OX Cloud hands-on provisioning

Simple guide to OX Cloud provisioning with code snippets (first Perl, Java can be added later).

General

How do I set up admin credentials?

For all examples on this page the provisioning needs admin credentials to be set up.

Required parameters

Admin name (Sub admin)

$oxaasadmname = "<brand>.<environment>.xion.oxcs.net" e.g: testbrand.staging.xion.oxcs.net

Admin password

$oxaasadmpw = <sub_admin_password_plain>

Code snippet

```
my $oxaascreds = SOAP::Data->type("Credentials")->value(
  \SOAP::Data->value(
   SOAP::Data->name( "login" => "$oxaasadmname" ),
   SOAP::Data->name( "password" => "$oxaasadmpw" )
)
);
```

How do I retrieve the service definitions?

To be able to use the webservices, you may need to set up the service definitions.

```
my $OXResellerContextService = SOAP::Lite->ns("http://soap.reseller.admin.openexchange.com") ->proxy(
"https://$oxaashost" . "/webservices/OXResellerContextService", ssl_opts => [ SSL_verify_mode => 1 ] );
my $OXResellerUserService = SOAP::Lite->ns("http://soap.reseller.admin.openexchange.com") ->proxy(
"https://$oxaashost" . "/webservices/OXResellerUserService", ssl_opts => [ SSL_verify_mode => 1 ] );
my $OXaaSService = SOAP::Lite->ns("http://soap.oxaas.admin.openexchange.com/") ->proxy( "https://$oxaashost" . "
/webservices/OXaaSService", ssl_opts => [ SSL_verify_mode => 1 ] );
```

Context

How do I provision a new context?

When you provision a new context, you will always have to specify a context admin for the new context as well as a maximum amount of quota that will be shared among the users of your context. This quota will be used for attachments of calendar appointments and contacts, e.g. and will not affect the user's unified quota. If you want to learn more about data managed in this storage, please check Data managed in file storage.

Required parameters

Context name

$ctxname = "<context_name>" e.g. testbrand.staging.xion.oxcs.net_context1.com

Admin credentials

How do I set up admin credentials?

Quota

$quota = <quota_MB> (-1 = unlimited)

Admin user

User object with at least the following properties:

- Name (<admname>)

- Password (<admpass>)
- Display name (<adm_display_name>)
- Surname (<adm_sur_name>)
- Given name (<adm_given_name>)
- Primary email address (<adm_primaryEmail>)
- Email address (<adm_email>)

Code snippet

```
my $context = SOAP::Data->type("Context")->value(
        \SOAP::Data->value(
            SOAP::Data->name( "name"     => $ctxname ),
            SOAP::Data->name( "maxQuota" => $quota ),
            SOAP::Data->name(
                "userAttributes" =>
                  \SOAP::Data->value( SOAP::Data->name( "entries" => @uattrs ) )
            )
        )
);

my $result = $OXResellerContextService->create(
        $context,
        SOAP::Data->value("User")->value(
            \SOAP::Data->value(
                SOAP::Data->name( "name"         => "<admname>" ),
                SOAP::Data->name( "password"     => "<admpass>" ),
                SOAP::Data->name( "display_name" => "<adm_display_name>" ),
                SOAP::Data->name( "sur_name"     => "<adm_sur_name>"),
                SOAP::Data->name( "given_name"   => "<adm_given_name>" ),
                SOAP::Data->name( "primaryEmail" => "<adm_primaryEmail>" ),
                SOAP::Data->name( "email1"       => "<adm_email>" )
            )
        ),
        $oxaascreds
    );
```

How do I change the Module Access Combination for a context?

Required parameters

Context object

$context = How do I get a context object for my context?

Mod access definition

$modaccess = cloud_pim | cloud_productivity | cloud_security | cloud_productivity_security

Admin credentials

How do I set up admin credentials?

