

# Timesheet user manual

- Installation
  - System Setup (Quick Start)
    - Data Structure
    - Project Setup
    - User Management
  - Time Entries Submit
  - Approval process
  - System Setup (Advanced Settings)
    - Holidays & Workdays
    - Task Type & Activity type
      - Task Type
      - Activity Type
    - Tag Types and Tags
    - Postgresql Settings

## Installation

1. Make sure you have Java properly installed. The PATH system variable should include a path to java executable files (java.exe or java). In order to check that java is installed properly, run `java -version` in the command line. If java version appears, then move to the next sage.
2. Download zip distributive.
3. Unzip the content to a target folder.
4. Run `start-lori.bat` (Windows) or `start-lori.sh` (Linux or MacOS).

Once all the steps above are successfully accomplished, the system will launch and available at <http://localhost:8080/app>. HSQL database is used by default. In order to change the database, follow the steps described in the [Postgresql settings](#) paragraph.

## System Setup (Quick Start)

### Data Structure

A minimal knowledge of the data structure is required to to get started with LoriTimesheets.

There are 5 main entities involved in the time tracking process:

- Clients; each client can be associated with several projects.
- Projects; each project is linked to a client and contain a number of tasks.
- Tasks; each task is linked to a project and is a subject to a time spent entry.
- Project participants; each project is associated with a number of project participants, who can add time spent entries for the defined project's tasks.
- Time entries; a time entry shows how much time was spent on a certain task, for a certain project by a certain user on a certain date.

Project participants can be granted one of the following roles:

- Manager can modify projects, tasks, participants, approve submitted timesheets and submit timesheets.
- Approver can approve and submit timesheets.
- Worker is permitted to submit timesheets.

Observer has a read only access to the system.

## Project Setup

To start using LoriTimesheets you need to specify the projects your team is working on. To do this, follow the steps below:

1. Define client
  - a. Open Menu > Project Management > Clients
  - b. Click Create
  - c. Input client name, then click Ok
2. Define project
  - a. Open Menu > Project Management > Projects

- b. Click Create project button
- c. Input project name, select client and click Ok
3. Define tasks
  - a. Select the project in projects table and click Create task
  - b. Input task name and click Ok
4. Define a project manager
  - a. Click Assign button and select the Manager role
  - b. On the dialog window select your system user and click Select. Congratulations, now you are a project manager!

Now you are able to track the time spent on the project.

The screenshot shows the 'Projects' interface. At the top, there is a search bar and a filter section. Below that, there are buttons for 'Create project', 'Edit', and 'Remove'. An 'Assign' dropdown menu is open, showing options: 'Worker' (highlighted), 'Manager', 'Approver', and 'Observer'. To the right, there is a 'Tasks' section with a 'Create task' button, 'Edit', 'Remove', and 'Close' buttons. A table below shows a single row with columns: Name, Code, Type, Status, Default tags, and Required tag types. The 'Name' is 'Platform' and 'Code' is 'PLATFORM'. The 'Status' is 'Open'. Red arrows point from text labels to the 'Create project' button, the 'Worker' option in the 'Assign' menu, and the 'Create task' button.

## User Management

The out-of-the-box database contains only one user (admin/admin). Make sure you change login/password of the default user before you go to production.

To add more users to the system, you'll need to follow the steps below:

1. Open Menu > Administration > Users
2. Click Create
3. Fill in the required user details
4. Specify security roles for the user: Manager, Approver, Worker or Observer (If user is a manager in one project and a worker in another - add both roles)
5. If you want users to change passwords on the first login - click Additional > Reset password.
6. If you want users to receive new temporary passwords by email - choose both Generate new passwords and Send emails with generated passwords
7. If no one option is selected, then users will be able to login with the specified password once the first login happens, after that they will be required to change the password.
8. Now you can assign users to the projects

The screenshot shows the 'Users' interface. At the top, there is a search bar and a filter section. Below that, there are buttons for 'Create user', 'Edit', 'Copy', 'Remove', 'Bulk edit', 'Additional', and 'Excel'. A table below shows a single row with columns: Login, Name, Position, Group, Email, Change Password at Next Logon, and Work Hours For Week. The 'Login' is 'admin', 'Name' is 'Administrator', 'Position' is 'Company', and 'Work Hours For Week' is '0'. A context menu is open over the 'admin' row, showing options: 'Copy settings', 'Change password', 'Reset passwords' (highlighted), and 'Reset 'remember me' tokens'. Red arrows point from text labels to the 'Create user' button and the 'Reset passwords' option in the context menu.

