|  |
| --- |
| Mr.Pres  핵심 기술 소개 |
| 한성대학교 컴퓨터공학과 설계 프로젝트 |
| FrontFloor  엄두성 / 김강성 / 한종빈 / 조이슬 |

HTML

하이퍼텍스트 마크업 언어(Hyper Text Markup Language) 라는 의미의 웹 페이지를 위한 마크업 언어이다. HTML은 제목, 단락, 목록 등과 같은 본문을 위한 구조적 의미를 나타내는 것뿐만 아니라 링크, 인용과 그 밖의 항목으로 구조적 문서를 만들 수 있는 방법을 제공한다. 그리고 이미지와 객체를 내장하고 대화형 양식을 생성하는 데 사용될 수 있다. HTML은 웹 페이지 콘텐츠 안의 꺾쇠 괄호에 둘러싸인 "태그"로 되어있는 HTML 요소 형태로 작성한다.

역사

1995년 11월 24일 HTML 2.0이 IETF의 RFC 1866로 발표되었다.   
1997년 1월 HTML 3.2 가 W3C 권고안으로 발표되었다. 이것은 1996년 9월에 폐쇄한 HTML 작업 그룹인 IETF와 같이 W3C에 의하여 배타적으로 개발되고 표준화된 첫 번째 버전이다. HTML 3.2에서 여러 종류의 겹치는 확장 수학 수식을 완전히 제외 시키고 넷스케이프의 비주얼 마크업 태그의 대다수를 채택했다. 넷스케이프의 블링크 요소와 마이크로소프트의 마퀴 요소는 두 회사의 상호 협의 하에 누락시켰다.[13] HTML에서 수학적 수식과 비슷한 마크업은 이후 MathML이 나올 때까지 14개월 동안 표준이 아니었다.  
1997년 12월 HTML 4.0이 W3C 권고안으로 출시되었다. HTML 4.0은 세 가지 문서 형태를 제공한다:  
 Strict(엄격): 사용이 중지된 요소를 금지한다.  
 Transitional(변이): 사용이 중지된 요소를 허용한다.  
 Frameset(프레임셋): 대부분 프레임 관련 요소에서만 허용한다.  
처음에 코드네임 "쿠가" 에서 HTML 4.0은 다양한 브라우저 특정 요소의 종류와 속성을 채택하였지만 동시에 스타일시트에 찬성하는 세력의 반대로 넷스케이프의 시각적 마크업 기능을 단계적으로 제거하기로 했다. HTML 4는 ISO 8879 - SGML에 따른 SGML 응용 프로그래밍 언어다.  
1998년 4월 HTML 4.0 버전이 증가되지 않고 조금 수정되어 재발표되었다.  
1999년 12월 HTML 4.01이 W3C 권고안으로 출시되었다. HTML 4.0과 마찬가지로 세 가지 문서 형태를 제공한다. 그리고 2001년 5월 12일에 그것에 대한 최종 오류 수정 사항 이 발표되었다.  
2000년 5월 ISO/IEC 15445:2000 ("ISO HTML", HTML 4.01 Strict 기반)이 ISO/IEC 국제 표준과 함께 출시되었다. ISO에서 이 표준안은 ISO/IEC JTC1/SC34 (ISO/IEC 공동 기술 위원회 1, 소위원회 34 - 문서 표현과 처리 언어)의 범위에서 떨어져 나왔다.  
2008년 1월HTML 5[26] 초안 작업은 W3C에 의해 발표되었다.비록 문법은 SGML의 문법과 밀접하게 연관되어 있지만, HTML 5는 난해한 SGML 문법과 호환되지 않는다. 그리고 HTML 5를 "html" 시리즈라고 명시적으로 정의되었다. 이 어플리케이션은 웹 서버에서 하이퍼텍스트 전송 프로토콜(HTTP)을 이용하는 웹 브라우저로 전송되는 HTML 문서를 이용한다. 그리고 웹 서버는 HTML 이외에도 이미지나 소리, 그리고 기타 내용물을 서비스하는 데 사용된다. 브라우저가 받는 각 문서를 어떻게 다른지 알려주게 하기 위해, 다른 정보가 문서와 함께 전송한다.

