

TYPO3: Extension für Google-Map mit festen Werten

von *Bernd Warken* bei *Fa. netcos AG*

0. Einleitung	3
<i>0.1 Vorwort</i>	3
<i>0.2 Lizenz</i>	3
1. Kennzahl für Google-Map	4
<i>1.1 Google Apis</i>	4
<i>1.2 Eintrag im TypoScript-Template</i>	4
2. Kickstarter	5
<i>2.1 Erstellung der Extension</i>	5
3. Quellcode anpassen	6
<i>3.1 Dateien ändern</i>	6

0. Einleitung

0.1 Vorwort

Dieses Dokument beschreibt die Hintergründe für die Erstellung der TYPO3-Extension **netcos_googlemap_fixed**. Dies ist eine offizielle Extension, die auf typo3.org erhältlich ist unter http://typo3.org/extensions/repository/view/netcos_googlemap_fixed/.

Diese Extension dient der Darstellung einer Google-Map innerhalb von TYPO3. Da mit festen Werten für Latitude, Longitude, Zoomlevel und Kartentyp gearbeitet wird, ist eine Eingabe beim Plugin nicht notwendig. Diese festen Werte werden eingestellt beim TypoScript-Template-Setup. Weniger gut, aber möglich, ist die direkte Änderung des Quellcodes.

0.2 Lizenz

Dieses Dokument gehört zum TYPO3-Projekt **netcos_googlemap_fixed**. Die aktuelle Version ist **2.0.2** vom 2. April 2008. Der Autor ist **Bernd Warken** bei Firma [netcos AG](#).

Alle Elemente des Projekts (dieses Dokument und der Quellcode) stehen unter der Lizenz **GPL (GNU General Public License) Version 3**. Der Originaltext dieser Lizenz ist erhältlich unter <http://www.gnu.org/licenses/gpl.html>, eine deutsche Übersetzung befindet sich unter <http://www.gnu.de/documents/gpl.de.html>.

Copyright 2007,2008 Bernd Warken

1. Kennzahl für Google-Map

Wenn man die Google-Map auf einer Domain benutzen will, muss man von Google eine Kennzahl für diese Domain besorgen.

1.1 Google Apis

Die Kennzahl wird bei <http://google.com/apis/maps> erstellt. Man muss dort eine URL für die Domain angeben und die Kennzahl damit erzeugen. Für jeden Domainnamen wird eine eigene Kennzahl mit etwa 80 Zeichen erstellt. Diese Kennzahl sollte man kopieren und sich merken. Wir speichern diese Kennzahl in der TypoScript-Konstanten:

```
google_map_key = ABQIAAAARokCoxWjWuwH4eEaCOyBkBQuBzwEBiC_X9c-oZXEt-cQgHLphTNVyn
```

Die hier notierte Kennzahl ist jedoch nicht vollständig, sie wurde auf Zeilenlänge gekürzt.

Wenn man seine TYPO3-Seite jedoch nur auf *localhost* ohne Domainnamen fahren möchte, so braucht man keine Kennzahl erstellen zu lassen. Jeder beliebige Wert ist passend für *localhost*.

1.2 Eintrag im TypoScript-Template

Die Werte des HTML-Headers lassen sich in TypoScript durch den Parameter **headerData** eingeben. Man muss hier ein JavaScript-Programm auf der Google-Website aufrufen. Zwei TypoScript-Befehle sind hierzu notwendig:

```
page.headerData.900 = TEXT
page.headerData.900.value (
    <script src="http://maps.google.com/maps?file=api&v=2&key={$google_map_key}"
        type="text/javascript">
    </script>
)
```

page bezieht sich hier auf das Objekt, das mit **PAGE** erzeugt wurde und auf **typeNum 0** läuft. Vielleicht heißt es bei Ihnen **seite**. Der Wert **900** ist willkürlich gewählt. Sie sollten nur darauf achten, dass dieser Wert nicht an einer anderen Stelle ebenfalls von **headerData** verwendet wird. **google_map_key** ist die Konstante, die die Kennzahl gespeichert hat.

2. Kickstarter

Der *kickstarter* ist das Werkzeug von TYPO3 zur Erstellung von Extensions.

2.1 Erstellung der Extension

Als erstes muss die Extension *kickstarter* installiert werden. Dadurch entsteht im Menü des *Extension Manager* der neue Eintrag *Neue Extension anlegen*, den wir anwählen.

Im entstehenden Formular gibt man zuerst den Namen im Feld *Enter extension key*: an, in diesem Fall **netcos_googlemap_fixed**. Bestätigung mit dem Button *Update...*

Im Folgenden wird die Optionsliste *KICKSTARTER WIZARD* im Formular bearbeitet. Man drückt zuerst das Pluszeichen von *General info*. Darin erstellt man den *Title* **Netcos googlemap fixed**. Als *Category* wählt man *Frontend Plugins*. *State* steht auf *Alpha*, die Extension ist jedoch mittlerweile im *Beta*-Zustand. Für *Dependencies* gibt man **cms** an. *Author* hinzufügen. Den Button *Update...* betätigen

Als nächstes das Pluszeichen von *Setup languages* und darin die Sprache *German* wählen. Bestätigung durch *Update...* Die Standardsprache der Extension ist immer Englisch, die gewählte deutsche Sprache ist eine Zusatzsprache.

