

# Decoupled Search with React.js

Natalia Justicia



# Hi, I am **Natalia**

Frontend Developer by **Netnode AG**

@nataliajustice - [drupal.org](https://drupal.org) / [drupal.slack.com](https://drupal.slack.com)





## Context



## Drupal Module



## Configuring the search



## Search Api export



## Building the search



## Conclusions



## Questions



# Context

patientendossier.ch

patientendossier.ch

Guest

EPD-Blog Publikationen Über uns Kontakt DE

Das EPD Informationen EPD eröffnen

EPD

elektronisches Patientendossier

# Meine Gesundheitsinfos, online verfügbar

Verwalten Sie Ihre eigenen Gesundheitsinfos und teilen Sie diese mit Ihren Gesundheitsfachpersonen.

So nutzen Sie das EPD  
(2:34)

Hausarzt

Spezialärztin

Apotheke

Spital

Physiotherapeut

Pflegeheim und weitere

patientendossier.ch

patientendossier.ch

Guest

EPD-Blog    Publikationen    Über uns    Kontakt    DE

Das EPD    Informationen    EPD eröffnen    X

Search...

MEINE GESELLSCHAFTSINFO,  
online verfügbar

Verwalten Sie Ihre eigenen Gesundheitsinfos und  
teilen Sie diese mit Ihren Gesundheitsfachpersonen.

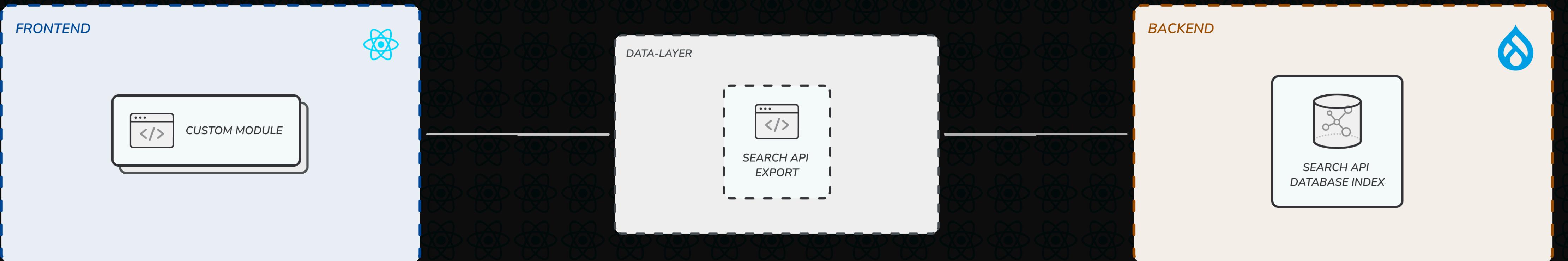
Hausarzt    Spezialärztin

Apotheke    Spital

Physiotherapeut    Pflegeheim und weitere

So nutzen Sie das EPD (2:34)

The screenshot shows the homepage of the patientendossier.ch website. At the top, there's a navigation bar with links for 'EPD-Blog', 'Publikationen', 'Über uns', 'Kontakt', and a language switcher 'DE'. Below the navigation is a logo for 'EPD elektronisches Patientendossier' featuring a blue circle with a white cross. A search bar with a magnifying glass icon is positioned above a large central banner. The banner has a blue gradient background and contains the text 'MEINE GESELLSCHAFTSINFO, online verfügbar' in large white letters. Below this, a subtext reads 'Verwalten Sie Ihre eigenen Gesundheitsinfos und teilen Sie diese mit Ihren Gesundheitsfachpersonen.' To the right of the text are several circular icons representing different healthcare professionals and facilities: 'Hausarzt' (male doctor), 'Spezialärztin' (female doctor), 'Apotheke' (pharmacy), 'Spital' (hospital), 'Physiotherapeut' (physiotherapist), and 'Pflegeheim und weitere' (nursing home and others). In the bottom left corner of the banner, there's a video thumbnail with the text 'So nutzen Sie das EPD (2:34)' and a play button icon. The overall design is clean and modern, using a white background for the main content area.





# Drupal Module

# Search Api

## search\_api

## Search API

[View](#) [Version control](#) [View history](#) [Automated testing](#)

### What does it do?

This module provides a framework for easily creating searches on any entity known to Drupal, using any kind of search engine. For site administrators, it is a great alternative to other search solutions, since it already incorporates facetting support and the ability to use the [Views module](#) for displaying search results, filters, etc. Also, with the [Apache Solr integration](#), a high-performance search engine is available for this module.

Developers, on the other hand, will be impressed by the large flexibility and numerous ways of extension the module provides. Hence, the growing number of [additional contrib modules](#), providing additional functionality or helping users customize some aspects of the search process.

### Examples & Demo

The prime example of a site using this module is [drupal.org](https://drupal.org) itself – the Search API, along with the [database backend](#) and [Views](#), is used to generate the issue queue listings.

We also showcase this ecosystem of Search API, including sliders, sorting, autocomplete and more at <https://drupalsearch.ch>. It is allowed to login (username/password: drupalsearch) and check out the configuration of the site to see some of the best practice configurations. The code for this demo site is available at <https://github.com/nickveenhof/drupal8-umami-search>.

### Note on security

In general, the Search API doesn't provide any kind of access restrictions (as this cannot be done on a generic level). It is therefore your responsibility to take care that only accessible items are indexed or results displayed (e.g., by using appropriate filters).

