

# 인수인계 문서

Web Service 및 Project Management에 관하여

개발부 김찬울 사원

2016년 7월 21일

---

# Table of Contents

Introduction	1.1
My Works	1.2
Recruit	1.3
Backup	1.4
Service Management	1.5
Communication	1.6
Internal	1.6.1
External	1.6.2
신입사원을 위한 세팅 가이드	1.6.3
Install Parallels Desktop on Mac	1.6.4
WebTex3D	1.7
Specification	1.7.1
How To Install	1.7.2
Youngwoo	1.7.3
stylefor.us	1.7.3.1
fabric.stylefor.us	1.7.3.2
shirts.stylefor.us	1.7.3.3
mobile.stylefor.us	1.7.3.4
Uniform	1.7.3.5
Yuka-Alpha	1.7.4
KEYUCA	1.7.4.1
Evegarden	1.7.4.2
Fukushin Dress	1.7.4.3
Sunregent	1.7.4.4
Royal	1.7.4.5
Elite	1.7.5
Specification	1.7.5.1
업무분담	1.7.5.2
Works	1.7.5.3
Texclub	1.8

---

---

Mail	1.8.1
Homepage	1.8.2
Install	1.8.3
Indexes	1.9
SNS Feed	1.9.1
Modal (Bootstrap)	1.9.2
My Favorite	1.9.3
Check Total Cost	1.9.4
Responsive Web	1.9.5
Print Cost Table from CSV Data	1.9.6
Stand Alone System (+ restart BAT file)	1.9.7
AJAX Loading bar	1.9.8
Export to PDF	1.9.9
Render Grid via Image Map	1.9.10
Print	1.9.11
Image Gallery	1.9.12
Not Using Hover on Mobile Device	1.9.13
Save Simulation Data on CSV	1.9.14
Search (KEYUCA)	1.9.15
Apache	1.9.16
Image Upload	1.9.17
Draw Text in Canvas	1.9.18
Mapping in Grid Group	1.9.19
Transfer Data to JoyCART	1.9.20
Multi-part Simulation	1.9.21

# 안녕하세요!

안녕하세요, 김찬울입니다.

이 문서는 Youngwoo CNI Web Part 직원 분들을 위한 업무 인수인계 문서의 종류입니다.

또한 웹문서의 수정 및 관리를 요구하실 경우 [studio@gar.am](mailto:studio@gar.am)으로 메일 주시면 빠르게 도와드릴 수 있도록 하겠습니다. 마지막으로 궁금하신 점이나 추가적인 정보를 요구하실 경우 이 또한 연락을 부탁드립니다.

아무쪼록, 더 멋진 개발자가 되기 위해 우리 화이팅 합시다 :)



## My Works

김찬울 사원의 주요 업무로 다음과 같은 내용들이 있습니다.

1. 개발팀에 오더되는 Web Project 제작 및 연구 [공동]
2. Texclub 서비스, 웹인증 서비스 관리 및 수정 [하영]
3. 하루에 두번 Webmaster Mail 확인 후 담당자에게 전달 [하영]
4. 프로젝트별 외부 담당자와의 커뮤니케이션 및 문서 작성 [진]
5. 프로젝트별 내부 담당자와의 기획 및 기능 설계 및 주간/팀장 회의 참여 [진]
6. Web 파트 지원자와의 면접 절차 진행 [공동]
7. Web Project의 Data 및 Database Backup 및 Project 설치를 위한 코드저장소, AWS, 도메인, 호스팅 관리 [하영]
8. Slack, Trello 회원 관리 및 신입사원을 위한 가이드 [하영]

# Recruit

이 Chapter는 개발 지원자 면접에 대한 내용을 포함하고 있습니다.

