



```
suspend fun <T> retrofitSafeCall(
    dispatcher: CoroutineDispatcher,
    timeout: Long = NETWORK_TIMEOUT,
    apiCall: suspend () -> T,
): DataState<T> {
    return withContext(dispatcher) { this: CoroutineScope
1      try {
2          withTimeout(timeout) { this: CoroutineScope
3              val call = apiCall.invoke()
4              DataState.Success(call) ^withTimeout
          } ^withContext
5      } catch (timeoutException: TimeoutCancellationException) {
6          Log.e(NETWORK_ERROR, timeoutException.localizedMessage, timeoutException)
7          DataState.Error(DataState.Cause.Timeout) ^withContext
8      } catch (ioException: IOException) {
9          Log.e(SERVER_ERROR, ioException.localizedMessage, ioException)
10         DataState.Error(ErrorHandler.handleException(ioException)) ^withContext
11     } catch (httpException: HttpException) {
12         Log.e(NETWORK_ERROR, httpException.localizedMessage, httpException)
13         DataState.Error(ErrorHandler.handleException(httpException)) ^withContext
14     }
15 }
```