UCOPIA

TURN YOUR WI-FI UP

UCOPIA portal API

Version 5.1

1 Table of contents

2 Introduction

This guide is intended for developers who want to integrate the features of the UCOPIA portal in a third-party application.

The UCOPIA portal API enables users to be registered using differents modes (One Click, Open access, etc.), authenticated, disconnected, as well as having account feedbacks (such as remaining time credit, etc.)

2.1 Global behavior (connection flow)

At the beginning end user browser tries to get Internet access by typing an URL using HTTP(S), this request will be captured by the UCOPIA controller in order to doing some verifications to derived the suitable portal settings, then redirected to the API which enables users to be authenticated or registered.

Getting Internet access at the beginning is similar to requesting the following URL:

```
https://<@UCOPIA controller host>/index.php
```

<@UCOPIA controller host>: can be the FQDN or the IP address of the UCOPIA controller.

This global flow sequence is illustrated in Figure 1.

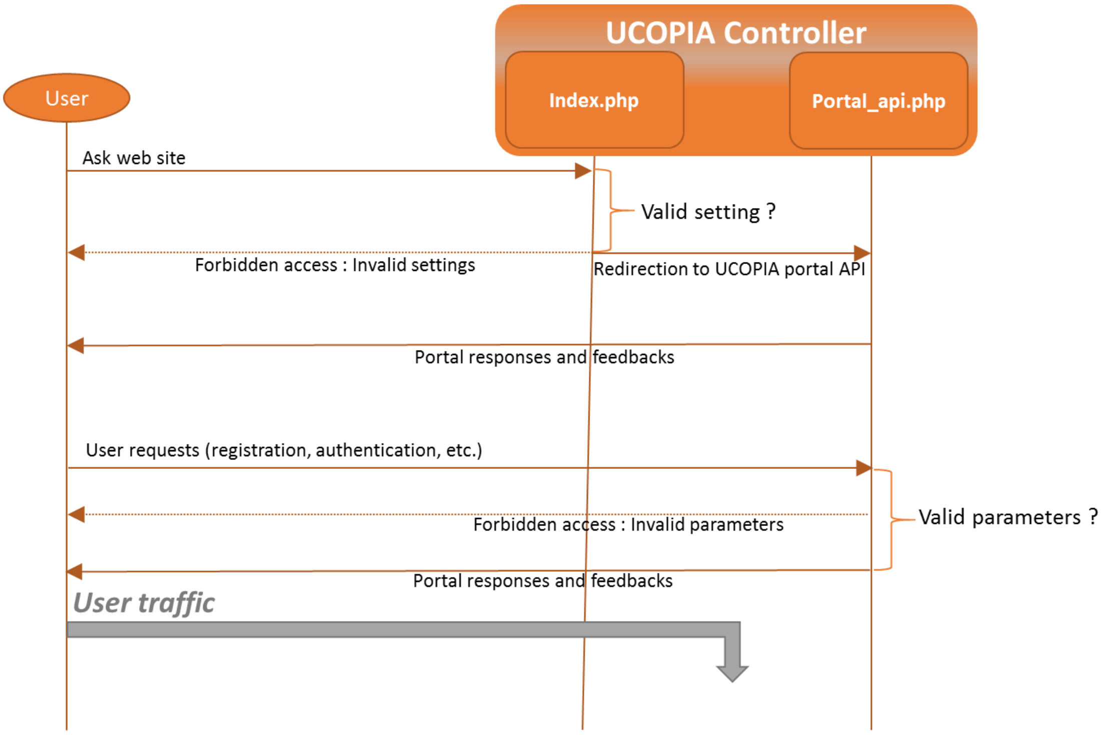

UCOPIA Controller
Index.php
Portal_api.php
User
Ask web site
Valid setting ?
Forbidden access : Invalid settings
Redirection to UCOPIA portal API
Portal responses and feedbacks
User requests (registration, authentication, etc.)
Valid parameters ?
Forbidden access : Invalid parameters
Portal responses and feedbacks
User traffic

Figure 1 : Connection Flow

Warning: index.php after checking settings create a SESSION_ID (PHP SESSION) which will be returned to the user into the session cookie. User must keep this cookie and reuse it for each request throughout the same session.

2.2 Error codes linked to an invalid settings

Before redirection to the API the UCOPIA controller checks the following settings:

- **HTTP_USER_AGENT**: UCOPIA controllers apply some restrictions on the user agent string (See Section 2.3).
- **INCOMING NETWORK**: UCOPIA controllers do some verifications on the user incoming network.
- **INCOMING ZONE AND ASSOCIATED PORTAL**: UCOPIA controllers do some verifications on the user incoming zone and the configured associated portal.

Based on all those valid settings UCOPIA controller redirect user to the API with all needed information. Else, the user will receive the HTTP error responses (See Table **1: HTTP error codes linked to an invalid settings**).

Table 1: HTTP error codes linked to an invalid settings

HTTP errors	Error message	Possible raisons
401	No portal is defined for this zone	Portal and zone configuration defined by the administrator is wrong
	You cannot access to the captive portal via this interface	The client is not connected through an incoming network
	You cannot access to captive portal of this controller	No license or passive redundancy controller
	Service Temporarily Unavailable. Please try later	Internal error, may be an LDAP access error
	An error occurred making the service unavailable. Please contact the network administrator.	May be a DNS problem
403	This user agent is not allowed to access the portal	User agent string is not authorized See Section 3.3
	Your current web browser cannot access this page	
404	The page you are looking for does not exist	Null or empty IP Address

2.3 HTTP user agent restrictions

For performance reason, the following rules regarding the user agent recognition must be respected.

- User agent string length not greater than 60 characters
- User agent string do not start with :
  - “iPhone”
  - “iPad”
  - “iPod”
  - “iPad”
  - “Apple-iPhone”
  - “Apple-iPad”
  - “Apple-iPod”
  - “Android”
  - “Skype WISPr”
  - “GeoServices”
  - “Mozilla”
  - “Opera”
  - “BlackBerry”

Once the session is open and the API is ready to serve requests, UCOPIA application can now send requests to interact with the API.

3 API description

The API is in the form of an HTTPS query using POST parameters with the following syntax:

```
https://<@UCOPIA_controller_host>/portal_api.php
```

All UCOPIA controller API responses are encoded in JSON format.

Regarding HTTPS security, the third party application can use:

- HTTPS without certificate check. In this case, all traffic is encrypted, but the UCOPIA controller is not authenticated. With this behavior, the third party application will work even if the UCOPIA controller name and/or certificate are modified.

- Strict certificate check. The controller name and the SSL must match.

