



Automatic configuration checking for TYPO3 extensions

using oelib

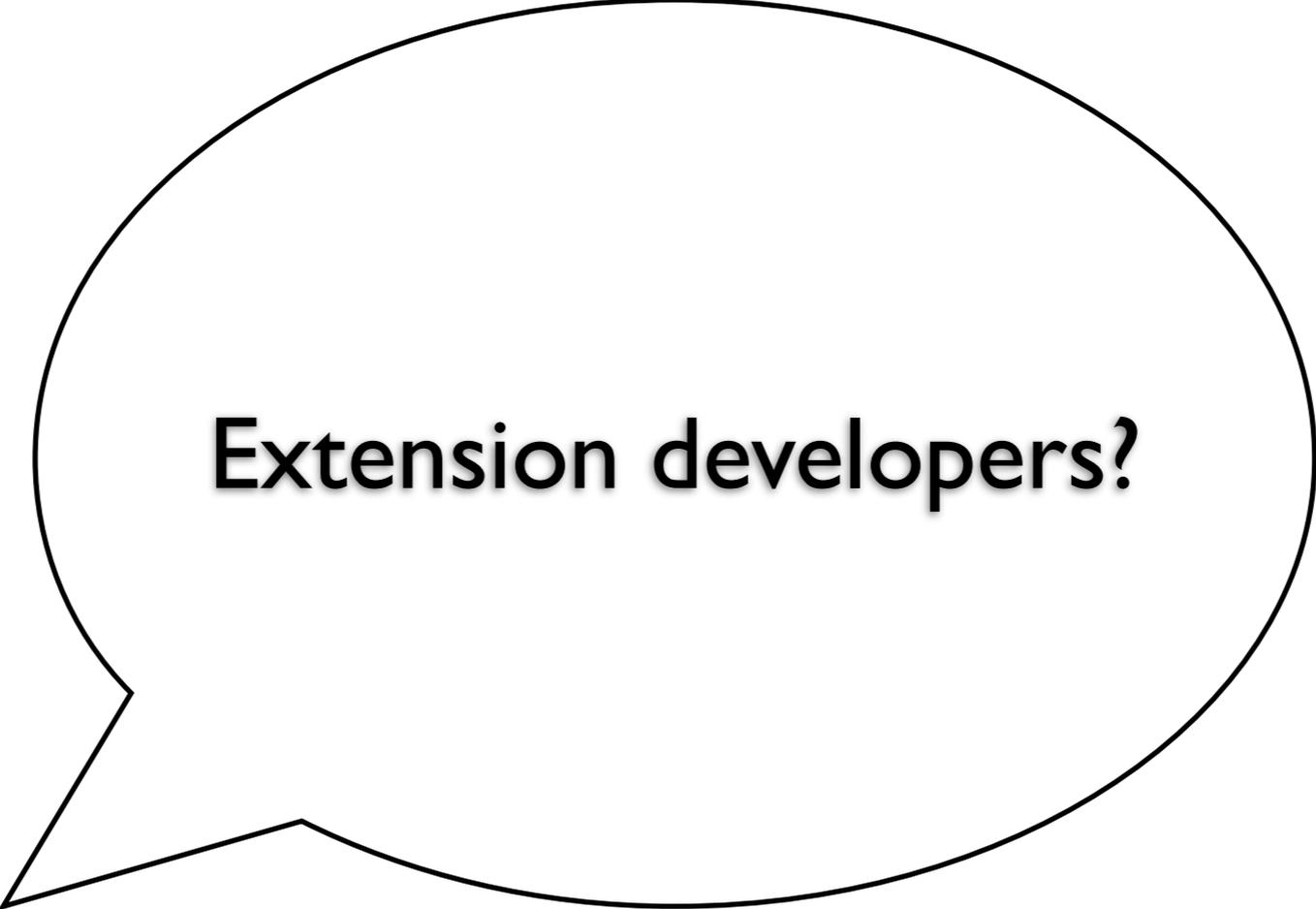
Oliver Klee, T3DD08, 2008-05-10

Schedule

1. Why bother?
2. What does it do?
3. Setting up the basics in your extension
4. A simple check example
5. Different flavors
6. All those different checks ...
7. Future plans



Why bother?



