

UNI•Login

Authentication

UNI•Login
Authentication

© UNI•C februar 2012

Index

1	Authentication with UNI•Login	2
1.1	How does it work?	2
1.2	How does the application identify itself to the server?	3
1.3	How does the ticket look?	4
1.4	How to log out of UNI•Login.	5
1.5	Single Login og group login	6
1.6	How to integrate UNI•Login grafically.....	6

1 Authentication with UNI•Login

UNI•Login is a service that provides authentication, access control and user administration to providers of web-based applications in the educational sector. This document describes the technical interface, which the applications must implement in relation to UNI•Login based authentication.

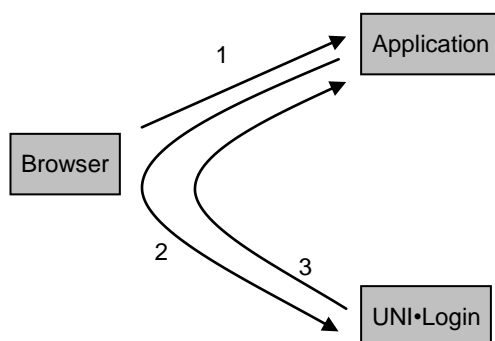
1.1 How does it work?

When an application wants to log in a new user via UNI•Login, the application redirects the user's browser to a UNI•Login-server. This server presents a login form to the user, who enters the username and password:



The image shows a login form for UNI•Login. At the top left, the text "UNI•Login" is displayed. Below it, there are two input fields: "Brugernavn" (Username) and "Adgangskode" (Password). To the right of these fields, there is a "Husk!" (Remember me) section with the text "For at logge ud, skal du lukke browseren." (To log out, you must close the browser.) and a "Support" link. Below the input fields is a "Log ind" (Log in) button. Underneath the button, it says "Du kan også logge ind med:" followed by "NEM ID" and "Digital Signatur". At the bottom of the form, the UNI•C logo is shown, with the text "DANMARKS IT-CENTER FOR UDDANNELSE OG FORSKNING" below it.

If the user's credentials are approved, UNI•Login redirects the user's browser back to the application along with a ticket, which proves that the user is authenticated. The ticket must be validated by the application, which then takes over the responsibility for the rest of the user's session. The principle is illustrated below:



- 1) The user contacts the application
- 2) The application redirects to UNI•Login
- 3) UNI•Login redirects back with a ticket.

UNI•Login supports Single Sign On. When the user has once logged into UNI•Login, he will not have to retype username and password for access to other UNI•Login-applications during the current browser session. The user will automatically be redirected back to additional requesting applications with an authenticating ticket. This functionality is often perceived by the users as if the different UNI•Login applications were integrated. It is possible to avoid this functionality by using Single Login instead of Single Sign On. In this case the user will always be prompted for a password when accessing the application.

1.2 How does the application identify itself to the server?

The application identifies itself to the UNI•Login-server by specifying the parameter "id" in the redirect. E.g. the application redirect could look like this:

<https://sso.emu.dk/unilogin/login.cgi?id=test>

Every application has a unique "id" with a corresponding unique common secret.

The return URL is often specified as a static URL, but some applications need UNI•Login to dynamically return the user to different places in the application.

This may be done by specifying the parameters "path" and "auth" in the redirect from the application to UNI•Login.

Parameter	Beskrivelse
id	Every application has a unique "id"
secret	Unique common secret between UNI•C and the provider of the application
returnURL	Default is specified statically by the configuration, but may be set dynamically by the "path" parameter
path	Contains returnURL. Calculated as $URI_ESCAPE(BASE64(returnURL))$
auth	A fingerprint that authenticates "path" Calculated as $MD5(returnURL+secret)$

Example of a redirect with a dynamic returnUrl:

```
https://sso.emu.dk/unilogin/login.cgi?id=test&
path=aHR0cDovL3d3dy5lbXUuZGsvYXBwbA%3D%3D&
auth=59169cb39fab40cb0ad6ade6a6eb491e
```

In the example "returnURL" is "http://www.emu.dk/appl" and "secret" is "abc123".

