

Inhaltsverzeichnis

| | |
|---|--------|
| Introduction | 1.1 |
| Vorarbeiten und generelle Informationen | 1.2 |
| Basiswissen und Ressourcen | 1.3 |
| Basiswissen NPM | 1.3.1 |
| Neue Features | 1.4 |
| Entwicklungsssetup | 1.5 |
| Architektur | 1.6 |
| Konfiguration | 1.7 |
| Globale Konfiguration | 1.7.1 |
| Portalkonfiguration | 1.7.2 |
| Masterportal Admin | 1.8 |
| Dokumentation | 1.9 |
| Druck | 1.10 |
| Übungsaufgaben | 1.10.1 |
| Addons | 1.11 |
| Übungsaufgaben | 1.11.1 |
| Migration V3 | 1.12 |
| Weitere Übungsaufgaben | 1.13 |
| Allgemeine Konfiguration | 1.13.1 |
| Layer | 1.13.2 |
| Suche | 1.13.3 |
| Suche | 1.13.4 |
| Build | 1.13.5 |
| Erweiterte Konfiguration | 1.14 |



Masterportal Logo

Einführung ins Masterportal

Herzlich Willkommen beim **Mastering the Masterportal** Workshop. 🙌

Dieser Workshop wurde für die Verwendung auf der [OSGeo-Live 16.0 DVD](#) entwickelt und soll Ihnen einen umfassenden Überblick über das Open Source Masterportal als Web-GIS-Lösung geben.

Allgemeines

Das Masterportal Projekt ermöglicht die Erstellung modularer und somit individueller Geoportale. Zum Aufbau einer Nutzer- und Entwicklungsgemeinschaft wurde die Implementierungspartnerschaft ins Leben gerufen, die mittlerweile aus über 35 PartnerInnen auf kommunaler, föderaler und Bundesebene besteht. Die wichtigsten Vernetzungs- und Entscheidungstreffen sind:

- Strategisches Komitee (steuert und kontrolliert die strategische Richtung des Masterportals)
- Technisches Komitee (unterstützt das Strategische Komitee in technischen Fragen)
- Produktpflege (Technische Weiterentwicklung, Release Management, etc.)
- Maintainergroup (unterstützt die Produktpflege bei der technischen Weiterentwicklung, Bearbeitung von PullRequests usw.)
- Produktmanagement (koordiniert organisatorische Angelegenheiten, Öffentlichkeitsarbeit, Veranstaltungsplanung usw.)

Neben den regelmäßig stattfindenden Gremiensitzungen werden verschiedene Workshops organisiert, zum Beispiel zur Ersteinrichtung der Software oder zu speziellen technischen Themen wie der Integration von sicheren Geodatendiensten.

Neben den Partnern aus der öffentlichen Verwaltung gibt es verschiedene Unternehmen, die Support und Wartung anbieten und zur Weiterentwicklung von Masterportal beitragen.

Zentrale Links:

- [Website](#)
- [Twitter](#)
- [Code](#)

Autoren

- Hannes Blitza (blitza@terrestris.de)

Zuarbeit

- LGV HH, u.a. Dirk Rohrmoser (dirk.rohrmoser@gv.hamburg.de)
- FOSSGIS 2024Last modified: 2024-03-21 13:08:21

Vorarbeitung und generelle Informationen

Bevor wir mit dem Workshop starten können, führen Sie bitte die folgenden Schritte aus:

- Rechner mit OSGeoLive-Medium hochfahren

Es wird angenommen, dass die OSGeoLive bereits installiert ist. Falls nicht, kann der Live-Modus gestartet werden:

- *Lubuntu ohne Installation ausprobieren* auswählen
- Benutzer: user; Passwort: user (wird vermutlich nicht benötigt)

Warning

⚠ Sollte die Tastaturbelegung noch auf **US** gestellt sein (siehe unten rechts in der Taskleiste), öffnen Sie ein Terminal (`strg+T`) und führen Sie folgenden Befehl aus: `setxkbmap -layout de`



Die Startansicht der OSGeo Live 16.0 auf Ihrem Rechner.

FOSSGIS 2024Last modified: 2024-03-21 13:08:21

Basiswissen

Dieser Teil dient eher als Nachschlagewerk, es kann direkt losgelegt werden mit dem [Entwicklungssetup](#)

- [npm](#)

Linksammlung

- <https://www.masterportal.org/>
- <https://bitbucket.org/geowerkstatt-hamburg/masterportal>
- <https://geoportal-hamburg.de/geo-online/>
- <https://geoportal.de>
- <https://maps.stuttgart.de/>
- <https://geoportal.freiburg.de/freigis/>
- <https://geoportal.muenchen.de/portal/master/>
- <https://geoportal.brandenburg.de/de/cms/portal/start>

Weitere Geoportale: <https://www.masterportal.org/referenzen.html>

FOSSGIS 2024Last modified: 2024-03-21 13:08:21

npm

npm ist der Paketmanager für Node.js (eine JavaScript-Laufzeitumgebung) und die weltweit größte Software-Registry (mehr als 600k Pakete) mit ca. 3 Milliarden Downloads pro Woche.



npm Logo

npm kann genutzt werden, um ...

- Pakete an Anwendungen anpassen oder diese so einbinden, wie sie sind.
- Eigenständige Tools herunterladen.
- Pakete ohne Herunterladen mit npx ausführen.
- Code mit jedem npm-User teilen, überall.
- Code für bestimmte Entwickler beschränken.
- Virtuelle Teams (orgs) bilden.
- Mehrere Versionen von Code und Code-Abhängigkeiten verwalten.
- Anwendungen einfach aktualisieren, wenn der zugrunde liegende Code aktualisiert wird.
- Andere Entwickler finden, die an ähnlichen Problemen arbeiten.

package.json

Der Befehl `npm init` in Ihrem Projektordner öffnet einen interaktiven Dialog zur Erstellung eines npm-Projekts. Das Ergebnis ist die `package.json` mit allen wichtigen Einstellungen, Skripten und Abhängigkeiten Ihres Projekts.

```
{
  "name": "name_of_your_package",
  "version": "1.0.0",
  "description": "This is just a test",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "repository": {
    "type": "git",
    "url": "http://github.com/yourname/name_of_your_package.git"
  },
  "author": "your_name",
  "license": "ISC"
}
```

Weitere Infos hier: [npm docs package.json](#).

Installieren von packages mittels npm

Der einfachste Weg, neue Pakete zu installieren, ist die Nutzung der [CLI](#):

```
npm install packagename
```

Das installierte Paket ist anschließend im Subfolder `node_modules` zu finden.

Node version manager NVM

- bash script um mehrere node Versionen zu verwalten
- Siehe [hier](#)

```
wget -qO- https://raw.githubusercontent.com/creationix/nvm/v0.34.0/install.sh | bash
nvm i v8
```

FOSSGIS 2024Last modified: 2024-03-21 13:08:21

Neue Features

Im Changelog werden detailliert sämtliche Änderungen neuer Releases gelistet. Es werden kontinuierlich Bugfixes und Sicherheitspatches bereitgestellt, zudem kommen neue Features hinzu.

[Link zum Changelog](#)

Neue Features der letzten Releases

- Print: An option "Improve scaling resolution" is implemented for 3d Layers to supply an improved and better resolution.
- Added translations for Sensor Theme.
- Added square as geometryType for drawingTool.
- New Parameter sldVersion for legend configuration to define a Styled Layer Descriptor for the GetLegendGraphic requests.
- possibility to keep more than one tool open at the same time
- Add additional wfs and wms layer param 'gfiThemeSettings' to provide the possibility of additional gfiTheme settings.
- New Tool 3D-Modeler has been added to draw/import 3D models into the 3D map.
- WebGL render pipeline supports now also VectorTile layers.
- Printing is now for 3d supported.
- Added new module login button that allows the user to login with an OIDC IDM. The resulting auth token is added to all service requests (can be configured).
- urls in gfiAttributes can now be customized and shown as html tags (e.g. a, img, iframe) with custom properties

Beispiel für einen Breaking Change

- **v2.34.0** The Virtual City Planner tool (virtualcityPLANNER) has been removed

Roadmap

- Es wird derzeit viel am 3.0.0 Release gearbeitet. Erste stabile Version Mitte 2024
- Support und Bugfixes bis Ende 2024