Code snippet

```
## This part is necessary if you are switching to or from a Module Access Combination
## that offers security features (cloud_security and cloud_productivity_security)

my $security_features = SOAP::Data->name(
    "entries" => \SOAP::Data->value(
        SOAP::Data->name( "key"   => "com.openexchange.capability.guard-mail" ),
        SOAP::Data->name( "value" => "true" )
    ),
    "entries" => \SOAP::Data->value(
        SOAP::Data->name( "key"   => "com.openexchange.capability.guard-drive" ),
        SOAP::Data->name( "value" => "false" )
    )
);

my $context = SOAP::Data->type("Context")->value(
    \SOAP::Data->value(
        SOAP::Data->name("id" => $cid),
                ## Add this if you are switching to or from a Module Access Combination that offers security
features
        SOAP::Data->name("userAttributes" => \SOAP::Data->value(
            SOAP::Data->name(
                "entries" => \SOAP::Data->value(
                          SOAP::Data->name( "key" => "config" ),
                          SOAP::Data->name( "value" => \SOAP::Data->value($security_features))
                         )
                )
            )
        )
    )
);
my $result = $OXResellerContextService->changeModuleAccessByName(
        $context,
        $modaccess,
        $oxaascreds
);

if ( $result->fault() ) {
    print STDERR "Error: ".$result->faultstring()."\n";
    next;
}

## You will have to execute this call only if you're switching to or from a Module Access Combination that
offers security features
my $result = $OXResellerContextService->change(
        $context,
        $oxaascreds
);

if ( $result->fault() ) {
    print STDERR "Error: ".$result->faultstring()."\n";
    next;
}
```

How do I allow users in a context to change their password?

This is usually not enabled for users because OX is considering password management to be handled by the partner's provisioning or IDM system. Especially flows like 'forgot password' cannot be covered within the OX Cloud solution. The password change offered here is technically only feasible in non-SSO setups and w/o very specific password policies. If the application internal solution turns out not to be sufficient this option must not be given to the users.

Required parameters

Context object

$context = How do I get a context object for my context?

Admin credentials

How do I set up admin credentials?

Code snippet

```
## Retrieve current module access for the context
my $modaccess = $OXResellerContextService->getModuleAccess(
        $context,
        $oxaascreds
);
my $modaccessperms = $modaccess->paramsall;
## Change editPassword permission
$modaccessperms->{'editPassword'} = "true | false";

## Create soap structure
my @newModuleAccess;
foreach my $key (keys %$modaccessperms) {
  push @newModuleAccess, SOAP::Data->name( $key    => $modaccessperms->{$key} )->type('boolean');
}
## Provision new module access for the context
my $result = $OXResellerContextService->changeModuleAccess(
        $context,
        SOAP::Data->type("UserModuleAccess")->value( \SOAP::Data->value(@newModuleAccess) ),
        $oxaascreds
);
```

How do I delete a context?

Required parameters

Context name

$ctxname = "<context_name>" e.g. testbrand.staging.xion.oxcs.net_context1.com

Admin credentials

How do I set up admin credentials?

Code snippet

```
my $context = SOAP::Data->type("Context")->value(
        \SOAP::Data->value(
            SOAP::Data->name( "name"     => $ctxname ),
            )
);

my $result = $OXResellerContextService->delete(
        $context,
        $oxaascreds
    );
```

How do I provision settings for a context?

Settings can be provisioned when you provision a context or afterwards by calling the change method for a context as described here. On OX Cloud there is only a limited amount of settings available that you can set for your context. You can find a detailed example on how to configure theming here: OX Cloud hands-on theming.

Required parameters

Context object

$context = How do I get a context object for my context?

Admin credentials

How do I set up admin credentials?

```
my %config = (
    <settings_name> => <settings_value>
);

my @attrs;
push @attrs, \SOAP::Data->value(
    SOAP::Data->name("key" => "config"),
    SOAP::Data->name("value" => \SOAP::Data->value(
      map {
        SOAP::Data->name("entries" => \SOAP::Data->value(
                SOAP::Data->name("key" => $_),
                SOAP::Data->name("value" => $config{$_})
        ))
          } keys %config
    ))
);

my $context = SOAP::Data->type("Context")->value(
    \SOAP::Data->value(
        SOAP::Data->name("id" => $cid),
        SOAP::Data->name(
        "userAttributes" =>
           \SOAP::Data->value( SOAP::Data->name( "entries" => @attrs ) )
        )
    )
);

my $result = $OXResellerContextService->change($context, $oxaascreds);
```

How do I find my context id?

Required parameters

Context name

$ctxname = "<context_name>" e.g. testbrand.staging.xion.oxcs.net_context1.com

Admin credentials

How do I set up admin credentials?

Code snippet

```
# get context id
my $context = $OXResellerContextService->list(
    $ctxname,
    $oxaascreds
);

if ( $result->fault() ) {
    print STDERR "Error geting context id with name $ctxname: ".$context->faultstring()."\n";
    next;
}

my @context = $context->paramsall;
my $cid = $context[0]->{'id'};
```

How do I get a context object for my context?

For most provisioning calls you will need a context object for the request.

Required parameters

Context ID

$cid = How do I find my context id?