User x

Users > User

|                      |                                       |                     |                                                    |
|----------------------|---------------------------------------|---------------------|----------------------------------------------------|
| Login                | <input type="text" value="new_user"/> | Group               | <input type="text" value="Company"/>               |
| New Password         | <input type="password"/>              | Position            | <input type="text"/>                               |
| Confirm New Password | <input type="password"/>              | Language            | <input type="text" value="English"/>               |
| First Name           | <input type="text" value="Doe"/>      | Time Zone           | <input type="text"/> <input type="checkbox"/> Auto |
| Middle Name          | <input type="text"/>                  | Permitted IP Mask   | <input type="text"/>                               |
| Last Name            | <input type="text" value="John"/>     | Work Hours For Week | <input type="text" value="40"/>                    |
| Name                 | <input type="text" value="John D."/>  | Active              | <input checked="" type="checkbox"/>                |
| Email                | <input type="text"/>                  |                     |                                                    |

Roles

**Add security roles**

| Role Name | Localized Name |
|-----------|----------------|
| Base role | Базовая роль   |
| Worker    | Сотрудник      |

Substituted Users

| Login | Name | Start Date | End Date |
|-------|------|------------|----------|
|       |      |            |          |

## Time Entries Submit

You can submit timesheets in different ways.

### 1. Weekly timesheets screen.

Click + button. Specify project and task in the dropdown lists. Record the time spent for specific days (8:30, 8h30m, 8 30, 8). Then click Submit All.

By default the screen shows the current week. If you need to submit timesheets for another week use < and > or Calendar (📅) buttons on the top-right corner.

Weekly Timesheets x

**Add timesheet**

E.g.: @PLAT #DEV \$TAG 2h30m or @PLAT #DEV \$TAG 2:30 (@Project #Task \$Tag XhYm or X:Y). Ctrl+Space - suggestions, Ctrl+Shift+A - apply

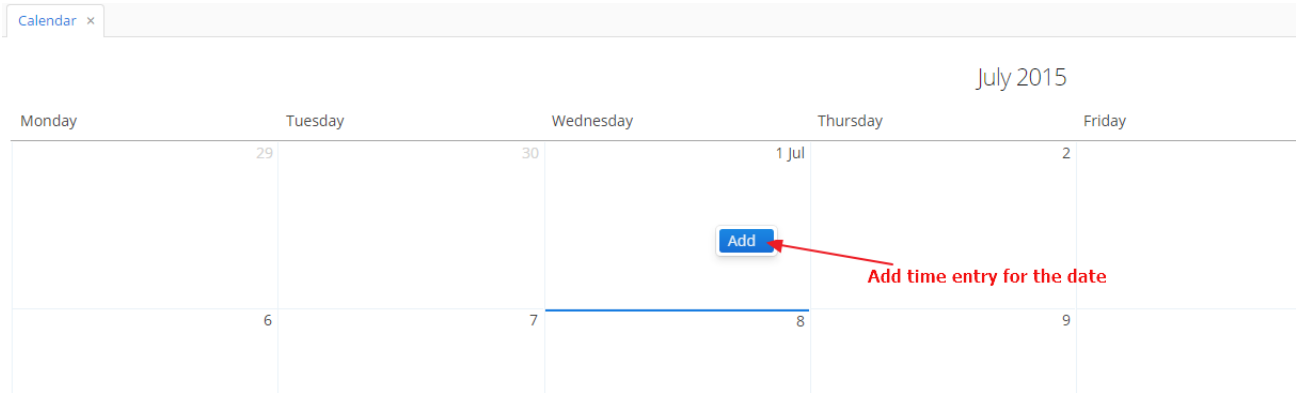
06/07/2015 - 12/07/2015  < Today > 📅

| Project                               | Task                                     | Mon 6  | Tue 7  | Wed 8  | Thu 9  | Fri 10 | Sat 11 | Sun 12 | Total  |
|---------------------------------------|------------------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|
|                                       |                                          | 0h 00m | 0h 00m | 0h 00m | 0h 00m | 0h 00m | 0h 00m | 0h 00m | 0h 00m |
| <input type="text" value="Platform"/> | <input type="text" value="Development"/> | 8 30   | 8:35   | 8h40m  |        |        |        |        | 0h 00m |

