

WebStatistics

Lehrziele:

- Erste Schritte mit Tasks
- Einstieg async/await

Beiliegend liegt eine Csv-Datei (sites.csv) vor, die 10 der meistbesuchten Websites enthält. Ihre Aufgabe ist es, die Startseiten zu laden und folgende Informationen zu ermitteln und ehestmöglich auszugeben:

Nummer der Website entsprechend der Csv-Datei

- Url
- Größe der Startseite
- Ladezeit seit Beginn des Versuchs
- ThreadId des Get-Tasks
- Anzahl der Threads

```
Web Statistics
=====

===== Http-Get with parallel Tasks =====
10 ja.wikipedia.org      Size:      89851 GetTime:   971ms ThreadId: 14 Threads: 22
 3 en.wikipedia.org      Size:      87172 GetTime:  1004ms ThreadId: 13 Threads: 22
 4 twitter.com           Size:     506386 GetTime:  1634ms ThreadId: 14 Threads: 22
 5 whatsapp.com          Size:     105252 GetTime:  1653ms ThreadId: 17 Threads: 22
 7 instagram.com         Size:      67150 GetTime:  1762ms ThreadId: 11 Threads: 22
 2 facebook.com          Size:     105247 GetTime:  1820ms ThreadId:  7 Threads: 22
 1 youtube.com           Size:     615897 GetTime:  1940ms ThreadId: 13 Threads: 22
 6 amazon.com            Size:     405621 GetTime:  2238ms ThreadId: 14 Threads: 22
 9 pinterest.com         Size:      90385 GetTime:  2378ms ThreadId: 14 Threads: 22
 8 live.com              Size:      27923 GetTime:  2478ms ThreadId: 14 Threads: 22

Auswertung für parallele Tasks:
-----
Gesamtzeit: 2554ms, längste Ladedauer: 2478
5 Threads für 10 Get-Tasks

===== Http-Get with async/await =====
 1 youtube.com           Size:     604152 GetTime:   501ms ThreadId: 11 Threads: 22
 2 facebook.com          Size:     105265 GetTime:   415ms ThreadId: 12 Threads: 22
 3 en.wikipedia.org      Size:      87172 GetTime:   545ms ThreadId: 17 Threads: 22
 4 twitter.com           Size:     506387 GetTime:  1138ms ThreadId:  7 Threads: 22
 5 whatsapp.com          Size:     105255 GetTime:   651ms ThreadId: 17 Threads: 22
 6 amazon.com            Size:     404715 GetTime:   793ms ThreadId: 16 Threads: 22
 7 instagram.com         Size:      67154 GetTime:   449ms ThreadId:  4 Threads: 22
 8 live.com              Size:      28585 GetTime:   866ms ThreadId: 14 Threads: 22
 9 pinterest.com         Size:      90385 GetTime:   789ms ThreadId:  9 Threads: 22
10 ja.wikipedia.org      Size:      89851 GetTime:   225ms ThreadId:  9 Threads: 22

Auswertung für async/await Tasks:
-----
Gesamtzeit: 6380ms, längste Ladedauer: 1138
8 Threads für 10 Get-Tasks
Beenden mit Eingabetaste ...
```

Führen Sie die Aufgabe zweimal durch. Beim ersten Versuch starten Sie die einzelnen Requests parallel über Task.Run() und sammeln die Ergebnisse in Continuations. Im zweiten Versuch werden die Requests mittels einer asynchronen Methode über async/await zwar blockadefrei aber nicht parallel gesammelt.

Interpretieren Sie die Ergebnisse in Form von Kommentaren am Beginn des Programms.

Schöne Herbstferien!