



Peppershop Direkt-Extern Modul

Anleitung

Datum

24. Oktober 2017

Version

1.5

Inhaltsverzeichnis

1. Einleitung.....	3
2. Installation.....	3
2.1 Systemanforderungen.....	3
2.2 Dateien kopieren und hochladen.....	3
2.3 Beschreibung.....	4
2.4 Anwendungsbeispiel externe Webseite.....	4
2.4.1 Aufrufe im CMS.....	4
2.5 API / Aufrufe.....	5
2.5.1 Komponenten (Login, Suche, Warenkorb, JavaScript, Session-Load, (Merkliste)).....	5
2.5.2 Suche: Direkte Quicksearch Suchanfrage.....	5
2.5.3 Kategorien.....	6
2.5.4 Kategorienbaum.....	7
2.5.5 Einzelartikel.....	8
2.5.6 Mehrere Artikel.....	9
2.5.7 Shop-Header.....	9
2.5.8 Hauptnavigation.....	9
2.5.9 Footer / Fusszeile.....	9
2.5.10 Session.....	10
3. Anhang.....	11
3.1 Beispielscript für Webseite.....	11
3.2 Aufruf von anderer Domain.....	12

PepperShop wird von Glarotech entwickelt und vertrieben. Seit 1998 ist das innovative Unternehmen im Internet tätig und auf E-Commerce spezialisiert. Sie als Kunde profitieren vom direkten Draht zu den Herstellern der Produkte.

Glarotech GmbH
Toggenburgerstrasse 156
CH-9500 Wil

info@glarotech.ch
Tel. +41 (0)71 923 08 58
www.glarotech.ch

1. Einleitung

Mit dem Direkt-Extern Modul kann man auf externen Webseiten einzelne Komponenten direkt aus dem Shop auslesen und anzeigen. So ist es zum Beispiel möglich, Artikel, ganze Artikellisten oder auch den Warenkorb auf einer Webseite anzuzeigen. Oft wird auch der Kategorienbaum des Shops ausgelesen, um z.B. in einer CMS Designintegration¹ im responsive Teil für das Mobile-Menü zu verwenden.

Aktuell bestehen folgende Möglichkeiten der Anbindung:

- Kategorien
- Kategorienbaum
- Einzelartikel
- Mehrere Artikel, optional gefiltert
- Shop-Header
- Hauptnavigation
- Footer (Fusszeile)
- Komponenten (Login, Suche, Warenkorb, JavaScript-Elemente, Session-Load/Sync)



2. Installation

2.1 Systemanforderungen

Um das Direkt-Extern Modul einzusetzen, ist mindestens ein PepperShop v.5.0 Professional erforderlich. Es empfiehlt sich ein möglichst aktueller Build.

2.2 Dateien kopieren und hochladen

Das Direkt-Extern Modul kann nach dem Bezug im Download Bereich des eigenen PepperShop Accounts heruntergeladen werden. Es besteht aus zwei Verzeichnissen, die man in seinen installierten PepperShop kopieren muss. Dazu entpackt man zuerst das ZIP-Archiv. Es sind folgende Dateien enthalten, die hier mit den Zielverzeichnissen dargestellt werden:

```
{shopdir}/shop/direct_extern.def.php
{shopdir}/shop/direct_extern_einzelartikel.def.php
{shopdir}/shop/direct_extern_kategorie.def.php
{shopdir}/shop/direct_extern_component.def.php
{shopdir}/shop/direct_extern_login.php
{shopdir}/shop/language/html_templates/direct_extern_einzelartikel.tpl.html
{shopdir}/shop/language/html_templates/direct_extern_kategorie.tpl.html
{shopdir}/shop/language/html_templates/direct_extern_kategorie_special.tpl.html
{shopdir}/shop/language/html_templates/directextern_kategorien.tpl.html
```

1: Siehe auch Modul Design-Connector: <http://qur.ch/gh>

```
{shopdir}/shop/language/html_templates/directextern_kategorien_xml.tpl.html  
{shopdir}/shop/language/html_templates/directextern_multi_artikel.tpl.html  
{shopdir}/shop/language/html_templates/direct_extern_component.tpl.html
```

Die Dateien lassen sich einfach via FTP zum Webserver hochladen. Der Platzhalter `{shopdir}` steht dabei für das Webshop-Verzeichnis auf dem Webserver, wo unter anderem die Datei `index.php` zusammen mit `README.txt` zu finden ist.