**Select project** **Select task** **Input time** **Submit timesheets**

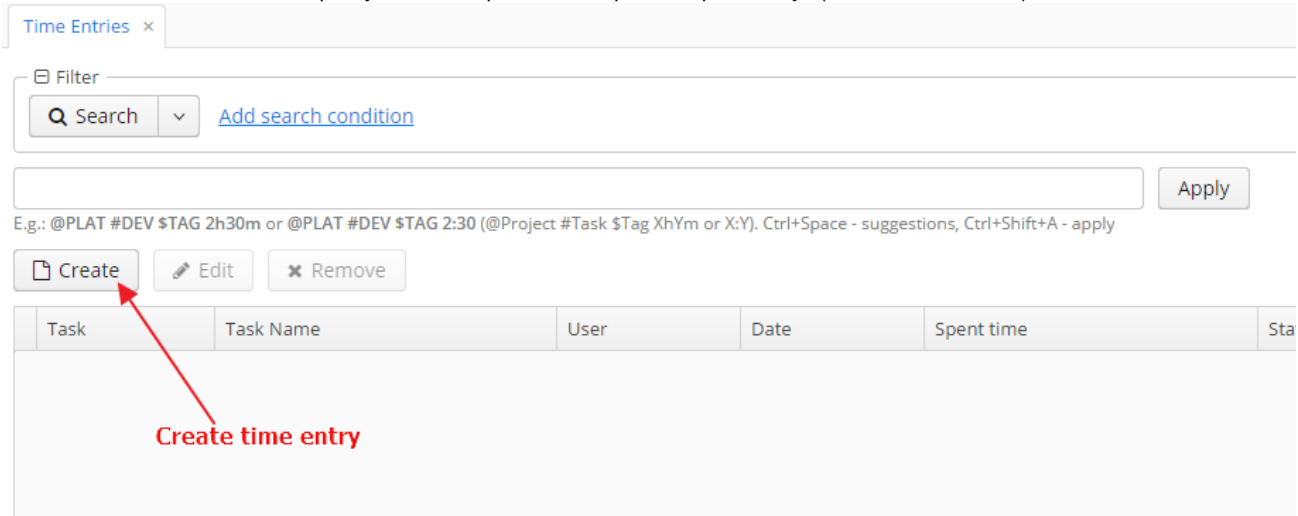
### 2. Calendar screen.

Right click on the date you want to submit timesheets to. Choose Add menu. Select task and input the time spent on specific days (8:30, 8h30m, 8 30, 8). Click Ok.



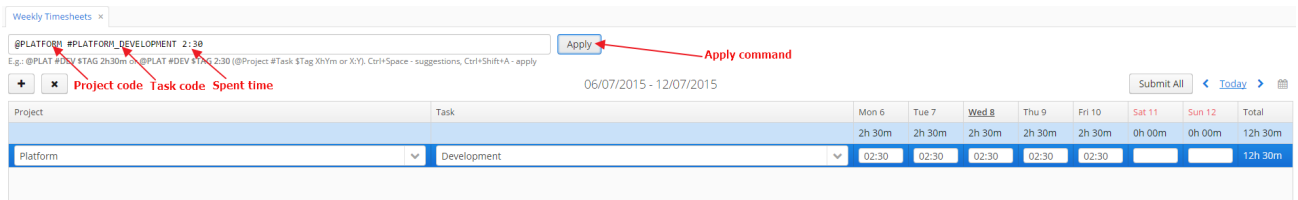
3. My Time Entries screen.

Click Create button. Select task, specify date and input the time spent on specific days (8:30, 8h30m, 8 30, 8). Click Ok.



4. Command line.

As you might have already noticed, all the submission screens have a command line. The command line enables a quick spent time submission. Note that the command line is hidden in the Calendar screen by default. You can show it by clicking the Gear (⚙️) button. To use the command line type the @ symbol. The command line will prompt the available projects you are assigned with. Then type #, the command line will prompt all the tasks you can submit timesheets to. Input time spent on tasks in 8h30m or 8:30 format. Then click Apply (or use Ctrl+Shift+A).



