

EXT: h_book

(Extension key: h_book)

Martin Helmich

16th September 2007

This document is published under the Open Content License available from

<http://www.opencontent.org/opl.shtml>

The content of this document is related to TYPO3 – a GNU/GPL

CMS/Framework available from *www.typo3.com*

Contents

1	Introduction	2
1.1	What does it do?	2
1.1.1	Text parsing	2
1.1.2	Security issues	2
1.2	Screenshots	3
2	Users manual	4
2.1	Installation	4
2.2	Inserting the h_book	4
2.2.1	Listing all entries	4
2.2.2	Creating new entries	4
2.2.3	Editing entries	4
3	Configuration	6
3.1	TypoScript Reference	6
3.1.1	General properties	6
3.1.2	Entry creation	6
3.1.3	Entry listing	8
4	Credits	9

1 Introduction

1.1 What does it do?

The extension H_BOOK offers a powerful, yet easy configurable guestbook with various features.

1.1.1 Text parsing

The H_BOOK automatically parses the entry text for common BBCode tags (like [B] [/B] for bold printing and so on...) and substitutes them with the proper HTML tags. Smilies like :) or ;) are substituted with a proper image. The H_BOOK also parses links and email addresses found in the entry text.

1.1.2 Security issues

The H_BOOK offers several ways of spam protection. Any entry containing a word found on a specific black list is not saved at all. This black list contains common spam words like "sex", "porn", and so on. Furthermore, an entry can optionally be verified by a randomly generated verification code that has to be copied manually from an image into an input field during entry creation. The administrator can block single IP addresses for a certain amount of time.

H_BOOK entries can be administrated either by backend users using the TYPO3 proprietary forms or by frontend users using a specific page protected by the TYPO3 frontend authentication.

1.2 Screenshots

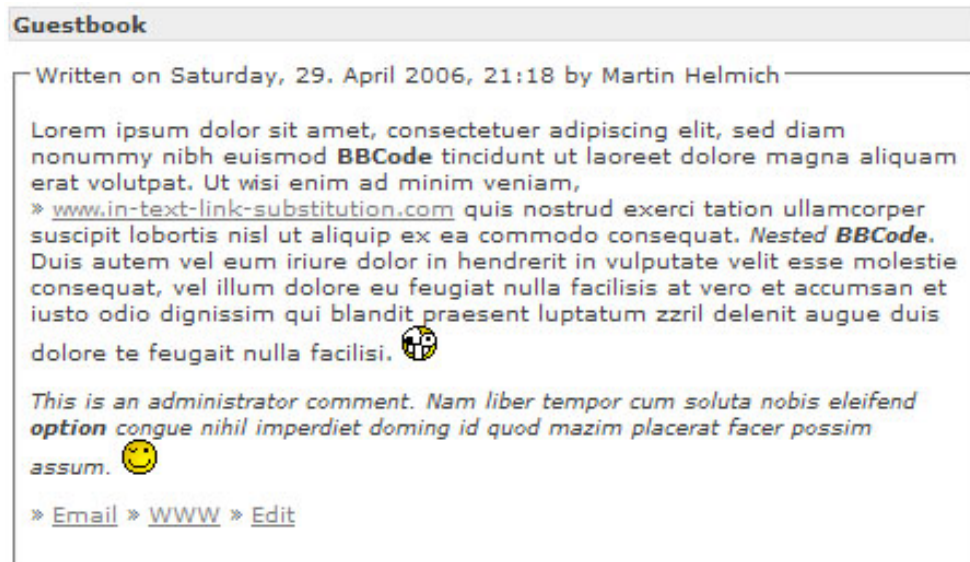


Figure 1: The default list view with smilie and BBCode support.



Figure 2: The entry creation form

2 Users manual

2.1 Installation

You can install the plugin quite easily via the TYPO3 Extension Manager. There is nothing special you have to consider during the installation.

After installing, you should customize and configure the H_BOOK to make it fit into your page design.

2.2 Inserting the h_book

2.2.1 Listing all entries

Now the time has come to include the H_BOOK into your website. Create a new page and insert a new content element. In the section *plugins* you will find the H_BOOK plugin. You can now configure this H_BOOK instance when editing this content record. At first, you should select *List all entries* in the *What to display* field in the *General settings* section. In this setting the H_BOOK will list all guestbook entries. You can change the sorting mode by changing the *Order By* field in the *Display* section.

2.2.2 Creating new entries

Now you can list all guestbook entries that were ever submitted. But this makes sense only, if visitors can submit entries at all. For this, you can either add *Create new entry* to the *What to display* field in the record you created in step 1, or create a new page with a new content element and then set *What to display* to *Create new entry* there.

Note: If you create a new page for entry creation you have to tell the H_BOOK where to store the guestbook entries. The best way is to create a SysFolder (create a new page and select *SysFolder* as page type) and set the *Entry storage page* in both plugin instances to this page.

