



Web



PSPDFKit Server



WEB &gt; GUIDES &gt; PSPDFKIT SERVER

# Manage secrets easily with PSPDFKit Server



PSPDFKit Server has been deprecated and replaced by [Document Engine](#). To migrate to Document Engine and unlock advanced document processing capabilities, refer to our [migration guide](#). Learn more about these enhancements on our [blog](#).

PSPDFKit Server offers the following options for authentication:

- ✧ Dashboard password — Used to access the [PSPDFKit Dashboard](#).
- ✧ Secret keybase — Used for generating secret keys for the purposes of authentication.
- ✧ Public key for authenticating JWT payloads — Used for [client authentication](#).

Collectively, these options are referred to as secrets.

## Change in behavior

Previously, you could configure secrets using environmental variables. However, this approach had the following disadvantages:

- ✧ You couldn't change secrets during runtime.
- ✧ You couldn't use more than one valid secret.
- ✧ You needed to restart PSPDFKit Server.



ASK AI

PSPDFKit Server 2023.1.0 introduced a new mechanism for managing secrets. There's always a "current" version of a secret that never expires. You can only change this current version using secret rotation. Next to the current version, you can have other secrets, but the non-current secrets need to have an expiration date.

By default, current secrets are sourced from the following environmental variables to retain old behavior:

- ✧ JWT\_PUBLIC\_KEY
- ✧ SECRET\_KEY\_BASE
- ✧ DASHBOARD\_PASSWORD

To change this behavior, set `REPLACE_SECRETS_FROM_ENV` to `false`. In this case, current secrets are only sourced from environmental variables if the secrets aren't yet set via the API.

`REPLACE_SECRETS_FROM_ENV` must be set to `false` to enable secret rotation. Otherwise, secrets set via the API are replaced on every restart.

Secrets can be managed via the PSPDFKit Server API. The secret APIs expect the secret `:type`, which can be `jwt`, `dashboard_password`, or `secret_key_base`.

## Adding a new secret

To add a new secret, use the following request:

```
1 POST /api/secrets/:type
2 Content-Type: application/json
3 Authorization: Token token="<secret token>"
4
5 {
6   "secret": "my new secret string",
7   "expiresAt": "2023-01-01T12:00:00.000000Z"
8 }
```

The `secret` field is the new secret. `expiresAt` sets the expiration date for the secret. The date and time are in UTC in ISO 8601 format.

## Listing secrets

To list the IDs and expiry dates of existing secrets, use the following request:

```
1 GET /api/secrets/:type
2 Authorization: Token token="<secret token>"
```



An example response is the following:

```
1 [
2   {
3     "id": 1,
4     "expiresAt": null
5   },
6   {
7     "id": 2,
8     "expiresAt": "2023-01-01T12:00:00.000000Z"
9   },
10  ...
11 ]
```



Only non-expired secrets are returned. For the current secret, the `expiresAt` field is `null`.

## Rotating secrets

Current secrets don't expire. To change current secrets, rotate them using the request below:

```
1 POST /api/secrets/:type/rotate
2 Content-Type: application/json
3 Authorization: Token token="<secret token>"
4
5 {
6   "secret": "my new secret string"
7 }
```



## Expiring existing secrets

To change the expiration date of secrets other than the current one, use the request below:

```
1 PATCH /api/secrets/:type/:id
2 Content-Type: application/json
```



```
3 Authorization: Token token="<secret token>"
4
5 {
6   "expiresAt": "2023-01-01T12:00:00.000000Z"
7 }
```

---

Was this helpful?

✓ YES

✗ NO

---

Questions? [Contact us](#)

