

## WebView

Import SDK header files

```
#import <JYouLoginKit/REDeLoginKit.h>
```

### 1. WebView opens the specified URL at a specific location and size

```
/** Opens a specified URL at a specific location
 * @param showView The view that loads the webView
 * Can be nil; if nil is passed, the keywindow will
 * be used.
 *
 * @param urlString The URL to be displayed.
 *
 * @param top Distance from the top of the screen
 *
 * @param left Distance from the left edge of the
 * screen
 *
 * @param width Web width
 *
 * @param height Web height
 *
 * @param eventBlock eventBlock Web callback to
 * native call
 *
 * @param closeBlock Close web callback.
 * `fromGame` YES: Client actively closes web; `NO`:
 * User or web client actively closes web.
 *
 * @param errorBlock Callback for URL opening
 * failure
 */
+ (void)openWebViewWithShowView:(UIView
*)showView url:(NSString *)urlString
withTop:(CGFloat)top left:(CGFloat)left
width:(CGFloat)width height:(CGFloat)height
eventBlock:(void (^)(NSString *, NSDictionary
*))eventBlock closeBlock:(void
```

```
(^)(BOOL))closeBlock errorCallback:(void (^)(NSError*))errorBlock;
```

Example Code:

```
[REDeLoginKit openWebViewWithShowView:nil
url:@"https://www.quicksdk.com/docsindex.html"
withTop:200 left:200 width:100 height:100
eventBlock:^(NSString *eventName, NSDictionary *dataDict) {

    } closeBlock :^( BOOL fromGame) {

    } errorCallback :^( NSError *error) {

    }
}];
```

## 2. Native calls methods in web pages

```
/** Call web method locally
@param jsonStr json string, example:
{"action":"loginNotify","data":{"uid":"userid","token":"usertoken"}}
*/
+ (void)nativeCallWebEvent:(NSString *)jsonStr;
```

Example Code:

```
NSDictionary * dict =
:@{@"action":@"loginNotify",@"data"::@{@"uid":@"userid", @"token":@"usertoken"}};
NSError *error = nil;
NSData *jsonData = [NSJSONSerialization
dataWithJSONObject:dict
options:NSJSONWritingPrettyPrinted error:&error];
if (error) {
    NSLog(@"dic to String
fail %ld: %@", (long)error.code,
error.localizedDescription);
    return;
}
NSString * jsonResult = [[NSString alloc]
initWithData:jsonData
encoding:NSUTF8StringEncoding];
[REDeLoginKit nativeCallWebEvent:jsonResult];
```

### 3. 网页调用 Native 方法

```
        window.webkit.messageHandlers.nativeHandler.postMessage({
            'action': 'method',
            'data': {'key': 'value'}
        });
        window.webkit.messageHandlers.nativeHandler.postMessage(JSON.stringify({
            'action': 'method',
            'data': {'key': 'value'}
        }, null, 2));
```

json 格式为 {"action":"close","data":{"key":"value"}}  
action 为 close 时，会关闭 webview 界面。

### 4. Close WebView method

When the game needs to actively close the current webView, this method can be called, and the closeBlock will be called back after the closing is completed.

```
/** Close webView */  
+ ( void )closeWebView;
```

Note: Only one webView can be opened at a time. You can only open another one after closing it.