Code snippet

```
my $context = SOAP::Data->type("Context")->value(
        \SOAP::Data->value(
            SOAP::Data->name( "id"     => $cid )
        )
    );
```

User

How do I provision a new user?

Required parameters

Admin credentials

How do I set up admin credentials?

Context object

$context = How do I get a context object for my context?

User object

User object with at least the following properties. Those are mandatory but not in every case reported as such if missing. Please be careful.

- User name (<username>)
- Display name (<displayname>)
- Sur name for the user (<surname>)
- Given name for the user (<givenname>)
- Password for the user (<password>)
- Primary mail address (<email>)
- Mail address (<email>)
- Language (e.g. en_US) (<language>)
- Timezone (e.g. Europe/Berlin) (<timezone>)
- Max filestorage quota in MB (<maxQuota>) (For Unified Quota this value MUST be defined during user creation and it must be identical with the quota defined for mail.)

 # Mod access definition

$modaccess = cloud_pim | cloud_productivity | cloud_security | cloud_productivity_security

Service Class

$serviceclass = Has to match $modaccess unless adviced differently

Code snippet

```
my @soapdata;
push @soapdata, SOAP::Data->name( "name"                    => <username> )->type('string');
push @soapdata, SOAP::Data->name( "password"                => <password> );
push @soapdata, SOAP::Data->name( "display_name"            => <displayname> )->type('string');
push @soapdata, SOAP::Data->name( "sur_name"                => <surname> )->type('string');
push @soapdata, SOAP::Data->name( "given_name"              => <givenname> )->type('string');
push @soapdata, SOAP::Data->name( "primaryEmail"            => <email> );
push @soapdata, SOAP::Data->name( "email1"                  => <email> );
push @soapdata, SOAP::Data->name( "language"             => <language> );
push @soapdata, SOAP::Data->name( "timezone"             => <timezone> );
push @soapdata, SOAP::Data->name( "maxQuota"             => <maxquota> );

my $qunified = SOAP::Data->name(
    "entries" => \SOAP::Data->value(
        SOAP::Data->name( "key"   => "com.openexchange.unifiedquota.enabled" ),
        SOAP::Data->name( "value" => "true" )
    )
);
my @classofservice;
push @classofservice, \SOAP::Data->value(
    SOAP::Data->name("key" => "cloud"),
    SOAP::Data->name("value" => \SOAP::Data->value(
        SOAP::Data->name("entries" => \SOAP::Data->value(
                SOAP::Data->name("key" => "service"),
                SOAP::Data->name("value" => $serviceclass)
        ))
    ))
);
push @soapdata,
    SOAP::Data->name(
    "userAttributes" => \SOAP::Data->value(
        SOAP::Data->name(
            "entries" => \SOAP::Data->value(
                     SOAP::Data->name( "key" => "config" ),
                     SOAP::Data->name( "value" => \SOAP::Data->value($qunified))
                    ),
                    @classofservice
            )
        )
    );

my $user = SOAP::Data->value("User")->value(
            \SOAP::Data->value(@soapdata)
            );

my $result = $OXResellerUserService->createByModuleAccessName(
        $context,
        $user,
        $modaccess,
        $oxaascreds
);

if( $result->fault() ) {
        print $cid;
        print $result->faultstring()."\n";
    exit(1);
}

my @results = $result->paramsall;

# set email quota for the new user or, if unified quota is enabled, activate
# unified quota valid for mail and filestorage
$result = $OXaaSService->setMailQuota(
        SOAP::Data->name( "ctxid" => $cid ),
        SOAP::Data->name( "usrid" => $results[0]->{'id'} ),
        SOAP::Data->name( "quota" => $maxQuota ),
        $oxaascreds
);
```

How do I change the Module Access Combination for a user?

Required parameters

Admin credentials

How do I set up admin credentials?

Context object

$context = How do I get a context object for my context?

Mod access definition

$modaccess = cloud_pim | cloud_productivity | cloud_security | cloud_productivity_security

