Fróðskaparsetur Føroya

Hagstovan

Bachelor Project Formulation

Create a new modern and responsive user interface for the Faroese statbank

Skapa eitt nýtt nútíðar nýslumarkamót fyri Hagtalsgrunnin

Author
Helgi Poulsen

Supervisor
Kári Holm Johannesen

August 6, 2019

Introduction

PXWEB

Statistics Faroe Islands is build on *PXWEB*. *PXWEB* is an API structure developed by Statistics Sweden together with other national statistical institutions, like Statistics Finland and Statistics Norway, to disseminate public statistics in a structured way. This enables downloading and usage of data from statistical agencies without using a web browser direct over HTTP/HTTPS.

The *PXWEB* *R* package connects any *PXWEB* API to *R* and hence facilitate the access, use and referencing of data from *PXWEB* APIs[1].

Statistics Faroe Islands, as well as other organizations use *PXWEB* to distribute hierarchical data.

Here is a list of a few data sets:

- Statistics Faroe Islands
- Statistics Sweden
- Statistics Finland

PXWEB API

The data in *PXWEB* APIs consists of a metadata part and a data part. Metadata is structured in a hierarchical node tree, where each node contains information about subnodes that are below it in the tree or, if the nodes are at the bottom of the tree structure, the data referenced by the node as well as what dimensions are available for the data at that subnode1.

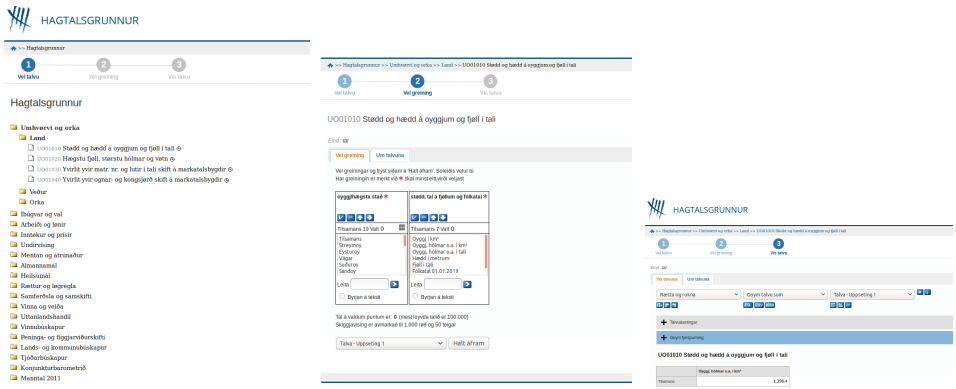
[1] PX-WEB API Interface for R

Problem

From user feedback of the Faroese statbank, we know that the Faroese statbank has lots of usability issues.

The UI[2] is not up to date with modern web standards and it does not contain any responsive modules, like visualization of the data. The UI is also legacy code[3].

This is a cross national problem since users in several countries, including all the Nordic countries, are facing the same usability problems.



(a) List (b) Selectors (c) Result

Figure 1: User Interface (Faroese Statbank)

If we start with figure11, we can see that the UI is not up to standard with modern web applications. The UI needs a new modern look.

As it is now, the user has to go through 3 pages to get to the result. This should be unnecessary. So to give the user a better experience, all 3 pages should be fit on 1 page.

[2]User Interface

[3] code without tests, which reflects the perspective of legacy code being difficult to work with in part due to a lack of automated regression tests

List1a The structure of the list is good and simple to understand. The real change here is to give it a modern style and try to make it more responsive.

Selectors1b The Selectors need a new styling and new functionality that give the user a instant result1c response. This means that the result changes when a selector changes.

Result1c The results only need new styling since the functionality of results is good.

1 Complex of unsolved problems

How can we utilize the API with modern web components, to improve the user experience and even rethink the way how it can be used.

2 Aim of project

The aim of the project is to build a new modern responsive UI prototype of the Faroese statbank as a concept.

If the project is a success, the concept will shown at the yearly $PXWEB$ conference[4], to show the other statistic organizations how $PXWEB$ can be utilized with modern components and not least give them inspiration of recreating their own modern statbank UI with this prototype as a template and share their experience.

3 Technology

To make the prototype a reality I will be using *JavaScript* as the main programming language. *JavaScript* is one of the worlds most used front-end programming language[5]. Addition to this, I will be using the *React*[6] library for building the user interface. *React* main maintainer is Facebook.

$Material-UI$[7] will also be used for the design.$Material-UI$ is a popular *React* UI Framework.

[4] held in Nov 2019 in Armenia
[5] Top 10 programming languages used in web development
[6] Reactjs
[7] Material-ui

TypeScript[8] is also being considered.

TypeScript is a strongly typed, object oriented, compiled language. It was designed by Anders Hejlsberg at Microsoft. *TypeScript* is both a language and a set of tools. *TypeScript* is a typed superset of *JavaScript* compiled to JavaScript. In other words, *TypeScript* is *JavaScript* plus some additional features[9].

Other programming languages where also considered:

.Net[10] Is a really good platform, but I find it best to use when working with databases and applications that use CRUD[11]. For example, a online store or accounting.

R Shiny[12] The package *RShiny* looks really interesting for visualizing data, but since one of the aims of the project is to give other statistic organizations inspiration of recreating their own modern statbank UI with this prototype as a template and share their experience. The best way forward is to use *JavaScript* that is probably worlds most used front-end programming language.

[8] TypeScript
[9] What is TypeScript
[11] Create,read,update,delete