## 현재 구인 구직 프로세스

1. 사람인, 잡코리아, 기타 교육기관을 통한 구인 구직 홍보
2. 지원자 대상으로 서류 및 포트폴리오 검토 (1차)
3. 1차 합격자 대상으로 현장 손코딩 테스트 및 개발진 면접 진행 (2차)
4. 2차 합격자 대상으로 임원진 면접 진행 (최종)
5. 최종 합격자 대상으로 수습기간 3개월을 토대로 실무 진행

## 면접관으로서 해야 할 일

상급자의 경우 '회사에서 오래 일할 수 있는', '지시에 잘 따르는', 등의 기준을 토대로 사람을 뽑으려 하는 경향이 있습니다. 이는 절대 잘못된 부분이 아닙니다.

하지만 우리는 팀에서 '같이 일하고 싶고', '빠르게 업무 처리를 해 낼수 있으며', '나아가 일당 백의 역할을 할 수 있는' 사람을 찾도록 노력해야 합니다. 그것이 임원진 면접 진행과 차별되는 개발진 면접 존재 이유입니다.

## 면접관 필수 요소

1. 성별, 나이, 비장애/장애 여부, 학교 등을 토대로 선입견을 가지지 맙시다. (이왕이면 포트폴리오만 선확인하세요.)
2. 자기 소개서의 감성팔이는 쳐 내고, 보이는 것 그대로만 봅시다. (사실 저는 서류검토 할 때 자기 소개서는 이제껏 한번도 안봤어요. 물론 자기 소개서는 중요합니다.)
3. 사실 프로그래머는 밀천이 **프로그래밍 능력**입니다. 능력을 보고 뽑으세요. (물론 면접관들도 그만큼 성장해야 합니다.)
4. 그렇지만 회사는 프리랜서 집단이 아닙니다. 팀 케미에 맞는 사람을 뽑으세요.
5. 단순히 코딩만 잘 하는 사람은 한계가 있습니다. 컴퓨터 사이언스를 사랑하는 사람을 만나요.
6. 1~6번 다 각설하고, 같이 일하고 싶은 사람을 뽑으세요. 향후 이 사람과 같이 일하면 얼마나 즐겁게 효율적으로 일할 수 있을지 생각하며 뽑읍시다.

## 면접관이 하지 말아야 할 일

1. 갑질, 압박 면접, 분위기 어둡게 하기
2. 너무 가볍고 장난기 있는 문제 혹은 개인적인 담화
3. 면접 후 오랜 기간 결과 연락 안하기

다 귀한 시간내서 오는 분들입니다. 또, 합격 문자를 받기 위해서 핸드폰을 잡고 기다리실 분들입니다.

최선을 다해서 면접에 임해주세요.

## 면접시 질문 사항

개발 면접을 진행하면서 사용했던 질문 레파토리는 보통 다음과 같습니다.

1. 코딩 테스트에 작성한 답변에 대한 질문 및 이해도 파악
2. 포트폴리오의 프로젝트에 대한 본인의 작업 범위 및 설명
3. Computer Science 전반적인 문제 (Design Pattern, Language, ...)
4. 회사에서 지원자분을 뽑아야 하는 이유가 뭘까요?

다음의 잡 인터뷰들을 보시면서, 좋은 지원자를 찾아낼 수 있을지 확인해 보세요.

- [Front-end Job Interview Questions](#)
- [Django Job Interview Questions](#)

## 향후 발전 방향

현재 코딩 테스트 시험지의 문제점

- Web과는 관련되지 않는 내용도 포함되어 있음
- Python, Django를 모르는 지원자들이 많아 함부로 Python, Django 문제를 낼 수 없음
- 유일하게 글로벌하게 평가 가능한 언어인 Javascript의 출제 문제는 사실 구글링 검색으로 풀 어낼 수 있는 문제
- 무엇보다도 실무 능력을 파악하기 위한 과정이 아닌 것을 느낌

좋은 개발자를 찾기 위해, 또한 코딩 테스트는 점점 의미가 퇴색됨에 따라 여러 회사에서 다양한 면접/평가 방법을 도입하고 있습니다. [비디오 면접](#)이라는 개념을 도입하고 있기도 하고, [알고리즘-디자인 패턴 질문을 구두로 진행하거나](#), [사내에서 작성하는 코드를 주고 리팩토링 하는 방식](#) 등, [새로운 방식](#)을 도입하기도 합니다.