FOSSGIS 2024Last modified: 2024-03-21 13:08:21

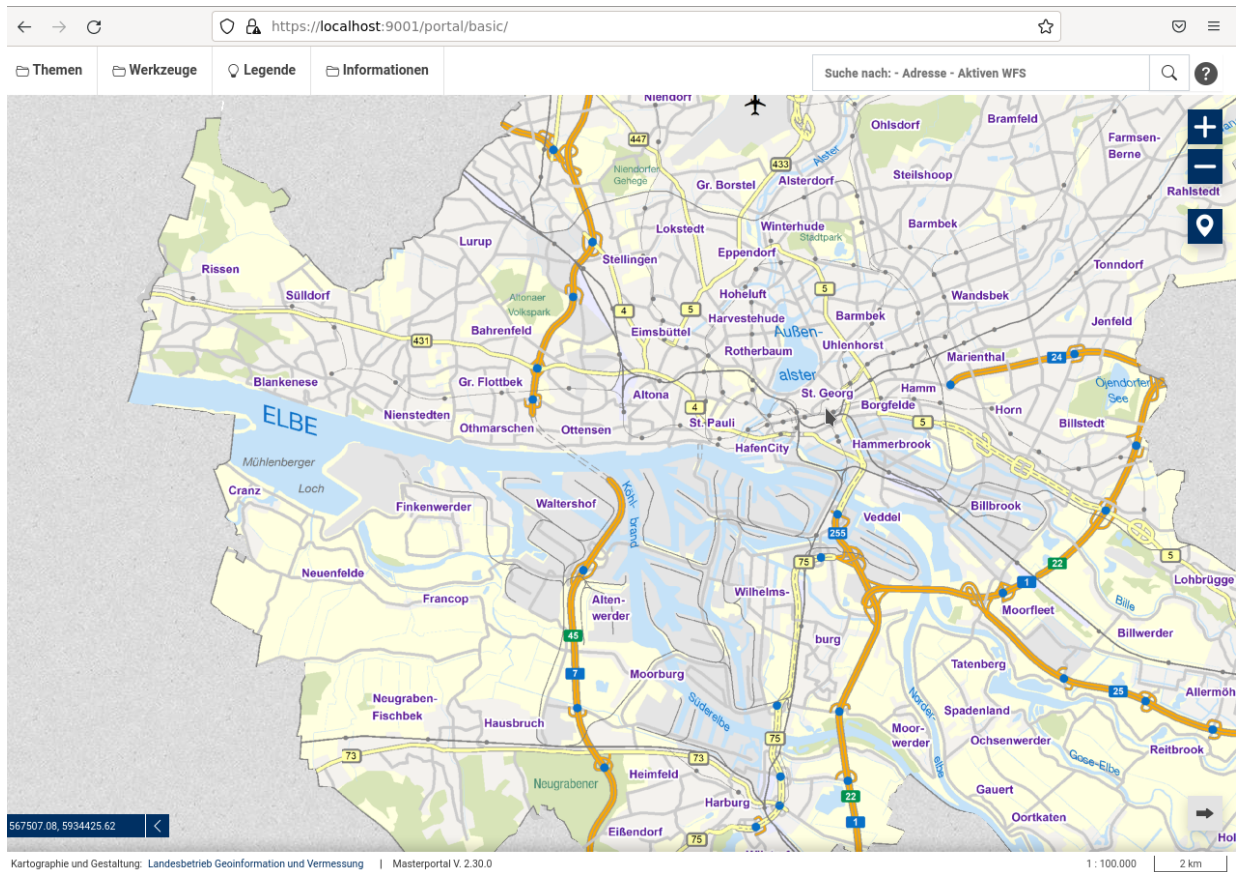
Repository und Entwicklungssetup

1. Öffnen Sie das Terminal und führen Sie den Befehl `pwd` aus.
2. Sie sollten sich im Pfad `/home/user` befinden.
3. Führen Sie den Befehl `git clone https://geowerkstatt@bitbucket.org/geowerkstatt-hamburg/masterportal.git` aus, um das Masterportal Repository auf Ihre Festplatte zu kopieren. Navigieren Sie anschließend in das neue Verzeichnis per Befehl: `cd masterportal`.

Wie in vielen modernen Javascript Projekte, wird auch für das Masterportal ein [Node.js](#) Framework zur Entwicklung genutzt. Mithilfe des Paketsmanager [npm](#) werden sämtliche Bibliotheken und Abhängigkeiten gemanaged und installiert, wie beispielsweise [webpack](#), der als *module bundler* fungiert.

Eine ausführliche Beschreibung dieser Entwicklungstools- und Frameworks würde den Rahmen dieses Workshops sprengen, die benötigten Infos werden im Rahmen dieses Workshops gegeben. Eine kurzen Überblick über npm ist [hier](#) zu finden.

1. Führen Sie `node -v`, um die installierte Version von `node` auszugeben. Falls `node` nicht installiert ist, oder die Version `< 18.16.0` oder `> 20.9.0` ist, folgende Schritte ausführen:
 - `wget -qO- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.7/install.sh | bash`
 - `source ~/.bashrc` Refresh der `.bashrc`. Notwendig, um neues command `nvm` auszuführen
 - `nvm install v20.9.0`
2. Es soll auf der Version `2.42.0` gearbeitet werden, hierzu sind folgend Befehle auszuführen:
 - `git fetch origin`
 - `git checkout v2.42.0`
3. Installieren Sie alle benötigten Abhängigkeiten des Masterportals-Projekts: `npm i`.
4. Starten Sie anschließend den Entwicklungsserver: `npm run start`.
5. Nun wird der Masterportal-Quellcode kompiliert und `webpack` erstellt den *dev build*, der anschließend - sobald die Nachricht `compiled successfully` im Terminal erscheint, im Browser unter der Adresse `localhost:9001/portal/basic` aufgerufen werden kann.
6. Möglicherweise tauchen viele Logs mit der Nachricht `ENOSPC: System limit for number of file watchers reached` auf. In diesem Fall `strg+c` drücken um den Dev-Server zu stoppen. Dann `echo fs.inotify.max_user_watches=524288 | sudo tee -a /etc/sysctl.conf && sudo sysctl -p` ausführen und anschließend den Dev-Server wieder starten mit `npm run start`.



Startansicht des Portals basic.



Weiterführende Infos zum Dev-Setup unter:

<https://bitbucket.org/geowerkstatt-hamburg/masterportal/src/latest/doc/setup.md>

