pt_extlist
documentation

Table of Contents

1. About the document

This is the first part of every documentation followed by "Reading this".

Table 1. Information about the document

author	Michael Knoll
version	1.0
status	draft
confidentiality	internal
filename	abbr_YYYYMMDD_Vxx
document type	description
start	18.10.2010
last date of editing	19.10.2010
last date of printing	-

disclaimer

publisher: punkt.de

publication: 2010

print-out: 1

person responsible: Lwam Berhane

2. Reading this

A documentation for users, administrators and developers.

The users will learn how to install and setup a demo list. The administrators will learn setting up a list and plugin configuration. The developers will learn what pt_extlist is about and how they can manipulate pt_extlist.

The users and administrators need the basic knowledge about programming. (write more!)

What is the author's intention?

3. Setup.txt

The setup-file includes typoscripts. The included scripts are demolists with which you can develop pt_extlist further.

A Demolist from static countries with typo3 backend and some filters.

3.1. Demolist_MySql_01

This template configures a demolist for use with pt_extlist.

3.2. Demolist_Typo3_01

This template configures a demolist for use with pt_extlist.

3.3. Demolist_Typo3_02

This template configures a demolist for use with pt_extlist.

3.4. Demolist_Extbase_01

This template configures a demolist for use with pt_extlist for rendering ExtBase Domain objects.

3.5. Demolist_Extbase_02

This template configures a demolist for use with pt_extlist for rendering ExtBase Domain objects.

4. Write the title action-orientated according to the target group.

If you want to make the user to do something use this section . Write a short introduction or if necessary a short summary for long instructions. A sentence mustn't be longer than 25 words. Use the same words for the same meaning, be consistent. Use as less specialist terms as possible according to the target group. An instruction can be linked to an example. If you need figures, use a caption. Observe the chronology.

1. Write one step each sentence, at least one predicat and one object. Use the imperative. Use the formal term of address somebody. Avoid passive formulation.
   - Write an alternative, at least one predicat and one object. Use the imperative. Use the formal term of address somebody .Avoid passive formulation.
2. Write one step each sentence, at least one predicat and one object. Use the imperative. Use the formal term of address somebody. Avoid passive formulation.

Caution

If there is a danger for human beings (element warning), you can use "do not...". If there is a danger for the product use a positive formulation. If there is a condition, use the term "when" at the beginning of the sentence.

3. Write one step each sentence, at least one predicat and one object. Use the imperative. Use the formal term of address somebody. Avoid passive formulation.

   --> Write a an interim result. E.g.:Dialog box opens.
4. Write one step each sentence, at least one predicat and one object. Use the imperative. Use the formal term of address somebody. Avoid passive formulation.

   => Write a result.E.g.:You installed program xy.

1. Write one step each sentence, at least one predicat and one object. Use the imperative. Use the formal term of address somebody. Avoid passive formulation.
2. Write one step each sentence, at least one predicat and one object. Use the imperative. Use the formal term of address somebody. Avoid passive formulation.

Warning

If there is a danger for human beings (element warning), you can use "Do not...".

   - Write an alternative, at least one predicat and one object. Use the imperative. Use the formal term of address somebody .Avoid passive formulation.

1. Write one step each sentence, at least one predicat and one object. Use the imperative. Use the formal term of address somebody. Avoid passive formulation.
2. Write one step each sentence, at least one predicat and one object. Use the imperative. Use the formal term of address somebody. Avoid passive formulation.
   - Write an alternative, at least one predicat and one object. Use the imperative. Use the formal term of address somebody .Avoid passive formulation.

4.1. Add subsections, if necessary.

5. example

Example 1.

If you want to illustrate a subject, use this section. This section can be preceded by an instruction or description. Figures are permitted.

```
Write here your code.
```

6. pt_extlist

TypoSript Reference

Name

plugin.tx_ptextlist.settings

plugin.tx_ptextlist.settings

Description

Main TS-key for all pt_extlist settings.

Datatype

Posible Values

Default

StdWrap

Prototype

Example

Child elements

listConfig,

Children of plugin.tx_ptextlist.settings:

Name

listConfig

listConfig

Description

Holds configuration for all list identifiers configured by array key.

Datatype

Associative Array (listIdentifier => listConfiguration)

Posible Values

Default

StdWrap

Prototype

Example

Child elements

[yourListId],

Children of listConfig:

Name

[yourListId]

[yourListId]

Description

Holds configuration for a single list identifier

Datatype

Posible Values

Default

StdWrap

Prototype

Example

```
here comes some sample code
```

Child elements

default, backendConfig, fields, columns, rendererChain, aggregateData, aggregateRows, filters, headerPartial, bodyPartial, agregateRowsPartial,

Children of [yourListId]:

Name

default

default

Description

List default values.