면접관으로서 사람을 만날 수 있는 경험은 흔치 않습니다. 특히 면접자와 면접관의 시각은 여러방면으로 매우 다릅니다! 좋은 경험이라 생각하시고, 이를 느껴보세요.

# Backup

이 Chapter는 Web Project의 Backup에 대한 내용을 포함하고 있습니다.

## 목적

Web Project의 Source는 Bitbucket을 통하여 관리되고 있습니다. 그럼에도 불구하고 Web Project 한정으로 Local에서 백업되어져야 하는 Data가 있습니다.

1. Project 납품 혹은 납품 후 업데이트 당시의 Data
  - i. Source Code
  - ii. File Data (txd, image file등)
  - iii. Database Dump
2. Project 개발 중의 Data
  - i. File Data (txd, image file등)
  - ii. Database Dump
3. Web Team 내부 문서, 프로그램
4. AWS에 올려져있는 Legacy 파일들 (2010~2014년 사이의 파일들)

## 방법

현재 WD My Passport 1TB를 사용하여 Backup 중입니다.

파일트리는 다음과 같습니다.

- WebTex3D
  - Yuka
    - keyuca
      - curtain
      - furniture
    - evegarden
    - fukushin
    - sunregent
  - Youngwoo
    - Styleforus
    - fabric
    - shirts
    - mobile

- uniform
- Elite

각 폴더 내에 트리는 다음과 같습니다.

- Project Folder
  - Latest (가장 최신에 백업된 내용)
    - Data
    - Sources
  - YYMMDD (YY년 MM월 DD일 백업)
    - Data
    - Sources
  - ...

## 백업 지침

1. 프로젝트에 대한 첫 백업은 프로젝트 이름의 폴더를 만들어 내부에 Latest라는 폴더를 만들어 주세요.
  - i. Data와 Source 폴더를 만들어 주세요.
  - ii. Data에는 프로젝트의 데이터 부분을 저장합니다.
    - i. 프로젝트의 media/data/ 안의 내용을 data.zip 압축하여 올려주세요.
    - ii. 해당 날짜의 Database를 dump하여 db\_dump 라는 파일명으로 올려주세요.
  - iii. Source는 프로젝트의 코어 부분 (mysite 혹은 프로젝트 이름으로 생성된 py 파일의 폴더)의 부분을 그대로 올려주세요.
2. 백업하려는 프로젝트가 처음이 아니라면 **Latest**폴더를 해당 폴더의 생성일로 변경하고, 새로 백업할 내용을 백업지침 1번을 따라 진행해주세요.
3. 프로젝트는 바로 내려받아 설치했을 때 작동할 수 있도록 무결성을 유지해야 합니다.
4. 백업 디스크의 여유 용량에 따라 오래된 백업 파일을 삭제 할 수 있습니다. 다만 **각 폴더에 Latest로 백업된 것은 존재**해야 합니다.
5. 백업 주기
  - i. 6개월에 한번씩 주기적 백업을 진행합니다.
  - ii. 납품된 프로젝트가 추가적으로 변경이 있었을때, 바로 핫픽스 백업을 진행합니다.

# Service Management

## Bitbucket 관리

Bitbucket은 형상 기억 관리 시스템 중 하나인 Git을 쓸수 있도록 도와주는 서비스입니다. (비슷한 것으로 Github라는 것이 있습니다)

### 사용 이유

- Git을 사용하여 프로젝트 단위별로 소스코드의 형상을 기억할 수 있도록
- Git의 Pull, Push를 원격 저장소 (Bitbucket 서버)에 두어 데이터의 안정성을 높이기 위해
- Pull Request를 효과적으로 사용하기 위해
- Private Repository 사용 가능 (5인 한정, github는 아쉽게도 무조건 유료여서 사용 불가)

### Git

Git은 웹 서비스를 구축하는 데 있어서, 개발자간의 소스코드 공유를 효과적으로 할 수 있도록 돕습니다. 현재 C에서는 비슷한 방식인 SVN을 사용하고 있고, 과거에 Web에서는 압축파일 (Zip) 을 이용해 코드를 공유했습니다.