2.3 Beschreibung

Die Shop-Komponenten werden über eine definierte Webadresse aus dem Shop ausgelesen. Der Aufbau der Adresse ist immer gleich:

```
Aufruf-URL: http://.../shop/direct_extern.def.php?op={op}&{zusatzparameter}
```

Der Parameter `op` definiert, welche Ausgabe gemacht wird. Mit den Zusatzparametern können dazu noch weitere Einstellungen übergeben werden. Nun folgt eine detaillierte Auflistung der möglichen Aufrufe und deren Parameter. Je nach Projekt werden nur einige der vorhandenen Parameter eingesetzt.

Es wird empfohlen, dass die Ausgaben der Aufrufe jeweils per AJAX geladen werden, womit auch die Session-Zuordnung automatisiert funktioniert. Das standardmässig verwendete Format der Informationen ist JSON. In Kapitel 3 wird ein einfaches Beispiel hierzu gezeigt.

2.4 Anwendungsbeispiel externe Webseite

Als praktisches Beispiel wird davon ausgegangen, dass zu einer Webseite ein Shop angebunden wird. Damit die Integration möglichst transparent für den Kunden wirkt, soll das Design der Webseite in den Shop übernommen werden. Dabei soll der Warenkorb und die Suche auf der Webseite integriert werden.

2.4.1 Aufrufe im CMS

Damit diese Komponenten im CMS angezeigt werden können, sind folgende Operationen notwendig:

- Komponenten: `op=shc`
- Kategorien: `op=skb&format=json`

Mit den Komponenten werden das Login, die Suche, der Warenkorb, das Javascript und (falls als Modul installiert) die Merkliste eingebunden.

Der Aufruf für die Kategorien ist notwendig, damit der Kategorienbaum des Shops im Mobile-Menü der Webseite erweitert wird.

Per AJAX kann der Kategorienbaum ausgelesen werden und das JSON direkt dem Mobile-Menü hinzugefügt werden, wenn die Shop-Kategorien darin enthalten sein sollen. Dasselbe gilt für die Komponenten, natürlich an entsprechend passender Stelle eingefügt.

2.5 API / Aufrufe

Nachfolgend werden die einzelnen Möglichkeiten aufgelistet und die Parameter, welche für die Anwendungen notwendig sind, beschrieben.

2.5.1 Komponenten (Login, Suche, Warenkorb, JavaScript, Session-Load, (Merkliste))

Einzelne Shop-Komponenten werden in einen Aufruf exportiert. Die Komponenten sind in einem JSON-Array mit direktem HTML-Code gelistet:

- session-load
- login
- quicksearch
- warenkorb
- javascript
- merkliste (falls Modul installiert)

Anmerkungen:

Für die Suchfunktion (Quicksearch) muss noch das Javascript und das `jquery-ui.css` CSS eingebunden werden.

Aufruf	<code>http://.../shop/direct_extern.def.php?op=shc&{zusatzparameter}</code>
Zusatzpar.	<ul style="list-style-type: none"> • <code>extsid={Externe Session-ID}</code> (optional) • für empfohlenen AJAX-Aufruf werden keine Parameter benötigt
Template	<p>Folgende Template-Dateien werden verwendet um die Ausgabe zu rendern / formatieren. Der Pfad bis zu den Dateien lautet: <code>{shop_dir}/shop/language/html_templates/</code></p> <ul style="list-style-type: none"> • login: <code>kundenaccount.tpl.html</code> • quicksearch: <code>quicksearch.tpl.html</code> • warenkorb: <code>warenkorb_kurzinfo.tpl.html</code>

2.5.2 Suche: Direkte Quicksearch Suchanfrage

Verwenden Sie die Schnellsuche des PepperShop Systems, um deren Resultate in Ihre CMS-Suchresultate einzubinden (Info: Diese Abfrage steht erst ab PepperShop v.6.0.0 Build 32 bereit):

Aufruf	<pre>http://.../shop/ajax_handler.php? ajax_handler_id=artikelsuche_search_name &fieldname=Suchstring &alt_label=Name &deeplinks=true &only-article=false &image=true &sprachwahl={sprachwahl} &term={Suchtext}</pre>
Zusatzpar.	<ul style="list-style-type: none"> • <code>term</code> = Ihr Suchstring (urlencodiert) • <code>sprachwahl</code> = Optionale Angabe der Sprache (de, en, fr, it, ..) • <code>only-article</code> = true false: Nur Artikel oder auch Kategorien? • <code>image</code> = true false: Links zu Bildern ausgeben?

