

estgames-common-framework (이스트 게임즈 아이폰 공통 라이브러리)

pod v1.0.9 license MIT platform ios

NEW 업데이트 (1.2.1)

- Localizable.string 파일을 프레임워크 안에 추가했습니다. 더 이상 추가 하지 않으셔도 됩니다. 없는 언어가 들어올 경우 영어로 기본설정 됩니다.
- region 값에 따라 유저로그인 정보를 따로 관리하게 수정했습니다.

NEW 업데이트 (1.2.0)

- EstgamesCommon
 - void setLanguage(String) 함수 추가 매개변수로 넘기는 언어코드로 언어설정, 없을 경우 기기의 기본설정 값으로 설정됩니다.
 - 배너 이미지 크기조절 모드가 변경되었습니다. 비율유지에 전체크기입니다. 비율유지라 잘리는 곳이 있습니다.
- GameAgent 추가
 - retrieveStatus(onReceived : (_ result: Bool) -> Void , onFail : (_ fail: Fail) -> Void) : API를 호출해 점검 모드인지 확인
 - 성공시 OnReceived 호출, 실패시 onFail 호출
 - OnReceived에 result값이 true면 오픈, false면 점검
 - onFail 결과값으로 오는 에러들
 - START_API_DATA_INIT : 스타트API에 오픈 유/무 URL를 받지 못함
 - API_BAD_REQUEST : 리전값이 잘못설정됨
 - API_UNKNOWN_RESPONSE : 서버에서 잘못된 값을 가져옴
 - API_REQUEST_FAIL : 네트워크 에러

NEW 업데이트 (1.1.0)

- EstgamesCommon
 - showEvent() 함수 추가 : 이벤트 페이지를 보여주는 웹뷰창입니다.
 - 배너에 웹뷰 기능 추가
 - 배너 자세히 보기 링크가 없을 경우 버튼이 나오지 않습니다.
 - 배너 자세히 보기 문구를 관리자에서 관리합니다.
 - 배너 이미지 크기조절 모드 변경
 - processShow에서 권한창이 제외되었습니다.
 - processShow에서 이용약관 미동의 시 더 이상 진행하지 않고 종료됩니다. 이 경우 estCommonFailCallBack을 호출하며 PROCESS_DENIED_CONTRACT 이 실패메시지로 들어옵니다.
- region, env 설정 추가, 아래 코드로 리전과 디버그 설정을 변경 할 수 있습니다.
 - 해당값은 EstgamesCommon 객체에 create() 함수 호출 전, UserService 객체에 startGame() 함수 호출 전에 변경하셔야 제대로 된 region, env로 나옵니다.

```
MpInfo.App.region = "mr.global.ls"
MpInfo.App.env = "live"
```

- Info.plist
 - ! MP - estapi : 기존 https://m-linker.estgames.co.kr/sdk-start-api 에서 https://m-linker.estgames.co.kr 로 수정이 필요합니다.
- Localizable.string
 - ! 'estcommon_authority_title' = "원활한 게임플레이를 위해\n아래 권한을 필요로 합니다."; 해당 키 수정 필요
 - English 번역 추가 필요

```
'estcommon_policy_title' = "Terms & Agreement";
'estcommon_policy_subTitle' = "Including the agreement for receiving advertisement";
'estcommon_policy_privacy' = "Privacy Policy";
'estcommon_policy_buttonText' = "Agree";

'estcommon_authority_title' = "Following permissions are needed for the smooth game
play.";
'estcommon_authority_confirm' = "Confirm";

'estcommon_userResult_title' = "Notification";
'estcommon_userResult_subTitle' = "Successfully Loaded";
'estcommon_userResult_titleMove' = "Move to the title page for restart";
'estcommon_userResult_confirm' = "Confirm";

'estcommon_userLoad_title' = "Load existing account";
'estcommon_userLoad_content' = "Please insert the words below if you would like to
delete the current game data([]) and load the previous game.";
'estcommon_userLoad_confirmText' = "Confirm Text: confirm";
'estcommon_userLoad_input' = "Insert";
'estcommon_userLoad_confirmButton' = "Confirm";
'estcommon_userLoad_input_wrong' = "Please insert correct words";

'estcommon_userLink_title' = "Account Interlink";
'estcommon_userLink_middelLabel' = "Previous game data exists in the
account..\nFacebookAccount: []\nWould you like to load the data?";
'estcommon_userLink_bottomLabel' = "Current playing game data([]) will be deleted";
'estcommon_userLink_confirm' = "Yes, I would like to load the previous game";
'estcommon_userLink_cancel' = "No, I would like to play the new game";

'estcommon_banner_closeButton' = "Close";
'estcommon_banner_linkButton' = "Detail view";
'estcommon_banner_oneDay' = "Don't show again today";

'estcommon_userGuest_title' = "Interlink Current Account";
'estcommon_userGuest_middle' = "Delete the previous game data([]) and interlink with the
current game.";
'estcommon_userGuest_bottom' = "\n[]";
'estcommon_userGuest_loginBt' = "Yes, I would like to login";
'estcommon_userGuest_beforeBt' = "Back to previous";
```

