Building Mathhub using React

Bachelor Thesis

Johannes-Sebastian See
Supervisor: Michael Kohlhase
Co-supervisor: Tom Wiesing
Friedrich-Alexander University, Erlangen Nürnberg, Germany

March 4, 2019

Abstract

Abstract will be added at the end

Contents

1 Introduction

1.1 Math Information Systems

1

1.2 Mathhub

1.3 Previous Implementation

Up until April 2018 the Mathhub frontend was realized with Drupal. Drupal is an open source content-management framework used by millions of different websites. Dealing with user interactions were handled by JavaScript modules in the JOBAD framework [KW14].

But in April 2018 a critical security flaw in the versions 6 to 8 went public. The problem was that the Drupal core in these versions accepts request parameters without any validation. This means the core processes any input from anybody [Tun18]. To exploit this weakness an attacker doesn't even need to log in or have any other privileges on a vulnerable website [McC18]. With this flaw it is possible to inject malicious code and compromise a website in multiple ways. This can be used to access, change and delete private data and create backdoors to make future attacks possible. The Drupal community called this weakness "Drupalgeddon2" while its official name was "CVE-2018-7600". Some code that was injected installed the program XMRig Monero miner, which is a cryptocurrency mining program, as well as deleting other mining programs on the compromised system [Kum18]. The National Institute of Standards (NIST) and Technology gave Drupal a "Highly Critical" Rating because of this vulnerability [NIS]. After this flaw was discovered a patch was published and a warning to update every website that used a vulnerable version was given.

Since there have been multiple flaws in Drupal before that compromised Mathhub, the decision to stop using it and rebuild Mathhub from the ground up to not be affected by future attacks, was made.

2 State of the art

2.1 Building an interactive Frontend

Before starting to build a completely new Mathhub frontend a different web framework had to be chosen.

Polymer is an open source JavaScript library developed and maintained by Google. It provides a set of features that make creating custom elements, that work like standard web components, easy. It is used for several Google

---

[1]EDNOTE: why are those special and different

services for example Youtube, Google Earth, Google Play Music etc. as well as Netflix, Electronic Arts and many other companies. [Wika]

Another open source web framework from Google is **Angular**. This TypeScript library has framework architectures that simplifies the development of new web applications. It also has Angular Material. A collection of UI components that work in browsers, on mobile and desktop. [Wikb]

After using Angular on several Google projects, Evan You decided to create his own JavaScript framework called **Vue.js** [Wikc] Depending on the project it can be scaled between a framework and a library. Vue.js separates its view layer library from its support libraries for complex applications, to create an easy approach to the framework. [Git]

In the end the decision was made to use **React** developed by Facebook. Further details about React and how it is used can be found in section 3.

2.2   Math Information Systems

- MathNet

- mathoverflow

- Wikidata

- Wikipedia

3   Preliminaries

3.1   The core concept of React

React is an open source JavaScript library owned and maintained by Facebook. It was created to build interactive user interfaces (UI). For example it is used for Facebook and Instagram. What makes React unique is its use of a virtual Document Object Model (DOM). The concept of the virtual DOM is that when updating a website not everything is rendered again. React computes the differences between the last and the next page and only changes the necessary parts. On top of that it has conditional rendering which means that an item will only be rendered if it is shown. The advantage of virtual and conditional rendering is that this makes updating a website fast, but it comes with high RAM costs. The actual interface is made up of many different elements and components. Since a website that uses React can have many different features it is helpful to build new components. [Inca] React does not have a styling system so it integrates Semantic UI to provide a consistent theme for the frontend.

