IOS下关于cookie的使用管理

1. iOS对cookie操作的相关类：NSHTTPCookie、NSHTTPCookieStorage
2. NSHTTPCookie对象代表一个HTTP cookie，这是一个不可改变的对象，从一个包含cookie属性的字典初始化，这个类可以用来手动创建cookie的properties：含如下属性

-(NSUInteger)version;

-(NSString \*)name;

-(NSString \*)value;

-(NSString \*)domain; //[url host] 服务器ip

-(NSString \*)path;

-(NSString \*)isSessionOnly;//等等

1. 可以通过手动赋值的方式创建cookie，如：

+(id)cookieWithProperties:(NSDictionary \*)properties;

-(id)initWithProperties:(NSDictionary \*)properties;

1. 可以从cookie中读到所有属性

-(NSDictionary \*)properties;

1. 使用类方法可将NSHTTPCookie实例与HTTP cookie header相互转化

+(NSDictionary \*)requestHeaderFieldsWithCookies:(NSArray \*)cookies;

+(NSArray \*)cookiesWithResponseHeaderFields:(NSDictionary \*)headerFields forURL:(NSURL \*)theURL;

注：第二个方法会忽略headerFields中与cookies无关字段，若cookies没指定domain，则使用theURL的domain，如果未指定path，则使用／

1. 除非NSURLRequest明确指定不使用cookie(HTTPShouldHandleCookies设为NO)，否者URL loading system会自动为NSURLRequest发送合适的存储cookie。从NSURLResponse返回的cookie也会根据当前的cookie访问策略 （cookie acceptance policy）接收到系统中
2. NSHTTPCookieStorage提供了管理所有NSHTTPCookie对象的接口。在OS X里，cookie是所有程序中共享的，而在ios里，cookie只在当前应用有效
3. 获取共享的NSHTTPCookieStorage单例对象

+(NSHTTPCookieStorage \*)sharedHTTPCookieStorage;

1. 获取当前所有存储的所有cookie或针对特定URL的cookie

-(NSArray \*)cookies;

-(NSArray \*)cookiesForURL:(NSURL \*)theURL;

1. 添加/删除cookie

-deleteCookie:

-setCookie:

-setCookies:forURL:mainDocumentURL:

1. 读取/修改cookie接收策略，默认为NSHTTPCookieAcceptPolicyAlways;

-(NSHTTPCookieAcceptPolicy)cookieAcceptPolicy;

-(void)setCookieAcceptPolicy:(NSHTTPAcceptPolicy)aPolicy;

1. 相关通知

NSHTTPCookieManagerCookiesChangedNotification:当单例中cookies变化时发出通知

NSHTTPCookieManagerAcceptPolicyChangedNotification:当cookie接收策略发生变化时发出通知

1. NSHTTPCookie requestHeaderFieldsWithCookies:得到的HTTP cookie header格式如下：{

Cookie = “ASP.NET\_SessionId=……;AccessToken=……；AccessUserId=……“  
}

1. 实例
2. 登陆cookie管理
3. 保存登陆session

-(void)saveLoginSession{

NSUserDefault \*defaults = [NSUserDefaults standardUserDefaults];

NSArray \*allCookies = [NSHTTPCookieStorage sharedHTTOCookieStorage] cookies];

for(NSHTTPCookie \*cookie in allCookies){  
 if([cookie.name isEqualToString:kServerSessionCookie]){  
 NSMutableDictionary \*cookieDictionary = [defaults dictionaryForKey:kLocalCookieName] mutableCopy};

[cookieDictionary setValue:cookie.properties forKey:”cookieDict”];

[default setObject:cookieDictionary forKey:kLocalCookieName];

[default synchronize];

break;

}

}  
}

1. 更新登陆session cookie，程序启动时调用，确保每次cookie是最新的

-(void)updateSession{

NSUserDefaults \*defaults = [NSUserDefaults standardUserDefaults];

NSDictionary \*cookieDict = [defaults dictionaryForKey:kLocalCookieName];

NSDictionary \*cookieProperties = [cookieDict valueForKey:@”cookieDict”];

If(cookieProperties){

NSHTTPCookie \*cookie = [NSHTTPCookie cookieWithProperties:cookieProperties];

[[NSHTTCookieStorage sharedHTTPCookieStorage] setCookie:cookie];  
}

}

1. 退出登陆时清除登陆session cookie

-(void)removeLoginSession{

NSUserDefault \*defaults = [NSUserDefaults standardUserDefaults];

[defaults removeObjectForKey:kLocalCookieName];

[defaults synchronize];//添加同步锁，防止多线程访问  
}

1. h5页面向原生页面请求cookie

-(NSString \*)getCookie{

NSArray \*cookies = [NSHTTPCookieStorage sharedHTTPCookiesStorage] cookiesForURL:url];

NSDictionary \*cookieDic = [NSHTTPCookie requestHeaderFieldsWithCookies:cookies];

Return cookieDic[@”Cookie”];  
}