



# Cloud User Management

Administration & Integration Guide

Deutsche Telekom Laboratories

Version: 1.3

Issued: 19.09.2011

State: Final



# Imprint

## **Publisher**

---

Deutsche Telekom Laboratories  
Ernst-Reuter Platz 7  
10587 Berlin

---

19.09.2011

---

# Revision History

---

<b>Version</b>	<b>Date</b>	<b>Editor</b>	<b>Revisions / Comment</b>
1.3	19.09.2011		Initial Version

---

# Content

<b>1</b>	<b>Cloud User Management Administration</b>	<b>3</b>
<b>2</b>	<b>Cloud User Management Configuration</b>	<b>4</b>
2.1	Attribute Settings .....	4
2.2	Password Policies.....	5
2.3	Application Configuration .....	5
2.4	3 <sup>rd</sup> Party Login .....	5
2.5	Other Security Options .....	7
<b>3</b>	<b>User Management</b>	<b>8</b>
<b>4</b>	<b>Cloud User Management Login Widget</b>	<b>9</b>
<b>5</b>	<b>Login Widget Integration</b>	<b>10</b>
5.1	Front-end Integration.....	10
5.1.1	Login Widget Integration.....	10
5.1.2	Login Widget Callbacks .....	10
5.1.3	Login Widget Functions .....	11
5.1.4	Login Widget Status .....	11
5.1.5	Login Widget Customization .....	12
5.2	Backend Integration.....	13
5.2.1	Access to User Attributes .....	14

# 1 Cloud User Management Administration

The Cloud User Management (CUM) is a consumer identity management solution offered as a Software-as-a-Service (SaaS). You can easily integrate it into your web applications and you get authentication (including 3<sup>rd</sup> parties), user self services and user profile attribute sharing.

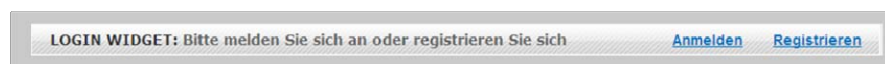
## Cloud User Management Portal

The Cloud User Management portal lets you manage your personal Cloud User Management (CUM) instance. You can access the portal from the Developer Center where you log in with your Developer Center user account.

Main features of the portal are:

- Configuration of your CUM, and
- Management of your CUM users.

Besides the SaaS aspects, the Cloud User Management offers a Login Widget designed for easy integration into most web apps.



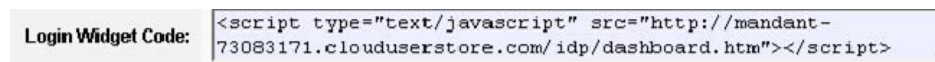
In the following sections we show you the configuration options and user management functions offered by the portal as well as a guideline for the integration of the CUM Login Widget into your web apps.

## 2 Cloud User Management Configuration

The Cloud User Management portal offers all configuration option on a single page. There you get all the information you need to further setup you instance.



One of the first things you need is your tenant-id that is required for the integration of the Login Widget into your web apps (for detailed description take a look at the Cloud User Management Integration Guideline). To make things easy, just copy and paste the little code snippet from the CUM portal:



### 2.1 Attribute Settings

User accounts in the CUM always have a unique **userid**, and they must have set a **password** and an **email** address. These attributes are the core of any account and can not be omitted. But for all other attributes offered by the CUM you can configure whether you want to enable them in your CUM instance and whether one or more of them shall be required, that is your users cannot leave these fields empty.

