



# Datenmodelle in GBD WebSuite 8

Georg Barikin  
Geoinformatikbüro Dassau GmbH

[barikin@gbd-consult.de](mailto:barikin@gbd-consult.de)

- Komplexe Datenstrukturen in Formularen abbilden
  - Beziehungen zwischen Strukturen abbilden
    - Eingaben flexibel validieren
      - Dateiverwaltung

# Beispieldatenstruktur

- Aufgabengebiet “Historische Häuser in Düsseldorf”
- eine Strasse hat mehrere Häuser (1:M Beziehung)
- ein Stadtteil hat mehrere Strassen, eine Strasse gehört zu mehreren Stadtteilen (M:N Beziehung über eine “link” Tabelle)
- an Häuser können Dokumente angehängt werden (M:1)
- Referenz-Spalten (M:1)

dokument			
id	haus_id	datei	titel
10	1	IMG002.jpeg	Ansicht vorne
20	2	Antrag.docx	Antrag
30	3	B123123.pdf	Beschreibung

haus				
id	strasse_id	nummer	baujahr	nutz_art
1	1	2a	1960	100
2	2	13	1970	100
3	20	5	1980	200
4	30	5b	1990	100
5	12	33	2010	300

strasse	
id	name
1	Kaiserstraße
2	Inselstraße
3	Sternstraße
4	Königsallee

nutzung	
id	nutzungsart
100	Wohnhaus
200	Industrie
300	Gewerbe
400	Kultur

strasse_stadtteil	
strasse_id	stadtteil_id
1	11
2	11
3	44

stadtteil	
id	name
11	Altstadt
22	Pempelfort
33	Derendorf

# Einfaches Formular: aus r7 bekannt

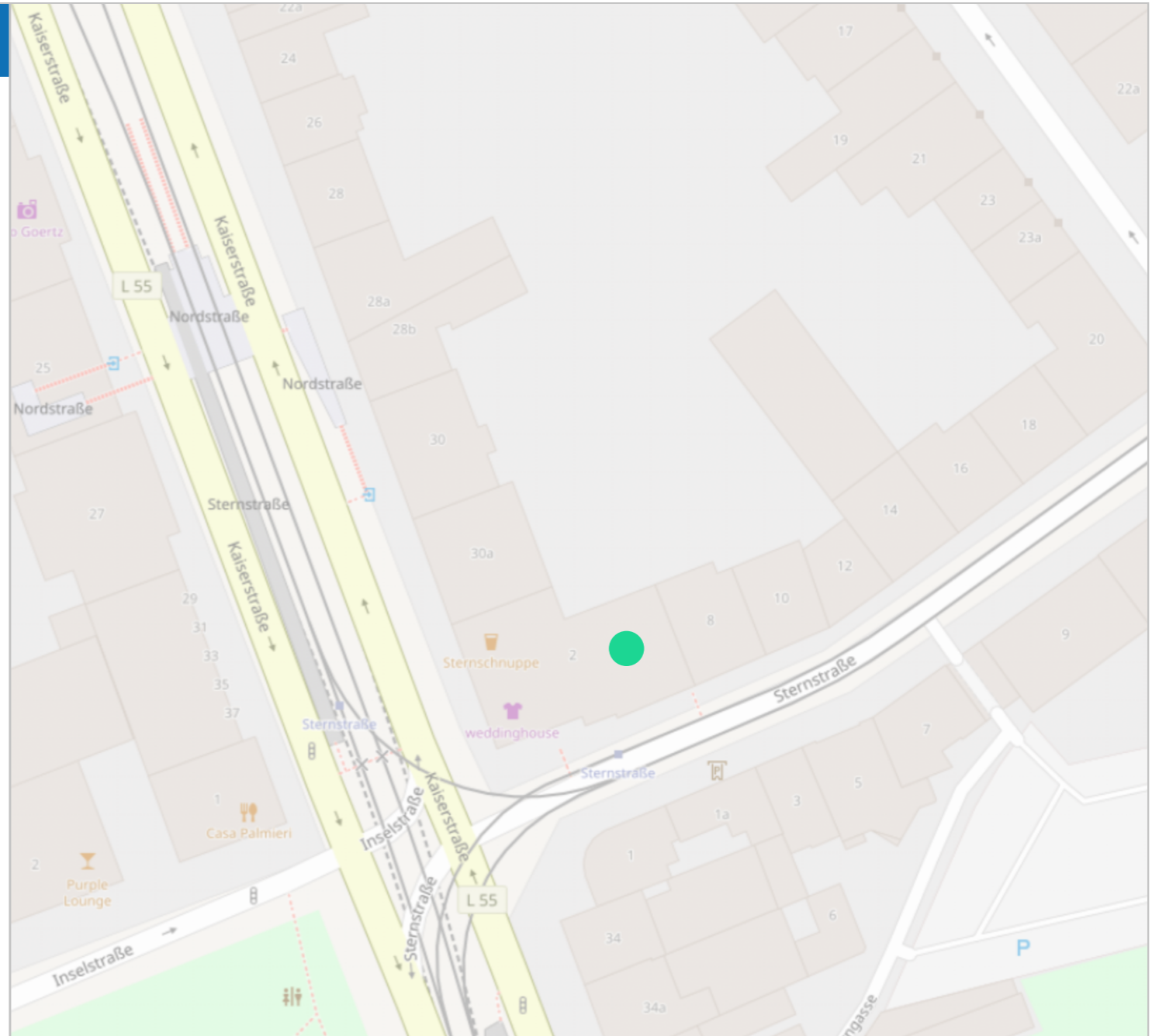
Haus #1

