Headless CMS, ReactJS, and REST API

build a new user interface for a toy museum in Eslöv

Namn: Savarin Sitisarn

Klass: WCMS17

Handledare: Jens Grips

Datum: 2 maj 2019

Innehållsförteckning

Sammanfatting

I combine my interest in Headless CMS and design. The method I used in the project is ReactJs, WordPress, and flexbox. ReactJs is used to build a new user interface and WordPress is only for store text and images. To display the content, I made a request to URL endpoint, which I got it from WordPress and I styled the website with flexbox. The purpose of this project is that I deepen my knowledge of JavaScript and CSS. Moreover, I gain a basic knowledge of ReactJS.

The website that I choose is a toy museum in Eslöv and I intend to use it as an exercise. Therefore, the website is uploaded on my server.

Headless CMS, ReactJS, and REST API

Syfte och mål

The purpose of this project is to learn more about JavaScript, HTML, and CSS. Back to the JavaScript class, I found that the last assignment was intriguing. I worked with REST API, WordPress, and jQuery and after the first internship, I realized I wanted to focus on frontend development.

Moreover, I see that React is one of the most popular JavaScript libraries used in the frontend. Therefore, I came up with this project and the idea of combining my interest in JavaScript and focusing on being frontend in the future. The website, which I choose to develop, is a toy museum in Eslöv (Eslöv Leksaksmuseum) and I intend to use their website as an exercise.

The goal of developing this website is that I gain more knowledge in JavaScript and design, and gain basic knowledge of React.

Avgränsning

I did not focus on editing the content because I only took original content from their website directly, without editing it.

Tidplan

The project started 1st January and ended 19th April 2019.

Task	Duration	Date
Do the design	10 hours	1 January - 7 January
Set up workspace	3 hours	7 January - 11 January
Learn React	40 hours	1 January - 29 March
Work on pages	48 hours	1 January - 19 April
Change the design	6 hours	5 March - 11 March
Responsive design	12 hours	8 April - 26 April
Web testing	6 hours	22 April - 26 April

Beskrivning av arbetets gång

Do the design

Before doing the design, I examined their website, for example, what kind of information they have, how the website structure, or how many pages they have. Then, I wrote down in paper to analyze with MOSCOW method.

It took me approximately 6 hours to sketch the layout on papers ,and then I transferred it to Adobe XD to complete the sketch with actual colors and images. Meanwhile, I do some research about theme colors, which is a good theme that would fit the website.

Set up workspace

The tools that I used during doing the project were:

- *Adobe XD*, where I did the design and sketch.
- *Trello*, where I managed tasks to help me keep track of lists.
- *Toggl* to manage the project in terms of timing.
- *Visual Studio Code*, where I wrote all the code.
- *Paper*, where I did the design and quick short notes.
- *FileZilla* to upload the web production.

Learn React

I bought two courses from Udemy where they teach how to build a website with React and REST API. Moreover, I watched many React tutorial videos on YouTube and many more in different forum and websites. The learning process took place during the website production.

Work on pages

When I finished with the design, I started to install the React app project in the local environment, and I followed React tutorial in Udemy at the same time. Furthermore, to get JSON from WordPress, I installed

WordPress with a default theme, Twenty Nineteen, to create pages, but later I did a new installation due to the problematic link.

Each page has it owns URL endpoint to get each page in WordPress. In React app, I created 4 pages components to represent 4 pages on the website; about us, collections, models, and contact. Also, the website consisted of 14 small components.

Responsive design

After the website had text, images and basic styles such as background-colors and background-images. I fully concentrated on this process because my goal is that the website would perfectly fit any screen.

Web testing

The last process of this project is to test the website by sending it to friends and a family to ask them to interact with the website and get feedback from them. Besides, I look up the website in different kind of screen and examined through the design.

Erfarenheter

Restart the project many time

At the beginning of learning React and setting up components, problems occurred ,and I could not solve the problems. So, I decided to restart the work instead because I tried to solve them for many days.

However, problems happened many time and I learned how to solve the problems and avoid what would cause them. Some problems appeared after I installed new dependencies in package.json. I solved the problem by reading the error messages, follow the instruction or google it.

A different way to manage and install modules

To install modules in React needs to add dependencies in package.json via npm. Some modules are easy to install and do not need to manually add dependencies. Meanwhile, some modules need to add dependencies by yourself and I think it is difficult for me to understand how to do it ,and sometimes it does not work as I expected. For example, I have installed Google Maps modules and the project stopped running, so I chose to remove the module and restart the project.

The design changed

There are two reasons I changed it because:

1. I finished the front page, and I felt that the design was incomplete. In the beginning, I tried to have the same structure as the original website, but the content was short, only contact information. I think that the new website should be more attractive, and it can give more information about the museum. Therefore, I borrowed some information from another websites that mentioned or wrote about the toy museum in Eslöv. As can be seen in the review section on the front page, I took reviews text from TripAdvisor. However, to make the front page more complete in design and more engaging for users, I generated dummy text in activities and exhibition sections.
2. I spent time around 15 hours on a responsive menu bar. The idea of the menu bar is hamburger fold-out menu in mobile and tablet. In React app does not provide readymade responsive menu bar, so I started building the menu bar from scratch and it was not easy as I thought. Eventually, I got a nice hamburger menu bar with pure CSS, but the links did not work at all. I attempted to solve the issue for two weeks and I finally had to change the design so that the project could continue. I redesigned the front page, and apply the navbar component in Bootstrap to get a responsive menu in mobile and tablet.

Learn more in CSS and flexbox

In this project, I aim to learn flexbox for styling and I found that it was easy to use and great. Most of the styling used flexbox.

