Politechnika Warszawska
Wydzia Elektroniki i Technik Informacyjnych
Instytut Informatyki

Academic year 2016/2017

Praca dyplomowa magisterska

Dennis Durairaj

Creating PDF Documents from Web Applications

Opiekun pracy:
Tytu Imi i Nazwisko

Ocena ....

....
Podpis Przewodniczcego
Komisji Egzaminu Dyplomowego

fotografia

Specjalno: Informatyka –
Inynieria oprogramowania
i systemy informacyjne

Data urodzenia: 1 stycznia 1980 r.

Data rozpoczcia studiw: 1 padziernika 2002 r.

yciorys

Nazywam si . . . .

. . . .
podpis studenta

Egzamin dyplomowy

Zoy egzamin dyplomowy w dn. . . . .

Z wynikiem . . . .

Oglny wynik studiw . . . .

Dodatkowe wnioski i uwagi Komisji . . . .

. . . .

Streszczenie

Praca ta prezentuje . . .

Sowa kluczowe: *sowa kluczowe.*

Abstract

Title: *Creating PDF Documents from Web Applications*

This thesis describes an in-depth research in handling creation of PDF documents through various web technologies in both the client-side as well as server-side.

Key words: *key words.*

Contents

1. Introduction

1.1. PDF - Portable Document Format

PDF was developed in the early 1990s as a way to share documents with people over different locations, which included formatting text as well as embedding inline images. Before the rise of the Internet and HTML, PDF was used mainly for publishing workflows.

Why Use PDF?

A well structured PDF will maintain the original text format, images, as well as the keep perfect layout of the document. PDF was mostly used by graphic designers and publishers for producing color page documents and designs. A PDF file can be shared, viewed, and printed by anyone using the freely available PDF reading softwares without depending on the type of the operating system, the original design application or fonts. However, with the changes happening in technology, nowadays PDF is used for any type data to be shared between users or applications. It is an open source file format specification and PDF is freely available to people who want to create tools for developing, viewing or manipulating PDF documents.

Why is PDF important?

Today from a user's stand-point, it is becoming increasingly easier to create PDF files as the process is become as simple as printing a document. To put it in other words, anything that can be done with a piece of paper can be done with a PDF. Offset printed documents are using the technoligy of PDF more frequently. Adding to mainstream usage is the fact that a large number of applications allow users to save, upload or download a document as a PDF, and you can also find a variety of PDF conversion softwares tools freely available. With the power to embed metadata in a PDF document, along with the use of password protection options and electronic signatures, PDF is also being used as a standard for data archiving. It may have not been the perfect solution in the beginning but with years of dedication and effort by the development team at Adobe, today a large number of people are turning to PDF as a solution for something no one thought of in the early 1900s.

Technical details about PDF

A PDF file is a 7-bit ASCII file, with the exception for certain elements that may have content in binary format. A PDF file begins with a header that contains a magic number and the version of the format. The format is a part of the COS ("Carousel" Object Structure) format. A COS tree file consists main of objects, of which there are eight types:

— Boolean values, representing true or false

— Numbers

— Strings, enclosed within parentheses ((...)), may contain 8-bit characters.

— Names, starting with a forward slash (/)

— Arrays, ordered collections of objects enclosed within square brackets ([...])

— Dictionaries, collections of objects indexed by Names enclosed within double pointy brackets

— Streams, usually containing large amounts of data, which can be compressed and binary

— The null object

1.2. Web Communication Methodologies

When facing frontend development, we start with the browser and the capabilities it offers. But with backend development, the field is much wider.

Firstly, we need to think about the language, with a plethora of server-side language available today, it is sometimes a problem to choose one with each having its own advantages. The language you choose will determine the operating system to install in the server. A web framework is a set of tools that can be used to help ease the process of setting up the server, since it provides functions and methods for the developer to work with and instead of creating his own from scratch. Hence it solves many web development problems and provides a good structure to begin with. This accelerates a lot of the initial setup. We shall touch on some backend frameworks in the following sections.