다음 링크를 통해 Git을 더 알아보세요.

- [Git이란?](#)
- [Git Tutorial](#)
- [Git Tutorial 2](#)
- [Git 전문가가 됩시다!](#)
- [Git VS SVN](#)

### Host on Bitbucket

[영우씨엔아이의 Bitbucket](#)에서 새로운 Project를 만들거나 Clone할 수 있습니다.

## Domain 설정

Stylefor.us군은 가비아를 통해, Texclub은 를 통해 서비스 되고 있습니다.

1. 가비아
  - \*.stylefor.us 의 관리

- 도메인 연장 (Domain Expiration Date: Sun Oct 23 23:59:59 GMT 2016)
- 네임플러스 서비스 사용 (신규 및 정식 프로젝트의 도메인 부여를 위해)  
 특히, 네임플러스 서비스(서브 도메인 설정)를 하여 \*.stylefor.us 형식의 도메인 접근을 가능토록 합니다. 설정 방법은 stylefor.us 서버의 Apache 설정 ( httpd.conf, extra/httpd-vhost 파일 )을 참고하거나 [Apache문서](#), 혹은 [외부 링크](#)를 참고합니다.

## 2. 텍크넷

- texclub.com 의 도메인 관리
- 도메인 연장 (Expiration Date: 15-jun-2017)

# Hosting 설정

## AWS

테스트 서버로 사용하고 있는 Hosting 서버입니다. AWS의 개념에 대해서는 [외부 링크](#)와 [나무 위키 문서](#)를 참조해주세요.

- 54.64.114.3
- 원래 Keyuca를 위한 테스트 서버로 사용하였으나 기간을 연장하여 영우씨엔아이에서 운영 중
- AWS 관리자에 대한 문서는 개발서버 참조
- Windows Remote Connection (혹은 윈도우 원격접속)을 통하여 접속 가능
- [Filezillar Server](#) 를 통해 Port를 개방 후 FTP를 통해 코드 및 데이터 업로드 가능
- 쓴 만큼 금액이 청구되는 종량제이므로, 장기간 사용하지 않을 경우 AWS 관리자 화면으로 로 그인하여 일시 중지 합니다.
- 적은 용량으로 인해, 새로운 프로젝트 테스트를 진행하기 위해 다른 프로젝트를 제거할 수 있음  
 이때 반드시 Apache Restart를 하여 다른 프로젝트, 혹은 새로운 프로젝트가 잘 동작하는지 테스트해야 합니다
- Port별로 프로젝트를 관리 중입니다. Apache를 통해 Port별로 프로젝트를 할당하는 방법은 Apache 설정 ( httpd.conf, extra/httpd-vhost 파일 )을 참고하거나 [Apache문서](#), 혹은 [외부 링크](#)를 참고합니다.

## Texclub

SFTP를 통해 texclub.com의 서버로 접근할 수 있습니다.

- Cyberduck, Filezillar 등의 클라이언트 프로그램으로 SFTP 접속
- root@texclub.com으로 접속하며 다음 문서를 통해 Texclub의 서버를 컨트롤하는 방법을 참조해 주세요.

- 개발서버/server/current/Document/11. 인수인계/18. 웹페이지 관리/웹 페이지 관리 인수인계.ppt
- 개발서버/server/current/Document/11. 인수인계/20. 개발 담당자 폴더/09. 정영수/서버, 메일, 홈페이지 관리.odt

