

Code-Generator

Luan Caduff

evernex
IT life services

Inhaltsverzeichnis

- ▶ Ausgangslage und Aufgabenstellung
 - Legacy SWO
 - Neues SWO
 - Code Generator Anforderungen
- ▶ Umsetzung
 - Struktur
 - Frontend
 - Backend
 - Snippet-Generierung
- ▶ Reflexion und Fazit
 - Persönliche Einschätzung
 - Alternativen
 - Zukunft





Aufgabenstellung

- ▶ Legacy SWO
- ▶ Neues SWO
- ▶ Code Generator Anforderungen



simpleWebOffice

Ingram Micro Distribution GmbH, Heisenbergbogen 3, Dornach bei München

Kunden Suche...

Mein Konto

Abmelden

Backend

Dashboard

Dashboard

xRM

CRM

Artikel - Lager

Technik

Einkauf

Vertrieb

Aufgaben

Leads

Kundenangebote

Kundenbestellungen

Support Packs (20)

Finanzen

Objekte

Kunden

5 Interessenten
293 Kunden

Detail Ansicht

Schnellsuche...

Artikel

44733 Neue Artikel
215791 Aktive Artikel

Detail Ansicht

Schnellsuche...

Lieferungen

0 Lieferungen Ausgehend
0 Lieferungen Eingehend

Detail Ansicht

Schnellsuche...

Rechnungen

195 Kunden Rechnungen
0 Lieferanten Rechnungen

Detail Ansicht

Schnellsuche...

Aufgaben

0 Neue Aufgaben
0 Offene Aufgaben

Detail Ansicht

Schnellsuche...

Aufträge

1 Neue Aufträge
372 Offene Aufträge

Detail Ansicht

Schnellsuche...

Projekte

0 Neue Projekte
0 Offene Projekte

Detail Ansicht

Schnellsuche...

Leads

0 Neue Leads
0 Offene Leads

Detail Ansicht

Schnellsuche...

Support Packs

0 Zu prüfende Support Packs
2000 Offene Support Packs

Detail Ansicht

Schnellsuche...

RMA

4 Neue RMA
0 Offene RMA

Detail Ansicht

Schnellsuche...

Tickets

8 Neue Tickets
1 Offene Tickets

Detail Ansicht

Schnellsuche...

Inkasso

0 Neue Inkasso Fälle
0 Offene Inkasso Fälle

Detail Ansicht

Schnellsuche...

© simpleWebOffice 2024



simpleWebOffice

Ingram Micro Distribution GmbH, Heisenbergbogen 3, Dornach bei München

Kunden Suche...

Mein Konto

Abmelden

Backend

xRM / Vertrieb / SupportPacks

Dashboard

xRM

CRM

Artikel - Lager

Technik

Einkauf

Vertrieb

Aufgaben

Leads

Kundenangebote

Kundenbestellungen

Support Packs (20)

Finanzen

Objekte

Support Packs - (Bearbeiten)

Suchen Resultat Daten Detail Detaildaten

Speichern Abbrechen Kopieren Hinzufügen Löschen Report Info E-Mail GS Umwandlung

Support Pack Nr.1202610

Zuständig

Kunden Nr. Käufer20052

Kunden Nr. Aktivierer

SpracheDeutsch

WährungEUR

Terminvorschlag

Beschreibung

Support Pack Datum08.05.202416:25:21

Gewünschtes Aktivierungsdatum

Service TypSupport

Support Pack StatusGekauft

Support Pack HerkunftPortal

Seriennummer

Support Pack Artikel Nr.POSSP050206005A

Support Pack KeyPOSSP050206005A202610-4

Typ / Modell

ArtikelbezeichnungSupport Pack: 5 jahre NBD, 5/13, für Synology Systemkosten bis 2,000.00, Synology Support Pack / Hardware / Spare Parts / POB (Nur Material gebucht)

HerstellerSynology

Laufzeit60 Monate

SLA5/13, NBD

Preiskategorie

Systempreis0.00



200+ Dateien! -

```
1 <?php
2 // *****
3 //  Verfasser:
4 //  Erstellt: 31.10.2018
5 // *****
6
7 // include("shop.openweboffice.head.inc.php");
8
9 require_once("global.config.inc.php");
10 require_once("../functions/simpinfo.global.inc.php");
11 require_once("../functions/simpinfo.browser.inc.php");
12 require_once("../functions/simpinfo.calendar.inc.php");
13 require_once("../functions/simpinfo.cryption.inc.php");
14 require_once("../functions/simpinfo.data.inc.php");
15 require_once("../functions/simpinfo.error.inc.php");
16 require_once("../functions/simpinfo.pdf.inc.php");
17 require_once("../functions/simpinfo.phpoffice.inc.php");
18 require_once("../functions/simpinfo.file.inc.php");
19 require_once("../functions/simpinfo.graphic.inc.php");
20 require_once("../functions/simpinfo.mail.inc.php");
21 require_once("../functions/simpinfo.ocr.inc.php");
22 require_once("../functions/simpinfo.object.inc.php");
23 require_once("../functions/simpinfo.permission.inc.php");
24 require_once("../functions/simpinfo.session.inc.php");
25 require_once("../functions/simpinfo.topal.inc.php");
26 require_once("../functions/simpinfo.variables.inc.php");
27
```

```
▼ scripts
  ▶ cora
  ▶ cronjob
    🐘 autostart.articleupdate.php
    🐘 browsererror.php
    🐘 controlcenter.openweboffice.action.php
    🐘 controlcenter.openweboffice.ajax.php
    📄 controlcenter.openweboffice.css
    🐘 controlcenter.openweboffice.function.inc.php
    🐘 controlcenter.openweboffice.head.inc.php
    📄 controlcenter.openweboffice.js
    🐘 controlcenter.openweboffice.php
    🐘 cron.php
    🐘 docsearch.php
    🐘 erp.article.alternative.cleaner.php
    🐘 erp.article.concerto.export.php
    🐘 erp.article.concerto.export.technogroup.php
    🐘 erp.article.download.config.inc.demosoag.php
    🐘 erp.article.download.config.inc.soag.php
    🐘 erp.article.download.config.inc.synorepair.php
    🐘 erp.article.download.php
    🐘 erp.article.download.prepare.php
    🐘 erp.article.export.level3.php
    🐘 erp.article.it-parts.import.php
    🐘 erp.article.manufacturer.cleaner.php
    🐘 erp.article.specialprice.import.php
    🐘 erp.article.statistics.update.php
    🐘 erp.article.structure.update.php
    🐘 erp.article.supportpack.cleaner.php
    🐘 erp.article.update.php
    🐘 erp.article.updater.email.php
    🐘 erp.article.updater.php
    📄 Export_Support_Pack.xlsx
    🐘 global.action.auth.inc.php
    🐘 global.action.customer.inc.php
    🐘 global.action.doc.inc.php
    🐘 global.action.importexport.inc.php
    🐘 global.action.inc.php
    🐘 global.action.meeting.inc.php
```

```

575 if($_REQUEST["Action"] == "SupportPackDetail" && $_REQUEST["Command"] == "add") {
576     $SQLQuery = "
577     select crm_om_supportpack.ObjectUUID,
578     crm_om_supportpack.SupportPackNo,
579     crm_om_supportpack.SupportPackKey
580     from crm_om_supportpack
581     where crm_om_supportpack.StateCd = 1
582     and ".fi_genSQLField("crm_om_supportpack", "SupportPackKey", $_REQUEST["FRM_SupportPackKey"], "Text", 1, 1).".
583     ";
584     $Result = fi_execSQLQuery($SQLQuery, __FILE__, __LINE__);
585     $row = mysqli_fetch_row($Result);
586     $TBObjectUUID = $row[0];
587     $TBSupportPackNo = $row[1];
588     $TBSupportPackKey = $row[2];
589     /* Neue Spport Pack Serviceadresse anlegen und selektieren */
590     $SQLQuery = "
591     update crm_om_supportpack set
592     ".fi_genSQLField("crm_om_supportpack", "SupportPackStatusCd", 5, "Integer", 1, 0).".
593     ".fi_getSQLInsertUser($GLOBALS["UserName"], "crm_om_supportpack")."
594     where StateCd = 1 and ".fi_genSQLField("crm_om_supportpack", "SupportPackKey", $TBSupportPackKey, "Text", 1, 1).".
595     ";
596     $Result = fi_execSQLQuery($SQLQuery, __FILE__, __LINE__);
597     $SQLQuery = "
598     insert into crm_om_supportpack_detail set
599     ".fi_genSQLField("crm_om_supportpack_detail", "ObjectUUID", $TBObjectUUID, "Text", 1, 0).",
600     ".fi_genSQLField("crm_om_supportpack_detail", "SupportPackNo", $TBSupportPackNo, "Integer", 1, 0).",
601     ".fi_genSQLField("crm_om_supportpack_detail", "SupportPackKey", $TBSupportPackKey, "Text", 1, 0).".
602     ".fi_getSQLInsertUser($GLOBALS["UserName"], "crm_om_supportpack_detail")."
603     ";
604     $Result = fi_execSQLQuery($SQLQuery, __FILE__, __LINE__);
605     /* Den neuen RecNo auslesen */
606     $SQLQuery = " select max(RecNo) from crm_om_supportpack_detail where StateCd = 1 and ".fi_genSQLField("crm_om_supportpack_detail", "SupportPackKey",
607     $Result = fi_execSQLQuery($SQLQuery, __FILE__, __LINE__);
608     $row = mysqli_fetch_row($Result);
609     $TBnextRecNo = $row[0];
610     $_REQUEST["Action"] = "SupportPackActivate";
611     $_REQUEST["Command"] = "check";
612     $_REQUEST["checkSupportPackDetailRecNo"] = $TBnextRecNo;
613 }

