



Writing your own: wrapping callbacks









```
override fun onReceive(call: Call<T>, response: Response<T>)
```


`suspend fun <T> C<T>.await(): T = suspend C<T>.await { c ->`

```
override fun onFailure(callee: Callable<T>, t: Throwable) {
```

cancel()
Cancel operation 🙅

c.*runWithException(HttpException(response))*

enqueue(object: Cancellable<T>){

```
if(response.isSuccessful){
```

c.invokeOnceLater }

c.resumeWithException(t)



} else {

```
c.resume(response.body)!!!Fast if null.
```



```
if(response.isSuccessful){
```