There is, however, built-in support for node access checks via the [Node access](#) data alteration (q.v. for details). Also, the *Search views* module contains an option to do additional access checks on all entities in search views.

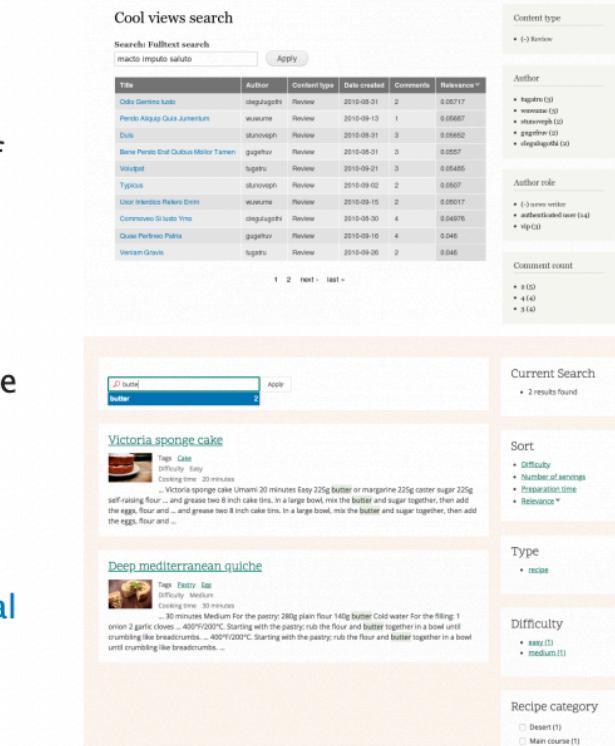
### Additional modules

At the moment, this project contains, apart from the core API module, the following extension modules:

#### Search views

(Drupal 7 only)

This module integrates the Search API with the [Views module](#), allowing searches on any index to be created and viewed via Views. All of an entity's properties, as well as those of related entities (e.g. a node's author's name), are available as fields, filters and arguments for all indexed fields



The screenshot shows a search interface titled 'Cool views search'. At the top, there are tabs for 'Search results', 'Search facets', and 'Search filters'. Below the tabs is a table with columns: 'Title', 'Author', 'Content type', 'Status', 'Created', and 'Updated'. The table lists several nodes, such as 'Ode to Gemini', 'Perle Alouette Quince Jam', 'Dish', 'Lemon Pepper Beet-Cauliflower Mashed Potatoes', 'Tzatziki', 'User interests (0)', 'Cinnamon', 'Cinnamon (0)', and 'Vanilla-Graham'. To the right of the table, there are sections for 'Content type', 'Author', 'Author role', 'Comment count', 'Current Search', 'Sort', 'Difficulty', and 'Recipe category'. The 'Content type' section includes links for 'Page', 'Image', 'View', 'Media', 'File', and 'Attachment'. The 'Author' section includes links for 'Anonymous', 'Moderator', 'Administrator', and 'Normal'. The 'Author role' section includes links for '1 or more website', '1 or more community event', and 'Normal'. The 'Comment count' section includes links for '0', '1 or 2', and '3 or 4'. The 'Current Search' section shows '2 results found'. The 'Sort' section includes links for 'Difficulty', 'Number of reviews', and 'Relevance'. The 'Difficulty' section includes links for 'Easy (1)', 'Medium (1)', and 'Hard (1)'. The 'Recipe category' section includes links for 'Dessert (1)' and 'Main course (1)'. At the bottom, there are navigation links for '1', '2', 'Next', and 'Last'.

 Star 177 

### Maintainers



borisson\_

dimitri\_m

### Issues for Search API

To avoid duplicates, please make sure you're not submitting a new issue if one exists.

#### Advanced search

All issues

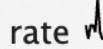
476 open, 6 RTBC, 10 closed

Bug report

199 open, 5 RTBC, 10 closed

Statistics

New issues 

Response rate 

1st response 

Open bugs 

Participants 

2 year graph, updated: 2022-07-12

### Resources

[Home page](#)

[Read license](#)

[Read complete log of changes](#)

[Try out a demonstration](#)

[http://drupal.org/project/search\\_api](http://drupal.org/project/search_api)

[Documentation for the latest version](#)

[Documentation for previous versions](#)

# Database Search

## search\_api\_db

## Search API

[View](#) [Version control](#) [View history](#) [Automated testing](#)

### What does it do?

This module provides a framework for easily creating searches on any entity known to Drupal, using any kind of search engine. For site administrators, it is a great alternative to other search solutions, since it already incorporates facetting support and the ability to use the [Views module](#) for displaying search results, filters, etc. Also, with the [Apache Solr integration](#), a high-performance search engine is available for this module.

Developers, on the other hand, will be impressed by the large flexibility and numerous ways of extension the module provides. Hence, the growing number of [additional contrib modules](#), providing additional functionality or helping users customize some aspects of the search process.

### Examples & Demo

The prime example of a site using this module is [drupal.org](https://drupal.org) itself – the Search API, along with the [database backend](#) and [Views](#), is used to generate the issue queue listings.

We also showcase this ecosystem of Search API, including sliders, sorting, autocomplete and more at <https://drupalsearch.ch>. It is allowed to login (username/password: drupalsearch) and check out the configuration of the site to see some of the best practice configurations. The code for this demo site is available at <https://github.com/nickveenhof/drupal8-umami-search>.

