

Zuwendungsempfängerin	Stadt Bielefeld
Vorhabensbezeichnung	Einführung des Serviceportals der Fa. regio iT bei der Stadt Bielefeld
Förderkennzeichen/Aktenzeichen	34.03.09-002/2019-023
Arbeitspaket	8 - Erweiterung des Bürgerserviceportals mit den Funktionalitäten des Geoportals der Stadt Bielefeld
ggf. Maßnahme	Dateibasierter grafischer Editor mit Raumbezug zur Unterstützung von eGovernment-Prozessen (Geo-Editor) – Schnittstelle zu Form-Solutions
Koordinator/in	Dr. Elmar Schalück

Einleitung
<p>Im Rahmen des Modellprojektes OWL ist ein dateibasierter grafischer Editor mit Raumbezug (geoEDITOR) entwickelt und als Open Source bereitgestellt worden.</p> <p>Kernaufgabe des geoEDITORS ist es, eine einfach zu bedienende Komponente auf Basis der existierenden Geodateninfrastruktur der Stadt Bielefeld zu schaffen. Mit dieser Komponente sollen Punkte, Linienzüge und Flächen auf dem Stadtplan eingezeichnet und mit Fachattributen versehen werden können. Die Resultate der beschriebenen Bearbeitungsfunktionen sollen anschließend in dem Format GeoJSON ausgegeben werden und automatisiert weiterverarbeitet werden können. Der geoEDITOR basiert auf dem bielefeldGEOCLIENT, dem Bielefelder WEB-GIS. Apps können frei definiert werden, um für die jeweiligen Anwendungszwecke optimierte Anwendungen bereitstellen zu können.</p> <p>Um den entwickelten geoEDITOR innerhalb von Online-Diensten einsetzen zu können, ist eine Integration in die eingesetzte Formularserverlösung der Firma Form-Solutions vorgenommen worden. Dazu wurden geeignete Mittel auf Seiten des Formularservers entwickelt.</p>
Architektonische Überlegungen
<ul style="list-style-type: none"> - Die geplante Infrastruktur sieht den Formularserver als Schnittstelle zur Bürgerin bzw. zum Bürger vor. Der ausgelieferte Inhalt vom Formularserver beinhaltet ein IFrame. Über dieses IFrame werden (für die Bürgerin bzw. den Bürger transparent) Inhalte vom geoEDITOR-Server nachgeladen. Das Zusammenspiel dieser zwei verschiedenen, voneinander unabhängigen Serverlandschaften in den Webbrowser der Bürgerin bzw. des Bürgers ist aus mehreren Gründen nicht trivial: Die Sicherheit zwischen beiden Servern ist die Hauptschwierigkeit, beide Systeme müssen in der Lage sein, in einem höchstmöglich abgeschotteten Bereich zu arbeiten. Keines der beiden Systeme darf ungeprüft Daten des anderen Systems übernehmen. Cross-site Scripting-Angriffe sind eine riesige Sicherheitsbedrohung und müssen unbedingt beachtet werden. - Ein System (hier der Formularserver) ist der Rahmen, über den das andere System (hier der geoEDITOR) aufgerufen und integriert wird. Basis ist für den Formularserver der Assistent, der als Webanwendung im Browser angezeigt wird. Auch der geoEDITOR ist im Browser dargestellt. - Für die sichere Kommunikation zwischen beiden Komponenten wurde entschieden, auf die Einbettung des geoEDITORS als IFrame in den Formularserver zu setzen und die Kommunikation auf die Technik postMessage (Window.postMessage() - Web APIs MDN (mozilla.org)) einzustellen. Diese Methode wird als sicher angesehen, wenn beide Komponenten genügende Sicherheitsmechanismen wie Validierung der Herkunft der Nachrichten nur vom erwarteten Kommunikationspartner beachten. <p>Sowohl auf Seite des Formularservers als auch im geoEDITOR sind Möglichkeiten vorhanden, JavaScript-Komponenten für die Implementierung von postMessage-Kommunikation zu hinterlegen.</p>

