

AZ Eingabeassistent

Technical Whitepaper



Version 1.10

23. Juli 2020

AZ Direct

Identifikation

Version	1.10
Datum	23.07.2020
Status	Final
Dokumentenname	AZ Eingabeassistent Technical Whitepaper.V1.10

Versionshistorie

Version	Datum	Verfasser	Änderung
1.0	07.02.2013	poep04	Initialerstellung
1.1	20.02.2013	fahl001	Testschnittstelle beschrieben, Hinweise für Mobilgeräte
1.2	16.05.2013	poep04	Namensänderung AZ Eingabeassistent
1.3	04.12.2013	scr04	Hinweis zur Registrierung
1.4	09.01.2014	poep04	Link für mobile Demoseite ergänzt
1.5	14.04.2014	poep04	Dokumentation von requestMapping ergänzt
1.6	16.04.2015	scr04	Fehler im Anhang Quellcode korrigiert
1.7	09.09.2015	poep04	Glossar ergänzt
1.8	04.03.2016	scr04	Ungültige Codierung im Anhang Quellcode beseitigt.
1.9	26.04.2017	poep04	Dokumentation des Parameters requestStyle ergänzt
1.10	23.07.2020	scr04	Alternative Verwendung ohne jQuery

Inhaltsverzeichnis

1	EINLEITUNG.....	4
1.1	Zweck des Dokuments	4
1.2	Was ist der AZ Eingabeassistent?	4
1.3	Quellcode, Umfang und Vorlagen	5
2	INTEGRATIONSANLEITUNG.....	6
2.1	JavaScript	6
2.2	HTML-Anpassungen	8
2.3	Registrierung bei der AZ Direct	10
2.4	Registrierung HTML Seite	11
2.5	AZ Eingabeassistent in Verbindung mit mobilen Endgeräten	12
3	KONFIGURATIONSDetails	13
3.1	JavaScript	13
4	ANHANG QUELLCODE.....	15
5	ANHANG GLOSSAR	16

1 Einleitung

1.1 Zweck des Dokuments

Dieses Dokument soll die Einbindung vom AZ Eingabeassistenten in eine Kundenwebseite beschreiben. Die Zielgruppe dieses Dokumentes sind deshalb Personen mit technischem Hintergrund. Grundkenntnisse von HTML und JavaScript werden vorausgesetzt. Der Leser des Dokumentes sollte anschließend in der Lage sein, den AZ Eingabeassistenten in seine Webseite zu integrieren und nach seinen Bedürfnissen zu konfigurieren.

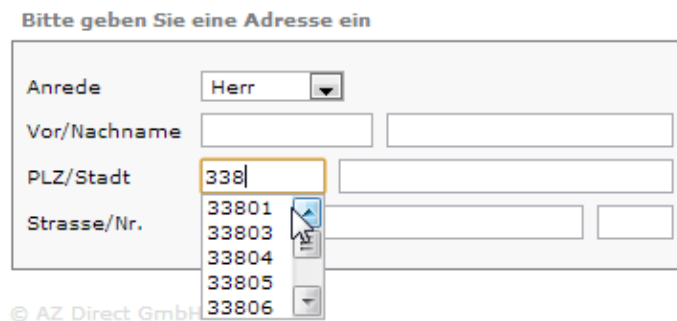
1.2 Was ist der AZ Eingabeassistent?

In Webseiten gibt es häufig Situationen, in denen Webseitenbesucher aufgefordert werden, ihre Adressen einzutragen. Dabei besteht die Möglichkeit, dass durch unterschiedliche Schreibweise von Adressen und Tippfehler nicht standardisierte bzw. inkorrekte Adressen übermittelt werden. Der AZ Eingabeassistent unterstützt bei der Eingabe von Adressen in Webformularen.

Für den **Webseitenbesucher** bedeutet das, dass er bereits bei der Eingabe von einigen Ziffern einer PLZ eine Auswahlliste aller gültigen PLZ-Nummern erhält. Wenn sich aus der PLZ eindeutig die Stadt ermitteln lässt, wird diese bereits beim Verlassen des PLZ Feldes im Stadt-Feld dargestellt. Ist der Name der Stadt nicht eindeutig (z.B. bei der PLZ 18581), dann wird ein Optionsmenü mit allen Orten dargestellt. Auch bei der Eingabe des Straßennamens wird nach nur wenigen Buchstaben eine Liste aller gültigen Straßennamen dargestellt.