## Texclub Database

테이블	실행	레코드수	종류	Collation	크기	부담
Member		3,007	MyISAM	latin1_swedish_ci	515.1 KB	-
cm2.Branch		0	MyISAM	latin1_swedish_ci	1.0 KB	-
cm2.BranchMember		0	MyISAM	latin1_swedish_ci	1.0 KB	-
cm2.Brand		1,068	MyISAM	latin1_swedish_ci	102.8 KB	-
cm2.BrandCommentUpdated		146	MyISAM	latin1_swedish_ci	5.6 KB	-
cm2.Client		866	MyISAM	latin1_swedish_ci	118.5 KB	-
cm2.Comment		7,911	MyISAM	latin1_swedish_ci	1.2 MB	-
cm2.Config		9	InnoDB	latin1_swedish_ci	16.0 KB	-
cm2.Coupon		21,010	MyISAM	latin1_swedish_ci	1.1 MB	-
cm2.Employ		4,457	MyISAM	latin1_swedish_ci	238.4 KB	-
cm2.Equip		1,624	MyISAM	latin1_swedish_ci	139.1 KB	-
cm2.Exam		22	MyISAM	latin1_swedish_ci	2.8 KB	-
cm2.ExamApply		7,467	MyISAM	latin1_swedish_ci	522.1 KB	-
cm2.Log		9,422	MyISAM	latin1_swedish_ci	772.0 KB	-
cm2.Migration		529	MyISAM	latin1_swedish_ci	15.7 KB	-
cm2.PostNumber		50,432	MyISAM	latin1_swedish_ci	4.4 MB	-
cm2.StringTable		620	MyISAM	latin1_swedish_ci	45.7 KB	-
cm2.SubComment		131	MyISAM	latin1_swedish_ci	2.1 KB	-
cm2.SubComment		23	MyISAM	latin1_swedish_ci	6.3 KB	-
cm2.SubComment		50	MyISAM	latin1_swedish_ci	2.6 KB	-
cm2.SubComment		7	MyISAM	latin1_swedish_ci	3.5 KB	-
cm2.Update		3	MyISAM	latin1_swedish_ci	2.2 KB	-
cm2.UpdateFile		3	MyISAM	latin1_swedish_ci	2.1 KB	-
cm2.User		6,218	MyISAM	latin1_swedish_ci	468.1 KB	-
cm2.UserComment		1,852	MyISAM	latin1_swedish_ci	149.4 KB	-
cm2.activation		16,943	MyISAM	latin1_swedish_ci	2.4 MB	-
cm2.activationEmail		12	MyISAM	latin1_swedish_ci	3.4 KB	-
cm2.activationTempKey		22	MyISAM	latin1_swedish_ci	3.5 KB	-
cm2.activationUsage		2,210	MyISAM	latin1_swedish_ci	85.4 KB	-
cm2.activationUsageLog		1,369,048	MyISAM	latin1_swedish_ci	48.5 MB	-
cm2.activationUser		1,920	InnoDB	latin1_swedish_ci	176.0 KB	-
cm2.activationWeb		34	InnoDB	latin1_swedish_ci	32.0 KB	-
cm2.activationWebUser		29	InnoDB	latin1_swedish_ci	32.0 KB	-
cm2.aspActivation		440	MyISAM	latin1_swedish_ci	54.2 KB	-
cm2.aspDiscountUser		1	MyISAM	latin1_swedish_ci	2.1 KB	-
cm2.aspGoods		4	MyISAM	latin1_swedish_ci	2.5 KB	-
cm2.aspGoodsProgram		4	MyISAM	latin1_swedish_ci	2.0 KB	-
cm2.aspGroup		12,397	MyISAM	utf8_general_ci	1.0 MB	-
cm2.aspOrder		467	MyISAM	latin1_swedish_ci	120.8 KB	-
cm2.aspOrderProgram		438	MyISAM	latin1_swedish_ci	41.5 KB	-
cm2.aspOrder_Card		271	MyISAM	latin1_swedish_ci	28.4 KB	-
cm2.aspOrder_Transfer		40	MyISAM	latin1_swedish_ci	3.8 KB	-
cm2.aspOrder_Virtual		113	MyISAM	latin1_swedish_ci	16.2 KB	-

이미 너무 많은 길을 걸어온 (...) Texclub의 Database에 대하여 서술합니다.