웹 서버에서 하이퍼텍스트 전송 프로토콜(HTTP)을 이용하는 웹 브라우저로 전송되는 HTML 문서를 이용하여 클라이언트 프로그램을 보인다. 그러나 HTTP는 HTML 이외에도 이미지나 소리, 그리고 기타 내용물을 서비스한다. 브라우저가 받는 각 문서를 어떻게 다른지 알려주게 하기 위해, 다른 정보가 문서와 함께 전송한다.

CSS

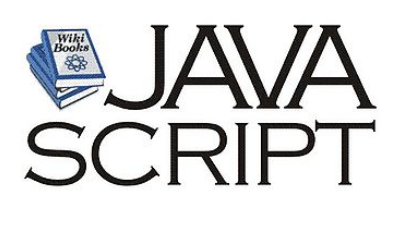
CSS 또는 캐스케이딩 스타일 시트(Cascading Style Sheets)는 마크업 언어가 실제 표시되는 방법을 기술하는 언어로, HTML과 XHTML에 주로 쓰이며, XML에서도 사용할 수 있다. W3C의 표준이며, 레이아웃과 스타일을 정의할 때의 자유도가 높다.  
 W3C CSS3 로드맵은 요약과 도입부를 제공하고 있다. 전체가 모듈화 되어 사용자 에이전트가 모듈에 대한 모든 모듈을 지원하지 않거나 자유롭게 선택할 수 있도록 하고 있으며, 다른 세로 글쓰기와 HTML 이외의 규격에 까지 관여하는 내용으로 되어 있다.

웹 문서의 전반적인 스타일을 미리 저장해 둔 스타일시트이다. 이를 사용하면 문서 전체의 일관성을 유지할 수 있고, 문서의 스타일 지정의 통합할 수 있다. 따라서 보다 풍부한 디자인으로 웹 페이지를 설계할 수 있고, 글자의 크기와 모양, 배경 색상, 줄 간격, 배열 위치 등을 자유롭게 선택하거나 변경할 수 있으며 문서 디자인의 유지 · 보수 또한 간편하게 할 수 있다.

bootstrap

부트스트랩은 트위터에 근무하는 두 명의 엔지니어 (Mark Otto와 Jacob Thornton)가 개발한 가볍고 간단한 오픈소스기반 (아파치2.0라이센스) 의 HTML5, CSS3 UI 프레임워크입니다. HTML5, CSS3, Javascript 로 구성되어 있고, 가볍게 트위터 디자인 비슷한 UI 를 빠르게 구현해낼 수 있는 각종 콤포넌트들을 제공해 주고 있습니다.

JavaScript

객체 기반의 스크립트 프로그래밍 언어이다. 이 언어는 웹 사이트에서 많이 사용하며, 다른 응용프로그램의 내장 객체에도 접근할 수 있는 기능을 가지고 있다. 자바스크립트는 본래 넷스케이프 커뮤니케이션즈 코퍼레이션의 브렌던 아이크(Brendan Eich)가 처음에는 모카(Mocha)라는 이름으로, 나중에는 라이브스크립트(LiveScript)라는 이름으로 개발하였으며, 최종적으로 자바스크립트가 되었다. 자바스크립트가 썬 마이크로시스템즈의 자바와 구문(syntax)이 유사한 점도 있지만, 이는 사실 두 언어 모두 C 언어의 기본 구문을 바탕했기 때문이고, 자바와 자바스크립트는 직접적인 관련성이 없다. 이름과 구문 외에는 자바보다 셀프와 유사성이 많다.

