Active.Directory.Utility: Exposed Methods

Summary of the API methods listed in Swagger. Generated from the project source, November 2026.

49 methods are exposed in Swagger. Use the Full Swagger Reference for every input field and schema.

Conventions

- **Base URL:** your API host, for example `https://api.corp.example.com`. HTTPS is required.
- **Authentication:** send your API key in the `X-Api-Key` header on every call. `/api/HealthCheck` does not need it unless the administrator turned that on.
- **Input and output:** JSON. Query-string parameter names are not case-sensitive.
- **Required parameters** are marked Yes in the Required column; the API returns HTTP 400 when one is missing or empty.
- **Response envelope:** every method answers with `{ "Success": true|false, "Data": ..., "Error": null|"message" }`. `Data` holds the method-specific result. `/api/HealthCheck` returns its report directly (see Schemas).
- **LDAP_Path:** `LDAP://OU=Users,DC=corp,DC=example,DC=com` or just the distinguished name. Any host part is ignored; the server always comes from configuration.
- **UserName:** a SAMAccountName or a UserPrincipalName.

HTTP status	Meaning
200	Success
400	Missing or invalid input, or the AD command failed
401	Missing or invalid `X-Api-Key`
404	The user, group, OU or object was not found
503	The PowerShell remoting (WinRM) endpoint could not be reached
504	The request exceeded the server time limit
500	Unexpected server error

Health

Method	URL	What it does
GET	`/api/HealthCheck`	**Health check.** Checks PowerShell on the web server, WinRM and LDAPS connectivity, and an Active Directory PowerShell session on the target server.

Users

Method	URL	What it does
POST	`/api/CreateUser`	**Create a user.** Creates a user in the OU given by LDAP_Path. AccountPassword is required and is never logged.
DELETE	`/api/DeleteUser`	**Delete a user.** Permanently deletes the user.
PUT	`/api/EnableUserAccount`	**Enable a user account.** Enables the account.
PUT	`/api/DisableUserAccount`	**Disable a user account.** Disables the account.
PUT	`/api/UnlockUserAccount`	**Unlock a user account.** Unlocks an account that is locked out.
GET	`/api/GetUsersInOU`	**Get users in an OU.** Lists users directly inside the OU (one level).
GET	`/api/ListOfUsers`	**List users.** Lists users under LDAP_Path whose name contains the search text, or all with *.
GET	`/api/Search4User`	**Search for users.** Finds users where the chosen attribute (SearchVIA) contains the search text.
GET	`/api/GetUserAccountStatus`	**Get a user's account status.** Returns enabled, locked-out, password and expiration details.
GET	`/api/GetUserAttribute`	**Get one user attribute.** Returns the value of a single LDAP attribute.
GET	`/api/GetUserAttributes`	**Get all user attributes.** Returns all attributes of the user (password hashes are never returned).
PUT	`/api/UpdateUserAttributes`	**Update user attributes.** Sets attributes from the JSON body. A null or empty value clears the attribute. Password attributes are blocked.
GET	`/api/GetUserPrincipalName`	**Get a user's UserPrincipalName.** Looks up the UPN for a sAMAccountName.
GET	`/api/GetUserSAMAccountName`	**Get a user's SAMAccountName.** Looks up the sAMAccountName for a UserPrincipalName.
GET	`/api/GetDaysUntilUserPasswordExpires`	**Days until password expires.** Returns the number of days before the password expires, or null if it never expires.
GET	`/api/AccountStatuses`	**Find accounts by status.** Finds users that are locked out, disabled, inactive, expiring, expired, etc. Statuses can be combined.
PUT	`/api/SetAccountExpiration`	**Set or clear account expiration.** Sets or clears the expiration date on all users matching Filter. Use Preview=true first.

Passwords and authentication

Method	URL	What it does
POST	`/api/AuthenticateUser`	**Authenticate a user.** Validates a user name and password. Failed attempts count toward account lockout.
PUT	`/api/ChangeUserPassword`	**Change a user's password.** Changes the password; the current password is required.
PUT	`/api/ResetUserPassword`	**Reset a user's password.** Sets a new password without needing the current one.