FOSSGIS 2024 Last modified: 2024-03-21 13:08:21

Architektur

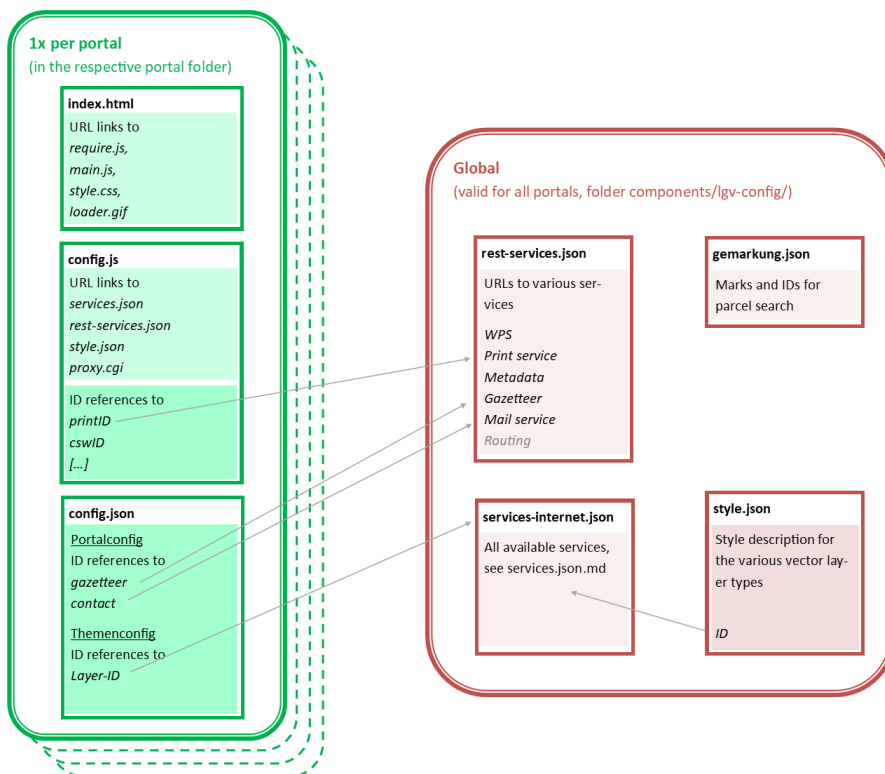
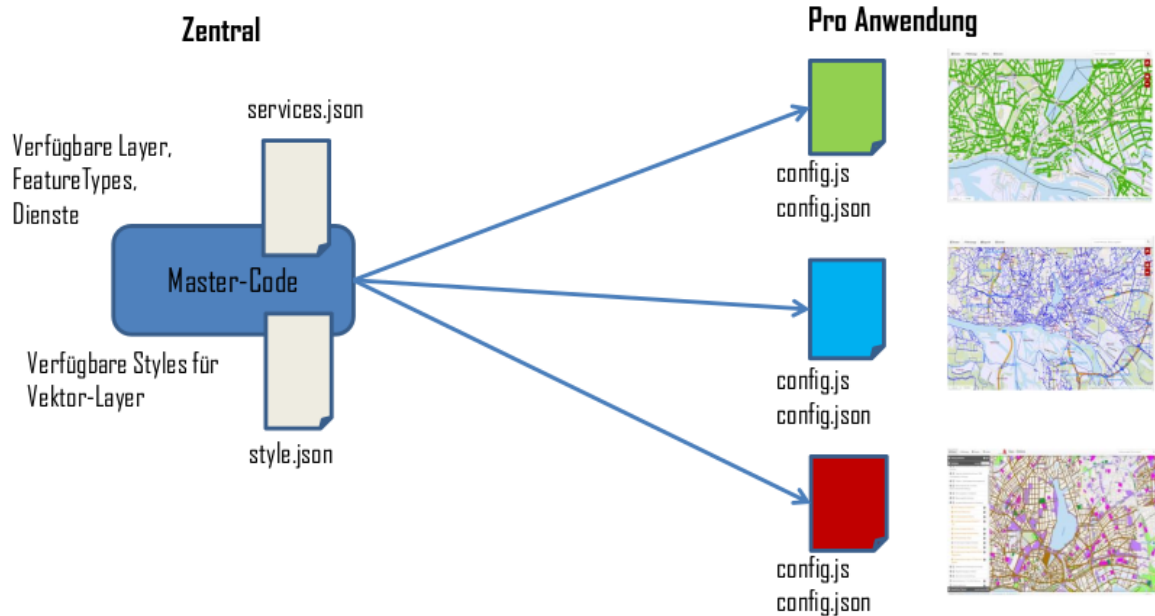
Paradigmen

- OpenSource (MIT)
- Standardbasiert (OGC Standards, Rest APIs)
- Modular und konfigurabel
- Responsiv
- Rein clientseitig
- Gut dokumentiert für NutzerInnen und Devs: `coding conventions` , `linting` , etc.
- Nutzung weit verbreiteter und *well maintained* Bibliotheken (z.B. OpenLayers, Vue.js, Vuex)

--> Schlanker Core. Zentrale Funktionen in der `MasterportalAPI` . Erweiterung durch Addons.

Konfiguration

Der Applikationskontext teilt sich in mehrere Dateien auf, die nach Belieben angepasst werden können. Teilweise können diese automatisch erstellt werden, darauf wird im nächsten Kapitel eingegangen.



Globale Konfiguration

services.json

Dies ist die zentrale Konfiguration für sämtliche Layer (WMS, WFS, WMTS, SensorThings-API, GeoJSON and weitere) die in den Portal dargestellt werden sollen. Sie wird in den jeweiligen Portalkonfigurationen (`config.js`) referenziert. Es kann auch auf einen API-Endpoint verwiesen werden, der die `services.json` Datei generiert - etwa über einen [Dienstemanager](#).

Jeder Layertyp benötigt unterschiedliche Parameter, wobei einige stets obligatorisch und andere optional sind.

WMS

Ein Beispiel für einen **WMS-Layer**:

```
{
  "id" : "8",
  "name" : "Aerial View DOP 10",
  "url" : "https://geodienste.hamburg.de/HH_WMS_DOP10",
  "typ" : "WMS",
  "layers" : "1",
  "format" : "image/jpeg",
  "version" : "1.3.0",
  "singleTile" : false,
  "transparent" : true,
  "tilesize" : "512",
  "gutter" : "0",
  "minScale" : "0",
  "maxScale" : "1000000",
  "gfiAttributes" : "ignore",
  "layerAttribution" : "nicht vorhanden",
  "legend" : false,
  "layerSequence": 1,
  "datasets" : [
    {
      "md_id" : "25DB0242-D6A3-48E2-BAE4-359FB28491BA",
      "rs_id" : "HMDK/25DB0242-D6A3-48E2-BAE4-359FB28491BA",
      "md_name" : "Digitale Orthophotos 10cm - FHHNET",
      "bbox" : "461468.97,5916367.23,587010.91,5980347.76",
      "kategorie_opendata" : [
        "Sonstiges"
      ],
      "kategorie_inspire" : [
        "nicht INSPIRE-identifiziert"
      ],
      "kategorie_organisation" : "Landesbetrieb Geoinformation und Vermessung"
    }
  ]
}
```

Hier sind zahlreiche Parameter angegeben, die den Layer optimal definieren. Beispielsweise bewirkt der `gfiAttributes: false` Parametereinstellung, dass keine Sachdatenabfrage (GetFeatureInfo) ausgeführt wird. `tilesize" : "512"` sorgt dafür, dass die Kacheln des WMS in einer Größe von 512x512 Pixeln abgefragt werden (default 256px).

Hinweis

Die Parameter bedingen sich teilweise gegenseitig bzw. sind voneinander abhängig. Bei ungewünschten Verhalten der Layer sollte die vollständige Dokumentation ausführlich studiert werden.

Ein Beispiel: `singleTile: true` hat zur Folge, dass der Ausschnitt als einzelne Kachel vom WMS abgefragt wird. `tileSize` hat dann natürlich keinen Effekt mehr.

WMTS

Ein Beispiel, das nur wenige Konfigurationsparameter bedarf ist ein `WMTS`, wenn der Parameter `optionsFromCapabilities` gesetzt ist:

```
{
  "id": "2020",
  "name": "EOC Basemap",
  "capabilitiesUrl": "https://tiles.geoservice.dlr.de/service/wmts?SERVICE=WMTS&REQUEST=GetCapabilities",
  "typ": "WMTS",
  "layers": "eoc:basemap",
  "optionsFromCapabilities": true
}
```

Hier muss bedacht werden, dass die Kacheln in der Projektion des ersten TileGrid angefragt werden, dass im Capabilities auftaucht (zumeist EPSG:3857). OpenLayers übernimmt im Client die [Reprojektion der Kacheln](#) in die aktuelle Kartenprojektion.

[Link zur vollständigen Dokumentation](#)

rest-services.json

Hier werden alle Services definiert, die nicht direkt für die visuelle Darstellung von Daten benötigt werden:

- Print services (MapFish)
- Metadata sources (CSW HMDK)
- BKG geocoding service
- Gazetteer URL
- WPS
- Email Services

```
[
  {
    "id": "1",
    "name": "CSW HMDK Summary",
    "url": "http://metaver.de/csw?service=CSW&version=2.0.2&request=GetRecordById&typeName=csw:Record&elementSetName=cs",
    "typ": "CSW"
  },
  {
    "id": "mapfish-terrestris",
    "name": "Testserver Print",
    "url": "https://10.133.7.xx/print/",
    "typ": "Print"
  },
  {
    "id" : "11",
    "name" : "Komoot Photon Suche",
    "url" : "https://photon.komoot.io/api/?",
    "typ" : "WFS"
  },
  {
    "id" : "80002",
    "name" : "Email Service by PHP",
    "url" : "https://geoportal-hamburg.de/smtp/sendmail.php",
    "typ" : "EmailService"
  }
]
```

style.json

Vektordaten wie WFS und GeoJSON werden clientseitig gestyled (gegenüber WMS und WMTS, die serverseitig gestyled werden).

Das Masterportal liest hierzu die `style.json` aus, in der x-beliebige Style im OpenLayers eigenen Stil-Format definiert werden. Bei der Konfiguration von Vektorlayern kann einem Layer ein bestimmter Stil zugewiesen werden, zudem können diese Stile für dynamisch hinzugefügte Layer (während der Laufzeit) verwendet werden oder Standard-Layer wie beispielsweise `MapMarker`.

