

CLI Reference

The PLANQK Command Line Interface (CLI) lets you interact with the PLANQK Platform directly from your terminal. We have installation instructions to guide you through the initial setup in our [quickstart](#) guide.

Installation

To install the PLANQK CLI, you must install Node.js 18 or higher and the npm command line interface using either a [Node version manager](#) or a [Node installer](#) .

Then install the PLANQK CLI globally using npm:

```
npm install -g @planqk/planqk-cli
```

bash

IMPORTANT

Make sure you have uninstalled the deprecated CLI:

```
npm uninstall -g @anaqor/planqk
```

bash

You can use the `--help` flag to get information about the supported commands:

```
planqk --help
```

bash

You may also get information about a specific command:

```
planqk <command> --help
```

bash

Authentication

Please copy your [personal access token](#) to your clipboard.

Login to your account using your access token:

bash

```
planqk login -t <your access token>
```

Usage

```
$ npm install -g @planqk/planqk-cli
$ planqk COMMAND
running command...
$ planqk (--version)
@planqk/planqk-cli/2.10.5 darwin-arm64 node-v20.14.0
$ planqk --help [COMMAND]
USAGE
  $ planqk COMMAND
...
```

Commands

- [planqk autocomplete \[SHELL\]](#).
- [planqk create-openapi](#)
- [planqk get-context](#)
- [planqk init](#)
- [planqk list-contexts](#)
- [planqk login](#)
- [planqk logout](#)
- [planqk run \[SERVICEID\]](#).
- [planqk serve](#)
- [planqk services](#)
- [planqk set-context \[CONTEXTID\]](#).
- [planqk up](#)

planqk autocomplete [SHELL]

display autocomplete installation instructions

USAGE

```
$ planqk autocomplete [SHELL] [-r]
```

ARGUMENTS

SHELL shell type

FLAGS

-r, --refresh-cache Refresh cache (ignores displaying instructions)

DESCRIPTION

display autocomplete installation instructions

EXAMPLES

```
$ planqk autocomplete
```

```
$ planqk autocomplete bash
```

```
$ planqk autocomplete zsh
```

```
$ planqk autocomplete --refresh-cache
```

See code: [@oclif/plugin-autocomplete](https://github.com/oclif/plugin-autocomplete)

planqk create-openapi

Creates the OpenAPI definition based on the sample data in the "input" directory. The generated OpenAPI definition is saved in the file "openapi-spec.yml". An existing "openapi-spec.yml" file is saved as "openapi-spec-saved.yml" before it is overwritten. After the generation you may check the generated OpenAPI definition and correct it manually if necessary.

USAGE

```
$ planqk create-openapi [--wd <value>]
```

FLAGS

`--wd=<value>` Working dir (e.g., for tests)

DESCRIPTION

Creates the OpenAPI definition based on the sample data in the "input" directory. saved in the file "openapi-spec.yml". An existing "openapi-spec.yml" file is saved it is overwritten. After the generation you may check the generated OpenAPI definit necessary.

EXAMPLES

```
$ planqk create-openapi
```

See code: [dist/commands/create-openapi/index.ts](#)

planqk get-context

Get the current context, i.e., the personal or organization account you are currently working with.

USAGE

```
$ planqk get-context
```

DESCRIPTION

Get the current context, i.e., the personal or organization account you are curren

EXAMPLES

```
$ planqk get-context
```



See code: [dist/commands/get-context/index.ts](#)

planqk init

Initialize a PLANQK project to create a service.

USAGE

```
$ planqk init [--non-interactive] [--name <value>]
```

FLAGS

```
--name=<value>      The name of the service
--non-interactive  Run it in non-interactive mode
```

DESCRIPTION

Initialize a PLANQK project to create a service.

EXAMPLES

```
$ planqk init
```

See code: [dist/commands/init/index.ts](#)

planqk list-contexts

Retrieves the available contexts, i.e., the personal or organizational accounts available to you to work with.

USAGE

```
$ planqk list-contexts
```

DESCRIPTION

Retrieves the available contexts, i.e., the personal or organizational accounts av

EXAMPLES

```
$ planqk list-contexts
```



See code: [dist/commands/list-contexts/index.ts](#)

planqk login

Login with your PLANQK Platform credentials

USAGE

```
$ planqk login [-t <value>]
```

FLAGS

-t, --token=<value> Your personal access token

DESCRIPTION

Login with your PLANQK Platform credentials

EXAMPLES

```
$ planqk login -t <personal access token>
```

See code: [dist/commands/login/index.ts](https://docs.planqk.de/dist/commands/login/index.ts)

planqk logout

Logout of the PLANQK Platform

USAGE

```
$ planqk logout
```

DESCRIPTION

Logout of the PLANQK Platform

EXAMPLES

```
$ planqk logout
```

See code: [dist/commands/logout/index.ts](https://docs.planqk.de/dist/commands/logout/index.ts)

planqk run [SERVICEID]

Creates a job execution of a PLANQK Service

USAGE

```
$ planqk run [SERVICEID] [-d <value>] [-p <value>] [--data-file <value>] [--params  
  [--detached]
```

FLAGS

-d, --data=<value> Input data as JSON string.
-p, --params=<value> Parameters as JSON string.
--data-file=<value> Relative path to file containing input data.
--detached Executes the job in detached mode, i.e., without waiting fo
--params-file=<value> Relative path to file containing params.

DESCRIPTION

Creates a job execution of a PLANQK Service

EXAMPLES

```
$ planqk run
```

```
$ planqk run --detached
```

```
$ planqk run -d '{"values": [10,12]}' -p '{"round_up": true}'
```

```
$ planqk run --data-file=./input/data.json --params-file=./input/params.json
```

See code: [dist/commands/run/index.ts](https://github.com/planqk/planqk/blob/master/dist/commands/run/index.ts)

planqk serve

Runs your current service code in a containerized environment to expose it through a local web server, similarly to how the PlanQK Platform would run your code. The local web server exposes the same RESTful HTTP endpoints to start a service execution, to check the status of running executions, to cancel executions, and to retrieve execution results.

USAGE

```
$ planqk serve [-p <value>]
```

FLAGS

-p, --port=<value> The port on which the local web server accepts requests

DESCRIPTION

Runs your current service code in a containerized environment to expose it through how the PLANQK Platform would run your code. The local web server exposes the same

service execution, to check the status of running executions, to cancel executions

EXAMPLES

```
$ planqk serve -p <port>
```

See code: [dist/commands/serve/index.ts](#)

planqk services

List all available services of the current selected context.

USAGE

```
$ planqk services
```

DESCRIPTION

List all available services of the current selected context.

EXAMPLES

```
$ planqk services
```

See code: [dist/commands/services/index.ts](#)

planqk set-context [CONTEXTID]

Set the current context, i.e., the personal or organization account you are currently working with.

USAGE

```
$ planqk set-context [CONTEXTID]
```

DESCRIPTION

Set the current context, i.e., the personal or organization account you are curren

EXAMPLES

```
$ planqk set-context
```



```
$ planqk set-context <context-id>
```

See code: [dist/commands/set-context/index.ts](#)

planqk up

Creates or updates a PLANQK Service

USAGE

```
$ planqk up [--silent]
```

FLAGS

`--silent` Suppresses all outputs, helpful when executed in a CI/CD pipeline.

DESCRIPTION

Creates or updates a PLANQK Service

EXAMPLES

```
$ planqk up
```

See code: [dist/commands/up/index.ts](#)

Help us improve this page!

Last updated: 22.01.25, 23:16

Previous page

[Service Access from Python](#)

Next page

[planqk.json Reference](#)