로컬라이징 적용방법

Localizable.string 파일을 선택하시면 오른쪽 Localization 메뉴에 로컬라이징 적용 버튼이 있습니다. 해당 버튼을 먼저 눌러주세요

프로젝트를 선택하시고 Info에 들어가시면 Localization 메뉴가 있는 데 +를 눌러서 원하시는 국가를 추가해 주시면 됩니다.



업데이트 (1.0.10)

- UserService
 - ! failCallBack은 이제 start 함수 실패에만 동작하는 콜백함수로 수정
 - ! goToLoginFailCallBack 함수 추가 goToLogin함수 동작 중 에러 시 동작
 - goToLogin(config: AWSAuthUIConfiguratrion) 함수 추가, 설정을 넘겨 로고이미지, 배경색을 변경할 수 있습니다.

- goToLogin(onComplete: (String?, String, String) -> Void, onFail : (Fail) -> Void, onCancel: () -> Void) 함수 추가
 - onComplete = goToLoginSuccessCallBack 에 설정됩니다.
 - onFail = goToLoginFailCallBack 에 설정됩니다.
 - onCancel = goToLoginCloseCallBack 에 설정됩니다.
 - 기존의 방식으로 하나하나 콜백함수를 설정하셔도 되고 해당 함수를 호출해서 한꺼번에 설정 + 계정연동 호출하셔도 됩니다.
- 계정 연동 팝업
 - 캐릭터 정보 적용, 문구 일부 수정
- 로그인 화면 커스텀 예제입니다.


```
@objc func emp_goToLogin() {
    print("[EMPNative.swift] emp_goToLogin()")
    estAccount.goToLoginCloseCallBack = emp_goToLogin_close_callback //계정연동중 중간에 X
버튼을 눌렀을 경우
    estAccount.goToLoginSuccessCallBack = emp_goToLogin_success_callback//계정연동 완료 팝업
까지 가서 닫기나, X버튼을 눌렀을 경우

    let config = AWSAuthUIConfiguration()

    config.enableUserPoolsUI = false //유저정보 저장 설정
    config.addSignInButtonView(class: AWSGoogleSignInButton.self)
    config.addSignInButtonView(class: AWSFacebookSignInButton.self) // 버튼 설정
    config.canCancel = true //취소 버튼
    config.isBackgroundColorFullScreen = true //배경색을 로고부분에만 적용할 건지 전체화면에 적용
할 건지 선택
    config.backgroundColor = UIColor.black //배경색 설정
    config.logoImage = UIImage(named: "logo-aws") //로고 이미지 설정

    estAccount.goToLogin(config: config)
    //estAccount.goToLogin()
}
```



업데이트 (1.0.9)

- EstgamesCommon
 - 객체 생성 시 생성자에서 startapi를 호출하는 방식에서 객체생성 후 crate() 함수를 호출해야 startapi를 호출하여 값을 설정합니다.
 - 이용약관 전부 동의시 show해도 더 이상 나오지 않음
 - 오류 처리는 estCommonFailCallBack 콜백함수로 리턴됩니다. ((Fail) -> Void 타입)
- UserService
 - 실패 시 호출되는 콜백함수가 failCallBack로 통합됩니다. failCallBack: (Fail) -> Void 타입으로 Fail enum 타입으로 에러코드가 리턴됩니다.
- Localizable.strings 파일에
 - 'estcommon_userLoad_input_wrong' = "정확한 단어를 입력하세요."; 를 추가해주세요
- info.plist 파일에
 - account_api = https://api-account.estgames.co.kr 추가 부탁드립니다
- Fail 에러 Enum 추가

