Handbook JAWS

Version: 0.0.4 (08/24/2014)

Flingelli, Markus

Version history

Version	Date of publication	Author
0.0.4	08/24/2014	Markus Flingelli
0.0.3	04/20/2014	Markus Flingelli
0.0.2	12/27/2013	Markus Flingelli
0.0.1	04/01/2013	Markus Flingelli

Handbook JAWS

Markus Flingelli

Table of Contents

List of Figures

List of Tables

List of Examples

Preface

JAWS (JAST Web Server) is the web application for managing JAST products via a web browser. JAWS can be deployed in an application server like Apache Tomcat or can run alone. In order to use JAWS you have to set up a database. At the moment following databases are supported:

- H2
- MySQL
- PostgreSQL

Chapter 1. Installation

1.1. Databse

In order to use JAWS you have to set up a database.

1.1.1. H2

H2 is a relational database management system written in Java. The software is available as open source software under modified versions of the Mozilla Public License or the original Eclipse Public License. The modification of the MPL is a shorter file header and the license name.

You can start the database simply with the following command:

```
java -jar h2-1.3.159.jar
```

For further information have a close look on the H2 homepage: H2 Databse Engine [http://www.h2database.com/html/main.html]

1.2. Application server

1.2.1. Apache Tomcat

If you have the rights to deploy a web application on an Apache Tomcat server you can upload the application via the web frontend.

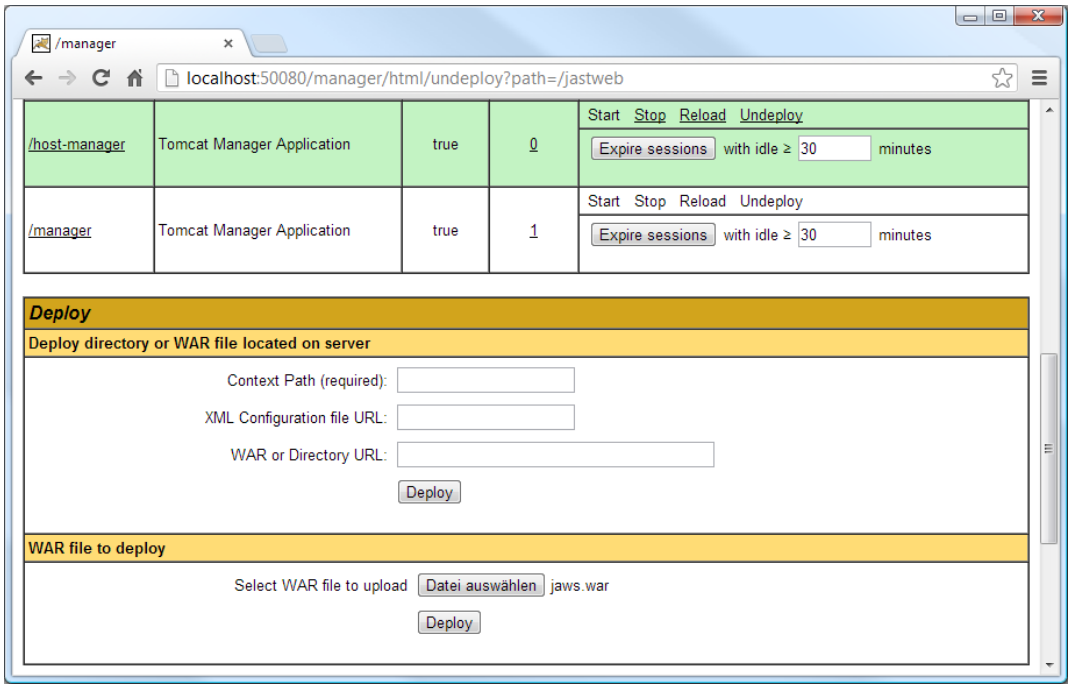


Figure 1.1. Installation of JAWS on Apache Tomcat

Now you can start the application, if you have configured the database.