Nutzer Attribute :	
Anzeigenname:	<input type="checkbox"/> Erforderlich <input checked="" type="checkbox"/> Sichtbar
Nachname:	<input type="checkbox"/> Erforderlich <input checked="" type="checkbox"/> Sichtbar
Vorname:	<input type="checkbox"/> Erforderlich <input checked="" type="checkbox"/> Sichtbar
Geschlecht:	<input type="checkbox"/> Erforderlich <input checked="" type="checkbox"/> Sichtbar
Straße:	<input type="checkbox"/> Erforderlich <input checked="" type="checkbox"/> Sichtbar
Postleitzahl:	<input type="checkbox"/> Erforderlich <input checked="" type="checkbox"/> Sichtbar
Land:	<input type="checkbox"/> Erforderlich <input checked="" type="checkbox"/> Sichtbar
Bundesland:	<input type="checkbox"/> Erforderlich <input type="checkbox"/> Sichtbar
Stadt:	<input type="checkbox"/> Erforderlich <input checked="" type="checkbox"/> Sichtbar
Geburtsdatum:	<input type="checkbox"/> Erforderlich <input checked="" type="checkbox"/> Sichtbar
Telefon Privat:	<input type="checkbox"/> Erforderlich <input checked="" type="checkbox"/> Sichtbar
Mobil:	<input type="checkbox"/> Erforderlich <input checked="" type="checkbox"/> Sichtbar
Sicherheitsfrage:	<input checked="" type="checkbox"/> Erforderlich <input checked="" type="checkbox"/> Sichtbar
Antwort:	<input checked="" type="checkbox"/> Erforderlich <input checked="" type="checkbox"/> Sichtbar

You can change your configuration whenever you want, it will be applied immediately to the running service.

## 2.2 Password Policies

Enforcement of strong passwords is always a good measure for more security – but there is no “one fits all” policy for password complexity. The CUM therefore allows you to configure password policy appropriate for your application area.

<b>Nutzer Passwort Regeln:</b>	Mindestlänge:	3
	Maximale Länge:	100
	Mindestanzahl Großbuchstaben:	1
	Mindestanzahl Kleinbuchstaben:	1
	Mindestanzahl Ziffern:	1
	Mindestanzahl Sonderzeichen:	1

You can change your configuration whenever you want, it will be applied immediately to the running service. The policy will take effect at all new passwords and all changes.

## 2.3 Application Configuration

The Login Widget provided by the CUM allows easy integration of the CUM in you web pages, but a typical web application would need access to the user's profile data. For a detailed description on how to implement this take a look at the Cloud User Management Integration Guideline. Here in the portal you can manage the access credentials, that is you create application ids and secrets, and you can delete them as well. If you're working with different applications, you can even create separate credentials for any of them. Just create and them, then copy & paste them to your application.

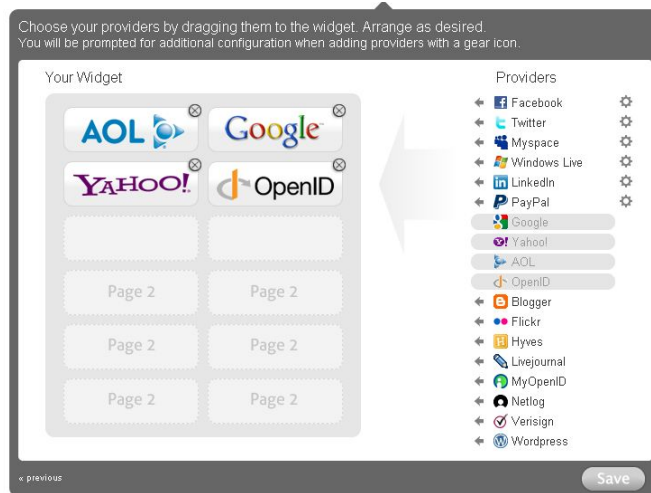
<b>Apps:</b>	ID:	aNhe94VC00EtJUQX6H5fBBWixJc48gPfdBCxYxv
	Name:	my demo webapp
	Secret:	8K82KfS57avHRepVv72xDIBtyKlZ3Le6zACV/aet2
		Aktualisieren   Löschen
<b>Apps Action:</b>	Anwendung hinzufügen	

Application credentials can be used immediately, and when you delete credentials, this takes effect immediately, too.

## 2.4 3<sup>rd</sup> Party Login

The concept of a 3<sup>rd</sup> party or social login allows your users to sign in with one of their favorite social networks accounts. With our integrated 3<sup>rd</sup> party login provided by Janrain you easily enable social login to your web sites. Simply request the creation of your own Janrain domain in the CUM portal, and it will be processed immediately. The activation process at Janrain requires us to pass your email address to Janrain, because they need it to send you an activation email. Follow their instructions to finish the activation.

