Templates und benutzerdefinierte Variablen

Tipps & Tricks zur Konfiguration der GBD WebSuite

Sebastian Vannahme

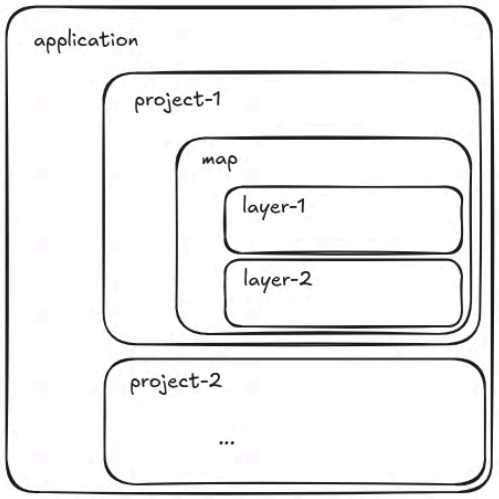
Düsseldorf, 4. November 2026

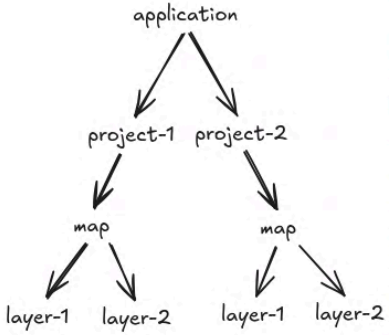
Agenda

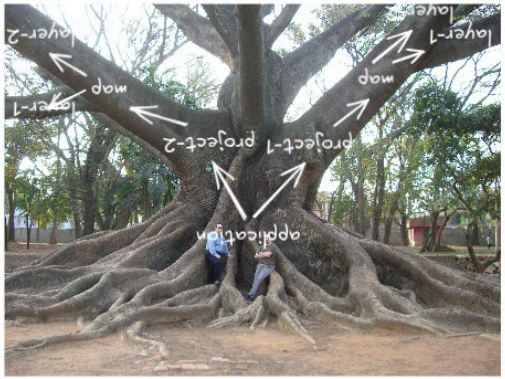
- Konfigurationsstruktur
- Konfiguration vs Laufzeit-Templates
- Features von Templates
- Beispiele: kombinieren der Features
- Vererbung
- Erben von Templates
- Benutzerdefinierte Variablen

Die GWS-Konfiguration ist ein Baum

```
application {
  projects+ {
    map {
      layers+ {
        ...
      }
    }
  }
}
```







Einstellungs-Vererbung

```
application {
  metadata.contact "gis@musterstadt.de"   ← hier definiert …

  projects+ {                             ← … gilt auch hier
    map {
      layers+ {
        ...                               ← … und auch hier
      }
      layers+ {                           ↙ … aber wird hier überschrieben
        metadata.contact "spezialist@musterstadt.de"
      }
    }
  }

}
```

- Gilt für: `metadata`, `client` – **und auch `templates`**

Templates in der Konfiguration

Darüber sprechen wir *nicht*:

```
{
    @PROJECT "my_project", "Mein Projekt"

    @POSTGRES_LAYER "public.my_table" "Meine Postgres Layer"

    @TILE_LAYER "https://osmtiles.gbd-consult.de/..." "Meine Tile Layer"

    @HINTERGRUNDKARTEN
}
```

… weil der Konfigurator kommt, siehe nächster Talk

Was sind (Laufzeit-) Templates?

