

Cheatsheet: Cloudfoundry UAA

VMWARE

- PDF Link: [cheatsheet-uaa-A4.pdf](#), Category: tools
- Blog URL: <https://cheatsheet.dennyzhang.com/cheatsheet-uaa-A4>
- Related posts: Bosh CheatSheet, PKS CheatSheet, #denny-cheatsheets

File me Issues or star this repo.

1.1 uaac

| Name | Command |
|-------------------------------|---|
| Run command | <code>uaac apt-install ruby ruby-dev build-essential rubygems; gem install uaac-cf</code> |
| Set a new target | <code>uaac target https://<uaa-dnsname>:8443 --skip-ssl-validation</code> |
| List all context | <code>uaac contexts</code> |
| Log in as a UAA admin | <code>uaac token client get admin -s <uaa-management-admin-client></code> Link: Log in as a UAA Admin |
| List users | <code>uaac users</code> |
| List all client registrations | <code>uaac clients</code> |
| Add user | <code>uaac user add alana --emails alana@example.com -p password</code> Link: Grant PKS Access to a User |
| Add user to group | <code>uaac member add pks.clusters.admin alana</code> Link: Grant PKS Access to a User |
| Get specific user account | <code>uaac user get alana</code> |
| Get user | <code>uaac user get pks-admin</code> |
| Reference | Bosh CheatSheet, Tile CheatSheet, UAA CheatSheet, PKS CheatSheet |

1.2 UAA Concept

| Name | Command |
|----------|---------|
| target | |
| context | |
| client | |
| token | |
| password | |
| secret | |

1.3 credhub

| Name | Command |
|---------------|---|
| CredHub | It centralizes and secures credential generation, storage, lifecycle management, and access. |
| Reference | GitHub: credhub, GitHub: credhub-cli, Link: credhub |
| credhub login | <code>credhub login --client-name credhub-admin --client-secret "admin_secret" --s "\${api_url}" --ca-cert "\${ca_file}"</code> |

<https://raw.githubusercontent.com/dennyzhang/cheatsheet.dennyzhang.com/master/cheatsheet-uaa-A4/credhub-service.png>

1.4 Online Help Usage

kubo@jumper:~\$ uaac --help

UAA Command Line Interface

Miscellaneous

| | |
|--------------------------------------|---|
| <code>version</code> | Display version |
| <code>help [topic command...]</code> | Display summary or details of command or topic |
| <code>target [uaa_url]</code> | Display current or set new target |
| | <code>-f --[no-]force</code> , set even if target does not respond |
| | <code>--ca-cert [file]</code> , use the given CA certificate to validate the target's SSL |
| | <code>--skip-ssl-validation</code> , do not attempt to validate ssl certificate |
| <code>targets</code> | Display all targets |
| <code>context [name]</code> | Display or set current context |

| | |
|----------------------------------|---|
| contexts | Display all contexts |
| System Information | |
| info | get information about current target |
| me | get authenticated user information |
| prompts | Show prompts for credentials required for implicit grant post |
| signing key | get the UAA's token signing key(s) -c --client <name> -s --secret <secret>, client secret |
| stats | Show UAA's current usage statistics -c --client <name> -s --secret <secret>, client secret |
| password strength [password] | calculate strength score of a password |
| Tokens | |
| token get [credentials...] | Gets a token by posting user credentials with an implicit grant request -c --client <name> --scope <list> |
| token client get [id] | Gets a token with client credentials grant -s --secret <secret>, client secret --scope <list> |
| token owner get [client] [user] | Gets a token with a resource owner password grant -s --secret <secret>, client secret -p --password <password>, user password --scope <list> |
| token sso get [client] | Gets a token based on a one time passcode after successful SSO via browser -s --secret <secret>, client secret --passcode <passcode> --scope <list> |
| token refresh [refreshtoken] | Gets a new access token from a refresh token -c --client <name> -s --secret <secret>, client secret --scope <list> |
| token authcode get | Gets a token using the authcode flow with browser -c --client <name> -s --secret <secret>, client secret --scope <list> --[no-]cf, save token in the ~/.cf_tokens file --port <number>, pin internal server to specific port |
| token implicit get | Gets a token using the implicit flow with browser -c --client <name> --scope <list> --[no-]cf, save token in the ~/.cf_tokens file --port <number>, pin internal server to specific port |
| token decode [token] [tokentype] | Show token contents as parsed locally or by the UAA. Decodes locally unless --key <key>, Token validation key -c --client <name> -s --secret <secret>, client secret |
| token delete [contexts...] | Delete current or specified context tokens and settings --[no-]all, remove all contexts |
| User Accounts | |
| users [filter] | List user accounts -a --attributes <names>, output for each user --start <number>, start of output page --count <number>, max number per page |
| user get [name] | Get specific user account --origin <identity provider origin, defaults to UAA> -a --attributes <names>, output for each user |