### Note on security

In general, the Search API doesn't provide any kind of access restrictions (as this cannot be done on a generic level). It is therefore your responsibility to take care that only accessible items are indexed or results displayed (e.g., by using appropriate filters).

There is, however, built-in support for node access checks via the [Node access](#) data alteration (q.v. for details). Also, the *Search views* module contains an option to do additional access checks on all entities in search views.

### Additional modules

At the moment, this project contains, apart from the core API module, the following extension modules:

#### Search views

(Drupal 7 only)

This module integrates the Search API with the [Views module](#), allowing searches on any index to be created and viewed via Views. All of an entity's properties, as well as those of related entities (e.g. a node's author's name), are available as fields, filters and arguments for all indexed fields

A screenshot of the drupal.org search interface. The search bar at the top has 'matt' typed into it. Below the search bar, there are several filter options: 'Content type' (set to 'Review'), 'Author' (list of authors including 'matt'), 'Author role' (list of roles including 'Moderator'), and 'Comment count' (list of counts including '0-10'). The main results table shows a list of reviews, each with columns for 'Title', 'Author', 'Content type', 'Created on', 'Comments', and 'Reactions'. One result is highlighted: 'Dude Gemini - Matt' by 'matt' on 2015-09-13. The page navigation shows '1 2 next - last'.

A screenshot of the drupal.org search interface. The search bar at the top has 'Victoria sponge cake' typed into it. Below the search bar, there are filter options: 'Sort' (set to 'Difficulty'), 'Type' (set to 'Issue'), and 'Difficulty' (set to 'Easy'). The main results table shows a list of reviews, each with columns for 'Title', 'Author', 'Content type', 'Created on', 'Comments', and 'Reactions'. Two results are visible: 'Victoria sponge cake' by 'matt' on 2015-09-13 and 'Deep mediterranean quiche' by 'matt' on 2015-09-13. The page navigation shows '1 2 next - last'.

Star 177

### Maintainers



borisson\_

d...

### Issues for Search API

To avoid duplicates, please make sure you're not submitting a new issue if one exists.

#### Advanced search

All issues

476 open, 6 RTBC, 1 closed

Bug report

199 open, 5 RTBC, 1 closed

Statistics

New issues

Response rate

1st response

Open bugs

Participants

2 year graph, updated

### Resources

[Home page](#)

[Read license](#)

[Read complete log of changes](#)

[Try out a demonstration](#)

[http://drupal.org/project/search\\_api\\_db](http://drupal.org/project/search_api_db)

[Documentation for the Search API](#)

[Documentation for the Search views module](#)

# Restful Web Services

## rest

## RESTful Web Services API overview

 Edit



Last updated on 28 September 2020

The RESTful Web Services API is new in Drupal 8.

For each REST resource, you can specify the supported verbs, and for each verb, you can specify the [serialization formats](#) & [authentication mechanisms](#).

This page is documenting capabilities of the REST module's API, about how to configure REST resource plugins, as well as how to create your own.

For information on how to actually *consume* the REST API from an application, see "[1. Getting started: REST configuration & REST request fundamentals](#)" — it is the first page of a complete how-to!

### # API features

Ordered by most to least frequently used APIs:

#### Configuring REST resources

##### # Specific version(s)

Before Drupal 8.2, this used to live in `rest.settings.yml`. The configuration in there is migrated automatically when you update to Drupal 8.2 or later. For details, see the change record: "[REST config converted to config entities](#)".



# Configuring the search

## <Backend />

# Creating the search api server

The image displays two screenshots of the Drupal administrative interface, specifically the Search API configuration page.

**Screenshot 1: Search API Overview**

This screenshot shows the main configuration page for the Search API. It lists two entries:

| Type   | Name            | Status  | Operations           |
|--------|-----------------|---------|----------------------|
| Server | Database Server | Enabled | <a href="#">Edit</a> |
| Index  | Content Index   | Enabled | <a href="#">Edit</a> |

At the top right, there are two buttons: **+ Add server** and **+ Add index**.

**Screenshot 2: Edit search server Database Server**

This screenshot shows the detailed configuration for the "Database Server" entry. The "Edit" tab is selected. The form fields are as follows:

- Server name\***: Database Server (Machine name: database\_server)
- Enabled**: A toggle switch is set to **Enabled**.
- Description**: A large text area for entering a description of the server.
- Backend\***: A radio button is selected for **Database**. A note below states: "Indexes items in the database. Supports several advanced features, but should not be used for large sites."
- Configure Database backend**: A collapsed section showing the current configuration: "Database" under "default > default".

# Creating the search api index

The screenshot shows two browser windows demonstrating the configuration of a search index in Drupal 8.

**Left Window: Search API Index List**

The URL is <https://demo-search.ch.indo.site/en/admin/config/search/search-api>. The page displays a table of indexes grouped by server. A 'Database Server' contains one 'Content Index'. Two buttons at the top right are visible: '+ Add server' and '+ Add index'.

| Type   | Name            | Status  | Operations           |
|--------|-----------------|---------|----------------------|
| Server | Database Server | Enabled | <a href="#">Edit</a> |
| Index  | Content Index   | Enabled | <a href="#">Edit</a> |

**Right Window: Edit search index Content Index**

The URL is [https://demo-search.ch.indo.site/en/admin/config/search/search-api/index/content\\_index/edit?destination=en/admin/config/search/search-api](https://demo-search.ch.indo.site/en/admin/config/search/search-api/index/content_index/edit?destination=en/admin/config/search/search-api). This is a modal dialog for editing the 'Content Index'.

