**DAS Mobile Library(IOS) 연동 규격서**

**[LG U+ Data Service Authentication System]**

Version 1.0.0.

Date: 2018/07/11

그림1.png

Approval

Author: 소속 직급 성명 서명

Reviewer: 소속 직급 성명 서명

Approver: 소속 직급 성명 서명

Revisions

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Name | Version | Contents |
| 2018/07/11 | 우희명 | 1.0.0 | 최초 작성 |

**List of Contents**

[1 Introduction 16](#_Toc513044343)

[1.1 Scope of this documents 16](#_Toc513044344)

[1.2 Intended Audience and Reading Suggestion 16](#_Toc513044345)

[2 Overview 17](#_Toc513044346)

[2.1 기능 17](#_Toc513044347)

[3 설치방법(모앱용) 18](#_Toc513044348)

[3.1 Library 적용 18](#_Toc513044349)

[3.2 xml 파일 적용 18](#_Toc513044350)

[3.3 One ID 아이콘 적용 18](#_Toc513044351)

[3.4 계정 및 동기화 이름 적용 18](#_Toc513044352)

[3.5 AndroidManifest 설정 18](#_Toc513044353)

[3.6 SharedPreferences 등록 20](#_Toc513044354)

[3.7 사용 Library 정보 20](#_Toc513044355)

[4 설치방법(일반앱용) 21](#_Toc513044356)

[4.1 Library 적용 21](#_Toc513044357)

[4.2 xml 파일 적용 21](#_Toc513044358)

[4.3 One ID 아이콘 적용 21](#_Toc513044359)

[4.4 계정 및 동기화 이름 적용 21](#_Toc513044360)

[4.5 AndroidManifest 설정 21](#_Toc513044361)

[4.6 SharedPreferences 등록 22](#_Toc513044362)

[4.7 사용 Library 정보 23](#_Toc513044363)

[5 모바일 라이브러리 인터페이스 규격 24](#_Toc513044364)

[5.1 WebView 인터페이스 소개 24](#_Toc513044365)

[5.2 CallBack 인터페이스 소개 24](#_Toc513044366)

[5.3 Service 인터페이스 소개 24](#_Toc513044367)

[5.4 Function 인터페이스 소개 25](#_Toc513044368)

[6 인터페이스 목록 26](#_Toc513044369)

[6.1 회원 가입 요청 29](#_Toc513044370)

[6.1.1 인터페이스 소개 29](#_Toc513044371)

[6.1.2 Message Specifications 30](#_Toc513044372)

[6.1.3 Sample 30](#_Toc513044373)

[6.2 M2M 가입정보 입력 회원 가입 요청 31](#_Toc513044374)

[6.2.1 인터페이스 소개 31](#_Toc513044375)

[6.2.2 Message Specifications 31](#_Toc513044376)

[6.2.3 Sample 32](#_Toc513044377)

[6.3 회원 가입 응답 33](#_Toc513044378)

[6.3.1 인터페이스 소개 33](#_Toc513044379)

[6.3.2 Message Specifications 33](#_Toc513044380)

[6.3.3 Sample 33](#_Toc513044381)

[6.4 자동로그인 요청 34](#_Toc513044382)

[6.4.1 인터페이스 소개 34](#_Toc513044383)

[6.4.2 Message Specifications 34](#_Toc513044384)

[6.4.3 Sample 35](#_Toc513044385)

[6.5 간편로그인 요청 36](#_Toc513044386)

[6.5.1 인터페이스 소개 36](#_Toc513044387)

[6.5.2 Message Specifications 37](#_Toc513044388)

[6.5.3 Sample 37](#_Toc513044389)

[6.6 간편로그인 응답 38](#_Toc513044390)

[6.6.1 인터페이스 소개 38](#_Toc513044391)

[6.6.2 Message Specifications 38](#_Toc513044392)

[6.6.3 Sample 39](#_Toc513044393)

[6.7 계정 및 동기화 개수 조회 40](#_Toc513044394)

[6.7.1 인터페이스 소개 40](#_Toc513044395)

[6.7.2 Message Specifications 40](#_Toc513044396)

[6.7.3 Sample 40](#_Toc513044397)

[6.8 계정 및 동기화 개수 응답 41](#_Toc513044398)

[6.8.1 인터페이스 소개 41](#_Toc513044399)

[6.8.2 Message Specifications 41](#_Toc513044400)

[6.8.3 Sample 41](#_Toc513044401)

[6.9 Default 로그인 요청 42](#_Toc513044402)

[6.9.1 인터페이스 소개 42](#_Toc513044403)

[6.9.2 Message Specifications 43](#_Toc513044404)

[6.9.3 Sample 43](#_Toc513044405)

[6.10 ID/PW로그인 요청 44](#_Toc513044406)

[6.10.1 인터페이스 소개 44](#_Toc513044407)

[6.10.2 Message Specifications 45](#_Toc513044408)

[6.10.3 Sample 46](#_Toc513044409)

[6.11 약관동의 및 기존ID 전환 요청 47](#_Toc513044410)

[6.11.1 인터페이스 소개 47](#_Toc513044411)

[6.11.2 Message Specifications 47](#_Toc513044412)

[6.11.3 Sample 48](#_Toc513044413)

[6.12 WebView 로그인 응답 49](#_Toc513044414)

[6.12.1 인터페이스 소개 49](#_Toc513044415)

[6.12.2 Message Specifications 49](#_Toc513044416)

[6.12.3 Sample 50](#_Toc513044417)

[6.13 기존ID로 로그인 요청 51](#_Toc513044418)

[6.13.1 인터페이스 소개 51](#_Toc513044419)

[6.13.2 Message Specifications 51](#_Toc513044420)

[6.13.3 Sample 51](#_Toc513044421)

[6.14 One ID 관리화면 요청 52](#_Toc513044422)

[6.14.1 인터페이스 소개 52](#_Toc513044423)

[6.14.2 Message Specifications 53](#_Toc513044424)

[6.14.3 Sample 53](#_Toc513044425)

[6.15 ID 찾기 요청 54](#_Toc513044426)

[6.15.1 인터페이스 소개 54](#_Toc513044427)

[6.15.2 Message Specifications 54](#_Toc513044428)

[6.15.3 Sample 54](#_Toc513044429)

[6.16 PW 찾기 요청 55](#_Toc513044430)

[6.16.1 인터페이스 소개 55](#_Toc513044431)

[6.16.2 Message Specifications 55](#_Toc513044432)

[6.16.3 Sample 55](#_Toc513044433)

[6.17 기존 ID 찾기 요청 56](#_Toc513044434)

[6.17.1 인터페이스 소개 56](#_Toc513044435)

[6.17.2 Message Specifications 57](#_Toc513044436)

[6.17.3 Sample 57](#_Toc513044437)

[6.18 기존 PW 찾기 요청 58](#_Toc513044438)

[6.18.1 인터페이스 소개 58](#_Toc513044439)

[6.18.2 Message Specifications 59](#_Toc513044440)

[6.18.3 Sample 59](#_Toc513044441)

[6.19 전환 가입 요청 60](#_Toc513044442)

[6.19.1 인터페이스 소개 60](#_Toc513044443)

[6.19.2 Message Specifications 60](#_Toc513044444)

[6.19.3 Sample 61](#_Toc513044445)

[6.20 전환 가입 응답 62](#_Toc513044446)

[6.20.1 인터페이스 소개 62](#_Toc513044447)

[6.20.2 Message Specifications 62](#_Toc513044448)

[6.20.3 Sample 62](#_Toc513044449)

[6.21 계정 및 동기화 삭제 요청 63](#_Toc513044450)

[6.21.1 인터페이스 소개 63](#_Toc513044451)

[6.21.2 Message Specifications 63](#_Toc513044452)

[6.21.3 Sample 63](#_Toc513044453)

[6.22 계정 및 동기화 삭제 응답 64](#_Toc513044454)

[6.22.1 인터페이스 소개 64](#_Toc513044455)

[6.22.2 Message Specifications 64](#_Toc513044456)

[6.22.3 Sample 64](#_Toc513044457)

[6.23 로그아웃 65](#_Toc513044458)

[6.23.1 인터페이스 소개 65](#_Toc513044459)

[6.23.2 Message Specifications 65](#_Toc513044460)

[6.23.3 Sample 65](#_Toc513044461)

[6.24 타사가입자 자사모바일 승급 66](#_Toc513044462)

[6.24.1 인터페이스 소개 66](#_Toc513044463)

[6.24.2 Message Specifications 66](#_Toc513044464)

[6.24.3 Sample 66](#_Toc513044465)

[6.25 UFLIX/HDTV 이용정보 조회 요청 67](#_Toc513044466)

[6.25.1 인터페이스 소개 67](#_Toc513044467)

[6.25.2 Message Specifications 67](#_Toc513044468)

[6.25.3 Sample 67](#_Toc513044469)

[6.26 페이지 종료 알림 68](#_Toc513044470)

[6.26.1 인터페이스 소개 68](#_Toc513044471)

[6.26.2 Message Specifications 68](#_Toc513044472)

[6.26.3 Sample 68](#_Toc513044473)

[~~6.27~~ ~~SNS ID 자동로그인 요청~~ 69](#_Toc513044474)

[~~6.27.1~~ ~~인터페이스 소개~~ 69](#_Toc513044475)

[~~6.27.2~~ ~~Message Specifications~~ 69](#_Toc513044476)

[~~6.27.3~~ ~~Sample~~ 69](#_Toc513044477)

[~~6.28~~ ~~SNS ID 로그인 응답~~ 70](#_Toc513044478)

[~~6.28.1~~ ~~인터페이스 소개~~ 70](#_Toc513044479)

[~~6.28.2~~ ~~Message Specifications~~ 70](#_Toc513044480)

[~~6.28.3~~ ~~Sample~~ 71](#_Toc513044481)

[~~6.29~~ ~~SNS ID 로그아웃 요청~~ 72](#_Toc513044482)

[~~6.29.1~~ ~~인터페이스 소개~~ 72](#_Toc513044483)

[~~6.29.2~~ ~~Message Specifications~~ 72](#_Toc513044484)

[~~6.29.3~~ ~~Sample~~ 72](#_Toc513044485)

[~~6.30~~ ~~SNS ID 로그아웃 응답~~ 73](#_Toc513044486)

[~~6.30.1~~ ~~인터페이스 소개~~ 73](#_Toc513044487)

[~~6.30.2~~ ~~Message Specifications~~ 73](#_Toc513044488)

[~~6.30.3~~ ~~Sample~~ 73](#_Toc513044489)

[~~6.31~~ ~~SNS ID 해지 요청~~ 74](#_Toc513044490)

[~~6.31.1~~ ~~인터페이스 소개~~ 74](#_Toc513044491)

[~~6.31.2~~ ~~Message Specifications~~ 74](#_Toc513044492)

[~~6.31.3~~ ~~Sample~~ 74](#_Toc513044493)

[~~6.32~~ ~~SNS ID 해지 응답~~ 75](#_Toc513044494)

[~~6.32.1~~ ~~인터페이스 소개~~ 75](#_Toc513044495)

[~~6.32.2~~ ~~Message Specifications~~ 75](#_Toc513044496)

[~~6.32.3~~ ~~Sample~~ 75](#_Toc513044497)

[~~6.33~~ ~~SNS ID 전환가입 요청~~ 76](#_Toc513044498)

[~~6.33.1~~ ~~인터페이스 소개~~ 76](#_Toc513044499)

[~~6.33.2~~ ~~Message Specifications~~ 76](#_Toc513044500)

[~~6.33.3~~ ~~Sample~~ 76](#_Toc513044501)

[~~6.34~~ ~~SNS ID 전환가입 응답~~ 77](#_Toc513044502)

[~~6.34.1~~ ~~인터페이스 소개~~ 77](#_Toc513044503)

[~~6.34.2~~ ~~Message Specifications~~ 77](#_Toc513044504)

[~~6.34.3~~ ~~Sample~~ 77](#_Toc513044505)

[~~6.35~~ ~~기 가입 One ID 사용 선택 응답~~ 78](#_Toc513044506)

[~~6.35.1~~ ~~인터페이스 소개~~ 78](#_Toc513044507)

[~~6.35.2~~ ~~Message Specifications~~ 78](#_Toc513044508)

[~~6.35.3~~ ~~Sample~~ 78](#_Toc513044509)

[6.36 간편인증 ONE ID의 정회원 전환 요청 79](#_Toc513044510)

[6.36.1 인터페이스 소개 79](#_Toc513044511)

[6.36.2 Message Specifications 79](#_Toc513044512)

[6.36.3 Sample 80](#_Toc513044513)

[6.37 간편인증 ONE ID의 정회원 전환 응답 81](#_Toc513044514)

[6.37.1 인터페이스 소개 81](#_Toc513044515)

[6.37.2 Message Specifications 81](#_Toc513044516)

[6.37.3 Sample 82](#_Toc513044517)

[6.38 서비스 ID/PW 로그인 화면 요청 Callback 83](#_Toc513044518)

[6.38.1 인터페이스 소개 83](#_Toc513044519)

[6.38.2 Message Specifications 83](#_Toc513044520)

[6.38.3 Sample 83](#_Toc513044521)

[6.39 휴대폰 로그인 요청 (IoT, 가족지킴이 전용) (CTN 인증 로그인 요청) 84](#_Toc513044522)

[6.39.1 인터페이스 소개 84](#_Toc513044523)

[6.39.2 Message Specifications 85](#_Toc513044524)

[6.39.3 Sample 85](#_Toc513044525)

[~~6.40~~ ~~ONE ID 변경 요청~~ 86](#_Toc513044526)

[~~6.40.1~~ ~~인터페이스 소개~~ 86](#_Toc513044527)

[~~6.40.2~~ ~~Message Specifications~~ 86](#_Toc513044528)

[~~6.40.3~~ ~~Sample~~ 86](#_Toc513044529)

[~~6.41~~ ~~ONE ID 변경 응답~~ 87](#_Toc513044530)

[~~6.41.1~~ ~~인터페이스 소개~~ 87](#_Toc513044531)

[~~6.41.2~~ ~~Message Specifications~~ 87](#_Toc513044532)

[~~6.41.3~~ ~~Sample~~ 88](#_Toc513044533)

[6.42 이벤트 알림 Callback 89](#_Toc513044534)

[6.42.1 인터페이스 소개 89](#_Toc513044535)

[6.42.2 Message Specifications 89](#_Toc513044536)

[6.42.3 Sample 90](#_Toc513044537)

[6.43 휴대폰 로그인 응답 (명칭변경 구. CTN 인증 로그인 응답) 91](#_Toc513044538)

[6.43.1 인터페이스 소개 91](#_Toc513044539)

[6.43.2 Message Specifications 91](#_Toc513044540)

[6.43.3 Sample 92](#_Toc513044541)

[6.44 임시ID One ID 변경 요청 93](#_Toc513044542)

[6.44.1 인터페이스 소개 93](#_Toc513044543)

[6.44.2 Message Specifications 93](#_Toc513044544)

[6.44.3 Sample 93](#_Toc513044545)

[6.45 임시ID ONE ID 변경 응답 94](#_Toc513044546)

[6.45.1 인터페이스 소개 94](#_Toc513044547)

[6.45.2 Message Specifications 94](#_Toc513044548)

[6.45.3 Sample 95](#_Toc513044549)

[6.46 권한 오류 알림 96](#_Toc513044550)

[6.46.1 인터페이스 소개 96](#_Toc513044551)

[6.46.2 Message Specifications 96](#_Toc513044552)

[6.46.3 Sample 96](#_Toc513044553)

[6.47 휴대폰 번호 로그인 요청 97](#_Toc513044554)

[6.47.1 인터페이스 소개 97](#_Toc513044555)

[6.47.2 Message Specifications 98](#_Toc513044556)

[6.47.3 Sample 99](#_Toc513044557)

[6.48 FIDO 추가인증 요청 101](#_Toc513044558)

[6.48.1 인터페이스 소개 101](#_Toc513044559)

[6.48.2 Message Specifications 101](#_Toc513044560)

[6.48.3 Sample 101](#_Toc513044561)

[7 Appendix A(결과코드표) 102](#_Toc513044562)

[8 Appendix B(결과코드에 따른 서비스 App 후처리) 103](#_Toc513044563)

[9 Appendix C(비밀번호 암호화 소스) 105](#_Toc513044564)

# Introduction

## Scope of this documents

이 문서는 DAS의 IOS Mobile Library 구성과 사용법에 대해 설명하고 있다.

## Intended Audience and Reading Suggestion

이 문서는 IOS Mobile Library를 통해 Data Service Authentication Service(이하 DAS) 시스템과의 연동을 하는 규약에 대한 설명을 하기 위해 작성된 문서입니다.

# Overview

## 기능

다양한 LG U+ 내부의 IOS Service App들의 로그인 계정을 One ID 로 통합하고 그것을 이용하여 기존에 사용하던 App 기능들을 그대로 사용 할 수 있도록 Mobile Library를 제공한다.

통합된 One ID 계정을 사용하기 위한 공통 UI를 제공하며 IOS 의 계정 및 동기화 기능을 활용하여 각 Service App 들 간에 간편하게 로그인을 할 수 있도록 제공한다.

# 설치방법(일반앱용)

## Library 적용

DasFramework.framework 파일을 프로젝트의 Build Phases > Link Binary With Libraries에 추가한다.

그 후 DasFramework.framework을 Build Phases > Embedded Framework 에 추가한다.

## xml 파일 적용

사용안함

## One ID 아이콘 적용

사용안함

## 계정 및 동기화 이름 적용

사용안함

## Info.plist 설정

프로젝트의 Info.plist 파일에 아래 내용을 추가한다.

|  |  |
| --- | --- |
| AndroidManifest.xml  (일반앱용) | <key>NSAppTransportSecurity</key>  <dict>  <key>NSAllowsArbitraryLoads</key>  <true/>  </dict> |

## NSUserDefaults 등록

SNS ID 사용여부, 아이핀 인증 사용여부, KMC인증 사용여부를 설정하기 위해

프로젝트의 viewDidLoad 의 vieDidLoad 함수에 아래 내용을 추가한다.