```

```

5 if($_REQUEST["Action"] == "SupportPackActivate" && $_REQUEST["Command"] == "check") {
    $FieldLabelsSupportPack = fi_getFieldLabels("crm_om_supportpack");

```



```

575 if($_REQUEST["Action"] == "SupportPackDetail" && $_REQUEST["Command"] == "add") {
576     $SQLQuery = "
577     select crm_om_supportpack.ObjectUUID,
578     crm_om_supportpack.SupportPackNo,
579     crm_om_supportpack.SupportPackKey
580     from crm_om_supportpack
581     where crm_om_supportpack.StateCd = 1
582     and ".fi_genSQLField("crm_om_supportpack", "SupportPackKey", $_REQUEST["FRM_SupportPackKey"], "Text", 1, 1).".
583     ";
584     $Result = fi_execSQLQuery($SQLQuery, __FILE__, __LINE__);
585     $row = mysqli_fetch_row($Result);
586     $TBObjectUUID = $row[0];
587     $TBSupportPackNo = $row[1];
588     $TBSupportPackKey = $row[2];
589     /* Neue Spport Pack Serviceadresse anlegen und selektieren */
590     $SQLQuery = "
591     update crm_om_supportpack set
592     ".fi_genSQLField("crm_om_supportpack", "SupportPackStatusCd", 5, "Integer", 1, 0).".
593     ".fi_getSQLInsertUser($GLOBALS["UserName"], "crm_om_supportpack")."
594     where StateCd = 1 and ".fi_genSQLField("crm_om_supportpack", "SupportPackKey", $TBSupportPackKey, "Text", 1, 1).".
595     ";
596     $Result = fi_execSQLQuery($SQLQuery, __FILE__, __LINE__);
597     $SQLQuery = "
598     insert into crm_om_supportpack_detail set
599     ".fi_genSQLField("crm_om_supportpack_detail", "ObjectUUID", $TBObjectUUID, "Text", 1, 0).",
600     ".fi_genSQLField("crm_om_supportpack_detail", "SupportPackNo", $TBSupportPackNo, "Integer", 1, 0).",
601     ".fi_genSQLField("crm_om_supportpack_detail", "SupportPackKey", $TBSupportPackKey, "Text", 1, 0).".
602     ".fi_getSQLInsertUser($GLOBALS["UserName"], "crm_om_supportpack_detail")."
603     ";
604     $Result = fi_execSQLQuery($SQLQuery, __FILE__, __LINE__);
605     /* Den neuen RecNo auslesen */
606     $SQLQuery = " select max(RecNo) from crm_om_supportpack_detail where StateCd = 1 and ".fi_genSQLField("crm_om_supportpack_detail", "SupportPackKey",
607     $Result = fi_execSQLQuery($SQLQuery, __FILE__, __LINE__);
608     $row = mysqli_fetch_row($Result);
609     $TBnextRecNo = $row[0];
610     $_REQUEST["Action"] = "SupportPackActivate";
611     $_REQUEST["Command"] = "check";
612     $_REQUEST["checkSupportPackDetailRecNo"] = $TBnextRecNo;
613 }

```

```

5 if($_REQUEST["Action"] == "SupportPackActivate" && $_REQUEST["Command"] == "check") {
    $FieldLabelsSupportPack = fi_getFieldLabels("crm_om_supportpack");

```



```

575 if($_REQUEST["Action"] == "SupportPackDetail" && $_REQUEST["Command"] == "add") {
576     $SQLQuery = "
577     select crm_om_supportpack.ObjectUUID,
578     crm_om_supportpack.SupportPackNo,
579     crm_om_supportpack.SupportPackKey
580     from crm_om_supportpack
581     where crm_om_supportpack.StateCd = 1
582     and ".fi_genSQLField("crm_om_supportpack", "SupportPackKey", $_REQUEST["FRM_SupportPackKey"], "Text", 1, 1).".
583     ";
584     $Result = fi_execSQLQuery($SQLQuery, __FILE__, __LINE__);
585     $row = mysqli_fetch_row($Result);
586     $TBObjectUUID = $row[0];
587     $TBSupportPackNo = $row[1];
588     $TBSupportPackKey = $row[2];
589     /* Neue Spport Pack Serviceadresse anlegen und selektieren */
590     $SQLQuery = "
591     update crm_om_supportpack set
592     ".fi_genSQLField("crm_om_supportpack", "SupportPackStatusCd", 5, "Integer", 1, 0).".
593     ".fi_getSQLInsertUser($GLOBALS["UserName"], "crm_om_supportpack")."
594     where StateCd = 1 and ".fi_genSQLField("crm_om_supportpack", "SupportPackKey", $TBSupportPackKey, "Text", 1, 1).".
595     ";
596     $Result = fi_execSQLQuery($SQLQuery, __FILE__, __LINE__);
597     $SQLQuery = "
598     insert into crm_om_supportpack_detail set
599     ".fi_genSQLField("crm_om_supportpack_detail", "ObjectUUID", $TBObjectUUID, "Text", 1, 0).".,
600     ".fi_genSQLField("crm_om_supportpack_detail", "SupportPackNo", $TBSupportPackNo, "Integer", 1, 0).".,
601     ".fi_genSQLField("crm_om_supportpack_detail", "SupportPackKey", $TBSupportPackKey, "Text", 1, 0).".
602     ".fi_getSQLInsertUser($GLOBALS["UserName"], "crm_om_supportpack_detail")."
603     ";
604     $Result = fi_execSQLQuery($SQLQuery, __FILE__, __LINE__);
605     /* Den neuen RecNo auslesen */
606     $SQLQuery = " select max(RecNo) from crm_om_supportpack_detail where StateCd = 1 and ".fi_genSQLField("crm_om_supportpack_detail", "SupportPackKey", $TBSupportPackKey, "Text", 1, 1).".
607     $Result = fi_execSQLQuery($SQLQuery, __FILE__, __LINE__);
608     $row = mysqli_fetch_row($Result);
609     $TBnextRecNo = $row[0];
610     $_REQUEST["Action"] = "SupportPackActivate";
611     $_REQUEST["Command"] = "check";
612     $_REQUEST["checkSupportPackDetailRecNo"] = $TBnextRecNo;
613 }

```

```

514 if($_REQUEST["Action"] == "SupportPackActivate" && $_REQUEST["Command"] == "check") {
515     $fieldLabelsSupportPack = fi_getFieldLabels("crm_om_supportpack");

```



SERVER AND OPTIONS

Master Data

Support Packs

...

Service Articles

Add Edit Info Delete | Filters: Manufacturer Select . . . | Search Enter search query . . .