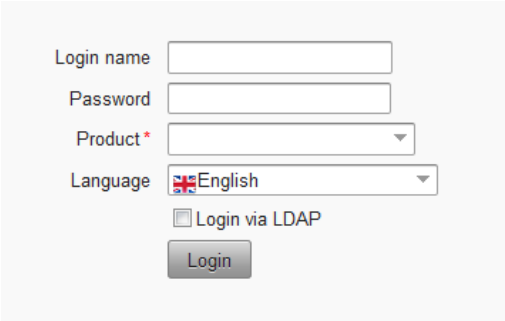


Figure 1.2. Login screen

1.3. Standalone

JAWS has an integrated web server (Jetty). Therefore you can start it by the following command:

```
java -jar jaws.war
```

1.4. Adjusting database

1.4.1. H2

You can see an example configuration of the database H2. The database is on the same machine as JAWS. The database is called `test`. If the database doesn't exist it will be created. The username for the database administrator is `sa` and the password is `secret`.

```
dataSource.driverClassName=org.h2.Driver
dataSource.dialect=org.hibernate.dialect.H2Dialect
dataSource.url=jdbc:h2:tcp://localhost/test;create=true
dataSource.username=sa
dataSource.password=secret
dataSource.hbm2ddl=update
dataSource.poolsize=2
```

1.4.2. MySQL

You can see an example configuration of the database MySQL. The database is on the same machine as JAWS. The database is called `test`. If the database doesn't exist it will be created. The username of the database administrator is `sa` and the password is `secret`.

```
dataSource.driverClassName=com.mysql.jdbc.Driver
dataSource.dialect=org.hibernate.dialect.MySQLDialect
dataSource.url=jdbc:mysql://localhost:3306/test;create=true
dataSource.username=sa
dataSource.password=secret
dataSource.hbm2ddl=update
dataSource.poolsize=2
```

1.5. Mail server

JAWS can send mails with appointments. You have to configure the IP address or the hostname of the mail server if you want to use it. The settings are stored in the file `mail.properties`. The mail server has to use port 25 and it must work without authentification.

You can see an example configuration. The mail server has the IP address `192.168.1.1`.

```
mail.smtp.host=192.168.1.1
```

1.6. LDAP server

JAWS can authentificate team members and web users against the local passwords and the passwords stored in a LDAP directory. The file `ldap.properties` contains the entries `ldap.host` and `ldap.port`.

If you set the value for the key `ldap.principal.prefix` the login is against the LDAP. On the login site you have only to insert the login name **`jdoe`** to login the user `jdoe`. You don't have to insert **`demo\jdoe`**. The prefix and the backslash will be automatically inserted.

```
ldap.host=ldap://192.168.1.2
ldap.port=389
ldap.principal.prefix=demo
```

Chapter 2. Configuration

2.1. Create product

You can easily create a new product by appending **`?createproduct=demo`** to the URL. A product called demo will be created. A user **`admin`** with the password **`password`** will be created, too. Now you must restart the application via **`?restartapplication`**.

Important

After the login you should change the password of the created user.