Jeder Style beginnt mit einer `styleID` und darauffolgend mit der Definition von Stilregeln.

```
{
  "styleId": "blue-point",
  "rules": [
    {
      "style": {
        "circleRadius" : 6,
        "circleStrokeColor": [51, 102, 255, 1],
        "circleStrokeWidth": 2,
        "circleFillColor": [51, 102, 255, 1]
      }
    }
  ]
},
```

Der Stilregel können `conditions` hinzugefügt werden, die letztendlich attributives Styling ermöglichen:

```
{
  "styleId": "blue-point",
  "rules": [
    {
      "conditions": {
        "properties": [
          {
            "attrName": " housenumber",
            "value": [0, 100]
          }
        ]
      },
      "style": {
        "circleRadius" : 6,
        "circleStrokeColor": [51, 102, 255, 1],
        "circleStrokeWidth": 2,
        "circleFillColor": [51, 102, 255, 1]
      }
    }
  ]
},
```

[Link zur vollständigen Beschreibung der Conditions](#)

[Vollständige Konfiguration](#)

FOSSGIS 2024Last modified: 2024-03-21 13:08:21

Portalkonfiguration

config.js

Hier werden sämtliche Konfigurationen vorgenommen, die nicht direkt auf UI-Elementen oder Layern bezogen sind:

- Pfade zu Backends und weiteren Konfigurationsdateien
- Projektionsdefinitionen im Portal
- Liste an Addons
- Proxy Einstellungen
- Footer
- mousehover

[Link zur vollständigen Dokumentation](#)

config.json

Die `config.json` enthält die gesamte Konfiguration der Portal-Oberfläche. In ihr wird definiert, welche Elemente sich wo in der Menüleiste befinden, wo das initiale Kartenzentrum liegen soll und welche Layer geladen werden sollen. Des weiteren wird die Liste der Tools und Addons definiert und die dazugehörige Start-Konfiguration festgelegt.

Portalconfig

- Titel & Logo (`portalTitle`)
- Art der Themenauswahl (`treeType`)
- Starteinstellungen der Kartenansicht (`mapView`)
- Schaltflächen auf der Kartenansicht sowie mögliche Interaktionen (`controls`)
- Menüeinträge sowie Vorhandenheit jeweiliger Tools und deren Reihenfolge (`menu`)
- Typ und Eigenschaften des genutzten Suchdienstes (`searchBar`)
- Löschbarkeit von Themen (`layersRemovable`)

Beispiel aus der `basic` Portalkonfiguration:

```

"Portalconfig": {
  "treeType": "light",
  "searchBar": {
    "komoot": {
      "minChars": 3,
      "serviceId": "11",
      "limit": 20,
      "lang": "de",
      "lat": 53.6,
      "lon": 10.0,
      "bbox": "9.6,53.3,10.4,53.8"
    },
    "visibleVector": {
      "layerTypes": [
        "WFS"
      ]
    },
    "tree": {},
    "startZoomLevel": 9,
    "placeholder": "Suche nach: - Adresse - Aktiven WFS"
  },
  "mapView": {
    "backgroundImage": "../resources/img/backgroundCanvas.jpeg",
    "startCenter": [
      561210,
      5932600
    ],
    "extent": [
      510000.0,
      5850000.0,
      625000.4,
      6000000.0
    ],
    "startZoomLevel": 1
  },
  "menu": {
    "tree": {
      "name": "Themen"
    },
    "tools": {
      "name": "Werkzeuge",
      "children": {
        "gfi": {
          "name": "Informationen abfragen",
          "active": true
        },
        "coordToolkit": {
          "name": "Koordinaten"
        },
        "measure": {
          "name": "Strecke / Fläche messen"
        },
        "draw": {
          "name": "Zeichnen / Schreiben"
        },
        "fileImport": {
          "name": "Datei Import"
        },
        "saveSelection": {
          "name": "Auswahl speichern"
        }
      }
    },
    "legend": {
      "name": "Legende"
    },
    "info": {
      "name": "Informationen",

```

```

        "children": {
          "staticlinks": [
            {
              "name": "Masterportal",
              "url": "https://masterportal.org"
            }
          ]
        }
      },
      "controls": {
        "zoom": true,
        "orientation": {
          "zoomMode": "once"
        },
        "attributions": true,
        "mousePosition": true
      }
    }
  }
}

```

Themenconfig

Die Themenconfig definiert, welche Inhalte an welcher Stelle im Themenbaum vorkommen.

Die Struktur ist abhängig von der Art des Themenbaums (ob flache Hierarchie oder Aufsplittung in Fach- und Hintergrundkarten).

Zudem können mehrdimensionale Daten (Time und 3D) separat definiert werden.

Ein Minimal-Beispiel (`treetype: custom`):

```

{
  "Themenconfig": {
    "Hintergrundkarten": {
      "Layer": [
        {
          "id": "452"
        },
        {
          "id": "432"
        }
      ]
    },
    "Fachdaten": {
      "Ordner": [
        {
          "Titel": "Fahrrad",
          "Layer": [
            {
              "id": "10882"
            }
          ]
        }
      ]
    }
  }
}

```

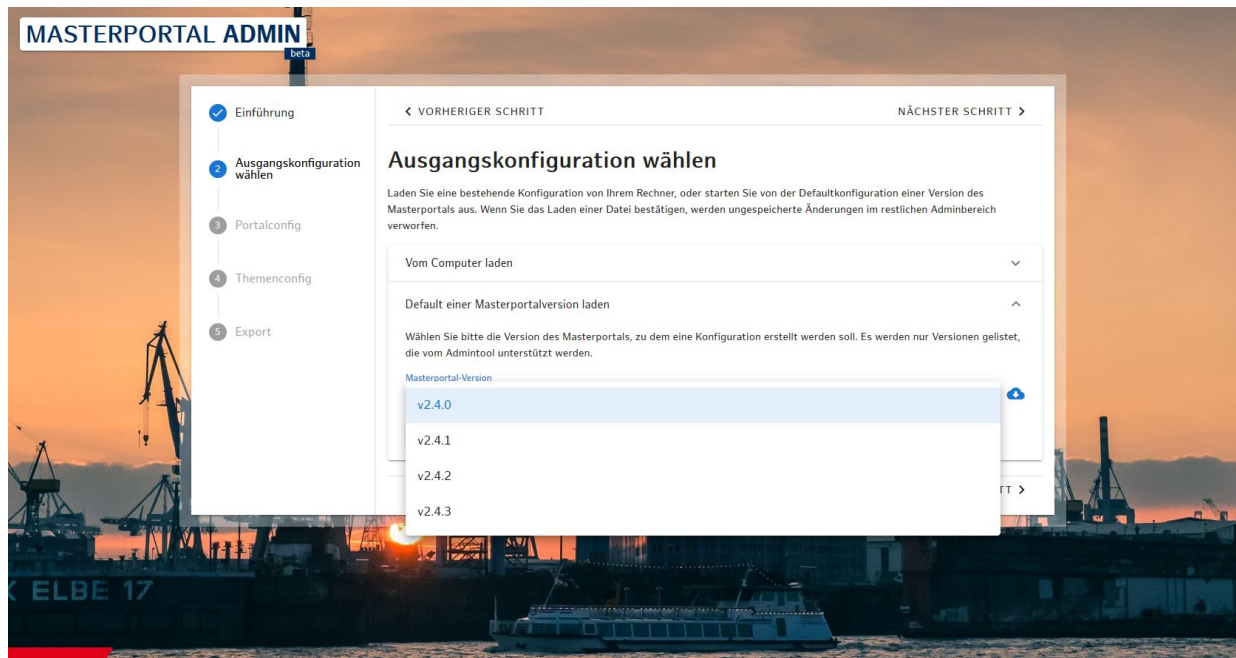
[Link zur vollständigen Dokumentation](#)

FOSSGIS 2024 Last modified: 2024-03-21 13:08:21

Masterportal Admin

Mit dem „Masterportal-Admin“ steht interessierten NutzerInnen eine webbasierte Anwendung mit grafischer Benutzeroberfläche zur Verfügung, die es ermöglicht, Konfigurationen für individuelle Geoportale dialoggeführt zu erstellen.

- [Kurzbeschreibung](#)
- [Repository](#)



MP Admin

FOSSGIS 2024Last modified: 2024-03-21 13:08:21

Dokumentation

Die Dokumentation für die Administration und Konfiguration des Masterportals ist in Markdown geschrieben (Ordner `doc`).