Extension developers?

Why bother?

From: John Doe
To: Paula Developer
Subject: Your extension

Hi,

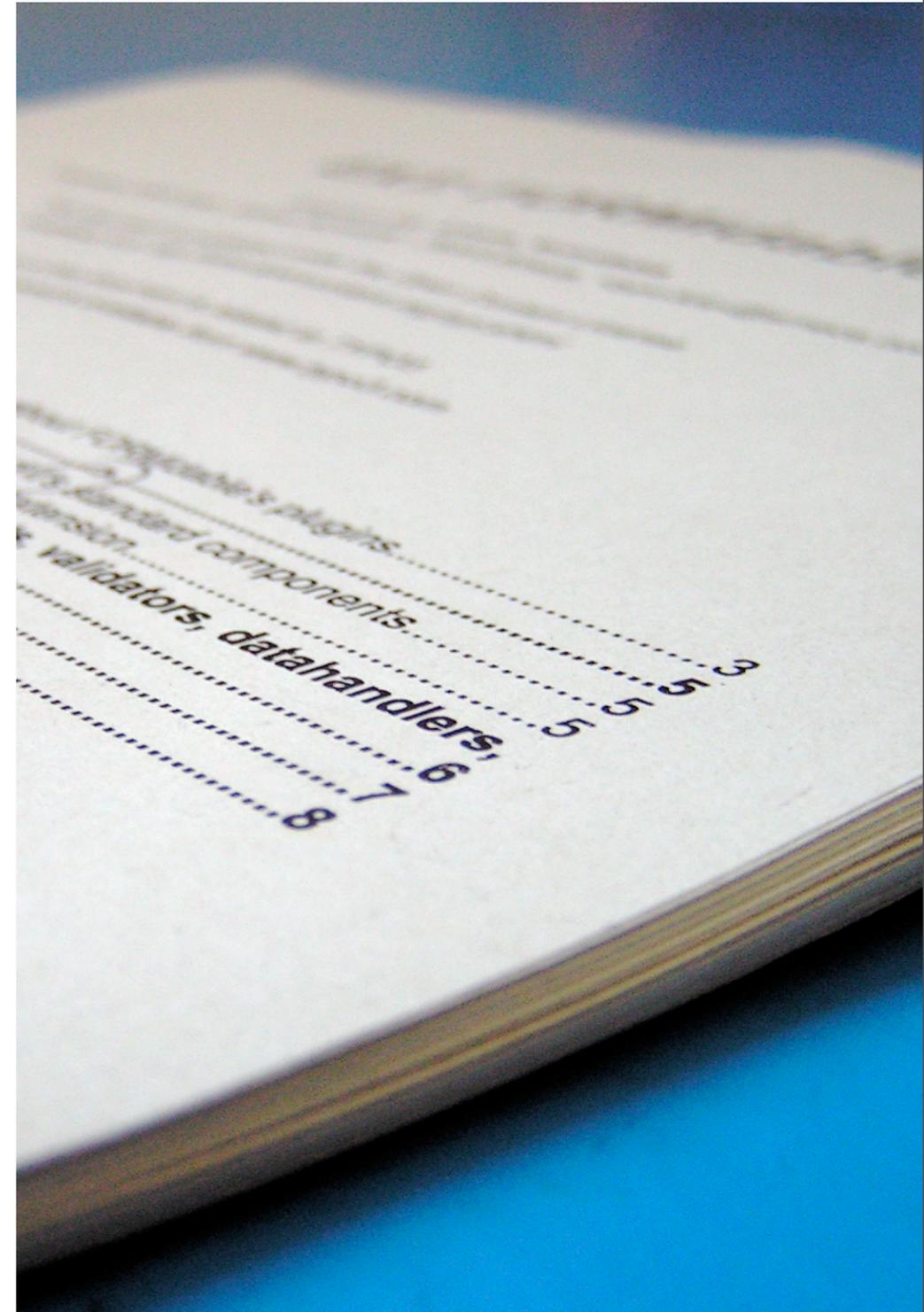
thanks for writing this great extension. :-)

Yet, I have trouble setting up the details view -
some fields don't get displayed. Do you have a
hint for me?