Datatype

Posible Values

Default

StdWrap

Prototype

Example

Child elements
sortingColumn,

Children of default:

Name

sortingColumn

sortingColumn

Description

The default sorting column while no other sortng ist set.

Datatype
String

Posible Values
Any column identifier

Default

StdWrap

Prototype

Example

Name

backendConfig

backendConfig

Description

Holds the configuration for the used data backend.

Datatype

Posible Values

Default

StdWrap

Prototype

Example

Name

fields

fields

Description

Defines raw datasource fields, wich can than combined and processed in to table fields.

Datatype

Associative array

Posible Values

Default

StdWrap

0

Prototype

Example

Child elements

[yourFieldId],

Children of fields:

Name

[yourFieldId]

[yourFieldId]

Description

Named definition of a single data field.

Datatype

String

Posible Values

Default

StdWrap

Prototype

Example

```
name_local {
 table = static_countries
 field = cn_short_local
 isSortable = 1
}
```

Child elements

table, field, special, isSortable, expandGroupRows,

Children of [yourFieldId]:

Name

table

table

Description

Datatype

Posible Values

Default

StdWrap

0

Prototype

Example

Name

field

field

Description

Datatype

Posible Values

Default

StdWrap
0

Prototype

Example

Name

special

special

Description

Insert a individual SQL snippet.

Datatype

Posible Values

Default

StdWrap

0

Prototype

Example

Name

isSortable

isSortable

Description

Datatype

Posible Values

Default

StdWrap
0

Prototype

Example

Name

expandGroupRows

expandGroupRows

Description

Datatype

Posible Values

Default

StdWrap
0

Prototype

Example

Name

columns

columns

Description

Holds the tables column definitions.

Datatype
    Array

Posible Values

Default

StdWrap
    0

Prototype

Example

Child elements
10,20,30,

Children of columns:

Name

10,20,30

10,20,30

Description

Datatype

Posible Values

Default

StdWrap

Prototype

Example

Child elements
fieldIdentifier, label, renderUserFunctions, renderTemplate, renderObj, sorting, sortingImageAsc, sortingImageDesc, sortingImageDefault, accessGroups, cellCSSClass,

Children of 10,20,30:

Name

fieldIdentifier

fieldIdentifier

Description

Datatype

Posible Values

Default

StdWrap
0

Prototype

Example

Name

label

label

Description

A label for this element.

Datatype

String

Posible Values

Default

StdWrap

1

Prototype

Example

Name

renderUserFunctions

renderUserFunctions

Description

A list of userfunctions to render the field value.

Datatype

Posible Values

Default

StdWrap

0

Prototype

Example

Name

renderTemplate

renderTemplate

Description

Datatype

Posible Values

Default

StdWrap
0

Prototype

Example

Name

renderObj

renderObj

Description

Datatype

Posible Values

Default

StdWrap

0

Prototype

Example

Name

sorting

sorting

Description

Datatype

Posible Values

Default

StdWrap

0

Prototype

Example

Name

sortingImageAsc

sortingImageAsc

Description

Datatype

Posible Values

Default

StdWrap
0

Prototype

Example

Name

sortingImageDesc

sortingImageDesc

Description

Datatype

Posible Values

Default

StdWrap
0

Prototype

Example

Name

sortingImageDefault

sortingImageDefault

Description

Datatype

Posible Values

Default

StdWrap
0

Prototype

Example

Name

accessGroups

accessGroups

Description

Datatype

Posible Values

Default

StdWrap
0

Prototype

Example

Name

cellCSSClass

cellCSSClass

Description

Datatype

Posible Values

Default

StdWrap
0

Prototype

Example

Name

rendererChain

rendererChain

Description

Holds the renderer configuration.

Datatype

Posible Values

Default

StdWrap

Prototype
    plugin.tx_ptextlist.prototype.rendererChain

Example

Child elements
enabled, rendererConfigs,

Children of rendererChain:

Name

enabled

enabled

Description

Datatype
Boolean

Posible Values
0,1

Default
1

StdWrap

Prototype

Example

Name

rendererConfigs

rendererConfigs

Description

A list of chained renderer classes that work on list data structures. The default renderer class uses the column configuration, to render the list of field data in a list of rows and columns. By default this renderer iscalled at position 100. All defined renderer before 100 work on a field data list, while renderer after 100 work on a column list data structure.

Datatype

Array (10,20,30)

Posible Values

Default

StdWrap

Prototype

Example

Child elements

[yourNumericRendererId],

Children of rendererConfigs:

Name

[yourNumericRendererId]

[yourNumericRendererId]

Description

Configuration of a single renderer.

Datatype

Posible Values
10,20,30

Default

StdWrap

Prototype

Example

Child elements
renderClassName,

Children of [yourNumericRendererId]:

Name

renderClassName

renderClassName

Description

The class name of the renderers php class.

Datatype

String

Posible Values

Default

StdWrap

Prototype

Example

Name

aggregateData

aggregateData

Description

Defines aggregates of data fields.