NSUserDefaults 를 등록한다.

* 타사 모바일 고객 SNS ID 사용여부 : OT\_SNS\_YN
* 자사 모바일 고객 SNS ID 사용여부 : LGT\_SNS\_YN
* 아이핀 사용여부 : IPIN\_YN (‘Y’ : 사용, ‘N’ : 미사용)

참고) 미설정 하였을 경우

자사/타사 모바일 고객 SNS ID : 미사용

아이핀 : 사용으로 설정됨

|  |  |
| --- | --- |
| VieController클래스의 viewDidLoad 함수  (모앱용) | NSUserDefaults\* defaults = [NSUserDefaults standardUserDefaults];  NSMutableDictionary\* dictionary = [NSMutableDictionary dictionary];  @try  {  if([defaults dictionaryForKey:AccountUtil.DAS\_LIBRARY] != nil) {  dictionary = [[defaults dictionaryForKey:@“DAS\_LIBRARY”] mutableCopy];  }  [dictionary setObject:@”Y” forKey:@"OT\_SNS\_YN"];  [dictionary setObject:@”N” forKey:@"LGT\_SNS\_YN"];  [dictionary setObject:@”Y” forKey:@"IPIN\_YN"];  [defaults setObject:dictionary forKey:AccountUtil.DAS\_LIBRARY];  if ( ![defaults synchronize] ) {  return false;  }  }  @catch (NSException \*exception)  {  NSLog(@"Caught %@%@", exception.name, exception.reason);  return false;  } |

## 사용 Library 정보

DasFramework.framework

# 모바일 라이브러리 인터페이스 규격

## CallBack 인터페이스 소개

1. 개요 : requestCode, resultCode, data 변수의 선언을 통해 WebView Activity의 결과 값을 받아온다.

2. 설명 : Library 내부에 있는 WebView Activity의 결과값을 호출된 Activity에게 CallBack으로 전달한다.

3 샘플 :

|  |  |
| --- | --- |
| Source | @interface MainActivity : UIViewController <UITableViewDelegate, UITableViewDataSource>  {  NSMutableDictionary \*data;  **int** requestCode; **int** resultCode;  }  @property (nonatomic, assign) **int** requestCode; @property (nonatomic, assign) **int** resultCode; @property (nonatomic, strong) NSMutableDictionary \*data;  @end  @implementation MainActivity  - (void)viewDidAppear:(BOOL)animated  {  if(resultCode == 0) {  return; } if(data != nil && requestCode == 1) {  if([data[@"RESULT"] isEqualToString:RC\_USER\_SIGNUP])  { NSString \*tmpRT = data[@"RT"]; NSString \*tmpRT\_MSG = data[@"RT\_MSG"]; NSString \*tmpUSER\_ID = data[@"USER\_ID"]; NSString \*tmpCLOSE\_ST = data[@"CLOSE\_ST"];  }  }  }  @end |

## WebView 인터페이스 소개

1. 개요 : NSMutableDictionary를 통해 Library 내부에 있는 WebView Activity를 호출한다.

2. 설명 : setObject 함수를 이용하여 파라메타를 전달한다.

3 샘플 :

|  |  |
| --- | --- |
| Source | NSMutableDictionary \*intent = [NSMutableDictionary dictionary]; [intent setObject: FUNCTION\_USER\_SIGNUP forKey:@"FUNCTION\_NAME"]; [intent setObject:SERVICE\_CD forKey:@"SERVICE\_CD"];  [Activity startActivityForResult:self class:[HybridWebView new] intent:intent requestCode:1]; |

## Function 인터페이스 소개

1. 개요 : DAS Mobile Library 내부에 있는 함수를 직접 호출한다.

2. 설명 : Library 내부에 있는 함수를 직접 호출하여 parameter를 전달하고 result값을 받는다.

3 샘플 :

|  |  |
| --- | --- |
| Source | OrderedDictionary  \*autoMap = [IntegrationService.getInstance autoLogin:SERVICE\_CD USER\_ID:@"USER\_ID" SSO\_KEY:@"SSO\_KEY"] |

# 인터페이스 목록

| **인터페이스 항목** | **호출방식** | **설명** |
| --- | --- | --- |
| 회원 가입 요청 | WebView | One ID 회원 신규가입 요청을 한다. |
| 가입정보 입력 회원 가입 요청 | WebView | One ID 회원이 직접 입력하는 신규가입 요청을 한다. |
| 회원 가입 응답 | Callback | One ID 회원 신규가입 요청에 대한 응답을 받는다. |
| 자동로그인 요청 | Function | One ID 회원 자동 로그인 요청을 한다. |
| 간편로그인 응답 | Callback | One ID 회원 간편 로그인에 대한 응답을 받는다. |
| 계정 및 동기화 개수 응답 | Callback | 계정 및 동기화에 등록된 One ID의 개수에 대한 응답을 받는다. |
| Default 로그인 요청 | WebView | 계정 및 동기화에 등록된 One ID 리스트를 이용하여 로그인 요청을 한다. |
| ID/PW로그인 요청 | WebView | One ID 회원 ID/PW 로그인 요청을 한다. |
| 약관동의 및 기존ID 전환 요청 | WebView | 로그인 응답이 약관 미동의 시 약관동의 화면 요청을 한다. |
| WebView 로그인 응답 | Callback | WebView에서 요청된 로그인 결과를 서비스 App에 알려준다. |
| 기존ID로 로그인 요청 | Callback | One ID 로그인 화면에서 '기존ID로 로그인'을 선택했을 경우 서비스 App에 알려준다. |
| One ID 관리화면 요청 | WebView | One ID 관리화면 요청을 한다. |
| ID 찾기 요청 | WebView | One ID 회원 ID 찾기 요청을 한다. |
| PW 찾기 요청 | WebView | One ID 회원 PW 찾기 요청을 한다. |
| 기존 ID 찾기 요청 | Callback | 기존 ID 회원 ID 찾기 요청을 한다. |
| 기존 PW 찾기 요청 | Callback | 기존 ID 회원 PW 찾기 요청을 한다. |
| 전환 가입 요청 | WebView | 기존 서비스의 ID 회원을 One ID 회원으로 전환 요청을 한다. |
| 전환 가입 응답 | Callback | 기존 서비스의 ID 회원을 One ID 회원으로 전환한 결과를 서비스 App에 알려준다. |
| 계정 및 동기화 삭제 요청 | Service | One ID에 해당하는 계정 및 동기화 정보 삭제 요청을 한다. |
| 계정 및 동기화 삭제 응답 | Callback | One ID에 해당하는 계정 및 동기화 정보 삭제에 대한 응답을 받는다. |
| 로그아웃 | Callback | 로그아웃 처리를 한다. |
| 타사가입자 자사모바일 승급 | WebView | 타사가입자를 자사모바일 가입자로 승급시킨다. |
| UFLIX/HDTV 이용정보 조회 요청 | WebView | One ID 회원 UFLIX/HDTV 상품 이용정보 조회 요청을 한다. |
| 페이지 종료 알림 | Callback | 페이지 우측 상단의 X 버튼을 클릭하여 종료 시 알려준다. |
| SNS ID 자동로그인 요청 | WebView | 서비스 앱이 보관 하고 있는 SNS ID 계정 정보를 이용하여 로그인 요청을 한다. |
| SNS ID 로그인 응답 | Callback | 라이브러리에 포함된 WebView Activity 의 SNS ID 로그인 결과를 서비스 App에 알린다. |
| SNS ID 로그아웃 요청 | WebView | 서비스에서 SNS ID 로그아웃 시 SNS(Facebook, Google+) 로그아웃 처리를 위해 요청한다. |
| SNS ID 로그아웃 응답 | Callback | 로그인 되어있는 SNS ID의 로그아웃 처리 결과를 서비스 App에 알린다. |
| SNS ID 해지 요청 | WebView | SNS ID 회원에 대한 서비스 해지/탈퇴 요청을 한다. |
| SNS ID 해지 응답 | Callback | SNS ID 회원에 대한 서비스 해지/탈퇴 처리 결과를 서비스 App에 알린다. |
| SNS ID 전환 가입 요청 | WebView | SNS ID 회원을 One ID 회원으로 전환 요청 한다. |
| SNS ID 전환 가입 응답 | Callback | SNS ID 회원을 One ID 회원으로 전환한 결과를 서비스 App에 알린다. |
| 기 가입 One ID 사용 선택 응답 | Callback | SNS ID 가입 시 email 정보와 동일한 기존 OneID 사용을 선택할 경우 서비스 앱에 전달. |
| 간편인증 ONE ID의 정회원 전환 요청 | WebView | 간편인증 One ID의 정회원전환을 위한 요청 |
| 간편인증 ONE ID의 정회원 전환 응답 | Callback | 간편인증 OneID의 정회원 전환 결과를 서비스 App에 알려준다. |
| 서비스 ID/PW 로그인 화면 요청 Callback | Callback | 서비스앱에서 OneID ID/PW 화면을 띄우게 하기 위해 DAS Lib가 요청함. |
| CTN 인증 로그인 요청 | WebView | CTN 인증을 통한 로그인 요청 |
| ONE ID 변경 요청 | WebView | ONE ID 변경 요청 |
| ONE ID 변경 응답 | Callback | ONE ID 변경 요답 |
| 이벤트 알림 Callback | Callback | DAS 화면에서 이벤트 발생을 알리기 위해 DAS Lib.가 요청 |
| CTN 인증 로그인 응답 | Callback | WebView에서 요청된 CTN 인증 로그인 결과를 서비스 App에 알려준다. |
| 휴대폰 로그인 응답 | Callback | WebView에서 요청된 CTN 인증 로그인 결과를 서비스 App에 알려준다. |
| 임시ID One ID 변경 요청 | WebView | 임시ID에 대한 One ID로의 변경 요청 |
| 임시ID One ID 변경 응답 | Callback | 임시ID를 ONE ID로 변경한 결과를 서비스 App에 알려준다. |
| 권한 오류 알림 | Callback | DAS Lib. 필수 권한 해지 시 오류 반환 |
| 휴대폰 번호 로그인 요청 | WebView | 휴대폰 번호 인증을 통한 로그인 요청 |

## 회원 가입 요청

### 인터페이스 소개

1. 개요 : One ID 회원 신규가입 요청을 한다.

2. 설명 : One ID 사용을 위한 동의절차 및 회원정보 입력을 요청을 한다.

DAS Lib. 사용 시 단말의 USIM 정보를 읽어 자사 또는 타사 OneID로 가입시킨다.

|  |
| --- |
| 용어설명 |
| \*자사OneID : OneID에 LG U+ 모바일 가입정보 맵핑. (ID\_TYPE : T00, LGT\_TYPE : Y)  \*타사OneID : 본인인증을 통한 OneID. 자사 모바일 가입정보 없음. (ID\_TYPE : T00, LGT\_TYPE : N)  \*간편인증OneID : email 인증 or SMS 인증을 통해 가입한 준회원개념의 ID. 고객정보 없음. (ID\_TYPE : T01 or T02)  \*임시ID : 고객에게 노출되지 않는 OneID. 제휴사 가입 또는 휴대폰 로그인 시 자동 생성됨. (ID\_TYPE : T00, LGT\_TYPE : N, VTID\_YN : Y) |

서비스앱에서 OneID 회원가입 기능 구현 방법

-. [6.1 회원 가입 요청] 사용 : 서비스앱 내에서 [OneID 회원가입] 버튼을 직접 구현할 경우 호출.

-. [6.10 ID/PW로그인 요청]을 통한 구현 : DAS의 ID/PW 로그인 화면 내에 회원가입 버튼이 노출됨.

|  |
| --- |
|  |
| [6.10 ID/PW로그인 요청]에 의한 DAS 로그인 화면 |

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | WebView |
| Request URI | FUNCTION\_USER\_SIGNUP |

### Message Specifications

#### Request

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
| SERVICE\_CD | String | 30 | O | One ID에서 부여한 서비스 코드 |
| SERVICE\_LOGIN\_TYPE | String | 4 | O | 서비스에서 사용할 로그인 타입   |  |  |  |  | | --- | --- | --- | --- | | OneID 용 | | 기존ID 용 | | | 자동로그인 | 간편로그인 | 자동로그인 | 간편로그인 | | 0 or 1 | 0 or 1 | 0 or 1 | 0 or 1 |   0:미사용, 1:사용  ex) 1100 (OneID 자동,간편사용, 기존ID 자동,간편미사용) |

### Sample

|  |  |
| --- | --- |
| Source | NSMutableDictionary \*intent = [NSMutableDictionary dictionary]; [intent setObject:FUNCTION\_USER\_SIGNUP forKey:@"FUNCTION\_NAME"]; [intent setObject:SERVICE\_CD forKey:@"SERVICE\_CD"]; [intent setObject:@"1100" forKey:@"SERVICE\_LOGIN\_TYPE"]; [Activity startActivityForResult:self class:[HybridWebView new] intent:intent requestCode:1]; |

## M2M 가입정보 입력 회원 가입 요청

### 인터페이스 소개

1. 개요 : One ID 회원이 직접 입력하는 신규가입 요청을 한다.

2. 설명 : M2M(012) 단말에서 자사 OneID 회원가입을 위해 요청한다.

[6.1 회원 가입 요청]과의 차이 : M2M단말(012번호)에 대해 단말정보(USIM, ICCID 등)를 단말에서 읽지 않고, 가입정보를 고객이 직접 입력하게 한다.

|  |
| --- |
|  |
| [6.2 M2M 가입정보 입력 회원 가입 요청] 시 최초 인증화면 |

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | WebView |
| Request URI | FUNCTION\_USER\_MANUAL\_SIGNUP |

### Message Specifications

#### Request

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
| SERVICE\_CD | String | 30 | O | One ID에서 부여한 서비스 코드 |
| SERVICE\_LOGIN\_TYPE | String | 4 | O | 서비스에서 사용할 로그인 타입   |  |  |  |  | | --- | --- | --- | --- | | OneID 용 | | 기존ID 용 | | | 자동로그인 | 간편로그인 | 자동로그인 | 간편로그인 | | 0 or 1 | 0 or 1 | 0 or 1 | 0 or 1 |   0:미사용, 1:사용  ex) 1100 (OneID 자동,간편사용, 기존ID 자동,간편미사용) |

### Sample

|  |  |
| --- | --- |
| Source | NSMutableDictionary \*intent = [NSMutableDictionary dictionary];  [intent setObject:FUNCTION\_USER\_MANUAL\_SIGNUP forKey:@"FUNCTION\_NAME"];  [intent setObject:SERVICE\_CD forKey:@"SERVICE\_CD"];  [intent setObject:@"1100" forKey:@"SERVICE\_LOGIN\_TYPE"];  [Activity startActivityForResult:self class:[HybridWebView new] intent:intent requestCode:1]; |

## 회원 가입 응답

### 인터페이스 소개

1. 개요 : One ID 회원 신규가입 요청에 대한 응답을 받는다.

2. 설명 : One ID 회원 신규가입에 대한 응답을 서비스앱에 전달한다.

OneID 회원가입이 완료되었을 경우의 응답. 성공응답을 받은 경우 서비스별 시나리오에 따라 다음 동작을 구현하면 됨. (ex. 회원가입 완료 후 로그인 요청)

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | Callback |
| Request URI | RC\_USER\_SIGNUP |

### Message Specifications

#### Response

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
| RT | String | 20 | O | 결과 코드([[Appendix A(결과 코드표)]](#_Appendix_A(결과코드)) 참조) |
| RT\_MSG | String | 500 | O | 결과 메시지([[Appendix A(결과 코드표)]](#_Appendix_A(결과코드)) 참조) |
| USER\_ID | String | 50 | O | 회원아이디(e-mail타입으로 @ 앞은 ‘-‘, ‘\_’, ‘.’ 특수문자 허용) |
| CLOSE\_ST | String | 1 | O | 0:확인버튼, 1:X버튼 |

### Sample

|  |  |
| --- | --- |
| Source | - (void)viewDidAppear:(BOOL)animated  {  if([data[@"RESULT"] isEqualToString:RC\_USER\_SIGNUP])  {  NSString \*tmpRT = data[@"RT"];  NSString \*tmpRT\_MSG = data[@"RT\_MSG"];  NSString \*tmpUSER\_ID = data[@"USER\_ID"];  NSString \*tmpCLOSE\_ST = data[@"CLOSE\_ST"] == nil?@"":data[@"CLOSE\_ST"];  }  } |

## 자동로그인 요청

### 인터페이스 소개

1. 개요 : One ID 회원 자동 로그인 요청을 한다.

2. 설명 : 서비스 앱이 보관 하고 있는 계정 정보를 이용하여 로그인 요청을 한다.

로그인 응답([6.6 간편로그인 응답], [6.12 WebView 로그인 응답], [6.43 휴대폰 로그인 응답] )의 LOGIN\_TYPE값 이 ‘1’인 경우는 고객이 자동로그인 기능을 사용하도록 설정한 것임.

이 후 서비스에서 자동로그인 기능을 구현해야 함.

자동로그인 구현 시 인증방법

-. [6.4 자동로그인 요청]을 하는 방법 : 로그인 응답에서 전달받은 SSO\_KEY를 서비스에서 저장한 후, 자동로그인이 필요한 시점에 [6.4 자동로그인 요청]을 통해 인증. 인증결과는 [6.6 간편로그인 응답]으로 전달됨.

-. Open API를 통해 인증하는 방법 : “기타\_ET05\_LGU\_DAS\_AND-03\_연동규격서\_v1\_0(OpenAPI)\_VXX” 규격의 [5.3 SSO 확인 (DasSsoConfirm)] 을 사용하여 인증. (Server-To-Server 방식)

DAS로부터 SSO Key 만료 되거나 찾을 수 없다는 응답을 받았을 경우, 로그인 요청을 구현하도록 권고함.

