

Inhalt

Teil I – Klassische Extensionprogrammierung

1	Grundlagen.....	3
1.1	Aufbau und Struktur von TYPO3	3
1.2	Aufbau einer Extension.....	6
1.3	Arten und Einteilung von Extensions.....	7
1.4	Der Extension-Key.....	8
1.5	TYPO3 Coding Guidelines	9
1.5.1	Namespace	10
1.5.2	PHP-Regeln.....	11
1.5.3	Dateistruktur.....	12
1.5.4	PHP-Syntax	15
1.5.5	phpDoc	21
1.5.6	Changelog.....	22
2	Eine Extension planen und entwickeln	23
2.1	Extensions planen	23
2.1.1	Wir planen eine Blog-Extension.....	24
2.2	Der Kickstarter.....	26
2.2.1	Die Datenbank.....	28
2.2.2	Datenbanktabellen und Formulare – eine Erklärung	34
2.2.3	Extend existing Tables	36
2.2.4	Frontend-Plug-ins.....	37
2.2.5	Backend-Module	40
2.2.6	Integrate in existing Modules	41
2.2.7	Clickmenu items.....	41
2.2.8	Services	42
2.2.9	Static TypoScript code	43

2.2.10	TSconfig.....	43
2.2.11	Die Extension erstellen.....	44
2.3	Extension-Konfiguration mit ext_emconf.php.....	45
2.4	Die weiteren Dateien der Extension.....	50
2.4.1	Verzeichnis doc.....	50
2.4.2	Verzeichnis mod#.....	50
2.4.3	Verzeichnis modfunc#.....	51
2.4.4	Verzeichnis pi#.....	52
2.4.5	Verzeichnis Static.....	52
2.4.6	Datei Changelog.....	53
2.4.7	Datei README.txt.....	53
2.4.8	Datei ext_icon.gif.....	53
2.4.9	Datei ext_localconf.php.....	53
2.4.10	Datei ext_tables.php.....	53
2.4.11	Datei ext_tables.sql.....	54
2.4.12	Datei ext_tables_static+adt.sql.....	55
2.4.13	Dateien Icon_tx_extkey_tabellenname.gif.....	56
2.4.14	Datei locallang.xml.....	56
2.4.15	Datei locallang_db.xml.....	56
2.4.16	Datei tca.php.....	56
3	Backend-Module erstellen	65
3.1	Module konfigurieren und im Hauptmenü anordnen.....	65
3.2	Einen eigenen Navigationsbaum erstellen.....	69
3.3	Backend-Modul erstellen.....	75
3.3.1	Das Modul initialisieren.....	75
3.3.2	Den Modulinhalt aufbauen.....	79
3.3.3	Extension-Konfiguration über ext_conf_template.txt.....	80
3.3.4	Backend-Formulare aufrufen.....	82
3.3.5	Eine Datensatzliste mit Bearbeitungsfunktionen erstellen.....	84
3.3.6	Aufgabe: Module für Kommentare, Blog und Blogroll.....	90
3.4	Backend User Object – \$BE_USER.....	91
3.5	Das Systemlog.....	92
4	Frontend-Plug-ins.....	95
4.1	Das Frontend-Plug-in im Überblick.....	95
4.2	Feldinhalte für die Ausgabe aufbereiten.....	103
4.2.1	Das Bild für Teaser Image.....	104
4.2.2	Ausgabe des Haupttextes in der Listenansicht, wenn erwünscht.....	105
4.2.3	Haupttext mit RTE formatieren.....	105
4.2.4	Haupttext statt Teasertext anzeigen, falls dieser nicht verfügbar ist.....	106
4.2.5	Verlinken des Autorennamens mit der Mail-Adresse des Autors.....	106
4.2.6	Kategorien als Namen anzeigen und verlinken (Relationen).....	107
4.3	HTML-Templates für das Plug-in.....	109
4.4	Die Kommentarfunktion mit AJAX und eID.....	114
4.4.1	Ausgabe vorhandener Kommentare.....	115

