SNAP JSON

SNAP is the name given to the protocol used to communicate with SiliconSafe's Password Protect managed authentication service. SNAP is not an acronym, the protocol is named after the card game Snap - when two passwords match SNAP!

The API accepts POST requests with JSON request bodies at URLs. E.g.

```
POST https://api.siliconsafe.com/api/user/authenticate
```

Every request to the API must be authenticated with a JSON Web Token (JWT). A unique JWT is issued to you when you sign-up to the service.

Include a header named "Authorization", with the value "Bearer *token*", replacing *token* with your JWT. E.g.

```
POST https://api.siliconsafe.com/api/...
Authorization: "Bearer yJjdHkiOiJKV1QiLCJlbmMiOiJBMjU2R0NNIG…"
Content-Type: application/json
{
    …
}
```

A SNAP message payload comprises JSON request bodies as follows:

Request type format

A typical request looks like this:

```
{
    "username": "user@example.com",
    "password": "V1QiLCJlbmMiOiJBM"
}
```

Response format

SNAP responds with a JSON response body.

A success response body looks like this:

```
{
    "success": true,
    "result": "CREDENTIALS_VALID"
}
```

A failure response body looks like this:

```
{
    "success": false,
    "result": "INVALID_REQUEST",
    "errors": [ "password.empty" ]
}
```

success	boolean	Indicates whether a positive result was received from Password Protect, indicating the user was successfully created/updated/deleted/authenticated. If false, Password Protect rejected the request.
result	string	A result code that indicates a successful request, or the reason for failure.
errors	array<string>	If our service rejected the request because fields in the request body were not supplied or in the incorrect format, and the request was not forwarded to Password Protect, validation errors will be listed.

SNAP JSON requests

Create user

POST https://api.siliconsafe.com/api/user/create

```
{
    "username": "user@example.com",
    "password": "V1QiLCJlbmMiOiJBM"
}
```

Standard results

USER_CREATED: The new user was successfully stored in Password Protect.
USER_EXISTS: A user with the given username already exists.

Authenticate user

POST https://api.siliconsafe.com/api/user/authenticate

```
{
    "username": "user@example.com",
    "password": "V1QiLCJlbmMiOiJBM"
}
```

```
}
```

Standard results

CREDENTIALS_VALID: The credentials are valid.
CREDENTIALS_INVALID: The username or password was invalid.
ACCOUNT_SUSPENDED: The credentials are valid but the account is suspended.

Update user password

```
POST https://api.siliconsafe.com/api/user/update

{
    "username": "user@example.com",
    "oldPassword": "V1QiLCJlbmMiOiJBM",
    "newPassword": "P02Jmk2H39GHEbbz1"
}
```

Standard results

USER_UPDATED: The new password was stored.
USERNAME_NOT_FOUND: The username does not exist.
PASSWORD_INVALID: The old password is incorrect.
ACCOUNT_SUSPENDED: The credentials are valid but the account is suspended.

Reset user password

```
POST https://api.siliconsafe.com/api/user/reset

{
    "username": "user@example.com",
    "newPassword": "P02Jmk2H39GHEbbz1"
}
```

Standard results

USER_RESET: The new password was stored.
USERNAME_NOT_FOUND: The username does not exist.
ACCOUNT_SUSPENDED: The credentials are valid but the account is suspended.

Delete user

```
POST https://api.siliconsafe.com/api/user/delete

{
```

```
    "username": "user@example.com"
}
```

Standard results

USER_DELETED: The user was deleted.
USERNAME_NOT_FOUND: The username does not exist.
ACCOUNT_SUSPENDED: The credentials are valid but the account is suspended.

Suspend user

```
POST https://api.siliconsafe.com/api/user/suspend
```

```
{
    "username": "user@example.com"
}
```

Standard results

USER_SUSPENDED: The user was suspended.
USERNAME_NOT_FOUND: The username does not exist.