Da bekannterweise „PROBIEREN ÜBER STUDIEREN“ geht, schlage ich dem Leser dieses Dokumentes vor, kurz auf folgende Webseite der AZ Direct zu gehen, um sich das Verhalten anzuschauen:

<http://www.az-eingabeassistent.de/>



Ganz wichtig zu erwähnen ist dabei: Sollte der Webseitenbesucher einmal einen Straßennamen eingeben wollen, den der AZ Eingabeassistent nicht kennt, ist das weiterhin jederzeit möglich.¹

Für den **Webseitenbetreiber** hat der AZ Eingabeassistent insbesondere den Vorteil, dass es sich mit sehr wenig Aufwand integrieren lässt. Außerdem erfolgen Datenupdates² ausschließlich auf Seiten der AZ Direct.

1.3 Quellcode, Umfang und Vorlagen

Eine einfache Webseite mit einem Eingabeformular für PLZ, Ort, Straße und Hausnummer lässt sich bereits mit ca. 30 Zeilen HTML aufbauen. Um in dieses Formular den AZ Eingabeassistenten zu integrieren sind lediglich ein paar zusätzliche Tags und ca. 30 Zeilen JavaScript erforderlich. Was genau in den Zeilen anzupassen ist und wie Sie die Funktionalität in Ihre Seite integrieren können, das soll in den nächsten Seiten erläutert werden.

Einen Beispiel-Quellcode können Sie sich jederzeit ansehen, in dem Sie auf die oben bereits erwähnte Webseite gehen und dort einfach mit der Maus über das Eingabeformular gehen und im Menü³ „Quelltext anzeigen“ aufrufen. Eine schlichte Version eines Webformulars befindet sich auch im Anhang.

¹ Grundsätzlich werden die Adressdaten einmal monatlich aktualisiert. Es kann aber sein, dass zum Beispiel eine Straße – etwa, weil es sich um ein Neubaugebiet handelt – noch nicht im Adressbestand ist und erst beim der nächsten Aktualisierung aufgenommen wird.

² In Deutschland gibt es pro Jahr durchaus mehrere tausend Änderungen an Straßen- und Ortsnamen sowie Postleitzahlen.

³ Der Text des Menüeintrages ist je nach Browser unterschiedlich.

2 Integrationsanleitung

2.1 JavaScript

Der AZ Eingabeassistent verwendet die weit verbreitete JavaScript Library „jQuery“. Falls Sie diese nicht bereits selber in Ihrer Webseite eingebunden haben, müssen Sie das zuerst im `<head>`-Tag tun:

```
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js" type="text/javascript"></script>
```

Hierbei muss es sich nicht unbedingt um die aktuellste Version dieser Library handeln. Der AZ Eingabeassistent funktioniert auch mit älteren Versionen ohne Probleme.

Als nächstes sollten Sie folgenden JavaScript Quellcode in Ihrer Webseite einbinden:

```
<script type="text/javascript">
  var service = 'AddressService';
  var requestMapping = 'integ.az-direct.com/asp/as'; // Integration AZ
  var speed = 'fast';
  var maxSelectSize = 5;
  var minLengthPlz = 3;
  var minLengthOrt = 0;
  var minLengthStr = 1;

  // startup initialization
  $(document).ready(function() {
    var group = 1;

    // adding various events to the input fields
    addInputFieldEvents(group, 'PlzField', 'OrtField', 'StrField');

    // adding various events to the select boxes
    addSelectBoxEvents(group, 'PlzField', 'OrtField', 'StrField');

    // initialize field styling selection boxes
    setSelectBoxStyles(group, 'PlzField', 'OrtField', 'StrField');
  });
</script>
```

In den hier dargestellten JavaScript Zeilen lässt sich die Funktionalität vom AZ Eingabeassistent weiter konfigurieren.

Der Link „integ.az-direct.com/asp/as“ ist die Integrations-Schnittstelle der AZ Direct, über die die AZ Eingabeassistent Kommunikation stattfindet.

