

网络请求

iOS 网络

- 网络请求
 - iOS 网络
 - NSURLConnection
 - NSURLSession
 - 三种模式
 - 三种任务
 - 事例代码

API调试工具：

去调试

马上下载SDK，简化API调用

接口地址：

http://apis.baidu.com/apistore/weatherservice/citylist

请求方法：

GET

请求参数(header)：

参数名	类型	必填	参数位置	描述
apikey	string	是	header	API密钥

请求参数(urlParam)：

参数名	类型	必填	参数位置	描述
cityname	string	是	urlParam	城市中文名称

NSURLConnection

NSURLConnection 提供了两种方式来实现连接,一种是同步的另一种是异步的,异步的连接将会创建一个新的线程,这个线程将会来负责下载的动作。而对于同步连接,在下载连接和处理通

讯时,则会阻塞当前调用线程。

许多开发者都会认为同步的连接将会堵塞主线程,其实这种观点是错误的。一个同步的连接是会阻塞调用它的线程。如果你在主线程中创建一个同步连接,没错,主线程会阻塞。但是如果你并不是从主线程开启的一个同步的连接,它将会类似异步的连接一样。因此这种情况并不会堵塞你的主线程。事实上,同步和异步的主要区别就是运行 runtime 为会异步连接创建一个线程,而同步连接则不会。

发送请求的三个步骤:

- 1.设置请求路径
- 2.创建请求对象
- 3.发送请求
 - 3.1发送同步请求 (一直在等待服务器返回数据,这行代码会卡住,如果服务器,没有返回数据,那么在主线程UI会卡住不能继续执行操作) 有返回值
 - 3.2发送异步请求:没有返回值

说明:任何NSURLRequest默认都是get请求。NSMutableURLRequest则可以选择POST并添加请求头。

```
1.  /**
2.   * 发送同步请求 GET
3.   */
4.  - (void)sync
5.  {
6.      NSString * httpUrl =
7.      @"http://apis.baidu.com/apistore/weatherservice/cityname";
8.      NSString * httpArg = @"cityname=%E5%8C%97%E4%BA%AC";
9.      NSString * apiKey = @"490c25080dfdadce6a8505b16825fda8";
10.     // 0.请求路径
11.     NSURL *url = [NSURL URLWithString:[NSString stringWithFormat:@"%s%s%s",
12.     @"",httpUrl,httpArg]];
13.
14.     // 1.创建请求对象
15.     // NSURLRequest *request = [NSURLRequest requestWithURL:url];
16.     NSMutableURLRequest * murequest = [NSMutableURLRequest requestWithURL:url];
17.     [murequest setValue:apiKey forHTTPHeaderField:@"apikey"];
18.     // 2.发送请求
19.     // sendSynchronousRequest阻塞式的方法,等待服务器返回数据
```

```

19.     NSHTTPURLResponse *response = nil;
20.     NSError *error = nil;
21.
22.     NSData *data = [NSURLConnection sendSynchronousRequest:mrequest re
turningResponse:&response error:&error];
23.
24.     // 3.解析服务器返回的数据（解析成字符串）
25.     NSString *string = [[NSString alloc] initWithData:data encoding:NSU
TF8StringEncoding];
26.     NSLog(@"%@ %@", string, response.allHeaderFields);
27. }

```

```

1.  /**
2.   * 发送异步请求 POST
3.   */
4.  - (void)async
5.  {
6.      NSString * httpUrl =
@"http://apis.baidu.com/apistore/weatherservice/cityname";
7.      NSString * httpArg = @"cityname=%E5%8C%97%E4%BA%AC";
8.      NSString * apiKey = @"490c25080dfdadce6a8505b16825fda8";
9.      // 0.请求路径
10.     NSURL *url = [NSURL URLWithString:[NSString stringWithFormat:@"%@?%
%@",httpUrl,httpArg]];
11.
12.     // 1.创建请求对象
13.     NSMutableURLRequest * mrequest = [NSMutableURLRequest requestWithU
RL:url];
14.     [mrequest setValue:apiKey forHTTPHeaderField:@"apikey"];
15.
16.     // 请求超时
17.     mrequest.timeoutInterval = 50;
18.     // 设置请求类型
19.     mrequest.HTTPMethod = @"POST";
20.     // 将请求体转成NSData
21.     NSDictionary * body = @{ @"cityname":@"深圳" };
22.     mrequest.HTTPBody = [NSJSONSerialization dataWithJSONObject:body o
ptions:NSJSONWritingPrettyPrinted error:nil];
23.     // 2.发送请求
24.     [NSURLConnection sendAsynchronousRequest:mrequest queue:
[[NSOperationQueue alloc] init] completionHandler:^(NSURLResponse
*response, NSData *data, NSError *connectionError) {
25.         // 请求完毕会来到这个block
26.         // 3.解析服务器返回的数据（解析成字符串）