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | Function |
| Request URI | [IntegrationService.getInstance autoLogin] |

### Message Specifications

#### Request

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
| SERVICE\_CD | String | 30 | O | One ID에서 부여한 서비스 코드 |
| USER\_ID | String | 50 | O | 회원아이디(e-mail타입으로 @ 앞은 ‘-‘, ‘\_’, ‘.’ 특수문자 허용) |
| SSO\_KEY | String | 100 | O | 통합 로그인 키 |

#### Response

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
| RT | String | 20 | O | 결과 코드([[Appendix A(결과 코드표)]](#_Appendix_A(결과코드)) 참조) |
| RT\_MSG | String | 500 | O | 결과 메시지([[Appendix A(결과 코드표)]](#_Appendix_A(결과코드)) 참조) |
| ONEID\_KEY | String | 13 | O | 회원일련번호(SEQ) |
| SERVICE\_KEY | String | 13 |  | U+ Box등 서비스의 고객 Unique ID 값 |
| NAME | String | 180 | O | 회원이름 |
| LGT\_TYPE | String | 1 | O | 자/타사 구분(Y:자사, N:타사) |
| PW\_UPDATE\_DT | String | 8 | O | YYYYMMDD |
| ID\_TYPE | String | 3 |  | ID구분 (NULL or T00: ONE ID, T01: 이메일 인증가입 ID, T02: 본인정보 입력가입 ID, …) |
| **VTID\_YN** | **String** | **1** |  | **임시ID 여부 (Y: 임시ID, N or NULL : ONE ID)** |
| **VTID\_RQST\_RSN\_CD** | **String** | **3** |  | **임시ID 신청 사유코드** |

### Sample

|  |  |
| --- | --- |
| Source | OrderedDictionary \*autoMap = [IntegrationService.getInstance autoLogin:SERVICE\_CD USER\_ID:dictionary[@"USER\_ID"] SSO\_KEY:dictionary[@"SSO\_KEY"]]; |

## 간편로그인 요청

### 인터페이스 소개

1. 개요 : One ID 회원 간편 로그인 요청을 한다.

2. 설명 : 서비스 앱이 보관 하고 있는 One ID를 이용하여 로그인 요청을 한다.

DAS의 Single-Sign-On 기능을 사용하는 경우에 사용 가능.

\* Single-Sign-On 기능 : OneID를 사용하는 서비스앱에서 로그인한 정보를 안드로이드 계정에 저장하여, 이 후 다른 서비스앱에서 간편하게 로그인 할 수 있는 기능. ([6.9 Default 로그인 요청] 참고)

|  |
| --- |
|  |
| 안드로이드 계정에 저장된 OneID 계정 정보 |

로그인 응답([6.6 간편로그인 응답], [6.12 WebView 로그인 응답], [6.43 휴대폰 로그인 응답] )의 LOGIN\_TYPE값 이 ‘2’인 경우 간편로그인 요청이 가능함. (고객이 자동로그인 사용체크하고, ‘계정및동기화’ 사용이 가능할 경우의 응답값)

[6.5 간편로그인 요청]을 하는 방법 : 로그인 응답에서 전달받은 USER\_ID를 서비스에서 저장한 후, 자동로그인이 필요한 시점에 인증요청. DAS Lib.는 계정및동기화의 ID에 맵핑된 SSO Key에 대해 인증함.

인증결과는 [6.6 간편로그인 응답]으로 전달됨.

고객 입장에서는 모두 자동로그인 기능이지만 이 경우 공통의 SSOKey를 사용하여 인증하는 차이가 있음.

DAS로부터 SSO Key 만료 되거나 찾을 수 없다는 응답을 받았을 경우, 로그인 요청을 구현하도록 권고함.

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | Service |
| Request URI | IntegrationConstants.FUNCTION\_EASY\_LOGIN |

### Message Specifications

#### Request

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
| SERVICE\_CD | String | 30 | O | One ID에서 부여한 서비스 코드 |
| USER\_ID | String | 50 | O | 회원아이디(e-mail타입으로 @ 앞은 ‘-‘, ‘\_’, ‘.’ 특수문자 허용) |

### Sample

|  |  |
| --- | --- |
| Source | Intent intent = new Intent("com.lg.das.intent.service");  intent.putExtra("FUNCTION\_NAME", IntegrationConstants.FUNCTION\_EASY\_LOGIN);  intent.putExtra("SERVICE\_CD", AccountUtil.SERVICE\_CD);  intent.putExtra("USER\_ID", tmpEasyID);  startIntent(intent); |

## 간편로그인 응답

### 인터페이스 소개

1. 개요 : One ID 회원 간편 로그인에 대한 응답을 받는다.

2. 설명 : One ID 로그인 결과를 반환한다.

[6.4 자동로그인 요청], [6.5 간편로그인 요청]에 대한 응답.

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | Callback |
| Request URI | RC\_EASY\_LOGIN |

### Message Specifications

#### Response

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
| RT | String | 20 | O | 결과 코드([[Appendix A(결과 코드표)]](#_Appendix_A(결과코드)) 참조) |
| RT\_MSG | String | 500 | O | 결과 메시지([[Appendix A(결과 코드표)]](#_Appendix_A(결과코드)) 참조) |
| USER\_ID | String | 50 | O | 회원아이디(e-mail타입으로 @ 앞은 ‘-‘, ‘\_’, ‘.’ 특수문자 허용) |
| SSO\_KEY | String | 100 | O | 통합 로그인 키 |
| ONEID\_KEY | String | 13 | O | 회원일련번호(SEQ) |
| SERVICE\_KEY | String | 13 | O | U+ Box등 서비스의 고객 Unique ID 값 |
| NAME | String | 180 | O | 회원이름 |
| LGT\_TYPE | String | 1 | O | 자/타사 구분(Y:자사, N:타사) |
| PW\_UPDATE\_DT | String | 8 | O | YYYYMMDD |
| ID\_TYPE | String | 3 |  | ID구분 (NULL or T00: ONE ID, T01: 이메일 인증가입 ID, T02: 본인정보 입력가입 ID, …) |

### Sample

|  |  |
| --- | --- |
| Source | - (void)viewDidAppear:(BOOL)animated  {  if([data[@"RESULT"] isEqualToString:RC\_EASY\_LOGIN])  {  NSString \*tmpRT = data[@"RT"];  NSString \*tmpRT\_MSG = data[@"RT\_MSG"];  NSString \*tmpLOGIN\_TYPE = data[@"LOGIN\_TYPE"];  NSString \*tmpUSER\_ID = data[@"USER\_ID"];  NSString \*tmpSSO\_KEY = data[@"SSO\_KEY"];  NSString \*tmpONEID\_KEY = data[@"ONEID\_KEY"];  NSString \*tmpSERVICE\_KEY = data[@"SERVICE\_KEY"];  NSString \*tmpNAME = data[@"NAME"];  NSString \*tmpLGT\_TYPE = data[@"LGT\_TYPE"];  NSString \*tmpPW\_UPDATE\_DT = data[@"PW\_UPDATE\_DT"];  NSString \*tmpID\_TYPE = data[@"ID\_TYPE"];  }  } |

## 계정 및 동기화 개수 조회

### 인터페이스 소개

1. 개요 : 계정 및 동기화에 등록된 One ID 의 개수를 요청한다.

2. 설명 : 계정 및 동기화에 등록된 One ID 의 존재 여부를 확인하여 Default 로그인 요청여부 결정

DAS Single-Sign-On 기능을 사용할 경우 사용.

이 요청으로 안드로이드 ‘계정 및 동기화’에 등록된 OneID 개수를 확인.

응답은 [6.8 계정 및 동기화 개수 응답]으로 전달.

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | Service |
| Request URI | IntegrationConstants.FUNCTION\_COUNT\_ACCOUNT |

### Message Specifications

#### Request

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

### Sample

|  |  |
| --- | --- |
| Source | Intent intent = new Intent("com.lg.das.intent.service");  intent.putExtra("FUNCTION\_NAME", IntegrationConstants.FUNCTION\_COUNT\_ACCOUNT);  startIntent(intent); |

## 계정 및 동기화 개수 응답

### 인터페이스 소개

1. 개요 : 계정 및 동기화에 등록된 One ID 의 개수에 대한 응답을 받는다.

2. 설명 : 계정 및 동기화에 등록된 One ID 의 존재 여부를 확인하여 Default 로그인 요청여부 결정

DAS Single-Sign-On 기능을 사용할 경우 사용.

[6.7 계정 미 동기화 개수 조회]에 대한 응답임.

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | Callback |
| Request URI | IntegrationConstants.RC\_COUNT\_ACCOUNT |

### Message Specifications

#### Response

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
| RT | String | 20 | O | 결과 코드([[Appendix A(결과 코드표)]](#_Appendix_A(결과코드)) 참조) |
| RT\_MSG | String | 500 | O | 결과 메시지([[Appendix A(결과 코드표)]](#_Appendix_A(결과코드)) 참조) |
| COUNT | String | 6 | O | 계정 및 동기화에 등록된 One ID 의 개수 |

### Sample

|  |  |
| --- | --- |
| Source | Override  protected void onActivityResult(int requestCode, int resultCode, Intent data) {  super.onActivityResult(requestCode, resultCode, data);  if(data.getStringExtra("RESULT").equals(IntegrationConstants.RC\_COUNT\_ACCOUNT)) {  String tmpRT = data.getStringExtra("RT");  String tmpRT\_MSG = data.getStringExtra("RT\_MSG");  String tmpCOUNT = data.getStringExtra("COUNT");  }  } |

## Default 로그인 요청

### 인터페이스 소개

1. 개요 : 계정 및 동기화에 등록된 One ID 리스트를 이용하여 로그인 요청을 한다.

2. 설명 : 계정 및 동기화에 등록된 One ID가 없으면 로그인 실패를 호출한다. 단, DAS 서버로부터 SSO Key 만료 되거나 찾을 수 없다는 응답을 받았을 경우 단말의 계정 및 동기화에서 계정을 삭제한다.

DAS Single-Sign-On 기능을 사용할 경우 사용.

먼저 [6.7 계정 및 동기화 개수 조회]로 ‘계정 및 동기화’에 OneID가 존재를 확인한 후 요청하도록 할 것.

다음의 화면으로 로그인이 가능하게 함.

OneID 선택 후의 로그인 응답은 [6.12 WebView 로그인 응답]으로 전달.

|  |
| --- |
|  |
| Default 로그인 요청 시 계정정보 존재할 경우의 로그인화면 |

Default 로그인 요청 시 에러코드 06002(계정 및 동기화에 계정정보가 존재하지 않습니다)를 전달받는 다면, 로그인 요청을 수행하도록 할 것.

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | WebView |
| Request URI | FUNCTION\_DEFAULT\_LOGIN |

### Message Specifications

#### Request

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
| SERVICE\_CD | String | 30 | O | One ID에서 부여한 서비스 코드 |
| SERVICE\_LOGIN\_TYPE | String | 4 | O | 서비스에서 사용할 로그인 타입   |  |  |  |  | | --- | --- | --- | --- | | OneID 용 | | 기존ID 용 | | | 자동로그인 | 간편로그인 | 자동로그인 | 간편로그인 | | 0 or 1 | 0 or 1 | 0 or 1 | 0 or 1 |   0:미사용, 1:사용  ex) 1100 (OneID 자동,간편사용, 기존ID 자동,간편미사용) |

### Sample

|  |  |
| --- | --- |
| Source | NSMutableDictionary \*intent = [NSMutableDictionary dictionary];  [intent setObject:FUNCTION\_DEFAULT\_LOGIN forKey:@"FUNCTION\_NAME"];  [intent setObject:SERVICE\_CD forKey:@"SERVICE\_CD"];  [intent setObject:@"1100" forKey:@"SERVICE\_LOGIN\_TYPE"];  [Activity startActivityForResult:self class:[HybridWebView new] intent:intent requestCode:1]; |

## ID/PW로그인 요청

### 인터페이스 소개

1. 개요 : One ID 회원 ID/PW 로그인 요청을 한다.

2. 설명 : One ID 및 PASSWORD를 입력하여 로그인 요청을 한다.

ID/PW로그인 화면에서 로그인 뿐만 아니라, 회원가입, 아이디 찾기, 비밀번호 찾기 기능도 제공됨.

|  |
| --- |
|  |
| ID/PW 로그인 요청 시의 로그인화면 |

휴대폰 로그인 기능이 추가됨에 따라 ‘휴대폰 로그인’과 ‘ID/PW 로그인’을 모두 사용할 경우 [6.39 휴대폰 로그인 요청]을 참고.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SERVICE\_LOGIN\_TYPE : 서비스에서 사용하고자 하는 로그인 타입 (서비스별 정책 확인 후 요청할 것)   |  |  |  |  | | --- | --- | --- | --- | | OneID 용 | | 기존ID 용 | | | 자동로그인 | 간편로그인 | 자동로그인 | 간편로그인 | | 0 or 1 | 0 or 1 | 0 or 1 | 0 or 1 |   0:미사용, 1:사용  OneID 도입되기 전에 사용하던 구아이디 체계를 ‘기존ID’라고 함.  ex) 요청 예  신규 서비스로 기존의 아이디 체계가 없다면 뒤의 두자리는 ‘00’.  자동로그인 기능만 사용할 경우 앞의 두자리는 ‘10’.  자동로그인과 Single-Sign-On 기능을 모두 사용할 경우 앞의 두자리는 ‘11’.  자동로그인 기능을 사용하지 않고 매번 로그인을 해야하는 서비스의 경우 앞의 두자리는 ‘00’. (이 경우 로그인 화면에 자동로그인 체크박스가 노출되지 않음) |

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | WebView |
| Request URI | FUNCTION\_IDPW\_LOGIN |

### Message Specifications

#### Request

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
| SERVICE\_CD | String | 30 | O | One ID에서 부여한 서비스 코드 |
| USER\_ID | String | 50 |  | 회원아이디(e-mail타입으로 @ 앞은 ‘-‘, ‘\_’, ‘.’ 특수문자 허용) |
| SERVICE\_LOGIN\_TYPE | String | 4 | O | 서비스에서 사용할 로그인 타입   |  |  |  |  | | --- | --- | --- | --- | | OneID 용 | | 기존ID 용 | | | 자동로그인 | 간편로그인 | 자동로그인 | 간편로그인 | | 0 or 1 | 0 or 1 | 0 or 1 | 0 or 1 |   0:미사용, 1:사용  ex) 1100 (OneID 자동,간편사용, 기존ID 자동,간편미사용) |

### Sample

|  |  |
| --- | --- |
| Source | NSMutableDictionary \*intent = [NSMutableDictionary dictionary];  [intent setObject:FUNCTION\_IDPW\_LOGIN forKey:@"FUNCTION\_NAME"];  [intent setObject:SERVICE\_CD forKey:@"SERVICE\_CD"];  [intent setObject:USER\_ID forKey:@"jun@girl.com"];  [intent setObject:@”1010” forKey:@"SERVICE\_LOGIN\_TYPE"];  [Activity startActivityForResult:self class:[HybridWebView new] intent:intent requestCode:1];; |

## 약관동의 및 기존ID 전환 요청

### 인터페이스 소개

1. 개요 : 로그인 응답이 약관 미동의 시 약관동의 화면을 요청을 한다.

2. 설명 : 로그인 이후 응답 값이 약관 미 동의로 넘어온 경우 요청한다.

로그인 화면을 서비스앱에서 별도로 만들고, API를 통해 로그인 요청하였을 경우에 사용.

로그인 응답 값이 약관 미동의 (Appendix A(결과코드표) 참고 : 00009)로 전달되었을 경우 요청.

(기본적으로 DAS Lib화면을 사용 시 로그인 과정에서 DAS 화면 내에서 약관동의 및 로그인 응답을 전달함)

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | WebView |
| Request URI | FUNCTION\_TOS\_REQUEST |

### Message Specifications

#### Request

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
| SERVICE\_CD | String | 30 | O | One ID에서 부여한 서비스 코드 |
| LOGIN\_TYPE | String | 1 | O | '3' |
| AUTH\_TYPE | String | 1 | O | I : ID/PW 인증, S : SSO Key 인증 |
| USER\_ID | String | 50 |  | 회원아이디(e-mail타입으로 @ 앞은 ‘-‘, ‘\_’, ‘.’ 특수문자 허용) |
| USER\_PASSWORD | String | 128 |  | **- 비밀번호 작성 규칙**  1. 8~12자리 영문 대/소문자+숫자 조합  2. 동일 문자/숫자 4자리 이상 연속 입력 불가  3. 연속된 숫자 4자리 이상 사용 불가  4. ID가 비밀번호에 포함 불가 (ex. abc@aaa.com일 경우 abc 비밀번호 포함 불가)  **- 최종 입력 받은 비밀번호를 SHA512로 인코딩 후 BASE64로 인코딩한 값** |
| SSO\_KEY | String | 100 |  | 통합 로그인 키 |
| ID\_TYPE | String | 3 |  | ID구분 (NULL or T00: ONE ID, T01: 이메일 인증가입 ID, T02: 본인정보 입력가입 ID, …) |

### Sample

|  |  |
| --- | --- |
| Source | NSMutableDictionary \*intent = [NSMutableDictionary dictionary];  [intent setObject:FUNCTION\_TOS\_REQUEST forKey:@"FUNCTION\_NAME"];  [intent setObject:SERVICE\_CD forKey:@"SERVICE\_CD"];  [intent setObject:@"3" forKey:@"LOGIN\_TYPE"];  [intent setObject:@ "I " forKey:@"AUTH\_TYPE"];  [intent setObject:@" jun@girl.com " forKey:@"USER\_ID"];  [intent setObject:@" 5sBEiSopxqyxjLVw4PclY1bDPLMX4QocKHIQkYnIGZ8lM0LgTX94CYrSHzidk5BnFbQdvW1mn8QaE/vNmveebg==" forKey:@"USER\_PASSWORD"];  [intent setObject:@" " forKey:@"SSO\_KEY"];  [Activity startActivityForResult:self class:[HybridWebView new] intent:intent requestCode:1]; |

## WebView 로그인 응답

### 인터페이스 소개

1. 개요 : WebView에서 요청된 로그인 결과를 서비스 App에 알려준다.