3.2 Building new components in React

React already has a large library with a lot of different components, but it is often necessary to make new ones that have the desired functionality. In JavaScript new components can be implemented by creating either a function or a class. Their input variables are called props and can only be read. Components return React elements that are ready to be rendered. Naturally a component can grow big rather quickly. Luckily it is possible to use components inside other components. This comes with the advantage that they can be reused in many different locations. The difference between creating a new component as a function and as a class is that a class can have a private internal state, which can be updated an any time. Since props are read-only, updating the state can only affect lower components. If it is necessary to also change something in a higher component it is possible to "lift up" the state. This means adding the state that causes the change to the state of the component on a higher level and giving it back to the lower levels as a prop. If the update should affect a component on the same level creating a new component with that state that consist of all the one that are affected will make this possible. [Incb]

3.3 MMT and OMDoc

The abbreviation MMT is short for either <u>m</u>eta-<u>m</u>eta-<u>t</u>heory or <u>m</u>eta-<u>m</u>eta-<u>t</u>ool. Whereby meta-meta-theory represents the theoretical and meta-meta-tool the practical part of MMT. Basically it is a knowledge representation framework that uses formal languages to create a scalable module system for mathematical theories. That means that in MMT the features of the syntax and the semantics of a language are defined as individual, reusable modules. This way of building individual languages leads to a high degree of abstraction of advanced algorithms.

MMT integrates the OMDoc format, which is a XML format that describes a design for an uniform language for knowledge.
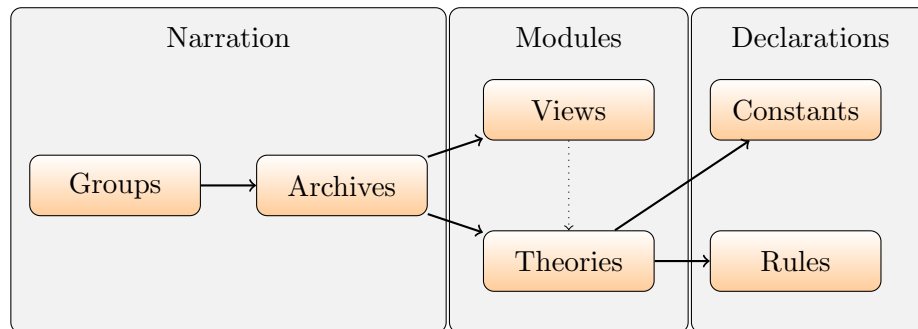

Figure 1: The MMT structure

4

To create a frontend that displays MMT it is important to understand its structure. First up on the highest level are the individual groups. Each group contains several archives. An archive can be described as a software project that provides a work flow for a language in MMT. [2] Groups and archives are just used for navigation and narration purposes. The modules that can be found in an archive, make up the actual content. These are either a theory or a view that shows the relation between different theories. A theory is defined by its rules and constants, the so called declarations. [Rab]