실패 메시지	설명
START_API_NOT_CALL	startAPI 네트워크 에러
START_API_DATA_FAIL	받은 데이터 오류
START_API_DATA_INIT	값이 초기화가 안된 상태
TOKEN_EMPTY	토큰이 없음
TOKEN_CREATION	토큰 생성 시 에러
TOKEN_INVALID	
TOKEN_EXPIRED	토큰 만료
CLIENT_UNKNOWN_PROVIDER	
CLIENT_NOT_REGISTERED	
API_REQUEST_FAIL	API 호출 에러
API_ACCESS_DENIED	API 호출 권한이 없음
API_OMITTED_PARAMETER	API 호출 파라미터 에러
API_UNSUPPORTED_METHOD	
API_BAD_REQUEST	잘못된 값 리턴
API_INCOMPATIBLE_VERSION	
API_CHARACTER_INFO	캐릭터 정보 API 에러
ACCOUNT_NOT_EXIST	계정정보가 없음
ACCOUNT_ALREADY_EXIST	이미 등록된 계정
ACCOUNT_INVALID_PROPERTY	
ACCOUNT_SYNC_FAIL	계정연동 중에 싱크 에러
SIGN_AWS_LOGIN_VIEW	로그인 창 불러오는 중에 에러
SIGN_GOOGLE_SDK	구글 로그인 에러
SIGN_FACEBOOK_SDK	페이스북 로그인 에러
SIGN_AWS_SESSION	aws 세션 에러
GOOGLE_CALLBACK_EMPTY	구글 이메일 콜백함수 초기화가 안되어 있음



업데이트 (1.0.6)

- EstgamesCommon 클래스에 국가, 언어 정보 리턴 함수 추가
- getNation()
- getLanguage()



업데이트 (1.0.4)



1.0.3버전 올릴 때 문제가 생겨 1.0.4버전으로 건너뛰었습니다.

- EstgamesCommon 클래스에 생성자 콜백함수 생성
- 생성자에서 startAPI를 호출해 값을 가져와 설정을 하는 데 설정되는 시간에 **show()를 호출해 창을 불러올 경우의 문제를 차단하기 위해 생성
- Objective-c 예제 프로젝트 추가(objc_ex)

- Objective-c에서 Swift코드 사용방법 링크 <http://beankhan.tistory.com/187>



업데이트 (1.0.2)

- 토큰 리프레쉬 부분 수정
- 웹뷰(공지사항, CSCenter(문의사항, FAQ))
- EstgamesCommon클래스에 합침
- egToken 수동으로 안 넣어도 됩니다.
- showFAQ() -> showCsCenter() 로 변경되었습니다.
- Info.plist 파일
- MP 에 env 추가, 값으로 [stage|live]가 설정 될 예정이며 해당 설정에 따라 stage, live API를 호출한다.



개발에 필요한 프로젝트 설정

1

cocoapods 라이브러리 등록 (Podfile에 추가)

- 이스트 공통 모듈
- pod 'estgames-common-framework', '~> 1.0.2'
- 구글 모듈
- pod 'GoogleSignIn', '~> 4.0.0'

2

info.plist에 설정등록



아래 값은 테스트에 사용되는 값이며 실제 MR설정 값은 웹플랫폼팀에 문의 부탁드립니다.

```
<key>MP</key>
<dict>
  <key>app_id</key>
  <string>mr</string>
  <key>region</key>
  <string>mr.global.ls</string>
  <key>app_name</key>
  <string>노바워즈</string>
  <key>client_id</key>
  <string>b9b2b750-ea07-3808-a7b1-9f9ca4a9ffab.mp.estgames.com</string>
  <key>account_api</key>
  <string>https://api-account.estgames.co.kr</string>
  <key>secret</key>
  <string>15624467fd9b22c7f592de53ca92c0ed49a3ba1945a40116926c4edf1209f75c</string>
  <key>estapi</key>
  <string>https://dvn2co5qnk.execute-api.ap-northeast-
2.amazonaws.com/live/start/mr.global.ls</string>
  <key>env</key>
  <string>stage</string>
</dict>
<key>AWS</key>
<dict>
  <key>CredentialsProvider</key>
  <dict>
    <key>CognitoIdentity</key>
    <dict>
      <key>Default</key>
      <dict>
        <key>PoolId</key>
        <string>ap-northeast-2:f3e54d03-7439-43ae-a846-d4bfd7e9c980</string>
        <key>Region</key>
```

```

        <string>ap-northeast-2</string>
      </dict>
    </dict>
  </dict>
  <key>MobileHub</key>
  <dict>
    <key>ProjectClientId</key>
    <string>MobileHub fcaa397e-1a87-44ca-ae36-dbc5e538b33d aws-my-sample-app-ios-
swift-v0.19</string>
  </dict>
  <key>PinpointAnalytics</key>
  <dict>
    <key>Default</key>
    <dict>
      <key>AppId</key>
      <string>6e716976d2d04a0daf7525fb4ca03b41</string>
      <key>Region</key>
      <string>us-east-1</string>
    </dict>
  </dict>
  <key>PinpointTargeting</key>
  <dict>
    <key>Default</key>
    <dict>
      <key>Region</key>
      <string>us-east-1</string>
    </dict>
  </dict>
</dict>
<key>FacebookAppID</key>
<string>1641813522553524</string>
<key>FacebookDisplayName</key>
<string>FFG-facebook</string>
<key>CFBundleURLTypes</key>
<array>
  <dict>
    <key>CFBundleTypeRole</key>
    <string>Editor</string>
    <key>CFBundleURLName</key>
    <string>com.amazon.MySampleApp</string>
    <key>CFBundleURLSchemes</key>
    <array>
      <string>com.googleusercontent.apps.915170168211-
65if7j1chf26id1vp2d2n975caau82ik</string>
      <string>fb1641813522553524</string>
      <string>com.amazon.mysampleapp</string>
    </array>
  </dict>
</array>

