

Async await

Wie funktioniert await?

Ein Thread ist wie eine Pipeline, die einen bestimmten Code in die CPU gibt.

Mal angenommen Thread 1 (Ui Thread) will etwas downloaden und dann auf der UI darstellen. Während des Http Requests ist die Ui gesperrt. Weil der Thread 1 darauf wartet, bis der Server geantwortet hat.

Mit await wird ein freier Thread genommen und der macht den Request. Thread 1 wird nun freigegeben. Der Benutzer kann z.B. weiter die Ui wie gewohnt nutzen.

Wenn Thread 2 fertig ist, sagt er, ich bin fertig und möchte zu dem Thread zurück kehren, der ihn erzeugt hat: Thread 1.

Thread 1 übernimmt und der Codeblock wird weiter ausgeführt.

Wenn es nun egal ist, welcher Thread den Codeblock weiter ausführen soll, kann man

```
[crayon-5d617ff34265b689529598/]
```

nutzen.

.Result sollte nicht genutzt werden, da im Exceptio Stacktrace Fehlermeldungen wie "MoveNext()" erscheinen. Diese kommen aus der kompilierten Statemachine. Besser ist es

```
[crayon-5d617ff342666793814434/]
```

zu nutzen, da dann unser Typ und unsere Exception zurück gegeben werden.

Handlebars render Funktion wie bei Mustache

[crayon-5d617ff343f0f437565076/]

Eigene Adblock uBlock Filterliste

Surface 2019-05-19

[crayon-5d617ff344279116317094/]

Zahl als String mit 2 Nachkommastellen und mit Punkt als Tausendertrennzeichen

[crayon-5d617ff344566220454265/]

Query für Rekursives Löschen

```
DECLARE @ObjectID UNIQUEIDENTIFIER = '18AAF1E3-
FBFC-4BB1-957A-60AD7EFC0276'
WITH RowsToDelete AS (
    SELECT [ID]
    FROM [Security].[PermissionLevel]
    WHERE [ID] = @ObjectID

UNION ALL

SELECT [Child].[ID]
FROM [Security].[PermissionLevel] Child
INNER JOIN RowsToDelete Parent ON Parent.[ID] =
[Child].[ParentID]
)
DELETE FROM [Security].[PermissionLevel]
FROM [Security].[PermissionLevel] h
INNER JOIN RowsToDelete d on h.ID = d.ID
go
```

Query um eine UI Tabelle darzustellen. Mit Filter pro

Spalte, Sortierung pro Spalte, variablen Paging und Anzahl Zeilen ohne Filterung

[crayon-5d617ff344814122257247/]

Spalteninformationen zu Prozeduren anzeigen

Der Profiler ist schon eine tolle Sache. Wenn man im SQL Management Studio eine Prozedur ausklappt, dann macht das Management Studio folgende Abfrage um die Informationen auszulesen:

[crayon-5d617ff344b0a391724025/]

Pokemon Tower Defense Tricks

// Letzte Trades

<http://www.ptdtrading.com/latestTrades.php?whichProfile=1>

JQuery hinzufügen dynamisch

[crayon-5d617ff3477fa239978023/]

Blendet alle Regulären Pokemon aus

[crayon-5d617ff347803895205249/]

Blendet Pidgeot aus

[crayon-5d617ff347806243572081/]

[crayon-5d617ff347808761473418/]

Alle Country Codes als SQL Query

Einfach eine Konsolenanwendung erstellen und folgendes Snippet reinkopieren:

[crayon-5d617ff34b1fa831888351/]

Folgender resultCultures Query wird generiert:

[crayon-5d617ff34b207514079671/]

bzw. Folgender resultRegions Query wird generiert:

[crayon-5d617ff34b275966304255/]

Eintrag in AppSettings zur Laufzeit ändern

1. Zunächst benötigt man eine Referenz auf "System.Configuration". Diese ist per Standard nicht in einem Projekt eingebunden
2. Man muss eine AppSettings.config erstellen, die in etwa

so aussieht:

```
[crayon-5d617ff34f11a638963749/]
```

3. Anschließend noch eine Hilfsfunktion:

```
[crayon-5d617ff34f125064087990/]
```

4. Die man ausführen kann:

```
[crayon-5d617ff34f129461037693/]
```

5. Auslesen kann man die AppSettings ganz einfach mit

```
[crayon-5d617ff34f12c140148884/]
```