Definition der verwendbaren PostMessage-Methoden

Folgende wiederkehrende (d.h. für verschiedene Antragsprozesse nutzbare) Methoden sind im geoEDITOR implementiert worden:

- **munimapReady** : Diese Methode wird vom geoEDITOR an seinen Parent (also den Formularserver) gesendet, wenn der geoEDITOR komplett geladen und initialisiert wurde.
Kommunikationsrichtung: geoEDITOR an Parent
- **finishGeoEditing**: Diese Methode erlaubt es, eine Anfrage an den geoEDITOR zu senden, ob alle Pflichtgeometrien und Pflichtattribute vom Anwender erstellt worden sind und im Positivfall wird die erstellte Struktur als GeoJSON-Objekt zurückgeliefert. Es müssen keine Parameter übergeben werden.
Kommunikationsrichtung: Parent an geoEDITOR
- **finishGeoEditing_response**: Diese Methode liefert den Rückgabeparameter success als true oder false (letzteres, wenn nicht alle Pflichtgeometrien und Pflichtattribute komplett sind). Im Erfolgsfall wird das GeoJSON als JavaScript-Objekt ausgeliefert.
Kommunikationsrichtung: geoEDITOR an Parent
- **printGeoEditing**: Diese Methode sendet dem geoEDITOR den Auftrag, die aktuell erstellte/n Geometrie/n als Datei (PNG oder PDF) zu generieren. Das Layout (z.B. „a4-portrait“), der Rand in Pixeln, die minimale Skalierung (Falls nötig wird der Maßstab automatisch vom geoEditor angepasst, z.B. wenn der ausgewählte Ausschnitt, in Relation zum gewählten Layout, für eine vollständige Darstellung sonst nicht ausreichen würde) und das Ausgabeformat („png“ oder „pdf“). Die Methode erstellt den Auftrag auf dem Server und sollte nur bei einer vollständigen und nicht gegenüber dem letzten Aufruf geänderten Geometrie aufgerufen werden, da sonst mehrfach Last erzeugt würde.
Kommunikationsrichtung: Parent an geoEditor
- **printGeoEditing_response**: Diese Methode liefert den Rückgabeparameter success als true oder false und die URL der erzeugten Datei als Download-URL. Achtung: Die Datei muss noch nicht auf dem Server fertig gerendert sein, dies kann einige Sekunden in Anspruch nehmen.
Kommunikationsrichtung: geoEditor an Parent

Ab hier kommen die Methoden, die eine App des geoEDITORS selbst beisteuern kann. Für das erste Beispiel sind dies:

- **centerOnCoord**: Diese Methode wird vom Parent auf dem geoEDITOR aufgerufen, um eine Positionierung auf eine spezifizierte Geokoordinate (Latitude, Longitude, Zoomstufe und Koordinatensystem) durchzuführen.
Kommunikationsrichtung: Parent an geoEDITOR
- **centerOnCoord_response**: Jede Methode, die vom Parent aufgerufen wird, liefert auch einen korrespondierenden Response zurück. Es werden hier die Werte success als true oder false und eine message mit einer Fehlerbeschreibung geliefert.
Kommunikationsrichtung: geoEDITOR an Parent
- **zoomToBBBox**: Diese Methode platziert den geoEDITOR auf ein angegebenes Rechteck (MinX, MinY, MaxX, MaxY und Koordinatensystem) und zoomt automatisch auf diesen Rahmen.
Kommunikationsrichtung: Parent an geoEDITOR
- **zoomToBBBox_response**: success als true oder false und eine message mit einer Fehlerbeschreibung.
Kommunikationsrichtung: geoEDITOR an Parent
- **zoomToGeom**: Diese Methode platziert den geoEDITOR auf eine angegebene GeoJSON Geometrie.
Kommunikationsrichtung: Parent an geoEDITOR
- **zoomToGeom_response**: liefert den Rückgabeparameter success als true oder false und eine message mit einer Fehlerbeschreibung.
Kommunikationsrichtung: geoEDITOR an Parent
- **save_geoedits_to_storage**: Diese Methode speichert die aktuell erzeugte Geometrie als Session-Cookie für diese geoEDITOR-App auf dem Clientrechner.
Kommunikationsrichtung: Parent an geoEDITOR
- **geoedits_in_storage_available**: Diese Methode prüft, ob ein Session-Cookie für diese geoEDITOR-App vorhanden ist.
Kommunikationsrichtung: Parent an geoEDITOR
- **geoedits_in_storage_available_response**: liefert den Rückgabeparameter success als true oder false und die Verfügbarkeit des Cookies mittels des Parameter available als true oder false.
Kommunikationsrichtung: geoEDITOR an Parent
- **remove_geoedits_from_storage**: Diese Methode löscht den Session-Cookie auf dem Clientrechner.
Kommunikationsrichtung: Parent an geoEDITOR.

