

Genboree

Installation instruction

Prerequisites

- Linux 64-bit OS only
- Kernel version $\geq 2.6.32$
- Glibc version ≥ 2.12
- Libstdc++ version $\geq 6.0.13$ (g++ 4.4.7)
- Bash
- Access to root account – needed for installation and starting/stopping Genboree services
- Directory `/usr/local/brl` must be available (it is created during installation)
- User “genboree” cannot be used (it is created during installation)
- The following TCP ports must be available:
 - 80 (HTTP)
 - 16001
 - 16002
- SELinux is not supported – must be disabled
- environment-modules package cannot be installed on the host system
- server must be able to reach itself by public address (it is not obvious for some firewall configuration) – sometimes extra rule must be added to IP table, e.g.:
`iptables -t nat -A OUTPUT -d 111.222.33.44 -j DNAT --to-destination 127.0.0.1`
- it is good to set max number of open file to threshold larger than default value 1024 (for CentOS 7 just edit the file `/etc/security/limits.conf` and add lines `'* soft nofile 65535'` and `'* hard nofile 65535'`)

Genboree installation

All commands below must be run from root account.

1. Run `install.sh` script from the main package's directory.
2. Go to `/usr/local/brl/local/etc/init.d`.
3. Run script `genboree` with parameter `start`. Usually you have to wait 2-3 minutes after this step before Genboree will be ready to use.
4. Start internet browser and open page on `localhost`. Front webpage of Genboree should appear.
5. You can log into Workbench as `genbadmin` with password `genbadmin`.

6. You can log into Redmine as *admin* with password *admin*.
7. Edit `/usr/local/brl/data/conf/settings.yaml` file to set basing settings for Genboree. You have to adjust this file to make Genboree visible from other hosts (by default, it works on *localhost* only). Genboree must be stopped and started again to make
8. You can stop Genboree by running `genboree` script from `/usr/local/brl/local/etc/init.d` with parameter *stop*.

Redmine configuration

1. Start internet browser and open page on *localhost/redmine*. Main Redmine page should appear.
2. Sign in as *admin* using password *admin*.
3. Go to “My account” → “Change password” and set new password for *admin* account (click “Save” then).
4. Go to “Administration” → “Settings” → “Email notification”, set “Emission email address” and “Email subject prefix”, click “Save” and check e-mail configuration by clicking “Sent a test e-mail”. (The e-mail from the previous step is used!).
5. Go to “Administration” → “Plugins” and click “Configure” link at the row entitled “Redmine Issue Checklist plugin”. Then activate “Save changes to issue log” option and click “Save”.

Google Calendar plugin Configuration (optional)

Go to Administration => Plugins => Google Calendar Plugin => Configure

1. Select and save the project custom field which will contain the Google Calendar-iframe code; in this example select the newly created project custom field "Google Calendar"
2. To add a Google Calendar tab to an project, you need to go get the IFrame based embed code from Google.
3. The calendar you want to embed MUST BE PUBLICLY viewable, else there will be some login stuff required.
4. While logged in to Google, go get the IFrame stuff from Google:
 - 4.1. Click on "Settings" in the upper left hand corner at Google.
 - 4.2. Click on the "Calendars" tab
 - 4.3. Select the calendar you want to use
 - 4.4. In the "Embed This Calendar" section, copy the entire `<tt><iframe></tt>` code from the text box

GenboreeKB configuration

Login to genboree account by running from root's shell:

```
su genboree
```

Create GenboreeKB database using command `createDbForGenboreKB.rb`, e.g.:

```
In createDbForGenboreKB.rb genbadmin genbadmin kbTest
```

It takes three parameters: name of genboree user (login), name of genboree group in which a database should be created and unique name of new GenboreKB database. When the command succeeds, a new genboree database will appear in given genboree group with name created from concatenation of “KB:” and given GenboreeKB name.

In every GenboreeKB database different collections can be defined. To define a collection a special

json file with collection's schema is needed. For now collections can be defined from console only. To create a new collection the script createCollection.rb must be used, e.g.:

```
createCollection.rb genbadmin kbTest collTest collSchema.json
```

The script takes four parameters: name of genboree group containing GenboreeKB database, name of the GenboreeKB database, name for new collection in given database and path to json file with schema.

To access the collection from user interface a Redmine project with GenboreeKB plugin is needed. Follow the steps above to create this kind of project:

1. login to Redmine as admin user
2. Create new project, it must contain the “Genboree kbs” plugin (check proper checkbox)
3. Connect created project with GenboreeKB database by filling fields in “GenboreeKB” tab in project's settings. Available fields should be filled with the following pattern:
 1. KB Name: name of the GenboreeKB (e.g.: kbTest)
 2. Description: can be leaved blank
 3. Genboree Group: genboree group containing KB database, e.g.: genbadmin
 4. Genboree Host: localhost
 5. Click “Create” to link the project with given KB database.
4. To use the GenboreeKB plugin you have to log as genboree user who has access to the KB database (e.g.: genbadmin). Before leaving the admin account make sure that this genboree user have access to created Redmine project. Proper access rights may be set in “Members” tab in project's settings.