

GSoC 2011: Static File Caching - Feature #27111

Major Feature # 26892 (Needs Feedback): Implement light weight frontend processing (TSFE light)

Implement object proxies and light weight TSFE

2011-05-28 14:10 - Oliver Hader

Status:	New	Start date:	2011-05-28
Priority:	Should have	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:	Mid of term		
Description			
There are several expensive objects in TSFE that are called directly by eg. \$GLOBALS['TSFE']->tmpl->whateverMethod(); Example/Idea: Create light object proxies for those objects - only if something is called/pushed/fetched from that object, the real instance will be created and used. The problems are in the correct initialization/bootstrap of those objects that needs to be done in after creating the instances as well. @class tslib_fe_light extends tslib_fe { /** * Defines an object proxy that will load the real object when it's required. * @var tx_eft_typo3_objectProxy */ public \$csConvObj; /** * Defines an object proxy that will load the real object when it's required. * @var tx_eft_typo3_objectProxy */ public \$sys_page; /** * Defines an object proxy that will load the real object when it's required. * @var tx_eft_typo3_objectProxy */ public \$tmpl; /** * Configuration array with TypoScript [config.] * @var array */ public \$config = array(); public function __construct() { \$this->csConvObj = t3lib_div::makeInstance('t3lib_utility_objectProxy', \$this, 't3lib_cs', PATH_t3lib . 'class.t3lib_cs.php'); } } class t3lib_utility_objectProxy { protected \$parent; protected \$className; protected \$classFile; protected \$initializeCallback; public function _get()/_set()__call() { // initialize if something shall be done with the object } } @			

History

#1 - 2011-05-28 14:10 - Oliver Hader

- Target version set to Mid of term