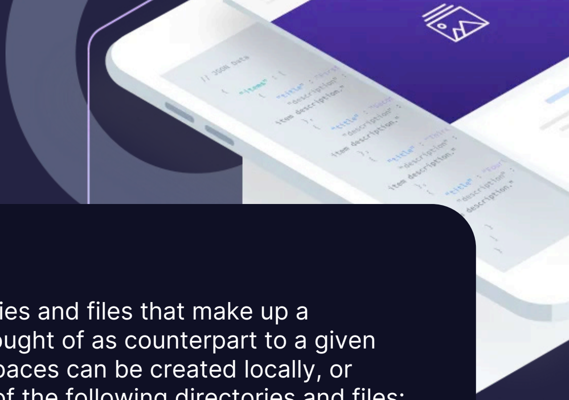
dotCMS CLI Cheat Sheet



Installation

```
npm install -g @dotcms/dotcli
```

See dotcms.com/docs/latest/cli for manual JAR download instructions.

Instance Configuration

Run the following to set the initial configuration required by all commands.

```
dotcli config
```

This will open a wizard for you to configure your instance.

```
Enter the key/name that will serve to identify
the dotCMS instance (must be unique) [local].
The name is [local]
Enter the dotCMS base URL (must be a valid URL
starting protocol http or https) [http://local-
host:8080]
The URL is [http://localhost:8080]
Are these values OK? (Enter to confirm or N to
cancel)  (Y/n)
```

Learn More

Visit the following pages on dotcms.com and GitHub to learn more:

dotCMS docs	dotcms.com/docs/latest/cli
GitHub docs	github.com/dotcms/core/tree/master/tools/dotcms-cli

Workspaces

The dotCMS CLI uses a set of directories and files that make up a workspace. The workspace can be thought of as counterpart to a given instance, managed via the CLI. Workspaces can be created locally, or remotely via GitHub repo and consist of the following directories and files:

`/content-types/`	Directory, stores Content Types.
`/files/`	Directory, stores file assets.
`/languages/`	Directory, stores Languages.
`/sites/`	Directory, stores sites.
`.dot-workspace.yml`	File, CLI workspace marker; indicates the root directory as a valid workspace.

Global Options

These options are available at all times within the CLI, regardless of command or subcommand.

-h, --help Provides a detailed explanation of a given command or subcommand.

```
dotcli login --help
dotcli site find -h
```

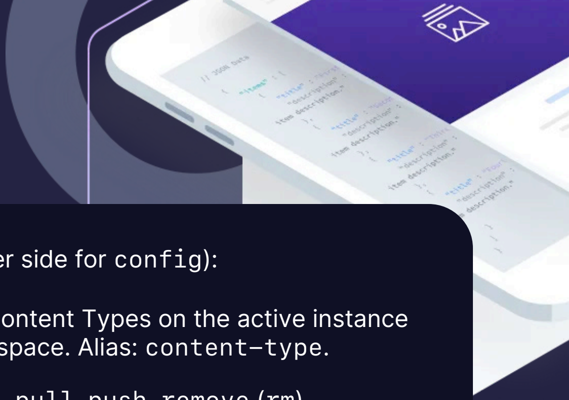
-V, --version With a Version flag in place, the CLI will simply print its current version information and exit.

```
dotcli -V
```

-e, --errors When a command invokes the Error option, it will return a stack trace detailing any error it encounters during execution.

```
dotcli status -e
```

dotCMS CLI Cheat Sheet



Main Commands

The dotCMS CLI currently has ten commands that can serve as its first argument (see other side for `config`):

`status` Prints current active dotCMS instance, current signed-in user, and API URL.

```
dotcli status
```

`instance` Prints a list of available dotCMS instances, as defined during instance configuration.

```
dotcli instance -act demo
```

`login` Signs in to a dotCMS instance.

```
dotcli login -u admin@dotcms.com -p
dotcli login -tk <dotcms-token>
```

`push` The push command is global, fully synchronizing a target instance with a given workspace—all Sites, Content Types, Languages, and Files.

As an optional argument, push takes a path to a local workspace, directory, or file; if no path argument is specified, it will default to the current working directory.

```
dotcli push ~/test-workspace --dry-run --retry-
attempts 3
```

`pull` The pull command is global, fully synchronizing a specified workspace with the active instance—all Sites, Content Types, Languages, and Files. Each item is saved in a descriptor file, either JSON or YAML.

```
dotcli pull --workspace ~/test-workspace
```

`ct` Performs actions on Content Types on the active instance and/or specified workspace. Alias: `content-type`.

Sub-commands: `find`, `pull`, `push`, `remove` (`rm`)

```
dotcli content-type find -n Blog
dotcli ct pull --format YAML
dotcli ct rm BlogAuthor
```

`site` Performs actions on Sites on the active instance and/or specified workspace. Alias: `host`.

Sub-commands: `find`, `pull`, `push`, `create`, `remove` (`rm`), `copy` (`cp`), `start`, `stop`, `archive`, `unarchive`

```
dotcli site find -n Home
dotcli host cp -in Site1 -cn Site1Copy --all
```

`lang` Performs actions on Languages on the active instance and/or specified workspace. Alias: `language`.

Sub-commands: `find`, `pull`, `push`, `remove` (`rm`)

```
dotcli language remove tlh-us
dotcli lang pull -h
```

`files` Performs actions on file assets in the active instance and/or specified workspace.

Sub-commands: `tree`, `ls`, `pull`, `push`

```
dotcli files pull --excludeAsset Rumpel*kin -p
dotcli files tree //demo.dotcms.com/application
```