UNIVERSITÄT
BERN

Implementing a Citation Search Engine in JavaScript

Student: Alexandru Filipescu

Supervisor: Prof. Oscar Nierstrasz

14.12.2021

Software Composition Seminar, Schützenmattstrasse 14.

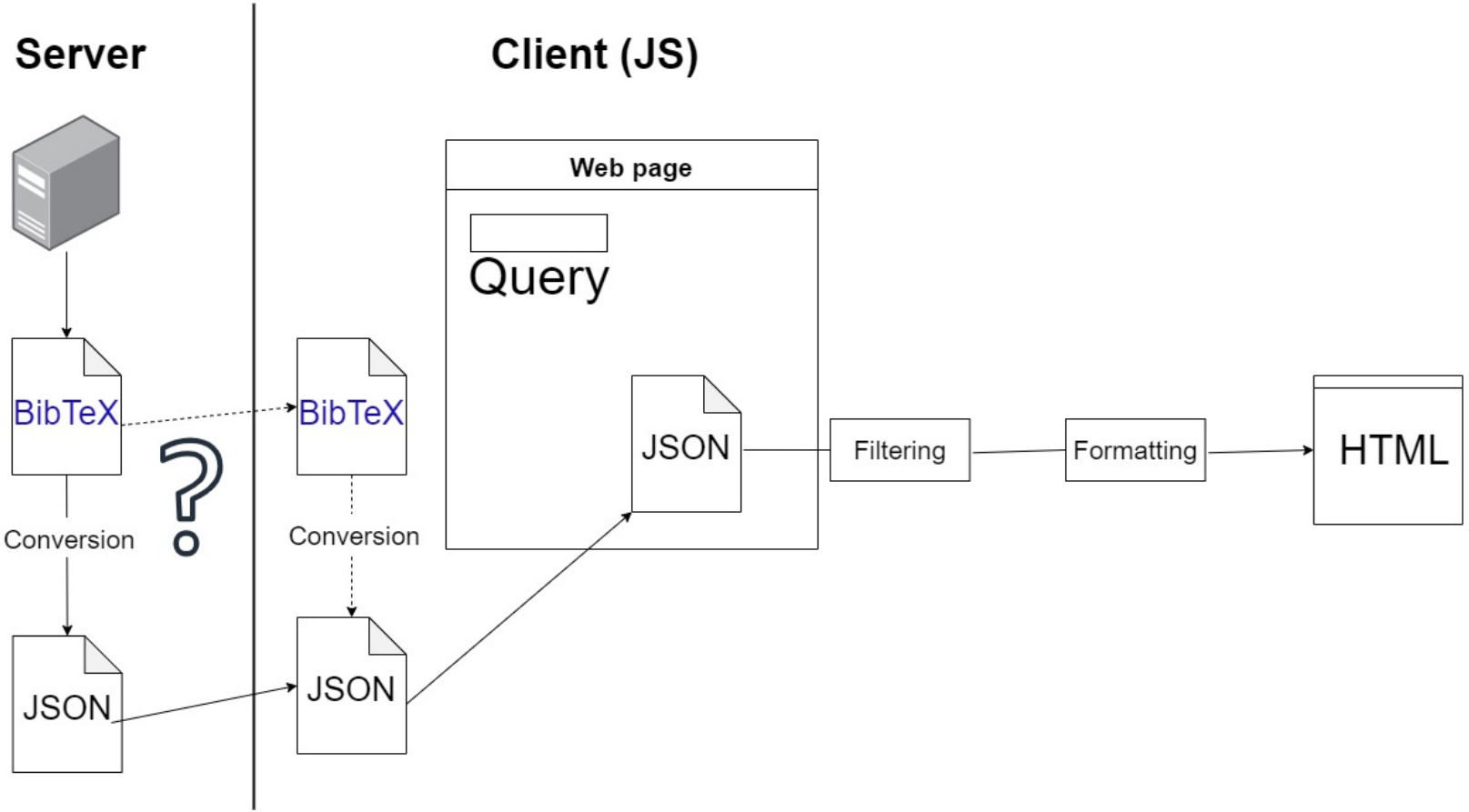
The challenge

1. **Responses retrieved in 10-20 seconds.**
2. **Unreliable server response (502 Bad Gateway).**
3. **Reorganize the filters.**

The solution?