```

MP

이름	타입	설명	(예)
app_id	String	앱아이디	mr
region	String		mr.global.ls
app_name	String	앱이름	노바워즈
client_id	String		b9b2b750-ea07-3808-a7b1-9f9ca4a9ffab.mp.estgames.com

이름	타입	설명	(예)
secret	String		15624467fd9b22c7f592de53ca92c0ed49a3ba1945a40116926c4edf1209f75c
estapi	String	공통 모듈에서 사용하는 api 주소	https://dvn2co5qnk.execute-api.ap-northeast-2.amazonaws.com/live/start/mr.global.ls
env	String	어떤 레벨에 API를 사용할 지 선택	stage
account_api	String	https://api-account.estgames.co.kr	

3 awsconfiguration.json 파일 추가

⚠ 아래 값은 테스트에 사용되는 값이며 실제 MR설정 값은 웹플랫폼팀에 문의 부탁드립니다.

```
{
  "UserAgent": "MobileHub/1.0",
  "Version": "1.0",
  "CredentialsProvider": {
    "CognitoIdentity": {
      "Default": {
        "PoolId": "ap-northeast-2:f3e54d03-7439-43ae-a846-d4bfd7e9c980",
        "Region": "ap-northeast-2"
      }
    }
  },
  "IdentityManager": {
    "Default": {}
  },
  "PinpointAnalytics": {
    "Default": {
      "AppId": "6e716976d2d04a0daf7525fb4ca03b41",
      "Region": "us-east-1"
    }
  },
  "PinpointTargeting": {
    "Default": {
      "Region": "us-east-1"
    }
  },
  "GoogleSignIn": {
    "ClientId-iOS": "915170168211-65if7j1chf26id1vp2d2n975caau82ik.apps.googleusercontent.com",
    "Permissions": "email,profile,openid"
  },
  "FacebookSignIn": {
    "AppId": "1641813522553524",
    "Permissions": "public_profile"
  }
}
```

4 Localizable.strings에 문자열 추가 (로컬라이징) (1.21 버전 이후 부터는 추가 하지 않아도 됩니다.)

```

'estcommon_policy_title' = "이용약관";
'estcommon_policy_subTitle' = "광고성 정보 수신 동의포함";
'estcommon_policy_privacy' = "개인 정보 취급방침";
'estcommon_policy_buttonText' = "동의합니다";

'estcommon_authority_title' = "원활한 게임플레이를 위해\n아래 권한을 필요로 합니다.";
'estcommon_authority_confirm' = "확인";

'estcommon_userResult_title' = "알림";
'estcommon_userResult_subTitle' = "불러오기 성공";
'estcommon_userResult_titleMove' = "게임 재 구동을 위해 타이틀 화면으로 이동합니다.";
'estcommon_userResult_confirm' = "확인";

'estcommon_userLoad_title' = "기존 계정 불러오기";
'estcommon_userLoad_content' = "현재 게스트 모드로 플레이 중인 데이터([ ])를\n삭제하고 기존 데이터를 불러 오시려면 아래 문자를 입력해주세요.";
'estcommon_userLoad_confirmText' = "확인 문자 : confirm";
'estcommon_userLoad_input' = "입력하기";
'estcommon_userLoad_confirmButton' = "확인";
'estcommon_userLoad_input_wrong' = "정확한 단어를 입력하세요.";

'estcommon_userLink_title' = "계정 연동";
'estcommon_userLink_middelLabel' = "입력하신 계정에 이미 플레이 중인 데이터가 있습니 다.\n\nFacebookAccount: [ ]\n위의 데이터를 불러오시겠습니까?";
'estcommon_userLink_bottomLabel' = "!현재 플레이 중인 게임데이터([ ])는 삭제됩니다";
'estcommon_userLink_confirm' = "네, 불러오겠습니다";
'estcommon_userLink_cancel' = "아니오, 새로 연동하기";

'estcommon_banner_closeButton' = "닫기";
'estcommon_banner_linkButton' = "자세히 보기";
'estcommon_banner_oneDay' = "오늘 다시보지않기";

'estcommon_userGuest_title' = "현재 계정 연동";
'estcommon_userGuest_middle' = "기존 연동된 데이터 ([ ])를 삭제하고 현재 플레이중인 게임 데이터로 계정연동 을 진행합니다.";
'estcommon_userGuest_bottom' = "\n[ ]";
'estcommon_userGuest_loginBt' = "네, 로그인합니다";
'estcommon_userGuest_beforeBt' = "이전으로 돌아가기";

