



ModuleStudio 0.5.2

News and showcases

Axel Guckelsberger [guite.de]



Events since last camp

- 2010-09 MOST 0.5.1 release
- 2011-02 Git migration
- 2011-08 MOST 0.5.2 beta stage
- 2011-09 MOST 0.5.2 release



Some numbers

- Commits since git migration
 - 1200 Generator
 - 2000 Other components
- Migrations
 - 4 smaller applications updated within 2 days.
 - More and more things can be achieved only by template overriding or simple extensions.



New features in 0.5.2

- Full support for Zikula 1.3
- Full support for Doctrine 2.1 and all extensions
- Reporting generator creating metrics for the project management and other artifacts
- Some editor improvements
 - Categories and descriptions for properties
 - A generic documentation field for all elements
 - Addition of missing label edit possibilities



Zikula 1.3 support

- Object-oriented module structure
- Includes recent core enhancements
 - Core event listeners
 - New hook system
 - Thumbnail creation with Imagine



Doctrine 2.1 support

- Entity and column mappings
 - Locking strategies combined with PageLock
- Association mappings
 - Unidirectional and bidirectional
 - Fetch types
 - Cascade options on application and database level
 - Ordered and indexed associations
- Supported but not stable yet in created controllers
 - Inheritance hierarchies
 - Aggregate fields
 - Composite keys



Generator – Field types

- Boolean
- Integer, Decimal, Float, User
- String, Text, Email, Url
- Datetime, Date, Time (past / future validation)
- Upload (naming schemes, allowed file types)
- Array, Object
- Many settings possible, explore the field properties.




Generator – More field types


- Enabled by String field validation properties
- Country fields
- Colour fields (form plugin for colour picker)
- Language fields (dropdown list with available locales)
- Regular expression (e.g. for ip addresses)



Generator – More field types

 RecipeManager

[Recipes](#) | [Image](#) | [Drink recipes](#) | [Menus](#) | [Menu positions](#) | [Ingredients](#) | [User ingredients](#) | [Categories](#) | [Units](#) | [Recipe steps](#) | [Recipe ingredients](#) | [Recipe step ingredients](#) | [Ingredient unit assignments](#) | [Recipe areas](#) | [Ingredient categories](#) | [Configuration](#) | [Hooks](#)

 **Edit category**

Content

Name *

Category name 2

Sort value *

1

Colour

#5E99B2

Langcode

Englisch

Alle

Deutsch

Englisch


Creation and update


Created by admin


Created on 29. Aug 2011

Updated by admin

Updated on 29. Aug 2011

 **Update category**

 **Delete category**

 **Cancel**

Powered by ModuleStudio 0.5.2

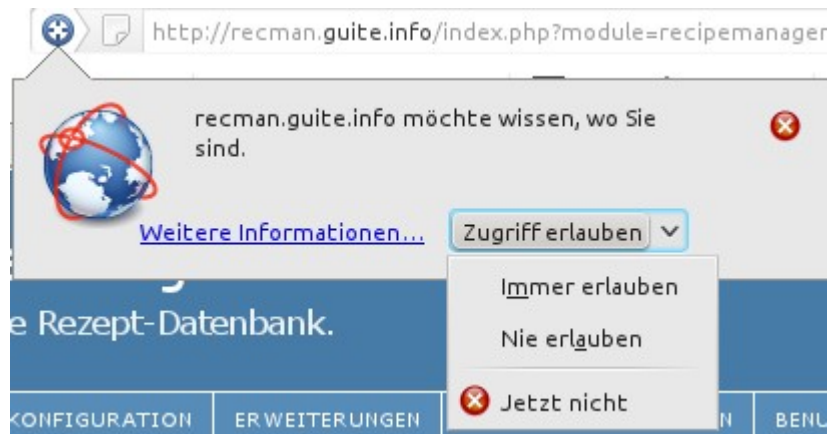
#5E99B2

Close



Extensions – Geographical

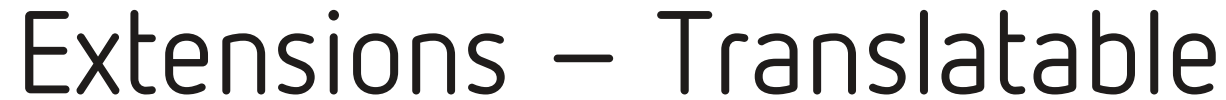
- Adds coordinate fields to an entity
- html5 geolocation support during creation for determination of default coordinates