Markdown nach HTML (docs "bauen")

1. Im Masterportal repo `npm run buildMdDocs` aufrufen.
2. Die Markdowns werden dann in den Ordner `docHtml` geschrieben und können bequem in einem Webbrowser betrachtet werden.
3. `cd docHtml` und `python3 -m http.server` ausführen.
4. Nun werden die Docs unter <http://localhost:8000> ausgeliefert. (z.B. <http://localhost:8000/config.json.html>)


Alternativen

- <https://www.masterportal.org/dokumentation.html>
 - <https://bitbucket.org/geowerkstatt-hamburg/masterportal/src/latest/>
 - Oder mit einem Editor (z.B. Visual Studio Code) die Markdowns rendern (Strg+Shift+v)
- FOSSGIS 2024Last modified: 2024-03-21 13:08:21

Print

Im Masterportal können [MapFish Print](#) Servlets angebunden werden. MapFish Print ist eine etablierte JAVA-Software zum Erstellen von Map-Reports. Es werden Jasper Reports Templates unterstützt, hierzu gibt es einen GUI-Editor: [Link zur Community Edition](#).

Features

- Druck von Raster- und Vektorkarten
- Nordpfeil
- Maßstabsleiste
- Koordinatengitter
- Legendendruck
- Tabellendruck
- Grafiken in Bildern
- Hinzufügen freier Layoutelemente (Boxen, Schriftzüge etc.)
- ...  [Beispielkonfigurationen](#)

Installation des Print Servlets

- Download des WAR-Files, Serve via Tomcat oder Jetty
- Docker: [Offizielles Image mit vielen Konfigurationsparamtern](#)

Minimal Beispiel Mapfish Print Docker Service

Voraussetzung: Print Templates

Quickstart:

```
docker run --rm --name "mapfish-print" -p 8080:8080 -
v pwd /print_templates/default:/usr/local/tomcat/webapps/ROOT/print-apps/ camptocamp/mapfish_print:3.30
```

Erläuterung des Befehls:

- Es wird ein neuer Container mit den Namen `mapfish-print` (`--name`) auf Basis des Images `camptocamp/mapfish_print:3.30` gestartet.
- Der Port 8080 des Containers wird auf den lokalen Port 8080 gemappt (`-p`)
- Der Container wird beim Beenden automatisch entfernt (`--rm`).
- Es wird das Verzeichnis `print_templates/default` im aktuellen Pfad gemounted (`-v`). Dies wird im Container in den Pfad `/usr/local/tomcat/webapps/ROOT/print-apps` gemounted.

Service in einem docker compose File:

```
print:
  image: camptocamp/mapfish_print:3.30
  container_name: print
  volumes:
    - ./print_templates/default:/usr/local/tomcat/webapps/ROOT/print-apps:Z
  ports:
    - 8080:8080
```

Konfiguration MapFish Print

1. Ein MapFish Print Servlet kann mehrere Print Apps haben. In den meisten Fällen reicht eine Print App, diese wird standardmäßig `default` genannt.
2. In der Printapp befindet sich die Konfigurationsdatei `config.yaml`, diese kann auch eine JSON Datei sein.
3. In der `config.yaml` werden verschiedene Templates konfiguriert, z.B. für DIN A4 Querformat oder DIN A4 Hochformat:

```
templates:
  A4 Querformat: !template
    reportTemplate: a4_landscape.jrxml
    attributes:
      title: !string
        default: "Countries"
      map: !map
        maxDpi: 400
        width: 786
        height: 539
        zoomLevels: !zoomLevels
          scales: [250, 500, 1000, 2500, 5000, 10000, 25000, 50000, 100000]
      northArrowDef: !northArrow
        size: 25
        default:
          backgroundColor: "rgba(214, 214, 214, 0)"
      scalebar: !scalebar
        width: 195
        height: 20
        createSubReport: true
        default:
          backgroundColor: "rgba(214, 214, 214, 0)"
          align: "center"
          barBgColor: "rgba(255, 255, 255, 255)"
          fontSize: 7
          barSize: 3
          color: "rgba(0, 0, 0, 255)"
          geodetic: false
          intervals: 4
          lineWidth: 1
          unit: "m"
      showLegend: !boolean
        default: false
      legend: !jsonData {}
    processors:
      - !reportBuilder
        directory: "."
      - !createMap {}
      - !createNorthArrow
      inputMapper:
        northArrowDef: northArrow
      - !createScalebar {}
```

1. Der Key `reportTemplate` verweist auf ein konkretes Template, z.B. `a4_portrait.jrxml`. Diese liegt in diesem Fall im selben Ordner wie die `config.yaml`.
2. Darüber hinaus können hier Geodaten abgelegt werden, die für Testdrucke genutzt werden können (ausführbar auf der Startseite bzw. Landingpage des Druck-Servlets).
3. Schriftarten und weitere Assets, wie etwa Grafiken für Logos, können in Unterordnern oder auch direkt auf der Ebene der Templates abgelegt werden.

Definition eines Druck Templates

Die Templates sind in `jrxml` verfasst und können wie jedes XML mit jedem Texteditor editiert werden. Eine Software mit GUI zur Verwaltung und Bearbeitung der Templates kann als Community-Edition heruntergeladen werden: [Link zur Community Edition](#).

Aus Zeitgründen wird in diesem Workshop nicht genauer auf die Templates eingegangen.

🇩🇪 Meldet euch gern bei den Workshop-Leitern, wenn ihr mehr Infos oder generell mehr Input hierzu benötigt.

Konfiguration im Masterportal

rest-services.json

```
{
  "id": "print",
  "name": "Print Service Test",
  "url": "http://localhost:8080/print/",
  "typ": "Print"
}
```

config.json

```
"print": {
  "name": "translate#common:menu.tools.print",
  "printServiceId": "print",
  "printAppId": "default",
  "filename": "Süderbrarup Ausdruck ${dd-MM-yyyy-HH-ss}",
  "title": "Mein Titel",
  "onlyDesktop": false
}
```

Vollständige Konfigurationsparameter (`doc/config.json.md`):

| Name | Required | Type | Default | Description | Enforced |
|----------------------|----------|--------|---------------------|--|----------|
| mapfishServiceId | yes | String | | <i>Deprecated in 3.0.0.</i> Print service id. Resolved using the rest-services.json file. | fa |
| printServiceId | yes | String | | Print service id. Resolved using the rest-services.json file. | fa |
| printService | no | String | "mapfish" | Flag determining which print service is in use. <code>plotservice</code> activates the <i>High Resolution PlotService</i> , if the parameter is not set, <i>Mapfish 3</i> is used. | fa |
| printAppCapabilities | no | String | "capabilities.json" | path for the configuration of the print service | fa |
| currentLayoutName | no | String | "" | Defines which layout is the default layout on opening the print tool, e.g. "A4 portrait format". If the given layout is not available oder none is provided, the first layout mentioned in the Capabilities is used. | fa |
| printAppId | no | String | "master" | Print service print app id. This tells the print service which template(s) to use. | fa |
| filename | no | String | "report" | Print result file name. | fa |
| title | no | String | "PrintResult" | Document title appearing as header. | fa |
| titleLength | no | Number | 45 | The length of the title characters. | fa |

| Name | Required | Type | Default | Description | Enforced |
|---------------------------|----------|------------------------------------|---------------|--|----------|
| isLegendSelected | no | Boolean | false | Defines whether a checkbox to print the legend is offered. Only used for print services supporting legend printing (Mapfish Print 3). | false |
| legendText | no | String | "Mit Legende" | Descriptive text for the legend print checkbox. | false |
| dpiForPdf | no | Number | 200 | DPI resolution for the map in the PDF file. | false |
| capabilitiesFilter | no | capabilitiesFilter | | Filter for the response of the configured print service. Possible keys are layouts and outputFormats. | false |
| defaultCapabilitiesFilter | no | capabilitiesFilter | | If there is no key set in capabilitiesFilter, the key from this object is taken. | false |
| useProxy | no | Boolean | false | <i>Deprecated in the next major release. GDI-DE recommends setting CORS headers on the required services instead of using proxies.</i> Defines whether a service URL should be requested via proxy. For this, dots in the URL are replaced with underscores. | false |
| printMapMarker | no | Boolean | false | If set to true, map markers visible in the print image section will be printed. They may obstruct the view to interesting information. | false |

| Name | Required | Type | Default | Description | Editor |
|------------------------|----------|----------|---------|---|--------|
| overviewmapLayerId | no | String | | Allows using a different layer for the overview map element. If no Id is specified, the first layer of the selected baselayer maps is used. | false |
| layoutOrder | no | String[] | | Defines the order in which the layouts should be shown in the dropdown. Only to be used in combination with the print service 'plotservice'. | false |
| isPrintDrawnGeoms | no | Boolean | false | Flag if it's possible to print geometries created by the Draw or Measure tool. Only to be used in combination with the print service 'plotservice'. | false |
| showInvisibleLayerInfo | no | Boolean | true | Defines whether an infobox is shown when layers will not be printed because they are invisible due to scale. | false |


FOSSGIS 2024Last modified: 2024-03-21 13:08:21

Übungsaufgaben Druck

Aufgabe 1: Starten Sie nach Anleitung vorheriger Folie einen lokalen Mapfish Print Server als Docker Container. Mounten Sie eine vorbereitete Print-App von terrestris: https://github.com/terrestris/masterportal-ws/print_templates

Aufgabe 2: Starten Sie ihr Dev-Setup (wenn noch nicht geschehen) und nehmen Sie eine exemplarische Konfiguration für das Druck-Template vor.