```

5 AppDelegate 에 소스 코드 추가

```

import estgames_common_framework // 이스트게임즈 프레임워크 추가
import GoogleSignIn //구글 프레임워크 추가

static let remoteNotificationKey = "RemoteNotification"
var estAppDelegate: EstAppDelegate = EstAppDelegate()

func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:[UIApplicationLaunchOptionsKey: Any]?) -> Bool {
    return estAppDelegate.application(application, didFinishLaunchingWithOptions: launchOptions)
}

func application(_ application: UIApplication, open url: URL, sourceApplication: String?, annotation: Any) -> Bool {
    return estAppDelegate.application(application, open: url, sourceApplication: sourceApplication, annotation: annotation)
}

```



```
func application(_ app: UIApplication, open url: URL, options:
[UIApplicationOpenURLOptionsKey : Any] = [:]) -> Bool {
    print("google: application")

    return GIDSignIn.sharedInstance().handle(url,
        sourceApplication: options[UIApplicationOpenURLOptionsKey.sourceApplication] as?
String,
        annotation: options[UIApplicationOpenURLOptionsKey.annotation])
}
```

6 구글 이메일 계정 가지고 오는 함수 생성

```
import GoogleSignIn //구글 프레임워크 추가

func googleEmail() -> String {
    if let user = GIDSignIn.sharedInstance().currentUser {
        return user.profile.email
    } else {
        return ""
    }
}
```

💣 주의 사항 - (info.plist, Localizable.strings, awsconfiguration.json) 파일은 같은 레벨안에 있어야 하며 **Build Phases - copy Bundle Resources**에 추가 되어 있는 지 확인 바랍니다.

💣 주의 사항 - 혹시 이미지 리소스 파일을 찾지 못할 경우 이미지를 클릭하고 오른쪽 창에 **Target Membership**에서 **estgames-common-framework**로 선택되어 있는 지 확인 바랍니다.



코드 구현

에러코드

❌ Fail enum

실패 메시지	설명
START_API_NOT_CALL	startAPI 네트워크 에러
START_API_DATA_FAIL	받은 데이터 오류
START_API_DATA_INIT	값이 초기화가 안된 상태
PROCESS_DENIED_CONTRACT	processShow 처리중 이용약관을 미동의하여 중간종료 되었을 경우
TOKEN_EMPTY	토큰이 없음
TOKEN_CREATION	토큰 생성 시 에러
TOKEN_INVALID	

실패 메시지	설명
TOKEN_EXPIRED	토큰 만료
CLIENT_UNKNOWN_PROVIDER	
CLIENT_NOT_REGISTERED	
API_REQUEST_FAIL	API 호출 에러
API_ACCESS_DENIED	API 호출 권한이 없음
API_OMITTED_PARAMETER	API 호출 파라미터 에러
API_UNSUPPORTED_METHOD	
API_BAD_REQUEST	잘못된 값으로 호출
API_INCOMPATIBLE_VERSION	
API_CHARACTER_INFO	캐릭터 정보 API 에러
API_UNKNOWN_RESPONSE	API 잘못된 정보 리턴
ACCOUNT_NOT_EXIST	계정정보가 없음
ACCOUNT_ALREADY_EXIST	이미 등록된 계정
ACCOUNT_INVALID_PROPERTY	
ACCOUNT_SYNC_FAIL	계정연동 중에 싱크 에러
SIGN_AWS_LOGIN_VIEW	로그인 창 불러오는 중에 에러
SIGN_GOOGLE_SDK	구글 로그인 에러
SIGN_FACEBOOK_SDK	페이스북 로그인 에러
SIGN_AWS_SESSION	aws 세션 에러
GOOGLE_CALLBACK_EMPTY	구글 이메일 콜백함수 초기화가 안되어 있음