2. 설명 : 라이브러리에 포함된 WebView Activity 의 로그인 결과를 서비스 App에 알려준다.

DAS의 로그인화면 (ID/PW 로그인, Default 로그인, 휴대폰 로그인)을 사용할 경우의 로그인 응답.

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | Callback |
| Request URI | RC\_LOGIN\_RESPONSE |

### Message Specifications

#### Response

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
| RT | String | 20 | O | 결과 코드([[Appendix A(결과 코드표)]](#_Appendix_A(결과코드)) 참조) |
| RT\_MSG | String | 500 | O | 결과 메시지([[Appendix A(결과 코드표)]](#_Appendix_A(결과코드)) 참조) |
| SSO\_KEY | String | 100 |  | 통합 로그인 키 |
| LOGIN\_TYPE | String | 1 |  | 로그인 유형(0:선택안함 1:자동로그인, 2:자동로그인+간편로그인, 3:이전선택유형과 변경없음) |
| USER\_ID | String | 50 |  | 회원아이디(e-mail타입으로 @ 앞은 ‘-‘, ‘\_’, ‘.’ 특수문자 허용) |
| ONEID\_KEY | String | 13 |  | 회원일련번호(SEQ) |
| SERVICE\_KEY | String | 13 |  | U+Box 등 서비스의 고객 Unique ID 값 |
| NAME | String | 180 |  | 회원이름 |
| LGT\_TYPE | String | 1 |  | 자타사구분(N:타사,Y:자사) |
| PW\_UPDATE\_DT | String | 8 |  | 비밀번호 마지막 갱신일 |
| TOS\_SERVICE\_CD | String | 30 |  | 추가 약관동의필수 서비스 코드 |
| ID\_TYPE | String | 3 |  | ID구분 (NULL or T00: ONE ID, T01: 이메일 인증가입 ID, T02: 본인정보 입력가입 ID, …) |

### Sample

|  |  |
| --- | --- |
| Source | - (void)viewDidAppear:(BOOL)animated  {  if([data[@"RESULT"] isEqualToString:RC\_LOGIN\_RESPONSE])  {  NSString \*tmpRT = data[@"RT"];  NSString \*tmpRT\_MSG = data[@"RT\_MSG"];  NSString \*tmpSSO\_KEY = data[@"SSO\_KEY"];  NSString \*tmpLOGIN\_TYPE = data[@"LOGIN\_TYPE "];  NSString \*tmpUSER\_ID = data[@"USER\_ID "];  NSString \*tmpONEID\_KEY = data[@"ONEID\_KEY "];  NSString \*tmpSERVICE\_KEY = data[@"SERVICE\_KEY"];  NSString \*tmpNAME = data[@"NAME "];  NSString \*tmpLGT\_TYPE = data[@"LGT\_TYPE "];  NSString \*tmpPW\_UPDATE\_DT= data[@"PW\_UPDATE\_DT""];  NSString \*tmpTOT\_SERVICE\_CD= data[@"TOS\_SERVICE\_CD "];  NSString \*tmpID\_TYPE = data[@"ID\_TYPE "];;  }  } |

## 기존ID로 로그인 요청

### 인터페이스 소개

1. 개요 : One ID 로그인 화면에서 '기존ID로 로그인'을 선택했을 경우 서비스 App에 알려준다.

2. 설명 : 서비스 앱의 기존 로그인 방식으로 로그인을 처리한다.

ID/PW 로그인 화면에서 이 메일 형식이 아닌 ID로 요청되었을 경우, 기존ID 체계로 판단하여 입력받은 ID/PW를 그대로 앱에 전달.

기존ID 체계가 있는 경우 전달한 ID/PW로 로그인 처리하거나, 기존ID 체계가 없는 경우는 무시 혹은 안내하도록 할 것.

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | Callback |
| Request URI | RC\_CALL\_OLD\_LOGIN |

### Message Specifications

#### Response

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
| SERVICE\_USER\_ID | String | 100 | O | 서비스 기존 회원 ID |
| SERVICE\_USER\_PW | String | 128 | O | 서비스 기존 회원 비밀번호 |
| LOGIN\_TYPE | String | 1 | O | 로그인 유형(0:선택안함 1:자동로그인, 2:자동로그인+간편로그인, 3:이전선택유형과 변경없음) |

### Sample

|  |  |
| --- | --- |
| Source | - (void)viewDidAppear:(BOOL)animated  {  if([data[@"RESULT"] isEqualToString:RC\_CALL\_OLD\_LOGIN])  {  NSString \*tmpSERVICE\_USER\_ID = data[@"SERVICE\_USER\_ID "];  NSString \*tmpSERVICE\_USER\_PW = data[@"SERVICE\_USER\_PW "];  NSString \*tmpLOGIN\_TYPE = data[@"LOGIN\_TYPE "];  }  } |

## One ID 관리화면 요청

### 인터페이스 소개

1. 개요 : One ID 관리화면 요청을 한다.

2. 설명 : 서비스 앱의 설정화면에서 One ID 관리화면을 호출한다.

(v2.0.2~ : Request SERVICE\_LOGIN\_TYPE 추가. OneID 관리화면에 간편로그인을 사용하지 않는 서비스에 대해 간편로그인 관리를 비노출하기 위해 추가)

로그인 된 사용자에 대한 관리화면으로 간편로그인 관리 / 이용중인 서비스 조회 / 회원정보 수정(비밀번호 변경/연락가능번호 변경/모바일인증/정회원전환 등)의 기능을 제공.

|  |
| --- |
|  |
| OneID 관리화면 호출 시 화면 |

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | WebView |
| Request URI | FUNCTION\_ONEID\_CONFIG |

### Message Specifications

#### Request

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
| SERVICE\_CD | String | 30 | O | One ID에서 부여한 서비스 코드 |
| ONEID\_KEY | String | 13 | O | 회원일련번호(SEQ) |
| USER\_ID | String | 50 | O | 회원아이디(e-mail타입으로 @ 앞은 ‘-‘, ‘\_’, ‘.’ 특수문자 허용) |
| SERVICE\_LOGIN\_TYPE | String | 4 | O | 서비스에서 사용할 로그인 타입   |  |  |  |  | | --- | --- | --- | --- | | OneID 용 | | 기존ID 용 | | | 자동로그인 | 간편로그인 | 자동로그인 | 간편로그인 | | 0 or 1 | 0 or 1 | 0 or 1 | 0 or 1 |   0:미사용, 1:사용  ex) 1100 (OneID 자동,간편사용, 기존ID 자동,간편미사용) |

### Sample

|  |  |
| --- | --- |
| Source | NSDictionary \*tmpMap = [AccountUtil getNormalAccount];  NSMutableDictionary \*intent = [NSMutableDictionary dictionary];  [intent setObject:FUNCTION\_ONEID\_CONFIG forKey:@"FUNCTION\_NAME"];  [intent setObject:SERVICE\_CD forKey:@"SERVICE\_CD"];  [intent setObject:@”” forKey:@"ONEID\_KEY"];  [intent setObject:@"jun@girl.com" forKey:@"USER\_ID"];  [intent setObject:@”1000” forKey:@"SERVICE\_LOGIN\_TYPE"];  [Activity startActivityForResult:self class:[HybridWebView new] intent:intent requestCode:1]; |

## ID 찾기 요청

### 인터페이스 소개

1. 개요 : One ID 회원 ID 찾기 요청을 한다.

2. 설명 : 서비스 앱에서 One ID 찾기 기능이 필요할 때 호출한다.

서비스앱에서 [ID 찾기] 기능을 직접 구현할 경우에만 요청. 그 외에는 DAS 로그인 화면 내에 ID 찾기 기능을 사용.

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | WebView |
| Request URI | FUNCTION\_ID\_SEARCH |

### Message Specifications

#### Request

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
| SERVICE\_CD | String | 30 | O | One ID에서 부여한 서비스 코드 |
| SERVICE\_LOGIN\_TYPE | String | 4 | O | 서비스에서 사용할 로그인 타입   |  |  |  |  | | --- | --- | --- | --- | | OneID 용 | | 기존ID 용 | | | 자동로그인 | 간편로그인 | 자동로그인 | 간편로그인 | | 0 or 1 | 0 or 1 | 0 or 1 | 0 or 1 |   0:미사용, 1:사용  ex) 1100 (OneID 자동,간편사용, 기존ID 자동,간편미사용) |

### Sample

|  |  |
| --- | --- |
| Source | NSMutableDictionary \*intent = [NSMutableDictionary dictionary];  [intent setObject:FUNCTION\_ID\_SEARCH forKey:@"FUNCTION\_NAME"];  [intent setObject:SERVICE\_CD forKey:@"SERVICE\_CD"];  [intent setObject:@”1110” forKey:@"SERVICE\_LOGIN\_TYPE"];  [Activity startActivityForResult:self class:[HybridWebView new] intent:intent requestCode:1]; |

## PW 찾기 요청

### 인터페이스 소개

1. 개요 : One ID 회원 PW 찾기 요청을 한다.

2. 설명 : 서비스 앱에서 One ID의 pw찾기 기능이 필요할 때 호출한다.

서비스앱에서 [PW 찾기] 기능을 직접 구현할 경우에만 요청. 그 외의 경우 DAS 로그인 화면 내에 비밀번호 찾기 기능을 사용.

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | WebView |
| Request URI | FUNCTION\_PASSWORD\_RESET |

### Message Specifications

#### Request

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
| SERVICE\_CD | String | 30 | O | One ID에서 부여한 서비스 코드 |
| SERVICE\_LOGIN\_TYPE | String | 4 | O | 서비스에서 사용할 로그인 타입   |  |  |  |  | | --- | --- | --- | --- | | OneID 용 | | 기존ID 용 | | | 자동로그인 | 간편로그인 | 자동로그인 | 간편로그인 | | 0 or 1 | 0 or 1 | 0 or 1 | 0 or 1 |   0:미사용, 1:사용  ex) 1100 (OneID 자동,간편사용, 기존ID 자동,간편미사용) |

### Sample

|  |  |
| --- | --- |
| Source | NSMutableDictionary \*intent = [NSMutableDictionary dictionary];  [intent setObject:FUNCTION\_PASSWORD\_RESET forKey:@"FUNCTION\_NAME"];  [intent setObject:SERVICE\_CD forKey:@"SERVICE\_CD"];  [intent setObject:@”1010” forKey:@"SERVICE\_LOGIN\_TYPE"];  [Activity startActivityForResult:self class:[HybridWebView new] intent:intent requestCode:1]; |

## 기존 ID 찾기 요청

### 인터페이스 소개

1. 개요 : 기존 ID 회원 ID 찾기 요청을 한다.

2. 설명 : 서비스 앱의 기존에 사용하던 회원 ID 찾기를 수행한다.

기존ID 체계를 사용하는 서비스에서 DAS 로그인 화면 내에 [아이디 찾기]를 선택 시 아래와 같은 화면이 노출됨.

|  |
| --- |
|  |
| ID 찾기 분기화면 |

이 화면에서 [기존ID 찾기] 버튼이 선택 됨을 서비스앱에 전달.

이 Callback을 전달받은 경우, 기존ID 체계의 ID찾기 기능을 수행하도록 할 것.

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | Callback |
| Request URI | RC\_SERVICE\_ID\_SEARCH |

### Message Specifications

#### Response

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
|  |  |  |  | 없음 |

### Sample

|  |  |
| --- | --- |
| Source | - (void)viewDidAppear:(BOOL)animated  {  if([data[@"RESULT"] isEqualToString:RC\_SERVICE\_ID\_SEARCH])  {  }  } |

## 기존 PW 찾기 요청

### 인터페이스 소개

1. 개요 : 기존 ID 회원 PW 찾기 요청을 한다.

2. 설명 : 서비스 앱의 기존에 사용하던 회원 PW 찾기를 수행한다.

기존ID 체계를 사용하는 서비스에서 DAS 로그인 화면 내에 [비밀번호 찾기]를 선택 시 아래와 같은 화면이 노출됨.

|  |
| --- |
|  |
| 비밀번호 찾기 분기화면 |

이 화면에서 [기존ID 재설정] 버튼이 선택 됨을 서비스앱에 전달.

이 Callback을 전달받은 경우, 기존ID 체계의 비밀번호 기능을 수행하도록 할 것.

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | Callback |
| Request URI | RC\_SERVICE\_PASSWORD\_RESET |

### Message Specifications

#### Response

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
|  |  |  |  | 없음 |

### Sample

|  |  |
| --- | --- |
| Source | - (void)viewDidAppear:(BOOL)animated  {  if([data[@"RESULT"] isEqualToString:RC\_SERVICE\_PASSWORD\_RESET])  {  }  } |

## 전환 가입 요청

### 인터페이스 소개

1. 개요 : 기존 서비스의 ID 회원을 One ID 회원으로 전환 요청을 한다.

2. 설명 : 서비스 앱의 기존에 사용하던 회원 ID를 One ID 로 전환 요청을 한다.

기존ID 체계가 존재하는 서비스에서 기존ID로 로그인된 계정을 OneID로 변경하고자 할 때 요청.

요청 시점은 서비스앱별 정책에 따름 (ex. 기존ID로 로그인 시 팝업으로 OneID 변경안내 등..)

CI, CTN, ENTR\_NO 를 전달하면 DAS에서는 전달받은 정보 기준으로 자사OneID로 생성할지, 타사OneID로 생성할지 판단하여 OneID생성하고 결과를 [6.20 전환 가입 응답]으로 전달.

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | WebView |
| Request URI | FUNCTION\_CONVERSION\_REQUEST |

### Message Specifications

#### Request

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
| SERVICE\_CD | String | 30 | O | One ID에서 부여한 서비스 코드 |
| CI | String | 88 |  | CI (연계정보) |
| CTN | String | 20 |  | 로그인 CTN (444타입 - 제외) |
| ENTR\_NO | String | 13 |  | 무선가입자식별번호 |
| SERVICE\_KEY | String | 13 | O | 서비스의 고객 Unique ID 값 |
| SERVICE\_USER\_ID | String | 50 | O | 기존 서비스 사용자 ID |
| SERVICE\_LOGIN\_TYPE | String | 4 | O | 서비스에서 사용할 로그인 타입   |  |  |  |  | | --- | --- | --- | --- | | OneID 용 | | 기존ID 용 | | | 자동로그인 | 간편로그인 | 자동로그인 | 간편로그인 | | 0 or 1 | 0 or 1 | 0 or 1 | 0 or 1 |   0:미사용, 1:사용  ex) 1100 (OneID 자동,간편사용, 기존ID 자동,간편미사용) |

### Sample

|  |  |
| --- | --- |
| Source | NSMutableDictionary \*intent = [NSMutableDictionary dictionary];  [intent setObject:FUNCTION\_CONVERSION\_REQUEST forKey:@"FUNCTION\_NAME"];  [intent setObject:SERVICE\_CD forKey:@"SERVICE\_CD"];  [intent setObject:@”” forKey:@"CI"];  [intent setObject:@”” forKey:@"SERVICE\_KEY"];  [intent setObject:@”” forKey:@"ENTR\_NO"];  [intent setObject:@”” forKey:@"SERVICE\_USER\_ID"];  [intent setObject:@”” forKey:@"SERVICE\_LOGIN\_TYPE"];  [Activity startActivityForResult:self class:[HybridWebView new] intent:intent requestCode:1]; |

## 전환 가입 응답

### 인터페이스 소개

1. 개요 : 기존 서비스의 ID 회원을 One ID 회원으로 전환한 결과를 서비스 App에 알려준다.

2. 설명 : 전환가입 처리 결과를 이용하여 서비스 앱이 후처리를 진행한다.

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | Callback |
| Request URI | RC\_CONVERSION\_RESPONSE |

### Message Specifications

#### Response

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
| RT | String | 20 | O | 결과 코드(문서 마지막에 코드값 첨부) |
| RT\_MSG | String | 500 | O | 결과 메시지(문서 마지막에 메시지값 첨부) |
| ONEID\_KEY | String | 13 | O | 회원일련번호(SEQ) |
| USER\_ID | String | 50 | O | 회원아이디(e-mail타입으로 @ 앞은 ‘-‘, ‘\_’, ‘.’ 특수문자 허용) |
| CLOSE\_ST | String | 1 | O | 0:확인버튼, 1:X버튼 |

### Sample

|  |  |
| --- | --- |
| Source | - (void)viewDidAppear:(BOOL)animated  {  if([data[@"RESULT"] isEqualToString:RC\_CONVERSION\_RESPONSE])  {  NSString \*tmpRT = data[@"RT"];  NSString \*tmpRT\_MSG = data[@"RT\_MSG"];  NSString \*tmpONEID\_KEY = data[@"ONEID\_KEY"];  NSString \*tmpUSER\_ID = data[@"USER\_ID"];  NSString \*tmpCLOSE\_ST = data[@"CLOSE\_ST"];  }  } |

## 계정 및 동기화 삭제 요청

### 인터페이스 소개

1. 개요 : One ID에 해당하는 계정 및 동기화 정보 삭제 요청을 한다.

2. 설명 : 안드로이드 환경설정에 등록되어있는 계정 및 동기화의 One ID 삭제를 요청한다. 단, 계정이 존재하지 않더라도 서비스에는 정상으로 응답한다.

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | Service |
| Request URI | IntegrationConstants.FUNCTION\_DELETE\_ACCOUNT |

### Message Specifications

#### Request

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
| SERVICE\_CD | String | 20 | O | One ID에서 부여한 서비스 코드 |
| USER\_ID | String | 50 | O | 회원아이디(e-mail타입으로 @ 앞은 ‘-‘, ‘\_’, ‘.’ 특수문자 허용) |

### Sample

|  |  |
| --- | --- |
| Source | Intent intent = new Intent("com.lg.das.intent.service");  intent.putExtra("FUNCTION\_NAME", IntegrationConstants.FUNCTION\_DELETE\_ACCOUNT);  intent.putExtra("SERVICE\_CD", AccountUtil.SERVICE\_CD);  intent.putExtra("USER\_ID", (String)tmpMap.get("USER\_ID"));  startIntent(intent); |

## 계정 및 동기화 삭제 응답

### 인터페이스 소개

1. 개요 : One ID에 해당하는 계정 및 동기화 정보 삭제에 대한 응답을 받는다.