Full command line syntax is: @ProjectCode #TaskCode \*ActivityTypeCode \$TagCode !DatesOrDaysOfWeek SpentTime "Description"  
 Examples:

- @Platform #Core \*Development \$6573 !1-15,18 4h "Bugfix for PL-6573"
- @Platform #Misc \*Meeting !Mon 1h "Usual meeting"
- @Platform #Core 8h

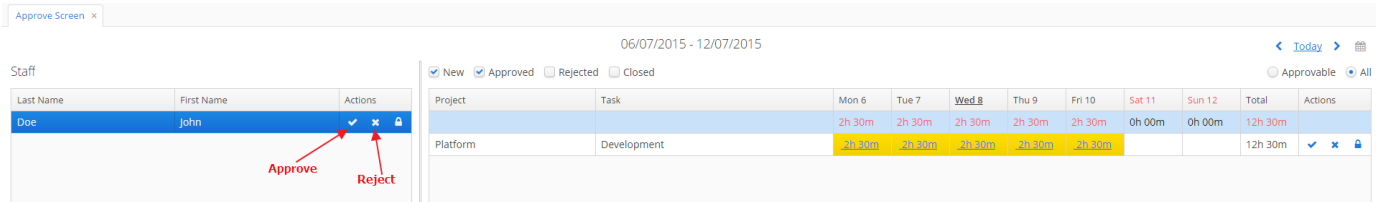
## Approval process

Often timesheets are used to calculate the salary of employees. Obviously, before the wage calculation is submitted, working time should be verified. In terms of LoriTimesheets the process of verifying time spent on tasks is called Timesheet Approval.

Only users with Manager or Approver role can undertake Approval process.

The Approval screen shows all the participants' spent time submissions related to your projects. Here you can approve or reject time submission. To approve the timesheet click V, X - to reject.

Rejected timesheets can be edited by the owner and then approved or rejected again.

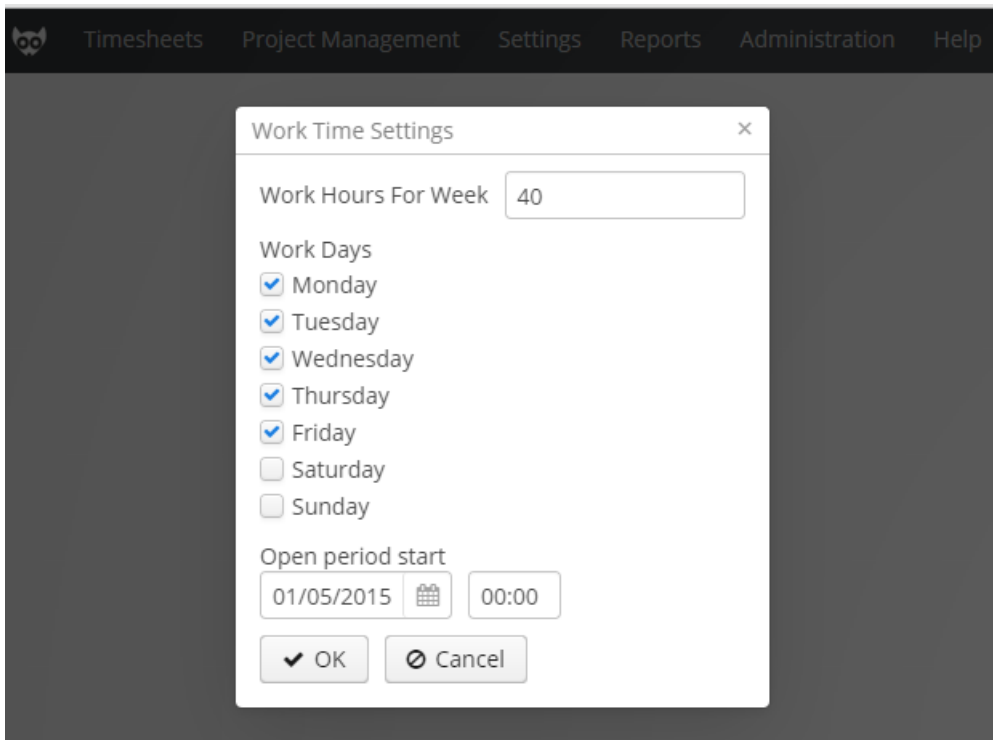


## System Setup (Advanced Settings)

### Holidays & Workdays

Different companies have different rules for weekly workhours and workdays.