```

```
27.         NSString *string = [[NSString alloc] initWithData:data encoding
    ::NSUTF8StringEncoding];
28.         NSLog(@"%@", string);
29.
30.     }];
31. }
```

NSURLSession

三种模式

- **默认会话模式**（default）：工作模式类似于原来的NSURLConnection，使用的是基于磁盘缓存的持久化策略，使用用户keychain中保存的证书进行认证授权。
- **瞬时会话模式**（ephemeral）：该模式不使用磁盘保存任何数据。所有和会话相关的 caches，证书，cookies等都被保存在RAM中，因此当程序使会话无效，这些缓存的数据就会被自动清空。
- **后台会话模式**（background）：该模式在后台完成上传和下载，在创建Configuration对象的时候需要提供一个NSString类型的ID用于标识完成工作的后台会话。

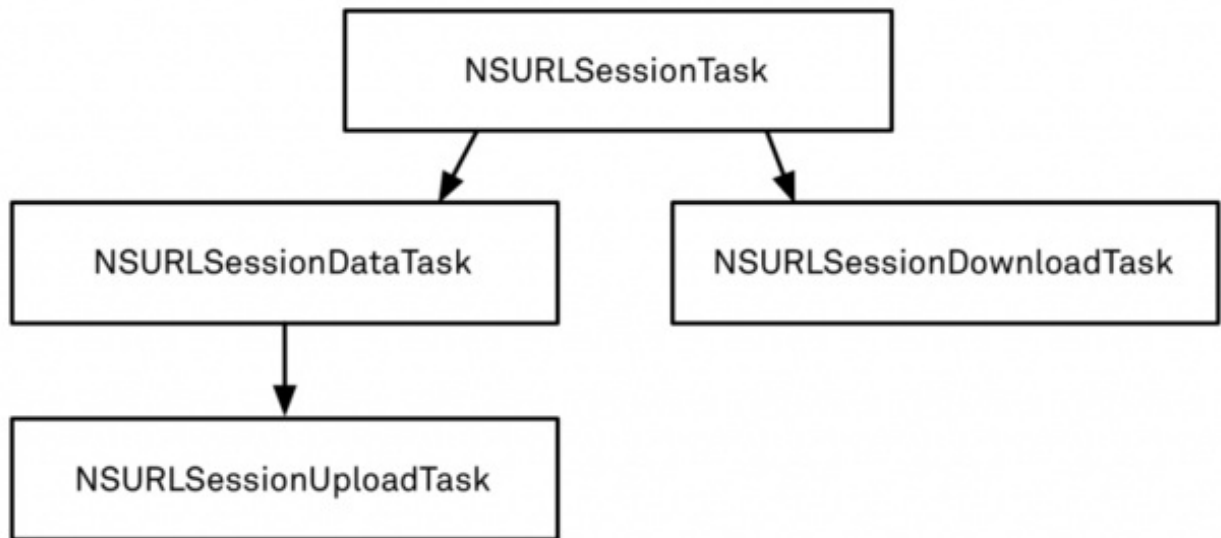
三种任务

NSURLSession使用NSURLSessionTask来具体执行网络请求的任务 NSURLSessionTask支持网络请求的取消、暂停和恢复，比如下载文件暂停之后再恢复就能够自动从上次的进度继续下载。

NSURLSessionTask还能获取数据的读取进度。

NSURLSessionTask的三种类型:

- 1> NSURLSessionDataTask 处理一般的NSData数据对象, 比如通过GET或POST方式从服务器获取JSON或XML返回等等, 但不支持后台获取.
- 2> NSURLSessionUploadTask 用于上传文件, 支持后台上传.
- 3> NSURLSessionDownloadTask 用于下载文件, 支持后台下载