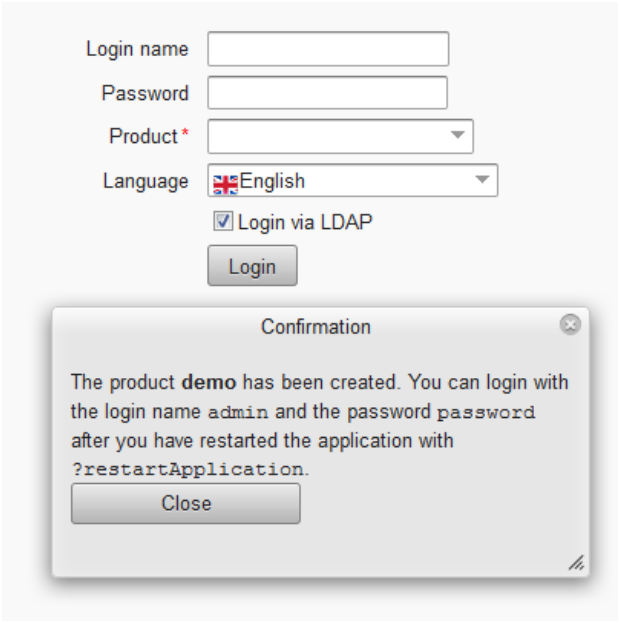


Figure 2.1. Create product

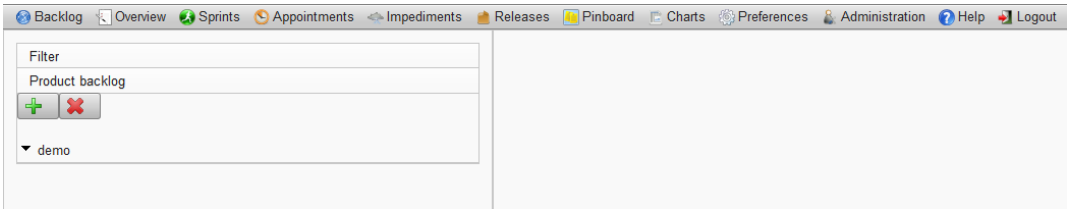


Figure 2.2. Product Backlog

2.2. Login

You can login locally or via LDAP. If you choose the option **`Login via LDAP`** the login works with LDAP. In this case an user with the corresponding user name in the product and the user name must exist in LDAP, too. The login checks the password that is stored in LDAP and not the locally stored password.

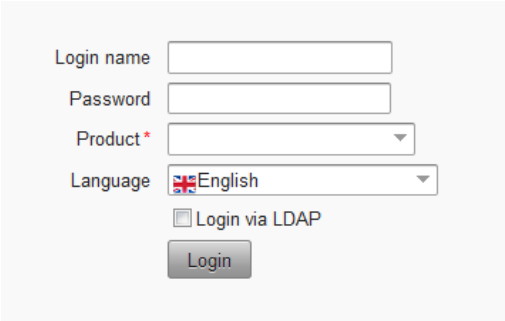


Figure 2.3. Login

Note

In the combobox **Product** only products are shown in which the given user exists. If the user specified in the textfield **Login name** exists only in one product this will be selected in the combobox **Product**.

2.3. Administration

The menu **Administration** consists of the three items **Export backlog**, **Logging** and **Version**.

If you select the menu item **Export backlog** the product backlog will be stored as XML file. This XML file can be opened and edited eg. with the application JAST.

The version of JAWS and its release date will be shown if you select the menu item **Version**.

If you choose the item **Logging** you can see the logging. On the web site you can only see the log items of the current day. Older log entries can be found on the the server in the sub directory `log`. The log entries will be put in ZIP archives. The latest 30 archive files will be stored.

Time	Level	Class	Message
20:06:11	INFO	de.flingelli.web.JastVaadinMain	User 'admin' successfully logged in to product 'demo'.

Figure 2.4. Logging

2.4. User management

A product contains members that are part of the Scrum team and so called web users. Members can be allowed to login via the web interface. Therefore the member must have a not empty login name and a password that has at least six characters.

Important

The login must be unique for all team members and web users.

There are three kinds of user rights implemented:

- Reader: This user is a web user and can only access read-only the backlog. He cannot see information like the login names of the web users and team members.
- Administrators: Administrators have full rights on the backlog. They can read and write.

2.4.1. Change password

If you want to change the password in JAWS the new one must at least consists of eight characters. No other requirements like password complexity are necessary. You can change the password by selecting **Preferences** in the menu bar. Now you choose the tab **Password** (see Figure 2.5, “Change password”). After you have inserted in the textfields **Password** and **Confirm password** the same password and pressed the button **Save** the password will be updated.

Important

If you change the password the local password will be updated. If you are logged in LDAP the local password will be changed and not the LDAP password.

Figure 2.5. Change password

Note

JAWS doesn't store the password in plain text. Each password will be concatenated with a so-called salt value. The hash value of that string will be stored. In the current version the hash alogrithm SHA-512 will be used.

2.4.2. Manage team members

In order to get to the team overview you have to select the product in the tree and than select the tab **Team**. You can add a new team member by pressing the button **Add**. Users can be edited or removed by selecting them first in the table and than press the corresponding button.