Template	<p>Es wird direkt JSON Code retourniert, bei einem Fehler wird eine Fehlermeldung retourniert.</p> <p>Beispiel Rückmeldung (Info: value steht für die interne Shop Artikel-/Kategorie-ID):</p> <pre>[{ "value": "14", "label": "Pfeffer", "category": "Kategorien", "deeplink": "http://demo.peppershop.com/ki.php/pfeffer.html" }, { "value": "67", "label": "PepperShop Pfefferschote", "category": "Artikel", "image": "http://demo.peppershop.com/shop/ProdukteBilder/67_kl.jpg", "deeplink": "http://demo.peppershop.com/pi.php/PepperShop-Pefferschote.html" }, { "value": "66", "label": "Wo der Pfeffer w\u00e4chst (Taschenbuch)", "category": "Artikel", "image": "http://demo.peppershop.com/shop/ProdukteBilder/pfeffer002_kl.jpg", "deeplink": "http://demo.peppershop.com/pi.php/Wo-der-Pfeffer-waechst-Taschenbuch.html" }, { "value": "65", "label": "Teufelsk\u00e4se. H\u00f6llisch scharfe Sachen. (Taschenbuch)", "category": "Artikel", "image": "http://demo.peppershop.com/shop/ProdukteBilder/65_kl.jpg", "deeplink": "http://demo.peppershop.com/pi.php/Teufelskueche-Hoellisch-scharfe-Sachen-Taschenbuch.html" }]</pre>
-----------------	--

2.5.3 Kategorien

Es wird der HTML-Code für eine bestimmte Kategorie und deren Artikel exportiert.

Es gibt hier jeweils einen Darstellungstyp ('normal' oder 'special'):

- **normal:** Die Kategorie rendert in einer Schleife die Artikel, welche ebenfalls im normal-Modus gerendert werden (einstufig, ohne Blättern)
- **special:** Hier wird das `direct_extern_kategorie_special.tpl.html` angewendet, welches direkt zur Artikel Renderingfunktion durchgereicht wird, das heisst, dass man in einem Template die Anpassung aller Darstellungsoptionen für die Kategorie und deren Artikel übernimmt, dies ermöglicht speziellere Darstellungen.

Aufruf	<code>http://.../shop/direct_extern.def.php?op=sk1&{zusatzparameter}</code>
Zusatzparameter	<ul style="list-style-type: none"> • entweder: <code>Kategorie_Nr={Kategorie_Nr}</code> • oder: <code>Kategorie_ID={Kategorie_ID}</code>
Template	<p>Folgende Template-Dateien werden verwendet um die Ausgabe zu rendern / formatieren. Der Pfad bis zu den Dateien lautet: <code>{shop_dir}/shop/language/html_templates/</code></p> <ul style="list-style-type: none"> • Darstellungstyp 'normal': <code>direct_extern_kategorie.tpl.html</code> • Darstellungstyp 'special': <code>direct_extern_kategorie_special.tpl.html</code>

Artikelfilter	<p>Falls nicht alle Artikel dieser Kategorie angezeigt werden sollen, kann man beliebig viele Filterregeln auf die Daten eines Artikelobjekts anwenden. Dazu wird ein Filterarray mit enthaltenen Filterregeln in <code>direct_extern.def.php</code> erzeugt und bei der Instanzierung des jeweiligen Objekts dem Konstruktor mit übergeben.</p> <p>Eine Filterregel besteht aus einem Array mit drei Elementen (assoziativ):</p> <ul style="list-style-type: none"> • <code>membervar</code>: Name der Membervariable des jeweiligen Artikelobjekts (z.B. <code>artikel_Nr</code>, <code>Zusatzfeld_5</code>, ...) • <code>operation</code>: Eine Vergleichsoperation mit zwei Operanden, z.B. <code><</code>, <code>></code>, <code><=</code>, <code>>=</code>, <code>==</code> • <code>operand</code>: Vergleichener Wert - meist ein vorgegebener Wert <p><i>Beispiel:</i></p> <pre>\$filter = array(array('membervar' => 'Zusatzfeld_1', 'operation' => '==', 'operand' => 'V', 'special' => 'ignore_case'));</pre>
----------------------	---

2.5.4 Kategorienbaum

Der HTML-Code oder JSON-Code für den Kategorienbaum wird exportiert.