- MySQL 5.5
- PhpMyAdmin 2.6.2 기반
- youngwoo Database에 관련 내용들이 저장됨
- 인증 및 각종 texclub의 서비스에 대한 database가 저장됨

CJK 텍스트이기 때문에 utf-8방식으로 저장되어야 했으나, collation을 보면

latin1\_swedish\_ci 로 지정되어 있습니다. 따라서 PhpMyAdmin에서 보게 되면 모든 글자들이 깨 집니다. (...) 수정을 해보려 했으나, 다음과 같은 문제점이 있었습니다.

- 현재 업무용으로 사용되는 Database이다 보니 건들기 곤란.



- Database collation 뿐만 아니라 php에서의 인코딩 또한 변경 필요.
- 미리 도전했던 사람들의 [문서](#) 참고

[백업](#)을 꼭 부탁드립니다.

# Communication

이 Chapter는 의사소통을 위한 내용을 포함하고 있습니다.

1. [Internal](#) : 내부적인 커뮤니케이션을 위한 내용이 기입된 문서입니다.
2. [External](#) : 대외적인 커뮤니케이션을 위한 내용이 기입된 문서입니다.
3. [신입사원을 위한 세팅 가이드](#) : 신입사원에게 진행해야 할 설치 가이드입니다.
4. [Install Parallels Desktop on Mac](#) : Mac 초기 설치시 우리 회사의 개발 환경에 맞출 수 있도록 도움을 주는 가이드입니다.

# Internal

이 Chapter는 회사 내부에서 의사소통을 위한 내용을 포함하고 있습니다. 거의 대다수가 알고 있는 중복된 내용을 피해, 간추린 부분만을 제공합니다.

## 개발팀 내부 의사소통

개발팀 내부에서는 다양한 방식으로 의사소통을 진행합니다. 다음 내용을 읽어주세요.

### 주간회의

매주 1회, 월요일 팀장 회의 종료 후 이루어집니다. 기존에는 개발팀장 주도하에 이루어졌습니다. 필요한 경우 원하는 주제를 가지고 상시 회의할 수 있습니다.

### Slack

업무 내용을 카카오톡 등의 개인적인 SNS로 주고 받는 것을 피하기 위해 업무용 이야기 채널인 [Slack](#)을 도입하였습니다. Slack이 좋은 이유는 [외부 링크](#)를 참조해주세요.

향후 사용 시 다양한 App을 붙일수 있는 점을 고려해 주세요. 현재 채널 #web에는 Bitbucket의 알림이름, #trello에는 Trello의 알림이름 붙여둔 상태입니다. 그 외에도 여러가지 App 들을 채널 혹은 팀에 붙일 수 있습니다.

#### 문제점

1. Slack에서 한 말, 첨부파일은 시간이 지나면 사라집니다

이 이유는 우리가 무료 계정을 사용하고 있기 때문입니다... 이 점 때문에 임시적인 Data 공유 및 의견 제시, 일상 이야기는 올려도 되지만, 꼭 반영구적으로 저장되어야 하는 Data는 개발팀 내부 서버 등을 이용해 저장해주세요.

### Trello

업무 현황을 한 눈에 파악하기 위해 Issue를 카드로 만들어 보여주는 [Trello](#)를 도입했습니다. Trello를 도입한 이유는 [외부 링크](#)를 참조해주세요.

Board를 만들고, 각 Board 내부에 List를 만들 수 있습니다. 그 List 안에 진행되는 Issue를 카드 형식으로 만들어 주세요.

완료되어 자주 살펴보지 않는 내용이라면 Archive라는 기능을 이용해 Archive 꾸러미에 담아둘 수 있습니다.

## 문제점

### 1. Trello의 List 구성이 이상적이지 않습니다.

- 기본 사용법, 다른 사람들이 트렐로를 사용하는 방법

보통 Trello는 해야할 일, 하는 중, 다 완료된 일, 기타 등 3-4가지 List로 구성하고, Card를 배치합니다. 현재의 경우 프로젝트 단위, C/Web 팀 단위로 구성이 되어 있습니다. 추후 이 부분을 C, Web 각각의 보드로 나누어 관리하는 것이 수월합니다.