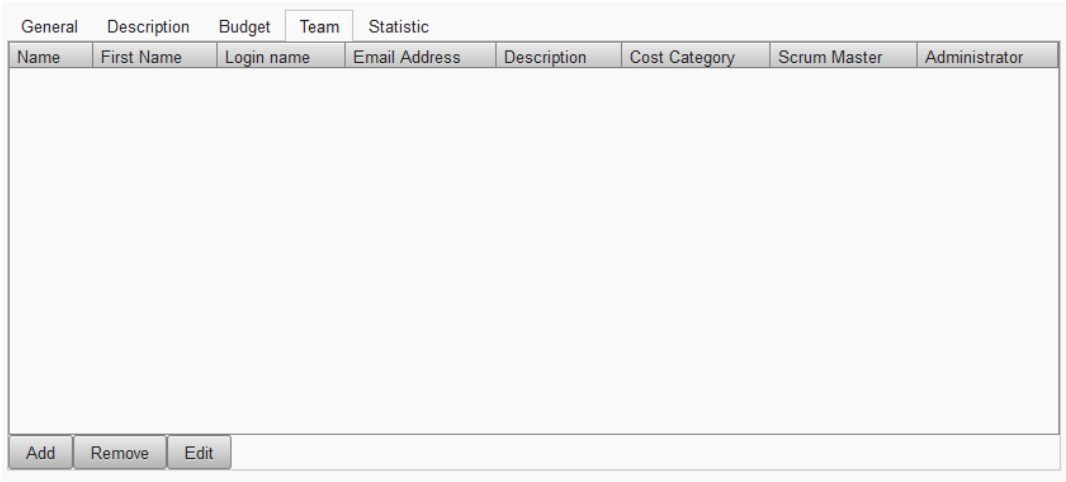
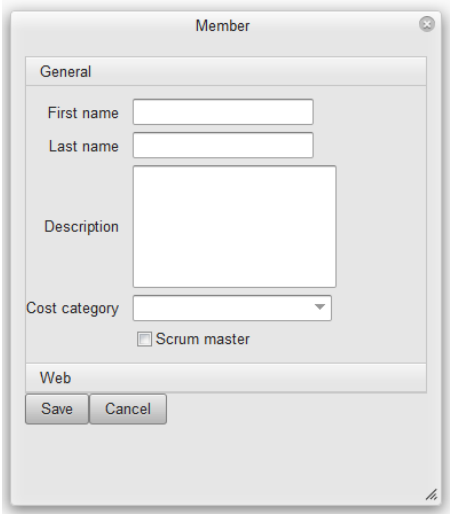


Figure 2.6. Team overview

If you press the button **Add** the member dialog will be shown (see Figure 2.7, “Member dialog: General”). In the section **General** you can insert the first name, the last name, the description and the cost category. You can set the option that the member becomes the Scrum master.



Important

Each team can consist of none or one Scrum master.

Figure 2.7. Member dialog: General

In the section **Web** you can insert the web relevant data: The login name, the password, the email address and you can set the option that the member becomes an administrator. Further information about the administrator rights are descriped in Section 2.4, “User management”.

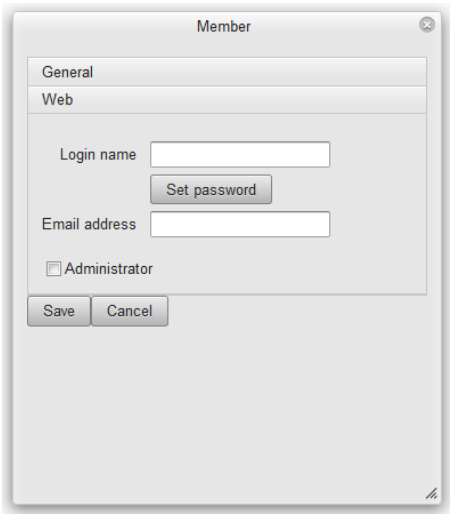


Figure 2.8. Member dialog: Web

Note

Only members with a non empty login name and a set password are able to login in JAWS.

Note

If you are logged in as team member and have administration rights you won't be able to remove this team member.

2.4.3. Manage web users

Web users are persons who have access to the product backlog without beeing team members. For example, the product owner could be a web user. If you set him as administrator he will be able to manage fully the product.

You get to the web user management by choosing the item **`Preferences`** in the menu bar than you have to select the tab **`Web users`**.

Figure 2.9. Overview of the web users

The dialog in order to add and edit web users is nearly identical with the dialog to add and edit team members. The only difference is the title of the dialog (see Section 2.4.2, “Manage team members”).

Important

Only webusers and team members with administration rights are allowed to add, edit and remove other webusers.

Note

Only web users with a non empty login name and a set password are able to login in JAWS.

Note

If you are logged in as web user and have administration rights you won't be able to remove this web user.

2.4.4. Manage own preferences

You can change the following data independently of you are a team member or a web user. You have to choose the menu item **`Preferences`**:

- First name,
- last name,
- email address,
- description and
- password.

In order to change other data you have to switch to the team member table or the list with the web users.

2.4.5. Login via LDAP

JAWS can authentificate team members and web users against a LDAP server. Therefore the file `ldap.properties` must be configured (see ???). If your LDAP domain is called `mycompany` and the user has the login name `jdoe` for example, you have to insert in the textfield **`Login name`** the string **`mycompany\jdoe`**.