Active in shop	Service type	Article Nr.	Article description	Manufacturer	Runtime
Yes	Support	SSP0515036132	Erweiterung zu SSP051503...	Synology	36 Months
Yes	Support	PWSSP050751213A	Support Pack: 1 Jahr 4h, 7/...	Synology	12 Months
Yes	Support	PWSSP050752413A	Support Pack: 2 Jahre 4h, 7...	Synology	24 Months
Yes	Support	PWSSP050751205A	Support Pack: 1 Jahr NBD, ...	Synology	12 Months
Yes	Support	PWSSP050752405A	Support Pack: 2 Jahre NBD,...	Synology	24 Months
Yes	Support	PWSSP050751203A	Support Pack: 1 Jahr 4h, 5/...	Synology	12 Months
Yes	Support	PWSSP050752403A	Support Pack: 2 Jahre 4h, 5...	Synology	24 Months
Yes	Support	PWSSP056501213A	Support Pack: 1 Jahr für Sy...	Synology	12 Months
Yes	Support	PWSSP056502413A	Support Pack: 2 Jahre für S...	Synology	24 Months
Yes	Support	PWSSP056501205A	Support Pack: 1 Jahr für Sy...	Synology	12 Months
Yes	Support	PWSSP056502405A	Support Pack: 2 Jahre für S...	Synology	24 Months
Yes	Support	PWSSP056501203A	Support Pack: 1 Jahr für Sy...	Synology	12 Months
Yes	Support	PWSSP056502403A	Support Pack: 2 Jahre für S...	Synology	24 Months
Yes	Support	PWSSP056001213A	Support Pack: 1 Jahr für Sy...	Synology	12 Months
Yes	Support	PWSSP056002413A	Support Pack: 2 Jahre für S...	Synology	24 Months
Yes	Support	PWSSP056001205A	Support Pack: 1 Jahr für Sy...	Synology	12 Months
Yes	Support	PWSSP056002405A	Support Pack: 2 Jahre für S...	Synology	24 Months
Yes	Support	PWSSP056001203A	Support Pack: 1 Jahr für Sy...	Synology	12 Months
Yes	Support	PWSSP056002403A	Support Pack: 2 Jahre für S...	Synology	24 Months
Yes	Support	PWSSP055501213A	Support Pack: 1 Jahr für Sy...	Synology	12 Months
Yes	Support	PWSSP055502413A	Support Pack: 2 Jahre für S...	Synology	24 Months
Yes	Support	PWSSP055501205A	Support Pack: 1 Jahr für Sy...	Synology	12 Months
Yes	Support	PWSSP055502405A	Support Pack: 2 Jahre für S...	Synology	24 Months
Yes	Support	PWSSP055501203A	Support Pack: 1 Jahr für Sy...	Synology	12 Months
Yes	Support	PWSSP055502403A	Support Pack: 2 Jahre für S...	Synology	24 Months
Yes	Support	PWSSP055001213A	Support Pack: 1 Jahr 4h, 7/...	Synology	12 Months
Yes	Support	PWSSP055002413A	Support Pack: 2 Jahre 4h, 7...	Synology	24 Months
Yes	Support	PWSSP055001205A	Support Pack: 1 Jahr NBD, ...	Synology	12 Months
Yes	Support	PWSSP055002405A	Support Pack: 2 Jahre NBD,...	Synology	24 Months
Yes	Support	PWSSP055002403A	Support Pack: 2 Jahre 4h, 5...	Synology	24 Months
Yes	Support	PWSSP055001203A	Support Pack: 1 Jahr 4h, 5/...	Synology	12 Months
Yes	Support	SSP087000613A	Support Pack Renewal: 6 M...	HPE	6 Months
Yes	Support	SSP090186005R1	Support Pack Renewal: 1 Ja...	Transtec Desktop	12 Months
Yes	Replace	SP0100033R	Replace Pack: 5 Jahre Voro...	Synology	60 Months
Yes	Replace	SP0100032R	Replace Pack: 5 Jahre Voro...	Synology	60 Months
Yes	Audit	SSP060010000W	WLAN Ausleuchtung für Ev...	Ubiquiti	-
Yes	Support	SSP043506005A	Support Pack Supermicro S...	Supermicro	60 Months
Yes	Support	SSP150741213A	Support Pack: 1 Jahr 4h, 7/...	Dell	12 Months

Edit Service Article «SSP0515036132»

Active in Shop	<input checked="" type="checkbox"/>
Service Type *	Support
Article Nr. *	SSP0515036132
Old Article Nr.	
Article description	Erweiterung zu SSP051503613A201362-9 auf Total CHF 25'000.00 - Synology 7x24 4h - Service Ende 04.04.2026
Manufacturer *	Synology
Runtime in months *	36
Service Level Agreement *	7/24, 4h
Only Work (labor only)	<input type="checkbox"/>
Only Material (parts only)	<input type="checkbox"/>
Defective Media Retention	<input type="checkbox"/>
Post Warranty	<input type="checkbox"/>
Indiv	<input type="checkbox"/>
Simple Activation	<input type="checkbox"/>
Indiv Text	<input checked="" type="checkbox"/>
NAS (?)	<input type="checkbox"/>
On-Site Engineer	<input checked="" type="checkbox"/>
Spare Parts	<input checked="" type="checkbox"/>