Hausnummer

12a

Baujahr

1960



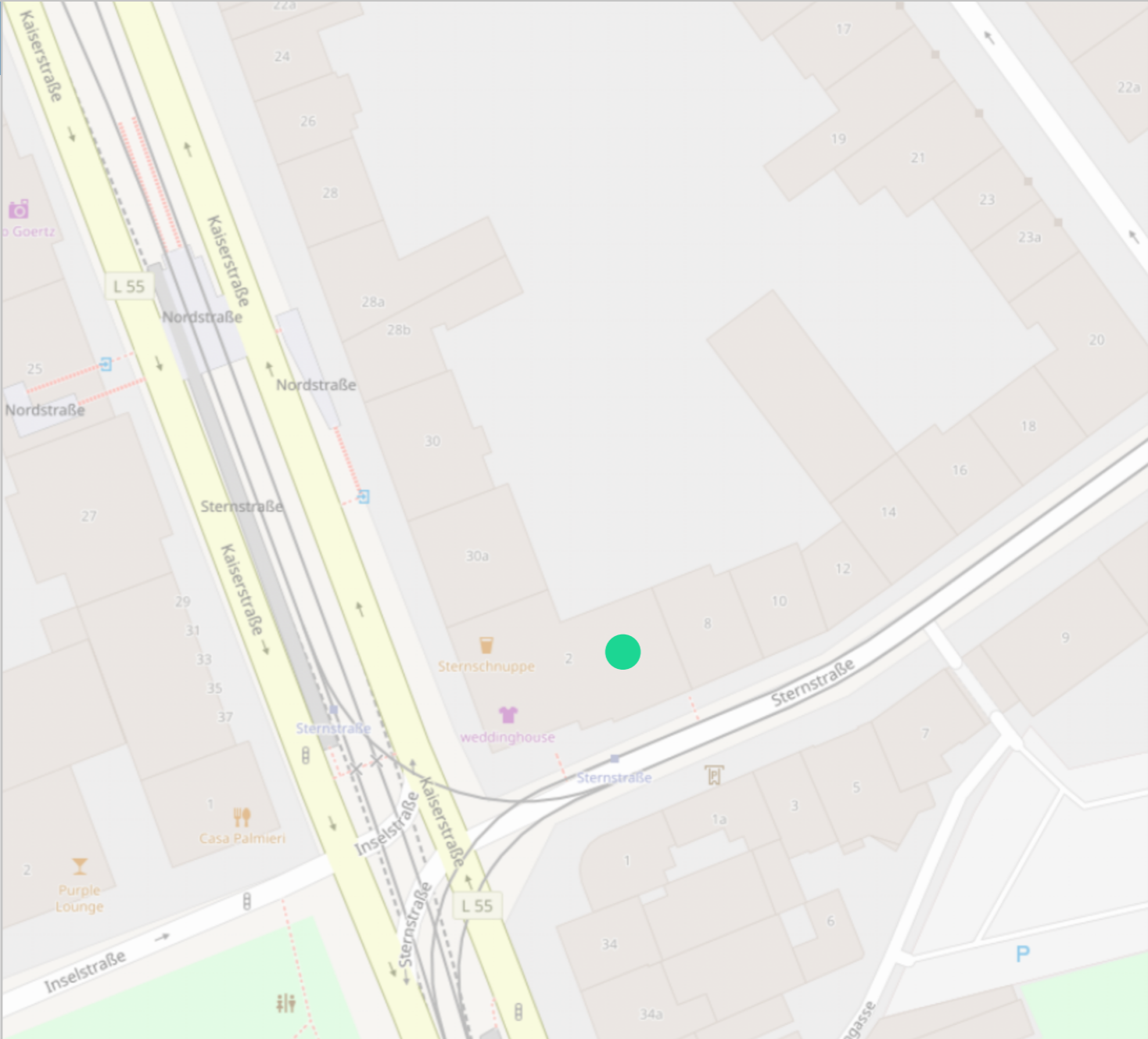


- jedes Model muss ein UID haben
- große Auswahl von Feld Typen und “widgets”

```
model {
  uid "hausModel"
  table "haus"
  fields+ {
    name "nummer"
    title "Hausnummer"
    type "string"
    widget.type "input"
  }
  fields+ {
    name "baujahr"
    title "Baujahr"
    type "integer"
    widget.type "integerInput"
  }
}
```

# neu in r8: Referenztabelle

<b>Haus #1</b>	
Hausnummer	<input type="text" value="12a"/>
Baujahr	<input type="text" value="1960"/>
Nutzung	<div><p><b>Gewerbe</b> ▾</p><p>Wohnhaus</p><p>Industrie</p><p><b>Gewerbe</b></p><p>Kultur</p></div>



- Feld Typ: **relatedFeature**
- Verknüpfung der Modelle über UIDs
- Widget "featureSelect"
- keine speziellen Anforderungen an Tabellen- und Spaltennamen

```
model {
  uid "hausModel"
  table "haus"
  fields+ {
    name "nutzung"
    title "Nutzung"
    type "relatedFeature"
    relation.modelUid "nutzungModel"
    foreignKey.name "nutz_art"
    widget.type "featureSelect"
  }
}

model {
  uid "nutzungModel"
  table "nutzung"
}
```

# M:1 Verknüpfung: "Haus" gehört zu "Strasse"

## Haus #1

Hausnummer

12a

Baujahr

1960

Strasse

Ca I

Cannstatter Str.

Capitostr.