Doctrine Extensions

- Loggable – revision management with version fields
- Sluggable – permalink handling
 - Different configuration options in the model
- Sortable – define fields as sortable position / group
- Timestampable – automatic date assignments
 - Triggers on creation, update or property changes



- [illegible]

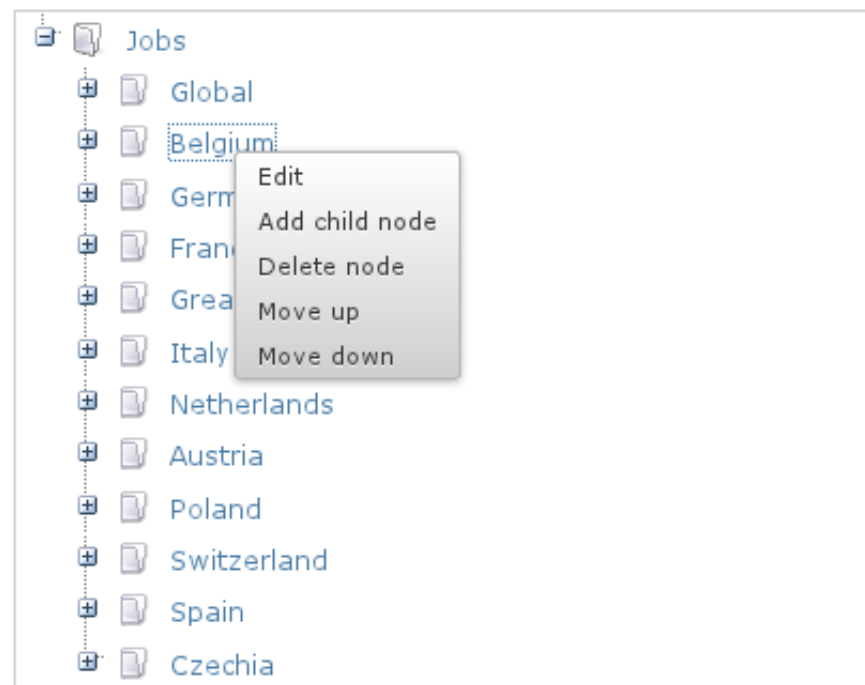
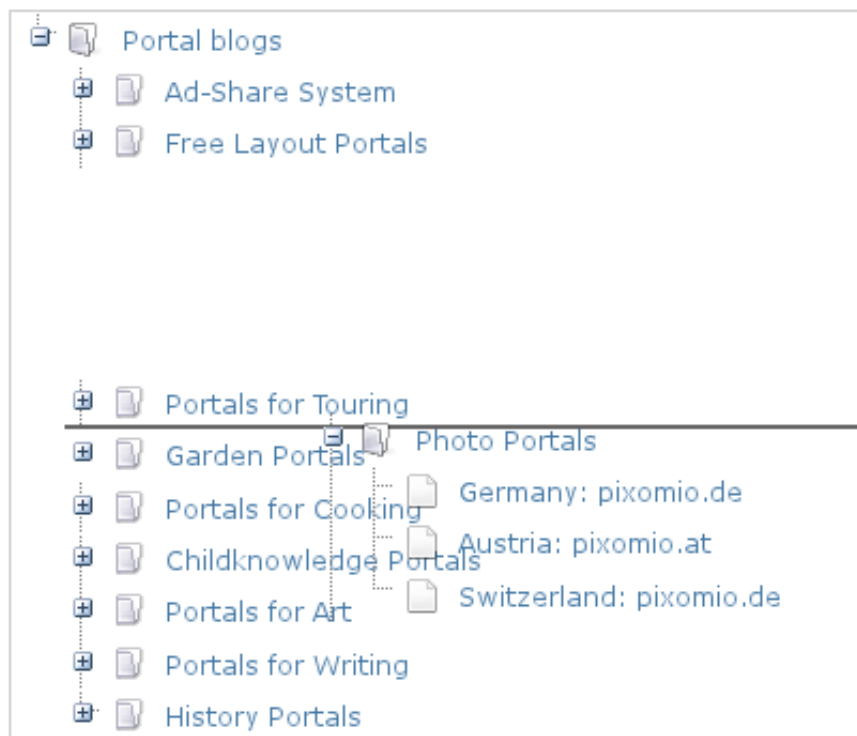


