

Prottern Documentation

- Introduction
 - Installation
 - First steps
 - How Prottern Works
- Commands
 - signup
 - login
 - logout
 - save
 - repository
 - describe
 - generate
 - delete
 - explore
 - get
 - pub
 - unpub
 - profile

Introduction

- Installation
- First steps
- How Prottern Works

Installation

Prottern ships only a single executable file. You can install it using the installers below, or download the release binary from the [releases page](#).

Chocolatey (Windows):

Build and install from source using **Cargo**:

First steps

Creating an User Account

After checking if the package is installed you need to create an user account

```
$ prottern signup
```

A signup form will appear:

```
Username: pultz-lucas
Email: prottern@email.com
Password: *****
Confirm your password: *****
```

Saving templates to the repository

In the beginning there will be no template saved in the repository, so let's see now how to save a template.

First I need to have real files and folders structured to use it as reference.

Below is my example structure:

```
ProjectWeb
├── src
│   ├── style.css
│   ├── script.js
│   └── index.html
```

My example is simple but larger structures of files and folders can be saved in the repository.

To save we will use the save command:

```
$ prottern save ProjectWeb project-web
```

if you look into your local repository the "project-web" will be there.

```
$ prottern repo
```

output:

```
+-----+-----+-----+-----+
Ã¢â€š name      Ã¢â€š owner  Ã¢â€š template_type Ã¢â€š      created_at      Ã¢â€š
+-----+-----+-----+-----+
Ã¢â€š project-web Ã¢â€š not-user Ã¢â€š Remote    Ã¢â€š 2021-07-29 21:44:19 Ã¢â€š
+-----+-----+-----+-----+
```

Generating a project from template

This process is very simple.

Having the folder where you want to generate the template, which in my case is **Website**, use the command **generate** or **gen** to instantiate the template:

```
$ prottern gen project-web Website
```

If generated successfully, now you have a project identical to the other.

```
Website
├── src
│   ├── style.css
│   ├── script.js
│   └── index.html
```

Ready! Now you know how to save templates and generate it with Prottern. :D

Let's use it in projects!

How prottern Works

Prottern basically manipulates templates created by the user.

templates are **metadata** to generate files, directories and execute console commands.

The user can save a template from an directory and utilise that to generate projects from scratch or a piece of an existing projects.

Commands

- signup
- login
- logout
- save
- repository
- describe
- generate
- delete
- explore
- get
- pub
- unpub
- profile

Signup

This command is used for signup an user account.

You need to have an user account because you will need one before save templates or publish templates.

To create an user account type:

```
$ prottern signup
```

A signup form will appear requesting your username, password and email.

Login

This command is used for authenticate your user account.

To authenticate your account type:

```
$ prottern login
```

A login form will appear requesting your username and password.

Logout

The **logout** command is used to exit of the current user account

```
$ prottern logout
```

output:

```
logged out.
```

Repository

Alias: repo

The **repository** command lists all the templates in your local repository.

For example:

- I have three templates saved in my repository, with the names of: project-web, project-mobile, project-desktop.
- When typing the templates command, the three templates will appear in my terminal.

```
$ prottern repository
```

All templates will appear in a pretty table.

Save

The **save** command saves a template in the repository.

It takes two parameters:

The 1st parameter receives the path of the project's root directory that will be used as a reference for generating the template.

The 2nd parameter receives the name of the template.

```
$ prottern save <directory-path> <template-name>
```

Generate

Alias: gen

The **generate** command generates a project from a saved template. It takes two parameters:

- The 1st parameter receives the name of an existing template on the repository
- The 2nd parameter receives the path of the output directory where the template will be generated.

```
$ prottern generate <template-name> <directory-path>
```

Describe

Alias: desc

The **describe** command shows information about the template passed as a parameter.

```
$ prottern describe <template-name>
```

output example:

```
index.html
src
src\script.js
src\style.css
```

Delete

Alias: del

The command **delete** deletes a template from the repository.

It receives the name of the template to be removed as a parameter.

```
$ prottern delete <template-name>
```

Explore

The **explore** command shows all public templates available to use on the platform.

```
$ prottern explore
```

output:

list of all platform's templates...

Get

This command is used to get public third party templates.

```
$ prottern get <template-name>
```

Pub

This command is used to publish a template.

```
$ prottern pub <template-name>
```

Profile

This command returns the public info of the current user account logged.

```
$ prottern profile
```

output example:

```
Name: username  
Email: username@email.com
```

Unpub

This command is used to unpublish a remote template.

```
$ prottern unpub <template-name>
```

Documentation

Alias: docs

This command will open the Prottern Documentation on github website.

```
$ prottern documentation
```

Version

Alias: v

This command will return the current Prottern version installed in your computer.

```
$ prottern version
```

Help

Alias: h

This command will return some info about commands available with a basic description, current Prottern version and the url of official Prottern Github repository for more information.

```
$ prottern help
```