Important

A login via LDAP with an empty password is not possible and the login will fail.

2.5. Manage Product Backlog

The product backlog is the fundamental source of information for a Scrum team. The product backlog consists of all the doing. It can be broken down to three idenure levels:

Epic	An epic arranges the backlog items. In pure Scrum there are no epics but it can make sense to introduce epics in large products.
Backlog item	A backlog item maps generally to a requirement.
Task	A task is assigned with exact one team member. A task is assigned with exact one backlog item. A backlog item consists at least of one task.

2.5.1. Manage product

If you select the product you see the tabs with the product specific information. You can see the product title, edit the product owner, choose if the estimation of the backlog items should be updated automatically and set the unit. There are at the moment the following units available: **`Hours`**, **`Days`**, **`Weeks`**, **`Weeks`**, **`Months`**, **`Function Points`** and **`Story Points`**.

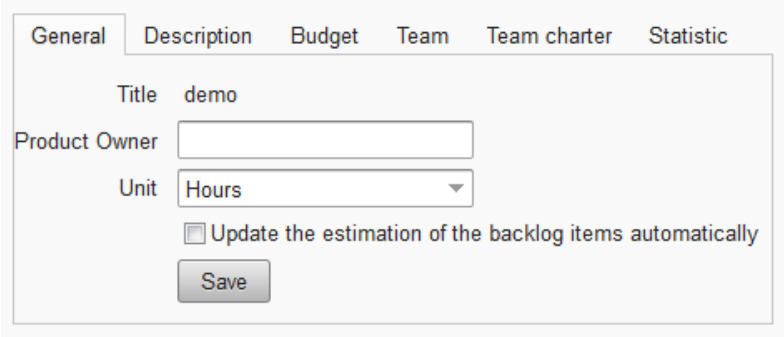


Figure 2.10. Product management

An optional description of the product can be set in the tab **`Description`**.

The tab **`Budget`** can be used for managing the product costs. You can choose the currency and set the budget of the project. You can add different cost categories.

Note

The name of a cost category must be unique.

The team overview behind tab **`Team`** was descriped in Section 2.4, “User management”.

You insert the charter of your team in the tab **Team charter**. The charter is valid while the whole product.

A summary of the product can be seen in the tab **Statistic**.

2.5.2. Create backlog item

You can add a backlog item by selecting the product and then press the add button (button with the plus sign). A backlog item can only been stored if the topic is unique in the product backlog.

Behind the text field **Estimation** there is a button to recalculate the estimation. If you press this button the estimation will be calculated from the estimation of the tasks.

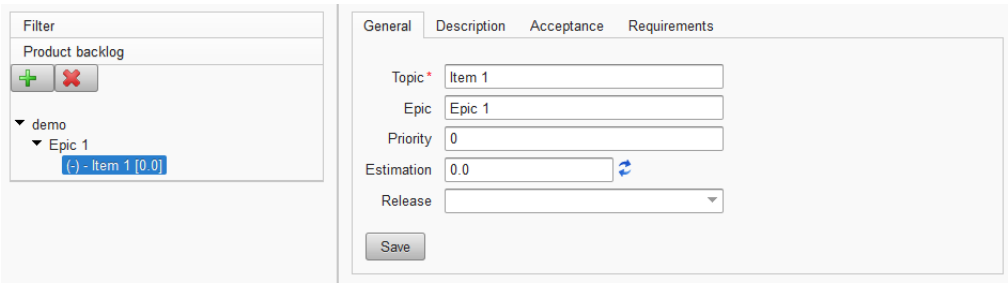


Figure 2.11. Add backlog item

2.5.3. Add task

You can add a task by selecting the corresponding backlog item and then press the add button (button with the plus sign). The title of the task must be unique within the backlog item. It is possible to have tasks with the same title if they are in different backlog items. The task title can not be empty.