- `restore_geoedits_from_storage`: Diese Methode lädt die im Session-Cookie gespeicherte Geometrie zurück in den geoEDITOR. Dies ist dann wichtig, wenn der Parent zwischenzeitlich eine andere Seite ohne den geoEDITOR aufgerufen hat, z.B. bei der Navigation zwischen mehreren Formular-Seiten.
Kommunikationsrichtung: Parent an geoEDITOR

Neue Möglichkeiten in Form-Solutions

Für das Projekt wurden von Form Solutions drei Dinge erstellt bzw. verbessert. Diese Änderungen sind seit der Version 4.69 für alle Nutzer verfügbar:

- Editor für JavaScript: Der Editor ist nun an drei Stellen verwendbar:
 - o Administration (Symbol Schraubenschlüssel) gilt für alle Assistenten
 - o Assistenten (Symbol JS in der zentralen Iconleiste)
 - o Panel (Symbol JS in der Panel-Iconleiste).
- Generierung eines festen Attributs im erzeugten HTML-Code für eine Komponente.
Über das Attribut `data-component-identifizier` kann im HTML z.B. über die Methode `document.querySelector("[data-component-identifizier='Eingabefeld'] input")` das Input-Element zum technischen Identifier Eingabefeld ermittelt werden. Dies gilt für alle Komponenten auf einem Panel, bei Objektgruppen ist das Attribut mehrfach aufzuführen, d.h. es sollte die Methode `document.querySelector("[data-component-identifizier='Gruppenbezeichnung'] [data-component-identifizier='Eingabefeld'] input")` genutzt werden.
- Erstellung einer neuen Komponente „Externe Ressource“: Diese Ressource bindet einen IFrame auf eine fremde URL ein. Breite und Höhe können über die speziellen Einstellungen definiert werden. Per CSS kann aber auch die Breite von 100% dafür festgelegt werden.

Umsetzung der Integration am Beispiel „Antrag auf Sondernutzung: Aufstellung von Schuttmulden“

Der erste produktiv gesetzte Anwendungsfall für die Integration des geoEDITORS wurde für die Aufstellung von Schuttmulden durchgeführt.

Thematische Beschreibung

Die Beantragung für die Aufstellung von einer oder mehreren Schuttmulden im öffentlichen Verkehrsraum (z.B. auf Bürgersteigen) erfordert eine Sondernutzungsgenehmigung des bei uns zuständigen Verkehrsamtes. Mit dem Verkehrsamt wurde vereinbart, dass eine Bürgerin, ein Bürger oder das beauftragte Entsorgungsunternehmen 1 bis maximal 5 Schuttmulden in einem Antrag mit dem geoEDITOR eingeben sollen. Für jede einzelne Mulde ist dabei die entsprechende Abmessung anzugeben. Der komplette Antrag mit der exakten Position der Mulde ist dann an das Verkehrsamt zu leiten, wo die Bescheidung des Antrags vorgenommen wird.