초기 설정

리전과 개발환경(env)를 설정한다. 설정하지 않으면 info.plist에 있는 값이 기본으로 들어가기 때문에 하지 않아도 무방하다.

```
MpInfo.App.region = "mr.global.ls"
MpInfo.App.env = "live"
```



공통으로 사용되는 창 : EstgamesCommon

이름	타입	설명
create()	void create()	StartApi를 호출, 데이터를 가져와 설정한다. 해당함수를 호출해서 데이터를 설정에 성공해야 다른 함수들이 호출된다.
initCallBack	(EstgamesCommon) -> Void	create 함수에 성공했을 때 호출되는 콜백함수
estCommonFailCallBack	(Fail) -> Void	EstgamesCommon 객체 안에 있는 함수 호출 시 실패 했을 경우 호출되는 콜백함수
processShow()	void processShow()	설정된 순서대로 배너, 이용약관..등이 출력됩니다.
processCallBack	() -> Void	processShow() 함수 호출 후 호출되는 콜백함수

이름	타입	설명
bannerShow()	void bannerShow()	배너를 출력합니다.
bannerCallBack	() -> Void	배너 호출 종료 시 호출됩니다.
authorityShow()	void authorityShow()	권한창을 출력합니다.
authorityCallBack	() -> Void	authorigyShow 호출 이후 콜백함수
policyShow()	void policyShow()	이용약관창을 출력합니다.
policyCallBack	() -> Void	policyShow호출 이후 콜백함수
contractService()	Boolean contractService()	이용약관 체크 유무(서비스)
contractPrivate()	Boolean contractPrivate()	이용약관 체크 유무(개인)
showNotice()	void showNotice()	공지사항창을 출력합니다.
showCsCenter()	void showCsCenter()	FAQ, 문의사항 창을 출력합니다.
showEvent()	void showEvent()	이벤트창을 출력합니다.
getNation()	String? getNation()	나라정보 리턴
getLanguage()	String? getLanguage()	기기의 언어정보 리턴

! 초기설정 (배너, 이용약관, 권한)관련 창

```
import estgames_common_framework    // 프레임워크 추가
//프레임 워크 선언
var estgamesCommon:EstgamesCommon!

override func viewDidLoad() {    //프레임 워크 초기화
    super.viewDidLoad()
    estgamesCommon = EstgamesCommon(pview: self) //배너, 이용약관, 권한..등을 띄울 뷰

    //EstgamesCommon에 create 함수를 호출 하고 값 설정이 성공했을 때 호출 되는 함수
    estgamesCommon.initCallBack = {(estcommon) -> Void in
        self.errorCode.text = "성공"
    }

    //EstgamesCommon 내에서 에러가 발생 했을 경우 호출되는 콜백함수
    estgamesCommon.estCommonFailCallBack = {(err) -> Void in
        switch (err) {
            case .START_API_NOT_CALL :
                self.errorCode.text = "API 호출 시 네트워크 에러"
                break
            case .START_API_DATA_FAIL :
                self.errorCode.text = "내려 받은 값 오류"
                break
            case .START_API_DATA_INIT :
                self.errorCode.text = "값이 초기화 되지 않았습니다."
                break
            case .PROCESS_DENIED_CONTRACT :
                self.errorCode.text = "이용약관을 동의해주세요"
                break
            default:

```

```

        break
    }
}

//스타트 api 호출, 내려받은 값으로 설정
estgamesCommon.create();

//processShow() 호출이 끝나고 호출하는 콜백함수
estgamesCommon.processCallBack = {() -> Void in
    self.callBack.text = "processShow 종료"
    self.con1.text = self.estgamesCommon.contractService().description
    self.con2.text = self.estgamesCommon.contractPrivate().description
}

//bannerShow() 함수 호출 이후 호출되는 콜백함수
estgamesCommon.bannerCallBack = {() -> Void in
    self.callBack.text = "bannerShow() 종료"
}

//authorityShow() 함수 호출 이후 호출되는 콜백함수
estgamesCommon.authorityCallBack = {() -> Void in
    self.callBack.text = "authorityShow() 종료"
}

//policyShow() 함수 호출 이후 호출되는 콜백함수
estgamesCommon.policyCallBack = {() -> Void in
    self.callBack.text = "policyShow() 종료"
    self.con1.text = self.estgamesCommon.contractService().description
    self.con2.text = self.estgamesCommon.contractPrivate().description
}
}

```

initCallBack, create 함수 콜백

create() 함수에서 StartAPI를 호출하여 결과를 받아 값을 만들 후 호출하는 함수, 해당 콜백이 호출 되기 전에 **show() 함수 관련 호출을 할 경우 아무 일도 일어나지 않는 다.



