



Documentation 0.0.1

- 1. Introduction
 - 1.1 What is commentBox?
- 2. Setup
 - 2.1 Download
 - 2.2 Dependencies
 - 2. CommentBox component
 - 2.2 Attributes (Properties)
 - 2.3 Attributes (Events)
- 4 Getting Started
- 5. Additional Features
 - 5.1 Pagination
 - 5.2 Live Features
 - 5.3 PrettyTime Support
- 6. License

1. Introduction

1.1 What is commentBox?

Commentbox is an open source JSF component that provides out of the box commentary functionality.

- All standard commentary functionalities (Create Comment, Reply, Edit, Delete)
- Advanced functionalities (Like, Report as Spam)
- Live Features (typing users, notification for new Comments, fetch new Comments)

2. Setup

2.1 Download

The commentBox component is provided as a single jar file. This file can be obtained by either compiling the code yourself (with maven) or by downloading the, latest and greatest, auto built.

https://github.com/nickrussler/commentbox#download

2.2 Dependencies

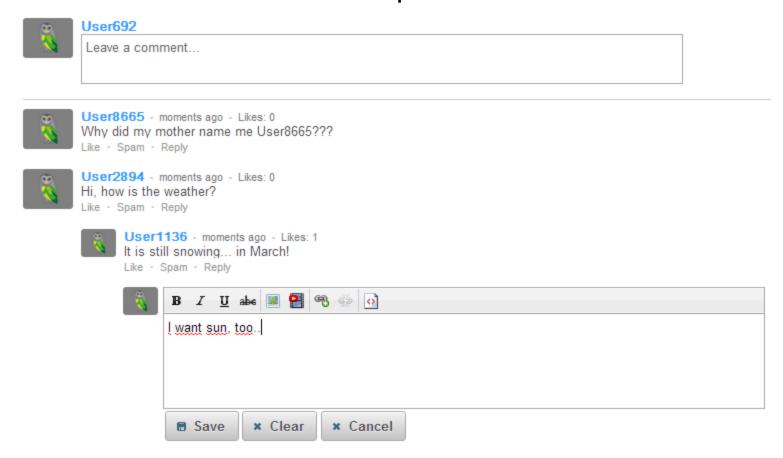
CommentBox has only a JSF 2.x implementation and PrimeFaces as mandatory dependencies, but there are other optional dependencies for advanced features.

JSF Runtime
PrimeFaces
PrettyTime
PrimeFaces Push

| Dependency | Version * | Туре | Description |
|---------------------------------|------------|-----------|-------------------------------|
| JSF Runtime | 2.0 or 2.1 | Mandatory | Mojarra or MyFaces |
| PrimeFaces | 3.5 | Mandatory | PrimeFaces |
| PrettyTime | 2.3.1 | Optional | nice date formatting |
| Primefaces Push (Atmosphere) | 1.0.11 | Optional | Live Features (new comments,) |

^{*} Listed versions are tested and known to be working with commentBox, other versions of these dependencies may also work but are not tested.

3. CommentBox component



3.1 Attributes (Properties)

| Name | Default | Туре | Description |
|----------------------|---------|--------------------------------------|---|
| contextID | null | String | Unique identifier of the component context |
| comments | null | java.util.List <comment*></comment*> | comments to display |
| currentPage | null | Long | current active page value |
| commentsPerPage | null | Long | Number of comments per page |
| commentCount | null | Long | Total number of comments |
| currentUserID | null | Object | Unique identifier of the current User |
| currentUserUsername | null | String | Username of the current User |
| currentUserAvatarUrl | null | String | URL of the avatar image of the current User |
| canEditAll | false | Boolean | If set to true the current user can edit posts of other users |
| canDeleteAll | false | Boolean | If set to true the current user can delete posts of other users |
| enableLiveFeatures | false | Boolean | If set to true Live features are enabled (e.g. new comment) |
| editorControls | []* | String | controls of the editors |

Comment* = info.whitebyte.component.commentbox.Comment [...]* = bold italic underline strikethrough | image youtube | link unlink | source

3.2 Attributes (Events)

| Name | Parameters | Туре | Trigger |
|--------------------|------------|---------|------------------------------------|
| onFetchNewComments | - | void | Load Comments |
| onFetchNewAnswers | Comment* | void | Hold the current active page value |
| onPageChange | Long | void | On Page Changes |
| onCreateComment | Comment* | boolean | On Creation of new Comment |
| onCreateAnswer | Comment* | boolean | On Creation of new Answer |
| onEditComment | Comment* | void | On Edit of Comment |
| onCommentLike | Comment* | void | On Like of Comment |
| onCommentSpam | Comment* | void | On Spam Report of Comment |
| onCommentDelete | Comment* | void | On Deletion of Comment |

Comment* = info.whitebyte.component.commentbox.Comment

These Java Side Events give a way to react to user interaction with the component.

Examples:

The user changes the page, the *comments* list has to be refilled with the comments of the next (or previous) page.

The user created a comment, since in most cases we want to persist the new comment, we have to save the new comment in some database.

5. Getting Started

Fill a list with comments (info.whitebyte.component.commentbox.Comment) you want to display in your backing bean.

```
<c:commentBox comments="#{backingbean.comments}" />
```

This is it, you should see your list of comments displayed.

A good way to understand how commentBox works, is to take a look at the <u>commentBox</u> <u>example</u>.

5. Additional Features

5.1. Pagination

To enable pagination, you need to specify a *commentCount* and how many comments you want to show per page by specifying *commentsPerPage*. You also need to specify a the *currentPage*. If the user changes the page the value of *currentPage* is changed by the commentBox component.

So if you need the current value of *currentPage* in your backing bean you need to bind *currentPage* to some value in your backing bean.

5.2. Live Features

You have to set enableLiveFeatures to enable the live features.

You also have to add the Atmosphere framework to your project. You can read how to do this in the Primefaces documentary under Primefaces Push Chapter.

5.3. PrettyTime Support

CommentBox offers optional support for PrettyTime. If you want to have your comment creation date formatted like this: moments ago instead of 13:10 2013 you have to add the prettytime library and the prettytime-integration-jsf library to your projects buildpath. CommentBox will automaticly detect the presence of the libraries and use them.

6. License

This project is open source and free, it is available under the Apache v2 License.