Avoine

Avoine SSO ohje 3. osapuolille

Initial setup

Avoine provides service_id and communications_key.

Login url is this:

https://tunnistus-avoine-fi.pwire.fi/sso-login/?service=<service_id>&return=<return_url>

<service_id> is provided by Avoine
<return_url> is encoded return url on 3rd party service, which will handle the login. Example value is https://www.example.com/processlogin/

When user is logged in succesfully, he will be returned to return url. Unique <ssoid> will be POSTed at the same request. After ssoid is received, 3rd party will need to validate that <ssoid>.

Endpoint

All requests should be POSTed as JSON data to https://tunnistus-avoine-fi.pwire.fi/mmserver

Validating ssoid

All communication between 3rd party service and Avoine SSO will be done through JSON-RPC. First thing to do is validating <ssoid> that is returned by service. You can validate it by posting this data:

```
{
   "id": <request_id_can_be_random>,
   "method": "GetUser",
   "params": [
      <communications_key>,
      <ssoid>
   ],
   "jsonrpc": "2.0"
}
```

Data that will be returned is along these lines:

```
{
   "jsonrpc": "2.0",
   "result": {
      "id": "6c0881f90f9189f9acbe70af040c78e8358b95bb",
      "user_id": "demo-rekisteri_antti@avoine.fi",
      "local_id": "5197",
      "name": "antti@avoine.fi",
      "realm": "demo",
      "idp": "demo-rekisteri",
      "source_service": "demo-service",
      "login_method": "master_login",
      "started": "2013-10-21 10:44:16",
      "finished": null,
```

```
      "last_request": null
   },
   "id": <request_id_returned>
}
```

Remember, that in some cases Avoine SSO has multiple idp:s, so users might come from office LDAP and/or member registry for example.

result should always be an array. Important stuff is inside that result array:

id is <ssoid>. This is also used when requesting more information about the user.
local_id is unique identifier for the user
name is whatever username user used when logged in. Usually this is email address or member number
user_id is combination of idp and name user used while logged in.
realm is the client account
idp is the identity provider that is used. Usually member registry and or office LDAP for example.
source_service is the service that original login emerged
login_method is the actual login method used. Original "username and password" combination is "master_login". Other possibilities are things like "facebook" or "google"

If validating user login is all that is needed, then all is fine now.

Getting more information about user

Usually more information from the user is required. You can get that by sending this request:

```
{
   "id": <request_id_can_be_random>,
   "method": "GetUserData",
   "params": [
      <communications_key>,
      <ssoid>
   ],
   "jsonrpc": "2.0"
}
```

And returned data is along these lines (these can be configured case by case, so there might be more or less data than listed here):

```
{
   "jsonrpc": "2.0",
   "result": {
      "demo-rekisteri.personid": "10000",
      "demo-rekisteri.p_id": "5197",
      "demo-rekisteri.lastname": "Peisa",
      "demo-rekisteri.firstname": "Antti",
      "demo-rekisteri.email": "antti@avoine.fi",
      "groups": [
         "demo-rekisteri.aktiiviset"
      ]
   },
   "id": <request_id_returned>
}
```

Data is always returned in syntax where <idp>.<fieldkey>

Groups is the most important part here, since many times users do get different rights based on the groups they belong to.

Logout

You can logout user by linking to https://tunnistus-avoine-fi.pwire.fi/sso-logout/

There is no way yet to provide return url after logout, but that will be added soon.