웹 브라우저에서 실행하는 스크립트 언어를 기술한다. 하이퍼텍스트 생성 언어(HTML) 문서를 작성하는 수준의 사용자가 사용하는 것을 주안점으로 하여 자바의 언어 규격으로부터 정수형이나 문자열 형 등과 같은 변수의 형을 생략하거나 새로운 클래스 정의를 할 수 없도록 하였다. 스크립트는 HTML 문서 속에 직접 기술하며, ‘script’라는 태그를 사용한다.

jQuery

jQuery(제이쿼리)는 브라우저 호환성이 있는 HTML 속 자바스크립트 라이브러리이며 클라이언트 사이드 스크립트 언어를 단순화 할 수 있도록 설계되었다. 존 레식이 2006년 뉴욕 시 바캠프(Barcamp NYC)에서 공식적으로 소개하였다. jQuery는 오늘날 가장 인기있는 자바스크립트 라이브러리 중 하나이다.  
 jQuery는 MIT 라이선스와 GNU 일반 공중 사용 허가서v2의 듀얼 라이선스를 가진 자유 오픈 소프트웨어이다. jQuery의 문법은 코드 보기, 문서 객체 모델 찾기, 애니메이션 만들기, 이벤트 제어, Ajax 개발을 쉽게 할 수 있도록 디자인되었다. 또한, jQuery는 개발자가 플러그인을 개발할 수 있는 기능을 제공한다.  
 마이크로소프트와 노키아는 자사 플랫폼에 jQuery를 포함하는 계획을 발표한 바 있다. 마이크로소프트는 비주얼스튜디오의 ASP.NET AJAX 프레임워크와 ASP.NET MVC 프레임워크에 적용했고, 노키아는 자사의 런타임 웹 위젯 개발 플랫폼에 통합하였다. 또한, jQuery는 미디어위키에도 1.16 버전부터 사용되고 있다.

jQuery는 다음과 같은 기능을 위해 사용한다:

DOM 요소 선택 기능.   
DOM 탐색 및 수정 (CSS 1-3 지원)  
CSS 셀렉터에 기반한 DOM 조작. 노드 요소 및 노드 속성(아이디 및 클래스)을 셀렉터 생성을 위한 기준으로 사용.  
이벤트  
특수효과 및 애니메이션  
AJAX  
JSON 파싱  
플러그인을 통한 확장성  
유틸리티  
호환성 메소드 (inArray(), each() 함수 등)  
멀티브라우저 지원 (크로스브라우저와는 다름)

Node.js

서버사이드 자바스크립트 (V8 엔진으로 컴파일) 프레임워크

이벤트 기반으로 동작 하며, Non-Blocking I/O 방식

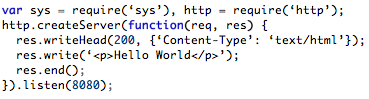
목적: 확장 가능한 네트워크 프로그램들을 간단하게 만드는 것

현재(2014.04.04) 버전 : v0.10.24

C와 C++로 만들어짐

단 하나의 스레드가 이벤트 루프를 구동

Node.js의 장점

가벼움 (아래의 코드로 간단한 http server 를 돌릴 수 있음)  


빠름

동시 접속 수에 따른 초당 요청 처리 수가 멀티쓰레딩에 비해 월등함.

100명의 접속자, 1메가 바이트 응답  
node 822 req/sec  
nginx 708 req/sec  
thin 85 req/sec  
mongrel 4 req/sec

많은 모듈: 현재 53919개(npm으로 공개 된 것들) – 개발 용이

자바스크립트로 다 할 수 있음: FrontEnd와 BackEnd가 Javascript로 대동단결

Node.js 의 단점

CPU 개수, Memory 용량을 높여도 성능이 크게 향상되지 않는다.  
 (노드 인스턴스를 여러 개 실행함으로써 해결 가능)

이벤트 구동 방식 프로그래밍(callback style)은 구현하기 쉽지 않음

실수로 동기적 코드를 이용하는 경우 치명적인 성능 저하를 유발할 수 있다.

아직 0.10.24이다 (불안정)

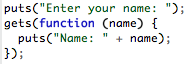
그동안 왜 사용하지 않았나?

문화적

event 방식 구현이 익숙치 않다.