Best regards,

Why bother?

- Users don't read.
- In particular, they don't read manuals.
- But they shouldn't need to bother the extension author for this.



Why bother?

- Errors are human.
- So are configuration mistakes.
- Common mistakes shouldn't cost much time:
 - forgot to include the static template
 - typo in the TS setup
 - mixed the pi1 and pi2 settings ...

Schedule

1. Why bother?
- 2. What does it do?**
3. Setting up the basics in your extension
4. A simple check example
5. Different flavors
6. All those different checks ...
7. Future plans

What does it do?

That's what it looks like:

Configuration check warning:

The TS setup variable **plugin.tx_seminars_pi1.pages** is empty, but needs to be non-empty. This value specifies the system folders that contain the event records for the list view. If this value is not set correctly, some events might not get displayed in the list view. Please correct the TS setup variable **plugin.tx_seminars_pi1.pages** in your TS template setup or via FlexForms.

When that is done, please empty the FE cache and reload this page.

The configuration check for this extension can be disabled in the extension manager.

What does it do?

- checks front-end plugins
- can check TypoScript setup and flexforms
- can check the settings in many ways:
 - numeric ranges
 - corresponds to DB column names
 - in a set of predefined strings
 - and many more ...

What does it do?

- can use different checks for different parts of the extension
- works even when the TS setup for an extension is missing completely
- can be enabled/disabled in the extension manager
- to improve performance
- to avoid warning messages on a live site

What does it do?

- outputs only one warning at a time
- procedure when installing a extension with configuration check:
 1. install the extension
 2. fix the first error and reload the page
 3. rinse and repeat as necessary

Schedule

1. Why bother?
2. What does it do?
- 3. Setting up the basics in your extension**
4. A simple check example
5. Different flavors
6. All those different checks ...