2.2.3 Editing entries

If you want to edit guestbook entries, you can either do this using the TYPO3 proprietary backend forms (go to the entry storage page you probably specified in step 2, select *List* and then select the H_BOOK entry you want to edit). If you want certain frontend users (for example moderators on a bigger page) to have the right to edit guestbook entries without giving them backend access, you can create a new page with a new H_BOOK instance. Here, you set *What to display* to *Edit entry*. Remember to set the *Entry storage page*.

In the plugin displaying the entry list, you should now set the *edit page* in the *General settings* section to the page you just created.

To prevent every ordinary visitor from editing guestbook entries, you should restrict the access to this page to a certain group of fe_users.

3 Configuration

The H_BOOK offers a wide range of configuration possibilities both on HTML Template and on TypoScript level.

3.1 TypoScript Reference

3.1.1 General properties

- templateFile** (*Data type: file*) — The template file
Default: EXT:h_book/pi1/h_book.template.html
- storagePID** (*Data type: page UID*) — The page the guestbook entries are stored on.
- local** (*Data type: boolean*) — If set, the H_BOOK stores guestbook entries on the page the plugin was instanciated on.
Default: 0 (false)
- ipBlockDuration** (*Data type: int*) — This defines the amount of time an IP address remains blocked.
Default: 86400 (1 day)
- smilies.display** (*Data type: IMAGE*) — This cObj is the basic image for all smilies substituted in the entry text.
- smilies.data** (*Data type: COA*) — An array of all smilies to be substituted in the entry text. Each item should inherit the properties of **smilies.display**. Only the *file* and the *tag* property should be unique for each smilie. The *tag* property defines the character sequence that is substituted with the image (like e.g. :) or ;))
- enableAdditionalFields** (*Data type: string*) — A comma-seperated list of optional fields to enable. Possible values: email, homepage, location, icq, aim, msn
Default: email, homepage, location, icq, aim, msn

3.1.2 Entry creation

The entry creation configuration can be accessed by `plugin.tx_hbook_pi1.create`

- required** (*Data type: string*) — A comma-seperated list of entry fields that are required when creating a new entry. Valid values are name, text, email and homepage.
Default: name,text

label (*Data type: stdWrap*) — All input field labels are parsed with these options

requiredLabel (*Data type: stdWrap*) — All required input field labels are parsed with these options. Regularly inherits all settings of `create.label`

ipblocking (*Data type: boolean*) — If set, IP blocking is enabled. This allows the administrator to temporarily block certain IP addresses from creating entries.
Default: 1 (true)

filter (*Data type: boolean*) — If set, entries are blocked if they contain a certain word from a spam word blacklist (`create.filter.words`)
Default: 1 (true)

filter.words (*Data type: string*) — A comma-separated list of unwanted words. All entries containing one of these words are not displayed.

spamCode (*Data type: boolean*) — If set, all visitors creating a guestbook entry have to verify that they are human and not a spam bot by manually copying a randomly generated code into an input field.
Default: 1 (true)

spamCode.length (*Data type: int*) — The spam protection code length
Default: 8

spamCode.background (*Data type: array*) — The background color of the image displaying the spam protection code. The color is defined by three properties called `.r`, `.g` and `.b`
Default: r=255;g=255;b=255

spamCode.text (*Data type: array*) — The text color of the image displaying the spam protection code. See `create.spamCode.background`
Default: r=0;g=0;b=0

errorList.item (*Data type: stdWrap*) — If an error occurred during entry creation, the HBook displays a list of errors. Each error is parsed with this `stdWrap` object.

inputField.width (*Data type: int*) — The width of the entry creation form input field width in characters
Default: 64

3.1.3 Entry listing

The entry listing configuration can be accessed by `plugin.tx_hbook_pi1.display`

`select` (*Data type: select*) — The select options for loading the guestbook entries

`labels.edit` (*Data type: cObj*) — The edit button

`labels.email` (*Data type: cObj*) — The email button

`labels.homepage` (*Data type: cObj*) — The homepage button

`labels.icq` (*Data type: cObj*) — The ICQ button

`labels.msn` (*Data type: cObj*) — The MSN button

`labels.aim` (*Data type: cObj*) — The AIM button

`location` (*Data type: cObj*) — The location

`text` (*Data type: stdWrap*) — The parsing options for the entry text.

`comment` (*Data type: stdWrap*) — The parsing options for the admin comment. This object usually inherits its attributes from the `text` object.

`author` (*Data type: stdWrap*) — StdWrap options for the author

`link` (*Data type: stdWrap*) — StdWrap options for in-text links

`date` (*Data type: stdWrap*) — StdWrap options for the entry date

`pagebrowser` (*Data type: stdWrap*) — StdWrap options for the page browser

`pagebrowser.item` (*Data type: stdWrap*) — StdWrap options for a single page link. The link texts for first, previous, next and last page are defined by the objects `.first`, `.prev`, `.last`, `.next`

4 Credits

Thanks to Katja Lampela for the finnish frontend translation.