Blocking (동기)  


vs

Nonblocking (비동기)  


인프라적

C는 익명함수도 없고 Closure도 없다. Callback도 만들기 어려움

대부분의 library들이 non-blocking i/o를 지원하지 않음

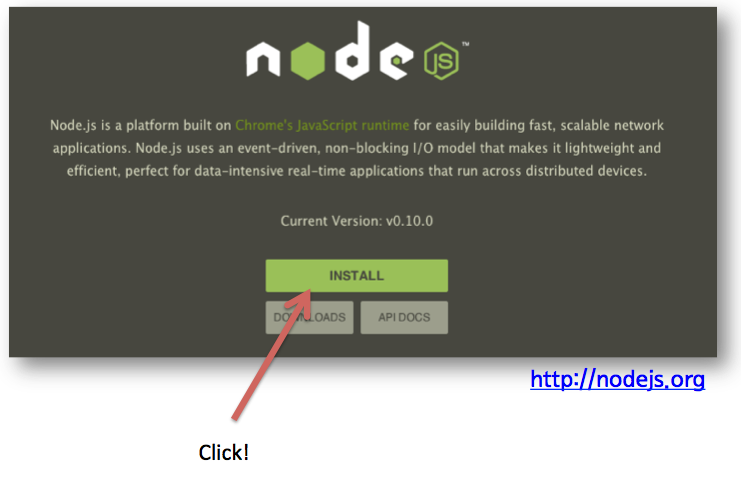
POSIX에서 비동기 파일 입출력을 이용할 수 없음

어디서 사용하나?

Storify  
Cloude IDE9  
Linked in: 전체 서버 스택이 모두 node 로 되어있음. 확장성/성능적 이점 때문에 사용  
Ebay & Paypal

Node.js 시작하기

Node.js 다운로드 및 설치



HTTP Server 만들기

Method 1: node.js http 모듈을 이용

var http = require('http');

var server = http.createServer( function(req, res) {

var body = 'Hello, World!';

res.writeHead(200, {

'Content-Length': body.length,

'Content-Type': 'text/plain'

});

res.end(body);

} );

server.listen(3000);

Method 2: Express, etc. npm 모듈을 이용

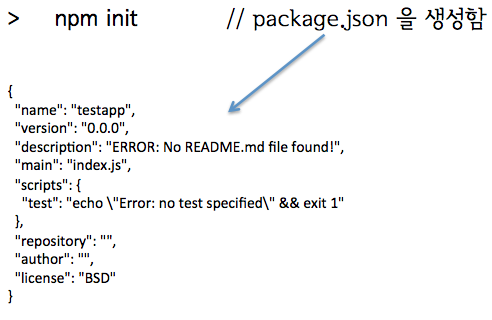
$ npm install -g express

$ express myapp

$ cd myapp && npm install

$ node app

Node 모듈을 시작하는 방법



필요한 모듈 가져다 쓰기

필요한 모듈 찾기: <https://npmjs.org>

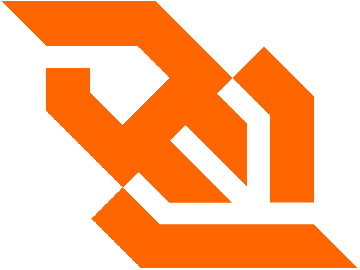
모듈 깔기: npm install {module name} {--save}

앱에 모듈 가져오기: var module = require(‘module name’)

적절한 모듈들

express – 웹 프레임워크

socket.io – 웹 소켓

Web Socket

* 자바스크립트의 WebSocket 객체를 제공.
* 웹 서버와 웹 브라우저가 지속적으로 연결된 TCP 라인을 통해 실시간으로 데이터를 주고 받을 수 있도록 하는 HTML5의 새로운 기술이다. (따라서, 웹 브라우저와 서버 간의 연결지향, 전이중 통신을 지원할 수 있다.)
* 웹 브라우저 사용자의 요청에 따른 또는 서버에서 발생한 이벤트를 사용자에게 통지하기 위한 메시지 교환 방식 사용한다. (실시간 이벤트를 즉시 받을 수 있다.)
* 일종의 PUSH 기술이다.
* 일반 통신: ws, 보안 통신: wss 프로토콜을 사용한다. (기본 포트 http,https와 동일한 80,443사용)
* WebSocket 속성