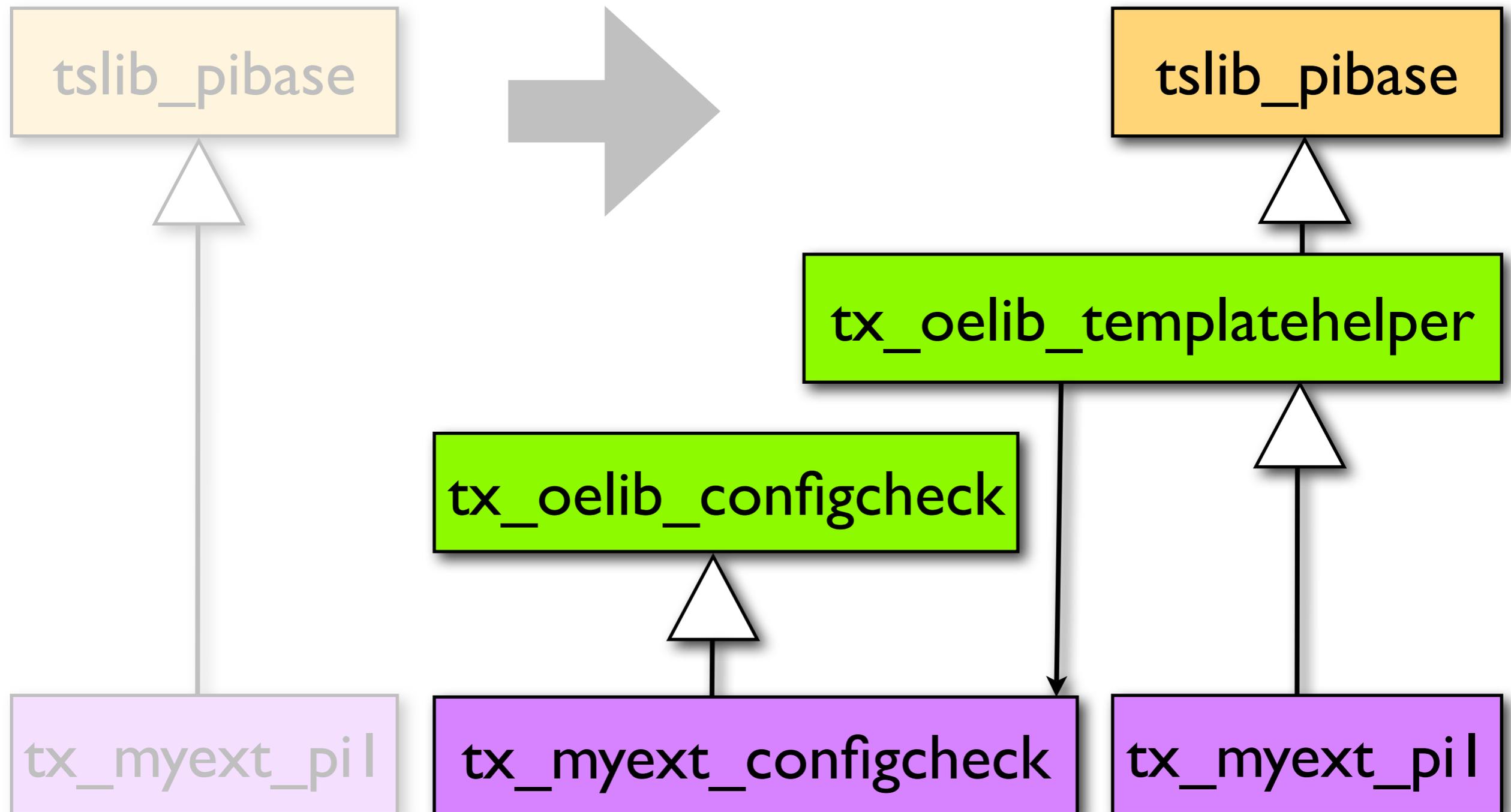
Setting up the basics

I. require oelib in ext_emconf.php

```
'shy' => '',  
'dependencies' => 'cms,oelib',  
'conflicts' => '',
```

```
'version' => '0.0.0',  
'constraints' => array(  
    'depends' => array(  
        'cms' => '',  
        'oelib' => '0.4.0-',  
    ),  
)
```

Setting up the basics



Setting up the basics

2. change the base class in the pi1 class

```
require_once(PATH_tslib.'class.tslib_pibase.php');
```

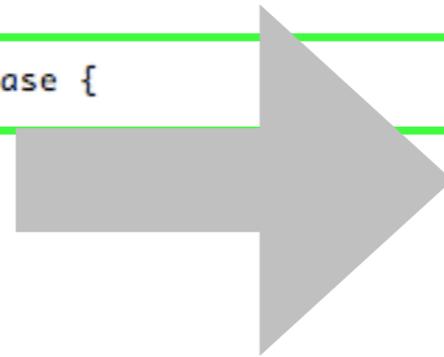
```
/**  
 * Plugin 'Test extension' for the 'myext' extension.  
 *  
 * @author Oliver Klee <typo3-coding@oliverklee.de>  
 * @package TYPO3  
 * @subpackage tx_myext  
 */
```

```
class tx_myext_pi1 extends tslib_pibase {
```

```
require_once(t3lib_extMgm::extPath('oelib').'class.tx_oelib_templatehelper.php');
```

```
/**  
 * Plugin 'Test extension' for the 'myext' extension.  
 *  
 * @author Oliver Klee <typo3-coding@oliverklee.de>  
 * @package TYPO3  
 * @subpackage tx_myext  
 */
```

