

TYPO3.Flow - Feature # 43569

Status:	Closed	Priority:	Should have
Author:	Christopher Hlubek	Category:	Error
Created:	2012-12-04	Assigned To:	Bastian Waidelich
Updated:	2014-12-16	Due date:	2013-04-13
PHP Version:			
Has patch:	No		
Complexity:			
Subject:	Exception Handler should respect format		
Description			
The DebugExceptionHandler and ProductionExceptionHandler do not respect other formats than HTML in a meaningful way. It should be possible to get syntactically correct exception / error messages in the most common formats (JSON, XML).			
Related issues:			
related to Base Distribution - Work Package # 45088: Improved REST support		Resolved	2011-05-29 2013-04-13
related to TYPO3.Flow - Feature # 43247: Request respects format		Closed	2012-11-23

History

#1 - 2013-02-22 10:54 - Bastian Waidelich

- Category set to Error
- Status changed from New to Accepted
- Assigned To set to Bastian Waidelich
- Parent task set to #45088

I have a basic prototype working and will try to fix this with #45088

#2 - 2013-02-22 10:57 - Bastian Waidelich

Bastian Waidelich wrote:

| I have a basic prototype working and will try to fix this with #45088

FYI: The current implementation fetches the current request (via RequestHandler->getRequest()) and checks it's format and/or Accept headers to determine a matching rendering format. This does not work on CLI therefore the default should probably be "plaintext".

Do have ideas/resources on how to format exceptions in JSON/XML for best compatibility?

#3 - 2013-03-28 16:21 - Aske Ertmann

- Parent task deleted (#45088)

#4 - 2013-03-28 16:22 - Aske Ertmann

- Parent task set to #45088

#5 - 2013-04-02 13:16 - Bastian Waidelich

- Due date set to 2013-04-06

#6 - 2013-04-02 13:17 - Bastian Waidelich

- Due date changed from 2013-04-06 to 2013-04-13

#7 - 2013-04-24 09:29 - Bastian Waidelich

- Parent task deleted (#45088)

#8 - 2013-04-24 09:31 - Bastian Waidelich

I'm still on this one, but I won't make it within the timeframe of #45088 (which is already over).

If anyone comes up with a good architecture for rendering exceptions according to the Accept header, please share ;)

#9 - 2014-05-13 15:46 - Bastian Waidelich

I still couldn't find a good hypermedia format for errors.. Maybe we could use [application/api-problem+json](#) but that was meant to be used for APIs and there is not XML counterpart it seems.

But probably a simple standard json/xml like

```
1 {
2  "errors": [
3    {
4      "message": "Foo Bar",
5      "code": 1234
6    }
7  ]
8 }
```

would already be much better than the html stacktrace that is currently rendered by default.

See https://www.mnot.net/blog/2013/05/15/http_problem for some more details

#10 - 2014-05-14 10:51 - Christopher Hlubek

Bastian Waidelich wrote:

```
I still couldn't find a good hypermedia format for errors.. Maybe we could use application/api-problem+json but that was meant to be used for APIs
and there is not XML counterpart it seems.
But probably a simple standard json/xml like
[...]
```

```
would already be much better than the html stacktrace that is currently rendered by default.
```

```
See https://www.mnot.net/blog/2013/05/15/http\_problem for some more details
```

Just briefly skipped through the blog article. Regarding the new media type: It makes sense, but for Flow it's too generic because a developer needs to create those URLs for problems and problem instance linking. Also the multiple errors feature is missing, but we handle only a single exception anyway, right? (Except for the planned multi-dispatch HTTP service for Neos)

So I'm in favor of a simple solution, but we should include the reference code to find the logged exception later.

#11 - 2014-05-14 13:03 - Bastian Waidelich

Christopher Hlubek wrote:

Regarding the new media type: It makes sense, but for Flow it's too generic because a developer needs to create those URLs for problems and problem instance linking.

As far as I understand most of the properties are optional. The type is expected to be an **URI**, not necessarily an **URL**. So it could be s.th. like <http://typo3.org/flow/exception-types/<the-type>>.

The instance could be the unique exception code we generate.

*Also the multiple errors feature is missing, but we handle only a single exception anyway, right?
(Except for the planned multi-dispatch HTTP service for Neos)*

True

So I'm in favor of a simple solution, but we should include the reference code to find the logged exception later.

+1

#12 - 2014-12-16 11:33 - Bastian Waidelich

- Status changed from Accepted to Closed

This ticket has been moved to <https://jira.typo3.org/browse/NEOS-108>