

NETWORKING WITH URLSession



PART 12: TESTING & METRICS

XCTestExpectation

```
// given
let promise = expectation(description: "Completion handler invoked")
// when
sessionUnderTest.dataTask(with: url!) {
    data, response, error in
    statusCode = (response as? HTTPURLResponse)?.statusCode
    responseError = error
    promise.fulfill()
}.resume()
waitForExpectations(timeout: 5, handler: nil)
// then
XCTAssertNil(responseError)
XCTAssertEqual(statusCode, 200)
```

FAKING WITH MOCK SESSIONS

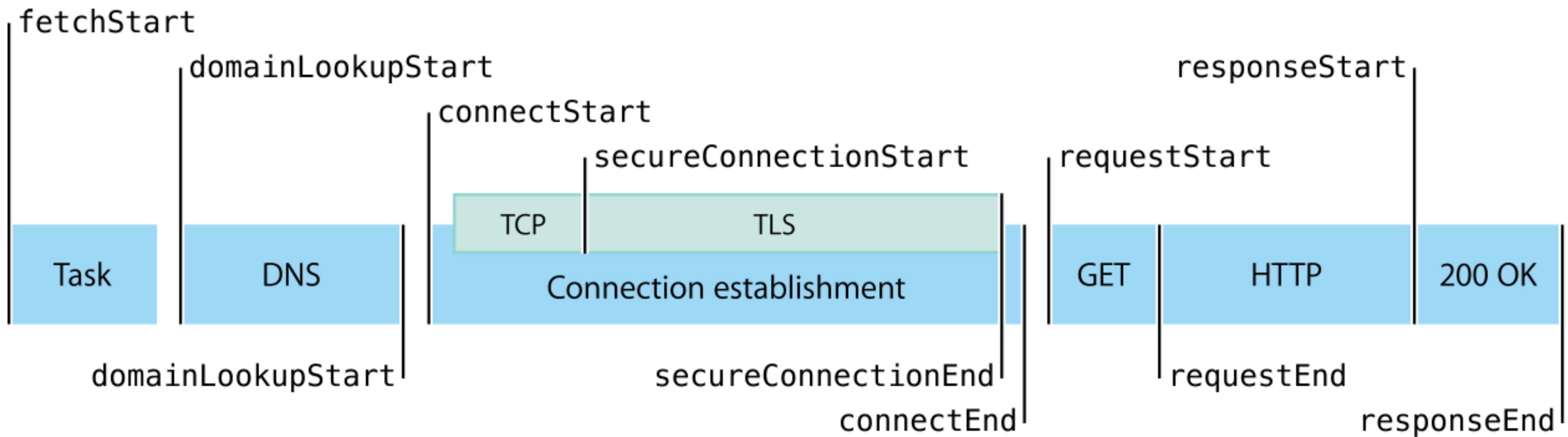
```
// Create fake data and response
// ...
// Create mock session
let sessionMock = URLSessionMock(data: data, response: urlResponse, error: nil)
apiManagerUnderTest.defaultSession = sessionMock
```



DEMO



URLSESSIONTASKMETRICS



```
open var transactionMetrics: [URLSessionTaskTransactionMetrics] { get }  
open var taskInterval: DateInterval { get }  
open var redirectCount: Int { get }
```

CHALLENGE TIME!

```
func urlSession(_ session: URLSession, task: URLSessionTask,  
    didFinishCollecting metrics: URLSessionTaskMetrics) {  
    print(metrics)  
}
```