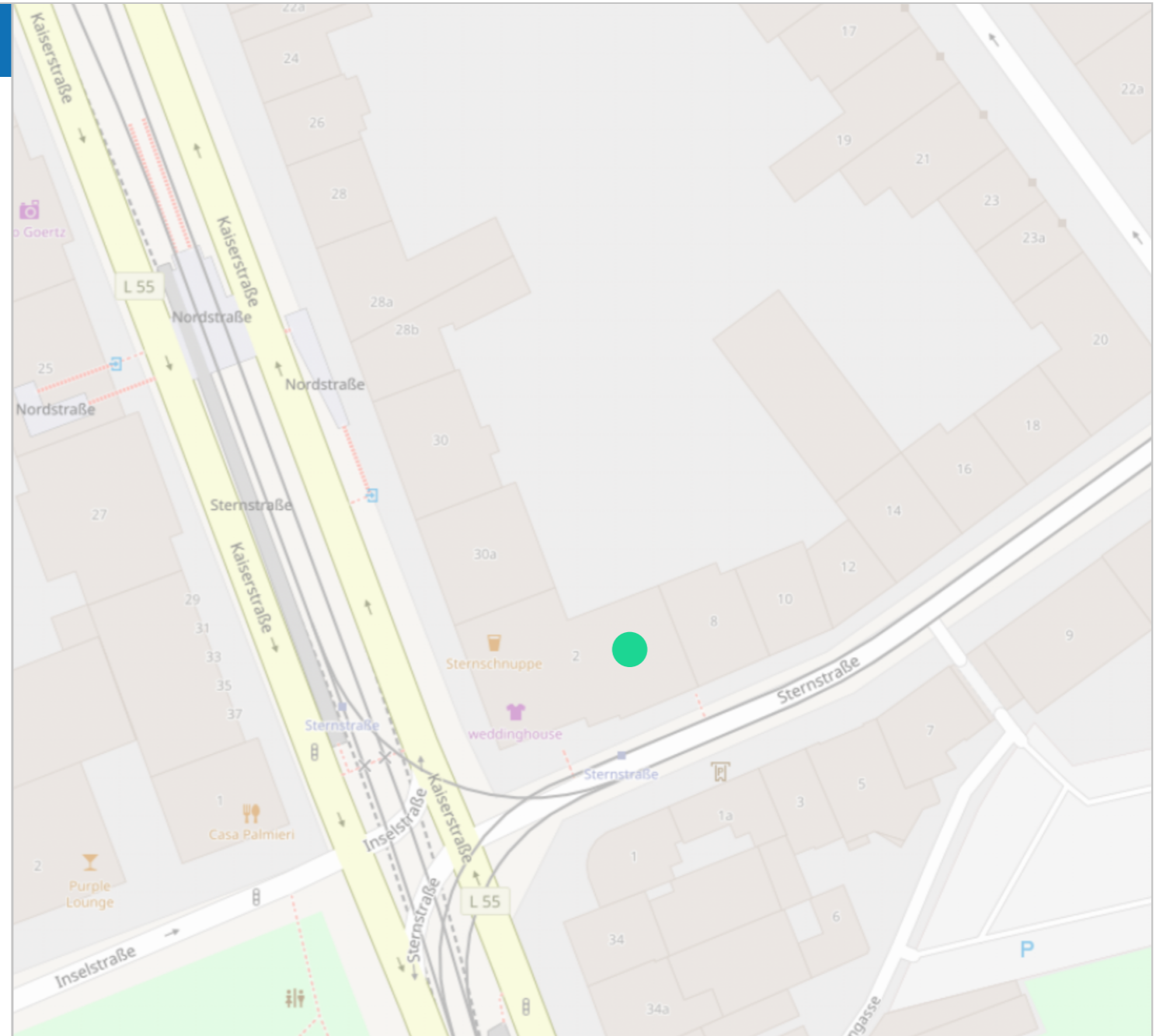
Cardaunstr.

Carlo-Schmid-Str.

Carlsplatz

Carlstor

Carmenstr.



- Konfiguration wie bei Referenztabellen
- Widget **featureSuggest** - dynamisches Laden und Autocomplete

```
model {
  uid "hausModel"
  table "haus"
  fields+ {
    name "strasse"
    title "Strasse"
    type "relatedFeature"
    relation.modelUid "strasseModel"
    foreignKey.name "strasse_id"
    widget.type "featureSuggest"
  }
}

model {
  uid "strasseModel"
  table "strasse"
}
```



# 1:M Verknüpfung: "Strasse" hat mehrere "Häuser"

**Sternstraße**

Name:

Häuser:

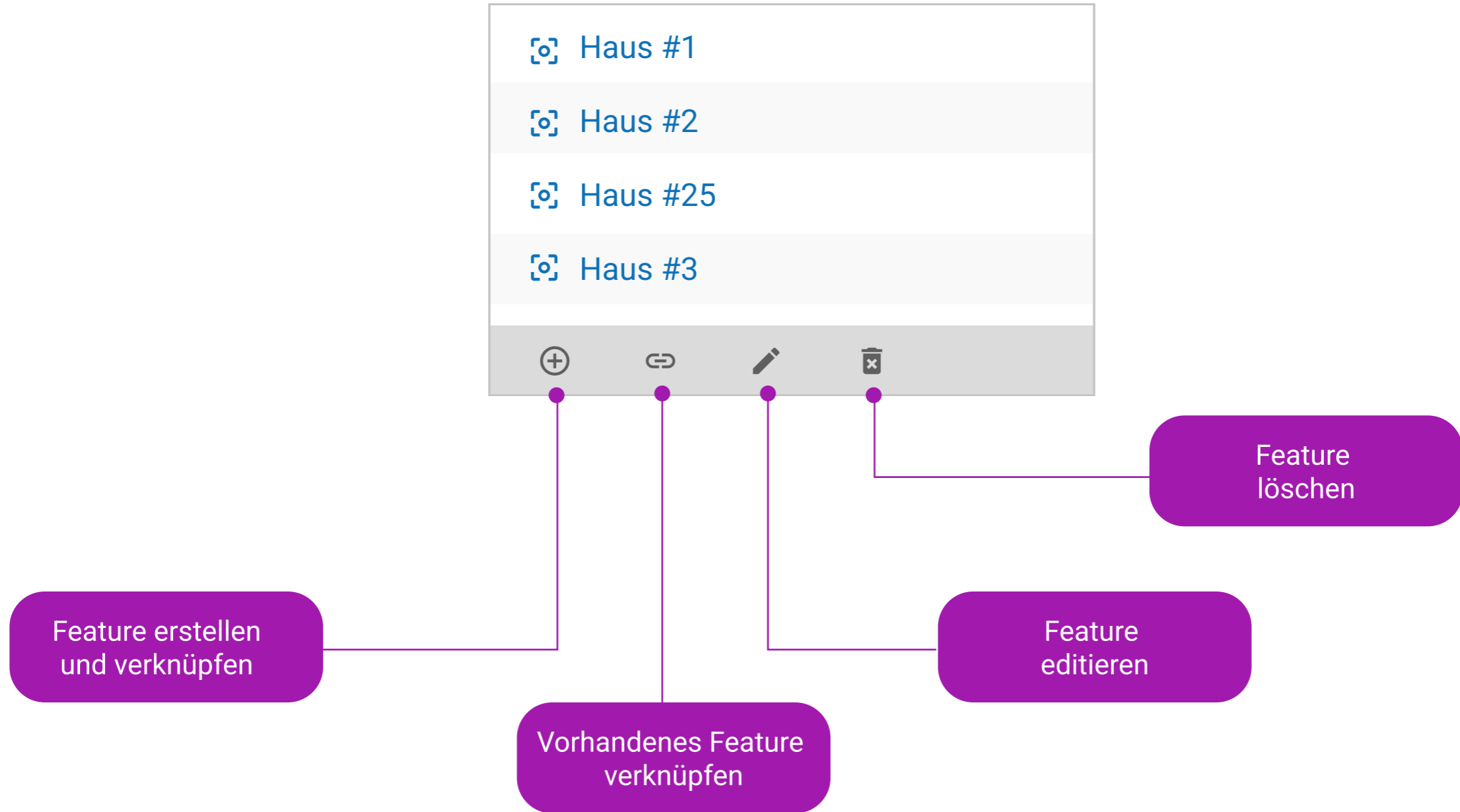
- Haus #1
- Haus #2
- Haus #25
- Haus #3
- Haus #5
- Haus #71
- Haus #123

- Feld Typ: **relatedFeatureList**
- beide Seiten der Verknüpfung konfigurieren:  
**hausModel.strasse <-> strasseModel.haus**
- Widget **featureList**

```
model {
  uid "hausModel"
  table "haus"
  fields+ {
    name "strasse"
    title "Strasse"
    type "relatedFeature"
    relation.modelUid "strasseModel"
    relation.fieldName "haus"
    foreignKey.name "strasse_id"
    widget.type "featureSuggest"
  }
}

model {
  uid "strasseModel"
  table "strasse"
  fields+ {
    name "haus"
    title "Häuser"
    type "relatedFeatureList"
    relation.modelUid "hausModel"
    relation.fieldName "strasse"
    widget.type "featureList"
  }
}
```

## Widget "featureList"





# N:M Verknüpfung: "Strassen" und "Stadtteile"

<b>Sternstraße</b>	
Name	<input type="text" value="Sternstraße"/>
Stadtteil	<ul style="list-style-type: none"><li><input type="checkbox"/> Altstadt</li><li><input type="checkbox"/> Pempelfort</li><li><input type="checkbox"/> Derendorf</li><li><input type="checkbox"/> Zoo</li></ul>
<p>⊕   🔗   ✎   🗑️</p>	