설정된 순서대로 (배너, 이용약관, 권한)

```

//콜백함수, 모든 로직이 완료된 후 호출하는 함수 구현
//구현하지 않으면 아무런 동작을 하지 않습니다.
//자료형 () -> Void
estgamesCommon.processCallBack = {() -> Void in}
//api 에서 제공하는 설정 순서대로 출력
estgamesCommon.processShow()

```



관련함수 및 매개 변수

이름	설명
processShow()	나라별로 설정된 순서로 출력하게 하는 함수
processCallBack: () -> Void	모든 출력이 끝난 후에 출력되는 콜백 함수



배너

```
//배너 콜백 함수
//자료형 () -> Void
estgamesCommon.bannerCallBack = {() -> Void in}
//배너 출력
estgamesCommon.bannerShow()
```

관련함수 및 매개 변수

이름	설명
bannerShow()	배너창 출력
bannerCallBack: () -> Void	배너창이 종료된 후 콜백함수

이용약관

```
estgamesCommon.policyCallBack = {() -> Void in} // policyShow 함수 종료 후에 호출되는 콜백 함수
estgamesCommon.policyShow()
estgamesCommon.contractService() //true or false
estgamesCommon.contractPrivate()
```

관련함수 및 매개 변수

이름	설명
policyShow()	배너창 출력
policyCallBack: () -> Void	이용약관창이 종료된 후 콜백함수
contractService() -> Bool	서비스 이용약관 동의 여부
contractPrivate() -> Bool	개인 이용약관 동의 여부


권한

```
estgamesCommon.authorityCallBack = {() -> Void in}
estgamesCommon.authorityShow()
```

관련함수 및 매개 변수

이름	설명
authorityShow()	권한확인창 출력
authorityCallBack: () -> Void	권한확인창이 종료된 후 콜백함수

웹뷰 관련창

-  EstgamesCommon으로 합쳐졌습니다. FAQ 호출 부분 함수 이름이 변경되었습니다. (1.0.2)
- ! 현재 웹뷰창 UI가 나오지 않아 적용되지 않았습니다.
- ! 테스트용 설정에서는 공지사항 게시판이 없어 에러가 납니다.
- ! 게임 시작 후 이지 아이디가 있어야 합니다.

공지사항

```
estgamesCommon.showNotice()
```

CSCenter(문의, FAQ)

```
estgamesCommon.showCsCenter()
```

나라, 언어 관련

```
estgamesCommon.getNation()  
estgamesCommon.getLanguage()
```



계정관련

이름	타입	설명
startGame()	void startGame()	게임시작, 토큰 생성
failCallBack	(Fail) -> Void	startGame에서 실패시 호출
goToLogin()	void goToLogin()	SNS 계정연동
goToLogin(config)	goToLogin(config: AWSAuthUIConfiguratrion)	매개변수로 설정파일을 넘겨 UI를 조정한다.
goToLogin(onComplete, onFail, onCancel)	goToLogin(onComplete: (String?, String) -> Void, onFail : (Fail) -> Void, onCancel: () -> Void)	"onComplete" = goToLoginSuccessCallBack, "onFail" = goToLoginFailCallBack, "onCancel" = goToLoginCloseCallBack 에 설정됩니다.
clearKey()	void clearKey()	로그인 정보 삭제
startSuccessCallBack	() -> Void	startGame성공 시 콜백함수
goToLoginSuccessCallBack	(egId:String?, resultType:String, provider:String) -> Void	goToLogin 성공 시 콜백함수 egId는 SNS 계정으로 연동 시에만 바뀐 아 이디가 들어옵니다. 게스트 계정으로 덮어쓰일 경우에는 nil값입니다. SNS 계정으로 연동 시 "LOGINBYSNS" 게스트 계정으로 덮어 씌우기 시 "LOGINBYFORCE" 으로 resultType값이 들어옵니다.
goToLoginFailCallBack	(Fail) -> Void	goToLogin 실패시 호출되는 콜백함수
goToLoginCloseCallBack	() -> Void	SNS 연동 중간에 X버튼을 눌렀을 경우에 콜백함수
clearSuccessCallBack	() -> Void	clearKey 함수 성공 후 콜백함수

