

Network 기초

Network

- 어떤연결을 통해 컴퓨터의 자원을 공유하는 것
- 다른컴퓨터와 데이터를 주고 받는것
- 데이터를 주고받는것이 네트워크

클라이언트 서버 모델(client - server model)

- Network architecture 중 하나
- server : client의 용청에 따라서 데이터를 제공해 주는 컴퓨터
- client : 서버로부터 요청한 데이터를 받는 컴퓨터
- 각각의 컴퓨터가 client, server의 역할에 맞게 구성되어 network통신이 이뤄진다면 우린 이걸 클라이언트 서버 모델이라고 부를수있다.
- 기본적으로 클라이언트가 요청을 해야 서버에서 응답이온다.

protocol

- 객체와 객체와의 통신
- 프로토콜은 컴퓨터끼리 또는 컴퓨터와 단말기 사이에 상호통신 할때 데이터를 에러없이 원활하고 신뢰성있게 주고받기 위해 필요한 약속을 규정하는 것으로서 **통신규약**이라고도 한다.
- 프로토콜의 종류
 - HTTP, HTTPS, FTP, SMTP, SSH(원격제어, 맥, 리눅스)

HTTP

- 클라이언트가 http protocol로 요청하면 서버에서 http protocol로 응답한다.

URL

- URL(파일식별자)은 네트워크 상에서 자원이 어디 있는지(위치)를 알려주기 위한 규약이다.
- URL구성
 - URL은 제일 앞에 자원에 접근할 방법을 정의해 둔 프로토콜 이름을 적는다.
 - 프로토콜 이름 다음에는 프로토콜 이름을 구분하는 구분자인 “:”을 적는다.
 - 만약 IP혹은 domain name정보가 필요한 프로토콜이라면 “:”다음에 “//”를 적는다.
 - 프로토콜명 구분자인 “:” 혹은 “:”다음에는 프로토콜 마다 특화된 정보를 넣는다.
- http://(프로토콜)naver.com(ip)/(lootfile)index.html(파일이 있는위치)
- 클라이언트가 http://naver.com/index.html로 요청하면 서버가 index.html파일 직접전달한다. (브라우저로 보여짐)

URI

- 통합 자원 식별자는 인터넷에 있는 자원을 나타내는 유일한 주소이다. URI의 존재는 인터넷에서 요구되는 기본조건으로서 인터넷 프로토콜 에 항상 붙어 다니다.
- URI vs URL
 - URL은 URI의 한 종류이다.
 - URL은 특정 리소스의 정확한 위치를, URI는 자원을 나타내는 식별자 역할
 - URL : 특정 자원의 위치값(실제파일이 있음)
http://naver.com/index.html
 - URI : restful구조에서 특정자원을 나타내는 함수(실제파일은 없음)
http://naver.com/basefile

참조사이트 : <http://sunychoi.github.io/java/2015/04/27/uri-url.html>

Rest

- URI를 통해 요청, 서버와 모바일 통신할때
- rest는 월드 와이드 웹과 같은 분산 하이퍼미디어 시스템을 위한 소프트웨어 아키텍처의 한 형식이다.
- 엄격한 의미로 rest는 네트워크 아키텍처 원리의 모음이다. 여기서 ‘네트워크 아키텍처 원리’란 자원을 정의하고 자원에 대한 주소를 지정하 는 방법 전반을 일컫는다.
- http://naver.com(<위치)/basefile(<URI)

참조사이트 : <http://egloos.zum.com/killins/v/3092502>

Rest 사용원칙

- Resources : URI들은 쉽게 자원의 내용을 이해할수 있게 만들어야 한다.
- Representations : 객체와 속성을 나타내는 데이터를 **JSON**(key, value타입)이나 XML구조로 전환해서 표현한다.
- Massage : **HTTP Method**를 사용한다.(GET, POST, PUT, DELETE)
- Stateless : server와 clinet의 요청사항은 저장하지 않는다. client는 session상태를 유지한다.

HTTP Request

- URI을 이용해서 Server에 데이터를 요청한다.
- 크게 header와 body로 구조를 나눌 수 있다.
- HTTP Method를 사용해서 요청 메시지를 보낸다.

참조사이트 : <https://blog.outsider.ne.kr/888>

HTTP Request 구조

- Request-line
- Header (general-header | request-header | entity-header)
- [message-body]

Request Header

- host
- accept : 응답
- user - agent : 기기정보, 자동으로 가져간다.

Contents Type

- 서버, 클라이언트 간의 어떠한 데이터를 주고 받을수 있는지를 명시하는 타입
- 대표 type 종류
 - application의 타입
 - content-type : application/json
 - content-type : application/x-www-form-urlencoded
 - multipart 타입
 - content-type : multipart/formed-data <—파일 첨부
 - 여러가지타입의 데이터, 복합적인, 주로 image합칠때 사용

Message Body

- GET HTTP Method의 파라미터는 URL에 포함시켜서 정보를 보낸다.
- ex) http://siteURL/age?fristName=kim&lastName=gh (파라미터)
- GET을 제외한 나머지 Method의 파라미터는 content-type에 맞는 형식으로 body message에 포함시켜 요청을 보낸다.