|  |  |
| --- | --- |
| 속성/메서드 | 설명 |
| url | 웹 소켓 프로토콜을(ws, wss) 사용해 연결한 서버의 URL 주소이다. |
| readyState | 웹 소켓 프로토콜을 사용해 서버와의 접속 상태를 알려주는 정수 속성 값으로, 다음 값 중 하나이다.  0 : 아직 연결이 되어 있지 않은 상태(CONNECTING)  1 : 메시지 교환이 가능한 상태(OPEN)  2 : 연결 해제 절차를 수행 중인 상태(CLOSING)  3 : 다시 연결이 불가능한 연결 해제 상태(CLOSED) |
| bufferedAmount | send 메서드에 의해 전송 버퍼에 놓여 전송 대기 중인 데이터의 바이트 수이다. |
| protocol | 서버의 요청에 의해 실행되는 서브 프로토콜의 이름이다. 서버와의 서브 프로토콜 협상이 필요하기 때문에 보통 사용하지 않는다. |
| send | 테이터를 서버에 전송하기 위한 메서드이다, |
| close | 연결 상태를 해제하기 위한 메서드이다. |
| onopen | 서버와의 웹 소켓 연결을 알리는 이벤트 핸들러이다. |
| onmessage | 서버로부터 전달된 데이터의 수신을 알리는 이벤트 핸들러이다, |
| onerror | 에러 발생 시 실행되는 이벤트 핸들러이다. |
| onclose | 연결 해제 시 실행되는 이벤트 핸들러이다. |

Web Socket Pres

Your web application has data that must flow bi-directional simultaneously.

Your web application must scale to large numbers of concurrent users.

Your web application must extend TCP-based protocols to the browser.

Your web application developers need an API that is easy to use.

Your web application must extend SOA over the Web and in the Cloud.

기존의 방식

i. 폴링(polling)

데이터 수신을 위해 서버가 클라이언트에게 전송해 주는 푸시(push)방식이 아니라 클라이언트가 서버에게 요청하는 방식이다.

ii. Comet(Long-Polling)

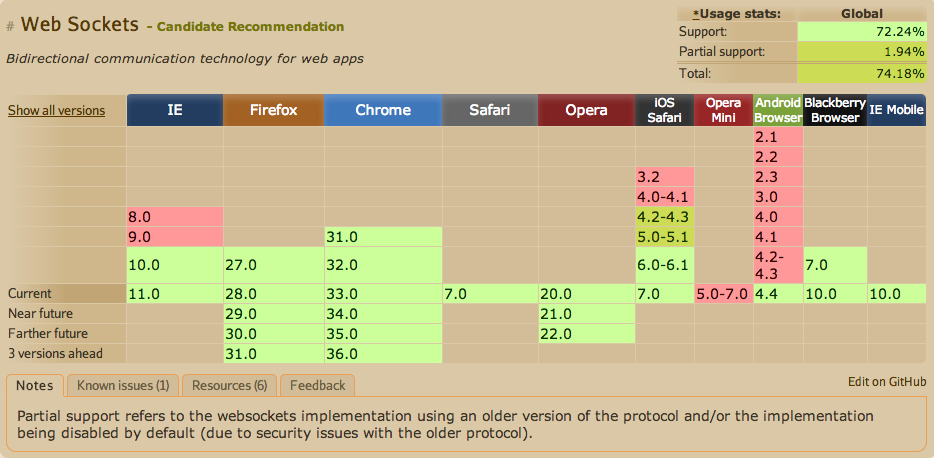
서버 데이터 수신 후 재 요청이 가능해져 불필요한 반복 요청의 비효율성은 개선하였다.