EdN:2

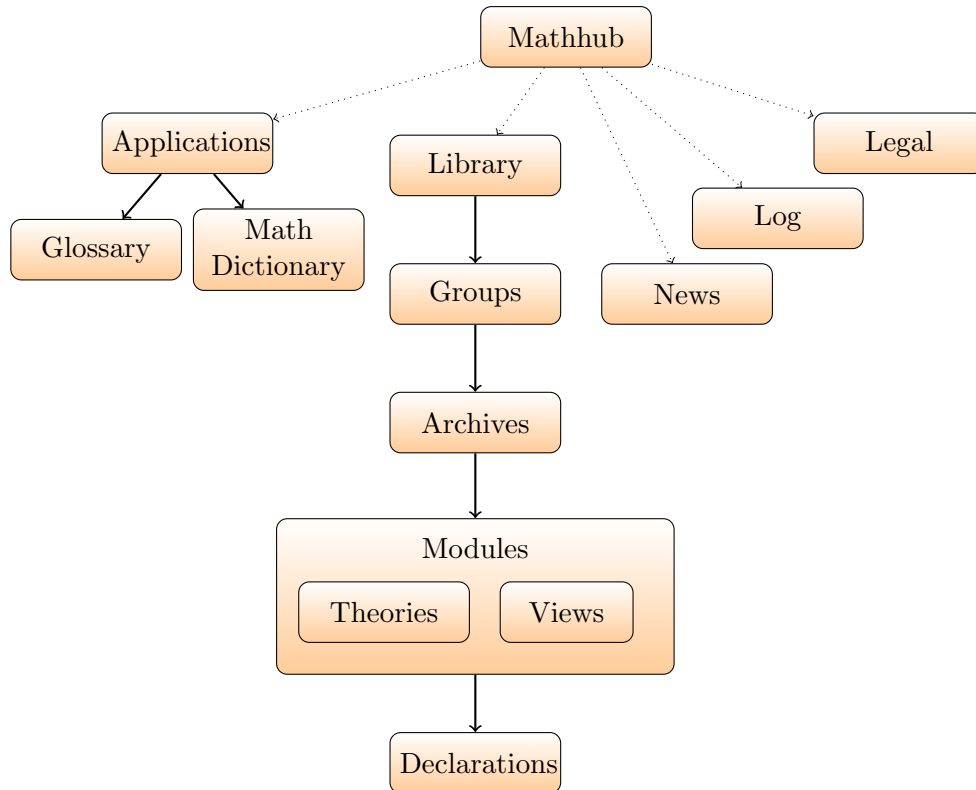# 4 The Architecture of Mathhub

4.1 Mathhub.info Routes

Figure 2: Mathhub components

To create a logical navigation system Mathhub.info is divided into numerous pages, that can be accessed from the Home-page.

---

[2]EdNote: is this to close to the source?

First of is the main content of Mathhub, the MMT library. It starts with the different groups. The next page displays the archives that belong to a specific group. Groups and archives only exists for narration and navigation purposes. So they don't have any actual content only meta information like names and descriptions. The actual content can be found in documents. These document consist of several theories and views. So the archive-page has a list of all the documents in an archive. On the document-page the modules are displayed with all their declarations.

With the huge mathematical library of theories there are a lot of technical terms. The glossary is a collection of these expressions. There wouldn't be much value in to just having a collection without any additional features. So the glossary also provides a definition for each term. Over the time many different authors have contributed to the theories, so it can happen that there are different terms that share a meaning. These synonyms can also be found in the glossary. Since many theories exist in multiple languages it makes sense to have a glossary available for every used language. Currently the biggest collection of terms are in the English glossary, followed by German and French. Smaller collections for Turkish and Romanian are also available as well as simplified and traditional Chinese.

Most of the times a user does not want to browse through the gigantic glossary to just find a single term. This is the reason why the Math Dictionary is a useful extension of the glossary. The main purpose of the Math Dictionary is to translate a term into another language and look up a definition of a specific expression.

There is also a page with all the latest news regarding Mathhub, [3] as well as pages for licenses and the privacy policy. At last there is a Log with the most recent messages from the backend. [4]

EdN:3

EdN:4

4.2   Layout

Every page consists of three parts: A header, a footer and the actual content in between.

The header is a menu with the routes to Home, the news, the glossary and the Math Dictionary as well some external links. Under the menu there are breadcrumbs for an easier navigation. In the footer the logos of the institutions that are involved in Mathhub can be found. There also are the routes to the Log, the licenses, the imprint and the privacy policy. The body itself depends on the page the user is currently on. [5]

EdN:5

---

[3]EdNote: or MMT? or KWARC?
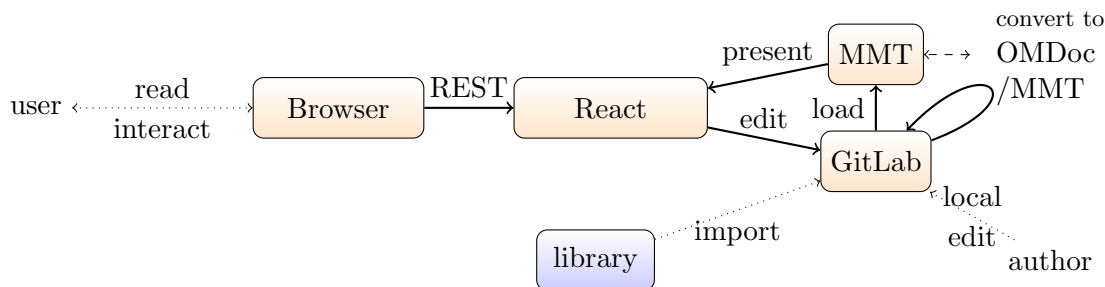[4]EdNote: is this right?
[5]EdNote: Screen shots