Technische Definition der geoEDITOR-App

In der App wurde festgelegt, dass von den verschiedenen Geometrietypen der Typ Point exakt einmal verwendet werden soll. Mit diesem Typ Point soll der Ort der Schuttmulde über das Symbol markiert werden. Mehrere Schuttmulden werden grafisch nicht dargestellt. Die Bezeichnung für den Typ wurde im geoEDITOR auf „Aufstellort“ gesetzt. Bei den Sachdaten zu diesem Typ wurde ein numerisches Pflichtfeld „Anzahl der Mulden“ abgefragt. Linienzüge und Flächen sollen nicht Verwendung finden.

Technische Umsetzung im Formularserver

Im Formularserver war ein neuer Assistent „KFAS_BI_Schuttmulden“ erstellt worden. Neben den allgemeinen Standards bei der Stadt Bielefeld (Datenschutz, Allgemeine Hinweise, Daten zum Antragsteller) werden weitere Daten über jeweils einzelne Panels für das Fachverfahren abgefragt:

- Angaben zum Antragsteller (Natürliche / juristische Person) und die Adresse, an der die Schuttmulde aufgestellt werden soll.

Mit der Funktion Autovervollständigung des Formularservers wird über drei Pflichtfelder die Adresse des Aufstellortes ermittelt. Dazu werden drei ComboBoxen (Postleitzahl, Straße und Hausnummer) verwendet, die aus einer Liste (BI_Strassen) im Formularserver gespeist werden. Die ComboBoxen in der Autovervollständigung haben dabei die Prioritätsstufen 1 bis 3.

Die hinterlegte Liste (BI_Strassen) kennt zu jeder Kombination aus Postleitzahl, Straße, Hausnummer zusätzlich die jeweiligen Geoinformationen (Latitude und Longitude). Über die Autovervollständigung des Formularservers wird anhand der Pflichtfelder die Latitude und Longitude automatisch ermittelt und jeweils in ein verstecktes und schreibgeschütztes Textfeld geschrieben. Die versteckten Felder haben beide die Prioritätsstufe 4 und werden im Formular und im PDF ausgeblendet.

Komponente "Auto-Vervollständigung" konfigurieren

Allgemeine Einstellungen
 FIM/OZG
 Spezielle Einstellungen

Art des Services: Vorkonfigurierte Datenliste

Liste auswählen: BI_Strassen

Art der Validierung: Auto-Vervollständigung und Validierung

Bezeichnung technisch	Listenfeld	Nur ab Feldanfang suchen?	Groß-Klein-Schreibung ignorieren?	Prioritätsstufe
PLZ	postalCode	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1
Strasse	street	<input type="checkbox"/>	<input checked="" type="checkbox"/>	2
Hausnummer	houseNumber	<input type="checkbox"/>	<input type="checkbox"/>	3
Lat	latitude	<input type="checkbox"/>	<input type="checkbox"/>	4
Lon	longitude	<input type="checkbox"/>	<input type="checkbox"/>	4

Weitere Angaben zur Muldenbestellung sind ebenfalls auf dem Panel, haben aber keinen Zusammenhang mit dem geoEDITOR.

Es gibt ein weiteres Panel, auf dem der geoEDITOR selbst aufgerufen wird. Darauf sind aus technischen Gründen eine Reihe von Komponenten eingetragen, in denen Hilfsinformationen für den geoEDITOR bereitgestellt werden. Hier ein Beispiel der Darstellung für die Bürgerin, den Bürger oder das beauftragte Entsorgungsunternehmen:



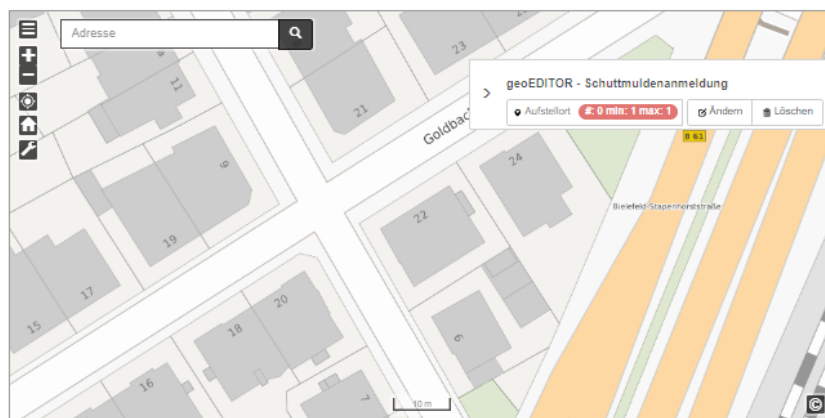
Antrag zum Abstellen von Schuttmulden

Assistentenvorschau - Nicht zur Veröffentlichung

Angaben zum Standort

Benötigen Sie Hilfe zur Bedienung des geoEDITORS?