Save

Cancel

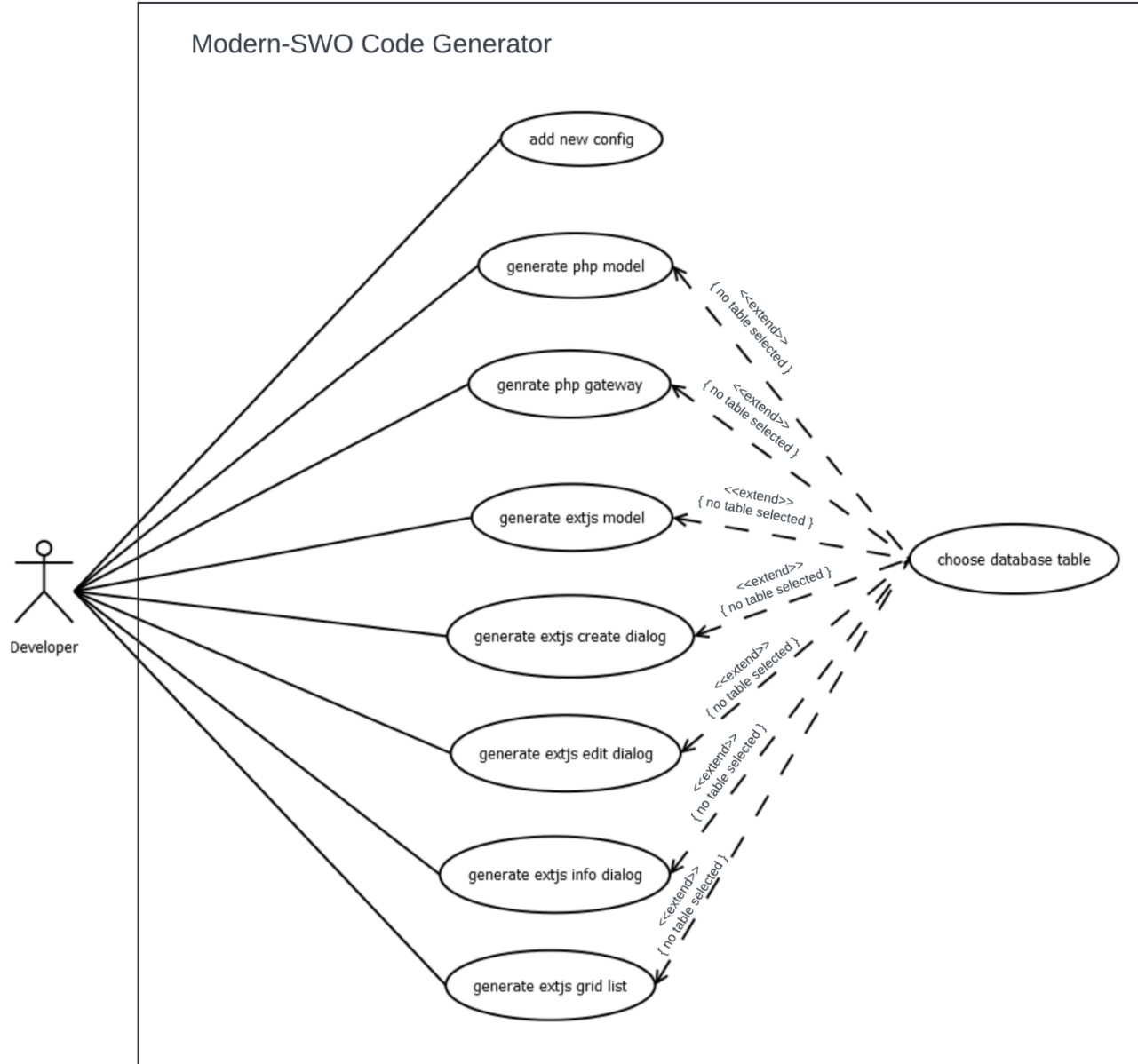


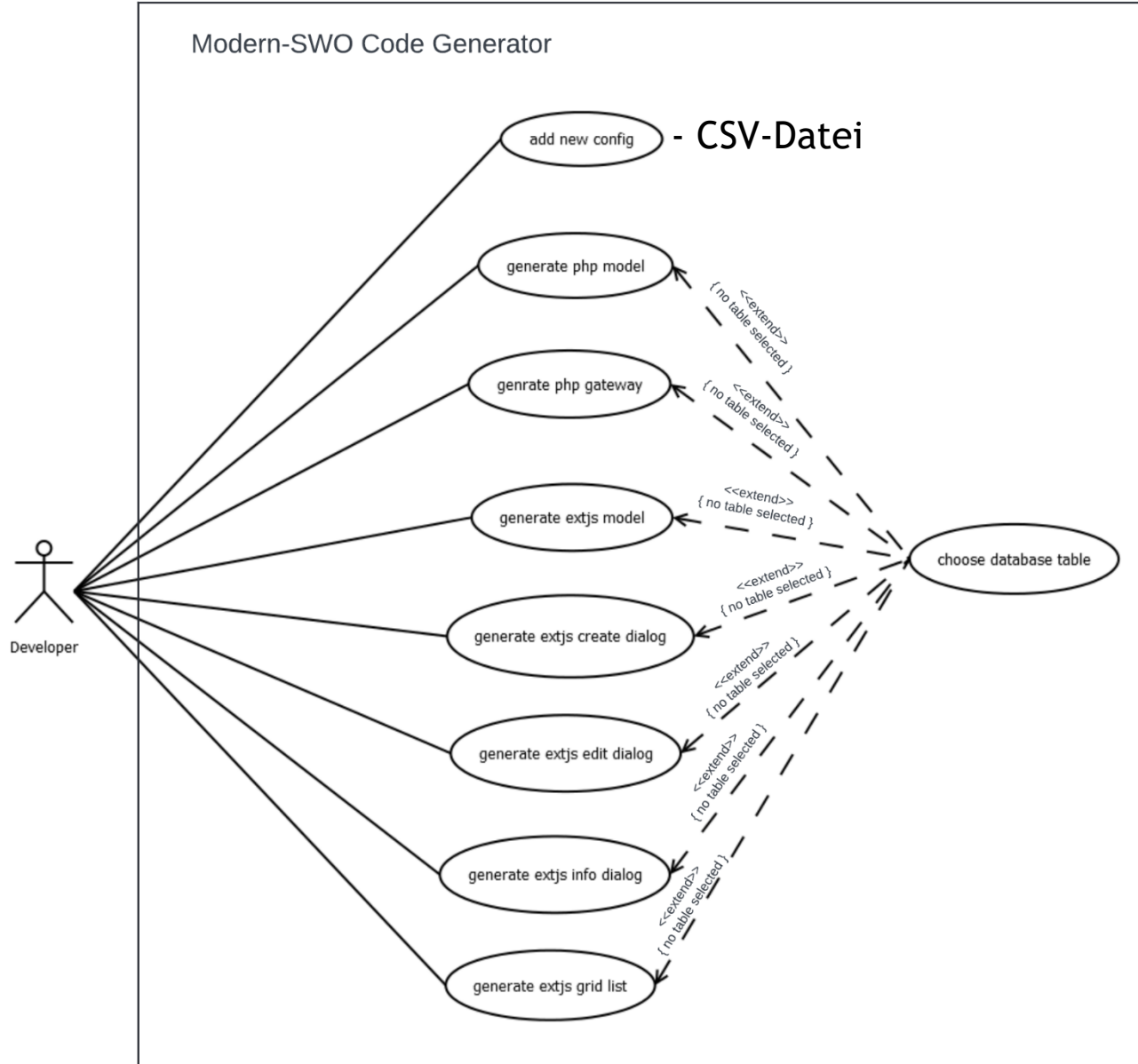
PHP

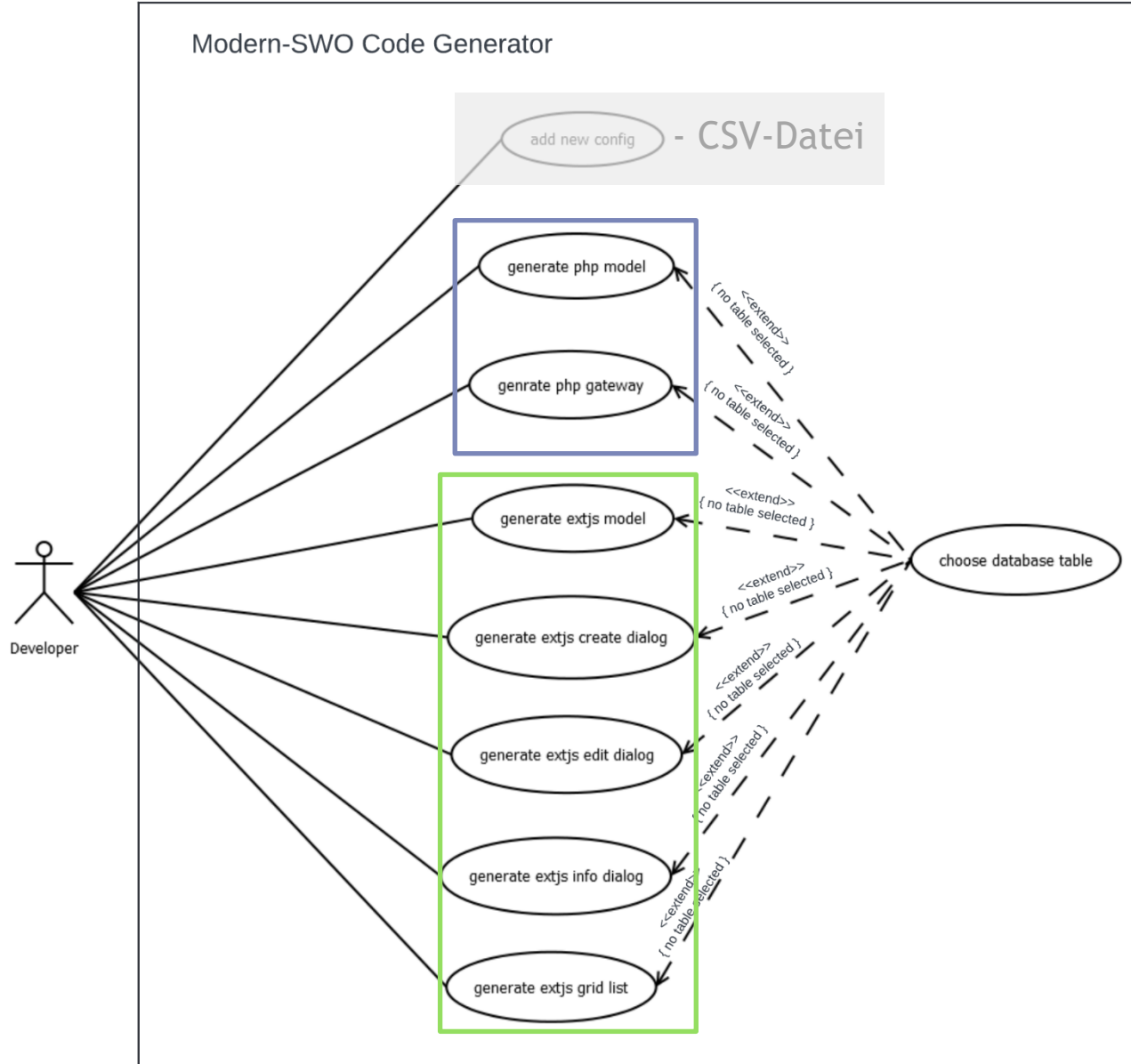
```
class ERParticleService extends BaseModel implements JsonSerializable {  
  
    private int      $id                = 0;  
    private int      $stateCd           = 1;  
    private int      $shopActiveCd      = 0;  
    private ?int     $serviceTypeCd     = NULL;  
    private string   $articleNo         = '';  
    private string   $articleNoOld      = '';  
    private string   $articleDescription = '';  
    private ?int     $supportPackManufacturerCd = NULL;  
    private ?int     $supportPackRunTimeCd = NULL;  
    private ?int     $slaCd              = NULL;  
    private ?int     $laborOnlyCd        = NULL;  
    private ?int     $partsOnlyCd        = NULL;  
    private ?int     $dmrCd              = NULL;  
    private ?int     $postWarrantyCd     = NULL;  
    private ?int     $indivCd            = NULL;  
    private ?int     $simpleActivationCd  = NULL;  
    private ?int     $indivTextCd        = NULL;  
    private ?int     $nasCd              = NULL;  
    private ?int     $onsiteEngineerCd   = NULL;  
    private ?int     $sparePartCd        = NULL;  
    private ?int     $hardwareCd         = NULL;  
    private string   $snCheck            = '';  
    private ?int     $activationTypeCd    = NULL;  
    private ?float   $systemPrice        = NULL;  
    private ?float   $servicePrice       = NULL;  
    private string   $systemType         = '';  
    private string   $articleNoManufacturer = '';  
    private string   $remarks            = '';  
    private ?DateTime $validBegin        = NULL;  
    private ?DateTime $validEnd          = NULL;  
    private string   $competence         = '';  
    private string   $permission         = '';  
    private string   $crtUser            = '';  
    private ?DateTime $crtDate           = NULL;  
    private string   $updUser            = '';  
    private ?DateTime $updDate           = NULL;  
}
```