**Form Fields:**

- Index name \***: Content Index (Machine name: content\_index)
- Data sources \***:
  - Contact message
  - Content
  - Content moderation state
  - Custom block
  - Custom menu link
  - File
  - Media
  - Search task
  - Shortcut link
- Configure the Content datasource**
- Configure the Default tracker**
- Server**:
  - No server
  - Database Server
- Enabled**: Only enabled indexes can be used for indexing and searching. This setting will only take effect if the selected server is also enabled.

# Adding the search api fields

The screenshot shows the 'Manage fields for search index Content Index' page in the Drupal admin interface. The page title is 'Content Index'. The 'Fields' tab is selected. There are two main sections: 'General' and 'Content'. The 'General' section contains two fields: 'Rendered HTML output' (Machine name: rendered\_item, Property path: rendered\_item, Type: Fulltext, Boost: 1.00) and 'URI' (Machine name: url, Property path: search\_api\_url, Type: String). The 'Content' section contains four fields: 'Content type' (Machine name: type, Property path: type, Type: String), 'ID' (Machine name: nid, Property path: nid, Type: Integer), 'Language' (Machine name: langcode, Property path: langcode, Type: String), and 'Title' (Machine name: title, Property path: title, Type: Fulltext, Boost: 1.00). A 'Data types' section is also present. At the bottom left is a 'Save changes' button.

| Label                | Machine name  | Property path  | Type     | Boost | Operations  |
|----------------------|---------------|----------------|----------|-------|-------------|
| Rendered HTML output | rendered_item | rendered_item  | Fulltext | 1.00  | Edit Remove |
| URI                  | url           | search_api_url | String   |       | Edit Remove |

| Label        | Machine name | Property path | Type     | Boost | Operations |
|--------------|--------------|---------------|----------|-------|------------|
| Content type | type         | type          | String   |       | Remove     |
| ID           | nid          | nid           | Integer  |       | Remove     |
| Language     | langcode     | langcode      | String   |       | Remove     |
| Title        | title        | title         | Fulltext | 1.00  | Remove     |

# Configuring the search api processors

The screenshot shows a web browser window titled "Manage processors for search". The URL is "demo-search.ch Indo.site/en/admin/config/search-api/index/content\_index/processors". The page header includes "Manage", "Shortcuts", and a warning message: "This site is intended for demonstration purposes." The user is logged in as "nadmin". The breadcrumb navigation shows: Home / Administration / Configuration / Search and metadata / Search API / Content Index.

## Manage processors for search index *Content Index* ☆

View Edit Fields **Processors**

Configure processors which will pre- and post-process data at index and search time. Find more information on the [processors documentation page](#).

**Enabled**

- Content access  
Adds content access checks for nodes and comments.
- Entity status  
Exclude inactive users and unpublished entities (which have a "Published" state) from being indexed.
- Highlight  
Adds a highlighted excerpt to results and highlights returned fields.
- HTML filter  
Strips HTML tags from fulltext fields and decodes HTML entities. Use this processor when indexing HTML data – for example, node bodies for certain text formats. The processor also allows to boost (or ignore) the contents of specific elements.
- Ignore case  
Makes searches case-insensitive on selected fields.
- Ignore characters  
Configure types of characters which should be ignored for searches.
- Number field-based boosting  
Adds a boost to indexed items based on the value of a numeric field.
- Reverse entity references

# Time to index

The screenshot shows a web browser window with the URL [https://demo-search.ch.Indo.site/en/admin/config/search/search-api/index/content\\_index](https://demo-search.ch.Indo.site/en/admin/config/search/search-api/index/content_index). The page title is "Content Index | Umami Food". The top navigation bar includes "Manage", "Shortcuts", and a warning message: "⚠ This site is intended for demonstration purposes." The user is logged in as "nadmin". The main content area is titled "Content Index ☆". It features tabs for "View" (selected), "Edit", "Fields", and "Processors". Below the tabs is a section titled "Index status" with a progress bar showing "38/38 indexed" and "100%". A table provides detailed configuration settings:

|                     |   |
|---------------------|---|
| Status              | enabled ( <a href="#">disable</a> )   |
| Datasource          | Content (38/38 indexed)   |
| Tracker             | Default   |
| Server              | Database Server   |
| Server index status | There are 38 items indexed on the server for this index. ( <a href="#">More information</a> ) |
| Cron batch size     | During cron runs, 50 items will be indexed per batch.   |

At the bottom of the page is a button labeled "▼ Start indexing now".



# Search Api export

## <DataLayer />

# Creating the search api export

The screenshot shows the Drupal admin interface for creating a 'Search API (Index Content Index)' view. The page title is 'Search API (Index Content Index)'. The 'Edit' tab is selected. The 'Displays' section contains a 'REST export' display. The 'Path settings' section shows the path as '/rest/api/search'. The 'Advanced' section includes a 'View REST export' button. The 'Fields' section lists various content fields like 'Content datasource: ID (indexed field)', 'Content datasource: Title (indexed field)', etc. The 'Filter criteria' section includes 'Index Content Index: Item language (= Interface text language selected for page)' and 'Search: Fulltext search (and)'. The 'Sort criteria' section includes 'Search: Relevance (desc)'. At the bottom are 'Save' and 'Cancel' buttons.

