PDF Research

Goal

The goal of this research is to find a solution to generating a module's PDF for frontend. The research should point out if the solution is adequate to the customers requirements or not.

Customer requirements

1. Should be possible to generate a module as PDF
2. Should be possible to generate a shortened module description
3. Should be possible to generate all modules at once
4. Should be possible to generate shortened versions of a module description at once
5. Generating PDF's is browser independent

Frontend PDF technologies researched

- jsPDF
- PDF.js
- PDFMake
- Native
- texlive.js

General Findings

Criteria 3 and 4 make things much harder on the frontend side. It's possible to generate all PDF's frontend. That would require retrieving a lot of data and calculations. All this effort which needs to be done frontend will increase stress on the frontend server which in turn might react slower.

Two approaches are found in creating a solution to the problem;

1. What you see is what you get
2. Template driven

What you see is what you get means that what is displayed as module is what will become the PDF. Template driven means that a template is used for creating a module.

There was also another problem found using these technologies which added complexity to create a PDF frontend. It is not possible to directly translate the html/css to pdf because angular code is not executed with these technologies.

Technology findings

jsPDF

jsPDF is a open source javascript framework created by James Hall. This library is ideal in many situations and is quite flexible. It meets all requirement. It could even use a what you see is what you get approach. However it has problems with translating angular code. You could still use this framework for a template driven approach.

The template driven approach needs to be calculated. Things like variable text sizes increases the complexity of this solution a lot. So altough it meets al requirements it takes an considerable amount of time to get a proper PDF.

Met Criteria

1. Should be possible to generate a module as PDF	X
2. Should be possible to generate a shortened module description	X
3. Should be possible to generate all modules at once	X
4. Should be possible to generate shortened versions of a module description at once	X
5. Generating PDF’s is browser independent	X
What you see is what you get approach	
Template driven approach	X

PDF.js

This technology does not fulfill any of the requirements. PDF.js is only usefull for rendering PDF’s properly on a browser. So altough this technology is not useful for this scenario. It could still be used in the future depending on the needs.

PDFMake

PDFMake Is another open source PDF library for javascript founded by Bartek Pampuch. This framework is very similair to jsPDF and faces the same difficulties as jsPDF. They use abit of a different approach. The main difference is that PDFMake is better documented.

1. Should be possible to generate a module as PDF	X
2. Should be possible to generate a shortened module description	X
3. Should be possible to generate all modules at once	X
4. Should be possible to generate shortened versions of a module description at once	X
5. Generating PDF's is browser independent	X
What you see is what you get approach	
Template driven approach	X

Native

Unlike the other technologies. Native is not a javascript framework but the capabilities of ones browser. With native technologies (css). It is possible to a PDF looking like a web page. It's possible to hide extra information and using checkboxes makes it easy to generate a shorthened module description. The problem of using this approach is that it is heavily browser dependant. Chrome and Edge seem to properly support this approach. Mozilla and safari have less options and can ignore certain parts of the CSS. Solving criteria 3 and 4 is also very difficult using this approach.

1. Should be possible to generate a module as PDF	X
2. Should be possible to generate a shortened module description	X
3. Should be possible to generate all modules at once	
4. Should be possible to generate shortened versions of a module description at once	
5. Generating PDF's is browser independent	
What you see is what you get approach	X
Template driven approach	

texlive.js

Texlive.js is yet another javascript framework that is capable of creating PDF documents frontend. Author of this framework is Manuel Scholling. Using this technology you don’t need to do the calculations to get everything at the right position as is the case with jsPDF and PDFMake. Using texlive.js you take on a latex approach of creating a PDF frontend.

1. Should be possible to generate a module as PDF	X
2. Should be possible to generate a shortened module description	X
3. Should be possible to generate all modules at once	X
4. Should be possible to generate shortened versions of a module description at once	X
5. Generating PDF’s is browser independent	X
What you see is what you get approach	
Template driven approach	X

Conclusion

As one might notice there are a lot of options to create PDF’s frontend. For the current problem however generating PDF’s frontend might not be the ideal solution with the given input of the customer. We started this research without having the knowledge of criteria 3, 4 and 5. If we had these criteria beforehand we might have chosen to generate PDF’s at the backend from the start.

For proof see log: 2017-12-12 Daily Meeting in which is implied the research already started (which it did on 11-12-2017). Later that day we asked the customer to discuss possible solutions and we wrote this down in the 2017-12-12 Daily Meeting log as well.