

Networking With NSURLSession

NSURL, NSURLRequest, and NSURLResponse

NSURL

https://raywenderlich.com/videos

Scheme

Host

Path

```
let url = NSURL(string: urlString)
```



NSURLComponents

<https://www.raywenderlich.com/115253/swift-2-tutorial-a-quick-start>

<https://www.raywenderlich.com/115279/swift-2-tutorial-part-2-a-simple-ios-app>

```
let components = NSURLComponents(string: "https://www.raywenderlich.com")!  
components.path = "/115253/swift-2-tutorial-a-quick-start"  
let url1 = components.URL  
components.path = "/115279/swift-2-tutorial-part-2-a-simple-ios-app"  
let url2 = components.URL
```



NSURLRequest

NSURLSessionConfiguration

allowsCellularAccess = false

NSURLRequest

allowsCellularAccess = ~~true~~e

GET /115279/swift-2-tutorial-part-2-a-simple-ios-app HTTP/1.1

Host: www.raywenderlich.com

Accept: text/html

Accept-Language: en-us

Accept-Encoding: gzip, deflate

Connection: keep-alive

NSURLRequest

NSURLSessionConfiguration

allowsCellularAccess = false

NSURLRequest

allowsCellularAccess = ~~true~~e

GET /115279/swift-2-tutorial-part-2-a-simple-ios-app HTTP/1.1

Host: www.raywenderlich.com

Accept: text/html

Accept-Language: en-us

Accept-Encoding: gzip, deflate

Connection: keep-alive

NSURLRequest

NSURLSessionConfiguration

allowsCellularAccess = false

NSURLRequest

allowsCellularAccess = ~~true~~e

GET /115279/swift-2-tutorial-part-2-a-simple-ios-app HTTP/1.1

Host: www.raywenderlich.com
Accept: text/html
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: keep-alive

NSURLRequest

NSURLSessionConfiguration

allowsCellularAccess = false

NSURLRequest

allowsCellularAccess = ~~true~~e

GET /115279/swift-2-tutorial-part-2-a-simple-ios-app HTTP/1.1

Host: www.raywenderlich.com

Accept: text/html

Accept-Language: en-us

Accept-Encoding: gzip, deflate

Connection: keep-alive

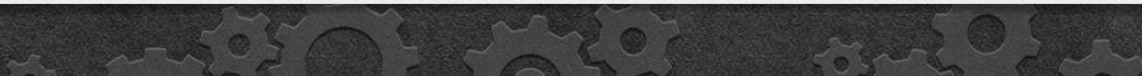
NSURLResponse

```
let task = session.dataTaskWithURL(url!) {  
    (data, response, error) in  
    if error != nil {  
        // Used for client-side errors  
    }  
    // Used for server-side errors  
    (response as? NSHTTPURLResponse)?.statusCode  
}
```


NSURLResponse

```
let task = session.dataTaskWithURL(url) {  
    (data, response, error) in  
    if error != nil {  
        // Used for client-side errors  
    }  
    // Used for server-side errors  
    (response as? NSHTTPURLResponse)?.statusCode  
}
```

Demo



Challenge Time!

The image displays three overlapping screenshots of the Charles 3.11.2 application, illustrating the process of analyzing a network request. The top window shows the 'Request' tab for a GET request to `https://api.flickr.com/services/feeds/photos_public.gne?format=json&jsoncallback=1`. The middle window shows the 'Response' tab for the same request, displaying the raw JSON data. The bottom window shows the 'Overview' tab, which provides a detailed summary of the request and response, including headers, timing, and size information.

Request Overview:

Name	Value
Protocol	HTTP/1.1
SSL	TLSv1.2 (TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256)
Method	GET
Host	api.flickr.com
Content-Type	application/x-javascript; charset=utf-8
Client Address	/127.0.0.1
Remote Address	api.flickr.com/216.115.110.119

Timing:

Event	Time
Request Start Time	2/24/16 17:13:54
Request End Time	2/24/16 17:13:54
Response Start Time	2/24/16 17:13:54
Response End Time	2/24/16 17:13:54
Duration	182 ms
DNS	369 ms
Connect	60 ms
SSL Handshake	57 ms
Request	1 ms
Response	3 ms
Latency	177 ms
Speed	24.18 KB/s
Response Speed	1,376.30 KB/s

Size:

Item	Size
Request Header	278 bytes
Response Header	1,114 KB (1137 bytes)
Request	-
Response	3.02 KB (3091 bytes)
Total	4.40 KB (4506 bytes)
Request Compression	-
Response Compression	82.4% (gzip)