Building Mathhub using React

Bachelor Thesis

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

February 11, 2019

Abstract

Abstract will be added at the end

Contents

1 Introduction

1.1 Mathhub

1.2 Previous Implementation

Up until April 2018 the Mathhub fronend 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 witht 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 this was not the first detected flaw in Drupal the decision to stop using it and rebuild Mathhub from the ground up to not be affected by future attacks was made.

1.3 Building an interactive Frontend - State of the art

- Polymer
- Angular
- Vue.js
- React

2 Preliminaries

2.1 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]

2.2 Building new components in React

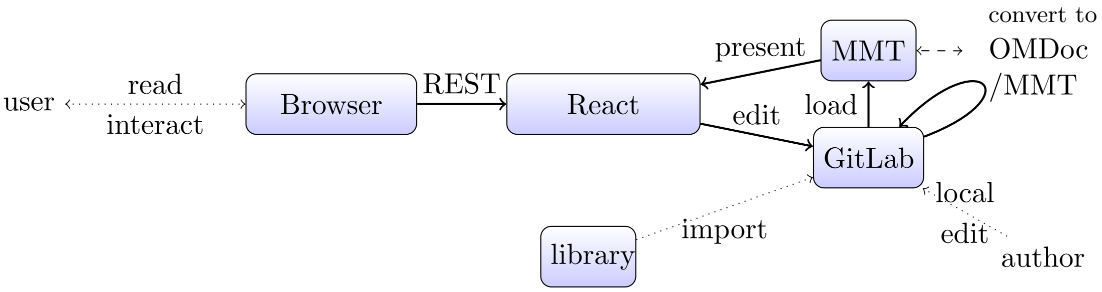
React already has a large library with a lot of different components, but it is often necessary to make new ones that have the desired functionality. In JavaScript new components can be implemented by creating either a function or a class. Their input variables are called props and can only be read. 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]

2.3 MMT and OMDoc

- OMDoc: XML format of how a uniform language for knowledge should be designed
- MMT is a scalable Module System for Mathematical theories
- MMT is a framework for knowledge representation using formal languages
- individual features are defined as reusable modules -¿ create individual languages
- high degree of abstraction of advanced algorithms
- archives: correspond to software projects; work flows for languages in MMT
- groups / libraries: collection of archives
- module: theory or view
- view: relations between theories
- theory: defined by declarations
- declaration: constants and rules
- groups and archives just for narration

- modules and declarations are actual content [Rab]

3 The Architecture of Mathhub



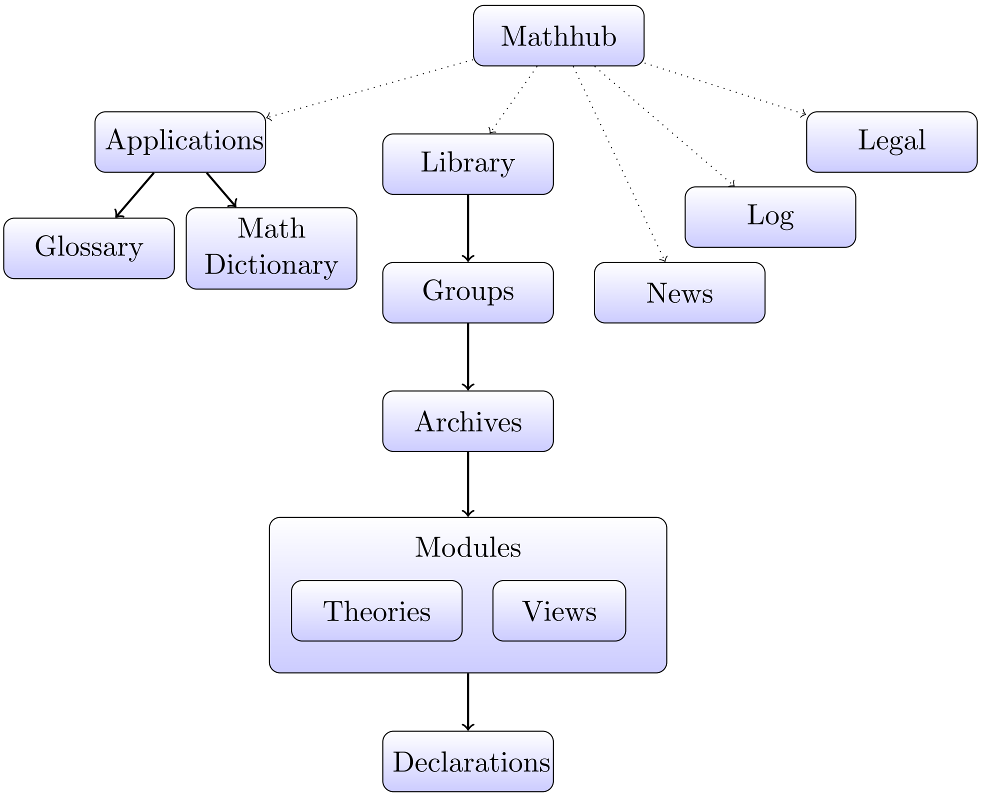
[1]

EdN:1

- React for building pages and interaction with the backend/MMT
- Semantic UI React for theming
- In the web-based system, semantic services (notation-based, presentation, definition lookup, relational navigation, dependency management, etc.) are provided by MMT and are made available to the user, primarily by dedicated React components.
- Gitlab used for versioned storage of the content documents, and organizes them into repositories
- conversion to OMDoc
- OMDoc functionalities and semantics for presentation

[1] EdNote: look for author and user pictures

3.1 Mathhub Routes



- Routes to every step in the MMT Library
- groups and archives no actual content only for navigation purposes
- Documents are a list of actual content like theories and declarations
- own page for Library → archives → documents
- some Legal stuff (Licenses and Privacy Policy)
- Log

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. 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 the used languages. 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.[2] EdN:2

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.

3.2 Layout

- Header: Menu with routes to Home, Applications (Glossary, Math Dictionary), Help and About
- Footer: Developed, Institutions, Funding, Logger and Legal [3] EdN:3
- body dependent on page

3.3 Communication with the Backend

- many clients communicate with a (MMT) backend
- Rest
- JSON

4 Mathhub Components

[4] [5] [6]

EdN:4
EdN:5
EdN:6

4.1 Groups

- link to source on gl.mathhub.info
- description text of the group
- list of archives in the group with short teaser
- every archive is a React component

[2] EdNote: What the hell even is zhs and zht?

[3] EdNote: Screenshots

[4] EdNote: what exactly have I done in the Frontend with these

[5] EdNote: don't go into to much detail about the implementation; eg entry has own react component is okay but don't talk about cards and stuff

[6] EdNote: get some nice screenshots

4.2 Archives

- link to source on gl.mathhub.info
- description text of the group
- list of the documents in the archive

4.3 Documents

- extendable modules
- extension shows declarations like structure and constants
- opaque elements: text for humans

4.4 Statistics

- groups, archives and documents can have statistics
- visible by button
- popup for explanation of keywords

5 The Applications of Mathhub

5.1 Glossary

- Different language tabs
- only language relevant terms are rendered

In the frontend 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 this language. The definition of a term is not immediately shown to create a better overview. To makes the definition visible the user can click on the entry.

5.2 Math Dictionary

To translate a term with the help of the Math Dictionary so the user has to select the language in which the term currently is and also the language to whichs 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.

6 Conclusion

7 Future Work

7.1 TGView

7.2 MathWebSearch

7.3 Subset Frontends

7.4 Issue report: Mathhub and content

References

[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/`.