事例代码

```
1.  - (void)download
2.  {
3.      NSString * httpUrl =
4.      @"http://apis.baidu.com/apistore/weatherservice/cityname";
5.      NSString * httpArg = @"cityname=%E5%8C%97%E4%BA%AC";
6.      // 获得NSURLSession对象
7.      NSURLSession *session = [NSURLSession sharedSession];
8.      // 获得下载任务
9.      NSURLSessionDownloadTask *task = [session downloadTaskWithURL:[NSURL
10.      URLWithString:httpUrl] completionHandler:^(NSURL *location,
11.      NSURLResponse *response, NSError *error) {
12.          // 文件将来存放的真实路径
13.          NSString *file = [[NSSearchPathForDirectoriesInDomains(NSCaches
14.          Directory, NSUserDomainMask, YES) lastObject]
15.          stringByAppendingPathComponent:response.suggestedFilename];
16.
17.          // 剪切location的临时文件到真实路径
18.          NSFileManager *mgr = [NSFileManager defaultManager];
19.          [mgr moveItemAtURL:location toURL:[NSURL URLWithString:file]
20.          error:nil];
21.      }];
22.
23.      // 启动任务
24.      [task resume];
25.  }
```

```

21. - (void)post
22. {
23.     NSString * httpUrl =
24.     @"http://apis.baidu.com/apistore/weatherservice/cityname";
25.     NSString * httpArg = @"cityname=%E5%8C%97%E4%BA%AC";
26.     // 获得NSURLSession对象
27.     NSURLSession *session = [NSURLSession sharedSession];
28.     // 创建请求
29.     NSMutableURLRequest *request = [NSMutableURLRequest requestWithURL:
30.     [NSURL URLWithString:httpUrl]];
31.     request.HTTPMethod = @"POST"; // 请求方法
32.     request.HTTPBody = [httpArg dataUsingEncoding:NSUTF8StringEncoding]
33.     ; // 请求体
34.     // 创建任务
35.     NSURLSessionDataTask *task = [session dataTaskWithRequest:request
36.     completionHandler:^(NSData *data, NSURLResponse *response, NSError *err
37.     or) {
38.         NSLog(@"%@", [NSJSONSerialization JSONObjectWithData:data optio
39.         ns:kNilOptions error:nil]);
40.     }];
41.     // 启动任务
42.     [task resume];
43. }
44.
45. - (void)get2
46. {
47.     NSString * httpUrl =
48.     @"http://apis.baidu.com/apistore/weatherservice/cityname";
49.     NSString * httpArg = @"cityname=%E5%8C%97%E4%BA%AC";
50.     NSURLSessionConfiguration * config = [NSURLSessionConfiguration
51.     defaultSessionConfiguration];
52.     [config setHTTPAdditionalHeaders:@{ @"cityname":@"深圳" }];
53.     // 获得NSURLSession对象
54.     NSURLSession *session = [NSURLSession
55.     sessionWithConfiguration:config];
56.
57.     // 创建任务
58.     NSURLSessionDataTask *task = [session dataTaskWithURL:[NSURL URLWit
59.     hString:httpArg] completionHandler:^(NSData *data, NSURLResponse
60.     *response, NSError *error) {
61.         NSLog(@"%@", [NSJSONSerialization JSONObjectWithData:data optio
62.         ns:kNilOptions error:nil]);
63.     }];
64.     // 启动任务
65.     [task resume];

```

```

54.     }
55.
56.     - (void)get
57.     {
58.         NSString * httpUrl =
59.         @"http://apis.baidu.com/apistore/weatherservice/cityname";
60.         NSString * httpArg = @"cityname=%E5%8C%97%E4%BA%AC";
61.         // 获得NSURLSession对象
62.         NSURLSession *session = [NSURLSession sharedSession];
63.         // 创建任务
64.         NSURLSessionDataTask *task = [session dataTaskWithRequest:[NSURLReq
65.         uest requestWithURL:[NSURL URLWithString:[NSString stringWithFormat:@"
66.         %@?%@",httpUrl,httpArg]] completionHandler:^(NSData *data,
67.         NSURLResponse *response, NSError *error) {
68.             NSLog(@"%@", [NSJSONSerialization JSONObjectWithData:data optio
69.             ns:kNilOptions error:nil]);
70.         }];
71.         // 启动任务
72.         [task resume];
73.     }

```

#