! 유저연동 초기 설정

- ! SNS 연동창을 띄우기 위해서는 Navigate Controller를 추가해 주어야 합니다.

```
import estgames_common_framework    // 프레임워크 추가  
  
var vc : UserService!    //유저 서비스 선언
```

```

override func viewDidLoad() {
    super.viewDidLoad()

    vc = UserService(pview: self, googleEmail: googleEmail)
    //매개변수 : 유저연동 창이 나올 뷰, 구글메일(String)을 리턴하는 함수

    vc.startSuccessCallBack = {() -> Void in
        print("startGame() 함수 호출 성공 시 호출되는 콜백함수")
    }

    vc.goToLoginSuccessCallBack = {(egId, resultType, provider) -> Void in
        print("goToLogin() 함수 호출 성공 시 호출되는 콜백함수")
    }

    vc.failCallBack = {(err) -> Void in //유저 서비스 부분 에러 콜백 함수
        switch (err) {
            case .TOKEN_EMPTY :
                self.errorCode.text = "토큰이 없음"
                break
            default:
                break
        }
    }
}

/**
구글 메일주소를 리턴하는 함수
*/
import GoogleSignIn

func googleEmail() -> String {
    if let user = GIDSignIn.sharedInstance().currentUser {
        return user.profile.email
    } else {
        return ""
    }
}

```

게임시작(토큰 만들기)

```

// 게임 시작 시 호출 필요
vc.startGame()
Mpinfo.Account.userId

```

관련함수 및 매개 변수

이름	설명
startGame()	토큰생성
startSuccessCallBack: () -> Void	토큰생성 성공시 콜백 함수

토큰 생성 후 유저 정보 불러오기

호출 코드	설명
Mpinfo.Account.egId	egId

호출 코드	설명
MpInfo.Account.egToken	egToken
MpInfo.Account.refreshToken	refreshToken
MpInfo.Account.principal	principal
MpInfo.Account.provider	[guest,facebook,google]
MpInfo.Account.email	이메일
MpInfo.Account.device	device
MpInfo.Account.userId	유저 아이디



sns 계정연동(토큰이 있어야 합니다.)

```
// 게임 토큰이 필요합니다. 없을 경우 에러 발생
vc.goToLogin()

// or 직접 UI 설정
import AWSAuthUI           //AWSAuthUIConfiguration
import AWSFacebookSignIn   //페이스북 버튼
import AWSGoogleSignIn     //구글 버튼

let config = AWSAuthUIConfiguration()

config.enableUserPoolsUI = false //유저정보 저장 설정
config.addSignInButtonView(class: AWSGoogleSignInButton.self)
config.addSignInButtonView(class: AWSFacebookSignInButton.self) // 버튼 설정
config.canCancel = true //취소 버튼
config.isBackgroundColorFullScreen = false //배경색을 로고부분에만 적용할 건지 전체화면에 적용할
건지 선택
config.backgroundColor = UIColor.blue //배경색 설정
config.logoImage = nil//UIImage(named: "UserIcon") //로고 이미지 설정

vc.goToLogin(config: config)
```



관련함수 및 매개 변수

이름	설명
goToLogin()	SNS 계정연동 시작
goToLogin(config)	goToLogin(config: AWSAuthUIConfiguratrion)
goToLogin(onComplete, onFail, onCancel)	goToLogin(onComplete: (String?, String) -> Void, onFail : (Fail) -> Void, onCancel: () -> Void)
goToLoginSuccessCallBack: (egId:String?, resultType:String, provider:String) -> Void	goToLogin 성공 시 콜백함수 egId는 SNS 계정으로 연동 시에만 바뀐 아이디가 들어옵니다. 게스트 계 정으로 덮어씌울 경우에는 nil값입니다. SNS계정으로 연동 시 "LOGINBYSNS" 게스트 계정으로 덮어 씌우기 시 "LOGINBYFORCE" 으로 resultType값이 들어옵니다.
goToLoginCloseCallBack: () -> Void	SNS 연동 중간에 X버튼을 눌렀을 경우에 콜백함수



게임에이전트

이름	타입	설명
retrieveStatus	retrieveStatus(onReceived : (_ result: Bool) -> Void , onFail : (_ fail: Fail) -> Void)	API를 호출해 점검 모드인지 확인
<ul style="list-style-type: none"> 성공시 OnReceived 호출, 실패시 onFail 호출 OnReceived에 result값이 true면 오픈, false면 점검 onFail 결과값으로 오는 에러들 START_API_DATA_INIT : 스타트API에 오픈 유/무 URL를 받지 못함 API_OMITTED_PARAMETER : 리전값이 잘못설정됨 API_BAD_REQUEST : 서버에서 잘못된 값을 가져옴 API_REQUEST_FAIL : 호출실패 		

! 초기설정

```
var gameAgent: GameAgent = GameAgent()
```

함수 호출 (예)

```
gameAgent.retrieveStatus(onReceived: {(result) -> Void in
    self.errorCode.text = result.description
}, onFail: {(fail) -> Void in
    print(fail)
})
```