# Creating the search api export

The screenshot shows a Drupal administrative interface for configuring a search filter criterion. The main title is "Configure filter criterion: Search: Fulltext search".

**Fields**: A sidebar on the left lists various fields and filters, including "Content data", "Search: URI", "Content data", "Index Cont...", "Search: Full...", "Sort criter...", and "Search: Rel...".

**Multiple words**: A dropdown menu set to "Multiple words". A note below says: "Choose how the search keys will be parsed."

**Filter identifier**: A text input field containing "query". A note below says: "This will appear in the URL after the ? to identify this filter. Cannot be blank. Only letters, digits and the dot ("."), hyphen ("‐"), underscore ("\_"), and tilde ("~") characters are allowed."

**Placeholder**: An empty text input field with a note: "Hint text that appears inside the field when empty."

**Expose searched fields**: A toggle switch that is off. A note below says: "Expose the list of searched fields. This allows users to narrow the search to the desired fields."

**Multiple words**: A section describing the parsing of search queries.

**Searched fields**: A list of selected fields: "Rendered HTML output" and "Content » Title". A note below says: "Select the fields that will be searched. If no fields are selected, all available fulltext fields will be searched."

**Buttons**: At the bottom are three buttons: "Apply" (blue), "Cancel" (blue), and "Remove" (red).

```
demo-search.ch.lndo.site/rest/+  
Not Secure | demo-search.ch.lndo.site/rest/api/search  
  
1 // 20220623141701  
2 // http://demo-search.ch.lndo.site/rest/api/search  
3  
4 [  
5 {  
6   "nid": "10",  
7   "typeId": "recipe",  
8   "title": "Borscht with pork ribs",  
9   "type": "Recipe",  
10  "langcode": "en",  
11  "url": "http://demo-search.ch.lndo.site/en/recipes/borscht-with-pork-ribs",  
12  "imageUrl": "/sites/default/files/borscht-with-pork-ribs-umami.jpg"  
13 },  
14 {  
15   "nid": "11",  
16   "typeId": "article",  
17   "title": "Give it a go and grow your own herbs",  
18   "type": "Article",  
19   "langcode": "en",  
20   "url": "http://demo-search.ch.lndo.site/en/articles/give-it-a-go-and-grow-your-own-herbs",  
21   "imageUrl": "/sites/default/files/home-grown-herbs.jpg"  
22 },  
23 {  
24   "nid": "12",  
25   "typeId": "article",  
26   "title": "Dairy-free and delicious milk chocolate",  
27   "type": "Article",  
28   "langcode": "en",  
29   "url": "http://demo-search.ch.lndo.site/en/articles/dairy-free-and-delicious-milk-chocolate",  
30   "imageUrl": "/sites/default/files/vegan-chocolate.jpg"  
31 },  
32 {  
33   "nid": "13",  
34   "typeId": "article",  
35   "title": "The real deal for supermarket savvy shopping",  
36   "type": "Article",  
37   "langcode": "en",  
38   "url": "http://demo-search.ch.lndo.site/en/articles/the-real-deal-for-supermarket-savvy-shopping",  
39   "imageUrl": "/sites/default/files/supermarket-savvy-umami.jpg"  
40 },  
41 {  
42   "nid": "14",  
43   "typeId": "article",  
44   "title": "The Umami guide to our favorite mushrooms",  
45   "type": "Article",
```

