



ADUCID Mobile Application Integration

Version 3.1.0.RC3

Release date

June 17, 2016

Table of Contents

Contents

Table of Contents	2
1. Server side	3
1.1. Creating custom server responses	3
2. Android implementation of Papi	3
2.1. How to make an authentication request?	3
2.2. Responses	4
2.3. Retrieve data when <i>authenticated</i>	4
2.3.1. Try to get data if not authenticated	5
2.4. Advanced requests	5
3. iOS implementation of Papi	5
3.1. Description	5
3.2. How integrate Papi framework	5
3.3. How authenticate with PEIG	6
3.4. Response ApiResult	7
3.5. Handle result from PEIG	8
3.6. Retrieving data when it is authenticated	8
3.6.1. Try to get data if it is not authenticated	8
3.7. How PEIG starts mobile application	9

This document describes integration of mobile application (Android, iOS) with ADUCID authentication. Two components are used: client side API called Papi (PEIG API) and server side ADUCID Web SDK.

1. Server side

Application based on ADUCID® Java SDK Web Platform provides basic authentication at URL:

```
/authenticateRemote
```

This URL is called using method `aducidAuthentication` in Papi.

To finish authentication the URL below is called:

```
/authenticateRemoteCheck
```

If you'd like to write your own application either provide these two URL using ADUCID® Java SDK Web Platform (see ADUCID® Java SDK manual) or use `aducidFullOperation` in Papi (see below). You can invoke a confirm dialog, create replicas, refresh identity, etc.

1.1. Creating custom server responses

Papi communicates with server using JSON format using these properties:

- **status:** send authentication result to Papi
 - OK
 - ERROR / other
- **data:** send data to be processed by Papi
 - Uri for PEIG
 - Custom message
- **redirect:** target Uri to retrieve result from
- **exception:** standard ADUCID exception

2. Android implementation of Papi

See *DemoBank MainFragment.java* example

Prerequisites: used external libraries:

- <https://github.com/loopj/android-async-http>
- <https://github.com/smarek/httpclient-android>

ADUCID PEIG API (Papi) provides simple way how to integrate ADUCID to custom client application. To keep session cookies, Papi uses http async client. (`papi.getAsyncClient()`)

2.1. How to make an authentication request?

1. First create an instance of Papi
(only one per application / set of requests)
2. Prepare a handler which is going to be called when Papi is finished (see below)
3. Call Papi method `aducidAuthentication ()`:

```
aducidAuthentication(String url, Handler handler)
```

- **url** is address of your **application** including protocol, e.g.
https://demobank.aducid.com/demobank
- **handler** is code that is called to evaluate authentication

In your Handler override **handleMessage**(Message msg) and get the result:

```
PapiResult result = msg.getData().getParcelable("result");
```

The evaluate it - see Responses:

2.2. Responses

Your code should evaluate these possible outcomes:

I.

```
if result.isOK()
```

Your session is authenticated

In DemoBank, result.getRedirect() contains link to page with account data

II.

```
else if (result.peigNotInstalled())
```

PEIG is not installed. To fix it, you can display a dialog and then open GooglePlay to install it:

```
getPapi().installPEIG(getActivity());
```

III.

```
else if (result.isNetworkError())
```

Communication error occurred – either phone has no connectivity or destination cannot be reached (wrong URL or server is not running).

IV.

```
else {other exception / redirect / state}
```

Custom application errors, authentication failed!

- result.getException() contains ADUCID exception, there are possible states:
 - result.notAcceptedByUser(): user rejected
 - result.isFactorAbsent(): user doesn't have personal factor (and server requires it)
- result.getData() contains error Message / reason
- result.getRedirect() contains page that could be opened using browser / contains additional content to download/ display to the user.

See PapiResult class for all details

2.3. Retrieve data when *authenticated*

In case I. your result is "OK" and your mobile application is authenticated. Use this session to contact your application server. AsyncHttpClient is used as http client to keep session cookies and GET/POST/PUT/DELETE operations. Following example shows how to use this client to GET a page when authenticated:

```
getPapi().getAsyncClient().get(String url, RequestParams parameters,
ResponseHandlerInterface responseHandler)
```

- **url** – target page / method URL, e.g. https://demobank.aducid.com/demobank/**payment**

- **parameters** – specify https(s) request parameters for your request if needed
- **AsyncHttpResponseHandler** – callback to be executed when data is loaded

2.3.1. Try to get data if not authenticated

To check your setup, test if you get error response from the server

2.4. Advanced requests

To accomplish a specific action like transaction use `aducidFullOperation()`

See DemoBank PaymentFragment.java example

```
aducidFullOperationPost(String url, String methodName, RequestParams parameters, Handler handler)
```

- **url** is address of your **application** including protocol, e.g.
`https://demobank.aducid.com/demobank`
- **methodName** is path to required page / method (e.g. `"/remotePayment"`)
- **parameters** are set of parameters which will be sent to the server, null if there is no parameter
- **handler** – see above – your callback after ADUCID is done

Get the result:

```
PapiResult result = msg.getData().getParcelable("result");
```

And parse it – see responses.

3. iOS implementation of Papi

3.1. Description

Integration of Papi for iOS platform is implemented by using framework in targeted application. Framework, named as *PeigApiLib.framework*, is part of the package. This framework is archived as ZIP file. Must be extracted on OSX system, because contains **symlinks** (symlinks doesn't work on Windows system).

It is recommended to use Xcode 6.4 for using PeigApiLib framework.

