OMPASS

REST API - UAF

Content

What is UAF?

Universal Authentication Framework (UAF) is a passwordless login that allows you to simply log in via OMPASS authentication without a password.

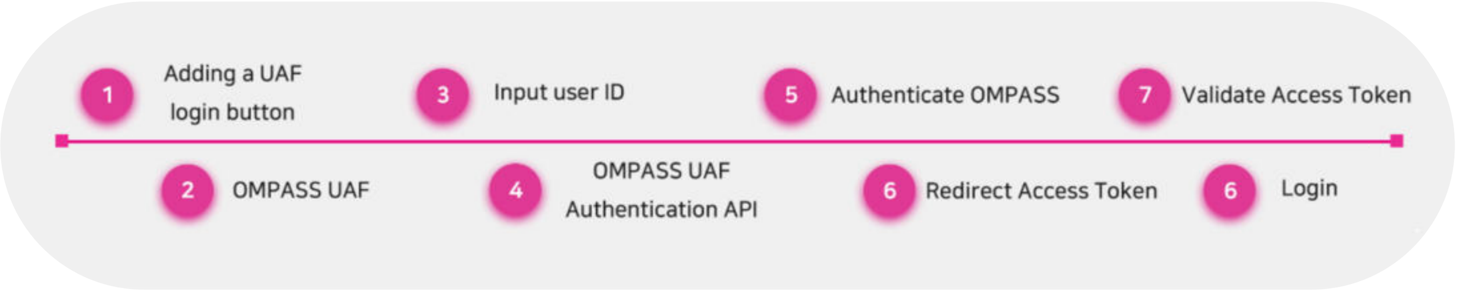
* Please note two requirements below
- UAF is only an optional extra.
- Set U2F first before UAF

Preparation

Please check the secret key frist from “Edit” that is automatically assigned when registering the application from “App Management”.

Please note that security problems may occur if the secret key is exposed to others, so make sure to be private.

Process for Applying OMPASS

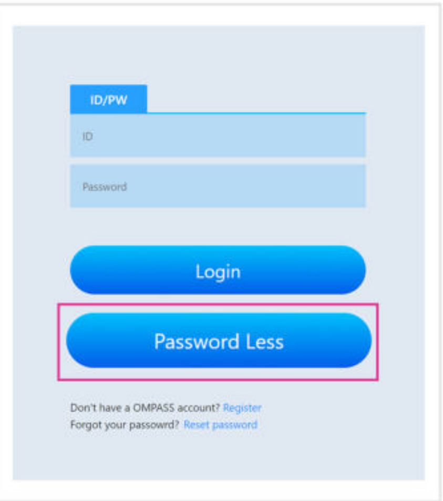


Add UAF login button

From the client-side (user)

In addition to the existing login button, a button for logging in using the UAF method will be added on the login page.
When the [Password Less] button is clicked, this gives a login request to the server-side.

Example



OMPASS-UAF

From the server-side

Once verifying user ID and password is completed from the server-side,
call OMPASS API including the secret key in “HTTP HEADER” and the user ID in “Request Body”.

OMPASS UAF Authentication API

```
POST
URL /v1/ompass/uaf
URL EXAMPLE – https://interface-api.ompasscloud.com/v1/ompass/uaf
```

Header

Key	Type	Description
Authorization	Bearer	· Secret Key assigned to the application · “Bearer” is required to be specified as the authorization type, and a space is required between “Bearer” and “Secret Key” · Example : Bearer djfk39dkfdl39dldjmgjd4idls83jflghidhs83jfk

Example of Request Body

Key	Type	Description
lang_init	String	English (EN) as an Initial language setting value of OMPASS URI to recieve a response

```
{
  “user_id” : “omsecurity”
  “lang_init” : “EN”
}
```


Response (JSON)

· In case of authentication success

Key	Type	Description
ompass_uri	String	A URI of the authentication page in case the user is already registered in OMPASS