Groups

Method	URL	What it does
POST	`/api/CreateGroup`	**Create a group.** Creates a group. GroupScope: DomainLocal, Global or Universal. GroupType: Security or Distribution.
DELETE	`/api/DeleteGroup`	**Delete a group.** Permanently deletes the group.
GET	`/api/GetGroupsInOU`	**Get groups in an OU.** Lists groups directly inside the OU (one level).
GET	`/api/ListOfGroups`	**List groups.** Lists groups under LDAP_Path whose name contains the search text, or all with *.
GET	`/api/Search4Group`	**Search for groups.** Finds groups whose name, sAMAccountName or display name contains the search text.

Group membership (role-based security)

Method	URL	What it does
PUT	`/api/AddUserToGroup`	**Add a user to a group.** Makes the user a member of the group.
DELETE	`/api/RemoveUserFromGroup`	**Remove a user from a group.** Removes the user from the group.
GET	`/api/IsUserInGroup`	**Is a user in a group.** Checks direct membership only (not nested groups or primary group).
GET	`/api/GetUserGroups`	**Get a user's groups.** Lists the groups the user is a member of.
GET	`/api/GetUsersInGroup`	**Get users in a group.** Lists the users that are direct members of a group.
PUT	`/api/AddGroupToGroup`	**Add a group to a group.** Makes ChildGroupName a member of ParentGroupName.
DELETE	`/api/RemoveGroupFromGroup`	**Remove a group from a group.** Removes ChildGroupName from ParentGroupName.
GET	`/api/IsGroupInGroup`	**Is a group in a group.** Checks direct membership only (not nested groups).
GET	`/api/GetGroupsInGroup`	**Get groups in a group.** Lists the groups that are direct members of a group.

Organizational units

Method	URL	What it does
POST	`/api/CreateOU`	**Create an OU.** Creates an organizational unit under LDAP_Path (or under Path, if supplied).
GET	`/api/GetOUsInOU`	**Get OUs in an OU.** Lists OUs directly inside the OU (one level).
GET	`/api/ListOfOUs`	**List OUs.** Lists OUs under LDAP_Path whose name contains the search text, or all with *.

Objects

Method	URL	What it does
GET	`/api/DoesObjectExists`	**Does an object exist.** Checks whether any object matches the name exactly (sAMAccountName, cn, name or UPN).
GET	`/api/GetObjectAttributes`	**Get all attributes of an object.** Returns all attributes of the object(s) matching the name.
GET	`/api/Search4Object`	**Search for objects.** Finds users, OUs, groups and/or computers whose name contains the search text.

Fine-grained password policies

Method	URL	What it does
POST	`/api/CreateFineGrainedPasswordPolicy`	**Create a fine-grained password policy.** Creates a policy (Precedence is required). TimeSpans use d.hh:mm:ss.
PUT	`/api/UpdateFineGrainedPasswordPolicy`	**Update a fine-grained password policy.** Changes only the settings supplied in the body.
DELETE	`/api/DeleteFineGrainedPasswordPolicy`	**Delete a fine-grained password policy.** Permanently deletes the policy.
GET	`/api/GetFineGrainedPasswordPolicy`	**Get a fine-grained password policy.** Returns the policy by name.
GET	`/api/GetFineGrainedPasswordPolicyEX`	**Get a fine-grained password policy (extended).** Returns the policy by distinguished name, plus the users/groups it applies to.
POST	`/api/ApplyFineGrainedPasswordPolicy`	**Apply a fine-grained password policy.** Applies the policy to a security group.
DELETE	`/api/RemoveFineGrainedPasswordPolicyFrom`	**Remove a fine-grained password policy.** Removes the policy from a user or group.
GET	`/api/GetUserFineGrainedPasswordPolicy`	**Get a user's password policy.** Returns the policy that effectively applies to the user.