Doctrine Extensions – Tree

- Two kinds: Nested set vs. Closure

Category hierarchy

 Add root node  Switch to table view





Zikula Extensions

- Additional extensions contributed by Zikula core
 - Attributes
 - Categories
 - Meta data
 - Standard fields
- Zikula 1.3.1-dev required

Categories

Category

- Blogging
- Musik und Audio
- Kunst und Fotografie
- Schreiben und Denken
- Kommunikation und Medien
- Reisen und Kultur
- Wissenschaft und Technik
- Sport und Aktivitäten
- Wirtschaft und Arbeit

Creation and update

- Created by admin
- Created on 29. Aug 2011
- Updated by admin
- Updated on 29. Aug 2011

Metadata

Title

Author


Subject


Keywords

Description

Publisher

Contributor

Start date 

End date 

Type

Format

Uri



Extensions – Combinations

- Example: Sluggable and Translatable
 - If a translatable field is defined as part of the slug then also the slug will be translatable.
- Example: Translatable and Tree
 - The translation table does not need to care about the tree-related fields as those are not defined as translatable.



Template output formats

- Normal templates
- Feeds
 - RSS
 - Atom

Latest recipes

A direct feed showing the list of recipes - Die neue Rezept-Datenbank.

[29. Aug 2011 - Drink recipe title 1](#)

Drink recipe description 1

[29. Aug 2011 - Drink recipe title 2](#)

Drink recipe description 2

[29. Aug 2011 - Drink recipe title 3](#)

Drink recipe description 3

[29. Aug 2011 - Drink recipe title 4](#)

Drink recipe description 4

[29. Aug 2011 - Drink recipe title 5](#)

Drink recipe description 5

[29. Aug 2011 - Recipe title 1](#)

Recipe description 1

[29. Aug 2011 - Recipe title 2](#)

Recipe description 2

Latest ingredients

A direct feed showing the list of ingredients

[Ingredient name 1](#)

29.08.2011 18:01

Ingredient description 1

[Ingredient name 2](#)

29.08.2011 18:01

Ingredient description 2

[Ingredient name 3](#)