4.4.2	Das Kommentarformular hinzufügen	118
4.5	Was sind Hooks?	123
4.6	Flexforms	125
4.6.1	Erzeugen der Flexform	126
4.7	Das Plug-in pi2	132
5	Dokumentation und Abschluß	133
5.1	Code dokumentieren mit extdeval	133
5.2	Extension-Dokumentation schreiben	134
5.2.1	Die Vorlage	135
5.2.2	Der Aufbau der Dokumentation	135
5.2.3	Vorlagenstile	136
5.2.4	Die Dokumentation verfassen	136
5.3	Upload ins TER	137

Teil II – Funktionsreferenz zur klassischen Extensionprogrammierung

6	Datenbank	141
6.1	\$GLOBALS['TYPO3_DB']	141
6.1.1	exec_SELECTquery	141
6.1.2	exec_SELECT_queryArray	141
6.1.3	exec_SELECTgetRows	142
6.1.4	exec_SELECT_mm_query	142
6.1.5	exec_INSERTquery	143
6.1.6	exec_UPDATEquery	143
6.1.7	exec_DELETEquery	143
6.1.8	sql_fetch_assoc	143
6.1.9	sql_fetch_row	144
6.1.10	searchQuery	144
6.1.11	listQuery	144
6.1.12	splitGroupOrderLimit	145
6.1.13	quoteStr	145
6.1.14	fullQuoteStr	145
6.1.15	fullQuoteArray	146
6.1.16	escapeStrForLike	146
6.1.17	cleanIntArray	147
6.1.18	cleanIntList	147
6.1.19	debug_lastBuiltQuery	147
6.2	tslib_pibase	148
6.2.1	pi_exec_query	148
6.2.2	pi_getPidList	149
6.2.3	pi_getRecord	150
6.2.4	pi_prependFieldsWithTable	150

6.3	cObj	151
6.3.1	DBgetDelete	151
6.3.2	DBgetInsert	151
6.3.3	DBgetUpdate	152
6.3.4	enableFields	153
7	Dateisystem.....	155
7.1	t3lib_div	155
7.1.1	dirname	155
7.1.2	fixWindowsFilePath	155
7.1.3	formatSize	156
7.1.4	get_dirs	156
7.1.5	getAllFilesAndFoldersInPath	157
7.1.6	getFileAbsFileName	158
7.1.7	getFilesInDir	158
7.1.8	getURL	159
7.1.9	isAbsPath	159
7.1.10	isAllowedAbsPath	159
7.1.11	mkdir	160
7.1.12	mkdir_deep	160
7.1.13	removePrefixPathFromList	160
7.1.14	resolveBackPath	161
7.1.15	rmdir	161
7.1.16	split_fileref	162
7.1.17	tempnam	162
7.1.18	unlink_tempfile	162
7.1.19	upload_copy_move	162
7.1.20	upload_to_tempfile	163
7.1.21	verifyFilenameAgainstDenyPattern	163
7.1.22	writeFile	164
7.1.23	writeFileToTypo3tempDir	164
8	Strings, Arrays und Umgebungsvariablen	165
8.1	Allgemeine Funktionen	165
8.1.1	_GET	165
8.1.2	_GETset	165
8.1.3	_GP	166
8.1.4	_POST	166
8.1.5	callUserFunction	166
8.1.6	clientInfo	167
8.1.7	compat_version	167
8.1.8	compileSelectedGetVarsFromArray	168
8.1.9	getHostname	168
8.1.10	getIndpEnv	168
8.1.11	getThisUrl	168
8.1.12	linkThisScript	169

