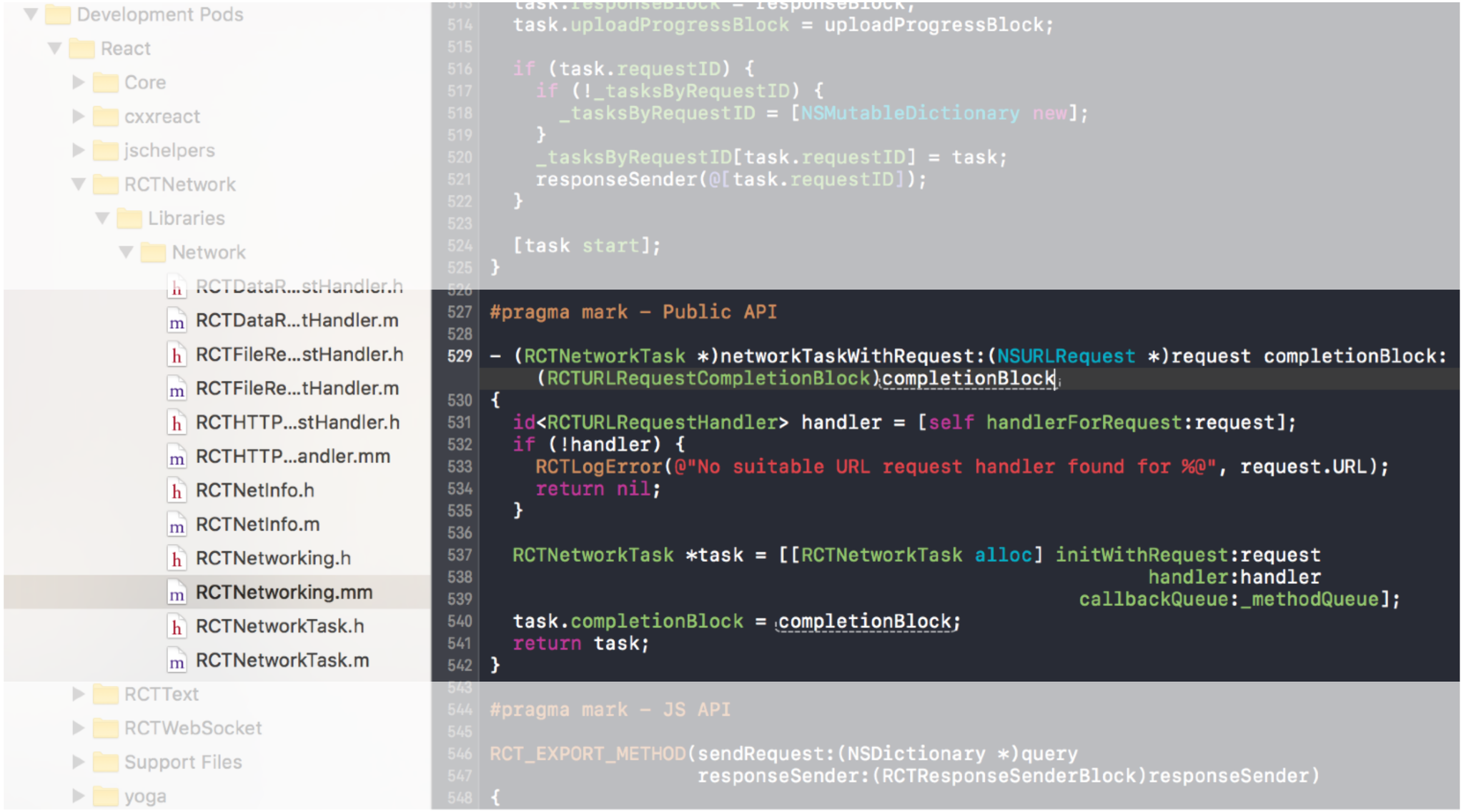
Development Pods
React
Core
cxxreact
jschelpers
RCTNetwork
Libraries
Network
RCTDataR...stHandler.h
RCTDataR...tHandler.m
RCTFileRe...stHandler.h
RCTFileRe...tHandler.m
RCTHTTP...stHandler.h
RCTHTTP...andler.mm
RCTNetInfo.h
RCTNetInfo.m
RCTNetworking.h
RCTNetworking.mm
RCTNetworkTask.h
RCTNetworkTask.m
RCTText
RCTWebSocket
Support Files
yoga
task.uploadProgressBlock = uploadProgressBlock;
if (task.requestID) {
if (!_tasksByRequestID) {
_tasksByRequestID = [NSMutableDictionary new];
}
_tasksByRequestID[task.requestID] = task;
responseSender(@[task.requestID]);
}
[task start];
}
#pragma mark - Public API
- (RCTNetworkTask *)networkTaskWithRequest:(NSURLRequest *)request completionBlock:
(RCTURLRequestCompletionBlock)completionBlock
{
id<RCTURLRequestHandler> handler = [self handlerForRequest:request];
if (!handler) {
RCTLogError(@"No suitable URL request handler found for %@", request.URL);
return nil;
}
RCTNetworkTask *task = [[RCTNetworkTask alloc] initWithRequest:request
handler:handler
callbackQueue:_methodQueue];
task.completionBlock = completionBlock;
return task;
}
#pragma mark - JS API
RCT_EXPORT_METHOD(sendRequest:(NSDictionary *)query
responseSender:(RCTResponseSenderBlock)responseSender)
{

```
22  import okhttp3.ConnectionSpec;
23  import okhttp3.OkHttpClient;
24  import okhttp3.TlsVersion;
25
26  /**
27   * Helper class that provides the same OkHttpClient instance that will be used for all networking
28   * requests.
29   */
30  public class OkHttpClientProvider {
31
32    // Centralized OkHttpClient for all networking requests.
33    private static @Nullable OkHttpClient sClient;
34
35    public static OkHttpClient getOkHttpClient() {
36      if (sClient == null) {
37        sClient = createClient();
38      }
39      return sClient;
40    }
41
```