29.08.2011 18:01

Ingredient description 3



Template output formats

- CSV
- XML
- JSON

```
-<recipes>
  -<recipe id="1" createdon="29. Aug 2011" updatedon="29. Aug 2011">
    <MenuPosition/>
    <title>Drink recipe title 1 </title>
    <description>Drink recipe description 1 </description>
    <amountOfPortions>1 </amountOfPortions>
    <author>1 </author>
    <parties>Drink recipe parties 1 </parties>
    <country>de </country>
    <difficulty>1 </difficulty>
    <downloadFile/>
    <totalDuration>0 </totalDuration>
    <additionalData>1 </additionalData>
    <additionalObject>Recipe additional object 1 </additionalObject>
    <recipeVersion>1 </recipeVersion>
  </recipe>
  -<recipe id="2" createdon="29. Aug 2011" updatedon="29. Aug 2011">
    <MenuPosition/>
```

```
{
  "date": "2011-08-29 17:01:01", "timezone_type": 3, "timezone": "Europe\\Berlin", "updatedAt": {
    "date": "2011-08-29 17:01:01", "timezone_type": 3, "timezone": "Europe\\Berlin"
  }, "_objectType": "recipe", "_idFields": [], "_validator": null, "_hasUniqueSlug": false, "title": "Drink recipe title 5", "description": "Drink recipe description 5", "amountOfPortions": 5, "author": 5, "parties": "Drink recipe parties 5", "country": "de", "difficulty": 5, "downloadFileMeta": {
    "downloadFile": "", "downloadFileFullPath": "", "downloadFileFullPathUrl": "", "totalDuration": 0, "additionalData": 5, "additionalObject": "Recipe additional object 5", "recipeVersion": 1, "slug": "drink-recipe-title-5-de", "metadata": null, "categories": {}, "areas": {}, "ingredients": {}, "steps": {}, "images": {}
  }, "_objectType": "recipe", "_idFields": [], "_validator": null, "_hasUniqueSlug": false, "id": 6, "title": "Recipe title 1", "description": "Recipe description 1", "amountOfPortions": 1, "author": 1, "parties": "Recipe parties 1", "country": "de", "difficulty": 1, "downloadFileMeta": {
    "downloadFile": "", "downloadFileFullPath": "", "downloadFileFullPathUrl": "", "totalDuration": 0, "additionalData": [], "additionalObject": "Recipe additional object 1", "recipeVersion": 1, "slug": "recipe-title-1-de", "metadata": null, "categories": {}, "createdUserId": 2, "updatedAt": {
      "date": "2011-08-29 17:01:01", "timezone_type": 3, "timezone": "Europe\\Berlin"
    }, "updatedAt": {
      "date": "2011-08-29 17:01:01", "timezone_type": 3, "timezone": "Europe\\Berlin"
    }, "areas": {}, "ingredients": {}, "steps": {}, "images": {}
  }, "_objectType": "recipe", "_idFields": [], "_validator": null, "_hasUniqueSlug": false, "id": 7, "title": "Recipe title 2", "description": "Recipe description 2", "amountOfPortions": 2, "author": 2, "parties": "Recipe parties 2", "country": "de", "difficulty": 2, "downloadFileMeta": {
    "downloadFile": "", "downloadFileFullPath": "", "downloadFileFullPathUrl": "", "totalDuration": 0, "additionalData": [], "additionalObject": "Recipe additional object 2", "recipeVersion": 1, "slug": "recipe-title-2-de", "metadata": null, "categories": {}, "createdUserId": 2, "updatedAt": {
      "date": "2011-08-29 17:01:01", "timezone_type": 3, "timezone": "Europe\\Berlin"
    }, "updatedAt": {
      "date": "2011-08-29 17:01:01", "timezone_type": 3, "timezone": "Europe\\Berlin"
    }, "areas": {}, "ingredients": {}, "steps": {}, "images": {}
  }, "_objectType": "recipe", "_idFields": [], "_validator": null, "_hasUniqueSlug": false, "id": 8, "title": "Recipe title 3", "description": "Recipe description 3", "amountOfPortions": 3, "author": 3, "parties": "Recipe parties 3", "country": "de", "difficulty": 3, "downloadFileMeta": {
    "downloadFile": "", "downloadFileFullPath": "", "downloadFileFullPathUrl": "", "totalDuration": 0, "additionalData": [], "additionalObject": "Recipe additional object 3", "recipeVersion": 1, "slug": "recipe-title-3-de", "metadata": null, "categories": {}, "createdUserId": 2, "updatedAt": {
      "date": "2011-08-29 17:01:01", "timezone_type": 3, "timezone": "Europe\\Berlin"
    }, "updatedAt": {
      "date": "2011-08-29 17:01:01", "timezone_type": 3, "timezone": "Europe\\Berlin"
    }, "areas": {}, "ingredients": {}, "steps": {}, "images": {}
  }
}
```



Generator – other features

- Overhauled validator framework
- Url parameter for redirects after editing
 - Example: link to edit form for a customer's address
 - `&returnTo=userDisplay` – redirect to address display page
 - `&returnTo=adminView` – redirect to admin list of addresses
 - `&returnTo=userCustomerDisplay` – customer display page
- Flexible routing-based short url support
- Client-side validation of unique fields with ajax



Generator – config page

- Modvars can now be modeled in sorted groups
- The generator creates a tabbed panel from that



Settings

[General settings](#) [Downloads](#) [Language](#) [Display](#) [Thumbnails](#) [Emails](#)

Download-related settings.

Use download function ☒

Use download function? (includes download counters)

Max attachment size MB

Is attachment mandatory ☒

Is the upload of the download file mandatory?