The calculations are as follows:

```
path = URI_ESCAPE(BASE64("http://www.emu.dk/appl")) =
URI_ESCAPE("aHR0cDovL3d3dy5lbXUuZGsvYXBwbA==")=
"aHR0cDovL3d3dy5lbXUuZGsvYXBwbA%3D%3D"
```

```
auth = MD5("http://www.emu.dk/applabc123") =
"59169cb39fab40cb0ad6ade6a6eb491e"
```

1.3 How does the ticket look?

The ticket is encoded into the URL, which UNI•Login redirects back to. Below is an example of this:

```
http://www.emu.dk/appl?user=testuser&timestamp=2003050512595&auth=5e55
280df202c8820a7092746b991088
```

The following three parameters are always present in the redirect:

Parameter	Beskrivelse
user	The username of the authenticated user. In the example the username is "testuser"
timestamp	timestamp" – A timestamp in the format YYYYMMDDhhmmss., which specifies the time of the ticket issue in UTC (GMT)
auth	"auth" – A fingerprint that authenticates "user" and "timestamp". The fingerprint is specified in hexadecimal and computes as MD5(timestamp+secret+user), where "secret" is a common secret shared by the application and the UNI•Login-server

When the application receives the ticket with these parameters, the application must verify that the fingerprint is correct and that the ticket has not previously been used.

The application verifies the fingerprint by recalculating MD5(timestamp+secret+user). The value must be the same as received in the

“auth” parameter. In the example above the example the common secret used is “abc123”. Thus the fingerprint gets the following value.

```
auth = MD5("20030505125952abc123testuser") =  
5e55280df202c8820a7092746b991088
```

It is possible to verify that the ticket has not previously been used by storing used tickets. An alternative method is to verify that the ticket was issued within a short time window, e.g. 60 seconds. The window has to be shorter than the time it takes to redirect the user’s browser. If the time is used to verify the validity of the ticket, it is important that the clock is correctly synchronized on the application server. UNI•Login is synchronized with central NTP-servers on the Internet.

If “auth” is valid and the ticket is not a reuse then “user” is authenticated and the application can now begin a session with user, e.g. by issuing a session cookie.

1.4 How to log out of UNI•Login

UNI•Login supports Single Sign On and consequently an application needs to do more than just close its own session with the user on logout. If the user contacts the application again, the user will get access without having to reenter credentials since UNI•Login will still issue a ticket based on the previous login.

In addition to closing the applications own session with the user, it is therefore necessary for the logout-process to redirect to UNI•Logins logout page in order to also close the users session with UNI•Login.

The URL to redirect to in order close the UNI•Login session is:

<https://sso.emu.dk/logout>

Note that UNI•Login does not automatically log the user out of other applications that the user might have visited directly or indirectly during the session. The only way to guarantee that every session is closed is to close the browser completely. Consequently it is the general recommendation when using UNI•Login Single Sign On to encourage the user to close the browser after use.

When logging out of UNI•Login the following warning in Danish will pop up:



1.5 Single Login og group login

If the application wants full control over the login-process it may use Single Login instead of Single Sign On. The consequence is that the users will always be prompted for a password when accessing the application, which may spoil the perception of integration between applications in e.g. a student portal or a portfolio.

Single Login can be used to e.g. implement group login by letting the users log in individually with Single Login, while their identities are tied together in a group by the application.

To use Single Login you must use the server address sli.emu.dk instead of sso.emu.dk. In every other aspect the documentation is unchanged.

1.6 How to integrate UNI•Login graphically

It is recommended that the textual links in the login button and the logout button are written in the font "Arial fat". The text is available in two versions: a short and a long that may be used as appropriate.

The font size is 10px/1em.

The color of the font should be chosen to ensure sufficient contrast to the background.

The buttons are shown below with respectively a light and a dark background:

