Table of Contents

WRAPP

1. WHAT IS IT ?!

A generic swiss knife wanna be bash / perl centric application wrapper for quicky packaging and deploying your tools, create new tools out of your existing ones , generating code for additonal functions, search and replace in both file paths and contents ... and all the rest not mentioned actions in the sfw/bash/wrapp/tests/all-wrapp-tests.lst file ...

2. INSTALLATION AND CONFIGURATION

The wrapp instances and clones have been running on *Nix boxes with bash > 3.0 … , but some actions require 4.0 and older …
Cygwin has been tested also …

2.1. Prerequisites

The must have binaries are:
bash, perl, zip
The nice to have are:
tmux, vim ,ctags
The examples are for Ubuntu - use you OS package manager …

```
apt-get autoclean
apt-get install --only-upgrade bash

sudo apt-get install -y perl

# optionally
sudo apt-get install -y excuberant-ctags
sudo apt-get install -y 7z

apt-get upgrade
```

2.2. Fetch the source

Fetch the source from git hub as follows:

```
# create your product dir:
mkdir -p /opt/csitea/wrapp
cd /opt/csitea/wrapp/

# fetch the source
git clone git@github.com:YordanGeorgiev/wrapp.git

# DO NOT CD into the new dir !!!!
```

2.3. Build the first wrapp instance

Build the wrapp instance by running the bootstrap script

```
# bootstrap the product instance dir
```

```
bash wrapp/src/bash/wrapp/bootstrap-wrapp.sh

# the script should prompt you to
cd /opt/csitea/wrapp/wrapp.1.1.5.dev.$USER
```

2.4. Check the runnable actions

You could check the functions which could be run - aka "actions" by issuing the following command.

```
# check the runnable with the -a cmd arg actions
find . -name '*.func.sh' | sort
```

2.5. Start hacking

Start hacking … or wait check at least the test call running all the functions of the tool …

```
# opionally if you are in the vim camp open the "project relative files list file"
vim meta/.dev.wrapp

# Ctrl + Z ,
bash sfw/bash/wrapp/test-wrapp.sh

# now clone your own instance
bash sfw/bash/wrapp/wrapp.sh -a to-app=my-tool
```