You can set the rules in Menu > Settings > Work Time settings:



Under certain circumstances, you might need to disable the ability to submit historical timesheets. LoriTimesheets allows you to set open period start date. The result is that the System will not allow the submission of timesheets for the dates before open period start.

Also, you can define official bank holidays in Menu > Settings > Holidays.

All these settings do not restrict spent time submission, but are used to highlight unexpected situations in Calendar and Weekly timesheets screen s.

Weekly Timesheets x

Apply

E.g.: @PLAT #DEV \$TAG 2h30m or @PLAT #DEV \$TAG 2:30 (@Project #Task \$Tag XhYm or X:Y). Ctrl+Space - suggestions, Ctrl+Shift+A - apply

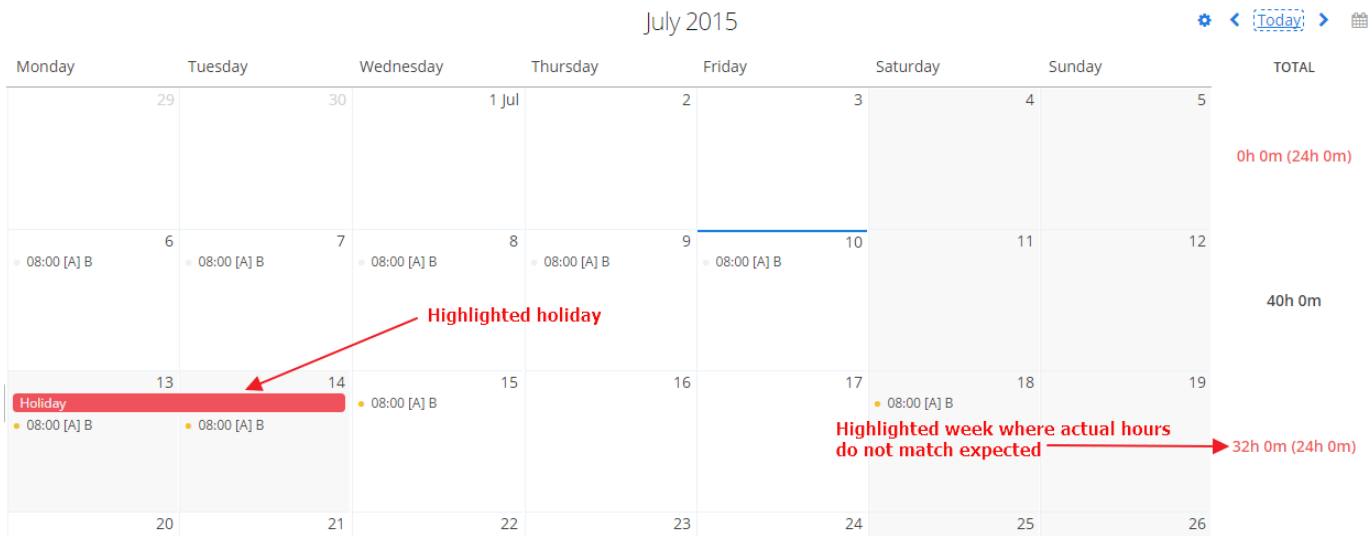
13/07/2015 - 19/07/2015

Submit All < Today >

| Project | Task | Mon 13  | Tue 14  | Wed 15  | Thu 16 | Fri 17 | Sat 18  | Sun 19 | Total   |
|---------|------|---------|---------|---------|--------|--------|---------|--------|---------|
|         |      | 8h 00m  | 8h 00m  | 8h 00m  | 0h 00m | 0h 00m | 8h 00m  | 0h 00m | 32h 00m |
| A       | B    | 08:00 * | 08:00 * | 08:00 * |        |        | 08:00 * |        | 32h 00m |

Highlighted time entry submitted to holiday

Highlighted time entry submitted to weekend



## Task Type & Activity type

### Task Type

Optionally, you can group tasks by assigning them to Task Types. For example, tasks can be roughly divided by Mobile App, Server Side, Web UI, Middleware and Misc task types. Assigning tasks to task types enables management to perform an analysis of which Task Type is the most labour-consuming.

You can setup Tasks Types in Menu > Settings > Task types screen. Note that task types are system wide and can be used across all projects.

### Activity Type

Some companies might require the classification of spent time by activity type. For instance, there is the Development task within the Mobile App task type. However, Development can be divided into particular activities, such as coding, bugfixing, refactoring, debug, underlying technology upgrade, autotests development, documentation, etc.