2. 설명 : 안드로이드 환경설정에 등록되어있는 계정 및 동기화의 One ID 삭제 응답결과

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | Callback |
| Request URI | RC\_DELETE\_ACCOUNT |

### Message Specifications

#### Response

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
| RT | String | 20 | O | 결과 코드(문서 마지막에 코드값 첨부) |
| RT\_MSG | String | 500 | O | 결과 메시지(문서 마지막에 메시지값 첨부) |

### Sample

|  |  |
| --- | --- |
| Source | - (void)viewDidAppear:(BOOL)animated  {  if([data[@"RESULT"] isEqualToString:RC\_DELETE\_ACCOUNT])  {  NSString \*tmpRT = data[@"RT"];  NSString \*tmpRT\_MSG = data[@"RT\_MSG"];  }  } |

## 로그아웃

### 인터페이스 소개

1. 개요 : 로그인 되어있는 OneID를 로그아웃 시킨다.

2. 설명 : 로그아웃 타입에 따라 계정 및 동기화에 등록되어있는 OneID는 자동 삭제된다.

관리화면에서 비밀번호 재설정 시 등에 DAS에서 호출한다.

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | Callback |
| Request URI | RC\_LOGOUT |

### Message Specifications

#### Response

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
|  |  |  |  | 없음 |

### Sample

|  |  |
| --- | --- |
| Source | - (void)viewDidAppear:(BOOL)animated  {  if([data[@"RESULT"] isEqualToString:RC\_LOGOUT])  {  //로그아웃 처리  }  } |

## 타사가입자 자사모바일 승급

### 인터페이스 소개

1. 개요 : 타사가입자를 자사모바일 가입자로 승급시킨다.

2. 설명 : 타사로 가입한 OneID 사용자가 LGT 모바일 가입자가 되었을 경우 자사가입자로 승급 처리시킨다.

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | WebView |
| Request URI | FUNCTION\_AUTH\_MOBILE |

### Message Specifications

#### Request

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
| SERVICE\_CD | String | 30 | O | One ID에서 부여한 서비스 코드 |
| ONEID\_KEY | String | 13 | O | 회원일련번호(SEQ) |
| USER\_ID | String | 50 | O | 회원아이디(e-mail타입으로 @ 앞은 ‘-‘, ‘\_’, ‘.’ 특수문자 허용) |

### Sample

|  |  |
| --- | --- |
| Source | NSDictionary \*tmpMap = [AccountUtil getNormalAccount];  SMutableDictionary \*intent = [NSMutableDictionary dictionary];  [intent setObject:FUNCTION\_AUTH\_MOBILE forKey:@"FUNCTION\_NAME"];  [intent setObject:SERVICE\_CD forKey:@"SERVICE\_CD"];  [intent setObject:@”” forKey:@"ONEID\_KEY"];  [intent setObject:@"jun@girl.com" forKey:@"USER\_ID"];  [Activity startActivityForResult:self class:[HybridWebView new] intent:intent requestCode:1]; |

## UFLIX/HDTV 이용정보 조회 요청

### 인터페이스 소개

1. 개요 : One ID 회원 UFLIX/HDTV 상품 이용정보 조회 요청을 한다.

2. 설명 : One ID 사용을 위한 UFLIX/HDTV 상품 이용정보 조회 요청을 한다.

비디오포털 서비스에서 이용정보 조회를 할 경우 요청.

조회된 화면에서 기존ID는 전환처리, OneID는 로그인 처리, 임시ID는 OneID로 변경 등이 진행됨.

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | WebView |
| Request URI | FUNCTION\_UFLIX\_INFO |

### Message Specifications

#### Request

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
| SERVICE\_CD | String | 30 | O | One ID에서 부여한 서비스 코드 |
| SERVICE\_LOGIN\_TYPE | String | 4 | O | 서비스에서 사용할 로그인 타입   |  |  |  |  | | --- | --- | --- | --- | | OneID 용 | | 기존ID 용 | | | 자동로그인 | 간편로그인 | 자동로그인 | 간편로그인 | | 0 or 1 | 0 or 1 | 0 or 1 | 0 or 1 |   0:미사용, 1:사용  ex) 1100 (OneID 자동,간편사용, 기존ID 자동,간편미사용) |

### Sample

|  |  |
| --- | --- |
| Source | NSMutableDictionary \*intent = [NSMutableDictionary dictionary];  [intent setObject:FUNCTION\_UFLIX\_INFO forKey:@"FUNCTION\_NAME"];  [intent setObject:SERVICE\_CD forKey:@"SERVICE\_CD"];  [intent setObject:@"1100" forKey:@"SERVICE\_LOGIN\_TYPE"];  [Activity startActivityForResult:self class:[HybridWebView new] intent:intent requestCode:1]; |

## 페이지 종료 알림

### 인터페이스 소개

1. 개요 : 페이지 종료방법 반환

2. 설명 : 페이지 종료 시 종료 방법을 반환한다.

DAS 화면이 닫혔을 경우 서비스앱에 전달합니다.

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | Callback |
| Request URI | RC\_CLOSE |

### Message Specifications

#### Response

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
| CLOSE\_ST | String | 1 | O | **0 : 확인, 완료 등 화면상의 작업완료를 나타내는 버튼으로 닫음**  **1 : X버튼, 백키로 닫음** |

### Sample

|  |  |
| --- | --- |
| Source | - (void)viewDidAppear:(BOOL)animated  {  if([data[@"RESULT"] isEqualToString:RC\_CLOSE])  {  NSString \*tmpCLOSE\_ST = data[@"CLOSE\_ST"];  }  } |

## ~~SNS ID 자동로그인 요청~~

### ~~인터페이스 소개~~

~~1. 개요 : SNS ID에 대한 자동로그인 요청.~~

~~2. 설명 : 서비스 앱 정책에 따라 서비스 앱이 보관 하고 있는 SNS ID 계정 정보를 이용하여 로그인 요청을 한다.~~

~~3. 프로토콜 상세~~

|  |  |
| --- | --- |
| **~~항목~~** | **~~내용~~** |
| ~~Call Type~~ | ~~WebView~~ |
| ~~Request URI~~ | ~~IntegrationConstants.FUNCTION\_SNS\_AUTO\_LOGIN~~ |

### ~~Message Specifications~~

#### ~~Request~~

| **~~매개 변수명~~** | **~~자료형~~** | **~~길이~~** | **~~필수~~** | **~~설명~~** |
| --- | --- | --- | --- | --- |
| ~~SERVICE\_CD~~ | ~~String~~ | ~~30~~ | ~~O~~ | ~~One ID에서 부여한 서비스 코드~~ |
| ~~SNSID\_KEY~~ | ~~String~~ | ~~13~~ | ~~O~~ | ~~SNS 회원 일련번호~~ |
| ~~SNS\_CD~~ | ~~String~~ | ~~3~~ | ~~O~~ | ~~S01 : facebook, S02 : Google+~~ |
| ~~SNS\_USER\_ID~~ | ~~String~~ | ~~50~~ | ~~O~~ | ~~SNS 회원 ID~~ |

### ~~Sample~~

|  |  |
| --- | --- |
| ~~Source~~ | ~~Intent intent = new Intent(ctCtext,HybridWebView.class);~~  ~~intent.putExtra("FUNCTION\_NAME", IntegrationConstants.FUNCTION\_SNS\_AUTO\_LOGIN);~~  ~~intent.putExtra("SERVICE\_CD", AccountUtil.SERVICE\_CD);~~  ~~intent.putExtra("SNSID\_KEY ", “11111111”);~~  ~~intent.putExtra("SNS\_CD ", “S01”);~~  ~~intent.putExtra("SNS\_USER\_ID", “12345678”);~~  ~~startActivityForResult(intent, 1);~~ |

## ~~SNS ID 로그인 응답~~

### ~~인터페이스 소개~~

~~1. 개요 : WebView에서 요청된 SNS ID 로그인 결과를 서비스 App에 알려준다.~~

~~2. 설명 : 라이브러리에 포함된 WebView Activity 의 SNS ID 로그인 결과를 서비스 App에 알린다.~~

~~3. 프로토콜 상세~~

|  |  |
| --- | --- |
| **~~항목~~** | **~~내용~~** |
| ~~Call Type~~ | ~~Callback~~ |
| ~~Request URI~~ | ~~IntegrationConstants.RC\_SNS\_LOGIN\_RESPONSE~~ |

### ~~Message Specifications~~

#### ~~Response~~

| **~~매개 변수명~~** | **~~자료형~~** | **~~길이~~** | **~~필수~~** | **~~설명~~** |
| --- | --- | --- | --- | --- |
| ~~RT~~ | ~~String~~ | ~~20~~ | ~~O~~ | ~~결과 코드(~~[~~[Appendix A(결과 코드표)]~~](#_Appendix_A(결과코드)) ~~참조)~~ |
| ~~RT\_MSG~~ | ~~String~~ | ~~500~~ | ~~O~~ | ~~결과 메시지(~~[~~[Appendix A(결과 코드표)]~~](#_Appendix_A(결과코드)) ~~참조)~~ |
| ~~SNSID\_KEY~~ | ~~String~~ | ~~13~~ | ~~O~~ | ~~SNS ID 회원 일련번호~~ |
| ~~SNS\_CD~~ | ~~String~~ | ~~3~~ | ~~O~~ | ~~S01 : facebook, S02 : Google+~~ |
| ~~SNS\_USER\_ID~~ | ~~String~~ | ~~50~~ | ~~O~~ | ~~SNS 회원 ID~~ |
| ~~NAME~~ | ~~String~~ | ~~180~~ |  | ~~회원이름~~ |
| ~~EMAIL~~ | ~~String~~ | ~~50~~ |  | ~~회원Email~~ |
| ~~TOS\_SERVICE\_CD~~ | ~~String~~ | ~~30~~ |  | ~~추가 약관동의필수 서비스 코드~~ |

### ~~Sample~~

|  |  |
| --- | --- |
| ~~Source~~ | ~~@Override~~  ~~protected void onActivityResult(int requestCode, int resultCode, Intent data) {~~  ~~super.onActivityResult(requestCode, resultCode, data);~~  ~~if(data.getStringExtra("RESULT").equals(IntegrationConstants.RC\_SNS\_LOGIN\_RESPONSE)) {~~  ~~String tmpRT = data.getStringExtra("RT");~~  ~~String tmpRT\_MSG = data.getStringExtra("RT\_MSG");~~  ~~String tmpSNSID\_KEY = data.getStringExtra("SNSID\_KEY");~~  ~~String tmpSNS\_CD = data.getStringExtra("SNS\_CD");~~  ~~String tmpSNS\_USER\_ID = data.getStringExtra("SNS\_USER\_ID");~~  ~~String tmpNAME = data.getStringExtra("NAME");~~  ~~String tmpEMAIL = data.getStringExtra("EMAIL");~~  ~~String tmpTOS\_SERVICE\_CD = data.getStringExtra("TOS\_SERVICE\_CD");~~  ~~}~~  ~~}~~ |

## ~~SNS ID 로그아웃 요청~~

### ~~인터페이스 소개~~

~~1. 개요 : SNS ID에 대한 로그아웃 요청.~~

~~2. 설명 : 서비스에서 SNS ID 로그아웃 시 SNS(Facebook, Google+) 로그아웃 처리를 위해 요청한다.~~

~~3. 프로토콜 상세~~

|  |  |
| --- | --- |
| **~~항목~~** | **~~내용~~** |
| ~~Call Type~~ | ~~WebView~~ |
| ~~Request URI~~ | ~~IntegrationConstants.FUNCTION\_SNS\_LOGOUT~~ |

### ~~Message Specifications~~

#### ~~Request~~

| **~~매개 변수명~~** | **~~자료형~~** | **~~길이~~** | **~~필수~~** | **~~설명~~** |
| --- | --- | --- | --- | --- |
| ~~SERVICE\_CD~~ | ~~String~~ | ~~30~~ | ~~O~~ | ~~One ID에서 부여한 서비스 코드~~ |
| ~~SNSID\_KEY~~ | ~~String~~ | ~~13~~ | ~~O~~ | ~~SNS 회원 일련번호~~ |
| ~~SNS\_CD~~ | ~~String~~ | ~~3~~ | ~~O~~ | ~~S01 : facebook, S02 : Google+~~ |
| ~~SNS\_USER\_ID~~ | ~~String~~ | ~~50~~ | ~~O~~ | ~~SNS 회원 ID~~ |

### ~~Sample~~

|  |  |
| --- | --- |
| ~~Source~~ | ~~Intent intent = new Intent(ctCtext,HybridWebView.class);~~  ~~intent.putExtra("FUNCTION\_NAME", IntegrationConstants.FUNCTION\_SNS\_LOGOUT);~~  ~~intent.putExtra("SERVICE\_CD", AccountUtil.SERVICE\_CD);~~  ~~intent.putExtra("SNSID\_KEY ", “11111111”);~~  ~~intent.putExtra("SNS\_CD ", “S01”);~~  ~~intent.putExtra("SNS\_USER\_ID", “12345678”);~~  ~~startActivityForResult(intent, 1);~~ |

## ~~SNS ID 로그아웃 응답~~

### ~~인터페이스 소개~~

~~1. 개요 : SNS ID 로그아웃~~

~~2. 설명 : 로그인 되어있는 SNS ID의 로그아웃 처리 결과를 서비스 App에 알린다.~~

~~3. 프로토콜 상세~~

|  |  |
| --- | --- |
| **~~항목~~** | **~~내용~~** |
| ~~Call Type~~ | ~~Callback~~ |
| ~~Request URI~~ | ~~IntegrationConstants.RC\_SNS\_LOGOUT~~ |

### ~~Message Specifications~~

#### ~~Response~~

| **~~매개 변수명~~** | **~~자료형~~** | **~~길이~~** | **~~필수~~** | **~~설명~~** |
| --- | --- | --- | --- | --- |
| ~~RT~~ | ~~String~~ | ~~20~~ | ~~O~~ | ~~결과 코드(~~[~~[Appendix A(결과 코드표)]~~](#_Appendix_A(결과코드)) ~~참조)~~ |
| ~~RT\_MSG~~ | ~~String~~ | ~~500~~ | ~~O~~ | ~~결과 메시지(~~[~~[Appendix A(결과 코드표)]~~](#_Appendix_A(결과코드)) ~~참조)~~ |
| ~~SNSID\_KEY~~ | ~~String~~ | ~~13~~ |  | ~~SNS ID 회원 일련번호~~ |

### ~~Sample~~

|  |  |
| --- | --- |
| ~~Source~~ | ~~@Override~~  ~~protected void onActivityResult(int requestCode, int resultCode, Intent data) {~~  ~~super.onActivityResult(requestCode, resultCode, data);~~  ~~if(data.getStringExtra("RESULT").equals(IntegrationConstants.RC\_SNS\_LOGOUT)) {~~  ~~String tmpSNSID\_KEY = data.getStringExtra("SNSID\_KEY");~~  ~~}~~  ~~}~~ |

## ~~SNS ID 해지 요청~~

### ~~인터페이스 소개~~

~~1. 개요 : SNS ID에 대한 해지 요청.~~

~~2. 설명 : 서비스에서 SNS ID 해지 시 SNS(Facebook, Google+) 앱 연결 해제 처리를 위해 요청한다.~~

~~3. 프로토콜 상세~~

|  |  |
| --- | --- |
| **~~항목~~** | **~~내용~~** |
| ~~Call Type~~ | ~~WebView~~ |
| ~~Request URI~~ | ~~IntegrationConstants.FUNCTION\_SNS\_REVOKE~~ |

### ~~Message Specifications~~

#### ~~Request~~

| **~~매개 변수명~~** | **~~자료형~~** | **~~길이~~** | **~~필수~~** | **~~설명~~** |
| --- | --- | --- | --- | --- |
| ~~SERVICE\_CD~~ | ~~String~~ | ~~30~~ | ~~O~~ | ~~One ID에서 부여한 서비스 코드~~ |
| ~~SNSID\_KEY~~ | ~~String~~ | ~~13~~ | ~~O~~ | ~~SNS 회원 일련번호~~ |
| ~~SNS\_CD~~ | ~~String~~ | ~~3~~ | ~~O~~ | ~~S01 : facebook, S02 : Google+~~ |
| ~~SNS\_USER\_ID~~ | ~~String~~ | ~~50~~ | ~~O~~ | ~~SNS 회원 ID~~ |

### ~~Sample~~

|  |  |
| --- | --- |
| ~~Source~~ | ~~Intent intent = new Intent(ctCtext,HybridWebView.class);~~  ~~intent.putExtra("FUNCTION\_NAME", IntegrationConstants.FUNCTION\_SNS\_REVOKE);~~  ~~intent.putExtra("SERVICE\_CD", AccountUtil.SERVICE\_CD);~~  ~~intent.putExtra("SNSID\_KEY ", “11111111”);~~  ~~intent.putExtra("SNS\_CD ", “S01”);~~  ~~intent.putExtra("SNS\_USER\_ID", “12345678”);~~  ~~startActivityForResult(intent, 1);~~ |

## ~~SNS ID 해지 응답~~

### ~~인터페이스 소개~~

~~1. 개요 : SNS ID에 대한 해지 응답.~~

~~2. 설명 : SNS ID 회원에 대한 서비스 해지/탈퇴 처리 결과를 서비스 App에 알린다.~~

~~3. 프로토콜 상세~~

|  |  |
| --- | --- |
| **~~항목~~** | **~~내용~~** |
| ~~Call Type~~ | ~~Callback~~ |
| ~~Request URI~~ | ~~IntegrationConstants.RC\_SNS\_REVOKE~~ |