Languages and Frameworks

a. **PHP**

PHP was, and arguably still is one of the most popular languages for web development. It comes pre-installed with almost all hosting services. PHP has a syntax similar to C and Java, so coming from these languages makes it easier to work with PHP.

PHP started as a procedural language, then introduced object orientation in version 4, and with version 5 it finally became a true object oriented language. Version 7 brought more features and improvements to the language.

PHP stands for Hypertext Preprocessor (it's a recursive acronym). It is used to develop dynamic and interactive content on the web and it is often used along with the Apache web server.

— Code written in PHP is not executed but interpreted at the server during runtime instead of being compiled by a computer's processor which reduces the stress on the client-side processing.

— It is used to interact with content on webpages

— PHP is server-side as opposed to client-side, unlike some other web languages. When view the source code of a web page, you will observe that it may have been written in different languages including HTML and Javascript. However, the source code you will be seeing is client-side code i.e. the code that executes on the browser. A webpage written using PHP will not display the PHP source code used to create the page when you try to view the source code of the page. Even though PHP executes on the server and not the browser, the PHP files will be returned to the browser as plain HTML.

b. **Python - Django Framework**

Python is considered to be one of the most flexible and dynamic programming languages. It can be used for developing a wide variety of software applications. Python is further considered to be the best programming language for developing scientific applications and applications that are required to process a huge amount of data. Thus, the programming provides a much higher level of flexibility than PHP. Python is currently one of the most secure and robust programming languages. It security features make it one of the best languages for writing complex and mission-critical software applications. Python was designed as a general-purpose programming language. The developers have to use add-on modules to easily write web applications in Python. They can still use a variety of third-party modules to use Python effectively for web development.

Its well tested, Google chose it to develop their services, and thats a good thing.

The most popular framework for Python is **Django**[1]. Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support.

i. **Complete**

   Django follows the "Batteries included" philosophy and provides almost everything developers might want to do "out of the box". Because everything you need is part of the one "product", it all works seamlessly together, follows consistent design principles, and has extensive and up-to-date documentation.

ii. **Versatile**

   Django can be (and has been) used to build almost any type of website from content management systems and wikis, through to social networks and news sites. It can work with any client-side framework, and can deliver content in almost any format (including HTML, RSS feeds, JSON, XML, etc). The site you are currently reading is based on Django!

   Internally, while it provides choices for almost any functionality you might want (e.g. several popular databases, templating engines, etc.), it can also be extended to use other components if needed.

iii. **Secure**

   Django helps developers avoid many common security mistakes by providing a framework that has been engineered to "do the right things" to protect the website automatically. For example, Django provides a secure way to manage user accounts and passwords, avoiding common mistakes like putting session information in cookies where it is vulnerable (instead cookies just contain a key, and the actual data is stored in the database) or directly storing passwords rather than a password hash.

   A password hash is a fixed-length value created by sending the password through a cryptographic hash function. Django can check if an entered password is correct by running it through the hash function and comparing the output to the stored hash value. However due to the "one-way" nature of the function, even if a stored hash value is compromised it is hard for an attacker to work out the original password.

   Django enables protection against many vulnerabilities by default, including SQL injection, cross-site scripting, cross-site request forgery and clickjacking (see Website security for more details of such attacks).

iv. **Scalable**

   Django uses a component-based shared-nothing architecture (each part of the architecture is independent of the others, and can hence be replaced or changed if needed). Having a clear separation between the different parts means that it can scale for increased traffic by adding hardware at any level: caching servers, database servers, or application servers. Some of the

[1] https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction

busiest sites have successfully scaled Django to meet their demands (e.g. Instagram and Disqus, to name just two).

v. **Maintainable**

Django code is written using design principles and patterns that encourage the creation of maintainable and reusable code. In particular, it makes use of the Don't Repeat Yourself (DRY) principle so there is no unnecessary duplication, reducing the amount of code. Django also promotes the grouping of related functionality into reusable "applications" and, at a lower level, groups related code into modules (along the lines of the Model View Controller (MVC) pattern).

vi. **Portable**

Django is written in Python, which runs on many platforms. That means that you are not tied to any particular server platform, and can run your applications on many flavours of Linux, Windows, and Mac OS X. Furthermore, Django is well-supported by many web hosting providers, who often provide specific infrastructure and documentation for hosting Django sites.

c. **C# - ASP .NET**

ASP .NET[2] is an open-source server-side web application framework designed for web development to produce dynamic web pages. It was developed by Microsoft to allow programmers to build dynamic web sites, web applications and web services.