### 2. 작성하는 사람만 작성하게 됩니다.

- 사용 현황 결과 이가현, 김찬울 사원 위주로 카드가 작성되는 것을 확인하였습니다.
- 사용하는 사람만 사용하게 되면 이 툴을 사용하는 목적이 없어집니다.
- 그렇다고, 전 사원이 카드를 작성하게 되면 콘텐츠의 중복 및 교란이 생깁니다.

따라서 각 팀의 관리자가 카드를 주도적으로 만들고, 팀원들이 세부 내용을 채워나도록 하는것이 좋습니다. 혹은 이 툴, 시스템이 우리팀에게는 안 맞다고 결정되면, 데이터를 백업 후 다른 시스템을 사용해 주세요.

## Naver Band

우리 회사 내부에서 사용하는 **네이버 밴드**는 Slack과 트렐로의 도입 이후 크게 사용하지 않습니다. 하지만 기존에 작성되었던 정보들이 있기 때문에 필요한 정보들을 살펴보시길 바랍니다.

사원 입사 혹은 퇴사의 경우 각 서비스의 계정 관리를 반드시 철저히 부탁드립니다.

## 팀간 의사소통

사내에서는 크게 두가지 종류로 의사소통이 진행됩니다.

### Naver Band

네이버 밴드의 **전체 회사 밴드**를 통해 사내 공지사항 등이 올라옵니다. 또한 웹쪽은 크게 이슈가 없지만, 개발 관련 협의를 위해 디자인팀 멤버들이 소속되어 있는 **개발톡 밴드**에서도 이야기가 진행됩니다.

### Interphone

인터폰은 좌석에 배치되어있는 전화를 통해 진행이 가능합니다. 전화연결을 원하는 코드를 입력하면 대상에게 전화가 진행됩니다. 또는 **9**번을 눌러 외부 전화로 연결이 가능합니다.

# External

## Yuka-Alpha

Yuka-Alpha Chapter의 해당 문서를 참조해 주세요.

## Elite

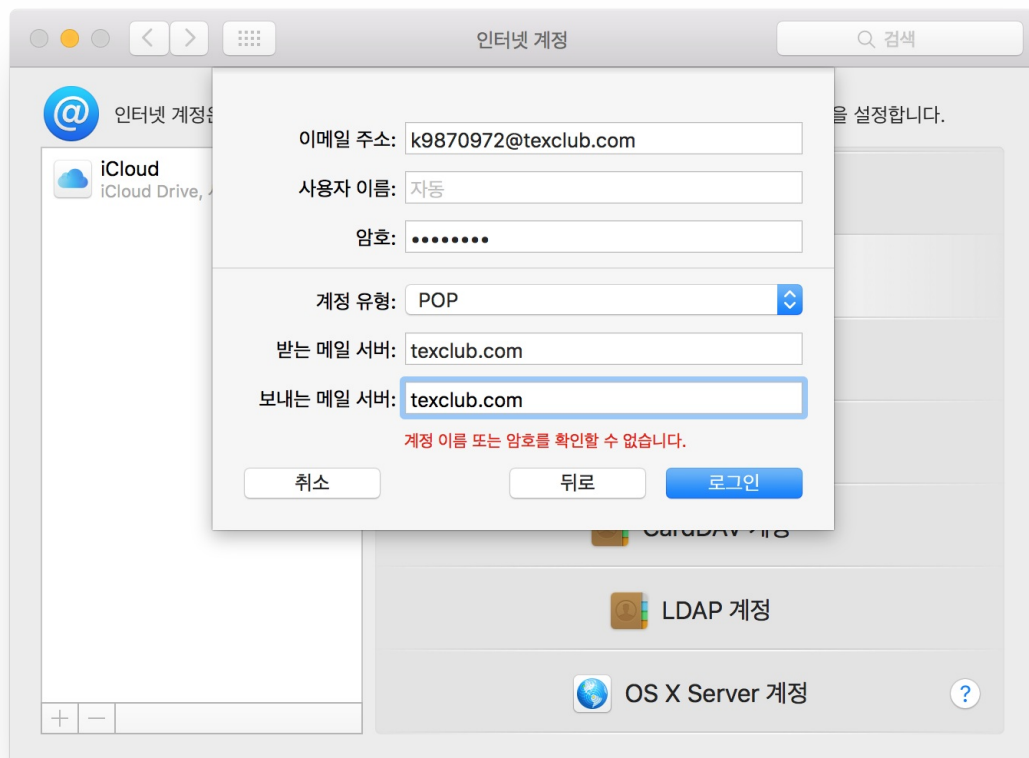
Elite 프로젝트 Chapter의 해당 문서를 참조해 주세요.