ExtJs

```
1 Ext.define('swo.model.ERParticleService', {  
2     extend: 'Ext.data.Model',  
3  
4     fields: [  
5         { name: 'stateCd',          type: 'int' },  
6         { name: 'articleNo',        type: 'string' },  
7         { name: 'articleNoOld',     type: 'string' },  
8         { name: 'articleDescription', type: 'string' },  
9         { name: 'supportPackManufacturerCd', type: 'int' },  
10        { name: 'supportPackRunTimeCd', type: 'int' },  
11        { name: 'slaCd',             type: 'int' },  
12        { name: 'laborOnlyCd',       type: 'int' },  
13        { name: 'partsOnlyCd',       type: 'int' },  
14        { name: 'dmrCd',             type: 'int' },  
15        { name: 'postWarrantyCd',    type: 'int' },  
16        { name: 'indivCd',          type: 'int' },  
17        { name: 'simpleActivationCd', type: 'int' },  
18        { name: 'indivTextCd',       type: 'int' },  
19        { name: 'nasCd',             type: 'int' },  
20        { name: 'onsiteEngineerCd',  type: 'int' },  
21        { name: 'sparePartCd',       type: 'int' },  
22        { name: 'hardwareCd',        type: 'int' },  
23        { name: 'snCheck',           type: 'string' },  
24        { name: 'activationTypeCd',  type: 'int' },  
25        { name: 'systemPrice',       type: 'float' },  
26        { name: 'servicePrice',     type: 'float' },  
27        { name: 'systemType',       type: 'string' },  
28        { name: 'articleNoManufacturer', type: 'string' },  
29        { name: 'remarks',           type: 'string' },  
30        { name: 'validBegin',        type: 'date' },  
31        { name: 'validEnd',          type: 'date' },  
32        { name: 'competence',        type: 'string' },  
33        { name: 'permission',        type: 'string' },  
34        { name: 'crtUser',           type: 'string' },  
35        { name: 'crtDate',           type: 'date' },  
36        { name: 'updUser',           type: 'string' },  
37        { name: 'updDate',           type: 'date' },  
38    ]  
39  
40 });  
41
```







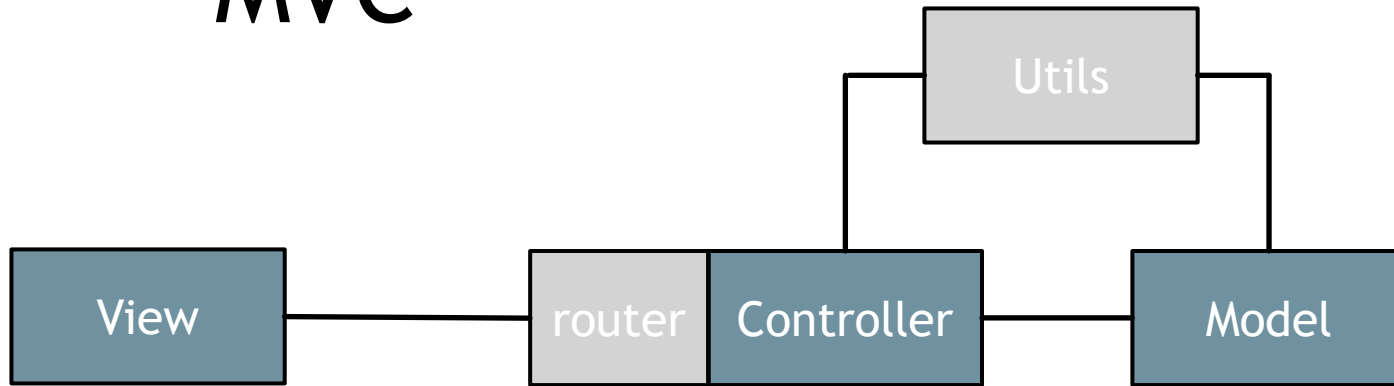


Umsetzung

- ▶ Struktur
- ▶ Frontend
- ▶ Backend
- ▶ Snippet-Generierung



MVC





```
1 $( document ).ready(function() {  
2  
3     // apply eventListeners  
4     applyClickListeners();  
5     applyFocusOutListeners();  
6     applyChangeListeners();  
7  
8  
9     fillConfigSelectbox();  
10  
11     /////////////////////////////////////////////////// START CLICK LISTENERS + FUNCTIONS ///////////////////////////////////  
12  
13     function applyClickListeners() {  
232     }  
233  
234     $('#copyButton').on('click', (event) => {  
239  
240         });  
241  
242     function openConfigForm() {  
254  
255     }  
256  
257     function closeConfigForm() {  
268     }  
269  
270     function clearConfigForm() {  
275     }  
276  
277     function selectText(containerid) {  
288     }  
289  
290     function clearSelection() {  
296     }  
297  
298     /////////////////////////////////////////////////// END CLICK LISTENERS + FUNCTIONS ///////////////////////////////////  
299  
300  
301     /////////////////////////////////////////////////// START FOCUSOUT LISTENERS + FUNCTIONS ///////////////////////////////////  
302  
303     function applyFocusOutListeners() {  
318  
319     }
```



router

```
if(isset($_GET['action']) && $_GET['action'] == 'getConfigs') {  
    $controller = new ConfigController();  
    $controller->getConfigs();  
    exit();  
}
```

Controller

```
public function getConfigs() {  
    $response = new Response();  
  
    $response->setStatus(Response::$STATUS_OK);  
    $response->setMessage('');  
    $response->setHttpStatusCode(Response::$HTTP_STATUS_OK);  
    $response->setData(ConfigManager::getInstance()->getConfigs());  
  
    $response->respond();  
}
```



router

```
if(isset($_GET['action']) && $_GET['action'] == 'getConfigs') {  
    $controller = new ConfigController();  
    $controller->getConfigs();  
    exit();  
}
```

Controller

Utils

```
public function getConfigs() {  
    $response = new Response();  
  
    $response->setStatus(Response::$STATUS_OK);  
    $response->setMessage('');  
    $response->setHttpStatusCode(Response::$HTTP_STATUS_OK);  
    $response->setData(ConfigManager::getInstance()->getConfigs());  
  
    $response->respond();  
}
```



