



Filter by keyword



Encrypting stored data

ON THIS PAGE

[storage.setSecret](#)[storage.getSecret](#)[storage.deleteSecret](#)

The Key-Value Store also lets you store key-value pairs in a secure, encrypted manner. Use the methods listed here to store sensitive data. Data stored through these methods can't be queried through the [query](#) method.

To start, import the Forge API package in your app, as follows:

```
1 import { storage } from '@forge/api';
```

Each installation of your app is subject to the Storage API's quotas and limits. See [Storage quotas](#) and [Storage limits](#) for more details.



The app storage API requires your app to have the `storage:app` scope. See [Add scopes to call an Atlassian REST API](#) guide to add new scopes to your app.

storage.setSecret

Similar to `storage.set`, `storage.setSecret` provides a way to store sensitive credentials. Values set with `storage.setSecret` can only be accessed with `storage.getSecret`.

The same limitation is applied: persisted JSON values may be up to 128 KiB in size and have a key of up to 500 bytes.

You do not need to identify your app or the active site/installation in your key, as Forge will do this automatically.

Write conflicts are resolved using a last-write-wins strategy.

Method signature

```
1 storage.setSecret(key: string, value: array | boolean | number | ob
```

Example

Sets the key `example-key` to one of the supported value types.

```
1 // array
2 storage.setSecret('example-key', [ 'Hello', 'World' ]);
3
4 // boolean
5 storage.setSecret('example-key', true);
6
7 // number
8 storage.setSecret('example-key', 123);
9
10 // object
11 storage.setSecret('example-key', { hello: 'world' });
12
13 // string
14 storage.setSecret('example-key', 'Hello world');
```

storage.getSecret

Gets a value by key, which was stored using `storage.setSecret`. If the key doesn't exist, the API returns `undefined`.

Method signature

```
1 storage.getSecret(key: string): Promise<array | boolean | number |
```

Example

Gets the secret value associated with the key `example-key`.

```
1 // Read the value for key `example-key`
2 await storage.getSecret('example-key');
```

storage.deleteSecret

Deletes a secret value by key, this succeeds whether the key exists or not.

-  Write conflicts are resolved using a last-write-wins strategy.

Method signature

```
1 storage.deleteSecret(key: string): Promise<void>;  
2
```

Example

Deletes the value associated with the key `example-key`, if it hasn't already been deleted.

```
1 // Delete the value with the key `key`  
2 await storage.deleteSecret('example-key');
```

Rate this page:     



[Changelog](#)

[System status](#)

[Privacy](#)

[Your Privacy Choices](#)

[Developer Terms](#)

[Trademark](#)

[© 2025 Atlassian](#)