RefereeManager

Developer Documentation

12. November 2017

Ekkart Kleinod

✉ ekleinod@edgesoft.de

Inhalt

1 Preface

This documentation focuses on the development aspect of the RefereeManager.

1.1 Used Language

The developer's documentation is in English. As I am a German native speaker, you can ask German questions every time.

Die Entwicklerdokumentation ist auf Englisch. Da ich Deutsch als Muttersprache spreche, können Sie natürlich jederzeit auch deutsche Fragen stellen.

2 Structure and Git repositories

The RefereeManager is written in Java 8+, using JavaFX, JAXB and language features introduced in Java 8.

2.1 Structure

The repository is structured as follows:

folder	content
`build`	Build files for releases.
`documentation`	User and developer documentation.
`files`	Released files (will change in the future).
`refereemanager`	Eclipse project with sources, ressources and tests.
`submodules`	Needed submodules (edgeutils for Ant and JAXB commons).

2.2 Git-Repository/github

The git repository is maintained using github at:

- https://github.com/tt-schiri/refereemanager/

Github is used for code maintenance and issue tracking.

Short information about the structure of the git repository:

The branches are constructed regarding the git branching model of

- http://nvie.com/posts/a-successful-git-branching-model/

This means, there are always at least three branches:

1. `master` - contains released versions
2. `develop` - main synchronisation branch for feature, release, and hotfix branches
3. `feature/work` - main working branch for development

Additionally, the following branches may occur:

- `feature/*` - writing a special feature
- `release/*` - synchronizing release versions between `develop` and `master`
- `hotfix/*` - fast bugfixes

3 Software

I develop under LinuxMint (Ubuntu) and Windows. Installer creation for Windows can be done in Windows and LinuxMint, for Linux (Ubuntu) in LinuxMint only.

In order to develop the RefereeManager, you will need:

- Java 8+
- Git
- Ant (Java)

I strongly recommend using:

- eclipse
- Multimarkdown
- LaTeX

In order to create releases, one needs:

- NSIS
- dpkg

3.1 Java 8+

The RefereeManager is written in Java 8+, using JavaFX, JAXB and language features introduced in Java 8.

3.2 Git

You need *git* for handling the developed code.

3.3 Ant

You need *ant* for execution of the JAXB and release scripts. *Ant* requires *Java*. You can choose not to use *ant*, in this case you have to execute the required scripts by hand.

3.4 eclipse

The development of thr RefereeManager takes place in eclipse, an eclipse project and launches are provided in the repository.

3.5 Multimarkdown

The pdf version of the documentation is created using Multimarkdown.

Additionally, many of the templates for letter, or list creation use Multimarkdown.

3.6 LaTeX

The pdf version of the documentation is created using LaTeX.

Additionally, many of the templates for letter, or list creation use LaTeX.

3.7 NSIS

The Windows installer is created with NSIS.

3.8 dpkg

The Debian (Ubuntu) installer is created using dpkg.

4 Setup your working environment

In short, do the following:

1. check out the git repository
2. update/check out the submodules
3. import the eclipse project `edgeutils`
4. install `edgeutils` via maven
5. import the eclipse project `refereemanager`
6. wait for the maven dependencies to install
7. start development

The repositories are:

- refereemanager at github[1]
- edgeutils at github[2]

[1] https://github.com/tt-schiri/refereemanager/

[2] https://github.com/ekleinod/edgeutils/

5 Implementation

Here, you find some details/tips about/for the implementation.

5.1 New Overview

An overview contains a list on the left, a detail view on the right with means to edit the data.

An example is the referee overview, visible using menu “Schiedsrichter” -> “Übersicht”, or Ctrl+Alt+R.

For a new overview you have to create the following files (example overview for trainees):

– DetailsTrainee.fxml
– DetailsTraineeController.java
– ListTrainees.fxml
– ListTraineesController.java
– OverviewTraineesController.java

For the editing features you have to implement:

– TraineeEditDialog.fxml
– TraineeEditDialogController.java

If needed, you have to implement sub-edit-dialogs, such as contact edit dialog, or wish edit dialog.