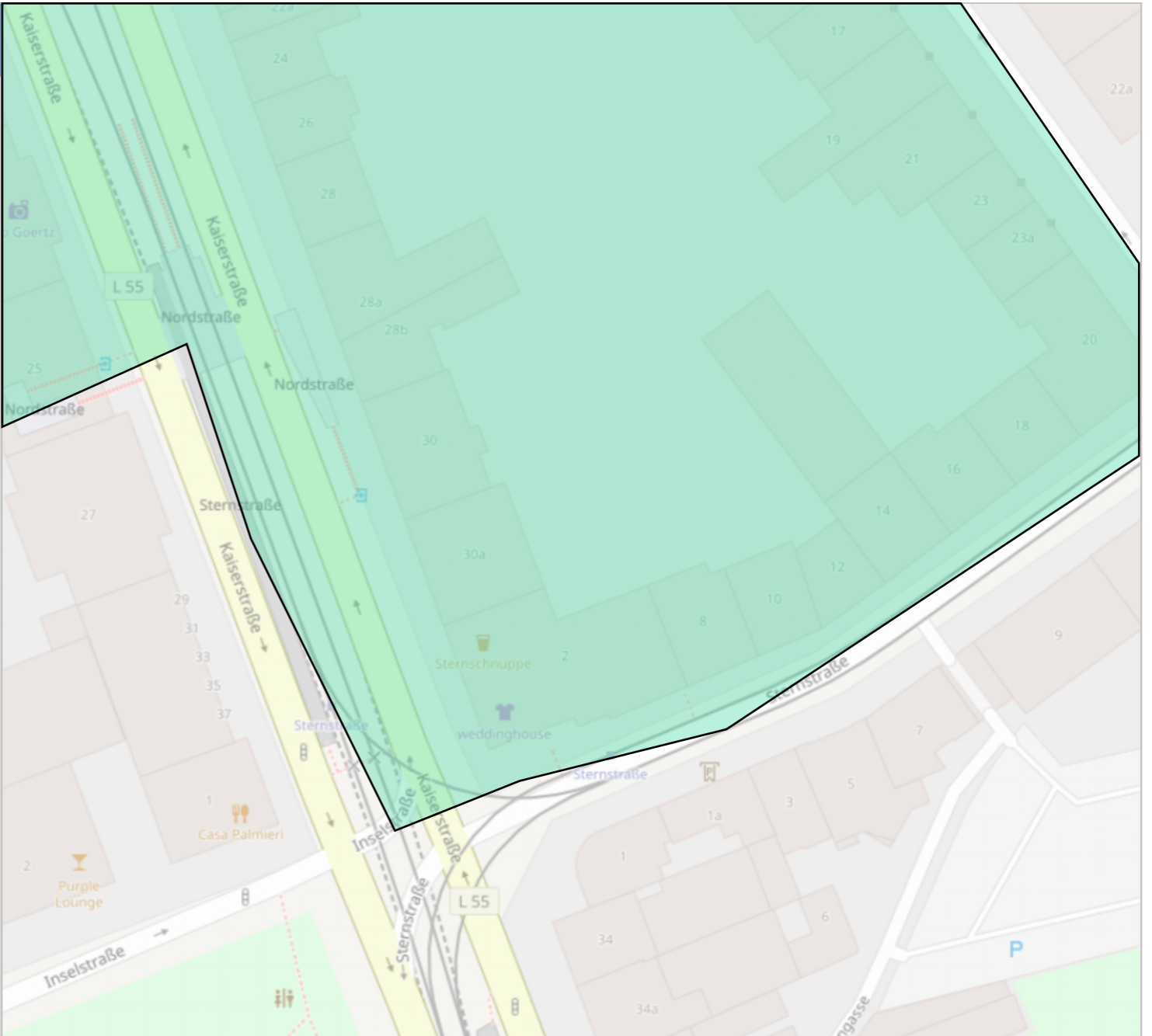
# N:M Verknüpfung: "Strassen" und "Stadtteile"

### Derendorf

Name:

Strassen:

- Brehmplatz
- Heinestraße
- Kaiserstraße
- Sternstraße



- Feld Typ: **relatedLinkedList**
- beide Seiten der Verknüpfung konfigurieren:

```
stadtteilModel.strasse <-> strasseModel.stadteil
```

- Linktabelle und Fremdschlüssel:

```
strasse_stadtteil.strasse_id <-> strasse.id
```

```
strasse_stadtteil.stadtteil_id <-> stadtteil.id
```

```
model {
  uid "strasseModel"
  table "strasse"
  fields+ {
    name "stadtteile"
    title "Stadtteile"
    type "relatedLinkedList"
    relation.modelUid "stadtteilModel"
    relation.fieldName "strasse"
    link.tableName "strasse_stadtteil"
    link.keyName "strasse_id"
  }
}
```

```
model {
  uid "stadtteilModel"
  table "stadtteil"
  fields+ {
    name "strassen"
    title "Strassen"
    type "relatedLinkedList"
    relation.modelUid "strasseModel"
    relation.fieldName "stadtteile"
    link.tableName "strasse_stadtteil"
    link.keyName "stadtteil_id"
  }
}
```