### ~~Message Specifications~~

#### ~~Response~~

| **~~매개 변수명~~** | **~~자료형~~** | **~~길이~~** | **~~필수~~** | **~~설명~~** |
| --- | --- | --- | --- | --- |
| ~~RT~~ | ~~String~~ | ~~20~~ | ~~O~~ | ~~결과 코드(~~[~~[Appendix A(결과 코드표)]~~](#_Appendix_A(결과코드)) ~~참조)~~ |
| ~~RT\_MSG~~ | ~~String~~ | ~~500~~ | ~~O~~ | ~~결과 메시지(~~[~~[Appendix A(결과 코드표)]~~](#_Appendix_A(결과코드)) ~~참조)~~ |

### ~~Sample~~

|  |  |
| --- | --- |
| ~~Source~~ | ~~@Override~~  ~~protected void onActivityResult(int requestCode, int resultCode, Intent data) {~~  ~~super.onActivityResult(requestCode, resultCode, data);~~  ~~if(data.getStringExtra("RESULT").equals(IntegrationConstants.RC\_SNS\_REVOKE)) {~~  ~~//해지처리 처리~~  ~~}~~  ~~}~~ |

## ~~SNS ID 전환가입 요청~~

### ~~인터페이스 소개~~

~~1. 개요 : SNS ID 회원의 One ID 회원 전환 요청.~~

~~2. 설명 : SNS ID 회원을 One ID 회원으로 전환 요청 한다.~~

~~3. 프로토콜 상세~~

|  |  |
| --- | --- |
| **~~항목~~** | **~~내용~~** |
| ~~Call Type~~ | ~~WebView~~ |
| ~~Request URI~~ | ~~IntegrationConstants.FUNCTION\_SNS\_CONVERSION\_REQUEST~~ |

### ~~Message Specifications~~

#### ~~Request~~

| **~~매개 변수명~~** | **~~자료형~~** | **~~길이~~** | **~~필수~~** | **~~설명~~** |
| --- | --- | --- | --- | --- |
| ~~SERVICE\_CD~~ | ~~String~~ | ~~30~~ |  | ~~One ID에서 부여한 서비스 코드~~ |
| ~~SNSID\_KEY~~ | ~~String~~ | ~~13~~ |  | ~~SNS ID 회원 일련번호~~ |
| ~~SNS\_CD~~ | ~~String~~ | ~~3~~ | ~~O~~ | ~~S01 : facebook, S02 : Google+~~ |
| ~~SNS\_USER\_ID~~ | ~~String~~ | ~~50~~ |  | ~~SNS 회원 ID~~ |

### ~~Sample~~

|  |  |
| --- | --- |
| ~~Source~~ | ~~Intent intent = new Intent(ctCtext,HybridWebView.class);~~  ~~intent.putExtra("FUNCTION\_NAME", IntegrationConstants.FUNCTION\_SNS\_CONVERSION\_REQUEST);~~  ~~intent.putExtra("SERVICE\_CD", AccountUtil.SERVICE\_CD);~~  ~~intent.putExtra("SNSID\_KEY ", “11111111”);~~  ~~intent.putExtra("SNS\_CD ", “S01”);~~  ~~intent.putExtra("SNS\_USER\_ID", “12345678”);~~  ~~startActivityForResult(intent, 1);~~ |

## ~~SNS ID 전환가입 응답~~

### ~~인터페이스 소개~~

~~1. 개요 : SNS ID의 One ID 회원 전환 결과 응답.~~

~~2. 설명 : SNS ID 회원을 One ID 회원으로 전환한 결과를 서비스 App에 알린다.~~

~~3. 프로토콜 상세~~

|  |  |
| --- | --- |
| **~~항목~~** | **~~내용~~** |
| ~~Call Type~~ | ~~Callback~~ |
| ~~Request URI~~ | ~~IntegrationConstants.RC\_SNS\_CONVERSION\_RESPONSE~~ |

### ~~Message Specifications~~

#### ~~Response~~

| **~~매개 변수명~~** | **~~자료형~~** | **~~길이~~** | **~~필수~~** | **~~설명~~** |
| --- | --- | --- | --- | --- |
| ~~RT~~ | ~~String~~ | ~~20~~ | ~~O~~ | ~~결과 코드(문서 마지막에 코드값 첨부)~~ |
| ~~RT\_MSG~~ | ~~String~~ | ~~500~~ | ~~O~~ | ~~결과 메시지(문서 마지막에 메시지값 첨부)~~ |
| ~~SNSID\_KEY~~ | ~~String~~ | ~~13~~ | ~~O~~ | ~~SNS ID 회원 일련번호~~ |
| ~~ONEID\_KEY~~ | ~~String~~ | ~~13~~ |  | ~~회원일련번호(SEQ)~~ |
| ~~USER\_ID~~ | ~~String~~ | ~~50~~ |  | ~~회원아이디(e-mail타입)~~ |
| ~~CLOSE\_ST~~ | ~~String~~ | ~~1~~ |  | ~~화면 닫기 유형 (0:확인버튼, 1. X버튼)~~ |

### ~~Sample~~

|  |  |
| --- | --- |
| ~~Source~~ | ~~@Override~~  ~~protected void onActivityResult(int requestCode, int resultCode, Intent data) {~~  ~~super.onActivityResult(requestCode, resultCode, data);~~  ~~if(data.getStringExtra("RESULT").equals(IntegrationConstants.RC\_SNS\_CONVERSION\_RESPONSE)) {~~  ~~String tmpRT = data.getStringExtra("RT");~~  ~~String tmpRT\_MSG = data.getStringExtra("RT\_MSG");~~  ~~String tmpSNSID\_KEY = data.getStringExtra("SNSID\_KEY");~~  ~~String tmpONEID\_KEY = data.getStringExtra("ONEID\_KEY");~~  ~~String tmpUSER\_ID = data.getStringExtra("USER\_ID");~~  ~~String tmpCLOSE\_ST = data.getStringExtra("CLOSE\_ST");~~  ~~}~~  ~~}~~ |

## ~~기 가입 One ID 사용 선택 응답~~

### ~~인터페이스 소개~~

~~1. 개요 : SNS ID 가입 시 email 정보와 동일한 기존 OneID 사용을 선택할 경우 서비스 앱에 전달.~~

~~2. 설명 : 로그인 시 SNS ID가 아닌 기존의 OneID 사용을 선택할 경우 서비스앱에 알린다.~~

~~(서비스 앱은 이 응답을 받았을 경우 USER\_ID를 요청에 포함하여 ID/PW로그인 요청을 한다.)~~

~~3. 프로토콜 상세~~

|  |  |
| --- | --- |
| **~~항목~~** | **~~내용~~** |
| ~~Call Type~~ | ~~Callback~~ |
| ~~Request URI~~ | ~~IntegrationConstants.RC\_SNS\_EXISTED\_ONEID\_RESPONSE~~ |

### ~~Message Specifications~~

#### ~~Response~~

| **~~매개 변수명~~** | **~~자료형~~** | **~~길이~~** | **~~필수~~** | **~~설명~~** |
| --- | --- | --- | --- | --- |
| ~~RT~~ | ~~String~~ | ~~20~~ | ~~O~~ | ~~결과 코드(문서 마지막에 코드값 첨부)~~ |
| ~~RT\_MSG~~ | ~~String~~ | ~~500~~ | ~~O~~ | ~~결과 메시지(문서 마지막에 메시지값 첨부)~~ |
| ~~USER\_ID~~ | ~~String~~ | ~~50~~ |  | ~~회원아이디(e-mail타입)~~ |

### ~~Sample~~

|  |  |
| --- | --- |
| ~~Source~~ | ~~@Override~~  ~~protected void onActivityResult(int requestCode, int resultCode, Intent data) {~~  ~~super.onActivityResult(requestCode, resultCode, data);~~  ~~if(data.getStringExtra("RESULT").equals(IntegrationConstants.RC\_SNS\_EXISTED\_ONEID\_RESPONSE)) {~~  ~~String tmpRT = data.getStringExtra("RT");~~  ~~String tmpRT\_MSG = data.getStringExtra("RT\_MSG");~~  ~~String tmpUSER\_ID = data.getStringExtra("USER\_ID");~~  ~~}~~  ~~}~~ |

## 간편인증 ONE ID의 정회원 전환 요청

### 인터페이스 소개

1. 개요 : 간편인증 One ID의 정회원전환을 위한 요청

2. 설명 : 본인인증 없이 가입된 간편인증 OneID를 외부인증(KMC, 아이핀)을 통해 정회원 OneID로 변경하기 위한 요청을 한다.

간편인증 OneID : 본인인증(자사 USIM의 경우 CAS인증, 타사의 경우 외부인증(아이핀 or 휴대폰인증)을 거치지 않고, email 인증 or SMS 인증 만으로 가입한 준회원개념의 OneID (ID\_TYPE : T01 or T02)

간편인증 OneID를 정회원 OneID로 변경하려 할 경우 호출. 변경 요청 시점은 서비스정책에 따름.

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | WebView |
| Request URI | FUNCTION\_ATHN\_USER |

### Message Specifications

#### Request

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
| SERVICE\_CD | String | 30 | O | One ID에서 부여한 서비스 코드 |
| USER\_ID | String | 50 |  | 회원아이디(e-mail타입으로 @ 앞은 ‘-‘, ‘\_’, ‘.’ 특수문자 허용) |
| ONEID\_KEY | String | 13 |  | 회원일련번호(SEQ) |
| ID\_TYPE | String | 3 |  | ID구분 (~~NULL or T00: ONE ID,~~ T01: 이메일 인증가입 ID, T02: 본인정보 입력가입 ID, …) |
| SERVICE\_LOGIN\_TYPE | String | 4 | O | 서비스에서 사용할 로그인 타입   |  |  |  |  | | --- | --- | --- | --- | | OneID 용 | | 기존ID 용 | | | 자동로그인 | 간편로그인 | 자동로그인 | 간편로그인 | | 0 or 1 | 0 or 1 | 0 or 1 | 0 or 1 |   0:미사용, 1:사용  ex) 1100 (OneID 자동,간편사용, 기존ID 자동,간편미사용) |

### Sample

|  |  |
| --- | --- |
| Source | NSMutableDictionary \*intent = [NSMutableDictionary dictionary];  [intent setObject:FUNCTION\_ATHN\_USER forKey:@"FUNCTION\_NAME"];  [intent setObject:SERVICE\_CD forKey:@"SERVICE\_CD"];  [intent setObject:@”jessica@9girls.com” forKey:@"USER\_ID"];  [intent setObject:@“1000000001” forKey:@"ONEID\_KEY"];  [intent setObject:@”T01” forKey:@"ID\_TYPE"];  [intent setObject:@"1100" forKey:@"SERVICE\_LOGIN\_TYPE"];  [Activity startActivityForResult:self class:[HybridWebView new] intent:intent requestCode:1]; |

## 간편인증 ONE ID의 정회원 전환 응답

### 인터페이스 소개

1. 개요 : 간편인증 OneID의 정회원 전환 결과를 서비스 App에 알려준다.

2. 설명 : 간편인증 OneID의 정회원 처리 결과를 성공을 응답받았을 경우 서비스별 후처리를 한다.

기존 로그인 정보를 갱신하기 위해 서비스 앱은 이 응답을 받았을 경우 기존 로그인계정을 로그아웃

처리하고 ‘간편인증 OneID의 정회원 전환 응답’의 Reponse 값인 USER\_ID를 요청에 포함하여

로그인 요청 (혹은 개별적으로 구현한 ID/PW 로그인 화면을 호출) 할 것.

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | Callback |
| Request URI | RC\_ATHN\_USER |

### Message Specifications

#### Response

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
| RT | String | 20 | O | 결과 코드(문서 마지막에 코드값 첨부) |
| RT\_MSG | String | 500 | O | 결과 메시지(문서 마지막에 메시지값 첨부) |
| ONEID\_KEY | String | 13 | O | 회원일련번호(SEQ) |
| USER\_ID | String | 50 | O | 회원아이디(e-mail타입으로 @ 앞은 ‘-‘, ‘\_’, ‘.’ 특수문자 허용) |
| CLOSE\_ST | String | 1 | O | 0:확인버튼, 1:X버튼 |

### Sample

|  |  |
| --- | --- |
| Source | - (void)viewDidAppear:(BOOL)animated  {  if([data[@"RESULT"] isEqualToString:RC\_ATHN\_USER])  {  NSString \*tmpRT = data[@"RT"];  NSString \*tmpRT\_MSG = data[@"RT\_MSG"];  NSString \*tmpONEID\_KEY = data[@"ONEID\_KEY"];  NSString \*tmpUSER\_ID = data[@"USER\_ID"];  NSString \*tmpCLOSE\_ST = data[@"CLOSE\_ST"];  }  } |

## 서비스 ID/PW 로그인 화면 요청 Callback

### 인터페이스 소개

1. 개요 : 서비스앱에서 OneID ID/PW 화면을 띄우게 하기 위해 DAS Lib가 요청함.

2. 설명 : DAS Lib에서 고객편의 등의 목적으로 ID/PW 로그인 화면을 띄워야 할 경우 서비스앱에 ID/PW로그인 화면을 호출하도록 Callback 호출을 한다.

서비스 앱은 이 응답을 받았을 경우 USER\_ID를 요청에 포함하여 로그인 요청 (혹은 개별적

구현한 ID/PW 로그인 화면을 호출) 하도록 구현.

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | Callback |
| Request URI | RC\_EXISTED\_ONEID\_RESPONSE |

### Message Specifications

#### Response

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
| RT | String | 20 | O | 결과 코드(문서 마지막에 코드값 첨부) |
| RT\_MSG | String | 500 | O | 결과 메시지(문서 마지막에 메시지값 첨부) |
| USER\_ID | String | 50 |  | 회원아이디(e-mail타입) |

### Sample

|  |  |
| --- | --- |
| Source | - (void)viewDidAppear:(BOOL)animated  {  if([data[@"RESULT"] isEqualToString:RC\_ATHN\_USER])  {  NSString \*tmpRT = data[@"RT"];  NSString \*tmpRT\_MSG = data[@"RT\_MSG"];  NSString \*tmpUSER\_ID = data[@"USER\_ID"];  }  } |

## 휴대폰 로그인 요청 (IoT, 가족지킴이 전용) (CTN 인증 로그인 요청)

### 인터페이스 소개

1. 개요 : 휴대폰 인증을 통한 로그인 요청

**IoT, 가족지킴이 등 (DAS와 별도화면 협의한 서비스)만 사용가능.**

**그 외 서비스는 [6.47 휴대폰 로그인 요청] 참조.**

2. 설명 : 휴대폰 로그인 요청을 통해 ‘휴대폰 번호 로그인’ (자사USIM은 CAS인증, 타사USIM은 외부 휴대폰인증(KMC))과 기존의 ‘U+ ONE ID 로그인(ID/PW 로그인)’ 중 선택하여 로그인 할 수 있게 한다.

**‘임시ID 생성’ 및 ‘임시ID 로그인을 허용’할 경우 반드시 VTID\_USE\_YN 을 ‘Y’로 전달해야 함. (V2.1.2 신규)**

|  |
| --- |
|  |
| 휴대폰 로그인 요청 시 DAS 화면 (IoT@Home의 예) |

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | WebView |
| Request URI | FUNCTION\_CTN\_ID\_LOGIN\_REQUEST |

### Message Specifications

#### Request

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
| SERVICE\_CD | String | 30 | O | One ID에서 부여한 서비스 코드 |
| USER\_ID | String | 50 |  | 회원아이디(e-mail타입으로 @ 앞은 ‘-‘, ‘\_’, ‘.’ 특수문자 허용)  (전달 시 ID/PW 로그인화면의 ID 입력란에 Default ID로 보여짐) |
| SERVICE\_LOGIN\_TYPE | String | 4 | O | 서비스에서 사용할 로그인 타입   |  |  |  |  | | --- | --- | --- | --- | | OneID 용 | | 기존ID 용 | | | 자동로그인 | 간편로그인 | 자동로그인 | 간편로그인 | | 0 or 1 | 0 or 1 | 0 or 1 | 0 or 1 |   0:미사용, 1:사용  ex) 1100 (OneID 자동,간편사용, 기존ID 자동,간편미사용) |
| EVENT\_CD | String | 3 |  | 로그인 화면에서 사용할 서비스 영역에서 서비스와 미리 정의된 EVENT 구분값  (ex. 001 : 홈IoT 체험하기 선택) |
| **SERVICE\_ITEM\_CODE** | **String** | **500** |  | **서비스상품코드**  **서비스상품가입 여부를 확인하기 위한 서비스 상품코드를 ‘|’ 구분으로 붙여서 전달.**  **전달한 상품코드 중에 고객이 가입되어있을 경우 ‘자사 휴대폰 로그인’ 신에서 생년월일입력 없이 로그인 진행함.** |
| **VTID\_USE\_YN** | **String** | **1** |  | **서비스 임시ID 기능사용 여부 (Y: 임시ID 기능 사용, N or NULL :임시ID 기능 사용 안함)** |

### Sample

|  |  |
| --- | --- |
| Source | NSMutableDictionary \*intent = [NSMutableDictionary dictionary];  [intent setObject:FUNCTION\_CTN\_ID\_LOGIN\_REQUEST forKey:@"FUNCTION\_NAME"];  [intent setObject:SERVICE\_CD forKey:@"SERVICE\_CD"];  [intent setObject:@“test@email.com” forKey:@"USER\_ID"];  [intent setObject:@”1000” forKey:@"SERVICE\_LOGIN\_TYPE"];  [intent setObject:@"001" forKey:@"EVENT\_CD"];  [Activity startActivityForResult:self class:[HybridWebView new] intent:intent requestCode:1]; |

## ~~ONE ID 변경 요청~~

### ~~인터페이스 소개~~

~~1. 개요 : ONE ID 변경 요청.~~

~~2. 설명 : ONE ID 변경이 필요할 경우 서비스앱에서 요청. IoT@Home의 임시 생성 ID인 @home.iot 계정만 사용이 가능함.~~