무의미한 반복 요청을 피하기 위한 연결유지 기법이 적용되었지만 일정 시간 이후에는 연결을 종료하고 다시 연결해야 한다

웹 브라우저의 주기적 요청에 의해 처리되는 서버 전송 이벤트 방식에 비해 더 빠르게 실시간 이벤트를 전달받아 처리될 수 있지만, 접속 요청을 대기하고 실시간 이벤트 메시지를 전송하는 서버 프로그램의 오버로드가 발생한다.

웹 소켓 작동

웹 브라우저 웹소켓 지원현황

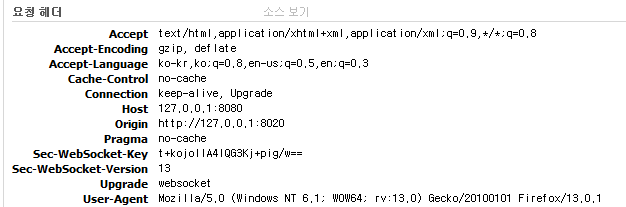


웹소켓 작동 방법: 서버로 연결하고, 서버에서 전해주는 이벤트를 지켜보다가 적절하게 반응.

1. 클라이언트의 TCP/IP 접속 요청에 의해 서버가 접속을 수락

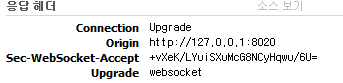
2. 핸드셰이크 요청

클라이언트는 HTML 프로토콜에서 처음 보내는 패킷과 유사한 패킷을 서버에 처음으로 전달한다.



3. 핸드셰이크 응답

서버의 핸드셰이크 요청 패킷에 따라 응답 패킷을 전송한다.



4. 데이터 교환

메시지 전달



<script>

var wSocket = new WebSocket("ws://127.0.0.1:8080");

wSocket.onmessage = function(e){ alert(e.data); }

wSocket.onopen = function(e){ alert("서버 연결 완료"); }

wSocket.onclose = function(e){ alert("서버 연결 종료"); }

function send(){ //서버로 데이터를 전송하는 메서드

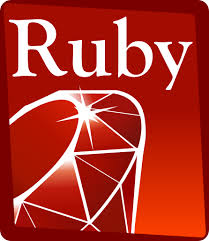
wSocket.send("Hello");

}

</script>

클라이언트 코드 만으로 데모는 실행되지 않는다. 별도의 서버가 필요.

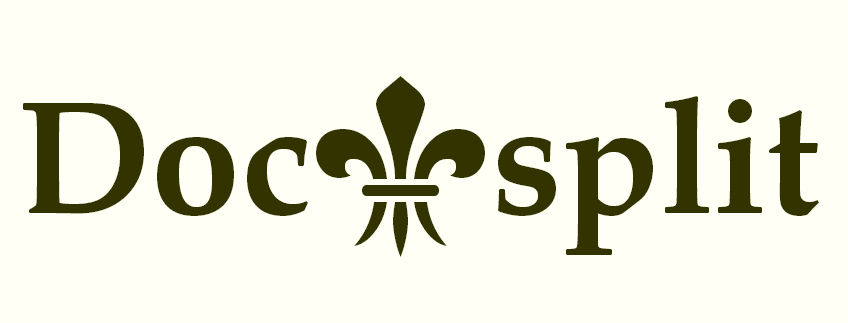
Ruby, Ruby on Rails

루비(Ruby)는 마츠모토 유키히로(松本行弘)가 개발한 동적 객체 지향 스크립트 프로그래밍 언어이다. 루비는 순수 객체 지향 언어이다. 루비에서의 정수나 문자열 등을 포함한 데이터 형식 등 모든 것이 객체이다.  
기능에는 클래스 정의, 가비지 컬렉션, 강력한 정규 표현식 처리, 다중 스레드, 예외 처리, 반복, 클로저, Mixin, 연산자 오버로드 등이 있다. 구문은 ALGOL계를 계승하면서 가독성을 중시하고 있다.  
오랫동안 언어 사양이 명문화되지 않고 마츠모토의 구현 언어 사양에 준하는 것으로 간주되어 왔지만, 2010년 6월 RubySpec으로 언어 사양을 책정 중이다.  
루비의 개발자 마츠모토 유키히로는 루비가 스트레스가 없는 쉬운 프로그래밍이고, 루비는 우리의 삶을 반영해야 하며, 프로그래밍 언어는 (컴퓨터가 아닌) 인간 중심으로 설계되어야 한다고 했다.  
루비는 이식성이 매우 높다. 대부분 GNU 리눅스에서 개발되었지만 유닉스, 맥 OS X, 윈도우, 도스, BeOS, OS/2 등에서 동작한다.