Aufruf	<code>http://.../shop/direct_extern.def.php?op=skb&{zusatzparameter}</code>
Zusatzparameter	<ul style="list-style-type: none"> • <code>Kategorie_Nr={Kategorie_Nr}</code> (optional) Standard ist 0, ganzer Baums • <code>deepness={deepness}</code> (optional) Standard ist 1 • <code>format={format}</code> (optional) Standard ist html <ul style="list-style-type: none"> ◦ mögliche Werte: html, xml, json • <code>extsid={Externe-Session-ID}</code> (optional)
Template	<p>Folgende Template-Dateien werden verwendet um die Ausgabe zu rendern / formatieren. Der Pfad bis zu den Dateien lautet: <code>{shop_dir}/shop/language/html_templates/</code></p> <ul style="list-style-type: none"> • Darstellungstyp 'html': <code>directextern_kategorien.tpl.html</code> • Darstellungstyp 'xml': <code>directextern_kategorien_xml.tpl.html</code> • Darstellungstyp 'json': Hat kein Template.

Beispiel JSON-Format:

```
{
  "category-entry": [
    {
      "name": "Pfeffer",
      "class": "kat_inaktiv",
      "href": "SHOPDIR/ki.php/Pfeffer-14.html"
    },
    {
      "name": "DVD",
      "class": "kat_inaktiv",
      "href": "SHOPDIR/ki.php/DVD-13.html"
    },
    {
      "name": "Pizzeria",
```

```

"class": "kat_inaktiv",
"href": "SHOPDIR/ki.php/Pizzeria-18.html",
"categories": {
  "category-entry": [
    {
      "name": "Pizza",
      "class": "kat_inaktiv",
      "href": "SHOPDIR/ki.php/Pizza-19.html"
    },
    {
      "name": "Spezial-Pizza",
      "class": "kat_inaktiv",
      "href": "SHOPDIR/ki.php/Spezial-Pizza-20.html"
    }
  ]
}
}
]
}
}

```

2.5.5 Einzelartikel

Der HTML-Code für einen einzelnen Artikel wird exportiert.

Einschränkungen:

- keine Mindestbestellmenge berücksichtigt
- keine Staffelpreise
- keine Standard PepperShop Variationen
- keine Standard PepperShop Zusatztexte
- keine Subartikel
- keine Cross-Selling Anzeige (manuell + automatisch)
- keine Artikelzusatzfelder
- keine Artikel mit Preis gleich 0.00

Aufruf	http://.../shop/direct_extern.def.php?op=sa&{zusatzparameter}
Zusatzparameter	<ul style="list-style-type: none"> • Artikel_Nr={Artikel_Nr} • Kategorie_Nr={Kategorie_Nr} (optional) • extsid={Externe Session-ID} (optional)
Template	<p>Folgende Template-Datei wird verwendet um die Ausgabe zu rendern / formatieren. Der Pfad bis zu der Datei lautet: {shop_dir}/shop/language/html_templates/</p> <ul style="list-style-type: none"> • direct_extern_einzelartikel.tpl.html

2.5.6 Mehrere Artikel

Der HTML-Code für eine Artikelliste wird exportiert. Es wird maximal die Anzahl Produkte aufgelistet und bei Promo diese prioritär. Die Ausgabe wird während 1 Stunde zwischengespeichert.

Aufruf	http://.../shop/direct_extern.def.php?op=sma&{zusatzparameter}
Zusatzparameter	<ul style="list-style-type: none"> • Kategorie_Nr={Kategorie_Nr} • anzahl={anzahl} (optional) Standard ist 3 • extsid={Externe Session-ID} (optional)
Template	Folgende Template-Datei wird verwendet um die Ausgabe zu rendern / formatieren. Der Pfad bis zu der Datei lautet: {shop_dir}/shop/language/html_templates/ <ul style="list-style-type: none"> • directextern_multi_artikel.tpl.html

2.5.7 Shop-Header

Der HEAD-Bereich des Shops wird als HTML exportiert.

Aufruf	http://.../shop/direct_extern.def.php?op=shh&{zusatzparameter}
Zusatzparameter	<ul style="list-style-type: none"> • title={Seitentitel} Standard Shopname • extsid={Externe Session-ID} (optional)
Template	Folgende Template-Datei wird verwendet um die Ausgabe zu rendern / formatieren. Der Pfad bis zu der Datei lautet: {shop_dir}/shop/language/html_templates/ <ul style="list-style-type: none"> • hauptseite.tpl.html

2.5.8 Hauptnavigation

Die ganze Hauptnavigation wird als HTML exportiert.

