Served by: DEVWEB01

Select a definition: Active.Directory.Utility v5

Active.Directory.Utility

v5 OAS 3.0

/swagger/v5/swagger.json

JSON API over PowerShell ActiveDirectory cmdlets. Required parameters are marked with *.



Authorize

Active.Directory.Utility

GET	/api/HealthCheck	Health check
PUT	/api/AddGroupToGroup	Add a group to a group
PUT	/api/AddUserToGroup	Add a user to a group
DELETE	/api/RemoveGroupFromGroup	Remove a group from a group
DELETE	/api/RemoveUserFromGroup	Remove a user from a group
GET	/api/IsGroupInGroup	Is a group in a group

Served by: DEVWEB01

GET	`/api/IsUserInGroup`	Is a user in a group
GET	`/api/GetGroupsInGroup`	Get groups in a group
GET	`/api/GetUsersInGroup`	Get users in a group
GET	`/api/GetUserGroups`	Get a user's groups
POST	`/api/CreateGroup`	Create a group
DELETE	`/api/DeleteGroup`	Delete a group
GET	`/api/GetGroupsInOU`	Get groups in an OU
GET	`/api/Search4Group`	Search for groups
POST	`/api/CreateOU`	Create an OU
GET	`/api/GetOUsInOU`	Get OUs in an OU
POST	`/api/CreateUser`	Create a user
DELETE	`/api/DeleteUser`	Delete a user
PUT	`/api/DisableUserAccount`	Disable a user account
PUT	`/api/EnableUserAccount`	Enable a user account
PUT	`/api/UnlockUserAccount`	Unlock a user account
GET	`/api/GetUsersInOU`	Get users in an OU
GET	`/api/Search4User`	Search for users

Served by: DEVWEB01

GET	`/api/GetUserAccountStatus`	Get a user's account status
GET	`/api/GetUserAttribute`	Get one user attribute
GET	`/api/GetUserAttributes`	Get all user attributes
PUT	`/api/UpdateUserAttributes`	Update user attributes
GET	`/api/GetUserPrincipalName`	Get a user's UserPrincipalName
GET	`/api/GetUserSAMAccountName`	Get a user's SAMAccountName
GET	`/api/GetDaysUntilUserPasswordExpires`	Days until password expires
POST	`/api/AuthenticateUser`	Authenticate a user
PUT	`/api/ChangeUserPassword`	Change a user's password
PUT	`/api/ResetUserPassword`	Reset a user's password
GET	`/api/Search4Object`	Search for objects
GET	`/api/AccountStatuses`	Find accounts by status
GET	`/api/ListOfOUs`	List OUs
GET	`/api/ListOfGroups`	List groups
GET	`/api/ListOfUsers`	List users
PUT	`/api/SetAccountExpiration`	Set or clear account expiration
GET	`/api/DoesObjectExists`	Does an object exist

Served by: DEVWEB01

GET /api/GetObjectAttributes Get all attributes of an object

POST /api/CreateFineGrainedPasswordPolicy Create a fine-grained password policy

PUT /api/UpdateFineGrainedPasswordPolicy Update a fine-grained password policy

DELETE /api/DeleteFineGrainedPasswordPolicy Delete a fine-grained password policy

GET /api/GetFineGrainedPasswordPolicy Get a fine-grained password policy

GET /api/GetFineGrainedPasswordPolicyEX Get a fine-grained password policy (extended)

POST /api/ApplyFineGrainedPasswordPolicy Apply a fine-grained password policy

DELETE /api/RemoveFineGrainedPasswordPolicyFrom Remove a fine-grained password policy

GET /api/GetUserFineGrainedPasswordPolicy Get a user's password policy

Schemas

AuthenticateUserRequest

```
{
    LDAP_Path*          [...]
    UserName*           [...]
    UserPassword*       [...]
}
```

ChangeUserPasswordRequest

```
{
    LDAP_Path*          [...]
    UserName*           [...]
    UserCurrentPassword* [...]
    UserNewPassword*    [...]
}
```

Served by: DEVWEB01

FineGrainedPasswordPolicyRequest {

Name* [...]

Precedence [...]

ComplexityEnabled [...]

LockoutDuration [...]

LockoutObservationWindow [...]

LockoutThreshold [...]

MaxPasswordAge [...]

MinPasswordAge [...]

MinPasswordLength [...]

PasswordHistoryCount [...]

ReversibleEncryptionEnabled [...]

ProtectedFromAccidentalDeletion [...]

DisplayName [...]

Description [...]

}

Served by: DEVWEB01

NewUserRequest {

SamAccountName*	[...]
AccountPassword*	[...]
Name	[...]
UserPrincipalName	[...]
GivenName	[...]
Surname	[...]
DisplayName	[...]
Description	[...]
EmailAddress	[...]
Title	[...]
Department	[...]
Company	[...]
Office	[...]
OfficePhone	[...]
MobilePhone	[...]
EmployeeID	[...]
Manager	[...]
Enabled	[...]
PasswordNeverExpires	[...]
ChangePasswordAtLogon	[...]
CannotChangePassword	[...]

}

ResetUserPasswordRequest {

LDAP_Path*	[...]
UserName*	[...]
UserPassword*	[...]

}

UserAttributeRequest {

Attributes*	{...}

}