

# How does RN really work?



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

513 task.responseBlock = responseBlock;

514 task.uploadProgressBlock = uploadProgressBlock;

515

516 if (task.requestID) {

517 if (!\_tasksByRequestID) {

518 \_tasksByRequestID = [NSMutableDictionary new];

519 }

520 \_tasksByRequestID[task.requestID] = task;

521 responseSender(@[task.requestID]);

522 }

523

524 [task start];

525 }

526

527 #pragma mark - Public API

528

529 - (RCTNetworkTask \*)networkTaskWithRequest:(NSURLRequest \*)request completionBlock:

530 (RCTURLRequestCompletionBlock)completionBlock;

531 {

532 id<RCTURLRequestHandler> handler = [self handlerForRequest:request];

533 if (!handler) {

534 RCTLogError(@"No suitable URL request handler found for %@", request.URL);

535 return nil;

536 }

537 RCTNetworkTask \*task = [[RCTNetworkTask alloc] initWithRequest:request

538 handler:handler

539 callbackQueue:\_methodQueue];

540 task.completionBlock = completionBlock;

541 return task;

542 }

543

544 #pragma mark - JS API

545

546 RCT\_EXPORT\_METHOD(sendRequest:(NSDictionary \*)query

547 responseSender:(RCTResponseSenderBlock)responseSender)

548 {

# How does RN really work?



```
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  }
```