The available primitives of the API are the following:

- Initialization (get information)
- User authentication
- User refresh
- User disconnection
- User registering

The following symbols will be used through all the specification to identify optional and mandatory fields:

Table 2 : optional and mandatory fields

Symbol	Description
!	Optional, could be missing, depending on the configuration defined by the administrator
!	Always Optional
none	Mandatory

The following string classes are defined using regular expression to validate the authorized characters of some form field values:

Table 3: Allowed chars

<table>
<tr><th>Class</th><th>Allowed chars</th></tr>
<tr><td>s-string</td><td>'^[a-zA-Z0-9àáâãäåæçèéêëìíîïðñòóôõöøùúûüýÿÀÁÂÃÄÅÆÇÈÉÊËÌÍÎÏÐÑÒÓÔÕÖØÙÚÛÜÝß\\-@_\\´\\\'\\./&]+$’</td></tr>
<tr><td>s-phone</td><td>‘^[0-9]+$’</td></tr>
<tr><td>s-email</td><td>'^[-a-zA-Z0-9!#$%&\'*+\\/=?^_`{|}~]+(\\.[-a-zA-Z0-9!#$%&\'*+\\/=?^_`{|}~]+)*@(([a-zA-Z0-9]([-a-zA-Z0-9]*[a-zA-Z0-9]+)?){1,63}\\.)+([a-zA-Z0-9]([-a-zA-Z0-9]*[a-zA-Z0-9]+)?){2,63}$'</td></tr>
</table>

3.1 Initialization (get information)

Get information from the UCOPIA controller.

- Parameters
  - Mandatory
    - **action** = init
  - Optional
    - None
- Result: API response includes several parameters: some are mandatory and others optional, depending on the configuration defined by the administrator.

  Array Description: (See Appendix 1: API “INIT” example for the complete “INIT” example.)

logon[hash]:	global parameters		
	policy [hash]: define a global service usage policy		
		display [boolean]:	**TRUE|FALSE**. Policy should be displayed for the user agreement?
		! displayLang []:	The available languages for display of a portal, See Table 4. Available languages for display of the portal.
		! <lang key> [hash]:	See Table 4. Available languages for display of the portal.
			! text [string]: Text to display
			! link [string]: A link to an URL
		! defaultLang [string]:	**<lang key>**: the default language
			! text [string]: Text to display
			! link [string]: A link to an URL
subscribe [hash]: available subscription modes. Possible values are : "one" | "direct"			
	count [integer]: number of available subscription modes.		
	One [hash]: 'One-Click Button' registration: all following fields are either optional or mandatory depending on the configuration defined by the administrator, but if one of them is allowed and mandatory the UCOPIA application must send it with the valid value		
		! userLogin [hash]: user identifier options, depending on the configuration defined by the administrator. If it is displayed and mandatory, the user must specify a login. If it is displayed and not mandatory, the login will be generated if not specified by the user	
			display [boolean]: **TRUE|FALSE**. userLogin should be displayed?
			mandatory [boolean]: **TRUE|FALSE** . userLogin should be mandatory?
		! userPassword [hash]: if it is displayed and mandatory, the user must specify a password. If it is displayed and not mandatory, the password will be generated if not specified by the user.	
			display [boolean]: **TRUE|FALSE**. userPassword should be displayed?

			mandatory [boolean]: **TRUE|FALSE**. userPassword should be mandatory?
		! **firstName [hash]:** user first name	
			display [boolean]: **TRUE|FALSE**. firstName should be displayed?
			mandatory [boolean]: **TRUE|FALSE**. firstName should be mandatory?
		! **lastName [hash]:** user last name	
			display [boolean]: **TRUE|FALSE**. LastName should be displayed?
			mandatory [boolean]: **TRUE|FALSE**. LastName should be mandatory?
		! **email [hash]:** user email	
			display [boolean]: **TRUE|FALSE**. email should be displayed?
			mandatory [boolean]: **TRUE|FALSE**. email should be mandatory?
		! **phone [hash]:** user phone number	
			display [boolean]: **TRUE|FALSE**. phone should be displayed?
			mandatory [boolean]: **TRUE|FALSE**. phone should be mandatory?
		! **organizationalUnitName [hash]:** Organizational unit name	
			display [boolean]: **TRUE|FALSE**. OrganizationalUnitName should be displayed?
			mandatory [boolean]: **TRUE|FALSE**. OrganizationalUnitName should be mandatory?
		! **postalAddress [hash]:** postal address.	

		⚠ **Warning**: if postalAddress is displayed and mandatory then the three following fields (postal-code [s-string], postal-locality-name[s-string(See Table 3: Allowed chars)], postal-country-name[s-string]) will be displayed and mandatory, and the two following fields : postal-postoffice-box[s-string] and postal_state_or_province_name[s-string] will be only displayed.		
			display [boolean]: **TRUE|FALSE**. Postal address should be displayed?	
			mandatory [boolean]: **TRUE|FALSE**. Postal address should be mandatory?	
		! **personalField_number [hash]**: number can be {1,2,3}		
			display [boolean]: **TRUE|FALSE**. personalField_ number should be displayed?	
			mandatory [boolean]: **TRUE|FALSE**. personalField_ number should be displayed mandatory?	
		! **policy [hash]**: define a policy governing the use of personal data (**private policy for current mode**)		
			display [boolean]: TRUE|FALSE. Private policy should be displayed for the user agreement?	
			! **displayLang []:** The available languages for display of a portal (See Table 4. Available languages for display of the portal.)	
			! **<lang key>** [hash]:	
				! **text [string]**: Text to display
				! **link [string]**: A link to an URL
user [hash]:				
	isAuthenticatedThroughShibboleth [boolean]: TRUE|FALSE. Authentification Shibboleth mode is enabled?			
	lang [<lang key>]: user default lang			
	incomingNetwork [hash]:			
		value [string]: incoming network label		

	incomingNetworkID [hash]:	
		value [string]: incoming network ID
	incomingZone [hash]:	
		value [string]: incoming zone label
auth_modes [hash]: set of all available authentication modes		
	standard [boolean]: **TRUE|FALSE** (login/password)	
	one [boolean]: **TRUE|FALSE** (One-Click)	
feedback [hash]: set of information to display or not after authentication success		
	caution [hash]:	
		display [boolean]: TRUE|FALSE. The following caution message "closing this window will disconnect you" should be displayed?
	requestedURL [hash]: User requested URL	
		display [boolean]: **TRUE|FALSE.** Should be displayed?
	login [hash]: User login	
		display [boolean]: **TRUE|FALSE.** Should be displayed?
	profile [hash]: User profile	
		display [boolean]: **TRUE|FALSE.** Should be displayed?
	services [hash]: Authorized services list	
		display [boolean]: **TRUE|FALSE.** Should be displayed?
	ipAddress [hash]: User IP address	
		display [boolean]: **TRUE|FALSE.** Should be displayed?
	incomingNetwork [hash]: User incoming network	
		display [boolean]: **TRUE|FALSE.** Should be displayed?
	incomingZone [hash]: User incoming zone	
		display [boolean]: **TRUE|FALSE.** Should be displayed?
	incommingVlan [hash]: User incoming VLAN	
		display [boolean]: **TRUE|FALSE.** Should be displayed?