## Haus #1

Hausnummer

12a

Baujahr

1960

Dokumente



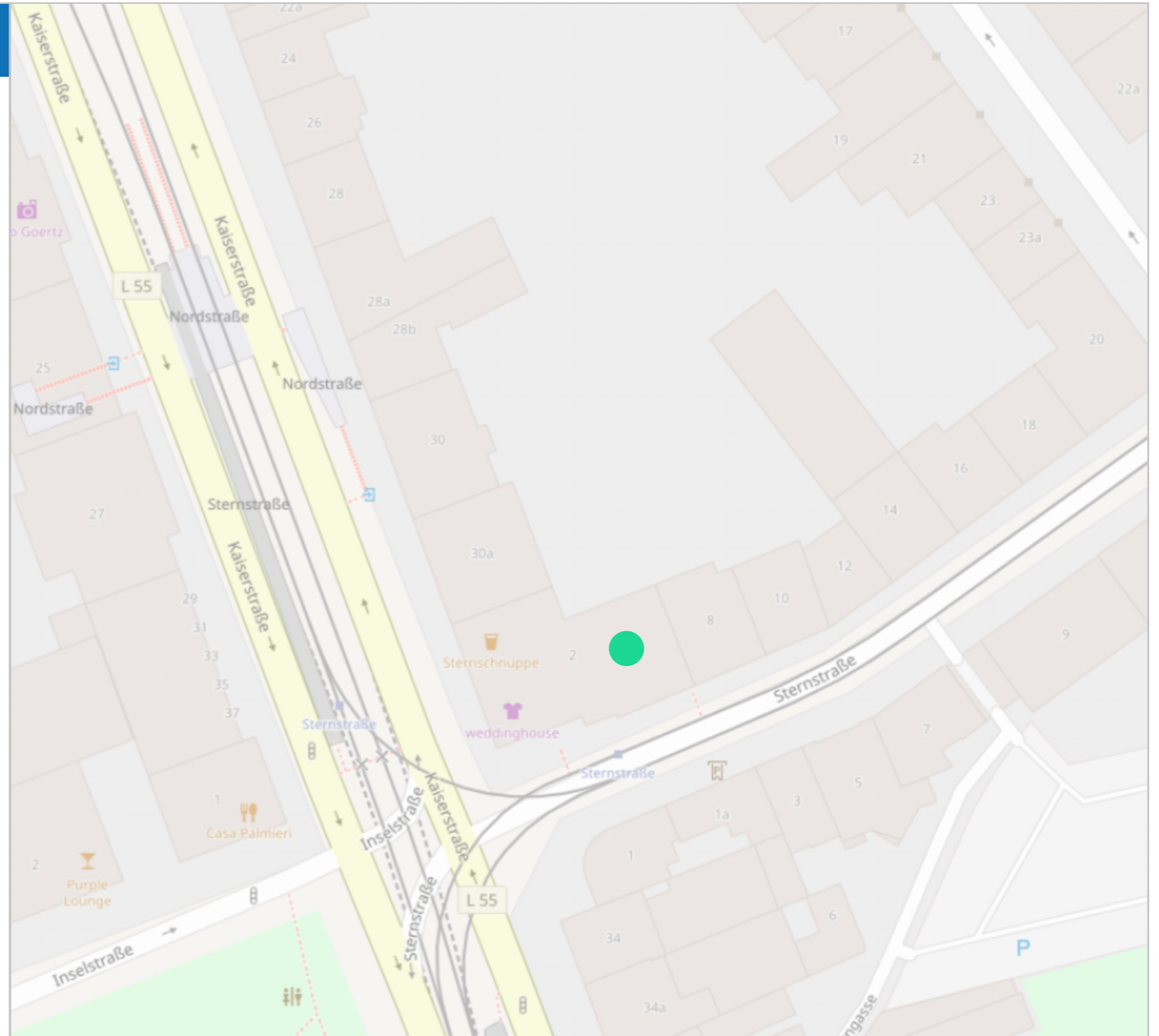
Ansicht vorne



Antrag



Beschreibung





Ansicht vorne

Haus

Haus #1

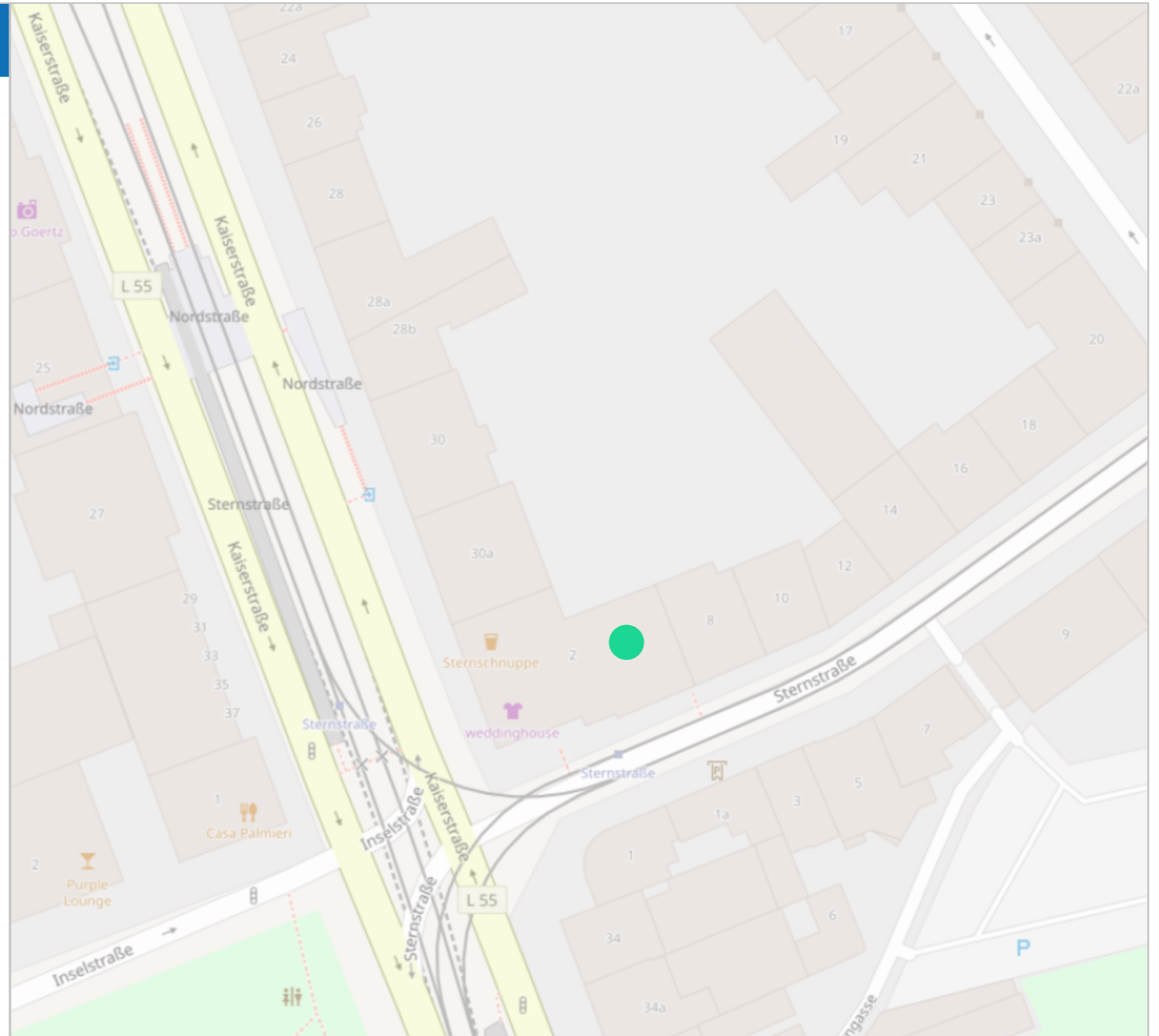
Titel

Ansicht vorne

Datei



IMG002.jpeg

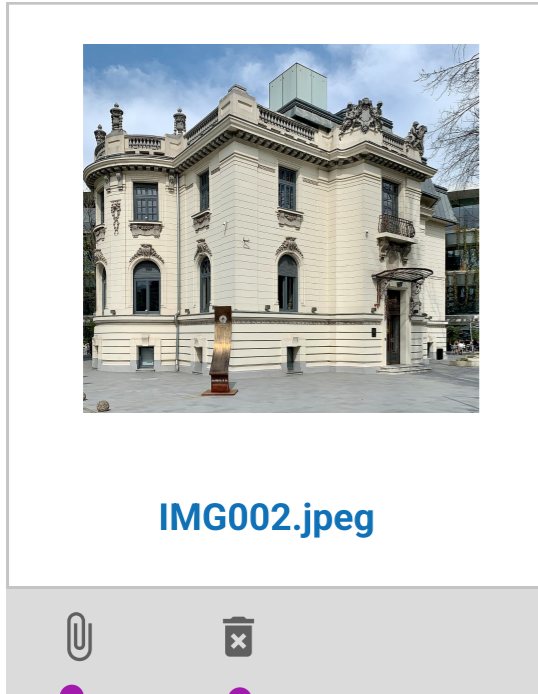


- Konfiguration wie bei 1:M
- Widgets **“file”** und **“documentList”**

```
model {
  uid "dokumentModel"
  table "dokument"
  fields+ {
    name "file"