8.1.13	linkThisUrl	169
8.1.14	locationHeaderUrl	169
8.1.15	makeInstance	170
8.1.16	makeInstanceService	170
8.1.17	rmFromList	170
8.1.18	sysLog	171
8.2	String-Funktionen	171
8.2.1	calcParenthesis	171
8.2.2	cmpFQDN	171
8.2.3	cmpIP	172
8.2.4	convUmlauts	172
8.2.5	csvValues	172
8.2.6	deHSCentities	173
8.2.7	expandList	173
8.2.8	fixed_lgd	173
8.2.9	fixed_lgd_pre	174
8.2.10	formatForTextarea	174
8.2.11	generateRandomBytes	174
8.2.12	get_tag_attributes	175
8.2.13	htmlspecialchars_decode	175
8.2.14	implodeArrayForUrl	175
8.2.15	implodeAttributes	176
8.2.16	inList	176
8.2.17	int_from_ver	177
8.2.18	intInRange	177
8.2.19	intval_positive	178
8.2.20	isFirstPartOfStr	178
8.2.21	md5int	178
8.2.22	milliseconds	179
8.2.23	modifyHTMLColor	179
8.2.24	modifyHTMLColorAll	179
8.2.25	normalizeIPv6	180
8.2.26	removeXSS	180
8.2.27	revExplode	180
8.2.28	rm_endcomma	181
8.2.29	shortMD5	181
8.2.30	split_tag_attributes	181
8.2.31	splitCalc	182
8.2.32	strtolower	183
8.2.33	strtoupper	183
8.2.34	substUrlsInPlainText	183
8.2.35	testInt	184
8.2.36	trimExplode	184
8.2.37	uniqueList	185
8.2.38	validEmail	185
8.2.39	validIP	185

8.2.40	validIPv4.....	186
8.2.41	validIPv6.....	186
8.3	Array-Funktionen.....	186
8.3.1	addSlashesOnArray.....	186
8.3.2	array_merge.....	187
8.3.3	array_merge_recursive_overrule.....	187
8.3.4	array2json.....	188
8.3.5	array2xml.....	188
8.3.6	arrayToLogString.....	189
8.3.7	explodeUrl2Array.....	190
8.3.8	inArray.....	190
8.3.9	print_array.....	190
8.3.10	removeArrayEntryByValue.....	191
8.3.11	slashArray.....	191
8.3.12	view_array.....	192
8.3.13	xml2array.....	192
8.3.14	xml2tree.....	193
8.3.15	xmlGetHeaderAttribs.....	194
8.4	String-Funktionen in Frontend-Plug-ins.....	195
8.4.1	calcAge.....	195
8.4.2	checkEmail.....	195
8.4.3	codeString.....	195
8.4.4	encryptEmail.....	196
8.4.5	HTMLcaseshift.....	196
8.4.6	keywords.....	196
8.4.7	linebreaks.....	197
8.4.8	processParams.....	197
8.4.9	uniqueHash.....	198
8.4.10	URLqMark.....	198
9	Bilder.....	199
9.1	Einbinden.....	199
9.2	Bearbeiten.....	200
9.2.1	Bild umrechnen (fürs Web optimieren).....	200
9.2.2	Abmessungen ändern.....	201
9.2.3	Zuschneiden.....	201
9.2.4	Graustufen.....	202
9.3	Erzeugen.....	203
9.3.1	Bild in bestehendes Bild einfügen.....	203
9.3.2	Text erzeugen.....	204
10	AJAX im Frontend.....	205
10.1	eID.....	205
10.1.1	Konfiguration.....	205
10.1.2	Basisdatei.....	206