3.2. How integrate Papi framework

Framework *PeigApiLib.framework* must be included into Xcode project, in project settings, choose Build Phases, section Link Binary With Libraries, see below image.

▼ Link Binary With Libraries (20 items) ×

Name	Status
 PeigApiLib.framework	Required ⬆
 AddressBook.framework	Required ⬆
 Security.framework	Required ⬆
 Social.framework	Optional ⬆
 Accounts.framework	Optional ⬆
 AdSupport.framework	Optional ⬆
 AVFoundation.framework	Required ⬆
 CoreVideo.framework	Required ⬆
 MediaPlayer.framework	Required ⬆
 QuartzCore.framework	Required ⬆
 SystemConfiguration.framework	Required ⬆
 CoreFoundation.framework	Required ⬆
 CFNetwork.framework	Required ⬆
 CoreLocation.framework	Required ⬆
 libz.dylib	Required ⬆

Framework contains next important classes (public headers):

- PeigApiLib.h - Contains main method for call ADUCID authentication and for another ADUCID operations
- LibServiceDelegate.h - Allow send message to another object, that ADUCID authentication (or another operation) is completed, Success or Failed
- ApiResult.h - Class manages ADUCID authentication result

In the source code, use the following import

```
#import <PeigApiLib/LibServiceDelegate.h>
```

```
#import <PeigApiLib/ApiResult.h>
```

Class LibServiceDelegate has two methods:

```
-(void) onSuccess:(ApiResult *) result;
-(void) onFailed:(NSString *) error;
```

3.3. How authenticate with PEIG

First, import class:

```
#import <PeigApiLib/PeigApiLib.h>
```

Thereafter call method:

```
[[PeigApiLib sharedInstance] aducidAuthenticate:url withDelegate:delegate
withDataSourceCompletionHandler:^(ApiResult* result) {

    } withDataSourceErrorHandler:^(NSString *error) {

    }
];
```

where:

- url - is **NSString**, in format [protocol]://[domain]/ebank/, in DemoBank application is <https://demobank.aducid.com/demobank/>
- delegate - is **id<LibServiceDelegate>**, typically reference to ViewController, see DemoBank, class LoginViewController.h

LoginViewController implements LibServiceDelegate and methods:

```

-(void) onSuccess:(ApiResult *) result
{
    ...
}

-(void) onFailed:(NSString *) error
{
    ...
}

```

See DemoBank application, class LoginViewController and other, eg. CHService.

To the evaluate ApiResult - see chapter 3.4.

3.4. Response ApiResult

Your code should evaluate these possible outcomes:

- if [result isOK]
 - Session is authenticated.
- else if [result peigNotInstalled]
 - PEIG is not installed, code


```
[[PeigApiLib sharedInstance] openAppStoreDetail]
```

 opens ITUNES detail page.
- else
 - other exception / redirect / state.
 - Custom application errors, authentication failed!
 - result.exception contains ADUCID exception, there are possible states:
 - [result notAcceptedByUser]: user rejected
 - [result isFactorAbsent]: user doesn't have personal factor (and server requires it)
 - result.data contains error Message / reason

- `[result doRedirect]`, `result.redirect` contains page that could be opened using browser / contains additional content to download / display to the user.

See `ApiResult` class for all details.

3.5. Handle result from PEIG

`PeigApiLib` works with URI scheme. It is important to forward URI request from PEIG to `PeigApiLib`. Please, see class `THAppDelegate` in `DemoBank` application.

```
- (BOOL)application:(UIApplication *)application handleOpenURL:(NSURL *)url
{
    [[PeigApiLib sharedInstance] handleOpenURL:url];
    return YES;
}
```

Method `handleOpenURL` is defined in protocol class `UIApplicationDelegate`.

3.6. Retrieving data when it is authenticated

In case `ApiResult` is "OK" and mobile application is authenticated, use this session to contact your application server by using `PeigApiLib` library.

You can use GET or POST operations. See `PeigApiLib` class.

```
- (void) getAducidOperation:(NSString*)url withParameters:(id)params
withDelegate:(id<LibServiceDelegate>)delegate withDataSourceCompletionHandler:(void (^)(
(ApiResult*))successDataHandler withDataSourceErrorHandler:(void
(^)(NSString*))failureDataHandler;
```

```
- (void) postAducidOperation:(NSString*)url withParameters:(id)params
withDelegate:(id<LibServiceDelegate>)delegate withDataSourceCompletionHandler:(void (^)(
(ApiResult*))successDataHandler withDataSourceErrorHandler:(void
(^)(NSString*))failureDataHandler;
```

url - target page or method URL, e.g. `https://demobank.aducid.com/demobank/payment`

params - reference to object with parameters, e.g. `NSDictionary` object in `DemoBank` application, class `CHService`

delegate - reference to delegate for callback when data is loaded

Please, have a look at class `PaymentViewController` in `DemoBank` application. There is the example how to use POST request.

3.6.1. Try to get data if it is not authenticated

Check your setup, test if you get error response from the server.

3.7. How PEIG starts mobile application

PEIG sends response back to mobile application via URI scheme. Please, define URI scheme for your mobile application in *info.plist* file. For DemoBank application is defined as *aducid-demobank*. See below image.

▼ URL types	▲▼	Array	(2 items)
▼ Item 0 (Editor)		Dictionary	(2 items)
Document Role	▲▼	String	Editor
URL identifier	▲▼	String	callback
▼ Item 1 (Editor)		Dictionary	(2 items)
Document Role	▲▼	String	Editor
▼ URL Schemes	▲▼	Array	(1 item)
Item 0		String	aducid-demobank