루비 온 레일즈는 루비 언어로 작성된, MVC 패턴을 이용하는 오픈소스 웹 어플리케이션 프레임워크이다. 흔히 줄여서 레일즈라고 부르기도 한다.  
액티브 레코드 기능을 활용한 데이터베이스 연동이나, 루비의 gem 기능을 활용해 쉽고 빠르게 추가적인 라이브러리/프레임워크 환경을 구축할 수 있는 등, 개발 속도를 대폭 높여주는 것으로 인기를 끌고 있다.  
웹 어플리케이션을 개발하는 데 있어 많은 부분을 자동화 및 단순화시켜 최대한 빠르게 웹 어플리케이션을 제작할 수 있다는 장점이 있다.

Why, How

루비와 루비 온 레일즈를 우리 프로젝트에 사용한 이유는 빠른 생산성과 Docsplit 라이브러리 사용에 있다.  
루비 온 레일즈의 가장 큰 장점은 단연 빠른 구현 속도에 있다. 우리 프로젝트에서 변환 서버가 맡는 역할은 매우 중요한 것이지만, 실제로 그 기능의 가지수가 많지는 않다. 처음부터 공을 들이기 보다는 기본적인 기능을 빠르게 뽑아 내고 실질적인 웹 어플리케이션 제작에 집중을 하는 것이 낫다.  
또한 여러가지 프레젠테이션 파일을 이미지로 변환하는 여러 라이브러리들을 살펴봤지만, 범용성과 구현 속도, 사용성 등을 고려했을 때 Docsplit만한 라이브러리가 없었다. 이 Docsplit 은 루비 라이브러리이고, 루비와 루비 온 레일즈에서 손쉽게 설정해 사용할 수 있다.

Docsplit

Docsplit은 ppt, pptx, pdf, odp, doc 등 많은 종류의 문서 파일을 이미지 혹은 pdf 형식으로 변환시켜주는 오픈소스 루비 라이브러리다.  
범용성, 구현 속도, 사용성 면에서 비슷한 기능의 다른 라이브러리들에 비해 우수하다.

Why, How

웹브라우저에서 프레젠테이션 파일을 보여 주기 위해서는 별도의 프로그램 설치가 필요한데, Docsplit을 통해 프레젠테이션 파일을 이미지 파일로 변환하면 별도의 프로그램 없이도 웹브라우저만을 통해 사용자에게 프레젠테이션을 보여줄 수 있다.  
 다른 프로그램(예를 들어, ppt/pptx 뷰어 프로그램)을 설치하게 하는 것보다 이미지로 변환하여 보여주는 것이 사용자 입장에서 덜 번거롭고 편하다. 또 특정 뷰어는 특정 형식의 파일을 지원하지 않는 등 문제가 있지만, Docsplit은 우리가 알고 있는 대부분의 문서에 대한 변환을 지원한다.

Resque

Resque는 백그라운드에서 실행되는 워커를 생성하고 실행하기 위한 루비 라이브러리다. 사용하기 위해서는 Redis의 설치가 필요하다.  
루비 혹은 루비 온 레일즈 어플리케이션에서 즉각적인 응답이 필요하지만 실질적으로 소요되는 시간이 많은 작업을 진행해야 할 경우 사용할 수 있다.  
루비는 Resque를 통해 워커를 만들어달라는 요청이 들어오는 순간 해당 요청을 Redis에 저장하고 사용자에게 응답을 함으로써 실행을 마친다. 이후 루비는 별도의 백그라운드 프로세스를 통해 Redis의 Multiple Queue를 지속적으로 검사하고, Resque를 통해 들어온 요청이 있다면 프로세스를 추가적으로 생성해 해당 작업을 수행한다.