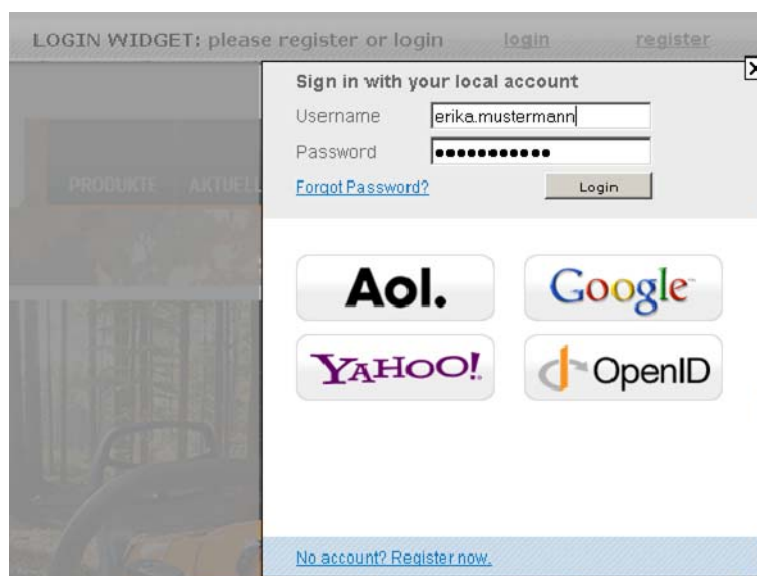
At the end of that process you'll be able to configure your 3<sup>rd</sup> party login in detail, that is you can select which social networks should appear in the Login Widget and which user attributes should be requested from them during login.



This kind of configuration is always made at the Janrain portal. For your convenience we include direct links to the corresponding pages at Janrain.

<b>Janrain Konfiguration:</b>	3rd Party Identity Provider List (@Janrain) 3rd Party Identity Provider Configuration
<b>Janrain App Domainname:</b>	my-demo.rpxnow.com
<b>Janrain App Schlüssel:</b>	0f92839ab-419d3f68f7f592abd4f6bf4e5cae39f
<b>Janrain Action:</b>	<a href="#">JanRain Link Löschen</a>

The Login Widget in you web pages will include 3<sup>rd</sup> login as configured without any further action.



## 2.5 Other Security Options

By integrating the Login Widget in you web pages your users do not only get the possibility to login and logout, they also can register themselves, modify their account data or request a new password when the current one has been forgotten.

To allow you to make registration and password reset more secure, the CUM provides an email confirmation mechanism. If enabled, users will have to follow a link sent to them via email to finish their registration or password reset process. In the meantime heir account will be blocked. To make it even more secure you can restrict the period of validity for this link.

	Sicherheitsfrage aktiviert:	ja
	Maximale Fehlversuche:	3
<b>Session Timeout (Minuten):</b>		30
<b>Email Bestätigung:</b>		nein
<b>Resetlink Timeout (Stunden):</b>		72

By activating the security question feature, password reset is possible only after the user has answered his personal security question. Good practice is to enable both the security question

And last but not least as with the password policies there is no “one-fits-all” best value for session timeouts. The default value should be acceptable in most cases, but you are encouraged to set a value suitable for your application area.

When you want to use your own CSS style sheets to adapt the appearance of the Login Widget to your web application, you have to register any URL you will pass as parameter to the Login Widget as a trusted URL first.

<b>Vertrauenswürdige CSS URLs:</b>	<pre>https://my.first.app.com:443/my-app/my_style.css http://my.second.app.com:80/another_app/another_style.css</pre>
------------------------------------	---