4.3 Realization

Figure 3: Mathhub architecture

The React based frontend that is displayed in the browser is written
in TypeScript. The use of Semantic UI React achieves an informal theme
throughout Mathhub.info. The React frontend is in constant communication
with an MMT server that provides the user with the actual content as well
as several semantic services.

Whereas React is used for the presentation of the content documents,
GitLab is used for their versioned storage. There the documents are orga-
nized into their proper repositories and converted from the source format
into the OMDoc/MMT format. These documents can be edited in a working
copy of the GIT repository. Afterwards the author can submit the changes
with a direct commit.

4.4 Communication with the Backend

As previously mentioned the frontend does not have any actual content.
It gets the data from the MMT backend. The frontend has several clients
that each communicate with the server when their specific functionality is
needed. The different clients are:

- a Library Client for everything related to the content of the library

- a Glossary Client that gets all the terms in a specific language

- a Translation Client that is used by the Math Dictionary to search for
  a translation of a term

- a News Client

---

[6]EDNOTE: name pending
[7]EDNOTE: look for author and user pictures
[8]EDNOTE: React doesn't edit anything yet, so maybe erase the edit-arrow?

- a Log Client

The answers that are received from the backend use the JavaScript Object Notation (JSON). The data objects in JSON consist of attribute-value pairs and arrays. The frontend then uses these objects to build React components.

5 Mathhub Library Components

This chapter is about the implementation of the different components of the Mathhub library. The first page of the library is a list of all the groups of Mathhub. Every group is rendered in its own React component that has the name of the group, a short teaser and links to the corresponding group-page.

5.1 Groups

Above the numerous archives the group page starts with a header that has the same structure for every group. It begins with a button that links to its source files on GitLab. After that follows a description that gives an overview of the groups content. The header ends with a list of e-mail addresses of the people that maintain the group.

Beneath that there is a list of all the corresponding archives. Every archive is its own React component that consists of a name and a short teaser that summarizes its content for the user. By clicking on an entry the user is taken to the corresponding archive-page.
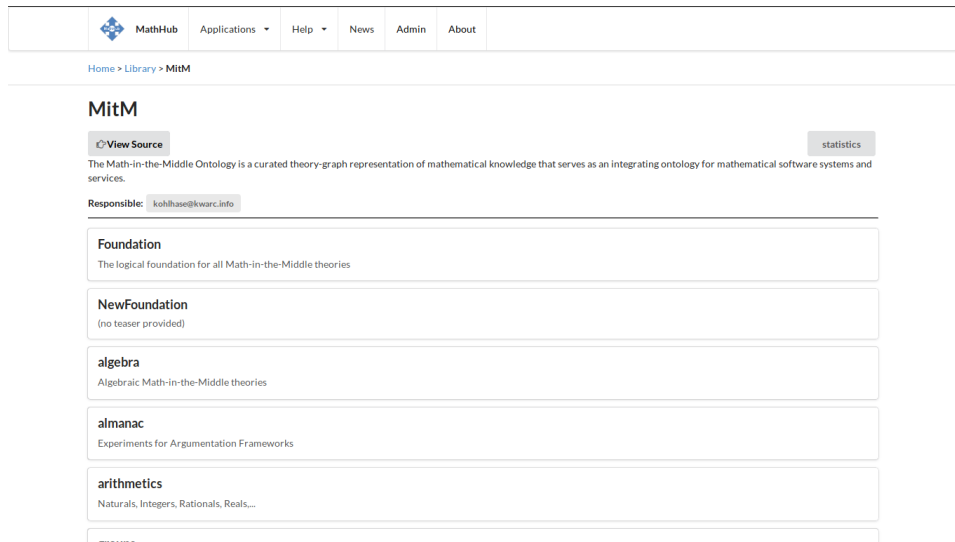

Figure 4: a group in the frontend

5.2 Archives

The structure of the archive-page is very similar to the group-page. It also starts with a header that consists of a button that links to the source files, a description of the archive and the e-mail addresses of the responsible people.