10.2	Zusammenspiel AJAX & eID	206
10.2.1	Aufbau der XML-Response.....	206
10.2.2	AJAX-Request und Verarbeitung.....	207
11	Frontend-Plug-ins	209
11.1	Cache	209
11.1.1	cacheExpires.....	209
11.1.2	clearPageCacheContent.....	209
11.1.3	clearPageCacheContent_pidList.....	209
11.1.4	get_cache_timeout.....	210
11.1.5	set_cache_timeout_default	210
11.1.6	set_no_cache	210
11.2	Content-Elemente.....	211
11.2.1	cImage	211
11.2.2	cleanFormName	211
11.2.3	cObjGet	211
11.2.4	cObjGetSingle	212
11.2.5	currentPageUrl.....	212
11.2.6	fileResource.....	213
11.2.7	getImgResource.....	213
11.2.8	getSlidePids	213
11.2.9	gifBuilderTextBox.....	214
11.2.10	stdWrap	214
11.3	Links	215
11.3.1	baseUrlWrap.....	215
11.3.2	getMailTo	215
11.3.3	getTypoLink	216
11.3.4	getTypoLink_URL	216
11.3.5	http_makelinks	217
11.3.6	imageLinkWrap.....	217
11.3.7	mailto_makelinks	218
11.3.8	pi_getPageLink.....	218
11.3.9	pi_linkToPage	218
11.3.10	pi_linkTP	219
11.3.11	pi_linkTP_keepPIvars	219
11.3.12	pi_linkTP_keepPIvars_url	219
11.3.13	pi_openAtagHrefInJSwindow	220
11.3.14	prefixLocalAnchorsWithScript.....	220
11.4	Listen	220
11.4.1	pi_list_browserresults	221
11.4.2	pi_list_linkSingle.....	222
11.4.3	pi_list_makelist	222
11.4.4	pi_list_modeSelector	223
11.4.5	pi_list_searchBox	223

11.5	JavaScript.....	224
11.5.1	additionalHeaderData.....	224
11.5.2	additionalJavaScript.....	224
11.5.3	JSeventFuncCalls.....	225
11.5.4	minifyJavaScript.....	225
11.5.5	quoteJSvalue.....	226
11.5.6	rawUrlEncodeJS.....	226
11.5.7	setJS.....	227
11.5.8	wrapJS.....	227
11.6	CSS.....	228
11.6.1	additionalCSS.....	228
11.6.2	additionalHeaderData.....	228
11.6.3	pi_getClassName.....	229
11.6.4	pi_setClassStyle.....	229
11.6.5	setCSS.....	229
11.7	Mehrsprachigkeit.....	230
11.7.1	getLLL.....	230
11.7.2	pi_getLL.....	230
11.7.3	readLLfile.....	231
11.8	Umgebungsvariablen und allgemeine Funktionen.....	232
11.8.1	absRefPrefix.....	232
11.8.2	additionalHeaderData.....	232
11.8.3	all.....	232
11.8.4	anchorPrefix.....	233
11.8.5	applicationData.....	233
11.8.6	ATagParams.....	233
11.8.7	baseUrl.....	234
11.8.8	baseUrlWrap.....	234
11.8.9	beUserLogin.....	234
11.8.10	clientInfo.....	234
11.8.11	content.....	235
11.8.12	defaultBodyTag.....	235
11.8.13	domainStartPage.....	235
11.8.14	extTarget.....	236
11.8.15	fePreview.....	236
11.8.16	id.....	236
11.8.17	imagesOnPage.....	237
11.8.18	intTarget.....	237
11.8.19	lang.....	237
11.8.20	lastImageInfo.....	238
11.8.21	loginUser.....	238
11.8.22	no_cache.....	238
11.8.23	page.....	239
11.8.24	printError.....	239
11.8.25	rootLine.....	240
11.8.26	siteScript.....	240