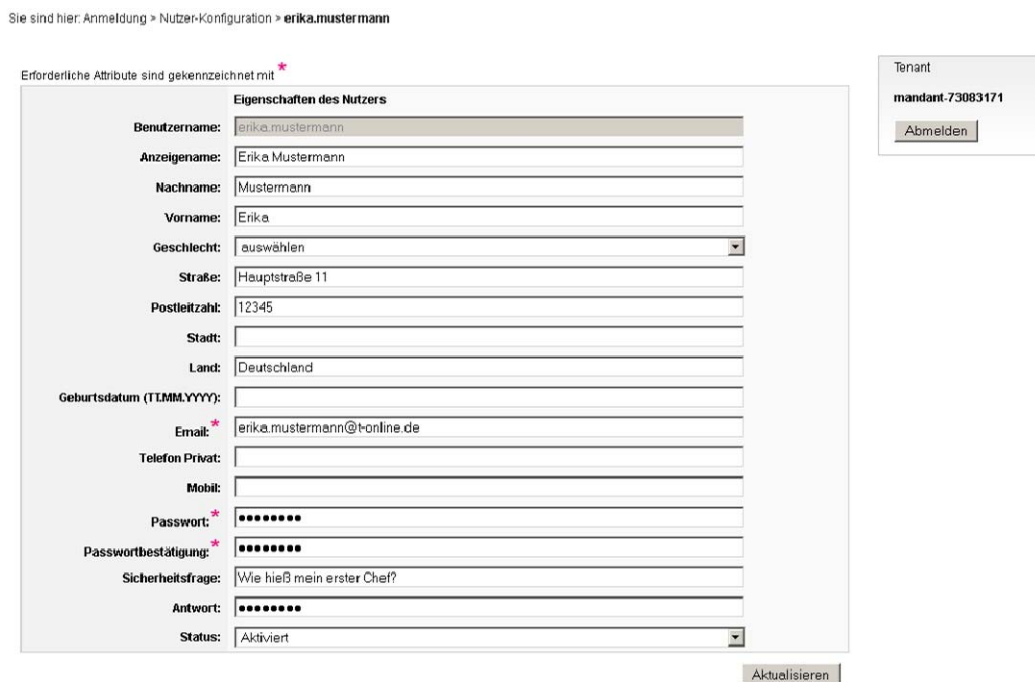
For security reasons the Login Widget will refuse to load any CSS URL that does not exactly match one of the registered trusted CSS URLs.

### 3 User Management

The user management capabilities of the CUM portal include the basic functions to create, list, search, modify and delete user accounts. The search filters include different name fields, the email address and the status.



The user management page is your tool to help your users when they have problems with their accounts.

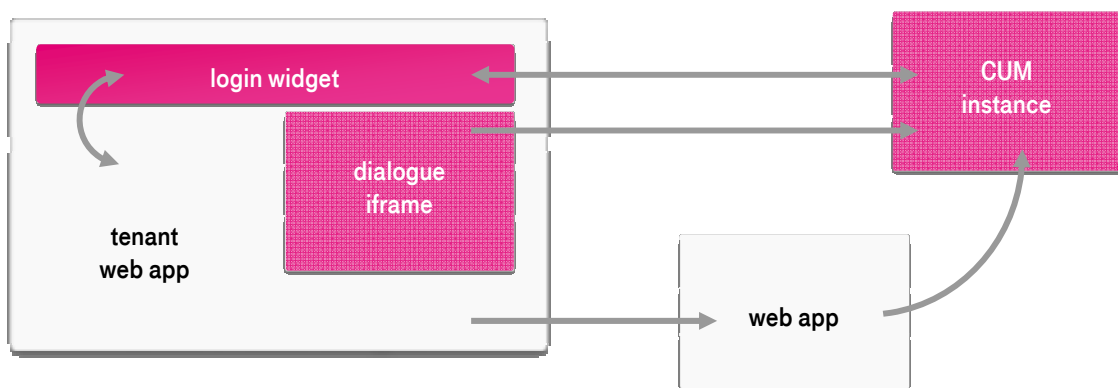


While you are the administrator of you CUM instance you can see and modify all user data with exception of the password: you can set it for a user, but you never can see any password.

## 4 Cloud User Management Login Widget

The Login Widget is the part of the service your users will see. It offers all the function they need from an identity management system:

- Authentication (login & logout).
- View and modify own account.
- Self services (registration, password reset).



The main features of the login widget are:

- It can be loaded from the Cloud User Management (CUM) by a single line of code.
- The widget places itself on top of the page.
- It communicates with page via Javascript.
- It accepts custom CSS per page.
- Its behavior is controlled by tenant's settings, e.g.
  - profile attributes (required/visible)
  - email verification option at registration
- It complies with cross-site-scripting security restrictions of modern browsers.

Trusted web applications can dynamically include the user profile resp. profiling information into existing web application business flows.

The following chapters explain how to integrate the login widget into a web app, how to customize it and how to use the HTTP-based interface for read and write access to the user profile.

For security and privacy reasons all communication with the Cloud User Management is encrypted (HTTPS).