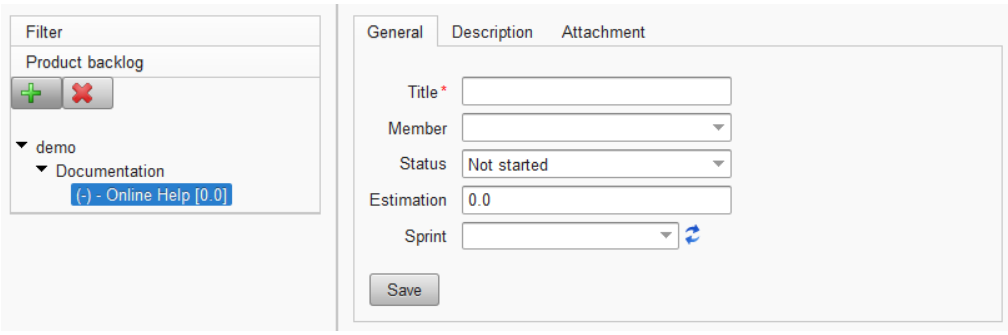


Figure 2.12. Add task

If you have administrations rights you can move tasks via drag and drop.

2.5.4. Delete item

You can delete a product, an epic, a backlog item or a task by selecting it and press the remove button (button with the red x). If you want to remove the product all the containing epics with their backlog items and tasks will be deleted. If you remove an epic all the containing backlog items and tasks will be deleted. If you remove a backlog item all the assigned tasks will be deleted.

Note

You have to confirm the remove action.

2.5.5. Filter backlog

If you press the item **`Filter`** you can filter the displayed backlog items and tasks. If you insert text in the textfield only backlog items and tasks will be shown which contains the text in the topic or the title (upper case and lower case will be ignored). You can show all the tasks of the current sprint, hide all assigned tasks. Furthermore you you can hide all done backlog items and all done tasks.

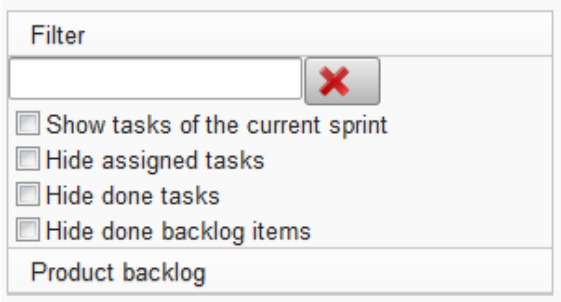


Figure 2.13. Filter backlog

2.6. Manage sprints

The sprints will be shown if you select **`Sprints`** in the menu bar.

Note

Sprints can only added and removed by users with administration rights.

2.6.1. Add sprint

You can add a new sprint by clicking the add button. The sprint title must be unique and the begin and end date must be set.

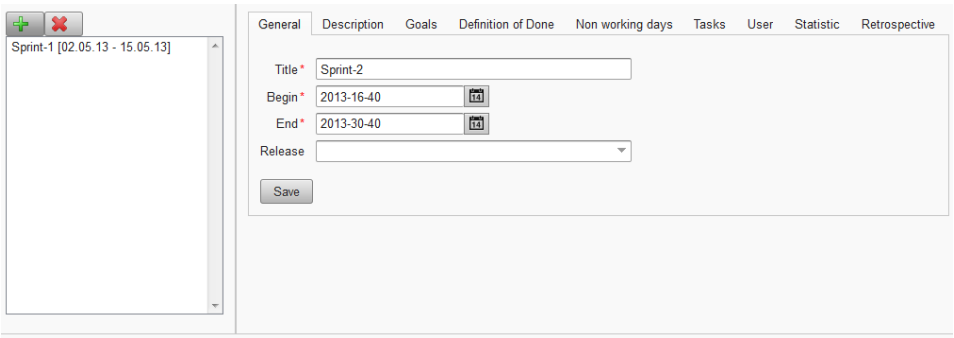


Figure 2.14. Add sprint

You can add a description, the definition of done, the goals and the non working days. Non working days are days that are involved in the product burndown chart and the other diagrams.

The result of the sprint retrospective can you save in the **retrospective** tab.

The sprint backlog is behind the tab **tasks**.

You can manage the effort in the tab **Users**. For each team member will be a tab created. If you select a task in the table you can edit the status of the task and you can add the effort.

Note

Only user with administration rights can edit the effort of all users. Team members without administration rights can only edit their own effort.

If you press the button **Create pinboard cards** in the the tab **Cards** pinboard cards with the tasks of the sprint will be generated. Per DIN A4 page will be two tasks printed.

2.6.1.1. Cover non working days

You can insert non working days by selecting the tab **Non working days**. Click with the left mouse button in the day you want to mark as a non working day. Then you have to confirm that the day should really marked as a non working day. You can delete a non working day by clicking with the left mouse button in the non working day.