Mit der Einstellung `minLengthPlz=3` legen Sie fest, dass nach 3 Ziffern der PLZ eine Vorschlagsliste angezeigt werden soll, die anderen `minLengthXxx` Einstellungen funktionieren analog.

Die Variable `service` definiert die Art des genutzten Service. Der Wert `'AddressService'` bietet die volle Funktionalität vom AZ Eingabeassistenten, benötigt allerdings eine Freischaltung. Der Testservice ist mit `'AddressMockup'` erreichbar.

Hierzu mehr unter *Registrierung bei der AZ Direct*.

In den anschließenden Aufrufen der JavaScript – Gruppierungsfunktionen (addInputFieldEvents, ...) werden sogenannte Gruppen gebildet. Sie haben beispielsweise auch die Möglichkeit eine Rechnungs- und Lieferadresse mit dem AZ Eingabeassistenten aufzubauen. Dann können Sie die zusammengehörenden Felder (Rechnungs-PLZ, Rechnungs-Ort und Rechnungs-Straße) hierüber verbinden. Dazu später noch mehr.

Jetzt sollten sie noch die beiden folgenden Zeilen einbinden:

```
<script src=https://www.az-direct.com/asp/as.js
  type="text/javascript"></script>
```

Diese Zeile ist für die Anmeldung verantwortlich. D.h. beim Aufruf dieses kleinen Scriptes wird überprüft, ob die Webseite bereits freigeschaltet ist (siehe weiter hinten) und es wird eine Art AZ Eingabeassistent Session-ID erzeugt, die für die weitere Verwendung benötigt wird. Da diese Zeile schon einiges an Kommunikation verursacht sollte sie nur auf den Seiten enthalten sein, die den AZ Eingabeassistenten wirklich verwenden.

```
<script src=http://oak.az-direct.com/oak/js/addressServices.js
  type="text/javascript"></script>
```

Diese Zeile bindet die AZ Eingabeassistent JavaScript Library ein, die für die Kommunikation zwischen dem Browser und dem AZ Eingabeassistenten verantwortlich ist. Diese Library kann direkt auf Ihrem Server abgelegt werden, oder alternativ mit folgender URL eingebunden werden:

<http://oak.az-direct.com/oak/js/addressServices.js>

Da es vermehrt Anfragen von Entwicklern gegeben hat, wie man den AZ Eingabeassistent ohne die „jQuery“ JavaScript Library verwenden kann gibt es mittlerweile eine entsprechende Alternative für diese Kommunikation. Diese kann alternativ zu der obigen Library verwendet werden, ohne das weitere Anpassungen notwendig sind:

<http://oak.az-direct.com/oak/js/addressServices-native.js>

2.2 HTML-Anpassungen

Als nächstes müssen Sie die 4 Eingabefelder in HTML aufbauen. Das kann etwas komplizierter werden, weil Sie auf Ihrer Seite sicherlich keine so schlichte Darstellung auswählen werden, wie wir sie hier im Dokument verwenden. Wir hoffen aber, dass dadurch die Erklärung einfacher ist. Anbei also ein Screenshot von einer sehr einfachen HTML-Eingabemaske.

PLZ/Stadt	<input type="text"/>	<input type="text"/>
Strasse/Nr.	<input type="text"/>	<input type="text"/>

Und nun zum dazugehörigen HTML-Quellcode.

```
<form>
<input type="hidden" id="LkzField1" value="de"/>
<table>
<tr>
<td>PLZ/Stadt</td>
<td>
<select id="SelectPlzField1" style="position:absolute;display:none;z-index:1000"
<input id="PlzField1" type="text" lkz="LkzField1" style="width: 65px;" t
<select id="SelectOrtField1" style="position:absolute;display:none;z-index:1000"
<input id="OrtField1" type="text" lkz="LkzField1" plz="PlzField1" style=
</td>
</tr>
<tr>
<td>Strasse/Nr.</td>
<td>
<select id="SelectStrField1" style="position:absolute;display:none;z-index:1000"
<input id="StrField1" type="text" lkz="LkzField1" plz="PlzField1" style=
<input id="HnrField" type="text" lkz="LkzField1" plz="PlzField1" style=
</td>
</tr>
</table>
</form>
```

Der HTML-Quellcode ist am rechten Rand zwar abgeschnitten, der grundsätzliche Aufbau lässt sich so aber besser erläutern. Wir haben in dieser Maske auf die Auswahl eines Landes verzichtet. Das Land muss in dem Hidden-Feld mit der id="LkzField1" eingestellt werden.