	multidevice [hash]: Number of authorized user devices	
		display [boolean]: **TRUE|FALSE.** Should be displayed?
	validity [hash]: User validities	
		display [boolean]: **TRUE|FALSE.** Should be displayed?
	forceDiscTimer [hash]: Force user disconnection option	
		display [boolean]: **TRUE|FALSE.** Should be displayed?
	totalTimeCredit [hash]: User initial time credit	
		display [boolean]: **TRUE|FALSE.** Should be displayed?
	totalConsumedTimeCredit [hash]: User consumed time credit	
		display [boolean]: **TRUE|FALSE.** Should be displayed?
	timeCredit [hash]: All user time credit details	
		display [boolean]: **TRUE|FALSE.** Should be displayed?
	consumedData [hash]: User consumed data (bandwidth)	
		display [boolean]: **TRUE|FALSE.** Should be displayed?
pwdRecovery [boolean]: **TRUE|FALSE.** User can try to recover his password?		
lang [hash]: Available languages, depending on the configuration defined by the administrator		
	displayLang [<lang key>]: key of each language (See Table 4. Available languages for display of the portal.)	
	defaultLang [<lang key>]: default language	
	customFieldsDefaultLang [<lang key>]: custom fields default language, can be missing (NULL)	
refreshInterval [integer]: by default 50000 milliseconds, interval of user refresh connection, depending on the configuration defined by the administrator		
step [string]: choice of the next authentication step, possible values: {“LOGON”, “SUBSCRIBE”, “FORCE_MODIFY_PWD”, “FEEDBACK”}		
type [string]: authentication step type, possible values: {“ONE”, “REFRESH”, “CONNECT”, “”}		
modifyPwd [boolean] : **TRUE|FALSE.** User must modify its password after the first connection? depending on the configuration defined by the administrator		

forceModifyPwd [boolean] : **TRUE|FALSE.** User may modify its password after the first connection ? depending on the configuration defined by the administrator	
postConnectRedirect [hash]: redirect user once connected? depending on the configuration defined by the administrator	
	url [string]: redirection URL
	delay [integer]: redirect after a delay [0-60 seconds]

Table 4. Available languages for display of the portal.

<lang key>	Corresponding language
ar	**Arabic**
de	**German**
fr	**French**
en	**English**
es	**Spanish**
it	**Italian**
nl	**Dutch**
pt	**Portuguese**
pl	**Polish**
zh_CN	**Simplified Chinese**
zh_TW	**Traditional Chinese**
ko	**Korean**
ja	**Japanese**
th	**Thai**
id	**Indonesian**
ru	**Russian**

3.2 User authentication

Authenticate a user.

- Parameters (mandatory)
  - action = **authenticate**
  - login = **user identifier**
  - password = **user password**
- Optional
  - policy_accept = **TRUE|FALSE**

- Result : API authentication response is either a returned error code Table 5 : authentication error codes, or the following array (See Appendix 2: API “AUTHENTIFICATION” example for the complete “authentication” example.):

authenticate_step [string]: last authentication step type, possible values: {“ONE”, “REFRESH”, “CONNECT”, “”}		
authenticate_type [string]: last authentication type, possible values: {“ONE”, “REFRESH”, “CONNECT”, “”}		
step [string]: Current of the next authentication step, possible values: {“LOGON”, “SUBSCRIBE”, “FORCE_MODIFY_PWD”, “FEEDBACK”}		
type [string]: authentication step type, possible values: {“ONE”, “REFRESH”, “CONNECT”, “”}		
user [hash]:		
	login [hash]: gather all user account information	
		value[s-string]: user identifier (See Table 3: Allowed chars)
	ipAddress [hash]:	
		value[string]: user IPv4 address
	profile [hash]:	
		value[string]: profile name
	service [hash]:	
		Value [string]: authorized services separated by coma (,)
	autoDisconnect [boolean] : the user’s device is in “automatic disconnection” mode ?	
	schedule [hash] : Authorized connection time slots	
		begin [hash]:

			day [string] : starting day
			hour [string] : starting hour (24-00)
			min [string] : starting minute (00)
		end [hash]:	
			day [string] : starting day
			hour [string] : starting hour (24-00)
			min [string] : starting minute (00)
	validity [hash]: validity duration (in seconds from epoq)		
	initTimeGMT [hash]:		
		value [integer]: initialization time of the current session	
	timeCredit [hash]:		
		value [integer]: total time credit (in seconds)	
		remaining [hash]:	
			value [integer]: current remaining time credit (in seconds)
		reneweach [hash]:	
			value [integer]: interval to re-initializing the time credit (in days)
		initialRemaining [hash]:	
			value [integer]: initial remaining time credit (in seconds)
	macAddress [hash]:		
		value [string] : user Mac address	
	incomingNetwork [hash]:		
		value [string] : incoming network label	
	incomingNetworkID [hash]:		
		value [string] : incoming network ID	
	incomingZone [hash]:		

		value [string] : incoming zone
	incomingVlan [hash]:	
		value [string] : incoming VLAN
	creationMode [hash]:	
		value [string] : define the users mode of creation, possible values : { “DELEG”, “portal-one”, “portal-paypal”, “portal-ogone”, “portal-pps”, “portal-direct”}
	universalTime [hash]: current time (in seconds from epoq) for the controller	
		value [string] : user Mac address
	timezoneOffse[hash]:	
		value [string] : time zone offset (in seconds) for the controller
	requestedURL [hash]:	
		value [string] : user requested URL
	lastName [hash]:	
		value [s-string] : user last name (See Table 3: Allowed chars)
	firstName [hash]:	
		value [s-string] : user first name (See Table 3: Allowed chars)
	email [hash]:	
		value [s-email] : user email (See Table 3: Allowed chars)
	phoneNumber [hash]:	
		value [s-phone] : user phone number (See Table 3: Allowed chars)
	organizationalUnitName [hash]:	
		value [s-string] : user organizational unit name
	countryName [hash]:	
		value [s-string] : user country name
	localityName [hash]:	
		value [s-string] : user locality name
	postalCode [hash]:	

		value [s-string] : user postal address code	
	postalAddress [hash]:		
		value [s-string] : user postal address	
	postOfficeBox [hash]:		
		value [s-string] : user postal office box	
	customFields [hash]:		
		fieldid [integer]: custom field ID	
		value [string] : custom field value	
		mandatory [boolean]: TRUE|FALSE. This custom field should be mandatory?	
		labels [hash]:	
			<lang key>: (See Table 4. Available languages for display of the portal.)
			defaultLang [<lang key>]: (See Table 4. Available languages for display of the portal.)