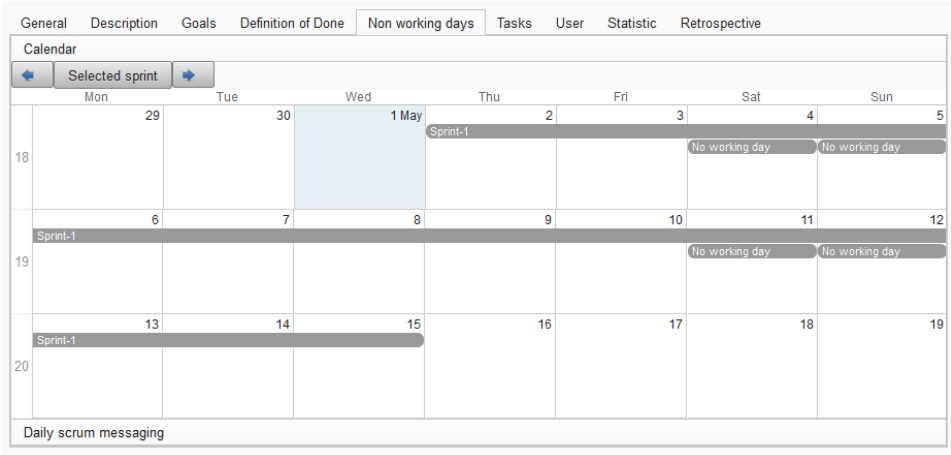


Figure 2.15. Non working days

Important

You can only insert non working days if you have administration rights.

2.6.1.2. Daily Scrum invitation

JAWS can send Outlook conform invitations. You have to select therefore the tab **Non working days** and select there the tab **Daily scrum messaging**. You can set the beginning of the Daily Scrum meeting in the textfield **Begin Daily Scrum**. The meeting will last 15 minutes. In the optional textfield **Location** you can give the location of the meeting. The combobox **Organiser** must be set. The organiser is typically the Scrum master. In the list **Invited participants** are the team members by default. You can add other participants from the group of the web users. The invitation starts at the second day of the sprint and ends one day before the sprint end. If the sprint contains weekends and all containing Saturdays and Sundays are marked as non working there will be sent only one invitation otherwise there will be sent an invitation for every single day.

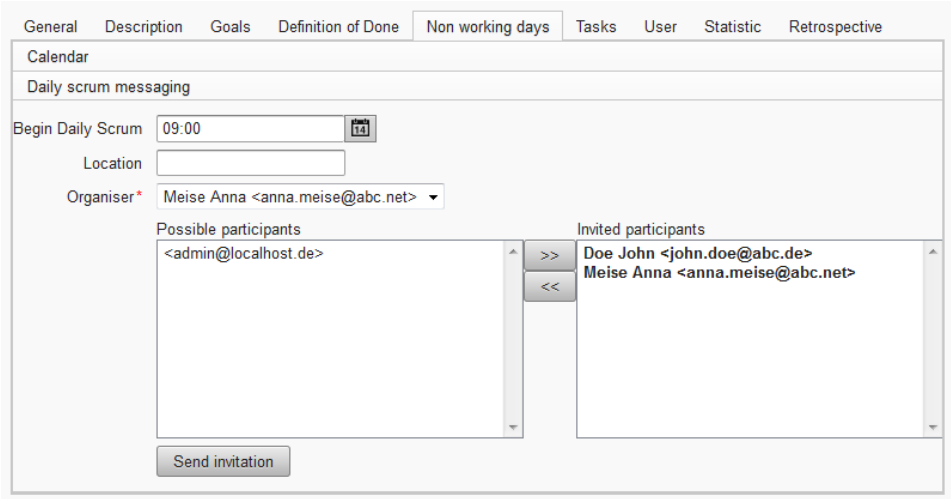


Figure 2.16. Daily Scrum invitation

Important

The sending of the invitation will only work if the SMTP is configured correctly. Every team member must have an e-mail address and there must exist a Scrum master in the team.

In order to cover the hours of work of a member you have to select the sprint and go to the tab **`User`**. In sub tabs you can see all users that have tasks in the sprint. Now you can go to the corresponding table and select the task by clicking the button **`Edit`**. You can insert the hours of work and afterwards you click the button **`Save`** in order to save the changes.