Aufruf	http://.../shop/direct_extern.def.php?op=shn&{zusatzparameter}
Zusatzpar.	<ul style="list-style-type: none"> • extsid={Externe Session-ID} (optional)
Template	Folgende Template-Dateien werden verwendet um die Ausgabe zu rendern / formatieren. Der Pfad bis zu den Dateien lautet: {shop_dir}/shop/language/html_templates/ <ul style="list-style-type: none"> • direct_extern_component.tpl.html • hauptnavigation.tpl.html

2.5.9 Footer / Fusszeile

Der Footer wird als HTML exportiert.

Aufruf	http://.../shop/direct_extern.def.php?op=sf&{zusatzparameter}
Zusatzpar.	<ul style="list-style-type: none"> • extsid={Externe Session-ID} (optional)
Template	Folgende Template-Dateien werden verwendet um die Ausgabe zu rendern / formatieren. Der Pfad bis zu den Dateien lautet: {shop_dir}/shop/language/html_templates/ <ul style="list-style-type: none"> • direct_extern_component.tpl.html • footer.tpl.html

2.5.10 Session

Info: Wird die Integration von Komponenten via empfohlenem AJAX-Aufruf getätigt, kann dieses Kapitel übersprungen werden.

Das Session-Management unterstützt die Zuordnung einer externen Session-ID (extsid) zu einer internen Session. Wenn man den Shop also ohne Session-Cookie mit nur `?extsid=xyz` aufruft und diese externe Session-ID ist zuvor an eine existierende Session gebunden worden, so erhält man dieselbe Session wieder. Umgekehrt wird bei der Erstellung einer neuen Session ebenfalls die externe Session-ID direkt verknüpft. Eine externe Session-ID kann immer nur einer einzigen PepperShop Session(-ID) zugewiesen werden

Abläufe:

Externe Ressource ist **innerhalb** des Shop-Verzeichnisses (PEPPERSESS Cookie ist sichtbar für externe Ressource)

Shop wird zuerst aufgerufen

- Shop wird mit neuem Browser aufgerufen (erhält PEPPERSESS=a)
 - Kunde legt Artikel in Warenkorb (PEPPERSESS=a)
 - Kunde wechselt zu externer Seite (PEPPERSESS=b + extsid=1) + JS/iFrame Call: (`op=s`: Via Cookie PEPPERSESS=a + extsid=1)
 - Synchronisation abgeschlossen. PEPPERSESS b wird nicht mehr verwendet und wird entsorgt werden
- Externe Seite wird zuerst aufgerufen
 - Kunde ruft externe Seite auf (PEPPERSESS=a + extsid=1) + JS/iFrame Call: (PEPPERSESS=b + extsid=1)
 - Synchronisation abgeschlossen. PEPPERSESS b wird nicht mehr verwendet und wird entsorgt werden

Externe Ressource ist **ausserhalb** des Shop-Verzeichnisses (PEPPERSESS Cookie ist *nicht sichtbar* für externe Ressource!)

- Shop wird zuerst aufgerufen
 - Shop wird mit neuem Browser aufgerufen (erhält PEPPERSESS=a)
 - Kunde legt Artikel in Warenkorb (PEPPERSESS=a)
 - Kunde wechselt zu externer Seite (PEPPERSESS=b + extsid=1) + JS/iFrame Call: (`op=s`: "Magic" PEPPERSESS=a + extsid=1)
 - Synchronisation abgeschlossen. PEPPERSESS b wird nicht mehr verwendet und wird entsorgt werden
- Externe Seite wird zuerst aufgerufen
 - Kunde ruft externe Seite auf (PEPPERSESS=a + extsid=1) + JS/iFrame Call: (PEPPERSESS=a + extsid=1)
 - Synchronisation abgeschlossen.