Service Class

$serviceclass = Has to match $modaccess

User

$user = Object containing the user's login (<user_login>)

```
my @soapdata;
push @soapdata, SOAP::Data->name( "name" => <user_login> )->type('string');

my @classofservice;
push @classofservice, \SOAP::Data->value(
    SOAP::Data->name("key" => "cloud"),
    SOAP::Data->name("value" => \SOAP::Data->value(
        SOAP::Data->name("entries" => \SOAP::Data->value(
                SOAP::Data->name("key" => "service"),
                SOAP::Data->name("value" => $serviceclass)
        ))
    ))
);

push @soapdata,
    SOAP::Data->name(
    "userAttributes" => \SOAP::Data->value(
        SOAP::Data->name(
            "entries" => @classofservice
            )
        )
    );

my $user = SOAP::Data->value("User")->value(
    \SOAP::Data->value(
        SOAP::Data->value(@soapdata)
    )
);

my $result = $OXResellerUserService->changeByModuleAccessName(
        $context,
        $user,
        $modaccess,
        $oxaascreds
);

if ( $result->fault() ) {
    print STDERR "Error: ".$result->faultstring()."\n";
    next;
}
## You will have to execute this call only if you're switching to or from a Module Access Combination that
offers security features
my $result = $OXResellerUserService->change(
        $context,
        $user,
        $oxaascreds
);

if ( $result->fault() ) {
    print STDERR "Error: ".$result->faultstring()."\n";
    next;
}
```

How do I enable user permissions?

Required parameters

Admin credentials

How do I set up admin credentials?

Context id

$cid = How do I find my context id?

User id

$uid = How do I get my user id?

Permissions to enable

@disablePerms = array of permissions that should be enabled

Code snippet

```
my @enablePerms;
push(@enablePerms, SOAP::Data->name( perms => "SEND" ));
push(@enablePerms, SOAP::Data->name( perms => "RECEIVE" ));
push(@enablePerms, SOAP::Data->name( perms => "MAILLOGIN" ));
push(@enablePerms, SOAP::Data->name( perms => "WEBLOGIN" ));

my $result = $OXaaSService->enablePermissions(
    SOAP::Data->name( "ctxid" => $cid ),
    SOAP::Data->name( "usrid" => $uid ),
    SOAP::Data->name( "perms" => @enablePerms ),
    $oxaascreds
);
```

How do I disable user permissions?

Before permanently deleting a user you may want to consider to just disable some or all of his permissions. This would not delete the user's data and so it could be reactivated without data loss afterwards.

Required parameters

Admin credentials

How do I set up admin credentials?

Context id

$cid = How do I find my context id?

User id

$uid = How do I get my user id?

Permissions to disable

@disablePerms = array of permissions that should be disabled

Code snippet

```
my @disablePerms;
push(@disablePerms, SOAP::Data->name( perms => "SEND" ));
push(@disablePerms, SOAP::Data->name( perms => "RECEIVE" ));
push(@disablePerms, SOAP::Data->name( perms => "MAILLOGIN" ));
push(@disablePerms, SOAP::Data->name( perms => "WEBLOGIN" ));

my $result = $OXaaSService->disablePermissions(
    SOAP::Data->name( "ctxid" => $cid ),
    SOAP::Data->name( "usrid" => $uid ),
    SOAP::Data->name( "perms" => @disablePerms ),
    $oxaascreds
);
```

How do I delete a user?

Before permanently deleting a user you may want to consider to just disable some or all of his permissions. This would not delete the user's data and so the user could be reactivated without data loss afterwards.

Required parameters

Context object

$context = How do I get a context object for my context?

User

$user = Object containing the user's login (<user_login>)

Admin credentials

How do I set up admin credentials?

Code snippet

```
my $user = SOAP::Data->value("User")->value(
            \SOAP::Data->value(
                SOAP::Data->name( "name"            => <user_login> )
                )
            );

my $result = $OXResellerUserService->delete(
        $context,
        $user,
        $oxaascreds
);
```

How do I provision a setting for a user?

Required parameters

Context object

$context = How do I get a context object for my context?

User

$user = Object containing the user's login (<user_login>)

Admin credentials

How do I set up admin credentials?

Code snippet

```
my %config = (
    <settings_name> => <settings_value>
);
my @attrs;
push @attrs, \SOAP::Data->value(
    SOAP::Data->name("key" => "config"),
    SOAP::Data->name("value" => \SOAP::Data->value(
        map {
            SOAP::Data->name("entries" => \SOAP::Data->value(
                    SOAP::Data->name("key" => $_),
                    SOAP::Data->name("value" => $config{$_})
            ))
        } keys %config
    ))
);

my $user = SOAP::Data->value("User")->value(
    \SOAP::Data->value(
        SOAP::Data->name("name"            => <user_login>),
        SOAP::Data->name(
        "userAttributes" =>
           \SOAP::Data->value( SOAP::Data->name( "entries" => @attrs ) )
        )
    )
);
my $result = $OXResellerUserService->change(
        $context,
        $user,
        $oxaascreds
    );
```

How do I change a user password?