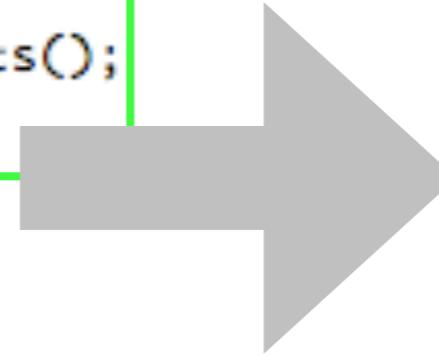
```
class tx_myext_pi1 extends tx_oelib_templatehelper {
```



Setting up the basics

3. use oelib's initialization in pi1::main()

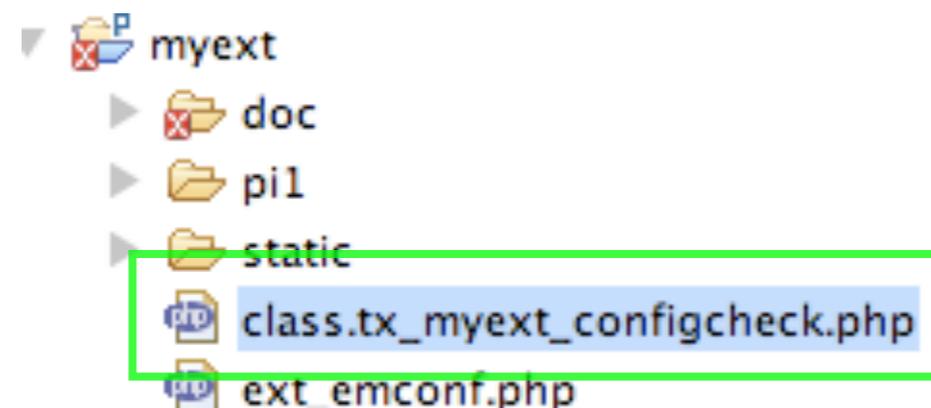
```
function main($content,$conf) {  
    $this->conf=$conf;  
    $this->pi_setPiVarDefaults();  
    $this->pi_loadLL();  
}
```



```
function main($content,$conf) {  
    $this->init($conf);  
}
```

Setting up the basics

4. add a configuration check class



```
require_once(t3lib_extMgm::extPath('oelib').'class.tx_oelib_configcheck.php');

class tx_myext_configcheck extends tx_oelib_configcheck {
    public function check_tx_myext_pi1() {
    }
}
```

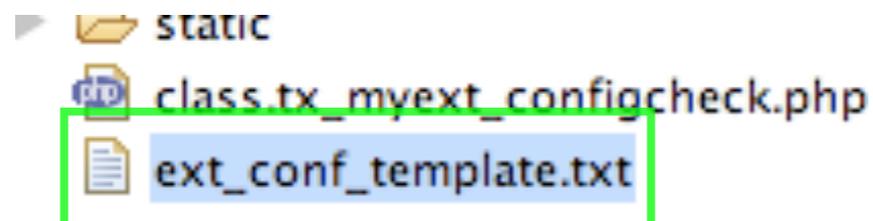
Setting up the basics

5. add a switch in the extension manager

CONFIGURATION:

(Notice: You may need to clear the cache after configu

Configuration check [enableConfigCheck]
Enables the configuration check in the front-end.



```
# cat=basic; type=boolean; label=Configuration check:Enables the configuratio
enableConfigCheck = 1
```

Setting up the basics

6. trigger and output the check in `pi1::main()`

```
$content .= $this->checkConfiguration();  
  
return $this->pi_wrapInBaseClass($content);  
}
```

(The front end does not change yet because we still have no concrete checks.)

Schedule

1. Why bother?
2. What does it do?
3. Setting up the basics in your extension
- 4. A simple check example**
5. Different flavors
6. All those different checks ...
7. Future plans

A simple check example

We want to check:

1. that the static template is included
2. that `.detailPID` (in the TS setup)
 - is set
 - points to an existing FE page

A simple check example

I. Check that the static template is included

- add a token to setup.txt

```
plugin.tx_myext_pi1 {  
    # Do not touch this variable! This is needed for checking whether this stat  
    isStaticTemplateLoaded = 1  
}
```

- check for that token in the config check

```
public function check_tx_myext_pi1() {  
    $this->checkStaticIncluded();  
}
```

A simple check example

I. Check that the static template is included

Configuration check warning:

The static template is not included. This has the effect that important default values do not get set. To fix this, please include this extension's template under *Include static (from extensions)* in your TS template.

