



Taskerville

Technical Documentation

Overview

About this project.....	1
License.....	1
Project lists.....	2
The backend.....	2
The frontend.....	2
Important url's.....	2
Portlets.....	3
Social activity.....	3
Manageability.....	3
Projects.....	3

About this project

Taskerville is a task management solution. Written using JEE technologies, such as

- EJB 3.1
- JPA 2.0
- JSF 2.0
- Portlet 2.0
- Apache [Maven](#) 2.x (switching to Maven 3 as soon as it is stable enough and runs under Hudson).
- [Hudson](#) (as CI)
- [Java](#) 1.6

License

Taskerville is considered to be written using the [GPL](#) 3.0 GNU General Public License.

Project lists

Taskerville is split into several sub projects.

Name	Description	Source location
Core	The backend core functionality.	/backend/
Frontend	The frontend webapp(s) to work with Taskerville in a browser, or remotely.	/frontend/
Bridges	The bridges to work with other task management solutions (as long as they support a remote interface). The bridges are for synchronizing taskerville tasks and projects with other task management solutions.	/bridges/
Connectors	The connectors, or remote facade to work with Taskerville remotely. This project depends on the tsk core and gives external interfaces (SOAP, Hessian, REST) to taskerville.	/remoting/
Deployment	This project contains all artifacts to get a deployable version of Taskerville.	/deployment/
I18N	All known translations for Taskerville. It is a plain project consisting of a single artifact, mostly containing property files.	/i18n/

The backend

The backend will be hosted under the backend project.

The frontend

The frontend is hosted under the frontend project.

Important url's

The main root mount point for the webapp will be

/tsk

Under this mount point, there will be several important views, such as

/tsk/p/PROJECTID

where PROJECTID will be the short string id of a project in Taskerville. In the above case, the project homepage will be displayed, listing all open versions, followed by all closed (released) versions, and also giving a list of all open and closed issues.

There are more important views:


- /tsk/p/project/issues = will list all issues about a project
- /tsk/p/project/version/issues = will list all issues regarding a given version
- /tsk/p/project/version/issues?solution=open

- `/tsk/p/project/version/issues?state=new`
- `/tsk/p/project/id` = see the next view
- `/tsk/t/TSK-1` = is a mirror for the above statement; lists a single task with the given task id (consisting of the project id TSK and the internal project task number 1)
- `/tsk/dashboard` = shows the dashboard of the current user
- `/tsk/dashboard/username` = shows the dashboard of the given user

Portlets

The frontend will use portlets. Portlets are considered a way to plug in functionality. Meaning, you can create a portlet which shows the tasks of a specific user. In another portlet, you can view the tasks of another group or user by changing the context of the portlet.

Still to define: which portlet container will be used. Considering:

- LifeRay 
- Apache Pluto
- JBoss Portlet

Social activity

Taskerville will use [Apache Shindig](#) to give some basic activity streams and social aspect to the platform. However, main focus is the task management.

Manageability

Projects

Taskerville can host a number of projects. Any project consists of a project owner (a single person) and can have a number of specific groups or members working on this project.