Controller - SnippetControllerExtJs

Utils

```
public function generateModel($configId, $databaseId, $tableId) {
    $response = new Response();

    $config = ConfigManager::getInstance()->getConfigById($configId);
    $database = Database::getById($config, $databaseId);
    $table = Table::getById($database, $tableId);

    $fields = $this->prepareFields($table->getFields());
    $className = $this->convertToClassname($table->getName());
    $newLine1 = '<br>';
    $newLine2 = '<br><br>';

    $header = $this->generateModelHeader($className);
    $body = $this->generateModelBody($fields);
    $footer = $this->generateFooter();

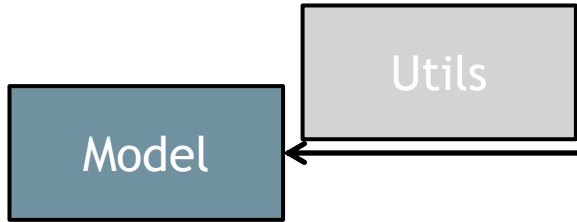
    $output = $header.
        $newLine2.
        $body.
        $newLine2.
        $footer;

    $response->setStatus(Response::$STATUS_OK);
    $response->setMessage('');
    $response->setHttpResponseCode(Response::$HTTP_STATUS_OK);
    $response->setData($output);

    $response->respond();
}
```



Controller - SnippetControllerExtJs



```
public function generateModel($configId, $databaseId, $tableId) {
    $response = new Response();

    $config = ConfigManager::getInstance()->getConfigById($configId);
    $database = Database::getById($config, $databaseId);
    $table = Table::getById($database, $tableId);

    $fields = $this->prepareFields($table->getFields());
    $className = $this->convertToClassname($table->getName());
    $newLine1 = '<br>';
    $newLine2 = '<br><br>';

    $header = $this->generateModelHeader($className);
    $body = $this->generateModelBody($fields);
    $footer = $this->generateFooter();

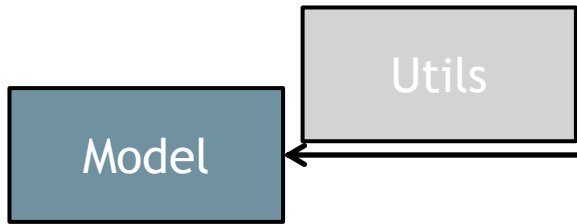
    $output = $header.
        $newLine2.
        $body.
        $newLine2.
        $footer;

    $response->setStatus(Response::$STATUS_OK);
    $response->setMessage('');
    $response->setHttpResponseCode(Response::$HTTP_STATUS_OK);
    $response->setData($output);

    $response->respond();
}
```



Controller - SnippetControllerExtJs



Member-
Methode

```
private function generateModelBody($fields): string {
    $content = "{ $this->indent(1)}fields: [";
    $content .= "<br>";

    foreach ($fields as $field) {
        $name = $field['COLUMN_NAME'];
        $lname = lcfirst($name);
        $type = $field['DATA_TYPE'];

        $typeSpaces = $field['DATA_TYPE_SPACES'];
        $nameSpaces = $field['COLUMN_NAME_SPACES'];

        $content .= "{ $this->indent(2)}{ name: '$lname', $nameSpaces type: '$type' },<br>";
    }
    $content = $this->removeLastOccurrence($content, ',');
    $content .= "{ $this->indent(1)}]";
    return $content;
}
```

```
public function generateModel($configId, $databaseId, $tableId) {
    $response = new Response();

    $config = ConfigManager::getInstance()->getConfigById($configId);
    $database = Database::getById($config, $databaseId);
    $table = Table::getById($database, $tableId);

    $fields = $this->prepareFields($table->getFields());
    $className = $this->convertToClassname($table->getName());
    $newLine1 = '<br>';
    $newLine2 = '<br><br>';

    $header = $this->generateModelHeader($className);
    $body = $this->generateModelBody($fields);
    $footer = $this->generateFooter();

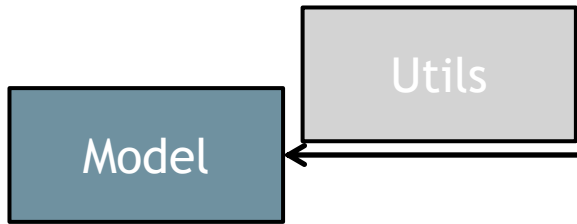
    $output = $header.
        $newLine2.
        $body.
        $newLine2.
        $footer;

    $response->setStatus(Response::STATUS_OK);
    $response->setMessage('');
    $response->setHttpStatusCode(Response::HTTP_STATUS_OK);
    $response->setData($output);

    $response->respond();
}
```




Controller - SnippetControllerExtJs



Member-
Methode

```
private function generateModelBody($fields): string {
    $content = "{ $this->indent(1)}fields: [";
    $content .= "<br>";

    foreach ($fields as $field) {
        $name = $field['COLUMN_NAME'];
        $lname = lcfirst($name);
        $type = $field['DATA_TYPE'];

        $typeSpaces = $field['DATA_TYPE_SPACES'];
        $nameSpaces = $field['COLUMN_NAME_SPACES'];

        $content .= "{ $this->indent(2)}{ name: '$lname', $nameSpaces type: '$type' },<br>";
    }
    $content = $this->removeLastOccurrence($content, ',');
    $content .= "{ $this->indent(1)}]";
    return $content;
}
```

```
public function generateModel($configId, $databaseId, $tableId) {
    $response = new Response();

    $config = ConfigManager::getInstance()->getConfigById($configId);
    $database = Database::getById($config, $databaseId);
    $table = Table::getById($database, $tableId);

    $fields = $this->prepareFields($table->getFields());
    $className = $this->convertToClassname($table->getName());
    $newLine1 = '<br>';
    $newLine2 = '<br><br>';

    $header = $this->generateModelHeader($className);
    $body = $this->generateModelBody($fields);
    $footer = $this->generateFooter();

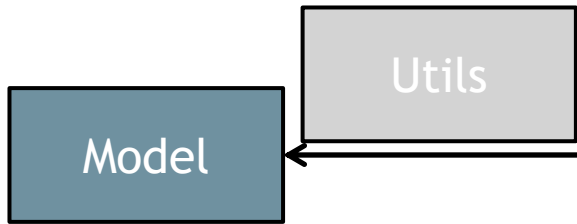
    $output = $header.
                $newLine2.
                $body.
                $newLine2.
                $footer;

    $response->setStatus(Response::STATUS_OK);
    $response->setMessage('');
    $response->setHttpStatusCode(Response::HTTP_STATUS_OK);
    $response->setData($output);

    $response->respond();
}
```



Controller - SnippetControllerExtJs



Member-
Methode

```
private function generateModelBody($fields): string {
    $content = "{ $this->indent(1)}fields: [";
    $content .= "<br>";

    foreach ($fields as $field) {
        $name = $field['COLUMN_NAME'];
        $lname = lcfirst($name);
        $type = $field['DATA_TYPE'];

        $typeSpaces = $field['DATA_TYPE_SPACES'];
        $nameSpaces = $field['COLUMN_NAME_SPACES'];

        $content .= "{ $this->indent(2)}{ name: '$lname', $nameSpaces type: '$type' },<br>";
    }
    $content = $this->removeLastOccurrence($content, ',');
    $content .= "{ $this->indent(1)}]";
    return $content;
}
```

```
public function generateModel($configId, $databaseId, $tableId) {
    $response = new Response();

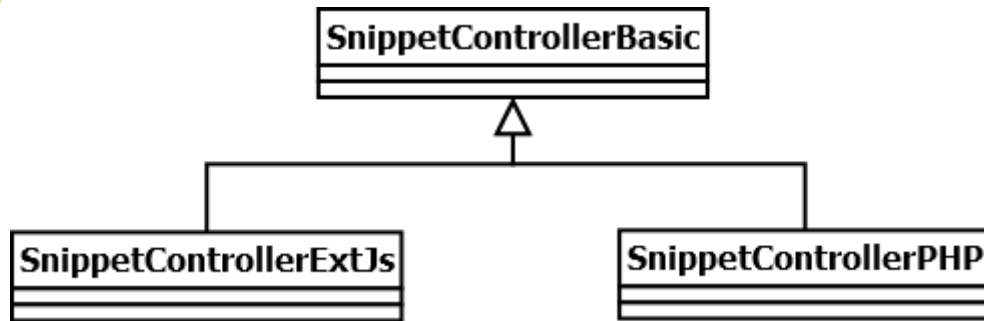
    $config = ConfigManager::getInstance()->getConfigById($configId);
    $database = Database::getById($config, $databaseId);
    $table = Table::getById($database, $tableId);

    $fields = $this->prepareFields($table->getFields());
    $className = $this->convertToClassname($table->getName());
    $newLine1 = '<br>';
    $newLine2 = '<br><br>';

    $header = $this->generateModelHeader($className);
    $body = $this->generateModelBody($fields);
    $footer = $this->generateFooter();

    $output = $header.
        $newLine2.
        $body.
        $newLine2.
        $footer;

    $response->setStatus(Response::STATUS_OK);
    $response->setMessage('');
    $response->setHttpStatusCode(Response::HTTP_STATUS_OK);
    $response->setData($output);
    $response->respond();
}
```



