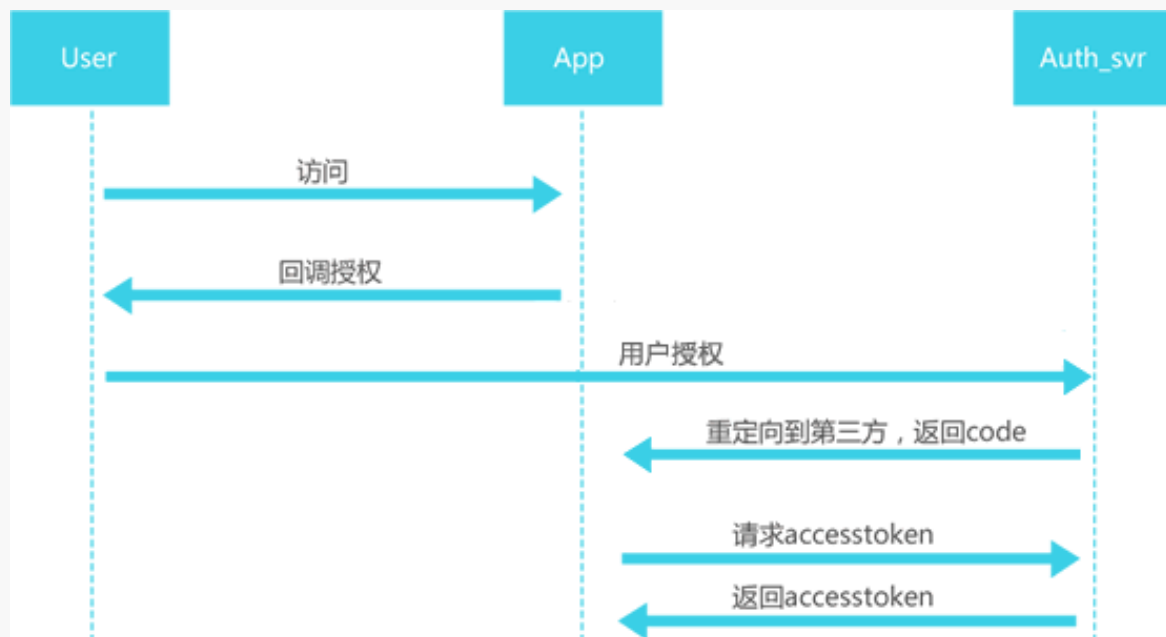


OAuth认证



- 引导需要授权的用户到如下地址:

```
https://api.weibo.com/oauth2/authorize?client_id=YOUR_CLIENT_ID&response_type=code&redirect_uri=YOUR_REGISTERED_REDIRECT_URI
```

- 如果用户同意授权, 页面跳转至 `YOUR_REGISTERED_REDIRECT_URI/?code=CODE`
- 换取Access Token

```
https://api.weibo.com/oauth2/access_token?client_id=YOUR_CLIENT_ID&client_secret=YOUR_CLIENT_SECRET&grant_type=authorization_code&redirect_uri=YOUR_REGISTERED_REDIRECT_URI&code=CODE
```

返回值 { "access_token": "SIAV32hkKG", "remind_in ": 3600, "expires_in": 3600 }

- 使用获得的OAuth2.0 Access Token调用API

目标

放弃新浪官方SDK; 自己实现认证流程, 登陆微博开放平台; 用AFNetworking进行HTTP通信;

实现

宏定义

```
#define kAppKey @"4244467607"
#define kRedirectURL @"http://weiji.info"
#define kAccessToken @"kAccessToken"
#define kLoginsuccess @"kLoginsuccess"
```

引入第三方库

使用CocoaPods引入第三方库 `AFNetworking`

比较优秀的第三方库: <http://cocoapods.wantedly.com/>

控制器框架

`ViewController` 根视图控制器，点击登陆按钮，用模态方式弹出一个视图控制器，用于将用户引导到登陆界面；登陆成功后，取消模态视图，返回根视图控制器。

登陆成功后，可以点击发微博按钮，用模态方式弹出一个视图控制器，用于发送微博，发送完成后，取消模态视图，返回根视图控制器。

根视图控制器

添加notification，用于保存登陆成功后获得的token

```
- (void)viewDidLoad {
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
    [[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(logInSuccess:) name:kLoginsuccess object:nil];
}

-(void)logInSuccess:(NSNotification *)notification{
    self.token = notification.object;
}
```

控制在登陆后才能进去发微博界面

```

-(BOOL)shouldPerformSegueWithIdentifier:(NSString *)identifier sender:(id)
sender{
    if ([identifier isEqualToString:@"sendMessage"]) {
        if (!self.token) {
            return NO;
        }
    }
    return YES;
}