Aufgabe 3: Editieren Sie das Druck-Template, so dass das Logo ihrer Institution / Firma angezeigt wird.

 **Bonus Aufgabe 4:** Konfigurieren Sie einen Filter, sodass nur ein einziges Layout für das Format PDF im Druck-Tool des Masterportals erscheint.

Hint

```
{
  "capabilitiesFilter": {
    "layouts": ["A4 Querformat"],
    "outputFormats": ["PDF"]
  }
}
```

FOSSGIS 2024Last modified: 2024-03-21 13:08:21

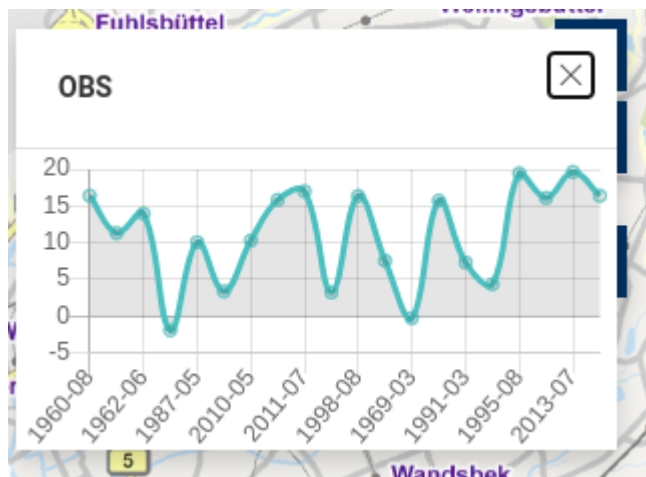
Addons

Mittels eines Addons lässt sich die Funktionalität des Masterportals beliebig erweitern, ohne dass der **Core** verändert werden muss. Es lassen sich eigenständige `Tools` und `GfiThemes` entwickeln, die zu Beginn der Laufzeit importiert werden und fortan wie eigenständige Module funktionieren.

Es empfiehlt sich, die [Vue.js devtools](#) zu installieren. Diese bieten u.a. einen Live-Zugriff auf den Store.

Im Rahmen dieses Workshops soll exemplarisch ein custom GFITheme Addon eingebunden und konfiguriert werden.

Addon-Code: <https://github.com/terrestris/masterportal-addons/tree/main/simpleLineChart>



FOSSGIS 2024Last modified: 2024-03-21 13:08:21

Übungsaufgaben Addon

Aufgabe 1: Integrieren Sie das `simpleLineChart` Addon in ihr Dev-Setup. Der erste Schritt ist der [Download des Addon-Codes](#). Dieser muss in den Ordner `addons` kopiert werden (`addons/simpleLineChart`). Starten Sie anschließend das Dev Setup neu, damit das Addon verfügbar ist.

Aufgabe 2: Definieren Sie das Addon in der `addonsConf.json` .

Hint

```
{
  "simpleLineChart": {
    "type": "gfiTheme"
  }
}
```

Aufgabe 3: Nehmen Sie die Konfiguration des Addons in der `config.js` vor.

Hint

```
{
  "simpleLineChart": {
    "type": "gfiTheme"
  }
}
```

Aufgabe 4: Konfigurieren Sie einen Layer, der das custom GFI Addon verwendet:

🚧 Für den DWD Geoserver muss zunächst ein Proxy eingerichtet werden:

- Erstellen sie eine `proxyconf.json` im Ordner `devsetup` mit folgenden Inhalt:

```
{
  "/dwd": {
    "target": "https://cdc.dwd.de/geoserver",
    "changeOrigin": true,
    "pathRewrite": {
      "^/dwd": ""
    }
  }
}
```

- Starten Sie den lokalen Dev-Server des Masterportals neu!
- Anschließend kann der folgende Layer definiert werden werden:

```

{
  "id": "obs_deu_p1m_t2m",
  "name": "OBS",
  "url": "https://localhost:9001/dwd/ows",
  "layers": "OBS_DEU_P1M_T2M",
  "typ": "WMS",
  "format": "image/png",
  "version": "1.3.0",
  "singleTile": false,
  "transparent": true,
  "transparency": 0,
  "tilesize": 256,
  "gutter": 0,
  "gfiAttributes": "showAll",
  "layerAttribution": "nicht vorhanden",
  "legendURL": "ignore",
  "gfiTheme": {
    "name": "simpleLineChart",
    "params": {
      "wfs": {
        "url": "https://localhost:9001/dwd/ows",
        "featureType": "CDC:OBS_DEU_P1M_T2M",
        "attributes": {
          "timestamp": "ZEITSTEMPEL",
          "value": "WERT"
        }
      }
    }
  }
},



```

Aufgabe 5: Testen Sie das Addon, in dem Sie den Layer in der `config.json` referenzieren, im Portal anschalten und einen GFI Request auslösen.

FOSSGIS 2024Last modified: 2024-03-21 13:08:21

Migration

Es wird ein Tool bereitgestellt, dass Konfigurationen von v2 nach v3 migriert.

  Der Status ist noch **Work in Progress**, es wird bis zum Release der finalen v3 Version noch angepasst.

1. Checken Sie den Tag v3.0.0-beta2 aus: `git checkout v3.0.0-beta2` .
2. Ändern Sie `node` Version auf `18.16.0` : `nvm use 18.16.0`
3. Installieren Sie die dependencies: `npm i`
4. Starten Sie das Kommandozeilen Tool mittels `npm run migrateConf` .
5. Geben Sie interaktiv den Source und Destination Path der Portalkonfiguration an.

Drawbacks: Es werden nicht alle Tools migriert, zum Teil weil diese in v3 nicht mehr verfügbar sind oder die noch nicht vom Migrationsskript abgedeckt werden

FOSSGIS 2024Last modified: 2024-03-21 13:08:21

Übungsaufgaben

! Startpunkt ist der `basic` Branch in der Entwicklungsumgebung:

Im Masterportal Folder (cloned Bitbucket-Repository) muss vorher ein `npm i` ausgeführt worden sein. Dann wird das Entwicklungssetup gestartet per `npm run start`. Das Starten dauert einige Zeit, es werden einige Warnings angezeigt, die aber getrost ignoriert werden können.

Sobald der Log `Compiled successfully` erscheint, kann losgelegt werden.

 Verwenden Sie `Featherpad` als Text-Editor für die folgenden Aufgaben.

Wir beginnen im nächsten Kapitel mit Aufgaben zur allgemeinen Konfiguration des Portals.

FOSSGIS 2024Last modified: 2024-03-21 13:08:21

Allgemeine Konfiguration

Aufg. 1: Ändern Sie das Logo der Anwendung in folgendes Bild um: `https://www.fossgis.de/logos/FOSSGIS@2x.png` . Ändern Sie den Titel des Portals zu `FOSSGIS Testportal` und wählen Sie geeignete Werte für `Link` und `Tooltip` .

Hint

Schauen Sie in der `config.json.md` nach `Portalconfig.portalTitle` !

Lösung

Definieren Sie in der `config.json` im Block `Portalconfig` den folgenden Block:

```
"portalTitle": {
  "title": "FOSSGIS Testportal",
  "logo": "https://www.fossgis.de/logos/FOSSGIS@2x.png",
  "link": "https://www.fossgis.de",
  "toolTip": "FOSSGIS Logo"
}
```

Aufg. 2: Ändern Sie den Titel der Webseite (taucht oben in der Browser Leiste auf).