SnippetControllerExtJs

```
public function generateModel($configId, $databaseId, $tableId) {
    $response = new Response();

    $config = ConfigManager::getInstance()->getConfigById($configId);
    $database = Database::getById($config, $databaseId);
    $table = Table::getById($database, $tableId);

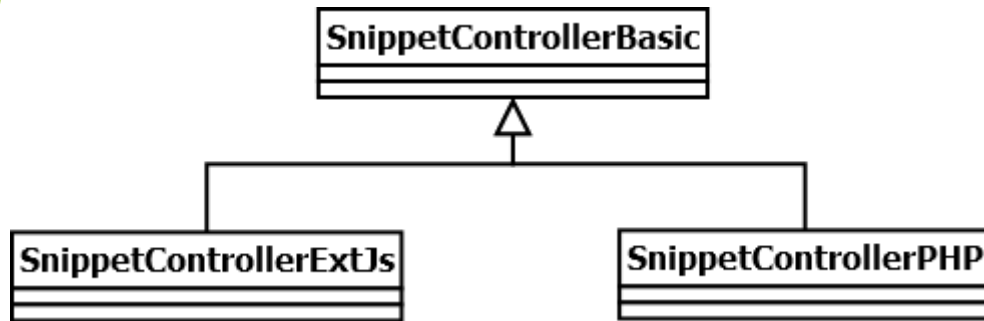
    $fields = $this->prepareFields($table->getFields());
    $className = $this->convertToClassname($table->getName());
    $newline1 = '<br>';
    $newline2 = '<br><br>';

    $header = $this->generateModelHeader($className);
    $body = $this->generateModelBody($fields);
    $footer = $this->generateFooter();

    $output = $header.
        $newline2.
        $body.
        $newline2.
        $footer;

    $response->setStatus(Response::$STATUS_OK);
    $response->setMessage('');
    $response->setHttpResponseCode(Response::$HTTP_STATUS_OK);
    $response->setData($output);

    $response->respond();
}
```



SnippetControllerBasic

```
protected function convertToClassname(string $tableName): string {
    $arr = explode('_', $tableName);
    for ($i = 0; $i < count($arr); $i++) {
        if ($i == 0) {
            $arr[$i] = strtoupper($arr[$i]);
        } else if ($i == 1) {
            $arr[$i] = strtolower($arr[$i]);
        } else {
            $arr[$i] = ucfirst($arr[$i]);
        }
    }
    return implode('', $arr);
}
```

SnippetControllerExtJs

```
public function generateModel($configId, $databaseId, $tableId) {
    $response = new Response();

    $config = ConfigManager::getInstance()->getConfigById($configId);
    $database = Database::getById($config, $databaseId);
    $table = Table::getById($database, $tableId);

    $fields = $this->prepareFields($table->getFields());
    $className = $this->convertToClassname($table->getName());
    $newline1 = '<br>';
    $newline2 = '<br><br>';

    $header = $this->generateModelHeader($className);
    $body = $this->generateModelBody($fields);
    $footer = $this->generateFooter();

    $output = $header.
        $newline2.
        $body.
        $newline2.
        $footer;

    $response->setStatus(Response::$STATUS_OK);
    $response->setMessage('');
    $response->setHttpResponseCode(Response::$HTTP_STATUS_OK);
    $response->setData($output);

    $response->respond();
}
```



SnippetControllerExtJs

```
private function generateModelBody($fields): string {  
    $content = "{$this->indent(1)}fields: [";  
    $content .= "<br>";  
  
    foreach ($fields as $field) {  
        $name = $field['COLUMN_NAME'];  
        $lname = lcfirst($name);  
        $type = $field['DATA_TYPE'];  
  
        $typeSpaces = $field['DATA_TYPE_SPACES'];  
        $nameSpaces = $field['COLUMN_NAME_SPACES'];  
  
        $content .= "{$this->indent(2)}{ name: '$lname',$nameSpaces type: '$type' },<br>";  
    }  
    $content = $this->removeLastOccurrence($content, ',');  
    $content .= "{$this->indent(1)}]";  
    return $content;  
}
```

SnippetControllerBasic

```
protected function indent(int $level) {  
    $indentSize = str_repeat('&nbsp;', self::$INDENT_SPACES);  
    return str_repeat($indentSize, $level);  
}
```



SnippetControllerExtJs

```
private function generateModelBody($fields): string {  
    $content = "{ $this->indent(1)}fields: [";  
    $content .= "<br>";  
  
    foreach ($fields as $field) {  
        $name = $field['COLUMN_NAME'];  
        $lname = lcfirst($name);  
        $type = $field['DATA_TYPE'];  
  
        $typeSpaces = $field['DATA_TYPE_SPACES'];  
        $nameSpaces = $field['COLUMN_NAME_SPACES'];  
  
        $content .= "{ $this->indent(2)}{ name: '$lname', $nameSpaces type: '$type' },<br>";  
    }  
    $content = $this->removeLastOccurrence($content, ',');  
    $content .= "{ $this->indent(1)}}";  
    return $content;  
}
```

SnippetControllerBasic

```
protected function indent(int $level) {  
    $indentSize = str_repeat('&nbsp;', self::$INDENT_SPACES);  
    return str_repeat($indentSize, $level);  
}
```

```
class SnippetControllerBasic {  
  
    // how many spaces is equal to one indent  
    protected static int $INDENT_SPACES = 4;  
}
```



SnippetControllerExtJs

```
private function generateModelBody($fields): string {  
    $content = "{${this->indent(1)}}fields: [";  
    $content .= "<br>";  
  
    foreach ($fields as $field) {  
        $name = $field['COLUMN_NAME'];  
        $lname = lcfirst($name);  
        $type = $field['DATA_TYPE'];  
  
        $typeSpaces = $field['DATA_TYPE_SPACES'];  
        $nameSpaces = $field['COLUMN_NAME_SPACES'];  
  
        $content .= "{${this->indent(2)}}{ name: '$lname',$nameSpaces type: '$type' },{br>";  
    }  
    $content = ${this->removeLastOccurrence($content, ','});  
    $content .= "{${this->indent(1)}}]";  
    return $content;  
}
```

SnippetControllerBasic

```
protected function indent(int $level) {  
    $indentSize = str_repeat('&nbsp;', self::$INDENT_SPACES);  
    return str_repeat($indentSize, $level);  
}
```

```
class SnippetControllerBasic {  
  
    // how many spaces is equal to one indent  
    protected static int $INDENT_SPACES = 4;
```

```
// remove only the last occurrence of a substring in a string  
protected function removeLastOccurrence(string $string, string $search) {  
    $offset = strrpos($string, $search);  
    if ($offset !== false) {  
        $length = strlen($search);  
        $string = substr_replace($string, '', $offset, $length);  
    }  
    return $string;  
}
```




SnippetControllerExtJs

```
private function generateModelBody($fields): string {  
  
    $content = "{$this->indent(1)}fields: [";  
    $content .= "<br>";  
  
    foreach ($fields as $field) {  
  
        $name = $field['COLUMN_NAME'];  
        $lname = lcfirst($name);  
        $type = $field['DATA_TYPE'];  
  
        $typeSpaces = $field['DATA_TYPE_SPACES'];  
        $nameSpaces = $field['COLUMN_NAME_SPACES'];  
  
        $content .= "{$this->indent(2)}{ name: '$lname',$nameSpaces type: '$type' },<br>";  
    }  
    $content = $this->removeLastOccurrence($content, ',');  
    $content .= "{$this->indent(1)}]";  
    return $content;  
}
```