If such a drill down approach is required, it is possible to specify Activity Types and assign them to time entries.

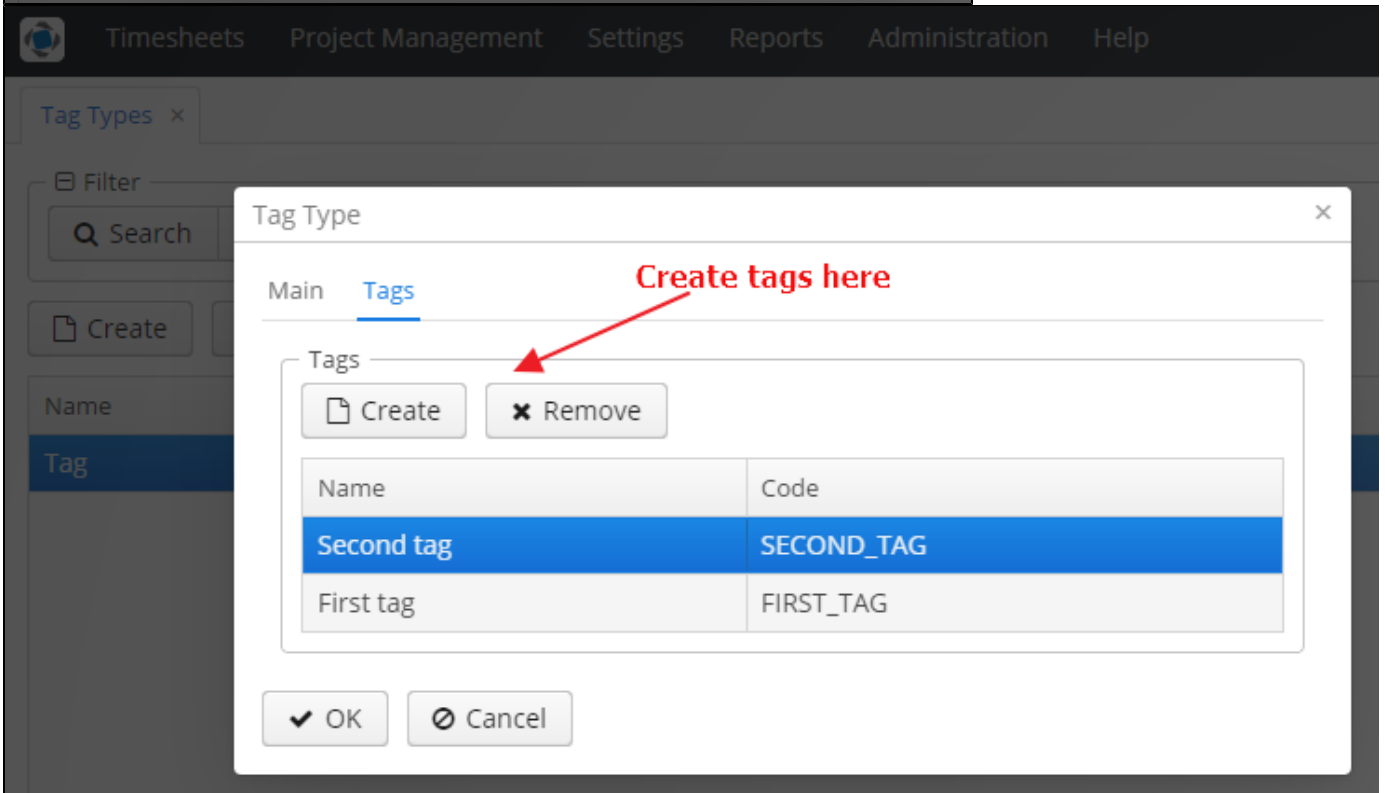
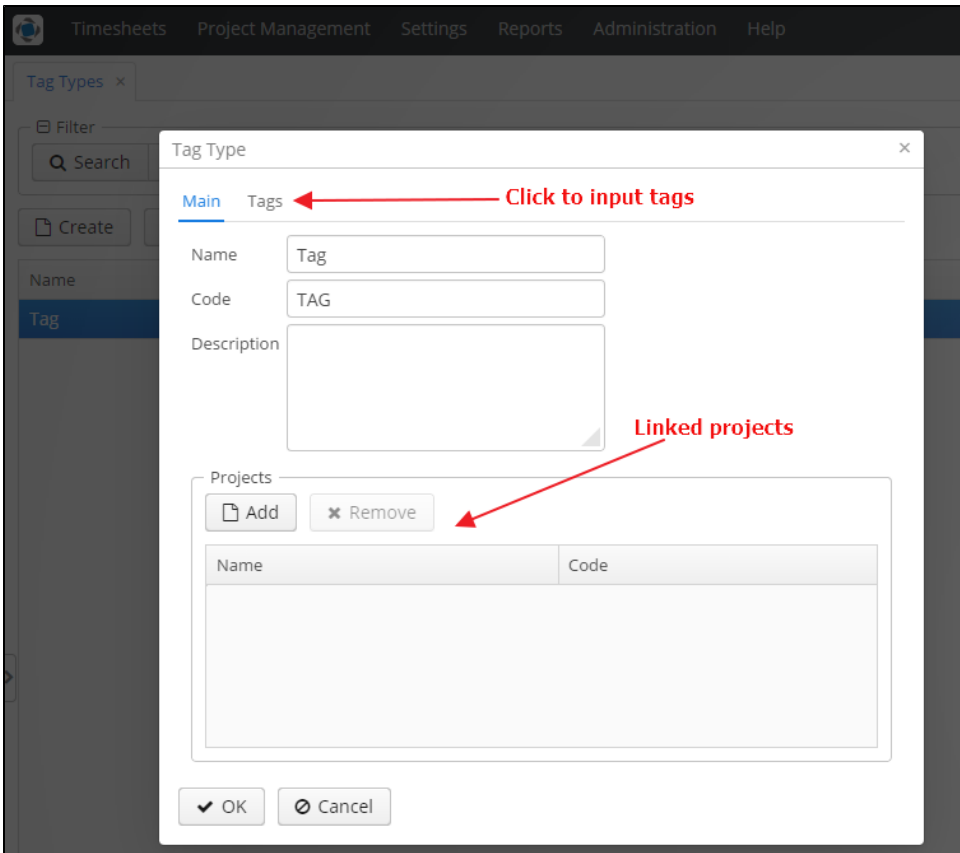
It is possible to setup Activity Types in Menu > Settings > Activity types. Activity types can be system wide (meaning they can be applied across all projects in the system) or project based. If an activity type is not linked with any project it is considered as system wide. Otherwise, if an activity type is linked with a set of projects, it is visible only for the time entries related to those projects.