After the header follows a list of the documents in that archive. There is the main difference between the group-page and the archive page. The document entries do not have a teaser text so they just link to the document-page.
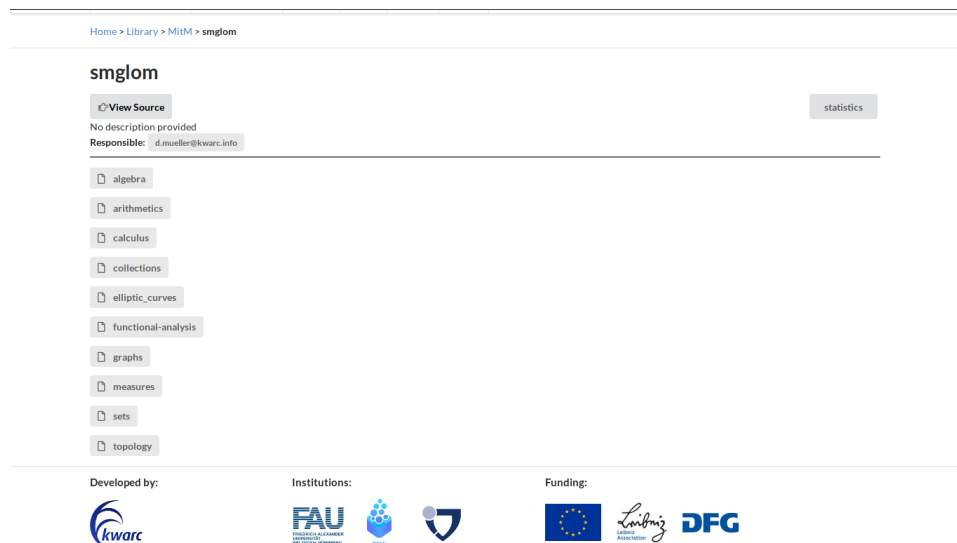

Figure 5: an archive in the frontend

5.3 Documents

Depending on the document the document-page can be either a list of OM-Docs or the different modules of an OMDoc. [9] If the document is in the OMDoc format there is also a button that links to its source file on GItLab. The modules of a document are expandable React components that consist of a name, a type, either theory or view and a button to show further details. Clicking on the button shows all the declarations and nested theories inside of a module. These declarations (structure elements or constants) and theories can be further extended to show their own content.

In between the modules there can be opaque elements. These are just text that help the user to understand the content of the document and do not serve any contribution to the actual theories
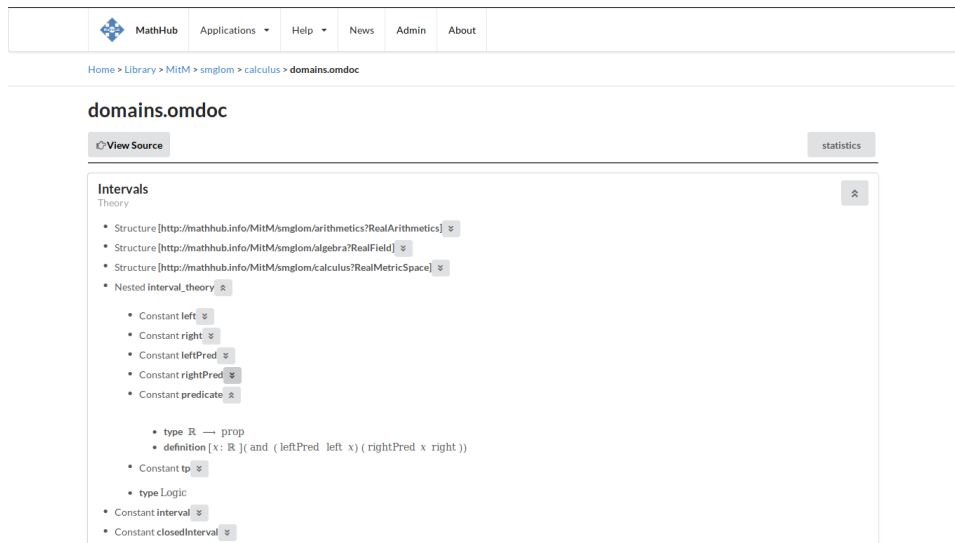
EdN:9