Output:

```
Ext.define('swo.model.ERParticleService', {  
    extend: 'Ext.data.Model',  
  
    fields: [  
        { name: 'id', type: 'int' },  
        { name: 'stateCd', type: 'int' },  
        { name: 'shopActiveCd', type: 'int' },  
        { name: 'serviceTypeCd', type: 'int' },  
        { name: 'articleNo', type: 'string' },  
        { name: 'articleNoOld', type: 'string' },  
        { name: 'articleDescription', type: 'string' },  
        { name: 'supportPackManufacturerCd', type: 'int' },  
        { name: 'supportPackRunTimeCd', type: 'int' },  
        { name: 'sLACd', type: 'int' },  
        { name: 'laborOnlyCd', type: 'int' },  
        { name: 'partsOnlyCd', type: 'int' },  
        { name: 'dmRCd', type: 'int' },  
        { name: 'postWarrantyCd', type: 'int' },  
        { name: 'indivCd', type: 'int' },  
        { name: 'simpleActivationCd', type: 'int' },  
        { name: 'indivTextCd', type: 'int' },  
        { name: 'nASCd', type: 'int' },  
        { name: 'onsiteEngineerCd', type: 'int' },  
        { name: 'sparePartCd', type: 'int' },  
        { name: 'hardwareCd', type: 'int' },  
        { name: 'sNCheck', type: 'string' },  
        { name: 'activationTypeCd', type: 'int' },  
        { name: 'systemPrice', type: 'float' },  
        { name: 'servicePrice', type: 'float' },  
        { name: 'systemType', type: 'string' },  
        { name: 'articleNoManufacturer', type: 'string' },  
        { name: 'remarks', type: 'string' },  
        { name: 'validBegin', type: 'date' },  
        { name: 'validEnd', type: 'date' },  
        { name: 'competence', type: 'string' },  
        { name: 'permission', type: 'string' },  
        { name: 'crtUser', type: 'string' },  
        { name: 'crtDate', type: 'date' },  
        { name: 'updUser', type: 'string' },  
        { name: 'updDate', type: 'date' }  
    ]  
});
```



SnippetControllerExtJs

```
private function generateModelBody($fields): string {  
    $content = "{$this->indent(1)}fields: [";  
    $content .= "<br>";  
  
    foreach ($fields as $field) {  
        $name = $field['COLUMN_NAME'];  
        $lname = lcfirst($name);  
        $type = $field['DATA_TYPE'];  
  
        $typeSpaces = $field['DATA_TYPE_SPACES'];  
        $nameSpaces = $field['COLUMN_NAME_SPACES'];  
  
        $content .= "{$this->indent(2)}{ name: '$lname', $nameSpaces type: '$type' },<br>";  
    }  
    $content = $this->removeLastOccurrence($content, ',');  
    $content .= "{$this->indent(1)}]";  
    return $content;  
}
```

Output:

```
Ext.define('swo.model.ERParticleService', {  
    extend: 'Ext.data.Model',  
  
    fields: [  
        { name: 'id', type: 'int' },  
        { name: 'stateCd', type: 'int' },  
        { name: 'shopActiveCd', type: 'int' },  
        { name: 'serviceTypeCd', type: 'int' },  
        { name: 'articleNo', type: 'string' },  
        { name: 'articleNoOld', type: 'string' },  
        { name: 'articleDescription', type: 'string' },  
        { name: 'supportPackManufacturerCd', type: 'int' },  
        { name: 'supportPackRunTimeCd', type: 'int' },  
        { name: 'sLACd', type: 'int' },  
        { name: 'laborOnlyCd', type: 'int' },  
        { name: 'partsOnlyCd', type: 'int' },  
        { name: 'dmRCd', type: 'int' },  
        { name: 'postWarrantyCd', type: 'int' },  
        { name: 'indivCd', type: 'int' },  
        { name: 'simpleActivationCd', type: 'int' },  
        { name: 'indivTextCd', type: 'int' },  
        { name: 'nASCd', type: 'int' },  
        { name: 'onsiteEngineerCd', type: 'int' },  
        { name: 'sparePartCd', type: 'int' },  
        { name: 'hardwareCd', type: 'int' },  
        { name: 'sNCheck', type: 'string' },  
        { name: 'activationTypeCd', type: 'int' },  
        { name: 'systemPrice', type: 'float' },  
        { name: 'servicePrice', type: 'float' },  
        { name: 'systemType', type: 'string' },  
        { name: 'articleNoManufacturer', type: 'string' },  
        { name: 'remarks', type: 'string' },  
        { name: 'validBegin', type: 'date' },  
        { name: 'validEnd', type: 'date' },  
        { name: 'competence', type: 'string' },  
        { name: 'permission', type: 'string' },  
        { name: 'crtUser', type: 'string' },  
        { name: 'crtDate', type: 'date' },  
        { name: 'updUser', type: 'string' },  
        { name: 'updDate', type: 'date' }  
    ]  
});
```



SnippetControllerExtJs

```
private function generateModelBody($fields): string {  
  
    $content = "{$this->indent(1)}fields: [";  
    $content .= "<br>";  
  
    foreach ($fields as $field) {  
  
        $name = $field['COLUMN_NAME'];  
        $lname = lcfirst($name);  
        $type = $field['DATA_TYPE'];  
  
        $typeSpaces = $field['DATA_TYPE_SPACES'];  
        $nameSpaces = $field['COLUMN_NAME_SPACES'];  
  
        $content .= "{$this->indent(2)}{ name: '$lname',$nameSpaces type: '$type' },<br>";  
    }  
    $content = $this->removeLastOccurrence($content, ',');  
    $content .= "{$this->indent(1)}]";  
    return $content;  
}
```

Output:

```
Ext.define('swo.model.ERParticleService', {  
    extend: 'Ext.data.Model',  
  
    fields: [  
        { name: 'id', type: 'int' },  
        { name: 'stateCd', type: 'int' },  
        { name: 'shopActiveCd', type: 'int' },  
        { name: 'serviceTypeCd', type: 'int' },  
        { name: 'articleNo', type: 'string' },  
        { name: 'articleNoOld', type: 'string' },  
        { name: 'articleDescription', type: 'string' },  
        { name: 'supportPackManufacturerCd', type: 'int' },  
        { name: 'supportPackRunTimeCd', type: 'int' },  
        { name: 'sLACd', type: 'int' },  
        { name: 'laborOnlyCd', type: 'int' },  
        { name: 'partsOnlyCd', type: 'int' },  
        { name: 'dmRCd', type: 'int' },  
        { name: 'postWarrantyCd', type: 'int' },  
        { name: 'indivCd', type: 'int' },  
        { name: 'simpleActivationCd', type: 'int' },  
        { name: 'indivTextCd', type: 'int' },  
        { name: 'nASCd', type: 'int' },  
        { name: 'onsiteEngineerCd', type: 'int' },  
        { name: 'sparePartCd', type: 'int' },  
        { name: 'hardwareCd', type: 'int' },  
        { name: 'sNCheck', type: 'string' },  
        { name: 'activationTypeCd', type: 'int' },  
        { name: 'systemPrice', type: 'float' },  
        { name: 'servicePrice', type: 'float' },  
        { name: 'systemType', type: 'string' },  
        { name: 'articleNoManufacturer', type: 'string' },  
        { name: 'remarks', type: 'string' },  
        { name: 'validBegin', type: 'date' },  
        { name: 'validEnd', type: 'date' },  
        { name: 'competence', type: 'string' },  
        { name: 'permission', type: 'string' },  
        { name: 'crtUser', type: 'string' },  
        { name: 'crtDate', type: 'date' },  
        { name: 'updUser', type: 'string' },  
        { name: 'updDate', type: 'date' }  
    ]  
});
```



Reflexion und Fazit

- ▶ Persönliche Einschätzung
- ▶ Alternativen
- ▶ Zukunft



Vielen Dank für Ihre Aufmerksamkeit!

Nun zur Demo . . .