Required parameters

Context object

$context = How do I get a context object for my context?

User

$user = Object containing the user's login (<user_login>) and the new password (<user_password>)

Admin credentials

How do I set up admin credentials?

Code snippet

```
my $user = SOAP::Data->value("User")->value(
            \SOAP::Data->value(
                SOAP::Data->name( "name"            => <user_login> );
                SOAP::Data->name( "password"      => <user_password> )
                )
            );

my $result = $OXResellerUserService->change(
        $context,
        $user,
        $oxaascreds
    );
```

How do I get my user id?

Required parameters

Context object

$context = How do I get a context object for my context?

User

$user = Object containing the user's login (<user_login>)

Admin credentials

How do I set up admin credentials?

```
my $user = $OXResellerUserService->getData(
    $context,
    SOAP::Data->value("User")->value(
        \SOAP::Data->value(
            SOAP::Data->name( "name"            => <user_login> )
        )
    ),
    $oxaascreds
);
my @user = $user->paramsall;

my $uid = $user[0]->{'id'};
```

Own Brand configuration

Beginning with version 7.10.5, it is possible to define some settings that used to be applied to every context before also into the own brand. You can now define all so called Configuration Cascade settings or `userAttributes` globally in your own brand entry.

Retrieve own brand configuration

The following example shows how to retrieve that data using the new SOAP API call `getSelfData`:

```
#!/usr/bin/perl -w

BEGIN { $ENV{PERL_LWP_SSL_VERIFY_HOSTNAME} = 0 }

use strict;
use SOAP::Lite;
use Data::Dumper;

my $soap = SOAP::Lite->ns("http://soap.reseller.admin.openexchange.com")->proxy("https://$oxaashost/webservices
/OXResellerService", ssl_opts => [ SSL_verify_mode => 0 ]);

my $oxaasadmname = "example.com";
my $oxaasadmpw = "secret";

my $creds = SOAP::Data->type("Credentials")->value(
                            \SOAP::Data->value(
                             SOAP::Data->name("login" => $oxaasadmname),
                             SOAP::Data->name("password" => $oxaasadmpw)));


my $ret = $soap->getSelfData($adminName, $adminId, $creds);
if( $ret->fault() ) {
    print $ret->faultstring()."\n";
} else {
    print Dumper($ret->paramsall);
}
```

Change own brand configuration

See the example below on how to add or remove settings from your own brand using the new SOAP API method `changeSelf`:

There's currently one caveat, that you have to know the Id of your own brand, but that cannot yet programmatically retrieved, so you have to ask us for it.

```
#!/usr/bin/perl -w

BEGIN { $ENV{PERL_LWP_SSL_VERIFY_HOSTNAME} = 0 }

use strict;
use SOAP::Lite;
use Data::Dumper;

my $soap = SOAP::Lite->ns("http://soap.reseller.admin.openexchange.com")->proxy("https://$oxaashost/webservices
/OXResellerService", ssl_opts => [ SSL_verify_mode => 0 ]);

my $oxaasadmname = "example.com";
my $oxaasadmpw = "secret";
my $oxaasadmid = "83";
my $configName = "com.openexchange.appsuite.servercontact";
my $configValue = "This server is operated by me";
my $add = 1;

my $creds = SOAP::Data->type("Credentials")->value(
                            \SOAP::Data->value(
                             SOAP::Data->name("login" => $oxaasadmname),
                             SOAP::Data->name("password" => $oxaasadmpw)));

my $config = SOAP::Data->name("entries" => \SOAP::Data->value(
            SOAP::Data->name( "key" => $configName ),
            SOAP::Data->name( "value" => $configValue)));

my $resellerAdm;
if ($add) {
    $resellerAdm = SOAP::Data->type("ResellerAdmin")->value(
                                \SOAP::Data->value(
                                SOAP::Data->name("name" => $oxaasadmname),
                                SOAP::Data->name("id" => $oxaasadmid),
                                SOAP::Data->name("configurationToAdd" =>  \SOAP::Data->value($config))
                                ));
} else {
    $resellerAdm = SOAP::Data->type("ResellerAdmin")->value(
                                \SOAP::Data->value(
                                SOAP::Data->name("name" => $oxaasadmname),
                                SOAP::Data->name("id" => $oxaasadmid),
                                SOAP::Data->name("configurationToRemove" =>  $configName)
                                ));
}


my $ret = $soap->changeSelf($resellerAdm, $creds);
if( $ret->fault() ) {
    print $ret->faultstring()."\n";
} else {
    print Dumper($ret->paramsall);
}
```