Datatype
    Associative array

Posible Values

Default

StdWrap
    0

Prototype

Example

Child elements
[yourAggregateFieldId],

Children of aggregateData:

Name

[yourAggregateFieldId]

[yourAggregateFieldId]

Description

Named definition of a single data field.

Datatype

Associative array

Posible Values

Default

StdWrap

0

Prototype

Example

```
sumPhone {
 fieldIdentifier = phone
 method = sum
 scope = query
}
```

Child elements

method, scope,

Children of [yourAggregateFieldId]:

Name

method

method

Description

Defined aggregate methods.

Datatype
String

Posible Values
min,max,sum,avg

Default

StdWrap
0

Prototype

Example

Name

scope

scope

Description

The scope for the aggregation can be either set to the current page or to the whole query. Aggregates for the current page are calculated internally without an additonal database query.

Datatype
: String

Posible Values
: page,query

Default

StdWrap
: 0

Prototype

Example

special fieldIdentifier

Name

aggregateRows

aggregateRows

Description

Holds the aggregates columns definitions.

Datatype
    Associative array

Posible Values
    All columnIdentifiers

Default

StdWrap

Prototype

Example

Child elements
[yourColumnId],

Children of aggregateRows:

Name

[yourColumnId]

[yourColumnId]

Description

Datatype

Posible Values

Default

StdWrap

Prototype

Example

Child elements
aggregateDataIdentifier,

Children of [yourColumnId]:

Name

aggregateDataIdentifier

aggregateDataIdentifier

Description

Datatype

String

Posible Values

Default

StdWrap

Prototype

Example

Name

filters

filters

Description

Holds all filterbox configurations.

Datatype

Associative array

Posible Values

Default

StdWrap

Prototype

Example

Child elements

[yourFilterBoxId],

Children of filters:

Name

[yourFilterBoxId]

[yourFilterBoxId]

Description

Datatype

Posible Values

Default

StdWrap

Prototype
plugin.tx_ptextlist.prototype.filterBox

Example

Child elements
showReset, showSubmit, filterConfigs,

Children of [yourFilterBoxId]:

Name

showReset

showReset

Description

Show a reset link for all filters of this filterBox.

Datatype
: Boolean

Posible Values
: 0,1

Default
: 1

StdWrap
: 0

Prototype

Example

Name

showSubmit

showSubmit

Description

Show a submit button for this filterBox.

Datatype
Boolean

Posible Values
0,1

Default
1

StdWrap
0

Prototype

Example

Name

filterConfigs

filterConfigs

Description

Holds the configuration of the filters of this filter box.

Datatype

Array

Posible Values

10,20,30...

Default

StdWrap

Prototype

Example

Child elements

[yourNumericFilterId],

Children of filterConfigs:

Name

[yourNumericFilterId]

[yourNumericFilterId]

Description

Datatype

Posible Values

Default

StdWrap

Prototype

Example

Child elements
filterIdentifier, defaultValue, filterClassName, partialPath, invert, invertable,

Children of [yourNumericFilterId]:

Name

filterIdentifier

filterIdentifier

Description

The unique identifier of this filter.

Datatype

String

Posible Values

Default

StdWrap

Prototype

Example

label

Name

defaultValue

defaultValue

Description

The default value which is shown or selected by default.

Datatype

String

Posible Values

Default

StdWrap

Prototype

Example

accessGroups fieldIdentifier

Name

filterClassName

filterClassName

Description

Name of the PHP Class of this filter.

Datatype

String

Posible Values

Default

StdWrap

Prototype

Example

```
Tx_PtExtlist_Domain_Model_Filter_StringFilter
```

Name

partialPath

partialPath

Description

Path to the fluid partial which renders this filter.

Datatype
    String

Posible Values

Default

StdWrap

Prototype

Example

```
Filter/String/StringFilter
```

Name

invert

invert

Description

Invert the constraint of this filter.

Datatype
: Boolean

Posible Values
: 0,1

Default
: 0

StdWrap

Prototype

Example

Name

invertable

invertable

Description

Show a controle to invert this filter.

Datatype
: Boolean

Posible Values
: 0,1

Default
: 0

StdWrap

Prototype

Example

Name

headerPartial

headerPartial

Description

Path to the header partial.

Datatype

String

Posible Values

Default

StdWrap

Prototype

Example

Name

bodyPartial

bodyPartial

Description

Path to the body partial.

Datatype

String

Posible Values

Default

StdWrap

Prototype

Example

Name

agregateRowsPartial

agregateRowsPartial

Description

Path to the aggregate row partial.

Datatype
String

Posible Values

Default

StdWrap

Prototype

Example

7. Glossary

Glossary

E

Extended Markup Language	A set of rules for encoding documents in machine-readable form. It is defined in the XML 1.0 Specification produced by the W3C, and several other related specifications, all gratis open standards.