☐ ja ☒ nein



[→ Weiter](#)

[|< Zum Anfang](#)

[← Zurück](#)

[X Abbruch](#)



Damit die Werte auch in JavaScript für die Kommunikation zum geoEDITOR genutzt werden können, ist es entscheidend, dass diese Komponenten sowohl im Assistenten angezeigt werden (d.h. von Form Solution als HTML mit gerendert werden) als auch beschreibbar (kein Schreibschutz, kein Pflichtfeld) sind. Das „Nicht-Anzeigen“ im Browser muss dann aber per JavaScript implementiert werden. Dies Vorgehen wird weiter unten beschrieben.

3_Aufstellung (Angaben zum Standort)

Latitude				Breite: 100%
Longitude				Breite: 100%
geoEDITOR				
Hilfe zu geoEDITOR				Breite: 100%
Hilfeerklärung				
extGeoEDITOR				
ImageURL				Breite: 100%
geoJSON				Breite: 100%
geoEditorInitialisiert				Breite: 100%
geoEditorKomplett				Breite: 100%
geoPositionierung				Breite: 100%
AnzahlGeometrien				Breite: 100%
Debug				Breite: 100%
AnzahlMulden				Breite: 100%
KoordinateLat				Breite: 100%
KoordinateLon				Breite: 100%
Webservice-Komponente				
Fehlermeldung				Breite: 100%
KoordinateBielefeld				Breite: 100%

Vom vorigen Panel werden per Kopierregel (Expertenregel Initialisierung des Panels) die Werte für Latitude und Longitude aus den versteckten Feldern übertragen und für die Zentrierung des Kartenausschnitts genutzt.

Die Objektgruppe „geoEDITOR“ fasst alle Komponenten zusammen, die direkt zum geoEDITOR-Assistenten gehören und auch für andere Assistenten so wiederverwendet werden können.

Weil der geoEDITOR nicht in allen Punkten selbsterklärend ist, wird eine per Aktivierungsregel anzeigbare Hilfeerklärung angeboten. Diese erläutert die Nutzung und kann je Assistent angepasst werden.

Benötigen Sie Hilfe zur Bedienung des geoEDITORS?

☒ ja ☐ nein

Durch einen Klick auf **Aufstellort** kann der gewünschte Aufstellort in die Karte eingezeichnet werden. Dieses erkennen Sie durch den veränderten Cursor innerhalb der Karte .

Ein Klick in die Karte markiert den gewünschten Standort. Es erscheint ein Fenster, in welchem Sie noch die gewünschte Anzahl der

Mulden eintragen müssen. Anschließend können Sie den Antragsprozess über fortsetzen.

Änderungen

Möchten Sie den gewählten Standort ändern, so klicken Sie zunächst auf und anschließend auf die entsprechende

Markierung . Dadurch ändert sich die Darstellung des Standortes und der kann bei gedrückter Maustaste frei auf der Karte bewegt werden.

Möchten Sie hingegen die gewünschte Anzahl der Mulden für den gewählten Standort ändern, können Sie das entsprechende Fenster

durch Klick auf die entsprechende Markierung erneut aufrufen.

Möchten Sie den gewählten Standort löschen, so klicken Sie zunächst auf und anschließend auf die entsprechende

Markierung .

