SKEDULO CLI

Cheatsheet

The Skedulo CLI is a modern, powerful tool designed to streamline workflows, enable source-code driven development, and simplify building, testing, and deploying custom solutions on the Skedulo platform.

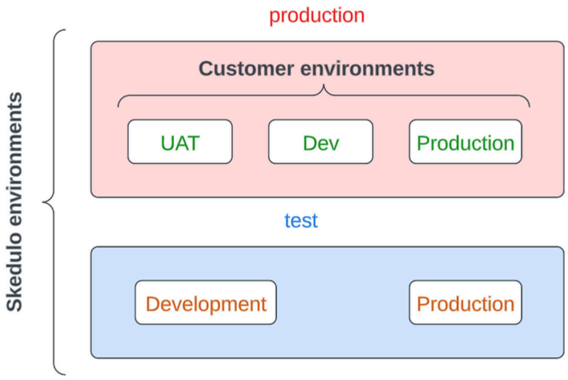
Installation

- Installation Guide
- About: **$ sked about**
- Available versions: **$ sked update --available**
- Upgrade latest version: **$ sked update**

Feature Flags

- **enablePackagingV2**: must be enabled to use CLI and Packaging
- **enableAlphaArtifacts**: must be enabled to deploy Alpha artifacts
- **enableBetaArtifacts**: enabled by default
- **enableMobileExtensionPackagingAdapter**: must be enabled to use Mobile Extension artifact.

Environments



The term environment varies by context.

In most commands, the environment flag refers to a Skedulo environment like production, test, perf, or staging, defaulting to production if unspecified.

Customers may have multiple environments or tenants, all within Skedulo's production environment, so the environment flag can often be ignored in the CLI.

Internal Plugin

- Install: **$ sked plugins install @skedulo/plugin-internal**
- Login: **$ sked internal login [-e <test/production>]**
- More Commands: **$ sked internal --help**

MORE INFO Skedulo CLI Internal Plugin

Tenant Commands

$ sked tenant login web -a <alias> -t <tenant-name> [-e <environment>] [--set-default]
Log in to Skedulo using the web interface

$ sked tenant login access-token -a <alias> -t <tenant> [-e <environment>] [--set-default]
Log in to Skedulo using a access token

$ sked tenant display [-a <alias>] [--permissions] [--roles]
Display authentication information for a given tenant

$ sked tenant logout [-a <alias>]
Removes the token for a given alias or default alias if not provided

$ sked tenant list
Display currently stored tenants

$ sked tenant set-default -a <alias>
Set default alias

Internal SDLC (Using Internal Token)

Use this command for local package deployment

$ sked internal register-package -s <github-reference-url>
Register a Skedulo package.

$ sked internal deploy-package -p package-name -t <team-name> [-v <version>]
Deploy a registered Skedulo package to one tenant.

$ sked internal deploy-package -p <package-name> -f <team-list-json-or-csv>
Deploy a registered Skedulo package to multiple tenants.

MORE INFO Creating a package as a Skedulo employee

Customer SDLC (Using Tenant Token)

$ sked package deploy local -p <package-dir> -a <alias>
Deploy a local package. Cannot deploy to deploy to a Production environment.

$ sked package register -s <github-reference-url>
Register a customer package. The Github token can be provided via command prompt or can be set via the GITHUB_ACCESS_TOKEN environment variable.

$ sked package deploy registered -p <package-name> [-v <version>]
Deploy a registered customer package. Use latest version if not provided.

MORE INFO Skedulo CLI SDLC for customers

SKEDULO CLI

Cheatsheet

What does a valid GitHub reference URL look like?

https://github.com/Skedulo/new-package/tree/main

<repository-base-ur>/tree/<branch-name|commit-hash>

https://github.com/Skedulo/new-package/tree/4ae834dc213ace7dcf307a6

⚠ The branch name must not contain a forward slash. If it does, use the commit hash instead.

You can [Install a package using the Admin Console]()

The API user must be configured to install a package on a Salesforce tenant.

Function Development

$ sked function generate -n <function-name> -o <output-dir>

Generate a function template

$ sked function dev [-a <alias>] [-p <port>] [-f <env-file>]

Run a function on local. Eg: sked function dev . -p 8080 (using default alias and .env file for configuration variables)

Test local function: the ***sked-api-server*** header is required authentication if the function calls other functions or Skedulo APIs.

```
$ curl -X GET "http://localhost:8080/ping" \
    -H "Authorization: Bearer <token>" \
    -H "sked-api-server: https://api.skedulo.com"
```

[Develope and Deploy Functions]()

Web Extension Development

$ sked web-extension generate-ssl-cert

Generate a self-sign SSL certificate

$ sked web-extension dev <root-dir>

Run a web extension on local.

View the local web extension at https://<team domain>/c-dev

You can find the template and instructions to run your first web extension [here]()

Common Errors