Hint

Werfen Sie einen Blick in die `index.html` im Portalordner `basic` . Es gibt einen

Lösung

Passen Sie die Zeile 10 in der `index.html` an:

```
<title>FOSSGIS Portal</title> <!-- enter your own Portal Title for the website at this -->
```

Aufg. 3: Konfigurieren Sie die Definition für das projizierte Koordinatensystem UTM für die Zone 33N. Dies soll standardmäßig genutzt werden, wenn das Portal aufgerufen wird.

Hint

Werfen Sie einen Blick in die `config.js` im Portalordner `basic` . Was steht im Block `namedProjections` ?

Lösung

Wir benötigen die proj.4 Defintion von UTM 33N. Diese kann mittels QGIS oder hier herausgefunden werden: <https://epsg.io/25833>.

Anschließend folgendes Element im `Array` (Liste) `namedProjection` hinzufügen (Komma nach dem ersten Listenelement nicht vergessen!):

```
["EPSG:25833", "+proj=utm +zone=33 +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +units=m +no_defs"]
```

Dann muss in der `config.json` im Block `mapView` der Wert `epsg` auf `EPSG:25833` gesetzt werden.

Zuletzt müssen noch die Startkoordinate und der Extent angepasst werden:

```
"startCenter": [  
  389932,  
  5819658  
],  
"extent": [  
  369149,  
  5806271,  
  411602,  
  5838442  
],
```

🤖 Die Karte bleibt grau! Woran kann das liegen?

FOSSGIS 2024Last modified: 2024-03-21 13:08:21

Hinzufügen neuer Layer

Aufg. 4: Definieren Sie einen neuen WMS-Layer in der `services.json` und integrieren Sie diese der Anwendung.

- Basis-Url: `https://sgx.geodatenzentrum.de/wms_basemapde?`
- Er soll initial sichtbar sein
- Kachelgröße 512x512 Pixel
- Kein `GetFeatureInfo`
- Legende soll angezeigt werden

Hint

Schauen Sie sich eine existierende WMS-Konfiguration an. Zum Beispiel die ID 452 (Digitale Orthophotos (belaubt) Hamburg). Kopieren Sie diesen Block und ändern Sie die entsprechenden Parameter. Einige können zunächst ausgelassen werden, z.B. `dataset` (für Metadaten).

Ferner kann die Dokumentation der [services.json](#) hinzugezogen werden.

Lösung

Eintrag in der `services.json` :

```
{
  "id": "1000",
  "name": "basemap.de Web Raster bunt",
  "url": "https://sgx.geodatenzentrum.de/wms_basemapde",
  "typ": "WMS",
  "layers": "de_basemapde_web_raster_farbe",
  "format": "image/png",
  "version": "1.3.0",
  "singleTile": false,
  "transparent": true,
  "gfiAttributes": "ignore",
  "tilesize": "512",
  "layerAttribution": "@ basemap.de / BKG 2023",
  "legend": true
},
```

Eintrag in der `config.json`

```
"Themenconfig": {
  "Hintergrundkarten": {
    "Layer": [
      {
        "id": "1000",
        "visibility": true
      },
    ],
  },
}
```

Aufg. 5: Definieren Sie einen WMTS Layer mit der Basis-URL `https://tiles.geoservice.dlr.de/service/wmts?SERVICE=WMTS&REQUEST=GetCapabilities` . Integrieren Sie diesen in das Portal.

Hint


Werfen Sie einen erneuten Blick in die [services.json](#) Dokumentation. Es werden für den WMTS Layertyp zwei Beispiele genannt.

Lösung

Definieren Sie den Layer wie folgt:

```
{
  "id": "2020",
  "name": "EOC Basemap",
  "capabilitiesUrl": "https://tiles.geoservice.dlr.de/service/wmts?SERVICE=WMTS&REQUEST=GetCapabilities",
  "typ": "WMTS",
  "layers": "eoc:basemap",
  "optionsFromCapabilities": true
},
```

Fügen Sie anschließend die ID in die Themenkonfiguration ein (siehe Aufg. 2).

 Wie bereits erwähnt unterstützt das Masterportal weitere Layerstandards wie WFS, Sensor Things API, OGC Feature API etc. Sie können jederzeit Layerkonfigurationen aus anderen Portal anschauen (z.B.

<https://geoportal.de>). Dazu öffnen Sie vor dem Start des Portals die Dev-Tools (F12) und suchen im Netzwerktab nach `services.json` / `services-internet.json`.

Aufg. 6: Fügen Sie zunächst folgenden WFS layer hinzu:

```
{
  "id": "2000",
  "name": "ALKIS Flurstücke",
  "url": "https://fbinter.stadt-berlin.de/fb/wfs/data/senstadt/s_wfs_alkis",
  "typ": "WFS",
  "featureType": "s_wfs_alkis",
  "version": "2.0.0",
  "featureNS": "fis",
  "layerAttribution": "nicht vorhanden",
  "legend": true,
  "gfiAttributes": {
    "zae": "Zähler Flurstücksnummer",
    "nen": "Nenner Flurstücksnummer",
    "fln": "Flurnummer",
    "afl": "Amtliche Fläche",
    "gmk": "Gemarkung",
    "bezeich": "Bezeichnung",
    "fsko": "fsko",
    "namgen": "Gemeinde",
    "statusdat": "Status",
    "gdz": "Gemeindezugehörigkeit"
  },
  "minScale": "500",
  "maxScale": "5000"
},
```

Es sollen verknüpfte Metadaten angezeigt werden.

Nehmen Sie folgenden CSW: <https://fbinter.stadt-berlin.de/fb/csw> und folgende Metadaten ID: 8519978f-6fa3-39e1-9616-4e4cfb340b09. Für den Link zum Metadatenkatalog wählen sie folgende Basis-URL:

<https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/>.

Lösung

Definieren Sie den Layer wie folgt:

```
{
  "datasets": [
    {
      "csw_url": "https://fbinter.stadt-berlin.de/fb/csw",
      "md_id": "8519978f-6fa3-39e1-9616-4e4cfb340b09",
      "show_doc_url": "https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/"
    }
  ],
},
```

Fügen Sie anschließend die ID in die Themenkonfiguration ein (siehe Aufg. 2).

Fügen Sie den Layer den Fachdaten hinzu (`config.json`).

FOSSGIS 2024Last modified: 2024-03-21 13:08:21

Konfiguration von Werkzeugen

Aufg. 7: Konfigurieren Sie das Messen-Tool so, dass für das Messen der Fläche stets auch Dezimeter angezeigt werden. Zudem sollen nautische Seemeilen auswählbar sein.

Hint

Schauen Sie in der `config.json.md` nach `Portalconfig.menu.tool.measure` !

Lösung

Ändern Sie die Konfiguration für das Messen-Werkzeug in der `config.json` wie folgt:

```
"measure": {
  "name": "translate#common:menu.tools.measure",
  "measurementAccuracy": "decimeter",
  "lineStringUnits": ["m", "km", "nm"]
}
```

Info: Wenn der Name mit einem `translate#` beginnt, folgt da hinter der Key in den Übersetzungsdateien `i18n`. In mehrsprachigen Modulen sollte dies immer der Fall sein, da ansonsten der Name des Tools im Portal nicht übersetzt wird.

Aufg. 8: Fügen Sie die Map-Control `overviewMap` dem Portal hinzu. Dies soll initial eingeblendet sein. Wählen Sie den Layer aus Aufgabe 1 für die Overview Map aus.

Hint

Schauen Sie in der `config.json.md` nach `Portalconfig.controls.overviewMap` !

Lösung

Ändern Sie die Konfiguration für die `overviewMap` Control folgendermaßen:

```
"overviewMap": {
  "layerId": "1000",
  "isInitOpen": true
}
```

Aufg. 9: Konfigurieren Sie das Messen-Tool so, dass für das Messen der Fläche stets auch Dezimeter angezeigt werden. Zudem sollen nautische Seemeilen auswählbar sein.

Hint

Schauen Sie in der `config.json.md` nach `Portalconfig.menu.tool.measure` !

Lösung

Ändern Sie die Konfiguration für das Messen-Werkzeug in der `config.json` wie folgt:

```
json "measure": { "name": "translate#common:menu.tools.measure", "measurementAccuracy": "decimeter",
"lineStringUnits": ["m", "km", "nm"] }
```

FOSSGIS 2024Last modified: 2024-03-21 13:08:21

Konfiguration der Suche

Aufgabe 10: Es soll eine Konfiguration für ein Gazeetteer erstellt werden. Hierfür nutzen wir die [Komoot API](#), die einen schnellen Zugriff auf OSM Daten ermöglicht.