Obwohl Sie in der Maske oben nur 4 Eingabefelder sehen, werden in dem HTML-Quellcode insgesamt 7 Eingabelemente definiert. Das liegt daran, dass teilweise ein Textfeld (<input type="text"...>) und ein OptionMenü (<select>) übereinanderliegen und je nach Bedarf in den Vorder- bzw. Hintergrund verschoben werden. Dass das OptionMenü zuerst unsichtbar ist, erkennen Sie an dem Style „display=none“.

Wir empfehlen hier, die entsprechenden Style-Einstellungen (position, z-index) so zu übernehmen, weil ansonsten die HTML Elemente eventuell nebeneinanderliegen. Keine Sorge: die Verwaltung, welches Element gerade sichtbar ist oder nicht, wird komplett vom AZ Eingabeassistenten übernommen.

Wichtig bei der Betrachtung des HTML-Quellcodes sind die Namen und die IDs, der Elemente. Diese sind wie folgt definiert:

- PlzField<Gruppennummer> und SelectPlzField<Gruppennummer> definieren das Postleitzahl-Feld.
- OrtField<Gruppennummer> und SelectOrtField<Gruppennummer> definieren das Orts-Eingabefeld.
- StrField<Gruppennummer> und SelectStrField<Gruppennummer> definieren das Straßen-Eingabefeld.

Jetzt wird die Verbindung zu den **JavaScript – Gruppierungsfunktionen** hoffentlich klar. D.h. wenn Sie Ihrem Besucher erlauben wollen, eine Rechnungs- und eine Lieferadresse anzugeben, dann definieren Sie einfach, dass die Rechnungsadresse die Gruppennummer 1 bekommt und die Lieferadresse die Gruppennummer 2.

Im JavaScript Bereich geben Sie dann folgendes an:

```
var rechnung = 1;
var liefer = 2;

addInputFieldEvents(rechnung, 'PlzField', 'OrtField', 'StrField');
addSelectBoxEvents(rechnung, 'PlzField', 'OrtField', 'StrField');
setSelectBoxStyles(rechnung, 'PlzField', 'OrtField', 'StrField');
addInputFieldEvents(liefer, 'PlzField', 'OrtField', 'StrField');
addSelectBoxEvents(liefer, 'PlzField', 'OrtField', 'StrField');
setSelectBoxStyles(liefer, 'PlzField', 'OrtField', 'StrField');
```

Bitte vergessen Sie beim Aufbau des HTML-Quellcodes nicht die für den AZ Eingabeassistenten sehr wichtigen HTML-Attribute:

- lkz
- plz

Diese geben die id des jeweiligen Feldes an den AZ Eingabeassistenten weiter, um bei der Anfrage einer Straße an den Server die Straßenergebnisse aus dem richtigen Ort und zur richtigen Postleitzahl zurückliefern.

2.3 Registrierung bei der AZ Direct

Wenn Sie, wie in den vorherigen Seiten beschrieben Ihr Webformular aufgebaut haben, dann müssen Sie sich noch bei der AZ Direct registrieren.

Dabei ist aus technischer Sicht insbesondere die URL wichtig, von der aus sie den AZ Eingabeassistenten verwenden. Da man in der Regel auf mehreren Webseiten den AZ Eingabeassistenten verwenden möchte, können auch sofort mehrere Webseiten registriert werden.

Für Testzwecke kann im oben genannten JavaScript-Teil die Variable `service='AddressMockup'` gesetzt werden. Dadurch liefert der AZ Eingabeassistent-Server unabhängig von der anfragenden Domain Testdaten zurück. **Hierfür ist keine Freischaltung notwendig!**

Daher kann diese Schnittstelle genutzt werden, um während des Webseitenaufbaus das Design an den AZ Eingabeassistenten anzupassen und dabei bereits Testdaten vom Server zu bekommen.