11.8.27	tmpl->config, tmpl->setup	241
11.8.28	type	241
11.8.29	TYPO3_CONF_VARS	241
11.8.30	uniqueString	242
11.9	Konfiguration mit Flexforms	242
11.9.1	Erstellen einer Plug-in_Flexform	242
11.9.2	Auslesen der Formulardaten	243
11.10	Konfiguration mit TypoScript	244
11.11	Konfiguration im Extension-Manager	244
12	Frontend-User & Sessions	245
12.1	Frontend-User	245
12.1.1	Daten des aktuell angemeldeten Users	245
12.2	Sessions	246
12.2.1	Daten speichern	246
12.2.2	Daten auslesen	246
12.2.3	Warenkorb aufbauen	246
13	Backend & Services	249
13.1	Eigene Flexforms	249
13.1.1	Mehrsprachigkeit	250
13.1.2	Tabs	251
13.1.3	Elemente	252
13.1.4	Flexform-Daten im Frontend	256
13.2	Backend-Module	257
13.2.1	Dokumententypen für \$this->doc	257
13.2.2	TypoScript einer Seite auslesen	257
13.2.3	AJAX im Backend	257
13.3	Services	259
13.4	cli	259
13.5	Debugging	260
13.5.1	debug	260
13.5.2	debug_ordvalue	261
13.5.3	debug_trail	261
13.5.4	debugRows	261
13.5.5	devLog	262

Teil III – Extbase und Fluid

14	Der neue Weg der Extension-Programmierung	265
14.1	Grundlagen der objektorientierten Programmierung	267
14.1.1	Klassen und Objekte	267
14.1.2	Vererbung von Klassen	271

14.1.3	Kontrollierte Vererbung – abstrakte und finale Klassen.....	273
14.1.4	Sichtbarkeiten: public, private und protected	274
14.1.5	Interfaces.....	275
14.1.6	Type Hints.....	277
14.1.7	Statische Methoden und Eigenschaften	277
14.1.8	Namespaces.....	278
14.2	Neue Konzepte.....	279
14.2.1	Domain Driven Design.....	280
14.2.2	Model-View-Controller.....	281
14.2.3	Modellierung.....	282
14.2.4	Gemeinsamens Vokabular – Ubiquitous Language.....	282
14.2.5	Die Elemente des Modells.....	285
15	Ein Rundgang durchs System.....	287
15.1	Installation von Extbase und Fluid.....	287
15.2	Installation der Extension blog_example	289
15.3	Datenstruktur im Blog-Example	291
15.4	Namenskonventionen.....	293
15.4.1	Verzeichnis- und Dateinamen	294
15.4.2	Klassennamen	294
15.4.3	Controller und Actions	295
15.5	Die Dateistruktur im Extension-Root-Verzeichnis	295
15.5.1	Die Datei ext_emconf.php.....	296
15.5.2	Die Datei ext_tables.php	297
15.5.3	Die Datei ext_localconf.php.....	301
15.5.4	Die Dateien ext_tables.sql und ext_icon.gif.....	302
15.6	Die Verzeichnisstruktur	302
15.6.1	Das Verzeichnis Classes.....	303
15.6.2	Das Verzeichnis Configuration	304
15.6.3	Das Verzeichnis Module	305
15.6.4	Das Verzeichnis Resources	305
15.6.5	Weitere Verzeichnisse.....	305
15.7	Der prinzipielle Ablauf	305
15.7.1	Aufruf des Dispatchers (Schritt 1).....	306
15.7.2	Aufruf des Controllers (Schritt 2).....	309
15.7.3	Ansprechen des Repositories (Schritt 3).....	309
15.7.4	Zurückliefern der Blog-Objekte (Schritt 4).....	309
15.7.5	Das Objekt wird dem View zugeordnet (Schritt 5)	311
15.7.6	Rückgabe der Template-Ausgabe an den Controller (Schritt 6).....	313
15.7.7	Rückgabe der Ausgabe an den Dispatcher (Schritt 7).....	314
15.7.8	Rückgabe der Ausgabe an TYPO3 (Schritt 8).....	314
16	Entwicklung eines eigenen Plug-ins.....	315
16.1	Aufbau einer minimal funktionstüchtigen Extension.....	315
16.1.1	Extension-Key.....	315
16.1.2	Verzeichnisse anlegen.....	316