Definieren Sie zunächst folgenden Block in der `rest-services-internet.json`.

```
{
  "id" : "1",
  "name" : "Komoot Photon Suche",
  "url" : "https://photon.komoot.io/api/?",
  "typ" : "WFS"
},
```

Anschließend muss in der `config.json` im Block `searchbar` folgender Block definiert werden:

```
"searchBar": {
  "komoot": {
    "minChars": 3,
    "serviceId": "1",
    "limit": 20,
    "lang": "de",
    "lat": 52.3879,
    "lon": 13.0582,
    "bbox": "13.13, 52.27, 13.64, 52.70"
  },
  "startZoomLevel": 9,
  "placeholder": "Suche nach: - Adressen"
},
```

Es sollen nur Straßen durchsucht werden, fügen Sie einen entsprechenden Parameter hinzu.

Hint

Schauen Sie in der `config.json.md` im Abschnitt `Portalconfig.searchBar.komoot` nach Filterung für OSM Tags.

Weiter Infos: <https://github.com/komoot/photon>.

Aufgabe 11:

Der eingebundene WFS soll durchsuchbar sein. Hierfür soll das Attribut `zae` verwendet werden.

Hint

In der Layer-Konfiguration muss der Parameter `searchField` hinzugefügt werden.

Falls nicht bereits vorhanden, muss noch in der `searchConfig` `visibleVector` ergänzt werden.

⚠ `visibleWFS` ist deprecated, ab dem nächsten Major Update wird nur noch `visibleVector` unterstützt.

FOSSGIS 2024Last modified: 2024-03-21 13:08:21

Masterportal Production-Build erstellen

Aufg. 12: Erstellen Sie einen `Production-Build` .

```
npm run build
```

Pfad zum Ordner mit Portalen ausgehend von "[...]/masterportal/": Enter drücken.

Was passiert hier?

- JS-Code wird transpiliert
- Code wird paketiert
 - Optimierung für Nutzung im Browser
 - einfache Verteilung
- `bundle` des Quell-Codes und der `assets` (z.B. Grafiken, CSS, Übersetzungsdateien)

Schauen Sie danach on den Ordner `dist/masterportalCode/2_30_0` .

Hier landet der fertige Production-Build.

Wozu brauche ich das?

Man kann den gebauten Code auch hier herunter laden: <https://bitbucket.org/geowerkstatt-hamburg/masterportal/downloads/>.

Wenn Sie das Masterportal in einem Projekt einsetzen, das zum einen individuelle Erweiterungen hat und in dem Pipelines eingesetzt werden, kann es Vorteile haben, den `build` selbst zu erstellen. Beispielsweise können somit `Addons` hinzugefügt werden.

FOSSGIS 2024Last modified: 2024-03-21 13:08:21

Erweiterte Konfiguration

1. UseCase

Ich möchte die Hauptfarben des Portals anpassen! Die Buttons sollen nicht rot, sondern blau sein - da dies besser zu unserem StyleGuide passt. 📖

Überlegung:

Es gibt im Ordner `mastercode/2_30_0/css` eine `masterportal.css`. Kann ich nicht einfach hier den Farbwert überschreiben? ?

⚠️ Woher weiß man, welche CSS-Properties alle angepasst werden? Die css Datei > 2.500 Zeilen und > 300.000 Zeichen!

😞 Keine gute Idee! Bei einem Update kann sich die Datei ändern - und man muss alle Anpassungen überprüfen / erneut durchführen..

Daher:

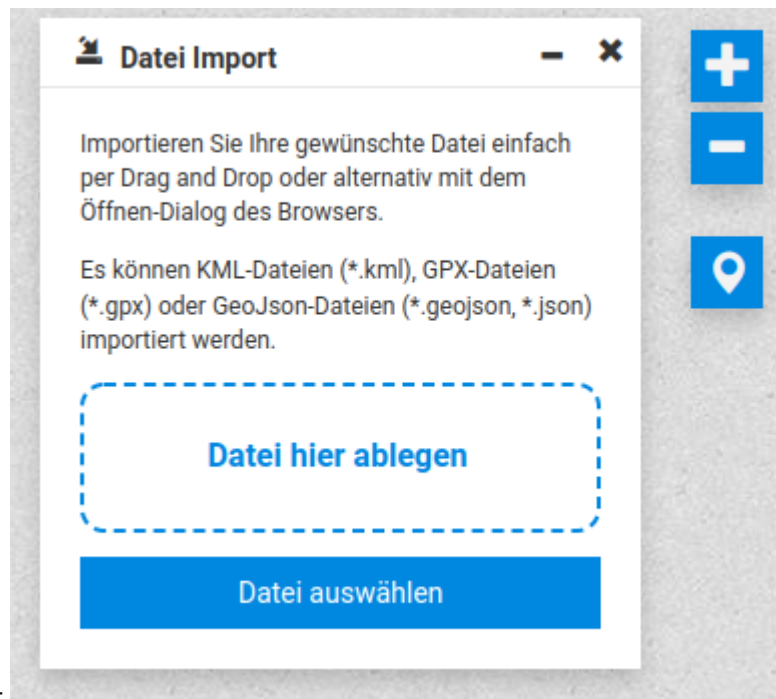
Wir passen eine zentrale Farbvariable an im Development Modus und bauen hinterher unser Portal neu! [TOP](#)

1. Öffnen Sie im Code-Editor die Datei `variables.scss`. Hier befinden sich sämtliche Farbwerte, aus denen die einzelnen Oberflächenelemente referenziert werden (Buttons, Hintergründe von Schaltflächen etc.).
2. Suchen sie nach dem Key `$primary` (Tipp: Zeile 94 🧐).
3. Passen Sie den Wert an. Zum Beispiel auf Hellblau: `#0087e0`.
4. Webpack erkennt, dass Sie eine Änderung vorgenommen haben und kompiliert den Code erneut. Das Ergebnis ist sofort sichtbar unter `localhost:9001/portal/basic`.

😞 Jetzt sind die Map-Controls blau, aber was ist mit den Tool-Buttons? Ich meine explizit das Tool Datei-Import.

5. Passen Sie die folgenden Variablen an:

```
$secondary_focus: #0087e0;
$accent: #0087e0;
$accent_hover: darken($accent, 5%);
```



6. Betrachten Sie das Ergebnis:
der Footer noch blassgrau. Ich möchte diesen gerne ebenfalls blau haben.

Nun ist

⚠ Warning Ändern Sie nicht zu viele Farbwerte. Die EntwicklerInnen des Masterportals und des Oberflächen-Frameworks (Bootstrap) haben sich bei der Wahl der Farben viele Gedanken gemacht. Es werden auch extra barrierearme color schemes verwendet, dies sollte bedacht werden. Es sollte auch immer betrachtet werden, welche Farbwerte voneinander abhängen!
Goldene Regel: Nach jeder Änderung, ausführlich das Portal testen!

2. UseCase

Der MapMarker beim GFI ist schön und hat einen hohen Wiedererkennungswert. Trotzdem brauche ich für ein Spezial-Portal einen Custom-Marker Style. 🙌 Was kann ich tun?

- Schauen Sie in der `config.js.md` nach dem Stichwort `mapMarker`. Sie finden dort folgende Erläuterungen:

| Name | Required | Type | Default | Description |
|----------------|----------|--------|---------------------------|--|
| pointStyleId | no | String | "defaultMapMarkerPoint" | StyleId to refer to a <code>style.json</code> point style. If not set, the <code>img/mapMarker.svg</code> is used. |
| polygonStyleId | no | String | "defaultMapMarkerPolygon" | StyleId to refer to a <code>style.json</code> polygon style. |

Example:

```
{
  "mapMarker": {
    "pointStyleId": "customMapMarkerPoint",
    "polygonStyleId": "customMapMarkerPolygon"
  }
}
```

Aha! 🙌

- Definieren Sie in der `style.json` einen Punkt Stil mit 3 Farbigen Kreisen:

```
{
  "styleId": "custom-point",
  "rules": [
    {
      "style": {
        "circleRadius": 10,
        "circleFillColor": [0, 132, 255, 1],
        "circleStrokeColor": [255, 102, 0, 1],
        "circleStrokeWidth": 5
      }
    }
  ]
}
```

2. Weisen Sie in der `config.js` dem Property `pointStyleId` den neuen Stil `custom-point` zu.
3. Hier muss das Dev-Setup ggf. neugestartet werden (Strg+C, dann erneut `npm run start`), um den Effekt zu sehen.

FOSSGIS 2024Last modified: 2024-03-21 13:08:21