Struggle with broken URL endpoint

The project had been done 50% and the website suddenly could not display content, which I made a request to get JSON from WordPress. The problem was “Access-Control-Allow-Origin”. I intensively struggled to solve the issue for weeks.

One day I realized I should look something different, therefore, I replaced URL endpoint with another WordPress URL to see how React app response to it. As a result, the website showed content. So, I installed a new WordPress and a new theme, Summertime Adventure, to avoid a broken URL endpoint.

Deploy the website

Another challenge for me is to publish the website because I read some articles, and it said that to upload React app to live server needs to do the configuration in some files.

I tried with many alternatives that they suggest in forum and websites, but none of them worked. However, I found an application deployment that could help me to upload React app easy, it called Netlify. I think it is uncomplicated to use, but I did use it. I finally deployed the website in FileZilla with a simple and easy way on my server.

Black and grey or red and yellow

As I mentioned above that I wanted to improve design skill because I think it is difficult to decide theme colors that would fit the toy museum website. I wanted the website looks fun and attractive. Using strong and bright colors makes the website alluring but I do not want to have too many colors in it and it would be difficult to control the styling. So, I decided to keep the website looked simple and pleasant for users.

Begrepp och terminologi

React app

React application also known as React.js or ReactJS is a JavaScript library for building user interfaces.

REST API

Representational State Transfer. A RESTful API -- also referred to as a RESTful web service -- is based on representational state transfer (REST) technology, an architectural style and approach to communications often used in web services development.

jQuery

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

Headless CMS

A headless CMS is a back-end only content management system (CMS) built from the ground up as a content repository that makes content accessible via a RESTful API for display on any device.

The term “headless” comes from the concept of chopping the “head” (the front end, i.e. the website) off the “body” (the back end, i.e. the content repository).

A headless CMS remains with an interface to add content and a RESTful API (JSON, XML) to deliver content wherever you need it. Due to this approach, a headless CMS does not care about how and where your content gets displayed. A headless CMS has only one focus: storing and delivering structured content.

Udemy

Udemy is an online learning and teaching marketplace.

MOSCOW method

The MoSCoW method is a prioritization technique used in management, business analysis, project management, and software development to reach a common understanding with stakeholders on the importance they place on the delivery of each requirement.

The term MoSCoW itself is an acronym derived from the first letter of each of four prioritization categories (Must have, Should have, Could have, and Won’t have), with the interstitial Os added to make the word pronounceable. While the Os are usually in lower-case to indicate that they do not stand for anything, the all-capitals MOSCOW is also used.

Endpoint

An endpoint is one end of a communication channel. When an API interacts with another system, the touchpoints of this communication are considered endpoints.

For APIs, an endpoint can include a URL of a server or service. Each endpoint is the location from which APIs can access the resources they need to carry out their function.

APIs work using ‘requests’ and ‘responses.’ When an API requests information from a web application or web server, it will receive a response. The place that APIs send requests and where the resource lives, is called an endpoint.

Flexbox

The Flexbox Layout officially called CSS Flexible Box Layout Module is new layout module in CSS3 made to improve the items align, directions and order in the container even when they are with dynamic or even unknown size.

Referenser och källförteckning

React tutorial: Intro to React. https://blog.bitsrc.io/8-react-application-deployment-and-hosting-options-for-2019-ab4d668309fd

React: AJAX and APIs. https://reactjs.org/docs/faq-ajax.html

Fetching API Data with React.js, Ethan Jarrell, August 7, 2017. https://blog.hellojs.org/fetching-api-data-with-react-js-460fe8bbf8f2

How to Fetch Data in React, July 6, 2018. https://www.robinwieruch.de/react-fetching-data/

Build a Basic CRUD App with Node and React, Braden Kelly, July 10, 2018. https://developer.okta.com/blog/2018/07/10/build-a-basic-crud-app-with-node-and-react

React + WordPress: REST API, Rashain Jayanetty, Febuary 2, 2019. https://blog.usejournal.com/react-wordpress-rest-api-c6e94595e4d7

REST API Handbook. https://developer.wordpress.org/rest-api/

WordPress REST API: What it is and how to get started using it, John Hughes, July 20, 2018. https://www.codeinwp.com/blog/wordpress-rest-api/

8 Ract Application Deployment and Hosting Solutions for 2019, Nwose Lotanna https://blog.bitsrc.io/8-react-application-deployment-and-hosting-options-for-2019-ab4d668309fd

React js Tutorial - How to get data from an API with React, Fullstack development, May 5, 2018. https://www.youtube.com/watch?v=hzLDsxPGctY

Beginner React (2019): create a movie web app, Thomas Weibenfalk. Udemy course.

Learn React - The Complete Guide to Master React in 2018, Hichem Med. Udemy course.

How does Access-Control-Allow-Origin header work?. https://stackoverflow.com/questions/10636611/how-does-access-control-allow-origin-header-work

Access-Control-Allow-Origin. https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Access-Control-Allow-Origin

Access-Control-Allow-Origin: Dealing with CORS Errors in React and Express, Dave Ceddia, July 11, 2018. https://daveceddia.com/access-control-allow-origin-cors-errors-in-react-express/

How to fix Access-Control-Allow-Origin (CORS origin) issue for your HTTPS enabled WordPress Site and MaxCDN. https://crunchify.com/how-to-fix-access-control-allow-origin-issue-for-your-https-enabled-wordpress-site-and-maxcdn/

A Complete Guide to Flexbox, March 14, 2019. https://css-tricks.com/snippets/css/a-guide-to-flexbox/