Gibt man also in das Postleitzahl-Feld „123“ ein, sollte der AZ Eingabeassistent automatisch zu „12345“ vervollständigen. Bei Wechsel in das Ort-Feld werden nun Fantasie-Ortsnamen aus dem Roman „Herr der Ringe“ von J. R. R. Tolkien als Orte vorgeschlagen. Als Straßen sind bei jedem dieser Orte wiederum Phantasiestraßen hinterlegt, die zurückgeliefert werden.

Diese Funktionsweise ist lediglich zum Testen während des Einbaus vom AZ Eingabeassistenten vorgesehen, es sollte also beachtet werden, dass vor der produktiven Nutzung unbedingt auf den Wert 'AddressService' umgeschaltet werden muss.

Hier ein Überblick über gültige Test-Eingaben:

PLZ	Ort	Straßen
12345	Buckelstadt	Am Bühl
	Hobbingen	Drei-Viertel-Stein
	Michelbinge	Grenzweg
	Wasserau	Oberbühl
	Wegscheid	Ost-Strasse
		Wasserauer Mühle
		Wasserauer See

2.4 Registrierung HTML Seite

Die Registrierung einer HTML Seite erfolgt üblicherweise mit folgender Anweisung:

```
<script src=https://www.az-direct.com/asp/as.js  
  type="text/javascript"></script>
```

Damit wird folgender Code Schnipsel in die HTML Seite eingefügt:

```
var asid = 'WdKw94aa06RzXiEJjirh03v7oUfd3R6XhbbFFvr2Do4';
```

Hier wird eine Variable angelegt, die den Registrierungsschlüssel als Inhalt bekommt. Folgende Variante kann als Alternative benutzt werden:

```
<script src=https://www.az-direct.com/asp/as.js?jsonp=funktionsname  
  type="text/javascript"></script>
```

Damit wird statt dessen folgender Code Schnipsel in die HTML Seite eingefügt:

```
funktionsname({'asid': ' WdKw94aa06RzXiEJjirh03v7oUfd3R6XhbbFFvr2Do4'});
```

Der `<funktionsname>` kann frei vergeben werden. Hier wird eine Funktion von Ihnen aufgerufen, die den Registrierungsschlüssel als Parameter übergeben bekommt.

In diesem Fall sind Sie allerdings selbst dafür verantwortlich das eine Variable mit der Bezeichnung `asid` angelegt wird und als Inhalt den Registrierungsschlüssel enthält.

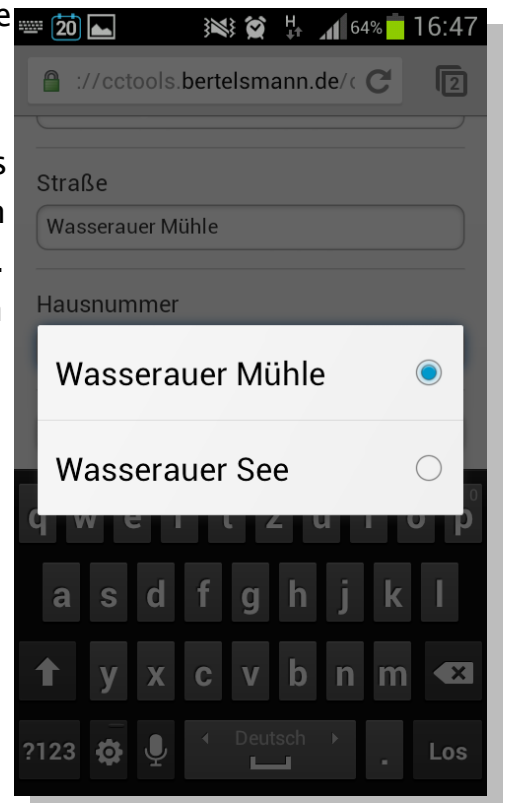
Unabhängig davon welche Variante Sie davon verwenden, Diese Anweisungen sollten nur auf den Seiten vorkommen die den Eingabeassistenten auch wirklich nutzen.