16.1.3	Die Datei ext_emconf.php	316
16.1.4	Die Dateien ext_localconf.php und ext_tables.php	317
16.1.5	Einrichten eines Standard-Controllers	318
16.1.6	Installieren der Extension und das erste Erfolgserlebnis	319
16.2	Hinzufügen eines Views	320
16.3	Entitäten einführen	322
16.4	Daten persistieren (Datenspeicher)	324
16.4.1	Datenbankstruktur – die Datei ext_tables.sql	325
16.4.2	Datenbankstruktur – das TCA	329
16.5	Anlegen eines Repositorys	333
16.6	Anlegen einer neuen Action	335
16.6.1	Anlegen der Action addAction()	336
16.7	Daten per Formular eingeben und auswerten	337
16.8	Objekte aus dem Repository entfernen	338
16.8.1	Zufügen der Action delete in der Konfiguration	338
16.8.2	Anpassen der Template-Datei index.html	338
16.9	Update von Objekten	339
16.9.1	Edit- und Update-Action hinzufügen	339
16.10	Der Query-Manager	341
16.11	Eingabevalidierung	343
16.12	Validatoren	343
16.12.1	Vordefinierte Validatoren	343
16.12.2	Eigene Validatoren	344
16.12.3	Ausgabe der Fehler	347
16.12.4	Optionale Argumente	347
16.13	Relationen zu anderen Tabellen	348
16.13.1	Erweiterung des TCA	348
16.13.2	Die Domain-Klasse Tx_Simpleblog_Domain_Model_Post	349
16.13.3	Registrieren der Actions in der Datei ext_localconf.php	350
16.13.4	Erstellung des Post-Controllers	350
16.13.5	Neue Templates und Template-Änderungen	353
16.13.6	Ändern der Blog-Identity	354
16.13.7	Aufruf der Extension im Frontend	356
16.14	Relationen zu anderen Tabellen m:n	356
16.15	Mehrsprachigkeit zufügen	361
16.16	Konfiguration mittels TypoScript	363
16.17	Backend-Module mit Extbase	365
16.18	Der Extbase-Kickstarter	366
16.18.1	Installation des Extbase-Kickstarters	367
16.18.2	Überblick über den Arbeitsbereich	368
16.18.3	Eingabe der Extension-Konfiguration	369
16.18.4	Modellierung	370
16.18.5	Anlegen des Post-Objekts	372
16.18.6	Anlegen des Tag-Objekts	372
16.18.7	Relationen festlegen	373

16.19	Weitere Extbase-Internas.....	374
16.19.1	StoragePid.....	374
16.19.2	MVC-Request	376
16.19.3	FlashMessages realisieren	377
17	Die Fluid-Template-Engine.....	379
17.1	Vorbereitung	380
17.2	Basissyntax und einfache Ausgabe	382
17.2.1	Arrays.....	382
17.2.2	Objekte.....	383
17.3	Fluid ViewHelper-Syntax	384
17.3.1	Namespace (Namensraum).....	384
17.3.2	Argumente.....	385
17.4	ViewHelper-Übersicht	387
17.4.1	alias	387
17.4.2	base	388
17.4.3	cObject	388
17.4.4	count.....	389
17.4.5	cycle.....	389
17.4.6	debug.....	390
17.4.7	else	390
17.4.8	for.....	390
17.4.9	form.....	391
17.4.10	format.....	402
17.4.11	groupedFor	406
17.4.12	if	407
17.4.13	image.....	408
17.4.14	layout.....	409
17.4.15	link	409
17.4.16	render	414
17.4.17	renderFlashMessages	414
17.4.18	section	415
17.4.19	then.....	415
17.4.20	translate	415
17.4.21	uri.....	416
17.5	Erstellen eines eigenen ViewHelpers.....	416
17.5.1	Der Dummytext-ViewHelper	417
17.5.2	Zugriff auf die übergebenen Argumente	418
17.5.3	Zufügen von Argumenten	419
17.5.4	Tag-basierende ViewHelper.....	419
17.5.5	Der Variablen-Container.....	420
17.6	Verwendung von Fluid in klassischen Extensions	421
17.7	Layouts und Partialas.....	423
	Register.....	427