To reimplement the search engine on the client-side. Once the database is loaded in the browser it will be possible to do local queries.

The Architecture



Challenges encountered

1. **Not enough BibTeX → JSON libraries**
2. **Concatenation complexity**
3. **Filtering on multiple criteria**

1) Not enough bibTeX → JSON libraries

Parser	bib2json (old)	bibtex-parse (new)
Speed/conversion	10-13 seconds	0.8 seconds

1. **bib2json** library had its last commit 4 years ago, while **bibtex-parse** 2 years ago.
2. **bib2json** was used in a synchronous, block mode as well as **bibtex-parse**.

2) Formatting complexity

Typical HTML manipulation with JavaScript:

```
objectArray.forEach( (item, index) =>{
  var url, title,bookTitle,month,address,publisher,s
  var li = document.createElement('li');
  li.className = 'list-group-item';
  li.innerHTML = '<b>Author:</b> ' + authorArray +
                                      url +
   ', '                              + title +
   ', '                              + bookTitle+
                                      url;

  list.appendChild(li);
```

3) Filtering on multiple criteria

Typical filter in JavaScript

```
function filterIt(searchTags)
{
    return objectArray.filter(object =>
        searchTags.every(tag => Object.values(object)
        .some(value => value.includes(tag)) //"Nierstr
        ));
```

The technologies used

1. jQuery is a fast, small, and feature-rich JavaScript library.
2. Bootstrap is an HTML, CSS & JS Library that focuses on simplifying the development of informative web pages.
3. Handlebars provides the necessary tools to let you build semantic templates effectively.

handlebars

What is Handlebarsjs?

Handlebars is a simple templating language.

It uses a template and an input object to generate HTML or other text formats. Handlebars templates look like regular text with embedded Handlebars expressions.

Input

```
{
  firstname: "Yehuda",
  lastname: "Katz",
}
```

Template

```
<p>{{firstname}} {{lastname}}</p>
```

Output

```
<p>Yehuda Katz</p>
```

Custom helpers

Helpers can be used to implement functionality that is not part of the Handlebars language itself.

```
Handlebars.registerHelper('loud', function (aString) {
    return aString.toUpperCase()
})
```

Template

```
<p>{{firstname}} {{loud lastname}}</p>
```

Output

```
<p>Yehuda KATZ</p>
```

How to use handlebars.js

We can iterate over a list by using the built-in helper *each.*

We can also use if helpers to conditionally render a block.

```
<template id="item-template">
 {{~#each items}}
    <div class="accordion-item">
      <h2 class="accordion-header" id="{{key}}random">

      <button class="{{#if ABSTRACT}}accordion-button {{else}} my-accordion-button {
            <div>

              {{#if (equals type "article")}} {{joinToEnd ', ' AUTHOR TITLE JOURNAL
              VOLUME NUMBER (join '' MONTH ' ' YEAR) PAGES NOTE }}
              {{/if}}
```

Live demo

http://scg.unibe.ch/download/oscar/Citation-Search-Engine/front-end/

What did we accomplish?

A working citation search engine that is faster than the original implementation.

What is left to be done?

- Further improving queries speed
- Implementing Regular expressions
- Adding case sensitive searches
- Alternative implementation with Elasticsearch (top-notch)

Lessons learned

1. Using a template engine.
2. Using state management would have made the development of the search engine a lot easier in the end.
3. List.js could have been a better alternative as it combines a templating engine, searching, sorting and filtering bundled together.