2.5 AZ Eingabeassistent in Verbindung mit mobilen Endgeräten

Der AZ Eingabeassistent sollte auf jedem Browser nutzbar sein, der JavaScript unterstützt. Momentan bieten alle gängigen Browser auf den üblichen Endgeräten diese Funktionalität, sofern der Nutzer dies nicht ausdrücklich unterbindet. Getestet wurde der AZ Eingabeassistenten auf den aktuellen Versionen von Microsoft Internet Explorer, Mozilla Firefox, Opera und Google Chrome auf Desktop-Rechnern.

Mit dem Einzug von JavaScript in den Bereich des mobilen Internets ist der AZ Eingabeassistent auch auf mobilen Geräten wie Tablets und Handys nutzbar. Um ein Formular zu erstellen, das mit mobilen Browsern kompatibel ist, sollte jedoch grundlegendes Vorwissen in diesem Bereich vorhanden sein.

Als Hinweis: Zusätzlich zu der jQuery-Library bietet sich auch die Nutzung von jQuery Mobile an. Hierbei sollte allerdings auf die passenden Versionen der beiden Frameworks geachtet werden. Hilfreich ist außerdem die Zuweisung von `data-role="none"` an die `<select>`-Felder, da jQuery-Mobile sonst unerwünschte Zusatzfelder erzeugt.



Unter Android und iOS bietet es sich weiterhin an, statt `<select>` und `<option>`-Tags mit `<div>` und ``-Containern zu arbeiten, da ansonsten eventuell betriebssystemeigene Varianten eingeblendet werden, die unerwünschte Ergebnisse liefern könnten. (siehe Screenshot)

Eine Beispielseite, die für die mobile Nutzung optimiert wurde finden Sie unter <https://oak.az-direct.com/oak/autosuggest/MobileDemo.jsp>

3 Konfigurationsdetails

3.1 JavaScript

speed

Legt die Darstellungsgeschwindigkeit fest, mit der das OptionMenü dargestellt werden soll. Mögliche Werte:

‘fast‘ (mit 200ms Verzögerung)

‘slow‘ (mit 600ms Verzögerung)

```
var speed = 'fast';
```

service

legt den benutzten Service fest, wahlweise Produktiv- oder Testservice

```
var service = 'AddressService'; // Produktivservice  
oder  
var service = 'AddressMockup'; // Testservice
```

requestMapping

Unterscheidung zwischen Produktions- oder Integrationssystem

```
var requestMapping = 'www.az-direct.com/asp/as'; // Produktion  
oder  
var requestMapping = 'integ.az-direct.com/asp/as'; // Integration
```

maxSelectSize

legt die Anzahl der Einträge für alle OptionMenüs fest, die sichtbar dargestellt werden sollen

```
var maxSelectSize = 5;
```

minLengthPlz

legt fest, ab der wievielten PLZ-Ziffer eine Vorschlagsliste ermittelt wird

```
var minLengthPlz = 3;
```

minLengthOrt

legt fest, ab dem wievielten Zeichen eine Ortsvorschlagsliste geholt wird

```
var minLengthOrt = 0;
```

minLengthStr

legt fest, ab dem wievielten Zeichen eine Straßenvorschlagsliste geholt wird

```
var minLengthStr = 1;
```

requestStyle

Mit diesem optionalen Attribut, können sie die Darstellung der OptionMenüs beeinflussen, d.h. die Auswahllisten, die erscheinen, wenn man z.B. zwei Ziffern der PLZ eingetragen hat und dann alle möglichen PLZs in einem „OptionMenü“ angezeigt werden.

Ist der requestStyle nicht gesetzt, werden Standard HTML <select>-Elemente verwendet. Wird die Variable requestStyle hingegen auf den Wert 'div' gesetzt, dann wird das OptionMenü aus anderen HTML-Elementen erstellt.

Der Vorteil: auf mobilen Endgeräten ist die Darstellung besser. Der Nachteil: die Verwendung von Cursor-Tasten ist nicht mehr gegeben (was auf mobilen Endgeräten auch nicht erforderlich ist.)

```
var requestStyle = 'div';
```

