

# [Dai:Si] - A modular framework for Dataset Search

---

This repository provides the source code for [Dai:Si] - a frontend framework for querying a dataset search index. The name is an abbreviation of 'Dataset Search' in its phonetic spelling. [Dai:Si] consists of two parts: a backend server (Node Server) and a frontend (Angular application). The Node middleware server handles the index requests and provides a convenient API for the frontend - a modular Angular app that can be easily adjusted for a specific domain. [Dai:Si] also provides a semantic search for biological datasets.

- [Angular app](#)
- [Node Server](#)

## Demo

A live demo is available here: <https://dev.gfbio.uni-jena.de/daisi/>

The Node Server and its API is also available: <https://dev.gfbio.uni-jena.de/daisi-api/api-docs/>

## How to setup Dataset Search UI for an own index

Please follow the instructions below if you want to use [Dai:Si] for your own search index. Make sure that your index is available

1. Create a new component for your index
2. Go to `app.component.ts` and add a new entry to the array of 'indexes' with a title and a URL (e.g., 'myNewIndex', '/myNewIndex').
3. Go to the `app-routing.module.ts`, import your newly created component and add an entry to the routes array, e.g.,

```
const routes: Routes = [ { path: '', component: GfbioComponent }, { path: '/myNewIndex', component: MyNewIndex } ];
```

4. Go to the html file of the component that you created and add the components you want to use. For instance, if you need the search-result component, which displays a search result and also includes pagination, you should add

```
<app-search-result [result]="result" (basket)="checkBox($event)"  
(from)="paginationClicked($event)"></app-search-result>
```

5. Then go to the `*.ts` file and implement from "SearchResult"(example:go to the `gfbio.component.ts`). You need to pass the content information through the "result" parameter (subscribed to the `communicationService.getResult()`). If a user clicks on an entry in the pagination component, you can get the click's action and the page number from the "paginationClicked" function. If the user clicks a check box in the results, you can get the click's action and the checked results, from "checkBox" function. If you need the filters component, you need to add the

```
<app-filters [result]="result" (filters)="filterSubmitted($event)"  
[resetFilters]="resetFilters"></app-filters>
```

Again you need to pass the results through a "result" parameter. If you need to clean the filters (example: new search key) by an action, you need to pass it by `resetFilters= true`. You can get the clicked filters by the "filterSubmitted" function.

If you need the search-input component, add

```
<app-search-input [checkboxValues]="basketValues" (basketChecked)="basketChecked()"
(searchKeyEmmit)="searchKeySubmitted($event)"></app-search-input>
```

You can add the elements that you want to put into the basket by `basketValues`. If the user clicks on the basket, the action can be received by the `basketChecked` function. If the user clicks on one of the search buttons (search, semantic), the action can be received by the `searchKeySubmitted` function. The input of the function is an array. The first item is the search key and the second one is a boolean value and represents if the search is semantic or not. Now you have all the information you need for sending the http request (such as search keys, filters, pagination, ...) - we are ready to send the request.

6. To send a request, you need a service which is responsible to map the results of the http request to the result object which is used in the search-result component to show the information. Create a new service under the services/local directory (example: `gfbio-preprocess-data.service.ts`). Go to the component that you want to send the request and inject the service in the constructor, e.g., "NodeService".
7. When calling the "search" method in the "NodeService", you need to pass 4 parameters (urlTerm, body, the service for mapping the result, parameters that you need in the mapping service). Check the example in the `Start-searching.Service.ts`.

## Issue Tracking

Please report bugs and issues in the GitHub issue tracker.

## Changelog

24.01.2021 v0.5a

- bugfix release of keycloak authentication service (angular & nodejs)

21.12.2021 v0.5

- integration of keycloak authentication service (angular & nodejs)
- REST API for basket(s) access
- persistent storage of baskets (via mysql/mariadb)

29.07.2021 pre-release 0.4\_beta

- Terminology widget added (info box in semantic search - explanation on expanded terms)
- dataset basket improvements: spinner added (creation of zip file can take some seconds), 'empty basket' function

07.07.2021 second official release 0.3

- minimap added (select a dataset and if coordinates are present the geographic location is shown on the map)
- dataset basket with download function (select datasets, open the basket and download all files as zip)
- related datasets/publications are displayed
- improved citation dialog (related datasets/publications added)
- bugfix semantic search: expanded terms are only displayed when they occur in the dataset
- config variables for Node and Angular are now provided in separate environment files

26.03.2021 first official release 0.2

- color and css revised
- additional information on demand (mouseover) added
- components revised for more modular structure
- bugfix semantic search - expanded terms highlighted in title and description
- collapse/expand for longer descriptions and parameters

05.03.2021 initial release 0.1

## License

[Dai:Si] is distributed under the terms of the GNU LGPL v3.0. (<https://www.gnu.org/licenses/lgpl-3.0.en.html>)

## Citation

Shafiei, F., Löffler, F., Thiel, S., Opasjumruskit, K., Grabiger, D., Rauh, P., König-Ries, B.: [Dai:Si] - A Modular Dataset Retrieval Framework with a Semantic Search for Biological Data, 2021

Sanfilippo, E. M., Kutz, O., Troquard, N., Hahmann, T., Masolo, C., Hoehndorf, R., Vita, R., Algergawy, A., Karam, N., Klan, F., Michel, F., Rosati, I. (Eds.), S4BioDiv 2021: 3rd International Workshop on Semantics for Biodiversity, held at JOWO 2021: Episode VII The Bolzano Summer of Knowledge, September 11–18, 2021, Bolzano, Italy

<http://ceur-ws.org/Vol-2969/paper4-s4biodiv.pdf>