    title "Datei"
    widget.type "file"
  }
  fields+ {
    name "haus"
    title "Haus"
    foreignKey.name "haus_id"
  }
}

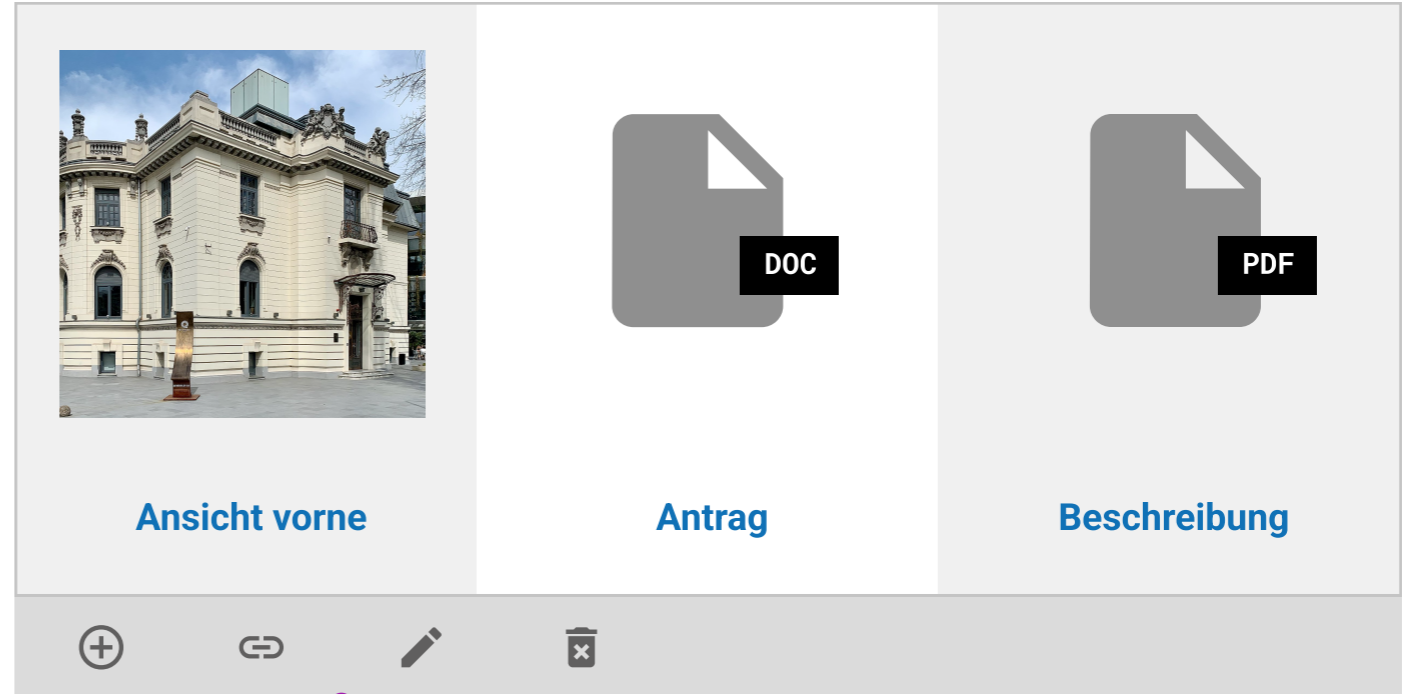
model {
  uid "hausModel"
  table "haus"
  fields+ {
    name "dokumente"
    type "relatedFeatureList"
    title "Dokumente"
    relation.modelUid "dokumentModel"
    relation.fieldName "haus"
    widget.type "documentList"
    widget.allowMime [ "pdf" "doc" "jpg" ]
  }
}
```