demo-search.ch.lndo.site/rest/

Not Secure | demo-search.ch.lndo.site/rest/api/search?query=mushrooms

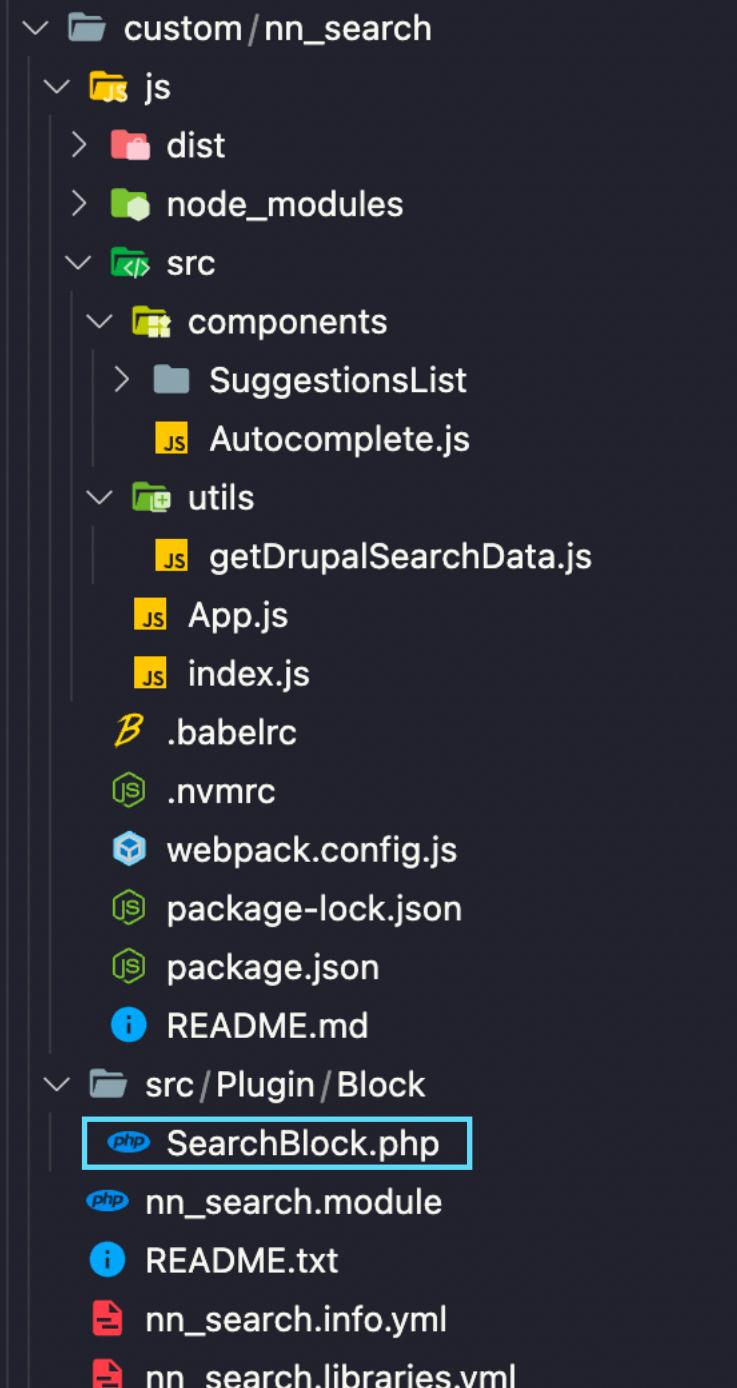
```
1 // 20220623141618
2 // http://demo-search.ch.lndo.site/rest/api/search?query=mushrooms
3
4 [
5 {
6   "nid": "14",
7   "typeId": "article",
8   "title": "The Umami guide to our favorite mushrooms",
9   "type": "Article",
10  "langcode": "en",
11  "url": "http://demo-search.ch.lndo.site/en/articles/the-umami-guide-to-our-favourite-mushrooms",
12  "imageUrl": "/sites/default/files/mushrooms-umami.jpg"
13 }
14 ]
```



# Building the search

## <Frontend />

# Custom module nn\_search

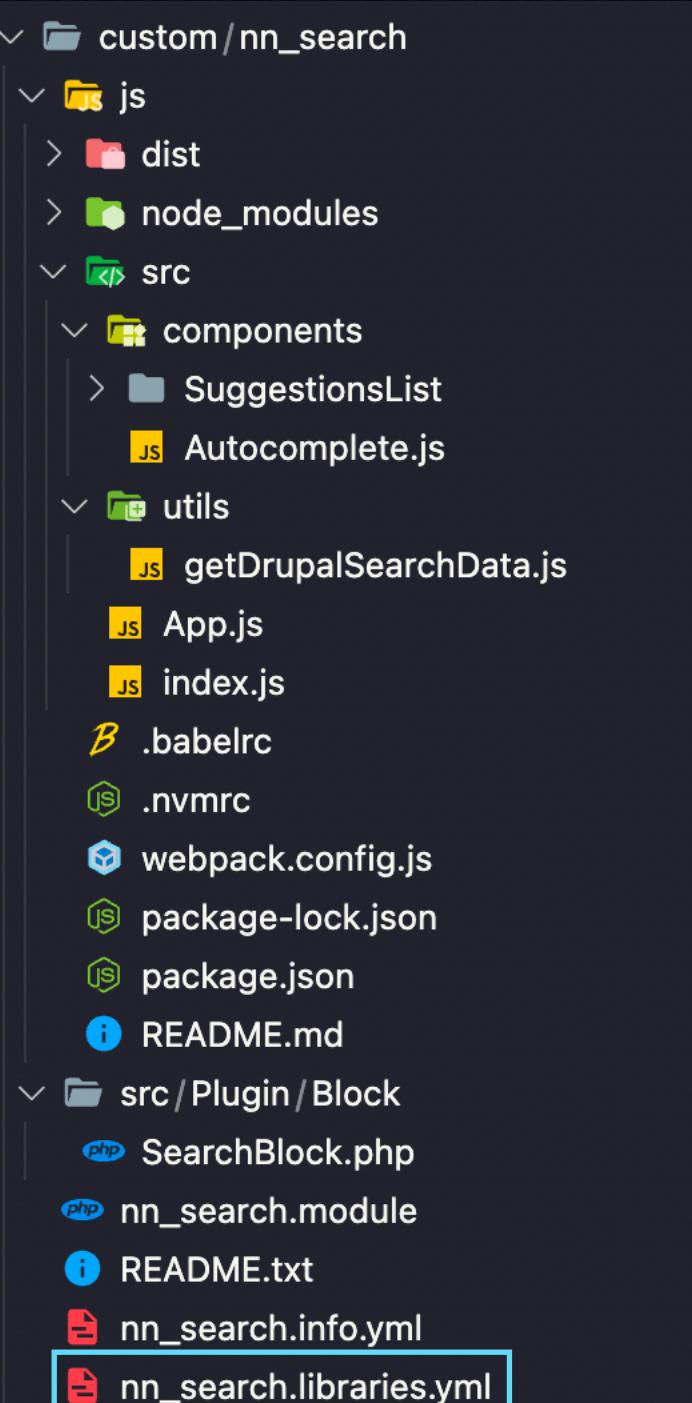


```
1  <?php
2
3  namespace Drupal\nn_search\Plugin\Block;
4
5  use Drupal\Core\Block\BlockBase;
6
7  /**
8   * Provides a 'Search Block' block.
9   *
10  * @Block(
11  *   id = "search_block",
12  *   admin_label = @Translation("Search Block")
13  * )
14  */
15 class SearchBlock extends BlockBase {
16
17  /**
18   * {@inheritDoc}
19  */
20  public function build() {
21    $language = \Drupal::languageManager()->getCurrentLanguage()->getId();
22
23    $build['search_block'] = [
24      '#markup' => '<section><div id="search-app"></div></section>',
25      '#attached' => [
26        'library' => 'nn_search/nn_search.react',
27        'drupalSettings' => [
28          'language' => $language
29        ],
30      ],
31    ];
32
33    return $build;
34  }
35
36 }
```