Nun das Pluszeichen von *Frontend Plugins* betätigen. Man gibt *title*: **Netcos googlemap fixed** in beiden Sprachen an. Anklicken *Add icon to 'New Content Element' wizard*:. Update.

Die Eingabe ist jetzt abgeschlossen. Daher den Feldbutton *View result* wählen. Dort auf *WRITE* klicken. Die Extension wird dann auf **typo3conf/ext/netcos_googlemap_fixed/** abgespeichert.

Die Extension ist jetzt erstellt und könnte auf *Install Extensions* installiert werden. Im Frontend würde jedoch noch keine Karte angezeigt, da nur ein Standardtext ausgegeben wird, der vom *Kickstarter*-Wizard erzeugt wurde. Daher muss der Quellcode in den erstellten Dateien angepasst werden.

3. Quellcode anpassen

Die Hauptarbeit bei der Erstellung der Extension ist die Anpassung des Quellcodes. Sobald eine Datei im Quellcode (in **typo3conf/ext/netcos_googlemap_fixed/**) geändert wird, darf der *Kickstarter* nicht mehr verwendet werden, da er alle Dateien mit Defaultwerten überschreibt.

3.1 Dateien ändern

Es scheint, als ob im Quellcode die Dateien **wizard_form.*** in **typo3conf/ext/netcos_googlemap_fixed/doc/** gelöscht werden können.

Quellcodedateien einer Extension können im *Ext Manager* geändert werden. Dazu wird der Menüeintrag *Install extensions* gewählt. Da die erstellte Extension bereits im Verzeichnis **typo3conf/ext** gespeichert wurde, wird sie hier auch aufgeführt. Wir klicken auf den Namen der Extension. Dadurch entsteht ein neues Menü, in dem wir *Edit files* wählen. Alle Quellcodedateien der Extension werden nun dargestellt. Wenn man nun auf den Knopf *Edit file* neben dem Dateinamen klickt, wird ein Editor mit dieser Datei geöffnet, in dem man die Datei ändern kann.

Die Hauptdatei zur Änderung ist die Datei **pi1/class.tx_netcosgooglemapfixed_pi1.php**. In ihr wird der Bereich mit den Defaultwerten aus der Erstellung mit dem *Kickstarter*, die Funktion *main*, ersetzt. Der Ersetzungscode lautet

```

function main($content,$conf) {
    $this->conf=$conf;
    $this->pi_setPiVarDefaults();
    $this->pi_loadLL();

    $empty = array(' ' => '');

    $width = intval($this->conf['width']);
    if ($width == 0) { $width = 400; }

    $height = intval($this->conf['height']);
    if ($height == 0) { $height = 400; }

    $latitude = floatval(strtr($this->conf['latitude'], $empty));
    if ($latitude == 0) { $latitude = 48.1456; }

    $longitude = floatval(strtr($this->conf['longitude'], $empty));
    if ($longitude == 0) { $longitude = 11.6156; }

    $zoom_level = intval($this->conf['zoom_level']);
    if ($zoom_level == 0) { $zoom_level = 10; }

    $map_type = strtoupper(strtr($this->conf['map_type'], $empty));
    switch ($map_type) {
        case 'HYBRID':
        case 'NORMAL':
        case 'SATELLITE':
            break;
        default:
            $map_type = 'HYBRID';
            break;
    }
}

/*
    $width = 400;
    $height = 400;
    $latitude = 48.1456;
    $longitude = 11.6156;
    $zoom_level = 15;
    $map_type = HYBRID;
*/
*/
$content="
    <div id='map' style='width: " .
        $width . "px; height: " . $height . "px'>
    </div>

    <script type='text/javascript'>
        <![CDATA[
            if (GBrowserIsCompatible()) {
                var map = new GMap2(document.getElementById('map'));
                map.addControl(new GLargeMapControl());
                map.addControl(new GMapTypeControl());
                map.setCenter(new GLatLng(" . $latitude . "," .
                    $longitude . ")," . $zoom_level . ",G_ " .
                    $map_type . "_MAP");
            }
        </script>
        </script>
";

return $this->pi_wrapInBaseClass($content);
}

```

Der JavaScript-Code beruht auf der TYPO3-DVD *Einstieg in TYPO3 4.0* von Thomas Kötter aus

dem *Galileo-Verlag*.

Der erste Teil dieses Codes enthält Abfragen von TypoScript-Werten, die in *\$this->conf* zur Verfügung stehen.

Der mittlere Teil besteht aus einem Kommentar, der die 6 Werte der Konfiguration enthält. Wird dieser Kommentar aktiviert, so wird die TypoScript-Konfiguration überschrieben und außer Kraft gesetzt. Bei einer alten Version der Extension war dies der Standard. Jetzt wird jedoch die TypoScript-Konfiguration vorgezogen. Sie bleibt auch bei einem Update der Extension erhalten, während die Quellcode-Konfiguration bei jedem Update aufgefrischt werden muss.

Der letzte Teil des Codes enthält den Aufruf des Skripts. Dieser ist unverändert geblieben gegenüber den alten Versionen der Extension.