- Steuern die Darstellung zur Laufzeit
- wenn ein Anwender mit der App interagiert

- Attributsdarstellung bei Abfrage (Description)
- Suchergebnisse und Listen (Teaser, Title)
- Druckvorlagen (HTML oder QGIS-Layout)
- Startseite der Anwendung

Templates einbinden – inline oder Datei

```
map.layers+ {
    type "postgres"
    title "Spielplätze"

    templates+ {
        subject "feature.title"
        type "html"
        text "Spielplatz: {name}"
    }

    templates+ {
        subject "feature.description"
        type "html"
        path "/data/templates/attributstabelle.cx.html"
    }
}
```

- Datei ist sinnvoll, wenn das Template komplex ist
- Datei kann von mehreren Layern geteilt werden

Vererbung von Templates

```
projects+ {
    templates+ {
        subject "feature.description"
        type "html"
        text "Attributsanzeige für diese Layer deaktiviert."
    }

    map.layers+ {
        title "Bäume"
        ## Kein Template → erbt vom Projekt
    }

    map.layers+ {
        title "Baustellen"
        ## Eigenes Template → überschattet Projekt-Template
        templates+ {
            subject "feature.description"
            type "html"
            path "/data/templates/baustellen_description.cx.html"
        }
    }
}
```

Template am Projekt gilt für alle Layer ohne eigenes Template

Features der Template-Sprache

`{feld}` – Attributwerte ausgeben

```
<tr>
    <th>Name:</th> <td>{name}</td>
</tr>
```

Zugriff auf Eigenschaften

```
{layer.title}

{project.map.rootLayer.layers[0].uid}
```

`@if` – bedingte Ausgabe

```
@if beschreibung != ''
    {beschreibung}
@end
```

Dokumentation: https://github.com/gebrkn/jump

Besonderheiten für Druckvorlagen

```
@page ( width="210" height="297" margin="20" )

@header
    <div class="kopf">Übersichtskarte</div>
@end

@footer
    <div class="fuss">Maßstab 1:{scale} | Seite {page} von {numpages}</div>
@end

@map ( width="180" height="120" )

@legend()
```

Benutzerdefinierte Variablen

Werte in der Konfiguration definieren, im Template lesen

```
vars {
    impressum_url  "https://www.musterstadt.de/impressum"
}
projects+ {
    vars {
        showEmail True
    }
    metadata.contact "vannahme@gbd-consult.de"
}
```

- Wert ändern → nur Konfiguration anfassen, nicht die Template-Datei
- Benutzerdefinierte Variablen werden *nicht* vererbt.

In der Template-Datei (`vorlage.cx.html`):

```
@if project.vars.showEmail
    Rückfragen bitte an: {project.metadata.contact}
@end

Unser Kontaktformular finden Sie unter {app.vars.impressum_url}.
```

Beispiel 1: @with

`@with … @else` behandelt den Fall, dass eine Variable nicht gesetzt ist

Konfiguration (Projekt mit spezifischer Legenden-Auswahl):

```
vars {
    legendLayers "my_layer_uid"
}
```

Druckvorlage:

```
@with project.vars.legendLayers as legendLayers
    @legend(layers=legendLayers)
@else
    @legend()
@end
```

- Projekte **mit** `legendLayers`: nur die gewählte Gruppe in der Legende
- Projekte **ohne** `legendLayers`: alle Layer in der Legende
- Eine gemeinsame Druckvorlage funktioniert für alle Projekte

Beispiel 2: Features aktivieren/deaktivieren

Druckvorlage:

```
@if project.vars.smallerLegend
    @let legendLayers
        {project.map.rootLayer.layers[0].uid}
    @end

    @legend(layers=legendLayers)
@else
    @legend()
@end
```

Konfiguration:

```
vars {
    smallerLegend True
}
```

Vielen Dank.
Haben Sie Fragen?

Geoinformatikbüro Dassau GmbH
Rethelstrasse 153
D-40237 Düsseldorf
Tel. +49 (0)211 699 377 50
Fax. +49 (0)171 468 75 40
info@gbd-consult.de

Die in der Präsentation enthaltenen Inhalte sind nicht zur Weitergabe oder Verbreitung legitimiert. Alle Rechte vorbehalten. Die verwendeten Logos, Marken und sonstige geschützte Inhalte unterliegen dem jeweiligen Urheberrecht.