Why

Docsplit이 파일 변환에 소요하는 시간이 긴 편은 아니지만, 그렇다고 일반적인 사용자가 참고 기다리기에 짧은 시간도 아니다. 따라서 우리는 사용자의 프레젠테이션 파일 업로드에는 즉각적인 응답을 반환하고, 별도의 프로세스로 프레젠테이션 파일 변환을 하기로 결정했다.  
Resque를 사용함으로서 사용자는 파일 변환 작업을 기다릴 필요가 없이, 하고자 하는 다른 작업을 자유롭게 진행할 수 있고, 파일 변환 작업은 백그라운드 워커에 의해 진행된다.

Redis

레디스(Redis)는 오픈 소스이며 네트워크화, 인메모리 데이터베이스, 선택적 영구성 키-값 데이터 저장소이다.  
데이터 모델은 외부층에서 레디스의 데이터 모델은 키를 값에 맵핑하는 자료구조의 dictionary 이다. 레디스 그리고 기타 구조화 된 스토리지 시스템 사이의 주요 차이점 중 하나는 레디스는 스트링 뿐만 아니라 자료구조의 Abstract Date Type도 지원을 한다.

Why, How

레디스는 인메모리 데이터베이스를 지원함으로서 빠른 속도를 가장 큰 장점으로 갖고 있다. 따라서 데이터베이스의 용도로도 쓰이지만 캐시 메모리로서의 용도로 사용될 때도 많다. Resque는 자신이 해야 할 작업의 순서 및 목록을 레디스에 캐시해두고 사용한다. 즉 레디스가 웹 어플리케이션의 메인 프로세스와 Resque의 백그라운드 워커 사이의 징검다리 역할을 해준다고 할 수 있다.  
Redis는 Resque가 작업해야 할 일의 목록을 큐 형식의 저장소에 저장해두었다가, Resque가 작업 목록을 요청할 때마다 진행해야할 작업을 건내주도록 구현되어 있다.

Google Drive API

구글 드라이브(Google Drive)는구글에서 제공하는 클라우드 기반 협업도구이자 파일저장/공유 서비스이다. 일반적인 대부분의 문서 파일을 볼 수 있고 여러 사용자와 공유가 가능하며, 구글 드라이브에서 지원하는 파일은 여러 유저의 동시 편집도 가능하다.  
구글 드라이브 API는 이러한 기능들을 개발자들이 사용할 수 있도록 구글에서 개방한 API이다. 사용자의 승인을 얻으면 구글 드라이브의 모든 기능을 프로그래밍 언어로서 컨트롤 할 수 있다.

Why, How

프레젠테이션 파일은 이미지를 비롯한 각종 멀티미디어 리소스들이 포함될 수 있고 따라서 용량도 매우 커질 수 있다. 프레젠테이션 서비스를 제공하기 위해서 변환된 이미지 파일은 변환 서버에서 관리하는 것이 당연하지만, 본래의 프레젠테이션 파일을 변환 서버에서 저장하고 있을 필요는 없다. 오히려 큰 용량 때문에 서버 공간이 낭비될 가능성이 크다.  
하지만 사용자 입장에서 원본 파일을 유지하고 서버에 저장하고 싶을 수도 있다. 따라서 사용할 수 있는 여러 클라우드 서비스를 조사한 결과, 구글 드라이브가 접근성, 사용성, 범용성 등에서 가장 좋은 서비스라는 결론을 내렸다.  
사용자가 프레젠테이션 파일을 업로드하면 변환 서버는 해당 파일을 잠시 저장해두고 이미지 파일로의 변환에 사용한 뒤에, 구글 드라이브 API를 통해 구글 드라이브 저장소에 업로드한 뒤, 변환 서버에서는 해당 파일을 지우게 된다.