Prottern Documentation

Introduction

Installation

First steps

Creating an User Account

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

```
$ prottern register
```

A signup form will appear:

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

Creating Repository

After creating an user account you need to create a repository that will be used to store templates.

```
$ prottern init
```

This command will create the template repository.

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 a real project to use as a reference.

Below is my example project:

```
ProjectWeb
   Ã‹Âª src
      Ã‹Âª style.css
      Ã‹Âª script.js
   Ã‹Âª index.html
```

My example project is simple but larger projects 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:

```
>> Local Templates
   | project-web

>> Remote Templates
```

Creating a project from template

Creating a project from template is very simple.

Having the root folder where will be created the project, which in my case will be **Website**, use the command **create** to instantiate the template:

```
$ prottern create project-web Website
```

If created successfully, a success message will appear and you will have a project identical to the other.

Ready! You can now save templates and create projects with Prottern.

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

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>
```

create

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

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

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

---

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>
```