It was first released in January 2002 with version 1.0 of the .NET Framework, and is the successor to Microsoft's Active Server Pages (ASP) technology. ASP.NET is built on the Common Language Runtime (CLR), allowing programmers to write ASP.NET code using any supported .NET language.

ASP.NET's successor is ASP.NET Core. It is a re-implementation of ASP.NET as a modular web framework, together with other frameworks like Entity Framework. The new framework uses the new open-source .NET Compiler Platform and is cross platform. ASP.NET MVC, ASP.NET Web API, and ASP.NET Web Pages (a platform using only Razor pages) have merged into a unified MVC 6.

ASP.NET is a web development platform, which provides a programming model, a comprehensive software infrastructure and various services required to build up robust web applications for PC, as well as mobile devices.

ASP.NET works on top of the HTTP protocol, and uses the HTTP commands and policies to set a browser-to-server bilateral communication and cooperation.

ASP.NET is a part of Microsoft .Net platform. ASP.NET applications are compiled codes, written using the extensible and reusable components or objects present in .NET framework. These codes can use the entire hierarchy of classes in .NET framework.

The ASP.NET application codes can be written in any of the following languages:

C# Visual Basic.Net Jscript J# ASP.NET is used to produce interactive, data-driven web applications over the internet. It consists of a large number of controls such as text boxes, buttons, and labels for assembling, configuring, and manipulating code to create HTML pages.

d. **JavaScript - NodeJS**

[2] https://nodejs.org/en/about/

Node.js is an open-source, cross-platform JavaScript runtime environment for developing a diverse variety of tools and applications. Although Node.js is not a JavaScript framework,[4] many of its basic modules are written in JavaScript, and developers can write new modules in JavaScript. The runtime e nvironment interprets JavaScript using Google's V8 JavaScript engine.

Node.js has an event-driven architecture capable of asynchronous I/O. These design choices aim to optimize throughput and scalability in Web applications with many input/output operations, as well as for real-time Web applications (e.g., real-time communication programs and browser games).

The Node.js distributed development project, governed by the Node.js Foundation, is facilitated by the Linux Foundation's Collaborative Projects program.

Node.js is a server-side platform built on Google Chrome's JavaScript Engine (V8 Engine). Node.js was developed by Ryan Dahl in 2009 and its latest version is v0.10.36. The definition of Node.js as supplied by its official **documentation**[3] is as follows

"Node.js is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices."

Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux.

Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

[3] https://nodejs.org/en/about/

1.2.1. An Overview

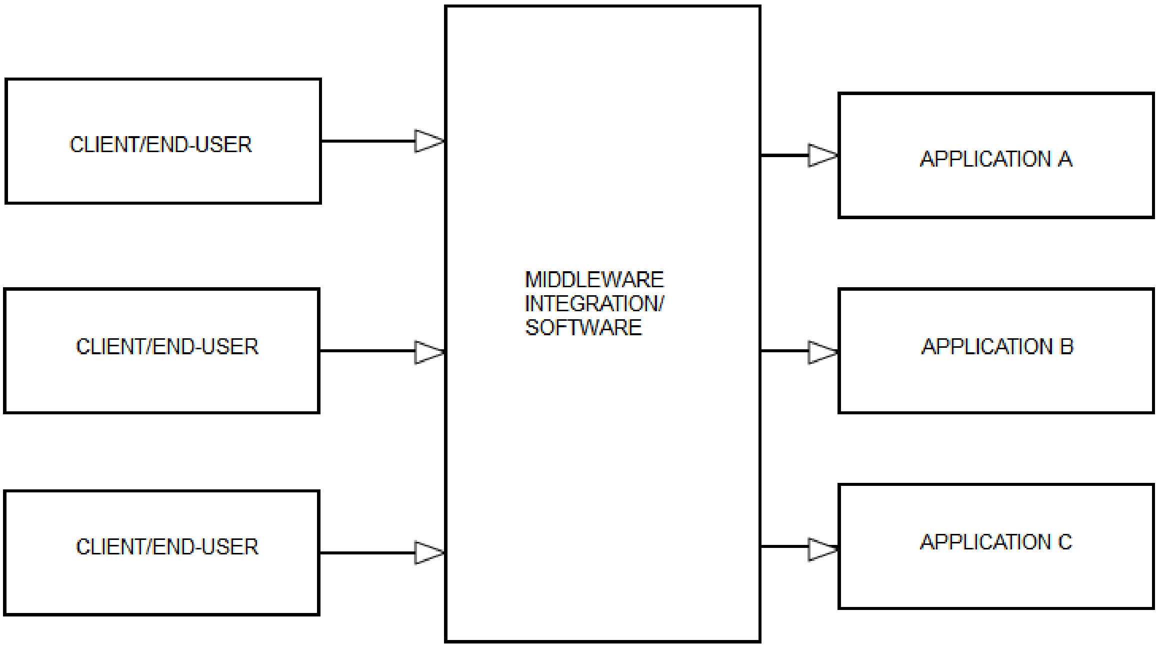

Figure 1.1. test

1.3. Existing methods to convert data to PDF

1.3.1. Frontend methods

1.3.2. Backend methods

2. Creating PDF documents in front-end

2.1. Frontend technologies

Front-end web development (also known as client-side development) is the practice of producing websites or Web Applications using HTML, CSS and JavaScript so that an end user can see and interact with them. The challenges that come with front end development is that the tools and techniques used to create the front end of a website change constantly and so the developer needs to constantly be aware of how the field is developing.

2.2. HTML5, CSS and JavaScript

a. **What is HTML5?**

HTML, or HyperText Markup Language, is the core element of the Internet. Its the language used to describe how a webpage should be structured. However, HTML on its own is bland because it can only display static pages; so in order to meet the increasing demand for more impressive web features, HTML with plugins like CSS, Flash, Java, Silverlight, etc. create the modern day Internet that we utilise in our daily life.

However, it has become something of a mess since different browsers have implemented these features in their own ways. With the advent of HTML5, it is meant to solve HTMLs big problems for a cleaner and more efficient web.

b. **Cascading Stylesheets**

CSS is a styling language that defines layout and design of HTML documents. For example, CSS covers margins, lines, width, background images, fonts, colours, height, advanced positioning and many other things. HTML can be used (or misused) to add layout to webpages. However, CSS offers additional options and is more accurate and sophisticated. CSS is completely supported by all browsers today.

In order to use HTML and CSS together, you use HTML to describe the body of the document and CSS to specify the document's layout, visual appearance, style,etc. not its content.

c. **JavaScript - The language of the Web**

JavaScript is a powerful client-side scripting language which has recently found its way into server-side scripting as well thanks to node.js which is a JavaScript based server-side programming framework. JavaScript is used mainly for improving the interaction of a client with the webpage. In other words, you can make your webpage more interactive and user-friendly, with the help of JavaScript. Today, from the browser to the server, JavaScript proves to be one of the most popular and versatile languages powering the modern web.

Client-side JavaScript makes use of the core language by providing special objects to control a browser using its Document Object Model (DOM). For instance, client-side extensions allow front-end developers to respond to user events such as clicks, scrolls, page navigation etc.

JavaScript has made its way to the server-side fairly recently. It extends the core language by supplying objects used to run JavaScript on a server. For instance, extensions on the server-side allow an application to provide passing of information from one invocation to another of the application, communicate with a database, or perform file manipulations on a server.

2.3. Built-in browser PDF converter

Modern browsers have the capability of saving webpages as a PDF file. This can come in hand in situations where one would like to save certain details. However, this is a read-only method of creating PDF files from webpages and involves no special application or code to achieve this feature. This method fails to save form details which requires more complex procedures since we need to handle validation to produce a valid form/document for the user.