## 5 Login Widget Integration

Integration of Cloud User Management in one or more Web applications is made by two steps:

1. Integrating the CUM login widget into the front-end part of the web app.
2. Preparing the back-end part of the web app for connecting to the Cloud User Management.

The widget integration is the visible part users will interact with, while the back-end part allows a web app to access the users profile including a couple of attributes that are not visible to the user but that are made available to web apps to store application specific profile information.

### 5.1 Front-end Integration

#### 5.1.1 Login Widget Integration

The login widget of a tenant can be integrated into a tenant web application page by inserting a single line of Javascript code:

```
<script src="https://TENANTID.clouduserstore.com/idp/dashboard.htm"></script>
```

TENANTID must be a valid tenantId, and can be obtained in the administration portal.

Variants of this procedure is to include parameters for

- providing a tenant web application-specific CSS for the login widget:

```
<script type="text/javascript"
  src="https://TENANTID.clouduserstore.com/idp/dashboard.htm? \
  css=http%3A%2F%2Fmywebapp.mydomain%2Fmystyle.css">
</script>
```

- or the option to keep the login widget invisible on the tenant web application:

```
<script type="text/javascript"
  src="https://TENANTID.clouduserstore.com/idp/dashboard.htm?dont_show_dashboard">
</script>
```

#### 5.1.2 Login Widget Callbacks

Importing the login widget into the pages of a web app should be accompanied by defining some JavaScript functions that will be used by the widget to communicate not only with the rest of the page in the browser but will also be used to communicate events like login or logout to the back-end part of the web app. This communication is necessary, because otherwise the back-end part of the web app can not access any user profile information hosted at the CUM.

The login widget has four standard JavaScript callback functions:

```
identityDashboard.callbacks.onLogin = function (sessionKey) { ... };
// send sessionKey to your webapp ...

identityDashboard.callbacks.onRegister = function () { ... };

identityDashboard.callbacks.onLogout = function () { ... };
// tell your webapp that user has logged out ...

identityDashboard.callbacks.onMyaccount = function () { ... };
```

The onLogin callback is the most important one, as it provides a sessionKey that is needed by the backend to access the user's profile. These functions can be implemented in several ways, e.g. AJAX calls working seamless in the background, or loading a new page on each event. The callback functions are optional, but it is strongly advised to implement at least the onLogin function.

### 5.1.3 Login Widget Functions

Besides the callback functions it is also possible to call widget functions from the web app. This feature is most important, when the widget is loaded in hidden mode as described above. In this scenario, the web app has to provide the appropriate page internal links or functions to the user.

The login widget dialogues can be triggered via JavaScript calls from a tenant web application using the following functions:

```
identityDashboard.showlogin(); // triggers the display of an overlay window for user login

identityDashboard.showregister(); // triggers the display of an overlay window for user
registration

identityDashboard.showlogout(); // triggers logout (an overlay window will be used but should
stay invisible)

identityDashboard.showmyaccount(); // triggers the display of an overlay window for editing user
profile information (resp. for visualization of adopted
attributes for non-native accounts)
```

### 5.1.4 Login Widget Status

Finally there are some status variables that can be read anytime to synchronize the web app with the widget.

For active user sessions with the Cloud User Management, the login widget offers three variables containing status information:

```
identityDashboard.loggedin; // boolean

identityDashboard.displayname; // String

identityDashboard.sessionKey; // String
```

## 5.1.5 Login Widget Customization

When importing the login widget without specifying additional options it will be displayed using a neutral default style that should be consistent with many web apps. Nevertheless, best results will be achieved by customizing the widget and the dialogues with custom CSS style sheets. The following style sheet shows the customizable elements classes and their relevant CSS style attributes.

