

Table of Contents

Table of Contents	1
ISSUE-TRACKER	2
1. WHAT IS IT ?!	2
2. INSTALLATION AND CONFIGURATION	2
2.1. Prerequisites	2
2.2. Fetch the source	2
2.3. run the boot-strap script	2
2.4. Apply the db and issue create scirpts	2
2.5. Install the required Perl modules	3
2.6. Start hacking	3
3. ADDITIONAL DOCS	3
4. AND FINALLY THE PROJECT STATUS	3

ISSUE-TRACKER

1. WHAT IS IT ?!

A tool to manage multiple projects' issues programmatically using txt files, xls files, store them into Postgres db and publish them into Google Sheets.

2. INSTALLATION AND CONFIGURATION

2.1. Prerequisites

The must have binaries are:

bash, perl, zip, postgres 9.6

The nice to have are:

tmux, vim, ctags

The examples are for Ubuntu - use your OS package manager ...

If you do not have postgres than you would have to follow the longer installation instructions :

<https://github.com/YordanGeorgiev/issue-tracker/blob/master/doc/md/issue-tracker-devops-guide.md#1-installations-and-configurations>

```
# use your OS package manager ... if you are not on Ubuntu

sudo apt-get autoclean
sudo 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

sudo apt-get upgrade
```

2.2. Fetch the source

Fetch the source from git hub as follows:

```
# got to a dir you have write permissions , for example:
cd ~

# fetch the source
git clone git@github.com:YordanGeorgiev/issue-tracker.git
```

2.3. run the boot-strap script

The bootstrap script will interpolate change the git deployment dir to a "product_instance_dir" (your instance of the issue-tracker, having the same version as this one, but running on a different host with different owner - your)

```
# define the latest and greates product_version
export product_version=$(cd issue-tracker;git tag|sort -nr| head -n 1;cd ..)

# run the bootstrap script :
bash issue-tracker/src/bash/issue-tracker/bootstrap-issue-tracker.sh

# now go to your product instance dir , note it is a DEV environment
cd /opt/csitema/issue-tracker/issue-tracker.$product_version.dev.$USER
```

2.4. Apply the db and issue create scirpts

If you do not have the PostgreSQL (v9.5 >) with current Linux user configured role installed check the instructions in the installations and configuratios section of the DevOps guide:

<https://github.com/YordanGeorgiev/issue-tracker/blob/master/doc/md/issue-tracker-devops-guide.md#1-installations-and-configurations>

If you do have it , apply the db and issue create scirpts as follows:

```
# apply the postgre sql scripts
bash src/bash/issue-tracker/issue-tracker.sh -a run-pgsql-scripts
```

2.5. Install the required Perl modules

Just run the prerequisites checker script which will provide you with copy pastable instructions

```
sudo perl src/perl/issue_tracker/script/issue_tracker_preq_checker.pl

# after installing all the modules check the perl syntax of the whole project:
bash src/bash/issue-tracker/issue-tracker.sh -a check-perl-syntax
```

2.6. Start hacking

Start usage:

```
doParseIniEnvVars cnf/issue-tracker-issues.dev.doc-pub-host.cnf

bash src/bash/issue-tracker/issue-tracker.sh -a txt-to-db
bash src/bash/issue-tracker/issue-tracker.sh -a db-to-xls

# now edit the files in the xls
bash src/bash/issue-tracker/issue-tracker.sh -a xls-to-db
bash src/bash/issue-tracker/issue-tracker.sh -a db-to-txt
export issues_order_by_attribute=start_time
export issues_order_by_attribute=prio
bash src/bash/issue-tracker/issue-tracker.sh -a db-to-txt

# publish to gsheet issues:
export do_truncate_tables=1 ; bash src/bash/issue-tracker/issue-tracker.sh -a db-to-gsheet -t
daily_issues,weekly_issues,monthly_issues,yearly_issues
```

3. ADDITIONAL DOCS

Additional docs could be found in the doc/md dir.

4. AND FINALLY THE PROJECT STATUS

The issue tracker project status could be tracked by the issue-tracker data stored in the following gsheet:

<https://docs.google.com/spreadsheets/d/e/2PACX-1vQ3ijqJkY03mDXiaT3mcvr96NkgRnsONSAHyBGwnukuRezhHTaAZsUxOcoQ6fHfZmcHXP2KpD6kfCPR/pubhtml>