Table 5 : authentication error codes

Return code	Description
permission-denied	Portal standard mode is missing? configuration error
too-many-attempts	User device is banned because number of failed attempts is reached
	value [integer] : user device is banned during (in minutes)
no-shib-login	Portal Shibboleth mode : user login is missing
restricted-acces-control	Portal Shibboleth mode : user profile is missing
bad-secure-password	Bad portal secure password
no-login	user login is missing
no-password	user password is missing
bad-policy-accept	Portal policy is not accepted
pms-user-account-not-allowed	Portal PMS mode : PMS user on invalidate portal configuration (no PMS portal)

no-pms-package	Portal PMS mode : user package is empty
paypal-user-account-not-allowed	Portal PAYPAL mode : PAYPAL user on invalidate portal configuration (no PAYPAL portal)
ogone-user-account-not-allowed	Portal OGONE mode : OGONE user on invalidate portal configuration (no OGONE portal)
no-paypal-package	Portal PAYPAL mode : user package is empty
cannot-connect-sql-database	Internal error
cannot-search-in-sql-database	Internal error
unknown-paypal-package	Portal PAYPAL mode : unknown user package
unknown-paypal-group	Portal PAYPAL mode : unknown user group
no-ogone-account	Portal OGONE mode : unknown user account
no-ogone-package	Portal OGONE mode : invalid user package
unknown-ogone-package	Portal OGONE mode : unknown user package
unknown-ogone-group	Portal OGONE mode : unknown user group
blocked-identification-code	Portal PPS mode: wrong user credentials (from PPS server)
expired-identification-code	Portal PPS mode: expired user credentials (from PPS server)
same-session-id	Portal PPS mode: same user session (from PPS server)
over-session-id	Portal PPS mode: number of authorized user sessions is reached (from PPS server)
cannot-connect-to-pps-server	Portal PPS mode: cannot connect to pps server
invalid-change-package	Portal PMS mode : failed user PMS package change
internal	package without validity | unhandled PPS return code

3.3 User disconnection

Disconnect a user.

- Parameters (mandatory)
  - action = **disconnect**
  - login = **user identifier**
- Optional
  - None

- Result : API disconnect response is either a returned error code Table 6 : disconnect error codes, or the following array :

  Array Description

- **step [string]:** current authentication step
- **type [string]:** current authentication type
- **info [hash]**:
  - code [string]: disconnect errors code, See Table 6 : disconnect error codes
- **login [hash]**:
  - value [string]: user identifier

Example

```
array (
   'step' => 'LOGON',
   'type' => 'ONE',
   'info' =>
   array (
      'code' => 'disconnect_success',
   ),
   'user' =>
   array (
      'login' =>
      array (
         'value' => 'ba',
      ),
   ),
```

Table 6 : disconnect error codes

Return code	Description
disconnect_success	
no-login	user login is missing
no-password	user password is missing (internal error)
failed to disconnect device	internal error

3.4 User refresh

Refresh a user connection each 50 seconds.

- Parameters (mandatory)
  - action = **refresh**
  - login = **user identifier**
- Optional
  - None

- Result : API refresh action returns the following two parameters :

  Array Description

  If success refresh:

  - **step [string]:** FEEDBACK
  - **type [string]:** REFRESH

  If success failed:

  - **step [string]:** LOGON
  - **type [string]:** “”

Examples:

SUCCES REFRESH:

```
array (
   'step' => 'FEEDBACK',
   'type' => ' REFRESH',
   ),
```

FAILED REFRESH: (internal error)

```
array (
   'step' => ' LOGON ',
   'type' => ' ',
   ),
```

3.5 User registering

Register a user (self-registering). Two registration modes are available: 'One-Click Button' and 'open direct registration'.

- Parameters (mandatory)
  - action = **subscribe**
  - type = **one** ('One-Click Button') or **direct** (open direct registration')
- Optional
  - **connect_policy_accept = TRUE | FALSE**
  - **policy_accept = TRUE | FALSE**
  - HTTP_POST parameters: array containing keys and associated values. See the 'init' action feedback (See 5.1 Initialization (get information)) for mandatory and optional parameters:

Example:

Table 7 : One click registration example

Keys	associated values
action	subscribe
type	one
user_login	jsmith
last_name	Smith
first_name	John
policy_accept	TRUE

- Result :

Array Description:

- info [hash]:
  - **code [string]**: If success API registering action returns :
    - **"info_one-subscribe_success"** for Open registration One-Click
    - "**info_direct-subscribe_success"** for Open direct registration
    - ! value [string]: self-registering error value, depending on the error code

    Else API registering action returns self-registering error codes See Table 8 : 'Self-registering' error codes

  - **subscribe [hash]**: new user account parameters
    - **login [string]** : identifier of the new user account
    - **password [string]** : password of the new user account

Example 1 with One-Click subscribing success

```
array (
   ‘info’=>
        array (
           'code' => ' info_one-subscribe_success',
           'subscribe' =>
               array (
                      'login' => 'ymrbm8c',
                      'password' => '37nr2BdU',
                 ),
                .....

        )
```

Example 2 with failed One-Click subscribing

```
array (
   ‘info’=>
        array (
           'code' => ' error_one-subscribe_already-used-login',
           'subscribe' =>
               array (
                      'login' => NULL,
                 ),

        )
```