Note: for security reasons you must register the URLs of your style sheets as trusted URLs in the Cloud User Management portal; CSS URLs that do not exactly match one of the registered trusted CSS URLs will not be loaded.

```

/* container element determining size and alignment of the dashboard */
/* default width with alignment centered */

.dashboard-container {
  width: 975px;
  margin-left: auto;
  margin-right: auto;
}

/* table that holds text, spacing and link elements of the dashboard */
/* default color white with gradient blend */

#dashboard-table {
  border-color:#FFFFFF;
  background-color:#FFFFFF;
  background-image:url(../images/dashback_01.png);
  height: 30px;
}

/* some text and spacing attributes */

.dashboard-text_links {
  font-family: Verdana; font-size: 14px; color: #333333; font-weight: bold;
}

.dashboard-text_mitte {
  font-family: Verdana; font-size: 14px; color: #666666; font-weight: bold;
}

.dashboard-abstand_links {
  width:18px;          /* left spacing */
}

.dashboard-abstand_rechts {
  width:18px;          /* right spacing*/
}

/* the links for login, logout, register, my account */
/* sequence link/visited/hover is important to make it work in all browsers */

a.dashboardLink:link {
  font-family: Arial, Helvetica, sans-serif;
  font-size: 14px;
  color: #0072bf;
}

a.dashboardLink:visited {
  font-family: Arial, Helvetica, sans-serif;
  font-size: 14px;
  color: #999999;
}

```

```
a.dashboardLink:hover {
  font-family: Arial, Helvetica, sans-serif;
  font-size: 14px;
  color: #FF0000;
}
```

The following style attributes are specific to the dialogue for login, registration etc.

```
/* limit the display size for attribute lists etc */

.dashboard-dialog-scrollarea {
  max-height:300px;
}

/* background styles for dialogue overlays */

.dashboard-dialog-farbe_1 {
  background-color:#ebebeb;
}

.dashboard-dialog-farbe_2 {
  background-color: #c5d8e7;
  background-image:url(../images/streifen_01.png);
}

/* headline text of dialogues */

.dashboard-dialog-headline {
  font-family: Arial, Helvetica, sans-serif;
  font-size: 14px;
  font-weight: bold;
  color: #666666;
}

/* text labels for input fields */

.dashboard-dialog-input_label {
  font-family: Arial, Helvetica, sans-serif;
  font-size: 14px;
  color: #666666;
}

/* all error messages */

.dashboard-dialog-error {
  font-family: Arial, Helvetica, sans-serif;
  font-size: 14px;
  color: #FF0000;
}
```

## 5.2 Backend Integration

This integration connects the Cloud User Management to web apps business logic. A prerequisite is the transfer of a sessionKey from the frontend into the backend (free choice of method). Together with an application id and an application secret, the web app then can access the user's profile (read and write). Application ids and secrets are created at the Cloud User Management administration portal, therefore only authorized web apps can access the user's profile. A tenant's admin can issue and revoke application ids at any time.

## 5.2.1 Access to User Attributes

User attributes can be read by a web app by posting a request to the Cloud User Management (added line breaks for better readability):

```
POST /idp/attributes/get.htm HTTP/1.1
Host: TENANTID.clouduserstore.com
Content-Type: application/x-www-form-urlencoded

format=json&
sessionKey=sessionKey&
application_id=APPLICATIONID&
application_secret=APPLICATIONSECRET
```

The user attributes are returned as JSON array of attribute names and values:

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store

{
  "ATTRIBUTE": "VALUE",
  "ATTRIBUTE": "VALUE",
  "ATTRIBUTE": "VALUE"
}
```

This is the preliminary list of attributes:

```
userid      (the userid of the WL-IdP account)
email       (the email address provided by the user)

displayname (display name as provided by the user)
givenname  (the first name / given name provided by the user)
surname    (the surname provided by the user)

street      (the street provided by the user)
province   (the state or province provided by the user)
zipcode    (the zip code provided by the user)
city       (the city provided by the user)
country    (the country information provided by the user)

birthday   (the user's birthday as provided by the user)
gender     (the gender information provided by the user, typically "male" or "female")

mobilephone (the mobile phone number provided by the user)
homephone  (the home phone number provided by the user)

profiling0001 -
profiling0010 (free text field to retrieve stored profiling information on the user)
```

Writing attribute values works analog, but the attributes to be changed and their new values are posted to a different URL in urlencoded style (added line breaks for better readability):

```
POST /idp/attributes/put.htm HTTP/1.1
Host: TENANTID.clouduserstore.com
Content-Type: application/x-www-form-urlencoded

sessionKey=sessionKey&
application_id=APPLICATIONID&
application_secret=APPLICATIONSECRET&
ATTRIBUTE=VALUE&
ATTRIBUTE=VALUE
```