Danach ist die externe Ressource definiert. Im Feld URL ist die Adresse des eigentlichen geoEDITORS notiert. ImageURL enthält die URL, die von der Methode printGeoEditing als Abruf-URL zurückgeliefert wird. GeoJSON beinhaltet den kompletten Datensatz der erstellten Geodaten im gleichnamigen GeoJSON Format (GeoJSON – Wikipedia). Das GeoJSON enthält ein Array von „Features“. In diesem Array befinden sich die verschiedenen Geometrien (Typen):

- Punkt (Point) als Array mit zwei Punkten als Koordinate,
- Linienzug (LineString). Eine Linie besteht aus mehreren (≥ 2) Punkten. Der Linienzug ist ein Array mit mehreren Arrays, bestehend jeweils aus zwei Punkten als Koordinate.
- Fläche (Polygon). Eine Fläche besteht aus mehreren (≥ 3) Punkten, die anders als der Linienzug eine geschlossene Fläche bilden können. Das Polygon ist ein Array mit mehreren Arrays, bestehend jeweils aus zwei Punkten als Koordinate. Das 1. und letzte Element des Arrays sind identisch und schließen damit die Fläche.

Neben den Punkt-, Linien- und Flächeninformationen enthält das „Feature“ Array auch sogenannte Properties für die oben beschriebenen Typen, hier: style mit Informationen zur Formatierung und formValues mit einer Name-Value-Liste der projektspezifischen Attribute. Dieses geoJSON wird als Komponente im Assistenten gespeichert, sobald alle Pflichteinträge gemacht worden sind.

geoEditorInitialisiert speichert temporär, ob der geoEDITOR schon komplett initialisiert ist. Dies passiert, nachdem die PostMessage „munimapReady“ vom geoEDITOR gesandt wurde.

geoEDITORKomplett speichert, ob alle Pflichtfelder und geometrischen Elemente im geoEDITOR gesetzt sind. Dieser Wert wird von einer Validierungsregel verwendet, die abschließend prüft, ob alles komplett ist.

geoPositionierung soll genutzt werden, um eine Positionierung auf die Geokoordinaten der vorher abgefragten Adresse im geoEDITOR nur dann durchzuführen, wenn dies nicht vorher geschehen ist.

AnzahlGeometrien wird aus dem geoJSON herausgeholt, um die Anzahl aller eingegebenen Geometrien zu ermitteln. Diese Information wird im hier beschriebenen Assistenten nicht benötigt.

Außerhalb der Objektgruppe „geoEDITOR“ sind die Werte aus der Fachlogik aufgeführt:

AnzahlMulden extrahiert aus dem geoJSON die vom Anwender angegebene Muldenzahl aus der ersten (einzigen) Geometrie und dem Property „formValues.count“. Die Anzahl muss in der Validierung zwischen 1 und 5 liegen.

KoordinateLat und KoordinateLon extrahiert die exakte Position der eingezeichneten Schuttmulde.

Auf einem weiteren Panel wurde die ImageURL genutzt, um die erzeugten Geometrien als Bild anzuzeigen (Expertenregel für die Initialisierung) und so auch in die generierte PDF-Datei einzubinden. Dies ist nicht zu 100% stabil, es kann sein, dass das Bild nicht geladen wird. In dem Fall muss im Browser die Seite aktualisiert werden.

Nutzung JavaScript im Formularserver

Im Formularserver ist ein komplexes JavaScript am Panel konfiguriert, mit dem die Kommunikation zwischen beiden Systemen (Formularserver und geoEDITOR) mit Hilfe der PostMessage-Methoden realisiert ist.

Ein Beispielskript ist im Paket enthalten.

Wichtige Methoden sind dort ausführlich dokumentiert (Ein- bzw. Ausblenden von Feldern oder der ganzen Gruppe; Asynchroner Aufruf der PostMessage-Methoden des geoEDITORS und Warten auf die Response; Extrahieren von Werten aus dem GeoJSON; Schreiben von Werten in die Formularserver-Komponenten; Umgang mit den Session-Cookies)

Zeitplanung

JAHR	2020												2021												2022												'23				
PLAN																																									
IST																																									

Erläuterung bei zeitlicher Abweichung

Bielefeld, 22.06.2023

gez. Schalück