Bitte beachten:

- Beim ersten Aufruf der externen Ressource ist die Synchronisation noch nicht ausgeführt und folglich wird der Warenkorbinhalt noch nicht korrekt angezeigt. Mögliche Lösung: Warenkorb dy-

namisch nachladen (warenkorb_kurzinfo.def.php, z.B. via `get_hauptnavigation()` in `direct_extern_component.def.php`)

- Externe Ressourcen werden nicht gecached (Session-Sync Calls gibts nur einmal – first call)

Aufruf	<code>http://.../shop/direct_extern.def.php?op=s&{zusatzparameter}</code>
Zusatzpar.	<ul style="list-style-type: none"> • <code>extsid={Externe Session-ID}</code> (optional)
Template	Keine sichtbare Ausgabe, nur Server-to-Server Kommunikation.

3. Anhang

3.1 Beispielscript für Webseite

Das folgende Beispiel PHP-Skript zeigt die Aufrufe und Anwendung einer typischen Integration von Shop-Komponenten in eine bestehende Webseite per AJAX.

Es werden per Aufruf (`http://.../shop/direct_extern.def.php?op=shc`) die Komponenten im JSON-Format geladen und danach in den Container mit der ID „results“ eingeordnet.

Testweise empfiehlt es sich, im entsprechenden Shop einen Artikel in den Warenkorb zu legen, sich als Kunde einzuloggen oder einen gültigen Suchbegriff abzusetzen.

```

<?php
// Modul: Direkt Externe Einbindung / Vorlage
//
// Autoren: Jose Fontanil & Reto Glanzmann & Roland Bruehwiler & Glarotech GmbH
//
// Zweck: Vorlagedatei:
//   - Proxysystem um PepperShop Inhalte zu rendern
//   - (Dieses Beispiel benoetigt: php.ini: allow_url_fopen=true)
// -----

// *****
// KONFIGURATION
$shop_verzeichnis = 'http://DOMAIN/SHOPVERZEICHNIS'; // Basis-URL fuer Calls (mit Trailing-Slash!)
// *****

?>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>CMS Vorlage</title>

<!-- Bootstrap core CSS -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css">
<script src="//code.jquery.com/jquery-1.11.3.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js"></script>
<link rel="stylesheet" href="//code.jquery.com/ui/1.11.4/themes/smoothness/jquery-ui.css">
<style>
.navbar {
border-radius: 0;
border-bottom: 4px solid #555555;
color: #FFFFFF;
}
.navbar-header .navbar-brand {
padding-top: 15px;
}
.footer {
border-top: 1px #DDDDDD solid;
margin-top: 20px;
}
</style>
</head>
<body>
<nav class="navbar navbar-inverse">
<div class="container">
<div class="navbar-header">
<button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#navbar" aria-expanded="false" aria-controls="navbar">
<span class="sr-only">Toggle navigation</span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>

```

```
<span class="icon-bar"></span>
</button>
<a class="navbar-brand" href="#">Webseite</a>
</div>
<div id="navbar" class="collapse navbar-collapse">
<ul class="nav navbar-nav">
<li class="active"><a href="#">Home</a></li>
<li><a href="#">Menu 1</a></li>
<li><a href="#">Menu 2</a></li>
<li><a href="#">Webshop</a></li>
</ul>
</div><!--/.nav-collapse -->
</div>
</nav>

<div class="container">
Hier Inhalt CMS<br>
Ajax-Load:<br>
<script>
$.ajax({
url: "<?php echo $shop_verzeichnis.'shop/direct_extern.def.php?op=shc';?>",
cache: false
})
.done(function( json_code ) {
var json_obj = jQuery.parseJSON(json_code);
$("#results").append(json_obj.warenkorb);
$("#results").append(json_obj.login);
$("#results").append(json_obj.javascript); // Fuer Quicksearch
$("#results").append(json_obj.quicksearch);
});
</script>
<div id="results">
</div>
</div>
<!-- Site footer -->
<div class="container">
<footer class="footer">
<p>Footer</p>
</footer>
</div>
</body>
</html>
```

3.2 Aufruf von anderer Domain

Wenn obiger Beispiel-Aufruf auf einer unterschiedlichen Domain durchgeführt wird, kommt die CORS Restriktion zum Einsatz. Fügen Sie in diesem Fall folgenden Code in die `.htaccess`-Datei auf dem Server ein, wo der PepperShop installiert ist:

```
Header set Access-Control-Allow-Origin "http://ANFRAGENDE DOMAIN"
#damit Cookies uebertragen werden können
Header set Access-Control-Allow-Credentials true
```

Die „ANFRAGENDE DOMAIN“ ist entsprechend der Server bzw. die URL gemeint, woher der AJAX-Aufruf abgesetzt wird.