When that is done, please empty the FE cache and reload this page.

The configuration check for this extension can be disabled in the extension manager.

- check for that token in the config check

```
public function check_tx_myext_pi1() {  
    $this->checkStaticIncluded();  
}
```

A simple check example

2. check .detailPID

- add it to setup.txt

```
# PID of the detail view page  
detailPID =
```

- in the configuration check

```
public function check_tx_myext_pi1() {  
    $this->checkStaticIncluded();  
    $this->checkDetailViewPid();  
}
```

```
private function checkDetailViewPid() {  
    $this->checkIfSingleFePageNotEmpty(  
        'detailPID',  
        // no flexforms, only setup.txt  
        false,  
        // no flexforms sheet name  
        '',  
        'This value specifies the PID of the page for the detail view. If '  
        .'this value is empty or invalid, the detail view is shown on '  
        .'the same page as the list view.'  
    );  
}
```

A simple check example

2. check .detailPID

Configuration check warning:

The TS setup variable **plugin.tx_myext_pi1.detailPID** is empty, but needs to be non-empty. This value specifies the PID of the page for the detail view. If this value is empty or invalid, the detail view is shown on the same page as the list view. Please correct the TS setup variable **plugin.tx_myext_pi1.detailPID** in your TS template setup.

When that is done, please empty the FE cache and reload this page.

The configuration check for this extension can be disabled in the extension manager.

```
private function checkDetailViewPid() {
    $this->checkIfSingleFePageNotEmpty(
        'detailPID',
        // no flexforms, only setup.txt
        false,
        // no flexforms sheet name
        '',
        'This value specifies the PID of the page for the detail view. If '
        .'this value is empty or invalid, the detail view is shown on '
        .'the same page as the list view.'
    );
}
```

Schedule

1. Why bother?
2. What does it do?
3. Setting up the basics in your extension
4. A simple check example
- 5. Different flavors**
6. All those different checks ...
7. Future plans

Different flavors

Different parts of the extension require different settings:

- a link in the **list view** needs `.detailPID`
- a link in the **detail view** needs `.listPID`
- the **contact form** needs `.defaultRecipient`
- ...

Different flavors

- set a config check flavor in the pi1 class

```
if ($this->piVars['showUid']) {  
    $this->setFlavor('detailView');  
} else {  
    $this->setFlavor('listView');  
}
```

- add flavored checks to the config check

```
public function check_tx_myext_pi1_listView() {  
    $this->checkStaticIncluded();  
    $this->checkDetailViewPid();  
}
```

```
public function check_tx_myext_pi1_detailView() {  
    $this->checkStaticIncluded();  
    $this->checkListViewPid();  
}
```

Schedule

1. Why bother?
2. What does it do?
3. Setting up the basics in your extension
4. A simple check example
5. Different flavors
- 6. All those different checks ...**
7. Future plans

All those different checks ...

Predefined complete checks

- `checkStaticIncluded`
- `checkTemplateFile`
- `checkCssFileFromConstants`
- `checkLocale`

All those different checks ...

Available functions for new checks

- `checkIfSingleInTableNotEmpty`
- `checkIfPidListNotEmpty`
- `checkIfIntegerInRange`
- `checkIfBoolean`
- `checkRegExp`
- ...

Schedule

1. Why bother?
2. What does it do?
3. Setting up the basics in your extension
4. A simple check example
5. Different flavors
6. All those different checks ...
- 7. Future plans**

Future plans

- make the warnings on the front end less intimidating ;-)
- show the total number of warnings
- add a link „show all warnings“
- check extension settings in the EM
- check TScnfig

Schedule

1. Why bother?
2. What does it do?
3. Setting up the basics in your extension
4. A simple check example
5. Different flavors
6. All those different checks ...
7. Future plans

Questions



Thank you!