

TYPO3.Neos - Task # 26598

Story # 11968 (Closed): Ext JS 4.0 integration

Status:	Rejected	Priority:	Could have
Author:	Christian Müller	Category:	UI
Created:	2011-05-03	Assigned To:	Regine Rosewich
Updated:	2011-05-12	Due date:	
Subject:	Run the current backend code with ExtJS4 in compatibility mode		
Description			
Related issues:			
blocks TYPO3.Neos - Task # 10276: Create ExtJS Theme for Phoenix	Closed	2010-10-15	2010-11-12
blocks TYPO3.Neos - Task # 11545: Make content-changed-number 100% visible al...	Closed		
blocks TYPO3.Neos - Bug # 11654: Fix key press handling in Content Dialog	Rejected	2010-12-22	

History

#1 - 2011-05-03 20:37 - Christian Müller

- Assigned To set to Regine Rosewich

#2 - 2011-05-05 16:29 - Sebastian Kurfuerst

- Project changed from Base Distribution to TYPO3.Neos

#3 - 2011-05-05 16:30 - Sebastian Kurfuerst

- Assigned To deleted (Regine Rosewich)

#4 - 2011-05-05 16:33 - Sebastian Kurfuerst

- Assigned To set to Regine Rosewich

#5 - 2011-05-05 16:36 - Sebastian Kurfuerst

- Status changed from New to Accepted

#6 - 2011-05-10 08:39 - Sebastian Kurfuerst

- Category set to UI

#7 - 2011-05-12 09:45 - Sebastian Kurfuerst

- Status changed from Accepted to Rejected

Regine has spent more than a week trying to get the Phoenix Backend run with Ext JS 4; and there are some considerable problems she discovered:

- as class handling has been completely changed, everything which uses xtype needs to be adjusted.
- the **ref**-Option has been dropped completely, and instead, **ComponentQuery** has to be used. This completely breaks our way of connecting the Backend components, and we'd need some alternative solutions there. We'd especially have to look at a consistent naming convention for component instances there.
- **ComponentQuery has quite some bugs**, which makes it very hard to work with this in certain circumstances.
- **CSS adjustments** or adjustments to the generated markup is **very tedious** at some places; This means we cannot adjust the markup as much as we'd like it to do.

Her suggestions now are the following (and I agree on them):

- **Wait** for bugfix releases; it's for us not yet usable.
- THEN, once a bugfix release is there, do NOT try to port the whole application **at once**, but instead **rebuild the UI** using Ext JS 4 components Top-Down.

However, as we cannot know when a bugfix release is given, we need to implement **new features in current Ext JS 3**, as we cannot have the migration block these tasks.

Regine said that **Sandboxing** (running ExtJS 3 and 4 in parallel) might be an option, which she will check when she has time.

She'll upload her current working state as zip file, so we can have a look at it.

Thanks Regine for your tremendous effort in this regard :-)

Greets,
Sebastian