Table 8 : 'Self-registering' error codes

<table>
<tr><th colspan="2">All retuned error code are shared by the two open registration modes and prefixed by the mode name, then:
- For One-click registration <b>prefix</b> = “<b>one-subscribe</b>”
- For Open direct registration <b>prefix = “direct-subscribe”</b></th></tr>
<tr><th>Return code</th><th>Description</th></tr>
<tr><td><b>prefix_no-last-name</b></td><td>Parameter "last-name" which is mandatory is missing</td></tr>
<tr><td><b>prefix_no-first-name</b></td><td>Parameter "first-name" missing which is mandatory is missing</td></tr>
<tr><td><b>prefix_no-email_address</b></td><td>Parameter "email_address" which is mandatory is missing</td></tr>
<tr><td><b>prefix_no-phone-prefix</b></td><td>Parameter "phone-prefix" which is mandatory is missing</td></tr>
<tr><td><b>prefix_no-phone-number</b></td><td>Parameter "phone-number" which is mandatory is missing</td></tr>
<tr><td><b>prefix_invalid-last-name</b></td><td>Parameter "last-name" is not valid (See Table 3: Allowed chars)</td></tr>
<tr><td><b>prefix_invalid-first-name</b></td><td>Parameter "first-name" is not valid (See Table 3: Allowed chars)</td></tr>
</table>

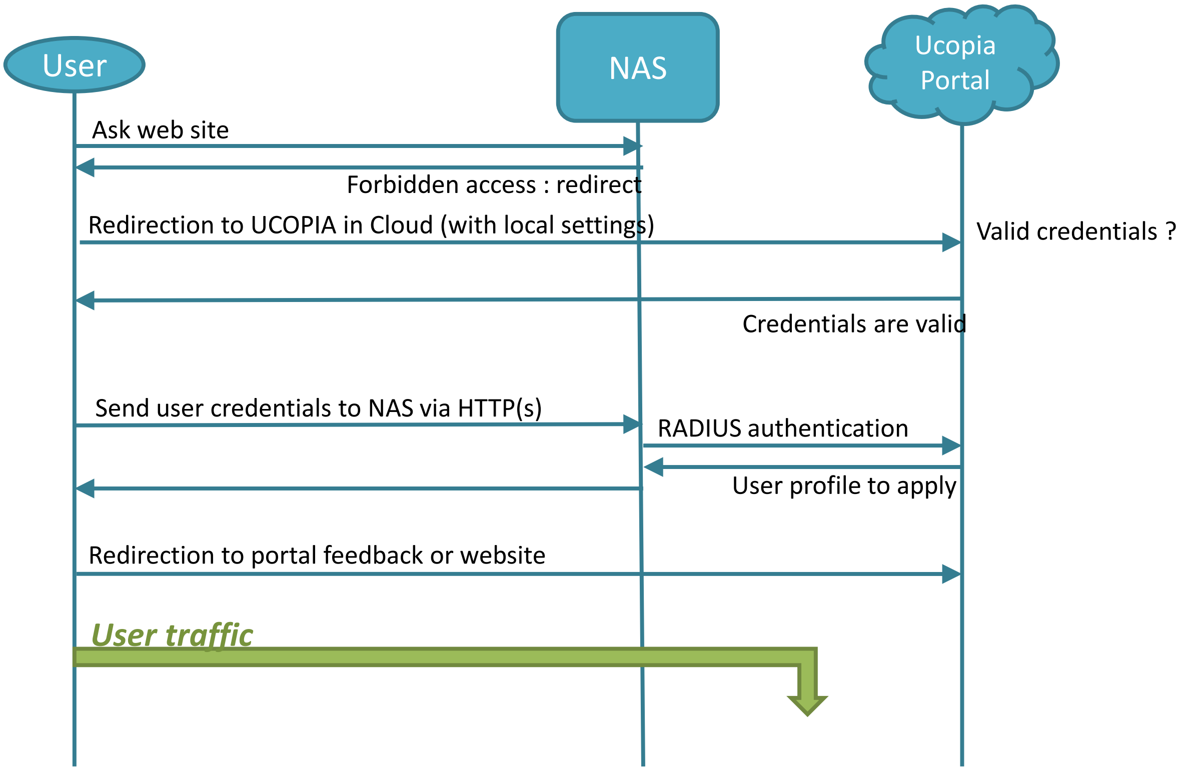
prefix_invalid-email-address	Parameter "email-address" is not valid (See Table 3: Allowed chars)
prefix_invalid-phone-prefix	Parameter "phone-prefix" is not valid (See Table 3: Allowed chars)
prefix_invalid-phone-number	Parameter "phone-number" is not valid (See Table 3: Allowed chars)
prefix_no-organizational-unit-name	Parameter " organizational-unit-name " which is mandatory is missing
prefix_invalid-organizational-unit-name	Parameter "organizational-unit-name" is not valid (See Table 3: Allowed chars)
prefix_no-postal-address	Parameter "postal-address" which is mandatory is missing
prefix_no-postal-code	Parameter "postal-code" which is mandatory is missing
prefix_no-postal-locality-name	Parameter "postal-locality-name" which is mandatory is missing
prefix_no-postal-country-name	Parameter "postal-country-name" which is mandatory is missing
prefix_no-custom-field	Parameter "custom-field " which is mandatory is missing, check the error value for the corresponding custom-field
prefix_no-policy-accept	Parameter "policy-accept " (private policy) which is mandatory is missing
prefix_no-connect-policy-accept	Parameter "connect-policy-accept" (global policy) which is mandatory is missing
prefix_already-used-login	Parameter "login" already exists
prefix_different-passwords	Parameters "user_password" and "user_password_confirm" are different
prefix_invalid-password	Parameter "password" is not valid
prefix_no-profile	Portal setting error, no profile defined
prefix_user-creation-failed	Internal error

4 UCOPIA In-line architecture

This is the standard architecture where the UCOPIA controller is in-line (i.e. The end-user traffic goes through the controller). UCOPIA portal should direct interact with the API, as explained in the Section: Global behavior (connection flow) and illustrated on the Figure 1 : Connection Flow.

5 UCOPIA Out-of-band architecture

The sequence diagram below describes the flow when UCOPIA is not in-line (for example in the Cloud) and the portal redirection is performed with a Wi-Fi equipment.


User
NAS
Ucopia Portal
Ask web site
Forbidden access : redirect
Redirection to UCOPIA in Cloud (with local settings)
Valid credentials ?
Credentials are valid
Send user credentials to NAS via HTTP(s)
RADIUS authentication
User profile to apply
Redirection to portal feedback or website
User traffic

And can be summarized into three steps:

Step 1: Initiates an HTTP or HTTPS request,

Step 2: Following the first redirection also by a HTTP or HTTPS request,

Step 3: Then, build and send requests to interact with the API (“init” | “authenticate” | “subscribe” | “refresh”| “disconnect”)

Note: The following is based on the “Out-of-band” architecture with Xirrus solution on premise.