---

[9]EDNOTE: can i write it like this?

Figure 6: a document in the frontend

5.4 Statistics

On every group-, archive- and document-page there also is a statistics button that shows some available statistics about that particular group, archive or document. When the cursor hovers over a keyword of a statistic that keyword is explained in a pop-up.
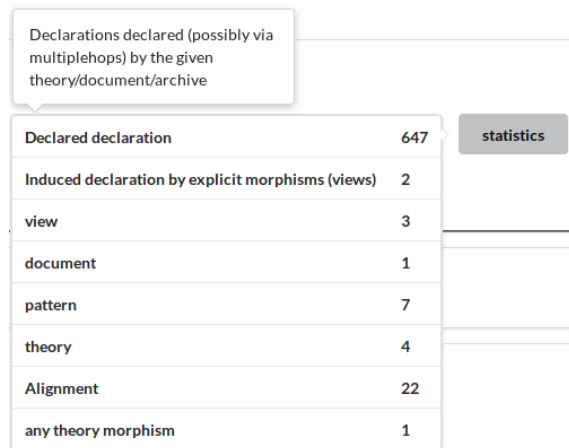

Figure 7: statistics in the frontend

6 The Applications of Mathhub

6.1 Glossary

The glossary-page has a tab for every available glossary. At any given time only the terms are rendered that have an entry in the currently selected language. Thus it is possible to change languages by either changing the language tab or clicking on a language-button inside an entry. If there is a button with a different language available, this means this particular entry also exists in that language. To create a better overview the definition of a term is not immediately shown. By clicking on an entry the definition becomes visible.
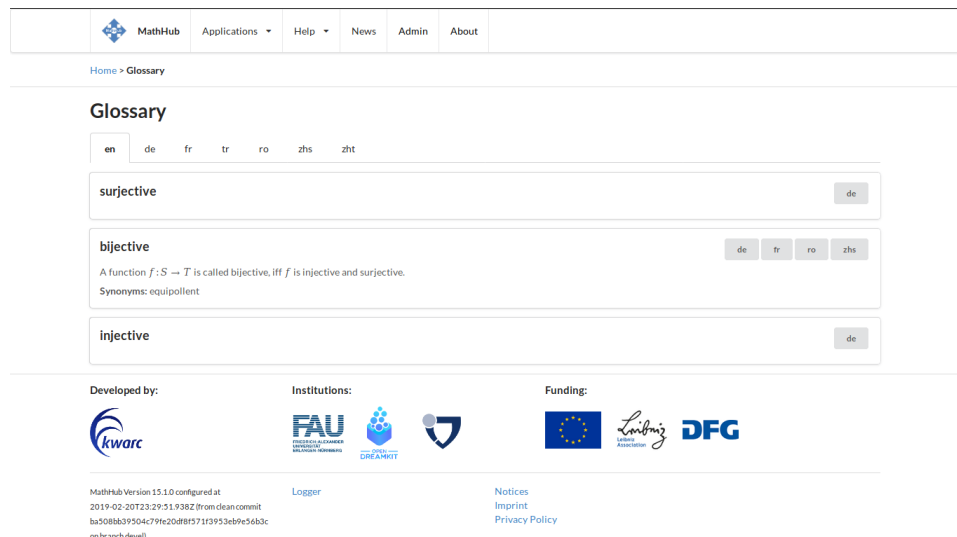

