# Office AMS: Manage users and groups

|  |  |
| --- | --- |
| Summary: | Applies to: |
| This scenario shows how you can manage groups and users inside a site collection | * Office 365 Multi-Tenant (MT) * Office 365 Dedicated (D) * SharePoint 2013 on-premises |
| Solution: | Core.GroupManagement, version 1.0 |
| Author: | Bert Jansen (Microsoft) |
| //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  // THIS CODE IS PROVIDED \*AS IS\* WITHOUT WARRANTY OF  // ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING ANY  // IMPLIED WARRANTIES OF FITNESS FOR A PARTICULAR  // PURPOSE, MERCHANTABILITY, OR NON-INFRINGEMENT.  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* | |

# General comments

This shows how you can work with groups and users by adding/removing them to the site collection. Adding users and groups in one thing, next thing is giving them a permission level so that they’ve access to SharePoint and that’s also shown in this sample. The actual implementation of these group/user/permission level actions is implemented in Office AMS Core as extension methods. This results in a very easy to use model which is shown via below code snippets.

## Adding groups and adding users to groups

cc.Load(cc.Web, web => web.CurrentUser);

cc.ExecuteQuery();

Microsoft.SharePoint.Client.User currentUser = cc.Web.CurrentUser;

if (!cc.Web.GroupExists("Test"))

{

Group group = cc.Web.AddGroup("Test", "Test group", true);

cc.Web.AddUserToGroup("Test", currentUser.LoginName);

}

## Removing groups

if (cc.Web.GroupExists("Test"))

{

cc.Web.RemoveGroup("Test");

}

## removing users from groups

cc.Load(cc.Web, web => web.CurrentUser);

cc.ExecuteQuery();

Microsoft.SharePoint.Client.User currentUser = cc.Web.CurrentUser;

if (cc.Web.GroupExists("Test"))

{

if (cc.Web.IsUserInGroup("Test", currentUser.LoginName))

{

cc.Web.RemoveUserFromGroup("Test", currentUser.LoginName);

}

}

## Add permission level to group

if (cc.Web.GroupExists("Test"))

{

cc.Web.AddPermissionLevelToGroup("Test", RoleType.Contributor);

}

## Add permission level to user

cc.Load(cc.Web, web => web.CurrentUser);

cc.ExecuteQuery();

Microsoft.SharePoint.Client.User currentUser = cc.Web.CurrentUser;

cc.Web.AddPermissionLevelToUser(currentUser.LoginName, RoleType.Reader);

## Remove permission level from group

if (cc.Web.GroupExists("Test"))

{

cc.Web.RemovePermissionLevelFromGroup("Test", RoleType.Reader);

}

## Remove permission level from user

cc.Load(cc.Web, web => web.CurrentUser);

cc.ExecuteQuery();

Microsoft.SharePoint.Client.User currentUser = cc.Web.CurrentUser;

cc.Web.RemovePermissionLevelFromUser(currentUser.LoginName, RoleType.Reader);