STEP 1:

Request : initiates an HTTP or HTTPS request using GET or POST parameters (www.google.com for example)

Response:

```
HTTP request sent, awaiting response… 302 Moved Temporarily
Location: REDIRECT_URL
```

REDIRECT_URL: which is sent by the NAS describes how to build requests to interact with the API (“init” | “authenticate” | “subscribe” |”refresh” | “disconnect”).

REDIRECT_URL is composed by several GET parameters attributes as following:

```
PROTOCOL://CENTRAL_CONTROLLER_ADRESS/zone/ZONE_NAME/?GET_ARGUMENT_
LIST
```

PROTOCOL: protocol (HTTP or HTTPS) to use when redirecting the user to the NAS

CENTRAL_CONTROLLER_ADRESS: central UCOPIA controller IP address or domain name

ZONE_NAME: user incoming zone name

GET_ARGUMENT_LIST (See Table 9: Xirrus GET ARGUMENT LIST):
res=$response&uamip=$uamip&uamport=$uamport&challenge=$challenge&
userurl=$userurl&mac=$mac&SSID=$SSID&vlan=$vlan

Table 9: Xirrus GET ARGUMENT LIST

Attribute	Description
res	User status (i.e., : success, failed, notyet, logoff, etc.), depending on the NAS constructor
uamip	Identify the current NAS constructor by the default IP address
uamport	Identify the current NAS constructor by the default PORT number
challenge	user challenge response (used during CHAP authentication)
userurl	URL requested by the user
mac	User's mac address
SSID	User's associated SSID
vlan	User's vlan

STEP 2:

Request : Following the REDIRECT_URL by a HTTP or HTTPS request using GET or POST parameters

Response :

```
HTTP/1.1 302 Found
Location: REDIRECT_PORTAL
```

REDIRECT_PORTAL: which is sent by the central UCOPIA controller specifies the portal settings to be used in the case of a browser client, and means that the session is now open and the API is ready to serve requests.

> **Warning**: from step 2 until the session end UCOPIA application must keep the first session cookie sent by the central UCOPIA controller and reuse it for each request throughout the session.

STEP 3:

Once the session is open and the API is ready to serve requests, UCOPIA application should send requests to interact with the API with the following format:

```
PROTOCOL://CENTRAL_CONTROLLER_ADRESS/portal_api.php?action=”API_PRIMITIVES”
```

Example:

STEP 1:

Request :

```
wget http://www.google.com
```

Response :

```
--2015-03-05 11:02:01--  http://www.google.com/
Resolving www.google.com (www.google.com)... 216.58.211.68,
2a00:1450:4007:806::1014
Connecting to www.google.com (www.google.com)|216.58.211.68|:80...
connected.
HTTP request sent, awaiting response...

302 Moved Temporarily

Location:
https://controller.ucopia.lan/zone/test/?res=notyet&uamip=185.0.0.1&
uamport=10000&challenge=491547e80adf07498546d902541bdb73&userurl=htt
p%3a%2f%2fwww.google.com%2f&mac=00-22-fb-d5-15-
ee&ssid=test_badis_doc_api&vlan=0
```

Then, UCOPIA application should followed this redirection to interact with the API:

STEP 2:

Request:

```
wget
https://controller.ucopia.lan/zone/test/?res=notyet&uamip=185.0.0.1&
uamport=10000&challenge=491547e80adf07498546d902541bdb73&userurl=htt
p%3a%2f%2fwww.google.com%2f&mac=00-22-fb-d5-15-
ee&ssid=test_badis_doc_api&vlan=0
```

Response:

```
Resolving controller.ucopia.lan (controller.ucopia.lan)... 10.1.10.2
Connecting to controller.ucopia.lan
(controller.ucopia.lan)|10.1.10.2|:443... connected.
HTTP request sent, awaiting response...
  HTTP/1.1 302 Found
  Location: /100/portal/
  Content-Length: 478
  Keep-Alive: timeout=2, max=100
  Connection: Keep-Alive
  Content-Type: text/html; charset=UTF-8
```

This redirection specifies the portal settings to be used (Location: /100/portal) and means that the session is now open and the API is ready to serve requests.

STEP 3:

Request:

```
Wget
https://controller.ucopia.lan/portal_api.php?action="init"
```

Response (JSON format):

{"logon":{"policy":{"display":false}},"subscribe":{"count":1,"one":{"userLogin":{"display":true,"mandatory":true},"lastName":{"display":true,"mandatory":true},"email":{"display":true,"mandatory":true},"policy":{"display":false}}},"user":{"isAuthenticatedThroughShibboleth":false,"lang":"fr","incomingNetwork":{"value":"native-out"},"incomingNetworkID":{"value":"out"},"incomingZone":{"value":"test"}},"auth_modes":{"standard":true,"one":true},"feedback":{"pms":{"customer":{"display":true},"message":{"display":true}},"caution":{"display":true},"requestedURL":{"display":true},"login":{"display":true},"profile":{"display":true},"services":{"display":true},"ipAddress":{"display":false},"incomingNetwork":{"display":false},"incomingZone":{"display":true},"incommingVlan":{"display":false},"incommingZone":{"display":true},"multidevice":{"display":true},"validity":{"display":true},"forceDiscTimer":{"display":true},"totalTimeCredit":{"display":true},"totalConsumedTimeCredit":{"display":true},"timeCredit":{"display":true},"consumedData":{"display":true}},"pwdRecovery":false,"lang":{"displayLang":["fr","de","en","es","it","nl","pt","pl","zh_CN","zh_TW","ar","ko","ja","th","ru","id"],"defaultLang":"fr","customFieldsDefaultLang":null},"refreshInterval":50000,"step":"SUBSCRIBE","type":"ONE","modifyPwd":true,"forceModifyPwd":true}

5.1 UCOPIA “OUT Of BAND”: authentication/disconnection

More steps are needed to do the authentication/disconnection requests in order to effective authenticate/disconnect from the NAS.

Request:

```
PROTOCOL://CENTRAL_CONTROLLER_ADRESS/?action=authenticate|disconne
ct&login=$login&password=$password
```

Response:

The same authentication response plus following parameters to be used to authenticate with the NAS:

….

- **login_URL_params [hash]**: gather all URL particular options to use for user connection
  - **type [string]** : type of the request: GET or POST
  - **uri [string]**: login or logout
  - **host [string]**: NAS IP address or domain
  - **port [string]**: **NAS port**
  - **protocol [string]**: protocol (http|https) to use when redirecting the user to the NAS at user (dis)connection
  - **parameters [hash]**:
    - **username [string]**: user login
    - **response [string]**: user challenge response (used during CHAP authentication)

……

Then, UCOPIA application should build the authentication|disconnection request using those **login_URL_params** and sent them to the NAS for effective (dis)connection:

Request:

```
Protocol://host:port/uri&username=$username&response=$response
```

Response:

```
HTTP request sent, awaiting response… 302 Moved Temporarily
Location: REDIRECT_URL
```

Then, following the **REDIRECT_URL** to have the last authentication/disconnection feedback from the central UCOPIA controller.