Figure 8: the glossary in the frontend

6.2 Math Dictionary

To translate a term with the help of the Math Dictionary the user has to select the language from a dropdown menu in which the term currently is and also the language to which it should be translated into Pressing the "translate" - button sends a translation request to the server. Until the servers responds, the message "translating" is shown and the button is disabled to prevent sending to many translation requests. If a translation exists then the translated term, its definition and potential synonyms are shown. By selecting the same language for "from" and "to" the Math Dictionary can also be used to get the definition for an expression without searching the glossary.
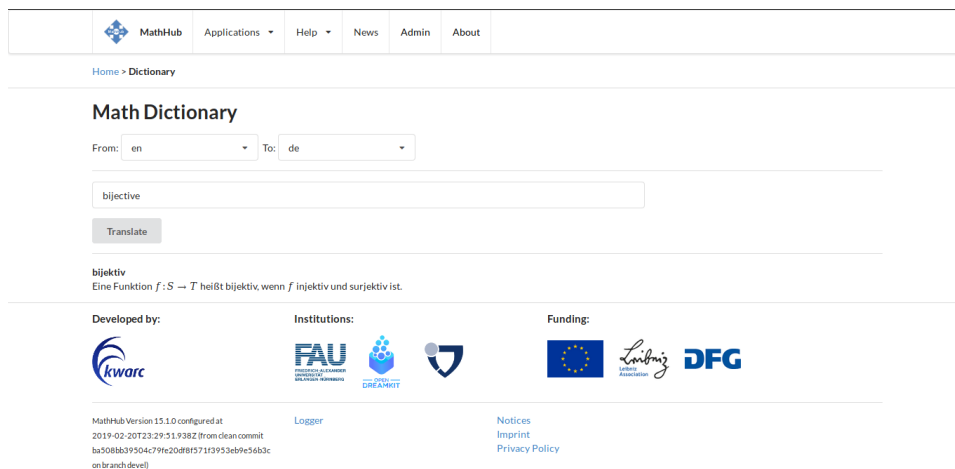
Figure 9: the Math Dictionary in the frontend

7 Conclusion

8 Future Work

8.1 TGView

8.2 MathWebSearch

8.3 Subset Frontends

8.4 Issue report: Mathhub and content

References

[Git]   Github. URL: https://github.com/vuejs/vue.

[Inca]   Facebook Inc. URL: https://reactjs.org/.

[Incb]   Facebook Inc. URL: https://reactjs.org/docs/getting-started.html.

[Kum18]   Mohit Kumar. "Hackers Exploiting Drupal Vulnerability to Inject
          Cryptocurrency Miners". In: *https://thehackernews.com/* (2018).
          URL: https://thehackernews.com/2018/04/drupal-cryptocurrency-hacking.html.

[KW14]   Michael Kohlhase and Tom Wiesing. *In-place computation in active documents*. Jacobs University Bremen, 2014.

[McC18]   Kieren McCarthy. "Running Drupal? You need to patch, patch, patch right now!" In: *https://www.theregister.co.uk/* (2018). URL: https://www.theregister.co.uk/2018/03/28/drupal_urgent_security_software_patch/.

[NIS]   NIST. URL: https://nvd.nist.gov/view/vuln/search-results?cpe=cpe%3A%2Fa%3Adrupal%3Adrupal%3A7.38&page_num=0&cid=3.

[Rab]   Florian Rabe. *The MMT Language and System*. URL: https://uniformal.github.io/doc/.

[Tun18]   Liam Tung. "Update Drupal ASAP: Over a million sites can be easily hacked by any visitor". In: *www.zdnet.com* (2018). URL: https://www.zdnet.com/article/update-drupal-asap-over-a-million-sites-can-be-easily-hacked-by-any-visitor/.

[Wika]   Wikipedia. URL: https://en.wikipedia.org/wiki/Polymer_(library).

[Wikb]   Wikipedia. URL: https://en.wikipedia.org/wiki/Angular_(web_framework).

[Wikc]   Wikipedia. URL: https://en.wikipedia.org/wiki/Vue.js.

EdN:10