Allowed file types

Comma separated list of file extensions, without dots. Leave blank to allow all file types.

Example: *jpg, png*


The following file types are always blocked due to security restrictions: *cgi, pl, asp, phtml, php, php3, php4, php5, exe, com, bat, jsp, cfm, shtml*

Download countdown duration seconds

Read exif data for description ☐

Preset the description with per breaks separated EXIF-data? (for photos)

If this setting is enabled the attachment page in item editing will be the first page, not the second.

 **Update configuration**

 **Cancel**



Documentation

- First draft for a user manual
- Includes validation and generator references
- Available in several versions
 - Help system in the application
 - HTML version (not published yet)
 - PDF document



The screenshot displays the "ModuleStudio User's Guide v0.5.2" web application. The interface includes a search bar at the top left, a navigation sidebar on the left, and a main content area on the right.

Sidebar:

- ModuleStudio User's Guide v0.5.2
 - First steps
 - Enhanced topics
 - Validation
 - Application generator
 - Introduction
 - Basic idea
 - How it works
 - The workflow
 - Generator reference
 - Application layer
 - Language elements
 - Combinations and edge cases
 - Model layer
 - Language elements
 - Entity** (highlighted)
 - Entity field
 - Derived field
 - Calculated field
 - Boolean field
 - Abstract integer field
 - Integer field
 - Decimal field
 - Float field
 - Abstract string field
 - String field

Main Content Area:

- A bulleted list of entity properties:
 - `readOnly` - A boolean specifying whether this entity is read-only or not. If set to `true` editing will not be possible.
 - `slugLength` - Length of slug field. Defaults to 255. An entity is sluggable as soon as at least one of its fields set `sluggable position` to a value greater than 0.
 - `slugSeparator` - Separator which will separate words in slug. Default value is `- like` in Zikula, too.
 - `slugStyle` - Which **slug style** is used.
 - `slugUnique` - A boolean specifying if the slug is unique or not. Default value is `true`.
 - `slugUpdatable` - A boolean specifying if the slug can be changed or not. Default value is `true`.
 - `tree` - Whether and which tree strategy is applied. More information about what the generator creates for trees can be found in the section about **entity tree types**.
 - `standardFields` - A boolean specifying whether the standard fields behaviour is used or not. If set to `true` the entity will get four additional fields for storing the id of the user who created the item, the id of the user who did the last update, as well as the creation and update dates. This information will be included on **display** and **edit** actions.
- A paragraph explaining inheritance: "Normally all created classes are generated twice. Thereby an empty concrete class inherits from an abstract base class containing the whole generator code. The motivation behind this separation is that your own code keeps free from generated artifacts."
- An example code snippet showing class inheritance:

```
class MyModule_Entity_Validator_Base_Person
    extends MyModule_Validator
{
    // generator code
}

class MyModule_Entity_Validator_Person
    extends MyModule_Entity_Validator_Base_Person
{
    // manual code
}
```
- A note on exceptions: "One exception for this scheme is **inheritance**."
- A warning: "Please note that arbitrary *filtering* is not possible at the moment until FilterUtil has been upgraded to support Doctrine 2."
- A concluding sentence: "Whenever you want to change the default implementation you can add corresponding extensions. If you recognise that you are doing the same changes again and again please submit them as patches for the generator."

Footer:

The footer contains the ModuleStudio logo and three columns of links:

- MOST**:
 - Homepage
 - Support
 - Demosite
- MOST Bugtracker**:
 - Generator
 - Dokumentation
 - Übersetzungen
- Zikula**:
 - Homepage
 - Wiki
 - Support



Outlook to 0.5.3

- Bugfixing for tooling and generator
- Some minor generator additions
 - Add map inclusion to Geographical support
 - Create a basic workflow
- Documentation
 - Maybe some help interaction improvements (like active help and context-sensitive help).



Questions?

- Homepage
 - <http://modulestudio.de>
- Github
 - <https://github.com/Guite/MostGenerator>
 - <https://github.com/Guite/MostHelp>
 - <https://github.com/Guite/MostExamples>
- Google Group
 - <http://groups.google.com/group/most-dev/>