## Widgets "file" und "documentList"



Datei hochladen

Datei löschen



Buttons wie bei "featureList" Widgets

# Eingaben validieren

**Sternstraße 12a**

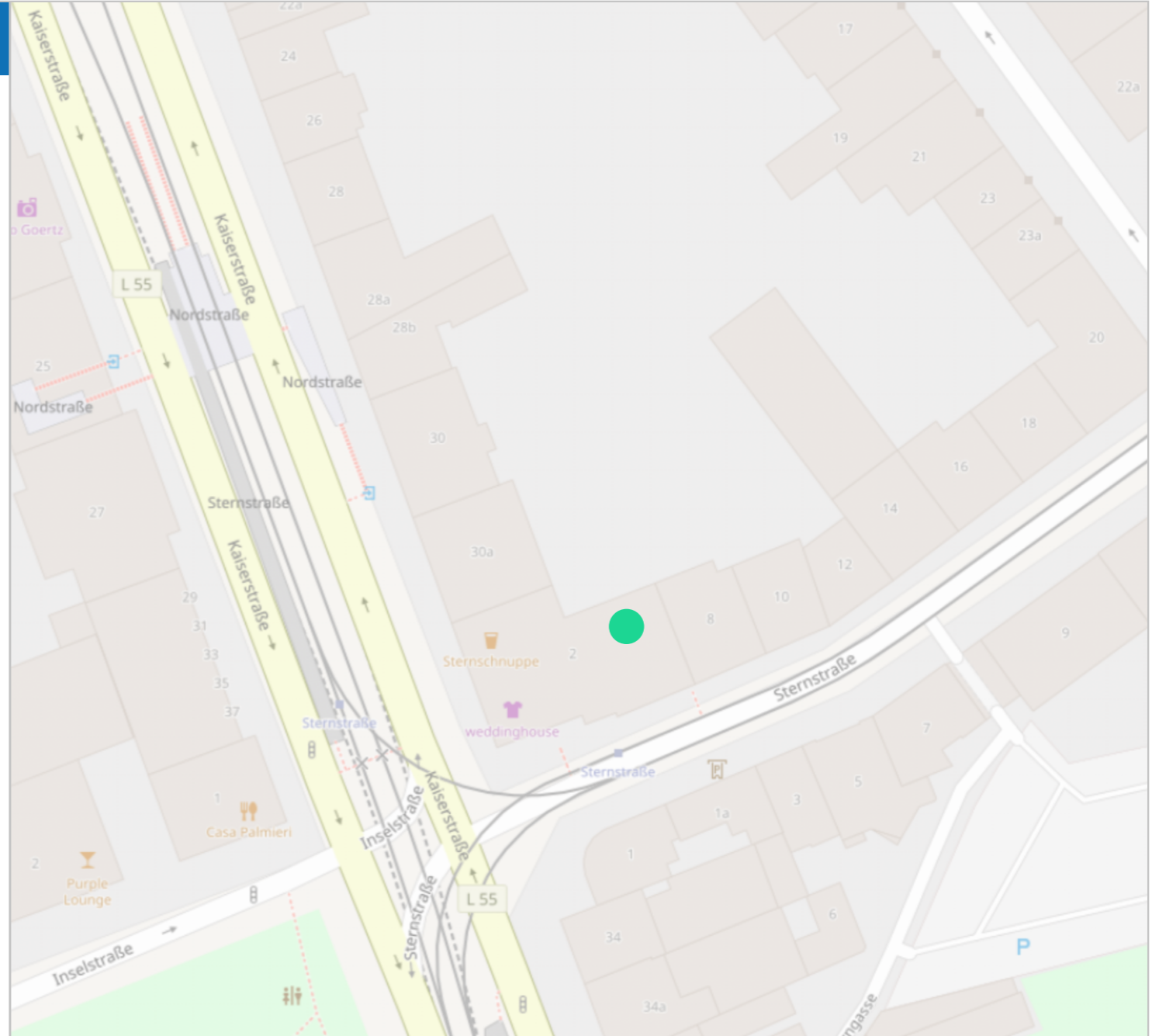
Hausnummer

Bitte Hausnummer eingeben

Baujahr

**3000**

Min. 1500, Max. 2020





- große Auswahl von Validierungsregeln
- konfigurierbare Fehlermeldungen
- statische Werte, z.B:

*Baujahr zwischen 1500 und 2020*

- dynamische Werte, z.B:

*Startdatum später als heute*

- geometrische Validierungen, z.B:

*der Punkt liegt innerhalb eines Polygons*

```
model {
  uid "hausModel"
  table "haus"
  fields+ {
    title "Hausnummer"
    validators+ {
      type "required"
      message
        "Bitte Hausnummer eingeben"
    }
  }
  fields+ {
    title "Baujahr"
    validators+ {
      type "range"
      message
        "Min. 1500, Max. 2020"
      minValue
        { type "static" value "1500" }
      maxValue
        { type "static" value "2020" }
    }
  }
}
```



<https://www.sqlalchemy.org>

Beliebte Datenbank und ORM Lösung für Python

**Vielen Dank für Ihre Aufmerksamkeit!**

