

OX Cloud Module Access Combinations and Service classes

In OX App Suite a Module Access Combination and a Service class defines which modules a user can access. For OX Cloud, there are 4 predefined Module Access Combinations that an administrator can choose from, when provisioning a user: **Webmail + PIM**, **Add-on Productivity**, **Add-on Security**, **Add-on Productivity & Security**.

The following matrix shows which modules are included in the packages:

Module	Mail & PIM (cloud_pim)	Add-on Productivity (cloud_productivity)	Add-on Security (cloud_security)
	(cloud_productivity_security)		
OX Portal	X		
OX Mail	X		
OX Address Book (shared and public)	X		
OX Calendar (shared and public)	X		
OX Tasks (shared and public)	X		
OX Document Viewer	X		
OX Sync (CalDAV/CardDAV, OX Sync App)	X		
OX Drive (shared and public) OX Drive App for different platforms OX Documents	-	X	-
OX Guard	-	-	X

How do I provision a Module Access Combination?

A module access combination can be provisioned using the create- or changeByModuleAccessName call in the OXResellerContextService (How do I retrieve the Service Definitions?).

In addition to the Module Access Combination a Service Class needs to be defined. Usually the name is identical to the Module Access Combination name (unless OX advises differently). How the Service Class is provisioned is shown in the examples below.

How do I provision a context with a Module Access Combination?

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)

Mod access definition

\$modaccess = cloud_pim | cloud_productivity | cloud_security | cloud_productivity_security

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>)

Sample code (Perl)

```

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

my $result = $OXResellerContextService->createModuleAccessByName(
    $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>" )
        )
    ),
    $modaccess,
    $oxaascreds
);

```

How do I provision a user with a Module Access Combination and a Service Class?

This guide will show you how to provision a user with a Module Access Combination that differs from the context's Module Access Combination. If your user should have the same as the context, you can also just provision a user with the [create call](#).

Required parameters

Context id

\$context = [How do I get a context object for my context?](#)

User

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

Mod access definition

\$modaccess = cloud_pim | cloud_productivity | cloud_security | cloud_productivity_security

Service Class

\$serviceclass = Has to match \$modaccess (unless adviced differently by OX)

Admin credentials

\$oxaascreds = Link to "How to define admin credentials?"

Sample code (Perl)

```

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

# use code below for unified quota
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 ( $result->fault() ) {
    print STDERR "Error: ".$result->faultstring()."\n";
    next;
}

# 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
);

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

```