

Base Distribution - Feature # 31638

Status:	Resolved	Priority:	Must have
Author:	Sebastian Kurfuerst	Category:	
Created:	2011-11-07	Assigned To:	
Updated:	2012-08-28	Due date:	

Subject: Rewrite Rendering Process based on Fluid and TypoScript

Description

The rendering process based on TypoScript and Fluid needs to be rewritten.

Current Rendering Process

Currently, **both** TypoScript and Fluid are involved in rendering. TypoScript controls the rendering while Fluid is just used to render the individual parts. Thus, TypoScript renders the content. The process is as follows:

1. TYPO3\TYPO3\Controller\Frontend\NodeController::show gets a node passed, and triggers render on the view
2. TYPO3\TYPO3\View\TypoScriptView picks up:
 1. it asks the TyposcriptObjectFactory to get a typoscript object name based on the current node type.
 2. then it checks if the current TypoScript tree has a **top-level object** for the which matches the expected TypoScript object name
 3. if yes, this is used. If no, we instantiate a default one.
3. TypoScript Page object renders a Template TypoScript object
 1. For that, it assigns the presentationPropertyNamees to the template, after wrapping them into a proxy object which evaluate **TS**

Processors

2. then, the Node object is assigned to the template
3. Then, the template is rendered (see below), and the result is possibly wrapped again.
4. TYPO3\TYPO3\TypoScript\Template is a Fluid TemplateView and renders the template.
 1. Especially important is the \TYPO3\TYPO3\TypoScript\Section (inheriting from TYPO3\TYPO3\TypoScript\ContentArray and \TYPO3\TypoScript\AbstractContentArrayObject), which creates TypoScript sub-objects for **every** sub node, by asking the \TYPO3\TypoScript\ObjectFactory again. This is how recursion is handled.

Drawbacks of the current rendering process

- very complex
- TypoScriptView is quite inflexible
- recursive rendering creates very many objects, but is not really extensible.

Requirements for revamped rendering

- Nodes can appear multiple times on a page. The rendering must be able to distinguish between these cases.
- When rendering a nodePath, we need to be able to render it completely out-of-context. I.e. render just a single content element being part of a bigger page. This could look somehow like:
.../somePage/someSection/text?typoscriptObjectPath=page.sections.someSection.text
 - This is needed for Varnish, ESI, etc.
 - Thus, the question is **how to address nodes and their rendering information without ambiguity**
- Make Fluid replaceable by other rendering system?? TODO
- TODO: TypoScript objects should be ONLY allowed on FOLDER nodes.
- Bonus: wouldn't it be cool to use TypoScript in combination with Fluid **outside** of Phoenix as well?

Pros and Cons

- **Fluid Template**
 - + easy to understand, easy to write
 - - only replaceable in an all-or-nothing manner
 - - not context dependent; so it is hard to implement things like "in the right column, all images should be 500 px wide maximum"
- **TypoScript:**
 - + adjustable and overridable on a very fine-grained level through use of TypoScript paths
 - + processors are a powerful way to fine-tune rendering.

- + presentationModel enables creating "virtual nodes" which are declared just through TypoScript.
- - hard to write
- - presentationModel is hard to grasp and adds a lot of boilerplate code

New Rendering Process

- **central problem: Addressing.** How do we address stuff like "render node X in context Y"? Y must be identified by a path.
- **central problem: collections.** How do we render a **collection of nodes**? Through an <f:for> loop? Or differently?
- TODO: continue here...

Related issues:

related to TYPO3.Neos - Feature # 11628: Make secondLevelMenu a submenu of cu...	Resolved	
related to TYPO3.Neos - Feature # 31796: Plugins have to be able to set page ...	Closed	2011-11-14
related to TYPO3.Neos - Bug # 26924: StopException in Plugins handled wrong	Closed	2011-05-20
related to TYPO3.TypoScript - Bug # 30097: Content must be renderable using a...	Closed	2011-09-20
blocks TYPO3.Neos - Task # 13544: Add possibility to select a page template	Resolved	

Associated revisions

Revision 9b696c95 - 2012-03-26 18:07 - Sebastian Kurfuerst

[TASK] Add toString() in Node, helpful for debugging

This is part of "completely refactored rendering", and helpful for outputting a node while debugging.