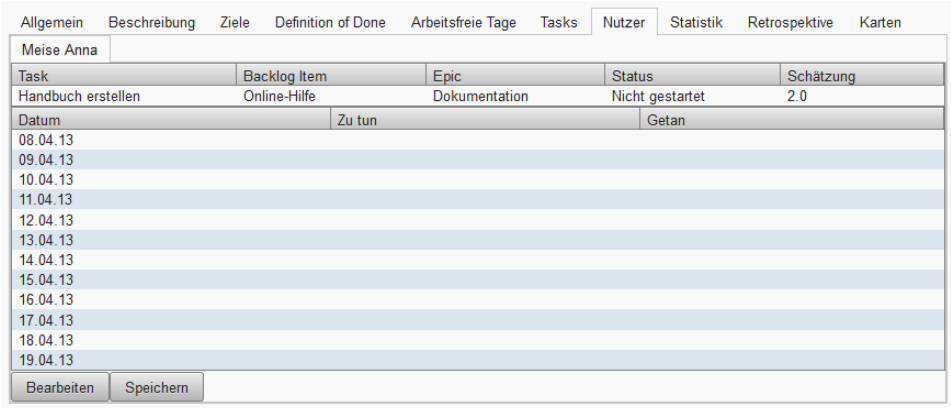


Figure 2.17. Cover costs

Note

You can only cover hours of works of the current day and days that are in the past.

2.7. Pinboard

The **`Pinboard`** shows the all tasks of the current sprint. The caption of a task card is the title of the task. Beneath the task title the topic of the backlog item is listed. In the left corner the name of the member is printed and in the left corner is listed how much of the task is already completed.

If you move the mouse cursor over the task title or the topic of the backlog item the task description will be shown.

Note

The pinboard will be updated every minute.

2.8. Diagrams

The following diagrams are available at the moment:

- Burndown,
- Sprint velocity,
- Member velocity and
- Sprint Burndown.

2.9. Formatting

In textfields with several input lines you can use a simple markup language. You can use simple itemizations, you can mark text bold and you can use the font Courier.

Original characters	Replaced by
`->`	→
`<-`	←
`<->`	↔
`=>`	⇒

`* Itemization level 1`
`  + Itemization level 2`
`  ~ Itemization level 3`
	• Itemization level 1
	  • Itemization level 2
	  • Itemization level 3

`**Bold print**`	**Bold print**
`'File name'`	`File name`
`---`	Horizontal line

`'public int sum(int n) {`
`  if (n <= 0) {`
`  return n;`
`  } else {`
`  return n + sum(n-1);`
`  }`
`}'`
	`public int sum(int n) {`
	`  if (n <= 0) {`
	`  return n;`
	`  } else {`
	`  return n + sum(n-1);`
	`  }`
	`}`

`` `public int sum(int n) {``
`  if (n <= 0) {`
`  return n;`
`  } else {`
`  return n + sum(n-1);`
` } ``
	`public int sum(int n) {`
	`  if (n <= 0) {`
	`  return n;`
	`  } else {`
	`  return n + sum(n-1);`
	`  }`

Original characters	Replaced by
}`	}

Table 2.1. Implemented Markup

Important

There will be no replacements in the program listing environment.

Note

Arrows are not replaced in the task cards.

2.9.1. Tables

You can code tables in JAWS with the pipe symbol (|). In tables you can use the markup for bold text and source code. The inline source cannot contain line breaks.

Note

Tables cannot be used inside of the itemization environment. You cannot use itemizations inside of tables.

Example 2.1. Table with two columns

The following code show a table consists of two columns.

```
|Colum 1|Colum 2|
|Value 1|Value 2|
```

By default all values in a cloumn are left-facing. You can adjust the alignment if you insert a line with the adjustment. The adjustment is valid till you add another adjustment line. If you insert `---` the text will be left-facing and `:---:` for justified and `---:` for right-facing.

Example 2.2. Adjust columns in tables

The following source code will adjust the content of columns. In the first column the values `Column 1` and `1` will be left-facing and the value `4` will be justfied. In the second column the values `Column 2` and `2` will be justified and the value `5` will be right-facing. In the third column the values `Column 3` and `3` will be right-facing and the value `6` will be left-facing.

```
|---      |:---:    |---:     |
|Column 1|Column 2|Column 3|
|1        |2        |3        |
|:---:    |---:     |---      |
|4        |5        |6        |
```