# Custom module nn\_search

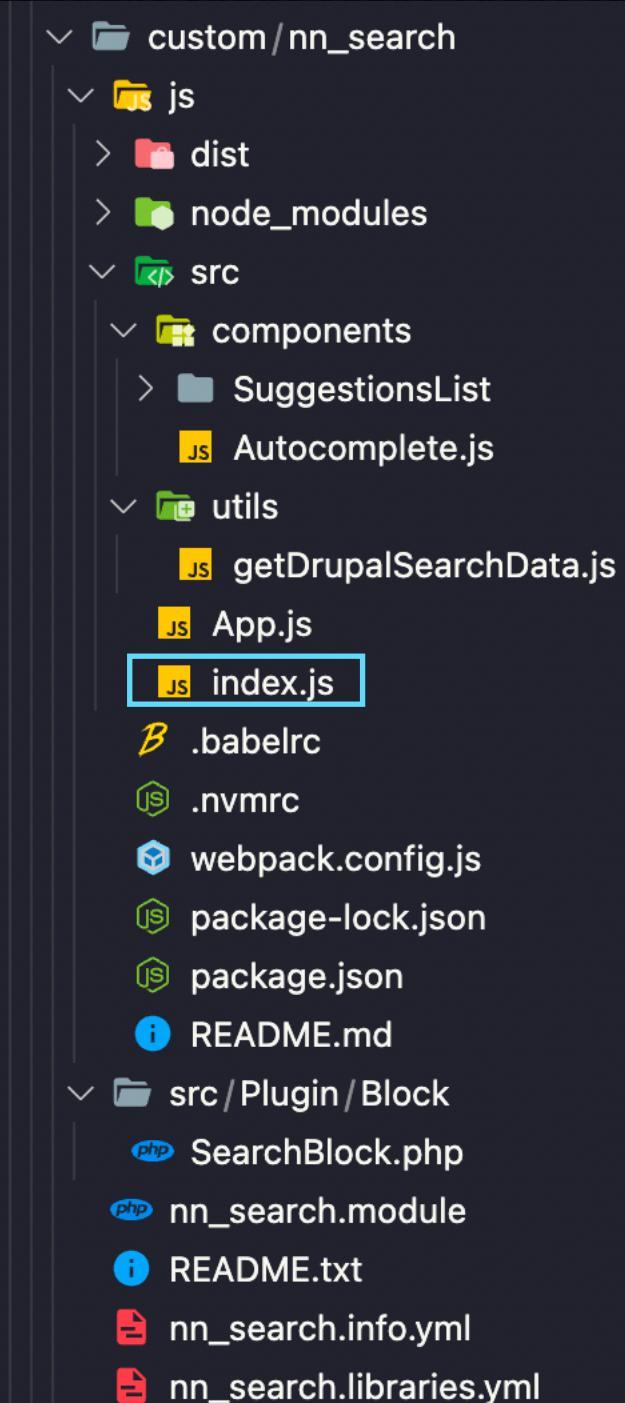
```
20 public function build() {
21   $language = \Drupal::languageManager()->getCurrentLanguage()->getId();
22
23   $build['search_block'] = [
24     '#markup' => '<section><div id="search-app"></div></section>',
25     '#attached' => [
26       'library' => 'nn_search/nn_search.react',
27       'drupalSettings' => [
28         'language' => $language
29       ],
30     ],
31   ];
32
33   return $build;
34 }
```

# Custom module nn\_search



```
● ● ●  
1 # React app.  
2 nn_search.react:  
3   js:  
4     js/dist/bundle.js: {}  
5   dependencies:  
6     - core/drupalSettings  
7
```