Change-Id: I54f7090ef07958dc8a054738f77b35d620813246

Related: #31638

Revision f6a02eb3 - 2012-03-30 14:46 - Sebastian Kurfuerst

[!!!][FEATURE] Refactored rendering

Completely refactored rendering, including the new TypoScript with lazy evaluation, Eel and FlowQuery.

Change-Id: Ia731e1822fb2d66500d83a523e115ee40cad8a9d

Resolves: #31638

Revision bb2cb76e - 2012-03-30 18:46 - Sebastian Kurfuerst

[!!!][FEATURE] Refactored rendering

Completely refactored rendering, including the new TypoScript with lazy evaluation, Eel and FlowQuery.

Change-Id: I087434541eae8277c1ea2a91e32d8d53a9d03978

Resolves: #31638

Revision 6524f9f3 - 2012-03-30 19:12 - Karsten Dambekalns

[TASK] Add TYPO3.Eel subpackage to distribution

Change-Id: Icf269ba34b0852d69e43933524830cf039579e1

Related: #31638

Revision e06665a0 - 2012-03-31 09:25 - Sebastian Kurfuerst

[BUGFIX] Make unit tests run through again, fix regression with links

Change-Id: I3ca8dfba80f2a929bf9d0c1d7daeccef39a008f0

Related: #31638

History

#1 - 2011-11-14 12:08 - Sebastian Kurfuerst

Had some discussions with Robert about it. Current state:

- We think that **three** trees are relevant for rendering. First, there is the **TYPO3CR Node Tree**, which gives us the nodes which we want to render. Second, the **TypoScript "Prototype Tree"**, which is the TypoScript configuration tree. This states things like "in the 3rd column, all images shall have a max width of 500 px". Third, there is the **Rendering Tree**, which is the instantiated TypoScript Prototype Tree with the TYPO3CR Nodes. This is closely mapped to the resulting page.

- The **root** of the trees are as follows:
 - In the **TYPO3CR Node Tree**, we use the **nearest TYPO3CR:Folder** as the node where the rendering starts
 - We also use this TypoScript configuration for rendering all subnodes.
- the TypoScript **instance tree** does not need to be buildable, as people could do this directly in the template. It's an edge case we do not need to focus on now.
- The TypoScript configuration needs some way to define **prototype inheritance**, and **collections**.
- maybe we can drop the explicit Presentation Model.

#2 - 2011-11-15 13:57 - Christian Müller

Some updates:

We have thought about some concepts and got some impressions from that:

- 1) CSS Selector syntax will not be enough as we need to be able to define NEW structures via Typoscript.
- 2) Prototypes for rendering are generally a good idea, we would need a way to override properties of prototypes in subobjects, so you can change the prototype for different concrete contexts of rendering (left/right section for example)
- 3) we need a way to create a new object from the prototypes via configuration and a way to render specific nodes (for example having a special element that renders the content and a menu of headlines for a page -> we need to render a Menu which is not defined in TYPO3CR + the TYPO3CR content but probably in another place)
- 4) for collection rendering a CSS like syntax would be nice, to have the ability to influence single node rendering (first node should have headline h1, and every third node should have a
 after, something like that - so a nicer syntax for v4 optionSplit-like stuff)

All in all the ideas tend to go in the right direction but the "feel" is not there yet, for know all these parts are too separate and not easy to describe. We need a holistic concept that involves all of the above and with an understandable syntax (second step after defining the overall concept).

#3 - 2011-11-21 12:46 - Christian Müller

TypoScript Object

- Instance of a TypoScript Prototype
- dynamically built at runtime, not declarable in TypoScript syntax

- not addressable in TypoScript

TypoScript Prototype

- Example: "TYPO3.TYPO3:Head", "TYPO3.TYPO3:Body", "TYPO3.TYPO3:Section"
- Blueprints for the runtime TypoScript objects used for rendering
- is not necessarily backed by a PHP class
- implement prototype inheritance: Com.FooBar:Text3 < TYPO3.TYPO3:Text
- with JavaScript semantics (DIFFERENT from v4):
 - Com.FooBar:Text3 < TYPO3.TYPO3:Text
 - TYPO3.TYPO3:Text.property ALSO affects Com.FooBar:Text3 (TODO: Dot Syntax!!)
- every **TypoScript Object** is responsible for rendering a corresponding **TYPO3CR:Node**. This node is resolved by a **Node Resolving Strategy**,

for example:

- getCurrentNode(): "Current" node in the Node Tree (which should be rendered)
- getByNodePath("..."): Fetch node with this path
- node(TYPE, PROPERTIES, SUBNODES): Create runtime node
- null() -- special case of a runtime node

TypoScript Prototype Resolver Strategy

- Answers the question: "I have a TYPO3CR Node, which TypoScript prototype shall I use for rendering this node?" (especially used in TYPO3.TYPO3:Section)
- has much context information available:
 - TYPO3CR Node (of course including context)
 - TypoScript Prototype Hierarchy from the last TYPO3CR:Folder node onwards (example: TYPO3.TYPO3:Page > TYPO3.TYPO3:Body > TYPO3.TYPO3:Section)
- must be exchangeable (Strategy Pattern)
- could implement for example some "CSS Selector like" syntax
- TODO: syntax still remains an open question

TypoScript rendering context

- contains all information to unambiguously render a given Node, under a given TypoScript Prototype hierarchy
- Example: /sites/phoenixdemotypo3org/download/sidebar/textuufas?typoScriptPath = TYPO3.TYPO3:Page > TYPO3.TYPO3:Body > TYPO3.TYPO3:Section > TYPO3.TYPO3:Text
 - Path until folder node: /sites/phoenixdemotypo3org/download
 - Path until content element in TYPO3CR: sidebar/textuufas <- this is returned by "getCurrentNode" in TS Object
 - TypoScript Prototype hierarchy: TYPO3.TYPO3:Page > TYPO3.TYPO3:Body > TYPO3.TYPO3:Section > TYPO3.TYPO3:Text

ROUGH EXAMPLE DEFAULTS FOR PROTOTYPES

```

TYPO3.TYPO3:Head {
  - node = getCurrentNode()
}
TYPO3.TYPO3:Body {
  - node = getCurrentNode()
  sections {
    left = Section
    main = Section
  }
}
TYPO3.TYPO3:Page {
  - node = getCurrentNode()
  head = Head (context = this.context)
  body = Body (context = this.context)
}
TYPO3.TYPO3:Section {
  - node = getCurrentNode()

```

```

}
TYPO3.TYPO3:Text {
  - template = file(...)
  - node = getCurrentNode()
  headlineLevel = 1
}

```

Own Prototypes derived from defaults

```

Com.FooBar:Text3 < TYPO3.TYPO3:Text {
  - template = file(...)
  - node = nodeWithUuid(...)
  headlineLevel = 1
}
Com.FooBar:StaticText < TYPO3.TYPO3:Text {
  - node = node(TYPO3:Text, foo=bar, baz=bar)
}
Com.FooBar:Page1 < TYPO3.TYPO3:Page
Com.FooBar:Page2 < TYPO3.TYPO3:Page

```

Some use cases but no code because the syntax is not there, but all use cases are possible with those concepts:

Use Case 1

2 column content rendering

TYPO3CR Node Hierarchy

- Page
 - Section: left
 - Text 1
 - Text N
 - Section: right
 - Text 1
 - Text N

Use case 2:

2 column content rendering with fixed right column content (right column static, same on all pages)

Var. A: Text in TS defined

Var. B: Text in a "global" text node

TYPO3CR Node Hierarchy

- Page
 - Section: main
 - Text 1
 - Text N

Use Case 3

2 column content rendering with right column content derived from left column (structure/index on right side)

TYPO3CR Node Hierarchy

- Page
 - Section: main
 - Text 1
 - Text N

#4 - 2011-11-23 13:55 - Bastian Waidelich

Just a rough idea for the "selector syntax" to target TS objects in a collection:

We could use a syntax similar to the [jQuery CSS selector syntax](#).