JSON data type(string)

- JSON은 속성-값 쌍으로 이루어진 데이터 오브젝트를 전달하기 위해 인간이 읽을 수 있는 텍스트를 사용하는 개방형 표준 포맷이다. 비동기 브라우저/서버통신을 위해, 넓게는 XML을 대체하는 주요 데이터 포맷이다. 특히, 인터넷에서 자료를 주고 받을 때 그 자료를 표현하는 방법으 로 알려져 있다. 자료의 종류에 큰 제한은 없으며, 특히 컴퓨터 프로그램의 변수값을 표현하는 데 적합하다.

IOS Network

NSURLSession(block형식)

- server에 data를 요청할수 있는 가장 손쉬운 API
- NSURLSession은 HTTP requests를 통해 데이터를 보내고 받는데 중요한 객체이다. NSURLSessionConfiguration을 통해 3가지의 타입의 Session을 만들수 있다.
- NSURLSession은 NSURLConnection을 대체하기 위해 나온 새로운 클래스로 보다 사용하기 쉽고, 더 강력한 기능(백그라운드 다운로드/업로드)을 지원한다.

NSURLSessionConfiguration설정

- NSURLSessionConfiguration은 session의 설정에 관련된 객체 입니다. 이 객체는 아래 3가지중 하나와 함께 생성되며, 그외에도 타임아웃, 캐시정책 또는 추가적인 HTTP헤더와 같은 여러 프로퍼티를 설정할 수 있게 해줍니다.
- defaultSessionConfiguration : 디폴트 configuration 객체를 생성합니다. 사용자들을 disk-persisted global cache, credential, cookie storage objects를 사용하게 됩니다.
- backgroundSessionConfiguration : 세션이 백그라운드에서 다운로드 작업과 업로드 작업을 수행할 수 있도록 합니다. 이 데이터 전송작업은 앱이 suspended나 not running일 때에도 수행됩니다.

NSURLSessionTask (요청 1개가 태스크 1, 작업의 단위)

- NSURLSessionTask는 세션의 작업 하나를 나타내는 추상클래스입니다. 세션은 데이터를 다운로드하거나 업로드하는 작업(task)를 생성합니다.
- NSURLSessionDataTask : HTTP GET요청으로 서버로부터 데이터를 가져오는 작업을 수행합니다.
- NSURLSessionUploadTask : 디스크로부터 웹서버로 파일을 업로드하는 작업을 수행합니다. 주로 HTTP POST, PUT메소드를 이용합니다.
- NSURLSessionDownloadTask : 리모트 서버로부터 파일을 다운로드하는 작업을 수행 합니다.

실행순서

- NSURLSession객체를 만든다. configuration 설정은 option
- 알맞는 NSURLSessionTask 객체를 만든다.
- [task resume] 메소드를 이용해서 테스크를 실행 시킨다.
- 테스크에 대한 respond가 task메소드 블록 안에서 실행된다.

```
//사진이미지가 있는 URL주소 지정
NSURL *imageurl = [NSURL URLWithString:@"https://lh3.googleusercontent.com/-_Q0T4pWDKZc/Vy6CpgwI1XI/AAAAAAAAUMg/SWApFadvA1A/s0/tumblr_o61v4cx1QP1qk12q0o1_1280.jpg"];

//인디게이터 시작 / 로딩중 표시
[self.indicator startAnimating];

//세션 설정
NSURLSession *session = [NSURLSession sessionWithConfiguration:
                        [NSURLSessionConfiguration defaultSessionConfiguration]];










//태스크 : 세션을 통해 보낸다.
NSURLSessionTask *task = [session dataTaskWithURL:imageurl
                        completionHandler:^(NSData * _Nullable data,
                                           NSURLResponse * _Nullable response,
                                           NSError * _Nullable error) {

    if (data) {
        //파일은 데이터 형식으로 온다
        UIImage *image2 = [UIImage imageData:data];
        if (image2) {
            sleep(1);
            //ui를 수정하기 위해서는 main queue에 해주어야한다.
            dispatch_queue_t queue = dispatch_get_main_queue();
            //비동기 방식
            dispatch_async(queue, ^{
                self.image1.image = image2;
                self.image1.contentMode = UIViewContentModeScaleToFill;

                //인디게이터 정지 후 숨김
                [self.indicator stopAnimating];
                [self.indicator setHidden:YES];
            });
        }
    }
});
// resume을 해야 위의 코드가 실행이된다.
[task resume];
```

ATS

- ios9이상의 버전에는 ATS기술이 기본적으로 적용된다. ATS는 앱과 웹 서비스 간 연결보안을 강화하는 기술로 이 기술이 적용되면 기존에 ios앱에서 사용하던 암호화되지 않는 http통신은 os내부에서 강제적으로 차단된다.

			 Network1 	 Network1 	 Info.plist 	No Selection
Key			Type		Value	
▼ Information Property List			Dictionary		(14 items)	
▼ App Transport Security Settings			Dictionary		(1 item)	
Allow Arbitrary Loads			Boolean		YES	