvenyek).*  
exclude:  
  - facebook.com.*
```

```
- twitter.com.*  
- x.com.*  
- plus.google.com.*  
limit: 10
```

Írjuk bele a konfigurálunkba ezeket a beállításokat, és indítsuk el a crawl! Nézzük meg az eredményt! Majd próbáljuk megismételni úgy, hogy a limitet felemeljük 30-ra! Hasonlítsuk össze a mentéseket!

(Ha nem adnánk meg limitet, akkor minden hírt lementenénk, csak az sokáig tartana.)

8. Képernyőkép-készítés

Ha szeretnénk egy webhelyről képernyőképet készíteni, akkor a konfigurációs fájlban ezt beállíthatjuk (a lehetséges értékek: "view", "thumbnail", "fullPage", "fullPageFinal"):

```
screenshot: "view,thumbnail,fullPageFinal"
```

Készítsünk ezzel a beállítással mentést a kezdőoldalról! Milyen hibát veszünk észre?

Orvoslat: adjuk hozzá blokkoló szabályként:

```
blockRules:  
- url: .*cookie.*.js
```

Ismételjük meg a mentést, és nézzük meg az eredményt!

9. Screencast

A Browsertrixben van lehetőség arra, hogy élőben kövessük a crawl-t (ez az ún. screencast). Ehhez a konfigurációban meg kell adni egy portot, amelyre a crawler kiteszi a képet, valamint a Docker-konténerhez is hozzá kell adni a portot:

```
screencastPort: 8080  
  
docker run --rm -d -p 8080:8080 -v  
"C:\Users\admin01\crawls:/crawls" -v  
"C:\Users\admin01\crawls\collections\oszk\crawl-  
config.yaml:/app/crawl-config.yaml" webrecorder/browsertrix-  
crawler crawl --config /app/crawl-config.yaml
```

Indítsuk el a crawl-t, és a böngészőben nyissuk meg: localhost:8080!

10. Behaviors

A Browsertrix beépítetten szimulál bizonyos felhasználói viselkedéseket: autoscroll, autofetch, autoplay, valamint oldalspecifikus viselkedéseket: siteSpecific. Használható még az autoclick behavior is, de az alapértelmezetten ki van kapcsolva. Ezekről itt lehet olvasni: <https://crawler.docs.browsertrix.com/user-guide/behaviors/>

Ezekon kívül lehetőség van saját viselkedéseket (custom behaviors) definiálni: a Chrome DevTools segítségével rögzíthetjük a felhasználói viselkedést, amelyet azután egy JSON-fájlként megadhatunk a Browsertrixnek, és a crawler minden URL-en lefuttatja az adott tevékenységsorozatot. Még haladóbb lehetőség az összetettebb viselkedések definiálására a Javascript (l. előadás).

Indítsunk egy crawl-t a regiritka.oszk.hu oldalról az alábbi konfigurációs beállításokkal!

```
url: https://regiritka.oszk.hu
scopeType: "page"
```

Az autoplay behaviornek köszönhetően a mentésben lejátszható a videó. Milyen hibát veszünk észre?

Ez utóbbi orvoslásához már custom behavior-t kellene definiálnunk.

11. Quality Assurance

A Browsertrixben van egy beépített minősbiztosítási modul, amelynek segítségével gyűjteményi szinten ellenőrizhetjük, hogy a mentett tartalom visszajátszása milyen minőségben lehetséges. Ehhez szükség van egy olyan mentésre, amelynek a kimenete a Browsertrix saját formátuma, a WACZ, valamint a mentésben kell legyen screenshot, és elő kell állítani a kinyerhető szöveget is, pl. ezzel a konfigurációval:

```
url: https://oszk.hu
collection: "oszk"
scopeType: "custom"
screenshot: "view"
include: https://oszk.hu/(hirek|rendezvenyek).*
exclude:
  - facebook.com.*
  - twitter.com.*
  - plus.google.com.*
  - x.com.*
blockRules:
  - url: .cookie..js
limit: 30
text: "to-warc"
generateWACZ: "true"
```

Majd kell futtatni egy QA crawl-t:

```
docker run --rm -v "C:\Users\admin01\crawls:/crawls"
webrecorder/browsertrix-crawler qa --qaSource
/crawls/collections/oszk/oszk.wacz --collection example-qa --
generateWACZ
```

Ennek eredménye megtekinthető a ReplayWeb.page-ben.

12. Egyéb beállítási lehetőségek

A Browsertrix számos egyéb konfigurálási lehetőséget tartalmaz. Ezek közül a legfontosabbak/legérdekesebbek:

- seedFile: Amennyiben nagyobb mennyiségű kiinduló URL-t szeretnénk megadni, akkor fájlból is beolvastathatjuk (igazán nagy mennyiség esetén fennakadást okozhat, ezen épp dolgoznak a fejlesztők).
- workers: Megadhatjuk az egyszerre indítandó böngészők számát, ami gyorsítja a crawl-t (viszont megnő az erőforrásigény!).
- blockAds: reklámblokkolás
- rolloverSize: Megadhatjuk a létrejövő WARC-fájlok maximális méretét (alapértelmezetten 1 GB).
- mobileDevice: mobil eszköz használatát lehet emulálni
- sizeLimit: a crawl méretének maximalizálása
- timeLimit: a crawl időtartamának maximalizálása