Note that this **REDIRECT_URL** includes the ‘**res**’ attribute which specifies the user status against RADIUS authentication (“success”, “already”, “failed”, “logoff”, etc.).

Example:

Request:

```
wget
https://controller.ucopia.lan/portal_api.php?action=authenticat
e&login=jsmith&password=1234
```

Response:

```
….
login_URL_params: array (
   'type' => 'GET',
   'uri' => 'login',
   'host' => '185.0.0.1',
   'port' => '10000',
   'protocol' => 'http',
   'parameters' =>
   array (
      'username' => 'jsmith',
      'response' => '2a91a1f9af386d2ebf8d42b755a53de1',
   ),
),
…..
```

Request:

```
wget
http://185.0.0.1:10000/login?username=jsmith&response=2a91a1f9a
f386d2ebf8d42b755a53de1
```

Response:

```
Connecting to 185.0.0.1:10000... connected.
HTTP request sent, awaiting response...
HTTP/1.0 302 Moved Temporarily
Location: https://controller.ucopia.lan/zone/test/?res=
success&uamip=185.0.0.1&uamport=10000&userurl=http%3a%2f%2fwww.gstatic.com%2fch
rome%2fprofile_avatars%2fNothingToDownload&mac=00-22-fb-d5-15-
ee&ssid=test_badis_doc_api&vlan=0
```

Request:

```
Wget

https://controller.ucopia.lan/zone/test/?res=
success&uamip=185.0.0.1&uamport=10000&userurl=http%3a%2f%2fwww.gstatic.com%2fch
rome%2fprofile_avatars%2fNothingToDownload&mac=00-22-fb-d5-15-
ee&ssid=test_badis_doc_api&vlan=0
```

Response (JSON format):

{"logon":{"policy":{"display":false}},"subscribe":{"count":1,"one":{"userLogin":{"display":true,"mandatory":true},"lastName":{"display":true,"mandatory":true},"email":{"display":true,"mandatory":true},"policy":{"display":false}}},"user":{"isAuthenticatedThroughShibboleth":false,"lang":"fr","incomingNetwork":{"value":"native-out"},"incomingNetworkID":{"value":"out"},"incomingZone":{"value":"test"}},"auth_modes":{"standard":true,"one":true},"feedback":{"pms":{"customer":{"display":true},"message":{"display":true}},"caution":{"display":true},"requestedURL":{"display":true},"login":{"display":true},"profile":{"display":true},"services":{"display":true},"ipAddress":{"display":false},"incomingNetwork":{"display":false},"incomingZone":{"display":true},"incommingVlan":{"display":false},"incommingZone":{"display":true},"multidevice":{"display":true},"validity":{"display":true},"forceDiscTimer":{"display":true},"totalTimeCredit":{"display":true},"totalConsumedTimeCredit":{"display":true},"timeCredit":{"display":true},"consumedData":{"display":true}},"pwdRecovery":false,"lang":{"displayLang":["fr","de","en","es","it","nl","pt","pl","zh_CN","zh_TW","ar","ko","ja","th","ru","id"],"defaultLang":"fr","customFieldsDefaultLang":null},"refreshInterval":50000,"step":"SUBSCRIBE","type":"ONE","modifyPwd":true,"forceModifyPwd":true}

6 Get user settings (portal feedback)

If UCOPIA application needs to recover settings once user is connected, it should request the following URL:

```
https://<@UCOPIA controller host>
```

7 Known limitations

The following section describes known limitations of the current UCOPIA controller's version (5.0):

7.1 Allowed user agent strings

Currently the accepted user agent strings are summarized in the Section 0: HTTP user agent restrictions.

7.2 Allowed special characters

Currently we restrict allowed special characters for string values, as explained in the Table 3: Allowed chars, Section 33: API description.

8 Appendix 1: API “INIT” example

URL = “https://@controller_host/portal_api.php”;

action = “init”

