

About

kickoff is a Command-Line tool, which can be used to easily create boilerplate code for new projects and other things based on archetypes stored in GitHub repos.

You can do the following things with kickoff:

1. Install kickoff in global mode.
2. Create the repository on github for the project template, including mustache templates.
3. Place the `.kickoff.yml` file, to define the questionnaire for collecting parameter values for template files.

In order to learn more about how to install and use the tool, read the [Get Started](#) page, then continue with reading the [documentation](#).

Get Started

Installation

Prerequisites:

kickoff needs [Node.js](#) and [NPM](#) installed on the machine, before start installing and using it.

Installation steps

The kickoff tool can be installed as any other node module, but you have to install to the global node_modules folder, so use the -g switch to npm.

To install kickoff, execute the following command:

```
$ npm install -g kickoff
```

To check, whether kickoff is installed successfully, run the following command:

```
$ kickoff -V
```

Usage of the kickoff utility

Check if kickoff is properly installed:

```
$ kickoff -h
```

```
Usage: kickoff [options]
```

```
Options:
```

-h, --help	output usage information
-V, --version	output the version number
-v, --verbose	Verbose mode
-a, --data-file <path>	Path to the yaml format datafile
-f, --folder <path>	Path to the source directory
-s, --src <username/repository>	Git repository
-d, --dest <dst>	Destination folder

Create your first project from a repository

The following command will create a new command line tool project with all the required boilerplate code, unsing the [ncli-archetype](#) repository:

```
$ kickoff -s tombenke/nccli-archetype -d new_app
? The name of the application: new_app
? The full name of the author of the application: Tamás Benke
? The email address of the author: tombenke@gmail.com
? The github account (username) of the author: tombenke
```

Check the results in the newly created new_app directory, and see also how the parameters requested through the console have substituted the mustache parameters in README.md and package.json files.

In order to learn more about how to use the tool, read the [documentation](#) pages.

Documentation

Installation

The prerequisites and installation of the tool is described in the [Get Started](#) page.

Create the template repository

This is a normal repository, which can contain anything, including mustache templates. These files will be copied to the destination folder without modification. During this transfer, each file is processed by the mustache template engine, so you can place parameters into these files, which will be replaced by actual values, that you can define through the command line interface of the tool.

The enquiry for these parameters can be described in the `.kickoff.yml` file, that you can place into the root of the repository.

© 2017 Tamás Benke. All rights reserved. MIT License. [View License](#)

See [ncli-archetype](#) repository on GitHub, that is a Project archetype for Command Line tools using Node.js.

The .kickoff.yml file

You can describe the questionnaire for template parameters within the .kickoff.yml file.

This is a sample configuration file for the kickoff utility:

```
description: |
  This is a project archetype for node.js based, cli applications.
  New projects can be generated from this archetype by the
  [kickoff](https://github.com/tombenke/kickoff) utility.

welcome: >

  You are generating a JavaScript project, which is
  using node.js, and running as a command-line application.

postscript: >

  Next steps:

  - Install the npm modules required by the newly generated application:

    cd <dest-folder>
    npm install

questions:
-
  type: input
  name: appname
  message: The name of the application
  default: anonymous
-
  type: input
  name: author_full_name
  message: The full name of the author of the application
  default: Anonymous
-
  type: input
  name: author_email
  message: The email address of the author
  default: anonymous@some-domain.com
-
  type: input
  name: author_github_account
  message: The github account (username) of the author
  default: anonymous

converters:
  mustache:
    - docs/about.md
    - docs/getHelp.md
    - docs/getStarted.md
    - docs/pageTemplate.html
    - index.html
    - package.json
    - README.md
```

You can add as many parameters as you need, using the same names you put into the mustache templates.

The description property

In the description property of the .kickoff.yml you can write down what the whole archetype projects is about. This field holds information mainly for those people who creates and/or maintain the archetype project itself.

The welcome and postscript messages

The welcome string will be written to the console, when the kickoff utility starts, in order to inform the user, what will happen during the process.

At the end, before exit, the kickoff will write the postscript string to the console, to inform the user about the succesful execution. Also you can place here reminders for things to do after kickoff run.

Questions

This section contains the list of questions and parameters the kickoff will collect during its working. The tool sets the variables to their default values that you can overwrite during the questioning, then the given parameters will be used by the tool to fill the templates during the generation of the target content.

File-name templates

The fileName is an optional list of from/to pairs, where from is a valid path relative to the base of the projects, and defines the name of the file which should be renamed during the processing. The to property defines the new name. It can be either a literal string value or a mustache template. The template values in the to filename will be substituted via the context object like in case of the template files in the project.

The next fragment demonstrates how you can define such rename operation using both constant values and file name templates:

```
fileNames:
```

```
from: docs/pageTemplate.html
to: docs/{{appname}}-page-template.html
```

```
from: README.md
to: ReadMe.md
```

Converters

By default the kickoff simply copies the files from the source to the target, but you can use converters to fill your data into the files that have been collected via questions. These converters can be templates or JavaScript functions. You can define which file can be taken as a converter and what type of conversion does it do. Currently the following converters can be defined:

converters: mustache: // Here you can define those files that are mustache templates, // for example in case of ncli-archetype these templates are used:

```
- docs/about.md
- docs/getHelp.md
- docs/getStarted.md
- docs/pageTemplate.html
- index.html
- package.json
- README.md
handlebars:
  // In case you need a more sophisticated template engine,
  // you can use the Handlebars engine
plugin:
  // Here you can refer to plain JavaScript modules which are exporting a single function,
  // that consumes a String and returns with a result String. The function also got the context
  // data that was defined via the questionnaire, or via the dedicated data file with the '-a' which.
```

In the test/src folder you can find samples to each kind of converter.

NOTE: The plugin type of converters currently work only in cases when you generating from local sources.

Load the context data from YAML file instead of querying

Let's suppose, you create a new_app executing the following command:

```
$ kickoff -s tombenke/ncli-archetype -d new_app
? The name of the application: new_app
? The full name of the author of the application: TamÅ¡s Benke
? The email address of the author: tombenke@gmail.com
? The github account (username) of the author: tombenke
```

You can achieve the same result, if you create a YAML file (for example: desc.yml), with the following content:

```
appname: new_app
author_full_name: TamÅ¡s Benke
author_email: tombenke@gmail.com
author_github_account: tombenke
```

With the -a, --data-file argument, you can load the context data in the YAML format file instead of using the interactive inquiry via the console, so execute the next command:

```
$ kickoff -a ncli.yml -s tombenke/ncli-archetype -d new_app
```

The result will be something similar to this:

You are generating a JavaScript project, which is using node.js, and running as a command-line application.

Loading data from ncli.yml instead of querying...

Next steps:

- Install the npm modules required by the newly generated application:

```
cd <dest-folder>
npm install
```

This method can be used, if the volume of the context data is big, and/or it has a complex structure (nested arrays and objects, etc.).

Get Help

Find answers and ask questions on the following StackOverflow tag <http://stackoverflow.com/questions/tagged/kickoff> or email me anytime at tombenke@gmail.com if you run into problems or have feedback.