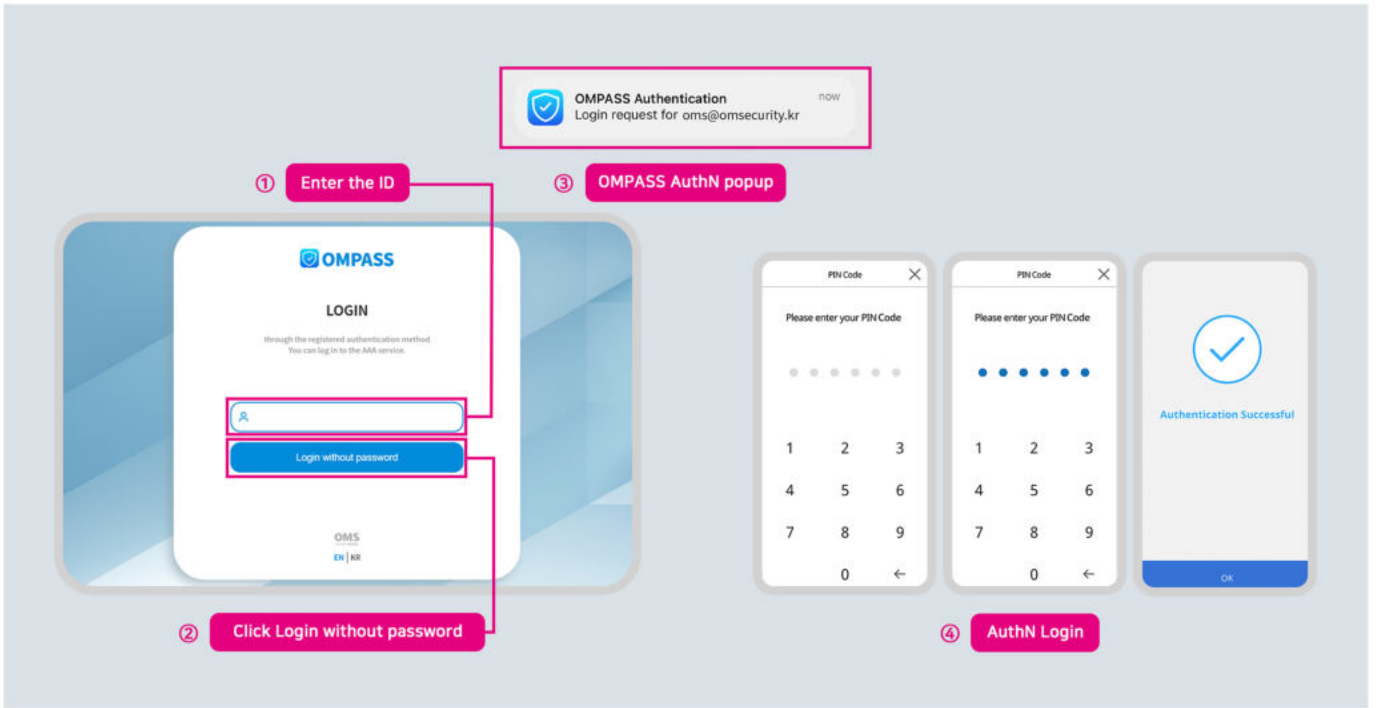
```
{
  "code" : 200,
  "message" : "ok",
  "data" : {
   "ompass_uri" : “https://interface-api.ompasscloud.com/register/did/14?do...”
  }
}
```

OMPASS UAF Authentication

From the client-side (user)

The browser from client-side calls OMPASS URI that is received a response.

Example of authentication interface call in pop-up window
- In case the user is not registered in OMPASS, the following pop-up window will be displayed.



❶ Enter the ID.

❷ Click the “Login without Password” button.

❸ Tap on the notification of OMPASS to launch the app from your phone.

❹ Complete the authentication using previously selected authentication methods.

Redirect Access Token

From the client-side (user)

Whenever OMPASS registration or authentication is successfully completed, HTTP redirects to the redirect URI designated in the application at the pop-up window of the OMPASS page,
and includes a Query String containing the access token.
Parsing the redirected authentication token from client-side will be passing to server-side.

Example

https://{redirect uri}?username={user ID}%20&access_token={Access Token}

Validate Access Token

From the server-side

The server will validate the token by calling the token validation API for OMPASS authentication including the access token passed from the client-side.
If the response for HTTP STATUS CODE of the API request is 200, login will proceed after verifying ID.

OMPASS Success Token Validation API

```
POST
URL /v1/ompass/token-verification
URL EXAMPLE – https://interface-api.ompasscloud.com/v1/ompass/token-verification
```

Header

Key	Type	Description
Authorization	Bearer	· Secret Key assigned to the application · “Bearer” is required to be specified as the authorization type, and a space is required between “Bearer” and “Secret Key” · Example : Bearer djfk39dkfdl39dldjmgjd4idls83jflghidhs83jfk

Request Body (JSON)

Key	Type	Description
user_id	String	User ID
access_token	String	The access token received as a redirect URI

Example of Request Body

```
{
  “user_id” : “omsecurity”,
  “access_token” : “dfj2ld92lldj29cldl29llduuufnbsd229312000dfl2ldio2o019029dj10wj”
}
```

Response (JSON)

· In case of authentication success

Key	Type	Description
user_id	String	User ID

```
{
  "code" : 200,
  "message" : "ok",
  "data" : {
   "user_id" : "omsecurity"
  }
}
```

API Error Messages

code	message
000	Required Request Body is missing.
001	Please make a request including the secret key.
002	Please make a request including the user ID.
003	Please make a request including the access token.
004	Invalid secret key.
005	The secret key format does not match. example) Bearer dl239d29dl292kmdjf139f2ds
006	User ID cannot exceed 30 digits.
011	The token has expired.
012	It is a token of an unsupported format.
013	The token is not configured correctly.
014	Failed to verify the existing signature.