~~3. 프로토콜 상세~~

|  |  |
| --- | --- |
| **~~항목~~** | **~~내용~~** |
| ~~Call Type~~ | ~~WebView~~ |
| ~~Request URI~~ | ~~IntegrationConstants.FUNCTION\_ID\_CHANGE\_REQUEST~~ |

### ~~Message Specifications~~

#### ~~Request~~

| **~~매개 변수명~~** | **~~자료형~~** | **~~길이~~** | **~~필수~~** | **~~설명~~** |
| --- | --- | --- | --- | --- |
| ~~SERVICE\_CD~~ | ~~String~~ | ~~30~~ | ~~O~~ | ~~One ID에서 부여한 서비스 코드~~ |
| ~~ONEID\_KEY~~ | ~~String~~ | ~~13~~ | ~~O~~ | ~~회원일련번호(SEQ)~~ |
| ~~USER\_ID~~ | ~~String~~ | ~~50~~ | ~~O~~ | ~~회원아이디(e-mail타입으로 @ 앞은 ‘-‘, ‘\_’, ‘.’ 특수문자 허용)~~ |

### ~~Sample~~

|  |  |
| --- | --- |
| ~~Source~~ | ~~Intent intent = new Intent(ctCtext,HybridWebView.class);~~  ~~intent.putExtra("FUNCTION\_NAME", IntegrationConstants. FUNCTION\_ID\_CHANGE\_REQUEST);~~  ~~intent.putExtra("SERVICE\_CD", AccountUtil.SERVICE\_CD);~~  ~~intent.putExtra("ONEID\_KEY", “1000000001”);~~  ~~intent.putExtra("USER\_ID", “jessica@9girls.com”);~~  ~~startActivityForResult(intent, 1);~~ |

## ~~ONE ID 변경 응답~~

### ~~인터페이스 소개~~

~~1. 개요 : One ID 변경에 대한 응답을 받는다.~~

~~2. 설명 : One ID 변경 결과를 로그인 결과와 함께 반환한다.~~

~~3. 프로토콜 상세~~

|  |  |
| --- | --- |
| **~~항목~~** | **~~내용~~** |
| ~~Call Type~~ | ~~Callback~~ |
| ~~Request URI~~ | ~~IntegrationConstants.RC\_ID\_CHANGE\_RESPONSE~~ |

### ~~Message Specifications~~

#### ~~Response~~

| **~~매개 변수명~~** | **~~자료형~~** | **~~길이~~** | **~~필수~~** | **~~설명~~** |
| --- | --- | --- | --- | --- |
| ~~RT~~ | ~~String~~ | ~~20~~ | ~~O~~ | ~~결과 코드(~~[~~[Appendix A(결과 코드표)]~~](#_Appendix_A(결과코드)) ~~참조)~~ |
| ~~RT\_MSG~~ | ~~String~~ | ~~500~~ | ~~O~~ | ~~결과 메시지(~~[~~[Appendix A(결과 코드표)]~~](#_Appendix_A(결과코드)) ~~참조)~~ |
| ~~CHG\_USER\_ID~~ | ~~String~~ | ~~50~~ | ~~O~~ | ~~변경된 회원아이디(e-mail타입으로 @ 앞은 ‘-‘, ‘\_’, ‘.’ 특수문자 허용)~~ |
| ~~SSO\_KEY~~ | ~~String~~ | ~~100~~ | ~~O~~ | ~~통합 로그인 키~~ |
| ~~ONEID\_KEY~~ | ~~String~~ | ~~13~~ | ~~O~~ | ~~회원일련번호(SEQ)~~ |
| ~~SERVICE\_KEY~~ | ~~String~~ | ~~13~~ |  | ~~U+ Box등 서비스의 고객 Unique ID 값~~ |
| ~~NAME~~ | ~~String~~ | ~~180~~ | ~~O~~ | ~~회원이름~~ |
| ~~LGT\_TYPE~~ | ~~String~~ | ~~1~~ | ~~O~~ | ~~자/타사 구분(Y:자사, N:타사)~~ |
| ~~PW\_UPDATE\_DT~~ | ~~String~~ | ~~8~~ | ~~O~~ | ~~YYYYMMDD~~ |
| ~~ID\_TYPE~~ | ~~String~~ | ~~3~~ |  | ~~ID구분 (NULL or T00: ONE ID, T01: 이메일 인증가입 ID, T02: 본인정보 입력가입 ID, …)~~ |

### ~~Sample~~

|  |  |
| --- | --- |
| ~~Source~~ | ~~Override~~  ~~protected void onActivityResult(int requestCode, int resultCode, Intent data) {~~  ~~super.onActivityResult(requestCode, resultCode, data);~~  ~~if(data.getStringExtra("RESULT").equals(IntegrationConstants.RC\_ID\_CHANGE\_RESPONSE)) {~~  ~~String tmpRT = data.getStringExtra("RT");~~  ~~String tmpRT\_MSG = data.getStringExtra("RT\_MSG");~~  ~~String tmpUSER\_ID = data.getStringExtra("CHG\_USER\_ID");~~  ~~String tmpSSO\_KEY = data.getStringExtra("SSO\_KEY");~~  ~~String tmpONEID\_KEY = data.getStringExtra("ONEID\_KEY");~~  ~~String tmpSERVICE\_KEY = data.getStringExtra("SERVICE\_KEY");~~  ~~String tmpNAME = data.getStringExtra("NAME");~~  ~~String tmpLGT\_TYPE = data.getStringExtra("LGT\_TYPE");~~  ~~String tmpPW\_UPDATE\_DT = data.getStringExtra("PW\_UPDATE\_DT");~~  ~~String tmpID\_TYPE = data.getStringExtra("ID\_TYPE”);~~  ~~}~~  ~~}~~ |

## 이벤트 알림 Callback

### 인터페이스 소개

1. 개요 : DAS 화면 내에서 서비스와 협의한 이벤트가 발생하였을 경우, 해당 이벤트 발생을 서비스앱에 알린다.

2. 설명 : DAS 화면 내에 서비스와 협의한 이벤트가 발생하였을 경우 (ex. 로그인 화면 내에 ‘홈IoT 체험하기 버튼’이 눌렸을 경우), 해당 이벤트가 발생했음을 서비스에 알린다. 서비스 앱은 이 응답을 받았을 경우 이벤트에 해당하는 동작을 구현한다.

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | Callback |
| Request URI | RC\_EVENT\_NOTI |

### Message Specifications

#### Response

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
| RT | String | 20 | O | 결과 코드(문서 마지막에 코드값 첨부) |
| RT\_MSG | String | 500 | O | 결과 메시지(문서 마지막에 메시지값 첨부) |
| EVENT\_CD | String | 3 | O | 서비스와 미리 정의된 EVENT 구분값  (ex. 001 : 홈IoT 체험하기 선택, 002 : ONE ID 한달 뒤 변경, …) |
| EVENT\_PARAM | String | 500 |  | 서비스와 미리 정의된 EVENT별 필요한 값이 있으면 전달  (구분자 ‘|’ 로 여러 값을 전달가능) |

### Sample

|  |  |
| --- | --- |
| Source | - (void)viewDidAppear:(BOOL)animated  {  If([data[@"RESULT"] isEqualToString:RC\_EVENT\_NOTI])  {  NSString \*tmpRT = data[@"RT"];  NSString \*tmpRT\_MSG = data[@"RT\_MSG"];  NSString \*tmpEVENT\_CD = data[@"EVENT\_CD"];  NSString \*tmpEVENT\_PARAM = data[@"EVENT\_PARAM"];  }  } |

## 휴대폰 로그인 응답 (명칭변경 구. CTN 인증 로그인 응답)

### 인터페이스 소개

1. 개요 : 휴대폰 로그인 결과를 서비스 App에 알려준다.

2. 설명 : 로그인 선택화면에서 ‘휴대폰 번호 로그인’을 선택하고 로그인 하였을 경우에 로그인 결과를 서비스 App에 알려준다.

(로그인 선택 화면에서 ‘U+ONE ID로 로그인’ 선택 후 로그인 시에는 [6.12 WebView 로그인 응답]으로 로그인 결과를 전달한다. 어느 방식으로 로그인 되었는지 서비스앱에서 구분하기 위함)

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | Callback |
| Request URI | RC\_CTN\_LOGIN\_RESPONSE |

### Message Specifications

#### Response

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
| RT | String | 20 | O | 결과 코드([[Appendix A(결과 코드표)]](#_Appendix_A(결과코드)) 참조) |
| RT\_MSG | String | 500 | O | 결과 메시지([[Appendix A(결과 코드표)]](#_Appendix_A(결과코드)) 참조) |
| SSO\_KEY | String | 100 |  | 통합 로그인 키 |
| LOGIN\_TYPE | String | 1 |  | 로그인 유형(0:선택안함 1:자동로그인, 2:자동로그인+간편로그인, 3:이전선택유형과 변경없음) |
| USER\_ID | String | 50 |  | 회원아이디(e-mail타입으로 @ 앞은 ‘-‘, ‘\_’, ‘.’ 특수문자 허용) |
| ONEID\_KEY | String | 13 |  | 회원일련번호(SEQ) |
| SERVICE\_KEY | String | 13 |  | U+Box 등 서비스의 고객 Unique ID 값 |
| NAME | String | 180 |  | 회원이름 |
| LGT\_TYPE | String | 1 |  | 자타사구분(N:타사,Y:자사) |
| PW\_UPDATE\_DT | String | 8 |  | 비밀번호 마지막 갱신일 |
| TOS\_SERVICE\_CD | String | 30 |  | 추가 약관동의필수 서비스 코드 |
| ID\_TYPE | String | 3 |  | ID구분 (NULL or T00: ONE ID, T01: 이메일 인증가입 ID, T02: 본인정보 입력가입 ID, …) |
| **REQ\_LOGIN\_CTN** | **String** | **12** |  | **휴대폰 로그인 인증 시 사용한 CTN (-제외된 444 번호API)**  **(ex. 010022335678)**  **휴대폰번호 로그인 시에만 전달되며, 이 후 SSO Key를 통한인증 (ex.자동로그인요청응답(Lib), SSO 확인(API)) 응답 시에는 전달되지 않음** |
| **VTID\_YN** | **String** | **1** |  | **임시ID 여부 (Y: 임시ID, N or NULL : ONE ID)** |
| **VTID\_RQST\_RSN\_CD** | **String** | **3** |  | **임시ID 신청 사유코드** |

### Sample

|  |  |
| --- | --- |
| Source | - (void)viewDidAppear:(BOOL)animated  {  if([data[@"RESULT"] isEqualToString: RC\_CTN\_LOGIN\_RESPONSE])  {  NSString \*tmpRT = data[@"RT"];  NSString \*tmpRT\_MSG = data[@"RT\_MSG"];  NSString \*tmpSSO\_KEY = nil;  NSString \*tmpLOGIN\_TYPE = data[@"LOGIN\_TYPE"];  NSString \*tmpUSER\_ID = data[@"USER\_ID"];  NSString \*tmpONEID\_KEY = data[@"ONEID\_KEY"];  NSString \*tmpSERVICE\_KEY = data[@"SERVICE\_KEY"];  NSString \*tmpNAME = data[@"NAME"];  NSString \*tmpLGT\_TYPE = data[@"LGT\_TYPE"];  NSString \*tmpPW\_UPDATE\_DT = data[@"PW\_UPDATE\_DT"];  NSString \*tmpTOS\_SERVICE\_CD = data[@"TOS\_SERVICE\_CD"];  NSString \*tmpID\_TYPE = data[@"ID\_TYPE"];  }  } |

## 임시ID One ID 변경 요청

### 인터페이스 소개

1. 개요 : 임시ID에 대한 One ID 변경 요청.

2. 설명 : 임시ID 계정(VTID\_YN이 ‘Y’인 계정)에 대해 One ID 변경로 변경하고자 할 때 요청함.

요청 시 고객이 임시ID에 대해 ID/PW를 설정하고 ONE ID로 변경이 가능하게 한다.

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | WebView |
| Request URI | FUNCTION\_VTID\_CHANGE\_REQUEST |

### Message Specifications

#### Request

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
| SERVICE\_CD | String | 30 | O | One ID에서 부여한 서비스 코드 |
| ONEID\_KEY | String | 13 | O | 회원일련번호(SEQ) |
| USER\_ID | String | 50 | O | 회원아이디(e-mail타입, 임시ID) |

### Sample

|  |  |
| --- | --- |
| Source | NSMutableDictionary \*intent = [NSMutableDictionary dictionary];  [intent setObject: FUNCTION\_VTID\_CHANGE\_REQUEST forKey:@"FUNCTION\_NAME"];  [intent setObject:SERVICE\_CD forKey:@"SERVICE\_CD"];  [intent setObject:@“1000000001” forKey:@"ONEID\_KEY"];  [intent setObject:@”aaa@email.com” forKey:@"USER\_ID"];  [Activity startActivityForResult:self class:[HybridWebView new] intent:intent requestCode:1]; |

## 임시ID ONE ID 변경 응답

### 인터페이스 소개

1. 개요 : 임시ID를 ONE ID로 변경한 결과를 서비스 App에 알려준다.

2. 설명 : 임시ID를 ONE ID로 변경한 결과를 서비스 App에 알려준다. ONE ID로 변경 시 해당 계정의SSOKey는 모두 삭제되며, 서비스 App은 로그아웃 처리 후 다시 로그인 요청을 해야 한다.

(이 응답 후 로그인 요청 (6.10 ID/PW로그인 요청 or 6.39 휴대폰 로그인 요청) 시 요청파라미터 USER\_ID에 전달받은 USER\_ID값을 넣어 요청하면 DAS ID/PW 화면에 ID가 노출됨)

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | Callback |
| Request URI | RC\_VTID\_CHANGE\_RESPONSE |

### Message Specifications

#### Response

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
| RT | String | 20 | O | 결과 코드(문서 마지막에 코드값 첨부) |
| RT\_MSG | String | 500 | O | 결과 메시지(문서 마지막에 메시지값 첨부) |
| ONEID\_KEY | String | 13 | O | 회원일련번호(SEQ) |
| USER\_ID | String | 50 | O | 변경된 회원아이디(e-mail타입으로 @ 앞은 ‘-‘, ‘\_’, ‘.’ 특수문자 허용) |
| CLOSE\_ST | String | 1 | O | 화면 닫기 유형 (0:확인버튼, 1. X버튼) |

### Sample

|  |  |
| --- | --- |
| Source | - (void)viewDidAppear:(BOOL)animated  {  if([data[@"RESULT"] isEqualToString: RC\_VTID\_CHANGE\_RESPONSE])  {  NSString \*tmpRT = data[@"RT"];  NSString \*tmpRT\_MSG = data[@"RT\_MSG"];  NSString \*tmpONEID\_KEY = data[@"ONEID\_KEY"];  NSString \*tmpCLOSE\_ST = data[@"CLOSE\_ST"];  }  } |

## ~~권한 오류 알림~~

### ~~인터페이스 소개~~

~~1. 개요 : DAS Lib. 필수 권한 해지 시 오류 반환~~

~~2. 설명 : 권한 오류 내용을 서비스앱에 알려준다.~~

~~DAS 화면이 닫혔을 경우 서비스앱에 전달합니다.~~

~~3. 프로토콜 상세~~

|  |  |
| --- | --- |
| **~~항목~~** | **~~내용~~** |
| ~~Call Type~~ | ~~Callback~~ |
| ~~Request URI~~ | ~~RC\_PERMISSION\_ERROR~~ |

### ~~Message Specifications~~

#### ~~Response~~

| **~~매개 변수명~~** | **~~자료형~~** | **~~길이~~** | **~~필수~~** | **~~설명~~** |
| --- | --- | --- | --- | --- |
| ~~PERMISSION~~ | ~~String~~ | ~~500~~ | ~~O~~ | ~~오류발생 권한~~ |

### ~~Sample~~

|  |  |
| --- | --- |
| ~~Source~~ | ~~- (void)viewDidAppear:(BOOL)animated~~  ~~{~~  ~~if([data[@"RESULT"] isEqualToString: RC\_CTN\_LOGIN\_RESPONSE])~~  ~~{~~  ~~NSString \*tmpRT = data[@"PERMISSION"];~~  ~~}~~  ~~}~~ |

## 휴대폰 번호 로그인 요청

### 인터페이스 소개

1. 개요 : 휴대폰 번호 인증을 통한 로그인 요청

2. 설명 : 휴대폰 로그인 요청을 통해 ‘휴대폰 번호 로그인’ (자사USIM은 CAS인증, 타사USIM은 외부 휴대폰인증(KMC))을 할 수 있게 한다.

‘임시ID 기능 사용’ 설정할 경우에는 로그인 시 고객명의의 U+ID가 없을 경우 임시ID를 생성/로그인 응답 처리한다.

**‘임시ID 생성’ 및 ‘임시ID 로그인을 허용’할 경우 반드시 VTID\_USE\_YN 을 ‘Y’로 전달해야 함.**

|  |
| --- |
|  |
| ID/PW 로그인 요청과 휴대폰 번호 로그인 요청을 구현할 경우 (예시) |

\* 휴대폰번호 로그인과 임시ID 생성기능을 사용하려면 DAS에 다음의 양식으로 선등록 요청을 하여야 한다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **구분** | | **작성** | **비고** | **예시** |
| 서비스코드 | |  | U-Cube 등록 서비스코드 | A45 |
| 서비스명 | |  | U-Cube 등록 서비스명 | 홈IoT |
| DAS 적용 방법 | Android App |  | DAS Lib. / DAS Mobile Web / 미출시 | DAS Lib. |
| iOS App |  | DAS Mobile Web / 미출시 | DAS Mobile Web |
| DAS Lib.  사용 시 | 휴대폰 번호 로그인 사용여부 |  | Y/N | Y |
| 임시ID  생성 여부 |  | Y/N | Y |
| DAS Mobile Web 사용 시 | 휴대폰 번호 로그인 사용여부 |  | Y/N | Y |
| 임시ID  생성 여부 |  | Y/N | Y |

-. 임시ID 사용여부는 요청파라미터로 전달.

-. 휴대폰 번호 로그인 사용 등록을 하지않으면, 휴대폰 번호 로그인 요청 시에도 ID/PW 로그인화면 노출.

-. 휴대폰 번호 로그인을 사용하지 않으면, 임시ID로그인과 임시ID 생성이 모두 불가함.

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | WebView |
| Request URI | FUNCTION\_CTN\_LOGIN\_REQUEST |

### Message Specifications

#### Request

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
| SERVICE\_CD | String | 30 | O | One ID에서 부여한 서비스 코드 |
| SERVICE\_LOGIN\_TYPE | String | 4 | O | 서비스에서 사용할 로그인 타입   |  |  |  |  | | --- | --- | --- | --- | | OneID 용 | | 기존ID 용 | | | 자동로그인 | 간편로그인 | 자동로그인 | 간편로그인 | | 0 or 1 | 0 or 1 | 0 or 1 | 0 or 1 |   0:미사용, 1:사용  ex) 1100 (OneID 자동,간편사용, 기존ID 자동,간편미사용) |
| **SERVICE\_ITEM\_CODE** | **String** | **500** |  | **서비스상품코드 (요금제코드 or 부가서비스코드)**  **서비스상품가입 여부를 확인하기 위한 상품코드를 ‘|’ 구분으로 붙여서 전달.**  **전달한 상품코드 중에 고객이 가입되어있을 경우 ‘자사 휴대폰 로그인’ 신에서 생년월일입력 없이 로그인 진행함.** |
| **VTID\_USE\_YN** | **String** | **1** |  | **서비스 임시ID 기능사용 여부 (Y: 임시ID 기능 사용, N or NULL :임시ID 기능 사용 안함)** |

### Sample

|  |  |
| --- | --- |
| Source | NSMutableDictionary \*intent = [NSMutableDictionary dictionary];  [intent setObject:FUNCTION\_CTN\_LOGIN\_REQUEST forKey:@"FUNCTION\_NAME"];  [intent setObject:SERVICE\_CD forKey:@"SERVICE\_CD"];  [intent setObject:@“1000” forKey:@"SERVICE\_LOGIN\_TYPE"];  [intent setObject:“OOOO|OOOO|OOOO” forKey:@"SERVICE\_ITEM\_CODE"];  [intent setObject:@“Y” forKey:@"VTID\_USE\_YN"];  [Activity startActivityForResult:self class:[HybridWebView new] intent:intent requestCode:1]; |

## FIDO 추가인증 요청

### 인터페이스 소개

1. 개요 : FIDO 추가인증 요청.

2. 설명 : 서비스앱에서 직접 FIDO 인증 요청을 하고자 할 경우 호출한다. 요청 U+ID 정보기반으로 FIDO 인증 수행 후 로그인응답으로 결과를 전달한다. FIDO 등록되어 있지 않은 경우, FIDO 등록 절차를 거친다.

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | WebView |
| Request URI | FUNCTION\_2ND\_CERTIFY\_REQUEST |

### Message Specifications

#### Request

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
| SERVICE\_CD | String | 30 | O | One ID에서 부여한 서비스 코드 |
| ONEID\_KEY | String | 13 | O | 회원일련번호(SEQ) |
| USER\_ID | String | 50 | O | 회원아이디(e-mail타입, 임시ID) |

### Sample

|  |  |
| --- | --- |
| Source | NSMutableDictionary \*intent = [NSMutableDictionary dictionary];  [intent setObject:FUNCTION\_2ND\_CERTIFY\_REQUEST forKey:@"FUNCTION\_NAME"];  [intent setObject:SERVICE\_CD forKey:@"SERVICE\_CD"];  [intent setObject:@“aaa@email.com” forKey:@"USER\_ID"];  [intent setObject:@“1000000001” forKey:@"ONEID\_KEY"];  [Activity startActivityForResult:self class:[HybridWebView new] intent:intent requestCode:1]; |

## FIDO 관리화면 요청

### 인터페이스 소개

1. 개요 : FIDO 관리화면을 직접 요청을 한다.

2. 설명 : 서비스 앱의 설정화면에서 FIDO 관리화면을 직접 호출한다. (ONE ID 관리화면 내에서도 고객이 FIDO 관리화면이 노출 가능하다. [6.14. One ID 관리화면 요청]이 구현되어있다면 이 요청을 구현하지 않아도 된다.

3. 프로토콜 상세

|  |  |
| --- | --- |
| **항목** | **내용** |
| Call Type | WebView |
| Request URI | FUNCTION\_FIDO\_CONFIG |

### Message Specifications

#### Request

| **매개 변수명** | **자료형** | **길이** | **필수** | **설명** |
| --- | --- | --- | --- | --- |
| SERVICE\_CD | String | 30 | O | One ID에서 부여한 서비스 코드 |
| ONEID\_KEY | String | 13 | O | 회원일련번호(SEQ) |
| USER\_ID | String | 50 | O | 회원아이디(e-mail타입으로 @ 앞은 ‘-‘, ‘\_’, ‘.’ 특수문자 허용) |
| SERVICE\_LOGIN\_TYPE | String | 4 | O | 서비스에서 사용할 로그인 타입   |  |  |  |  | | --- | --- | --- | --- | | OneID 용 | | 기존ID 용 | | | 자동로그인 | 간편로그인 | 자동로그인 | 간편로그인 | | 0 or 1 | 0 or 1 | 0 or 1 | 0 or 1 |   0:미사용, 1:사용  ex) 1100 (OneID 자동,간편사용, 기존ID 자동,간편미사용) |

### Sample

|  |  |
| --- | --- |
| Source | NSMutableDictionary \*intent = [NSMutableDictionary dictionary];  [intent setObject:FUNCTION\_FIDO\_CONFIG forKey:@"FUNCTION\_NAME"];  [intent setObject:SERVICE\_CD forKey:@"SERVICE\_CD"];  [intent setObject:@"jun@girl.com" forKey:@"USER\_ID"];  [intent setObject:@"" forKey:@"ONEID\_KEY"];  [intent setObject:@”1000” forKey:@"SERVICE\_LOGIN\_TYPE"];  [Activity startActivityForResult:self class:[HybridWebView new] intent:intent requestCode:1]; |

# Appendix A(결과코드표)

|  |  |  |  |
| --- | --- | --- | --- |
| **결과코드** | **결과메시지** | **성공여부** | **비고** |
| 00000 | 성공 | SUCCESS |  |
| 00009 | 서비스 약관 미동의 | SUCCESS |  |
| 00011 | SSO Key 발급에 실패하였습니다. | SUCCESS |  |
| 01001 | 요청한 전문형식이 잘못되었습니다. | FAIL |  |
| 01002 | 필수 파라미터 항목이 누락되었습니다. | FAIL |  |
| 01003 | 파라미터 값이 부적절합니다. | FAIL |  |
| 01004 | API 인증에 실패했습니다. | FAIL |  |
| 01005 | 내부 오류가 발생하였습니다. | FAIL |  |
| 01006 | DB에 오류가 발생하였습니다. | FAIL |  |
| 01007 | 인증되지 않은 IP입니다. | FAIL |  |
| 02001 | U-Cube 연동 시 오류가 발생하였습니다. | FAIL |  |
| 02002 | CAS 조회 시 오류가 발생하였습니다. | FAIL |  |
| 02003 | 서비스 연동 시 오류가 발생하였습니다. | FAIL |  |
| 03001 | ID 또는 Password가 잘못되었습니다. | FAIL |  |
| 03002 | 5회 이상 비밀번호를 틀렸습니다. | FAIL |  |
| 03003 | ID가 미인증 상태입니다. | FAIL |  |
| ~~03004~~ | ~~ID 변경에 실패하였습니다.~~ | ~~FAIL~~ |  |
| 06001 | SSO\_KEY가 만료되었습니다.(개인정보 보호를 위해 자동 로그아웃 되었습니다.) | FAIL |  |
| 06002 | 계정 및 동기화에 계정정보가 존재하지 않습니다. | FAIL |  |
| 90000 | 22:00 ~ 24:00 까지 서비스 안정화 작업 중 입니다. 이용에 불편을 드려 죄송합니다. | FAIL | 문구는 config에 반영 |
| 99999 | 정의되지 않은 오류가 발생하였습니다. | FAIL |  |

# Appendix B(결과코드에 따른 서비스 App 후처리)

|  |  |  |  |
| --- | --- | --- | --- |
| **결과코드** | **결과메시지** | **고객 Noti내역** | **App 후처리** |
| 00000 | 성공 | - | N/A |
| 00009 | 서비스 약관 미동의 | N/A | 서비스 App에서 약관 동의 페이지 호출(자체 로그인 UI 구현한 경우에 해당) |
| 00011 | SSO Key 발급에 실패하였습니다. | 시스템 오류가 발생하였습니다. [00011] | 서비스 App에서 팝업 메시지 |
| 00012 | SNS ID의 email과 동일한 OneID가 존재합니다. | N/A | ID/PW 로그인 요청 (넘겨받은 EMAIL 사용) |
| 01001 | 요청한 전문형식이 잘못되었습니다. | 시스템 오류가 발생하였습니다. [01001] | 서비스 App에서 팝업 메시지 |
| 01002 | 필수 파라미터 항목이 누락되었습니다. | 시스템 오류가 발생하였습니다. [01002] | 서비스 App에서 팝업 메시지 |
| 01003 | 파라미터 값이 부적절합니다. | 시스템 오류가 발생하였습니다. [01003] | 서비스 App에서 팝업 메시지 |
| 01004 | API 인증에 실패했습니다. | 시스템 오류가 발생하였습니다. [01004] | 서비스 App에서 팝업 메시지 |
| 01005 | 내부 오류가 발생하였습니다. | 시스템 오류가 발생하였습니다. [01005] | 서비스 App에서 팝업 메시지 |
| 01006 | DB에 오류가 발생하였습니다. | 시스템 오류가 발생하였습니다. [01006] | 서비스 App에서 팝업 메시지 |
| 01007 | 인증되지 않은 IP입니다. | 시스템 오류가 발생하였습니다. [01007] | 서비스 App에서 팝업 메시지 |
| ~~01008~~ | ~~등록되지 않은 도메인 입니다.~~ | ~~시스템 오류가 발생하였습니다.[90001]~~ | ~~DAS에서 팝업 메시지~~ |
| 02001 | U-Cube 연동 시 오류가 발생하였습니다. | 시스템 오류가 발생하였습니다. [02001] | 서비스 App에서 팝업 메시지 |
| 02002 | CAS 조회 시 오류가 발생하였습니다. | 시스템 오류가 발생하였습니다. [02002] | 서비스 App에서 팝업 메시지 |
| 02003 | 서비스 연동 시 오류가 발생하였습니다. | 시스템 오류가 발생하였습니다. [02003] | 서비스 App에서 팝업 메시지 |
| 03001 | ID 또는 Password가 잘못되었습니다. | ID 또는 Password가 잘못되었습니다. | DAS에서 팝업 메시지 |
| 03002 | 5회 이상 비밀번호를 틀렸습니다. | ID 또는 Password 입력 5회 오류로 인하여 로그인이 제한 되었습니다. PW 재설정으로 Password를 초기화 하시기 바랍니다. | DAS에서 팝업 메시지 |
| 03003 | ID가 미인증 상태입니다. | ID가 미인증 상태입니다. | DAS에서 팝업 메시지 |
| 06001 | SSO\_KEY가 만료되었습니다.(개인정보 보호를 위해 자동 로그아웃 되었습니다.) | N/A | 서비스 App에서 로그인 호출 |
| 06002 | 계정 및 동기화에 계정정보가 존재하지 않습니다. | N/A | 서비스 App에서 로그인 호출 |
| 90000 | 22:00 ~ 24:00 까지 서비스 안정화 작업 중 입니다. 이용에 불편을 드려 죄송합니다. | Ex) 22:00 ~ 24:00 까지 서비스 안정화 작업 중 입니다. 이용에 불편을 드려 죄송합니다. | 서비스 App에서 팝업 메시지(메시지의 경우 Lib.에서 리턴한 MSG 그대로 사용) |
| 99999 | 정의되지 않은 오류가 발생하였습니다. | 시스템 오류가 발생하였습니다. [99999] | 서비스 App에서 팝업 메시지 |

# Appendix C(비밀번호 암호화 소스)

|  |
| --- |
| //createSHA512 -> dataFromHexString -> lowercase -> createBASE64  //패스워드를 sha512 해시 후 hex 스트링 변환 후 소문자 변환 한뒤 base64인코딩한다.  + (NSString \*) createSHA512:(NSString \*)input  {    const char \*cstr = [input cStringUsingEncoding:NSUTF8StringEncoding];  NSData \*data = [NSData dataWithBytes:cstr length:input.length];  uint8\_t digest[CC\_SHA512\_DIGEST\_LENGTH];  CC\_SHA512(data.bytes, (CC\_LONG)data.length, digest);    NSMutableString\* output = [NSMutableString stringWithCapacity:CC\_SHA512\_DIGEST\_LENGTH \* 2];  for(int i = 0; i < CC\_SHA512\_DIGEST\_LENGTH; i++)  [output appendFormat:@"%02x", digest[i]];  return output;  }  + (NSString \*) createBASE64:(NSData \*) data  {  NSString \*encoding = nil;  unsigned char \*encodingBytes = NULL;    @try {  static char encodingTable[64] = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/";  static NSUInteger paddingTable[] = {0,2,1};  // Table 1: The Base 64 Alphabet  //  // Value Encoding Value Encoding Value Encoding Value Encoding  // 0 A 17 R 34 i 51 z  // 1 B 18 S 35 j 52 0  // 2 C 19 T 36 k 53 1  // 3 D 20 U 37 l 54 2  // 4 E 21 V 38 m 55 3  // 5 F 22 W 39 n 56 4  // 6 G 23 X 40 o 57 5  // 7 H 24 Y 41 p 58 6  // 8 I 25 Z 42 q 59 7  // 9 J 26 a 43 r 60 8  // 10 K 27 b 44 s 61 9  // 11 L 28 c 45 t 62 +  // 12 M 29 d 46 u 63 /  // 13 N 30 e 47 v  // 14 O 31 f 48 w (pad) =  // 15 P 32 g 49 x  // 16 Q 33 h 50 y    NSUInteger dataLength = [data length];  NSUInteger encodedBlocks = dataLength / 3;  if( (encodedBlocks + 1) >= (NSUIntegerMax / 4) ) return nil; // NSUInteger overflow check  NSUInteger padding = paddingTable[dataLength % 3];  if( padding > 0 ) encodedBlocks++;  NSUInteger encodedLength = encodedBlocks \* 4;    encodingBytes = malloc(encodedLength);  if( encodingBytes != NULL ) {  NSUInteger rawBytesToProcess = dataLength;  NSUInteger rawBaseIndex = 0;  NSUInteger encodingBaseIndex = 0;  unsigned char \*rawBytes = (unsigned char \*)[data bytes];  unsigned char rawByte1, rawByte2, rawByte3;  while( rawBytesToProcess >= 3 ) {  rawByte1 = rawBytes[rawBaseIndex];  rawByte2 = rawBytes[rawBaseIndex+1];  rawByte3 = rawBytes[rawBaseIndex+2];  encodingBytes[encodingBaseIndex] = encodingTable[((rawByte1 >> 2) & 0x3F)];  encodingBytes[encodingBaseIndex+1] = encodingTable[((rawByte1 << 4) & 0x30) | ((rawByte2 >> 4) & 0x0F) ];  encodingBytes[encodingBaseIndex+2] = encodingTable[((rawByte2 << 2) & 0x3C) | ((rawByte3 >> 6) & 0x03) ];  encodingBytes[encodingBaseIndex+3] = encodingTable[(rawByte3 & 0x3F)];    rawBaseIndex += 3;  encodingBaseIndex += 4;  rawBytesToProcess -= 3;  }  rawByte2 = 0;  switch (dataLength-rawBaseIndex) {  case 2:  rawByte2 = rawBytes[rawBaseIndex+1];  case 1:  rawByte1 = rawBytes[rawBaseIndex];  encodingBytes[encodingBaseIndex] = encodingTable[((rawByte1 >> 2) & 0x3F)];  encodingBytes[encodingBaseIndex+1] = encodingTable[((rawByte1 << 4) & 0x30) | ((rawByte2 >> 4) & 0x0F) ];  encodingBytes[encodingBaseIndex+2] = encodingTable[((rawByte2 << 2) & 0x3C) ];  // we can skip rawByte3 since we have a partial block it would always be 0  break;  }  // compute location from where to begin inserting padding, it may overwrite some bytes from the partial block encoding  // if their value was 0 (cases 1-2).  encodingBaseIndex = encodedLength - padding;  while( padding-- > 0 ) {  encodingBytes[encodingBaseIndex++] = '=';  }  encoding = [[NSString alloc] initWithBytes:encodingBytes length:encodedLength encoding:NSASCIIStringEncoding];  }  }  @catch (NSException \*exception) {  encoding = nil;  NSLog(@"WARNING: error occured while tring to encode base 32 data: %@", exception);  }  @finally {  if( encodingBytes != NULL ) {  free( encodingBytes );  }  }  return encoding;  }  + (NSData \*) dataFromHexString:(NSString \*) target  {  NSMutableData \*data= [NSMutableData new];  unsigned char whole\_byte;  char byte\_chars[3] = {'\0','\0','\0'};  int i = 0;  NSUInteger length = target.length;    while (i < length-1) {  char c = [target characterAtIndex:i++];  if (c < '0' || (c > '9' && c < 'a') || c > 'f')  continue;  byte\_chars[0] = c;  byte\_chars[1] = [target characterAtIndex:i++];  whole\_byte = strtol(byte\_chars, NULL, 16);  [data appendBytes:&whole\_byte length:1];  }    return data;  } |