Example:

```
[...]
<node type="TYPO3.TYPO3:Page" nodeName="somepage">
  <properties>
    <title>Some Page</title>
  </properties>
  <node type="TYPO3.TYPO3:Section" nodeName="sidebar">
    <node type="TYPO3.TYPO3:Html" nodeName="text1">
      <properties>
        <source><![CDATA[some text]]></source>
      </properties>
    </node>
  </node>
</node>
```

title of the somepage page:

```
#somepage > properties > title
```

first html node of sidebar section:

```
#sidebar node[type="TYPO3.TYPO3:Html"]:first
```

There are PHP libraries for that (which probably need some adjustments)

#5 - 2012-01-06 10:22 - Sebastian Kurfuerst

TYPO3 v4 Concepts in Relation to TYPO3 Phoenix concepts

CONTENT / RECORDS

Find one or multiple nodes depending on current context, AND TRIGGER RENDERING
-> MISSING: Concept to describe "Collection of Nodes"
-> roughly corresponds to *Node Resolving Strategy*, just without the render triggering

CASE

-> roughly: 'TypeScript Prototype Resolver Strategy'
-> CREATES NEW RUNTIME INSTANCES of TypeScript objects

CONTENT_OBJECT_ARRAY (COA) ??

"Collection of TypeScript objects"

HMENU / TMENU / ...

Menu -> Render "Node Structure"

Content Objects:

FILE / FORM / HTML / TEXT / IMAGE / IMAGERESOURCE / IMAGETEXT / MEDIA / MULTIMEDIA / USER / USER_INT

PAGE / TEMPLATE / FLUIDTEMPLATE

"meta-stuff"

(EDITPANEL) ??

REGISTER??

Consequences

- **CONTENT finds nodes**
- **TYPO3v4** creates TypeScript objects **LAZILY** (i.e. when rendering); Phoenix does it **EAGERLY** --> loss of flexibility

```
foo = CONTENT
foo.resolvingStrategy = childNodes
```

```
foo.renderingConfiguration = CASE
foo.renderingConfiguration.[type] = TYPE ## MUST BE MORE INTELLIGENT, merge configuration ....
```

```
Content::render($renderingContext) {
    NodeCollection $nodes = $this->findNodesSomehow();
    return $nodes->render($this->renderingConfiguration);
}
```

#6 - 2012-01-06 10:26 - Sebastian Kurfuerst

IDEA: We split TypeScript into a TypeScript PARSER which returns one giant array like in v4, and a TypeScript EVALUATOR which LAZILY turns the array into TypeScript objects and renders them.

#7 - 2012-01-06 14:13 - Sebastian Kurfuerst

* "exports.*" -> *exports.page1 = Page* --> then, "page1" is renderable.

```
$typescript = array(
    'page1' => array(
        '__tsObjectName' => 'TYPO3:Page',
        'implementationClassName' => ....
        '__tsDefaults' => array(
            'TYPO3:Page' => array(
                )
            )
        )
    'template' => 'pageTemplate.html',
```

```

'sections' => array(
  '__tsDefaults' => array(
    '__tsObjectName' => 'TYPO3:NodeCollection'
  ),
  'main' => array(
    '__tsObjectName' => 'TYPO3:Text'
  )
)
);

```

```

* $typoScriptString depends on current page....
* TypoScriptRuntime $ts = TypoScriptParser::parse($typoScriptString, $context)
* $ts->render('page1');
** $tsArray['page1'] --> what is the TS Object at this point?
** $pageTsObject = new Page($ts, 'page1');
** // $pageTsObject->setXY... # HERE; processors are evaluated
** return $pageTsObject->render();
*** # In Page
*** $variableContainer->add('context', $context);
*** return $fluidTemplate->render();
*** # Durch ViewHelper
*** $sectionNode = $this->getSection($sectionName); --> Get CR node
*** $ts->pushContext($sectionNode);
*** $sectionContent = $ts->render($currentTsPath . '.sections.' . $sectionName);
*** $ts->popContext();
* $ts->renderPath

```

```

*TypoScript* --> *RenderTree* // Node / ContextSpecific Config

```

```

pageTemplate.html
<html>
<body>
<ts:page.section name="main" />
--oder--
<f:for each="{context.node.sections.main}" as="node">
# Relative by default, prepend "/" for global path
<ts:render path="sections.main" context="{node}" />
</f:for>
</bod>
</html>

```

#8 - 2012-02-20 11:51 - Gerrit Code Review

Patch set 1 for branch **master** has been pushed to the review server.