## Tag Types and Tags

Occasionally, there may be a requirement to assign unstructured marks to a time entry, which cannot be defined in activity types, nor in task types. E.g. TBD (to be discussed), long chat, etc. For this purpose LoriTimesheets supports Tags.

To enable tagging go to setup Tag Types in Menu > Settings > Tag types and define a set of tags for each Tag Type.

You can link a certain tag type to a number of projects, in order to limit its applicability only to this one set. If a projects set is not specified from some tag type, then all of its tags will be applicable for all projects in the system. Furthermore, there is a possibility to define Required Tag Types and Default Tags for a certain task.



## Postgresql Settings

By default LoriTimesheets uses embedded HSQL as a data storage tier, because it doesn't require any additional setup. However, for the production environment it is assumed that users would prefer an enterprise database. For that purpose LoriTimesheets supports the Postgresql d

atabase. To enable Postgresql the following steps should be conducted:

1. Install Postgresql
2. Create a database named ts
3. Create local.app.properties file in the \$LoriTimesheetsInstallationDir/tomcat/conf/app-core directory. It should have the following content

```
cuba.dbmsType = postgres
```

4. Create app-core.xml file in the \$LoriTimesheetsInstallationDir/tomcat/conf/Catalina/localhost/ directory. It should have the following content.

```
<Context>
  <!-- Database connection -->
  <Resource driverClassName="org.postgresql.Driver" maxActive="20" maxIdle="2" maxWait="5000"
name="jdbc/CubaDS"
    password="root" type="javax.sql.DataSource" url="jdbc:postgresql://localhost/ts" username="root"/>
  <!-- Switch off session serialization -->
  <Manager className="org.apache.catalina.session.PersistentManager" debug="0" distributable="false"
    saveOnRestart="false">
    <Store className="org.apache.catalina.session.FileStore"/>
  </Manager>
</Context>
```

5. Adjust the following parameters in the app-core.xml file if needed: username, password, url