

IOS 下关于 cookie 的使用管理

1. iOS 对 cookie 操作的相关类：NSHTTPCookie、NSHTTPCookieStorage

(1) 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

```
-(NSInteger)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;
```

2) 可以从 cookie 中读到所有属性

```
-(NSDictionary *)properties;
```

3) 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 /

4) 除非 NSURLRequest 明确指定不使用 cookie (HTTPShouldHandleCookies 设为 NO)，否则 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

1) 获取共享的 NSHTTPCookieStorage 单例对象

```
+(NSHTTPCookieStorage *)sharedHTTPCookieStorage;
```

2) 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;
```

3) 添加/删除 cookie

```
-deleteCookie:
```

```
-setCookie:
```

```
-setCookies:forURL:mainDocumentURL:
```

4) 读取/修改 cookie 接收策略，默认为 NSHTTPCookieAcceptPolicyAlways;

```
-(NSHTTPCookieAcceptPolicy)cookieAcceptPolicy;
```

```
-(void)setCookieAcceptPolicy:(NSHTTPAcceptPolicy)aPolicy;
```

5) 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

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

```
Cookie = "ASP.NET_SessionId=.....;AccessToken=.....; AccessUserId=..... "
```

```
}
```

3. 实例

(1) 登陆 cookie 管理

1) 保存登陆 session

```
-(void)saveLoginSession{
```

```
    NSUserDefaults *defaults = [NSUserDefaults standardUserDefaults];
```

```
    NSArray *allCookies = [NSHTTPCookieStorage sharedHTTPCookieStorage]
    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;
    }
}
}
}

```

2) 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];
        [[NSHTTPCookieStorage sharedHTTPCookieStorage] setCookie:cookie];
    }
}

```

3) 退出登陆时清除登陆 session cookie

```

-(void)removeLoginSession{
    NSUserDefaults *defaults = [NSUserDefaults standardUserDefaults];
    [defaults removeObjectForKey:kLocalCookieName];
}

```

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

(2) h5 页面向原生页面请求 cookie

```
-(NSString *)getCookie{  
    NSArray *cookies = [NSHTTPCookieStorage sharedHTTPCookieStorage]  
cookiesForURL:url];  
  
    NSDictionary *cookieDic = [NSHTTPCookie  
requestHeaderFieldsWithCookies:cookies];  
  
    Return cookieDic[@" Cookie" ];  
}
```