It is available at <http://review.typo3.org/9119>

#9 - 2012-02-20 11:52 - Gerrit Code Review

Patch set 1 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9120>

#10 - 2012-02-20 14:08 - Gerrit Code Review

Patch set 1 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9122>

#11 - 2012-02-20 14:09 - Gerrit Code Review

Patch set 2 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9120>

#12 - 2012-02-20 14:09 - Gerrit Code Review

Patch set 2 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9119>

#13 - 2012-02-20 14:10 - Gerrit Code Review

Patch set 3 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9119>

#14 - 2012-02-21 06:27 - Gerrit Code Review

Patch set 3 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9120>

#15 - 2012-02-21 06:41 - Gerrit Code Review

Patch set 4 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9120>

#16 - 2012-02-21 06:48 - Gerrit Code Review

Patch set 4 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9119>

#17 - 2012-02-21 06:49 - Gerrit Code Review

Patch set 5 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9120>

#18 - 2012-02-21 08:02 - Gerrit Code Review

Patch set 5 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9119>

#19 - 2012-02-21 08:03 - Gerrit Code Review

Patch set 6 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9120>

#20 - 2012-03-23 06:22 - Gerrit Code Review

Patch set 2 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9122>

#21 - 2012-03-23 06:30 - Gerrit Code Review

Patch set 7 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9120>

#22 - 2012-03-23 06:33 - Gerrit Code Review

Patch set 8 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9120>

#23 - 2012-03-23 06:41 - Gerrit Code Review

Patch set 6 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9119>

#24 - 2012-03-23 10:02 - Gerrit Code Review

Patch set 9 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9120>

#25 - 2012-03-23 10:34 - Gerrit Code Review

Patch set 7 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9119>

#26 - 2012-03-23 10:47 - Gerrit Code Review

Patch set 8 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9119>

#27 - 2012-03-23 11:13 - Gerrit Code Review

Patch set 10 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9120>

#28 - 2012-03-23 11:19 - Gerrit Code Review

Patch set 9 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9119>

#29 - 2012-03-26 12:05 - Gerrit Code Review

Patch set 10 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9119>

#30 - 2012-03-26 12:06 - Gerrit Code Review

Patch set 11 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9120>

#31 - 2012-03-26 12:52 - Gerrit Code Review

Patch set 12 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9120>

#32 - 2012-03-26 13:46 - Gerrit Code Review

Patch set 11 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9119>

#33 - 2012-03-26 13:46 - Gerrit Code Review

Patch set 13 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9120>

#34 - 2012-03-26 13:59 - Gerrit Code Review

Patch set 14 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9120>

#35 - 2012-03-26 14:08 - Gerrit Code Review

Patch set 15 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9120>

#36 - 2012-03-26 14:10 - Gerrit Code Review

Patch set 16 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9120>

#37 - 2012-03-26 14:39 - Gerrit Code Review

Patch set 17 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9120>

#38 - 2012-03-26 16:31 - Gerrit Code Review

Patch set 18 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9120>

#39 - 2012-03-26 16:32 - Gerrit Code Review

Patch set 12 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9119>

#40 - 2012-03-26 16:32 - Gerrit Code Review

Patch set 19 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9120>

#41 - 2012-03-30 13:53 - Gerrit Code Review

Patch set 13 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9119>

#42 - 2012-03-30 14:03 - Gerrit Code Review

Patch set 14 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9119>

#43 - 2012-03-30 14:04 - Gerrit Code Review

Patch set 20 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9120>

#44 - 2012-03-30 14:43 - Gerrit Code Review

Patch set 21 for branch **master** has been pushed to the review server.
It is available at <http://review.typo3.org/9120>

#45 - 2012-03-30 14:46 - Gerrit Code Review

Patch set 15 for branch **master** has been pushed to the review server.

It is available at <http://review.typo3.org/9119>

#46 - 2012-03-30 18:47 - Gerrit Code Review

Patch set 22 for branch **master** has been pushed to the review server.

It is available at <http://review.typo3.org/9120>

#47 - 2012-03-31 02:39 - Sebastian Kurfuerst

- *Status changed from New to Resolved*

- *% Done changed from 0 to 100*

Applied in changeset commit:f6a02eb32c5caeb68428ac4da8583703f9f00098.

#48 - 2012-08-28 11:41 - Robert Lemke

- *Tracker changed from Major Feature to Feature*