# Custom module nn\_search

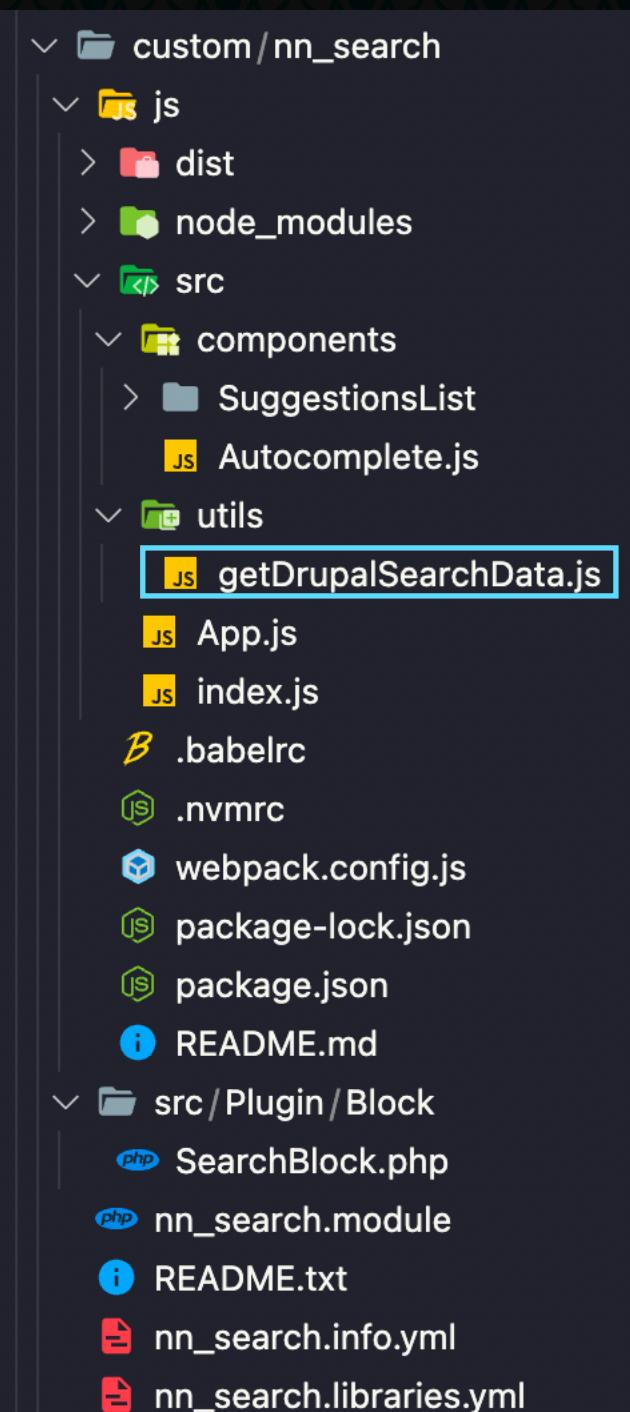


```
1 import React from "react";
2 import { createRoot } from "react-dom/client";
3 import App from "./App";
4
5 const container = document.getElementById("search-app");
6 const root = createRoot(container);
7
8 root.render(
9   <React.StrictMode>
10     <App />
11   </React.StrictMode>
12 );
13
```

# Custom module nn\_search

```
1 import React from "react";
2 import { createRoot } from "react-dom/client";
3 import App from "./App";
4
5 const container = document.getElementById("search-app");
6 const root = createRoot(container);
7
8 root.render(
9   <React.StrictMode>
10     <App />
11   </React.StrictMode>
12 );
```

# Custom module nn\_search



```
const API_URL = `/${drupalSettings.language}/rest/api/search`;

const getDrupalSearchData = ({ query }) => {
  const apiURL = `${API_URL}?query=${query}`;

  return fetch(apiURL)
    .then((res) => res.json())
    .then((result) => {
      return result;
    });
};

export default getDrupalSearchData;
```

patientendossier.ch

patientendossier.ch

Guest

EPD-Blog    Publikationen    Über uns    Kontakt    DE

Das EPD    Informationen    EPD eröffnen    X

Search...

MEINE GESELLSCHAFTSINFO,  
online verfügbar

Verwalten Sie Ihre eigenen Gesundheitsinfos und  
teilen Sie diese mit Ihren Gesundheitsfachpersonen.

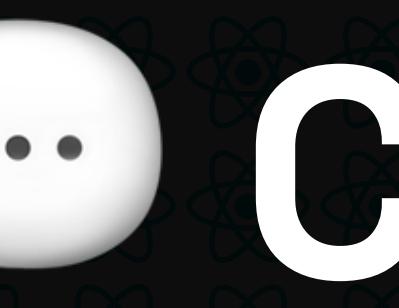
Hausarzt    Spezialärztin

Apotheke    Spital

Physiotherapeut    Pflegeheim und weitere

So nutzen Sie das EPD (2:34)

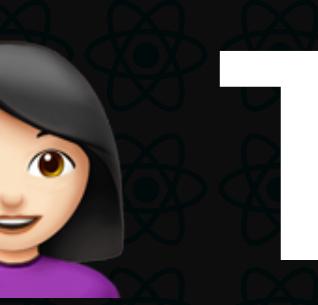
The screenshot shows the homepage of the patientendossier.ch website. At the top, there's a navigation bar with links for 'EPD-Blog', 'Publikationen', 'Über uns', 'Kontakt', and a language switcher 'DE'. Below the navigation is a search bar with a magnifying glass icon. The main headline reads 'MEINE GESELLSCHAFTSINFO, online verfügbar'. Below the headline, there's a subtext: 'Verwalten Sie Ihre eigenen Gesundheitsinfos und teilen Sie diese mit Ihren Gesundheitsfachpersonen.' To the right of the text are several circular icons representing different healthcare professionals and facilities: 'Hausarzt' (male doctor), 'Spezialärztin' (female doctor), 'Apotheke' (pharmacy), 'Spital' (hospital), 'Physiotherapeut' (physiotherapist), and 'Pflegeheim und weitere' (nursing home and others). In the center, there's an illustration of a person holding a tablet displaying the EPD logo, surrounded by medical icons like a stethoscope and pills. A video thumbnail at the bottom left shows a play button and the text 'So nutzen Sie das EPD (2:34)'.



# Conclusions

# freedom, flexible & powerful





# Thank you!

If you want to join us [✉️ jobs.netnode.ch](mailto:jobs.netnode.ch)