| | |
|---|---|
| user add [name] | Add a user account --given_name <name> --family_name <name> --emails <addresses> --phones <phone_numbers> --origin <identity provider origin, defaults to UAA> -p --password <password>, user password |
| user update [name] | Update a user account with specified options --given_name <name> --family_name <name> --emails <addresses> --phones <phone_numbers> --origin <identity provider origin, defaults to UAA> --del_attrs <attr_names>, list of attributes to delete |
| user delete [name] | Delete user account --origin <identity provider origin, defaults to UAA> |
| user ids [username id...] | Gets user names and ids for the given users |
| user unlock [name] | Unlocks the user account |
| user deactivate [name] | Deactivates user |
| user activate [name] | Activates user |
| password set [name] | Set password -p --password <password>, user password |
| password change | Change password for authenticated user in current context -o --old_password <password>, current password -p --password <password>, user password |
| Groups | |
| groups [filter] | List groups -a --attributes <names>, output for each user --start <start>, show results starting at this index --count <count>, number of results to show |
| group get [name] | Get specific group information -a --attributes <names>, output for each user |
| group add [name] | Adds a group |
| group delete [name] | Delete group |
| group mappings | List all the mappings between uaa scopes and external groups --start <start>, show results starting at this index --count <count>, number of results to show |
| group map [external_group] | Map uaa groups to external groups --id <id>, map uaa group using group id --name <name>, map uaa scope using group name --origin <origin>, map uaa scope to external group for this origin. Default |
| group unmap [group_name] [external_group] | Unmaps an external group from a uaa group --origin <origin>, map uaa scope to external group for this origin. Default |
| member add [name] [users...] | add members to a group |
| member delete [name] [users...] | remove members from a group |
| Client Application Registrations | |
| clients [filter] | List client registrations -a --attributes <names>, output for each user --start <start>, show results starting at this index --count <count>, number of results to show |
| client get [id] | Get specific client registration -a --attributes <names>, output for each user |
| client add [id] | Add client registration --name <string> --scope <list> --authorized_grant_types <list> |

| | |
|-------------------------|--|
| | <pre> --authorities <list> --access_token_validity <seconds> --refresh_token_validity <seconds> --redirect_uri <list> --autoapprove <list> --signup_redirect_url <url> --clone <other>, get default settings from other -s --secret <secret>, client secret -i --[no-]interactive, interactively verify all values </pre> |
| client update [id] | Update client registration |
| | <pre> --name <string> --scope <list> --authorized_grant_types <list> --authorities <list> --access_token_validity <seconds> --refresh_token_validity <seconds> --redirect_uri <list> --autoapprove <list> --signup_redirect_url <url> --del_attrs <attr_names>, list of attributes to delete -i --[no-]interactive, interactively verify all values </pre> |
| client delete [id] | Delete client registration |
| secret set [id] | Set client secret |
| | <pre> -s --secret <secret>, client secret </pre> |
| secret change | Change secret for authenticated client in current context |
| | <pre> --old_secret <secret>, current secret -s --secret <secret>, client secret </pre> |
| CURL | |
| curl [path] | CURL to a UAA endpoint |
| | <pre> -X --request <method>, request method type, defaults to GET -d --data <data>, data included in request body -H --header <header>, header to be included in the request -k --insecure, makes request without verifying SSL certificates </pre> |
| Global options: | |
| -h --[no-]help | display helpful information |
| -v --[no-]version | show version |
| -d --[no-]debug | display debug information |
| -t --[no-]trace | display extra verbose debug information |
| --config [string file] | file to get/save configuration information or yaml string |
| -z --zone <subdomain> | subdomain of zone to manage |

1.5 More Resources

License: Code is licensed under MIT License.

<https://docs.cloudfoundry.org/credhub/>

<https://github.com/cloudfoundry/uaa>