4 Anhang Quellcode

```

<html>
<head>
  <title>AZ Eingabeassistent</title>
  <script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"
    type="text/javascript"></script>
  <script src="http://oak.az-direct.com/oak/js/addressServices.js"
    type="text/javascript"></script>
  <script src="https://integ.az-direct.com/asp/as.js" type="text/javascript"></script>
  <script type="text/javascript">
    var speed = 'fast';
    var service = 'AddressService';
    var requestMapping = 'integ.az-direct.com/asp/as';
    var maxSelectSize = 5;
    var minLengthPlz = 3;
    var minLengthOrt = 0;
    var minLengthStr = 1;

    // startup initialization
    $(document).ready(function() {
      var group = 1;

      // adding various events to the input fields
      addInputFieldEvents(group, 'PlzField', 'OrtField', 'StrField');

      // adding various events to the select boxes
      addSelectBoxEvents(group, 'PlzField', 'OrtField', 'StrField');

      // initialize field styling selection boxes
      setSelectBoxStyles(group, 'PlzField', 'OrtField', 'StrField');
    });
  </script>
  <script type="text/javascript">
    <!-- JavaScript Code fuer Browser, die kein Scripting unterstuetzen ausblenden
    //autocomplete für input deaktivieren
    $("input").attr({ autocomplete:"off" });
    // Ende der JavaScript Sektion -->
  </script>
</head>
<body>
<form>
  <input type="hidden" id="LkzField1" value="de"/>
  <table>
  <tr><td>PLZ/Stadt</td>
  <td>
    <select id="SelectPlzField1" tabindex="105"
      style="position:absolute;display:none;z-index:1000;height:auto;padding-left:0;">
    </select>
    <input id="PlzField1" tabindex="104" style="width: 65px;"
      lkz="LkzField1" type="text" value="" maxlength="5" autocomplete="off"/>
    <select id="SelectOrtField1" tabindex="107"
      style="position:absolute;display:none;z-index:1000;height:auto;padding-left:0;">
    </select>
    <input id="OrtField1" tabindex="106" style="width: 175px; margin-left: 3px;"
      lkz="LkzField1" plz="PlzField1" type="text" value=""
      maxlength="60" autocomplete="off"/>
  </td>
</tr>
  <tr>
  <td>Strasse/Nr.</td>
  <td>
    <select id="SelectStrField1" tabindex="108"
      style="position:absolute;display:none;z-index:1000;height:auto;padding-left:0;">
    </select>
    <input id="StrField1" tabindex="107" style="width: 200px;" lkz="LkzField1"
      plz="PlzField1" type="text" value="" maxlength="60" autocomplete="off"/>
    <input id="HnrField" tabindex="109" style="width: 40px; margin-left: 3px;"
      type="text" value="" maxlength="5" autocomplete="off"/>
  </td>
</tr>
</table>
</form>
</body>
</html>

```

5 Anhang Glossar

Request

Der AZ Eingabeassistent unterscheidet in der Regel zwischen einem Request und einer Transaction. Jedes Mal, wenn die JavaScript Library `as.js` eingebunden wird, wird ein sogenannter „Request“ gezählt. Ein Request ist quasi das Recht, den AZ Eingabeassistent nutzen zu können. Es wird deshalb empfohlen, wirklich nur auf den Webseiten, auf denen der AZ Eingabeassistent zum Einsatz kommen soll, `as.js` einzubinden.

Transaction

Im Gegensatz zum Request wird im AZ Eingabeassistent Umfeld von einer Transaction gesprochen, wenn die eigentliche Expansions-Funktionalität erstmalig genutzt wird. Voraussetzung ist hier, dass ein Request vorliegt und z.B. die ersten 3 Ziffern der Postleitzahl eingetippt wurden. In diesem Fall wird erstmalig die Vorschlagsliste für alle Postleitzahlen angefordert und der Transaction-Zähler wird hochgezählt. Sollten in derselben Web-Session dann mehrere Adressen über den AZ Eingabeassistenten expandiert werden, wird der Transaction-Zähler nicht erneut hochgezählt.

Länderkennzeichen

Ein Länderkennzeichen ist für den AZ Eingabeassistent unbedingt notwendig. Bitte beachten Sie, dass Sie hier die zweibuchstabigen Codes aus dem ISO-3166 (ALPHA2) Standard verwendet werden müssen. Groß/Kleinschreibung ist dabei irrelevant. Aktuell werden folgende Länder unterstützt: de-Deutschland, at-Österreich, ch-Schweiz, nl-Niederlande, be-Belgien, lu-Luxemburg