```

在发送微博时，给发送界面提供token

```

-(void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender{
    if ([segue.identifier isEqualToString:@"sendMessage"]) {
        UINavigationController *nav = segue.destinationViewController;
        SendMessage *message = (SendMessage *)[nav viewControllers] firstObject];
        message.token = self.token;
    }
}

```

登陆视图控制器

将用户引导到登陆界面,在这里，不再是访问，而是引导到登陆的HTML界面

```

- (void)viewDidLoad {
    [super viewDidLoad];
    // Do any additional setup after loading the view.

    //不将视图向四周延伸
    self.edgesForExtendedLayout = UIRectEdgeNone;

    NSString *urlString =[NSString stringWithFormat:@"https://api.weibo.com/oauth2/authorize?client_id=%@&redirect_uri=%@&response_type=code&display=mobile",kAppKey,kRedirectURL];

    NSMutableURLRequest *request = [[NSMutableURLRequest alloc] initWithURL:[NSURL URLWithString:urlString]];
    [self.webView loadRequest:request];
}

```

授权后，会重定向到另一个URL (YOUR_REGISTERED_REDIRECT_URI/?code=CODE)，而这个URL中就包含了我们上面所说的要返回的数据（重点是code这个参数）；在webView的delegate方法中可以获取重定向的URL，从这个URL中截取code参数值，将code这个参数值添

加到获取access_token这个API的参数中后调用，调用后获取返回的参数，对返回的参数进行处理进而获取access_token。

```
#pragma mark - web delegate

-(BOOL) webView:(UIWebView *)webView shouldStartLoadWithRequest:(NSURLRequest *)request navigationType:(UIWebViewNavigationType)navigationType
{
    NSURL *resultURL = [request URL];
    NSString *tempStr = [resultURL absoluteString];
    NSLog(@"resultstr:%@",resultURL);
    //如果访问的URL的字符串是以回调地址开头,获取code
    if([tempStr hasPrefix:kRedirectURL]){
        NSLog(@"back url: %@",tempStr);

        NSRange rangeOne;
        rangeOne = [tempStr rangeOfString:@"code="];
        NSRange range = NSMakeRange(rangeOne.length+rangeOne.location, tempStr.length-(rangeOne.length+rangeOne.location));
        NSString *codeString = [tempStr substringWithRange:range];
        NSLog(@"code = %@",codeString);
        //获取access_token
        NSDictionary *parameters = @{@"client_id":kAppKey,
                                     @"client_secret": @"cb551568671f50ab311e8eddb96d0137",
                                     @"grant_type": @"authorization_code",
                                     @"redirect_uri": kRedirectURL,
                                     @"code": codeString,
                                     @"type": @"focus-c"};

        //普通HTTP请求
        AFHTTPRequestOperationManager *manager = [AFHTTPRequestOperationManager manager];
        // 设置请求格式
        manager.requestSerializer = [AFHTTPRequestSerializer serializer];
        // 设置返回格式
        manager.responseSerializer = [AFJSONResponseSerializer serializer];

        manager.responseSerializer.acceptableContentTypes = [NSSet setWithObject:@"text/plain"];

        [manager POST:@"https://api.weibo.com/oauth2/access_token" parameters:parameters success:^(AFHTTPRequestOperation *operation, id responseObject) {
```

```

        NSLog(@"JSON: %@", responseObject);

        NSDictionary *info = (NSDictionary *)responseObject;
        NSString *token = [info objectForKey:@"access_token"];

        [[NSNotificationCenter defaultCenter] postNotificationName:kLoginSuccess
        object:token userInfo:nil];
        [self dismissViewControllerAnimated:YES completion:nil];
    } failure:^(AFHTTPRequestOperation *operation, NSError *error) {
        NSLog(@"responseString:%@ Error: %@", operation.responseString, error);
        [self dismissViewControllerAnimated:YES completion:nil];
    }];

    }
    return YES;
}

@end

```

发送微博控制器

该控制器包含一个**textView**，点击发送按钮，触发事件将用户输入内容发送到微博平台

```

#define kUpdateUrl @"https://api.weibo.com/2/statuses/update.json"

- (IBAction)send:(id)sender {
    NSString *message = self.textView.text;
    NSLog(@"%@", message);
    AFHTTPRequestOperationManager *manager = [AFHTTPRequestOperationManager manager];
    NSDictionary *parameters = @{@"access_token":self.token, @"status":message};
    [manager POST:kUpdateUrl parameters:parameters success:^(AFHTTPRequestOperation *operation, id responseObject) {
        NSLog(@"%@", responseObject);
        [self dismissViewControllerAnimated:YES completion:nil];
    } failure:^(AFHTTPRequestOperation *operation, NSError *error) {
        NSLog(@"%@", error);
    }];
}

```