API Response: (PHP Array)

```
array (
   'logon' =>
   array (
      'policy' =>
      array (
         'display' => true,
         'displayLang' =>
         array (
            0 => ' en',
            1 => ' fr ',
            …
         ),
         'en' =>
         array (
            'text' => ' perso_data_policy'',
            'link' => NULL,
         ),
         'defaultLang' => 'en',
   ),
   'subscribe' =>
   array (
      'count' => 2,
      'one' =>
      array (
         'userLogin' =>
         array (
            'display' => true,
            'mandatory' => false,
         ),
         'userPassword' =>
         array (
            'display' => true,
            'mandatory' => false,
         ),
         'lastName' =>
         array (
            'display' => true,
            'mandatory' => false,
         ),
         'firstName' =>
         array (
            'display' => true,
            'mandatory' => false,
         ),
         'email' =>
         array (
            'display' => true,
            'mandatory' => false,
         ),
         'phone' =>
         array (
            'display' => true,
            'mandatory' => false,
         ),
         'organizationalUnitName' =>
         array (
            'display' => true,
            'mandatory' => false,
         ),
         'postalAddress' =>
         array (
            'display' => true,
            'mandatory' => false,
         ),
         'personalField_1' =>
```

```
array (
  'display' => true,
  'mandatory' => true,
  'labels' =>
  array (
    'fr' => ' carte de fidelité',
    'en' => ' Fidelity card',
  ),
 …
),
'policy' =>
array (
  'display' => true,
  'displayLang' =>
  array (
    0 => 'ar',
    1 => 'nl',
    2 => 'en',
    3 => 'fr',
    4 => 'de',
    5 => 'id',
    6 => 'it',
    7 => 'ja',
    8 => 'ko',
    9 => 'pl',
    10 => 'pt',
    11 => 'ru',
    12 => 'zh_CN',
    13 => 'es',
    14 => 'th',
    15 => 'zh_TW',
  ),
  'ar' =>
  array (
    'text' => 'French',
    'link' => 'http://www.ucopia.com',
      ),
      'nl' =>
      array (
        'text' => 'French',
        'link' => ' http://www.ucopia.com',
      ),
      'en' =>
      array (
        'text' => 'French',
        'link' => 'http://www.ucopia.com',
      ),
      'fr' =>
      array (
        'text' => 'French',
        'link' => 'http://www.ucopia.com',
      ),
       ……..
      ),
    ),
  ),
  'direct' =>
  array (
    'userLogin' =>
    array (
      'display' => true,
      'mandatory' => false,
    ),
     ……..
      ),
    ),
  ),
),
'user' =>
array (
  'isAuthenticatedThroughShibboleth' => false,
  'lang' => 'fr',
  'incomingNetwork' =>
  array (
    'value' => 'native-in',
  ),
  'incomingNetworkID' =>
  array (
    'value' => 'in',
  ),
  'incomingZone' =>
  array (
    'value' => 'Default-in',
  ),
),
'auth_modes' =>
array (
  'standard' => true,
  'one' => true,
),
'feedback' =>
array (
  'pms' =>
  array (
    'customer' =>
    array (
      'display' => true,
    ),
    'message' =>
    array (
      'display' => true,
    ),
  ),
  'caution' =>
  array (
    'display' => true,
  ),
  'requestedURL' =>
```

```
array (
   'display' => true,
),
'login' =>
array (
   'display' => true,
),
'profile' =>
array (
   'display' => true,
),
'services' =>
array (
   'display' => true,
),
'ipAddress' =>
array (
   'display' => false,
),
'incomingNetwork' =>
array (
   'display' => false,
),
'incomingZone' =>
array (
   'display' => true,
),
'incommingVlan' =>
array (
   'display' => false,
),
'incommingZone' =>
array (
   'display' => true,
),
'multidevice' =>
```

```
   array (
      'display' => true,
   ),
   'validity' =>
   array (
      'display' => true,
   ),
   'forceDiscTimer' =>
   array (
      'display' => true,
   ),
   'totalTimeCredit' =>
   array (
      'display' => true,
   ),
   'totalConsumedTimeCredit' =>
   array (
      'display' => true,
   ),
   'timeCredit' =>
   array (
      'display' => true,
   ),
   'consumedData' =>
   array (
      'display' => true,
   ),
),
'pwdRecovery' => true,
'lang' =>
array (
   'displayLang' =>
   array (
      0 => 'fr',
      1 => 'de',
      2 => 'en',
         3 => 'es',
         4 => 'it',
         5 => 'nl',
         6 => 'pt',
         7 => 'pl',
         8 => 'zh_CN',
         9 => 'zh_TW',
         10 => 'ar',
         11 => 'ko',
         12 => 'ja',
         13 => 'th',
         14 => 'ru',
         15 => 'id',
      ),
      'defaultLang' => 'fr',
      'customFieldsDefaultLang' => 'ar',
   ),
   'refreshInterval' => 50000,
   'postConnectRedirect' =>
   array (
      'url' => ' http://www.wi-fi.org/ ',
      'delay' => '20',
   ),
   'step' => 'SUBSCRIBE',
   'type' => 'ONE',
   'modifyPwd' => true,
   'forceModifyPwd' => true,
)
```

9 Appendix 2: API “AUTHENTIFICATION” example

URL = “https://@controller host/portal_api.php”;

Parameters :

action = “**authenticate**”

login **= “jsmith”**

password **= “xxxx”**

API Response: (PHP Array)

```
array (
  'authenticate_step' => 'LOGON',
  'authenticate_type' => 'ONE',
  'step' => 'FEEDBACK',
  'type' => 'CONNECT',
  'user' =>
  array (
    'login' =>
    array (
      'value' => 'ba',
    ),
    'ipAddress' =>
    array (
      'value' => '192.168.100.1',
    ),
    'profile' =>
    array (
      'value' => 'profile_1',
    ),
    'services' =>
    array (
      'value' => 'Full_Access, Mail, Remote_Access, SSH, Web_Proxy',
    ),
    'autoDisconnect' =>
    array (
      'value' => false,
    ),
    'schedule' =>
    array (
      'value' =>
      array (
        0 =>
        array (
          'begin' =>
          array (
            'day' => 'monday',
            'hour' => '00',
            'min' => '00',
          ),
          'end' =>
          array (
            'day' => 'sunday',
            'hour' => '24',
            'min' => '00',
          ),
        ),
      ),
    ),
    'validity' =>
    array (
      'value' => '0',
    ),
    'initTimeGMT' =>
    array (
      'value' => '1425305092',
    ),
    'timeCredit' =>
    array (
      'value' => '7200',
      'remaining' =>
      array (
        'value' => 5146,
       ),
       'reneweach' =>
       array (
          'value' => '0',
       ),
       'initialRemaining' =>
       array (
          'value' => 5146,
       ),
    ),
    'macAddress' =>
    array (
       'value' => '00:20:xx:xx:xx:xx',
    ),
    'incomingNetwork' =>
    array (
       'value' => 'native-in',
    ),
    'incomingNetworkID' =>
    array (
       'value' => 'in',
    ),
    'incomingZone' =>
    array (
       'value' => 'Default-in',
    ),
    'incomingVlan' =>
    array (
       'value' => 'native-in',
    ),
    'incommingVlan' =>
    array (
       'value' => 'native-in',
    ),
    'incommingZone' =>
    array (
       'value' => 'Default-in',
    ),
    'creationMode' =>
    array (
       'value' => 'portal-one',
    ),
    'universalTime' =>
    array (
       'value' => 1425305092,
    ),
    'timezoneOffset' =>
    array (
       'value' => '3600',
    ),
    'requestedURL' =>
    array (
       'value' => 'http://www.cambridgeenglish.org/ ',
    ),
    'allowModPwdBySelf' => false,
    'managePersonalSettings' => false,
    'lastName' =>
    array (
       'value' => 'John',
    ),
    'firstName' =>
    array (
       'value' => 'Smith',
    ),
    'email' =>
    array (
       'value' => 'jsmith@gmail.com',
    ),
    'phoneNumber' =>
    array (
       'value' => '',
    ),
    'organizationalUnitName' =>
    array (
       'value' => 'ucopia',
    ),
    'countryName' =>
    array (
       'value' => 'France',
    ),
    'localityName' =>
    array (
       'value' => 'Montrouge',
    ),
    'postalCode' =>
    array (
       'value' => '75',
    ),
    'postalAddress' =>
    array (
       'value' => '20',
    ),
    'postOfficeBox' =>
    array (
       'value' => '92100',
    ),
    'customFields' =>
    array (
       0 =>
       array (
         'fieldid' => '1',
         'value' => ' Fidelity card’,
         'mandatory' => true,
         'label' =>
         array (
           'en' => ' Fidelity card ',
            …
           'defaultLang' => 'en',
                    ),
                  …
                ),
            'manageAccount' => false,
        ),
    …….
            ),
        ),
```