

# 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-641826a0aac91375913551/]
nutzen.
```

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

```
[crayon-641826a0aac95378564407/]
```

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