*Additional contents will provided in 1~2days Later.* 추가내용 보충 예정.

# 신입사원을 위한 세팅 가이드

웹 파트 신입사원의 경우 다음과 같은 순서를 따라 지시해 주시면 됩니다.

1. Texclub 회원 가입 및 등급 설정
2. Texclub 메일 계정 생성 및 Mail 클라이언트 설정 (이후 진행되는 서비스 계정 생성은 무조건 texclub계정으로 진행)



3. 필수 App 설치
  - 한글 2002
  - [OpenOffice](#)
  - [Slack](#)
  - [SourceTree](#)
  - JetBrains 제품군
    - [PyCharm](#)
    - [WebStorm](#)
4. Slack, Trello, Naver Band 초대 및 설치, 로그인
5. Bitbucket 계정 생성 및 Project 합류
6. 개발 환경 세팅

7. Bitbucket 프로젝트 내려받기 및 작동 테스트
8. 개발팀 내부 공유 서버 연결
9. 회사 내부 공유 서버 연결

## Mac을 쓰는 사원의 경우

기존 사원이 Mac을 사용하게 될 경우 다음과 같은 순서를 따라 행동하시면 됩니다.

1. Apple 계정 생성 (혹은 기존에 사용하던 계정 등록)
2. Mail App을 통해 texclub 계정 설정
3. 필수 App 설치
  - [Karabiner](#) (한영 키를 위해)
  - [Slack](#)
  - [SourceTree](#)
  - [OpenOffice](#)
  - iWorks 제품군
    - [Pages](#)
    - [Numbers](#)
    - [Keynote](#)
  - [Parallels](#)
    - i. [Parallels의 윈도우 세팅](#)
    - ii. [Parallels의 개발환경 세팅](#)
    - iii. [Parallels에서 Symbolic Link를 이용하여 Windows와 Mac간의 개발 폴더 공유](#)
      - i. Mac에서는 원하는 작업 구역에 'www'라는 폴더를 만듭니다. 이 폴더는 추후 git Repository의 파일들을 내려받는 공간이 될 것입니다.
      - ii. [Parallels의 윈도우에서는 맥의 'www'폴더가 \\Mac\Home\Documents\www 형식으로 인식됩니다. 이는 공유 폴더로 인식하기 때문입니다.](#)
      - iii. 이 폴더를 [symbolic link](#) 로 c:\에 만들어줍니다. 주소는 본인 컴퓨터에 맞게 작성하여 cmd에 넣으세요. `mklink /d "c:\www" " \\Mac\Home\Documents\www"`
      - iv. 처음엔 `python manage.py runserver`로, 그 후에는 `apache`로 시뮬레이션을 테스트하여 잘 작동하는지 체크하세요.
        - [에러가 생길 수 있는 이유](#)
          - i. `sim.ini` 체크
          - ii. `wsgi` 설치 여부 및 `wsgi.py`의 경로 체크
          - iii. `setting.py`의 경로 체크
          - iv. `Apache Log` 확인
  - [Microsoft Remote Desktop](#)
  - [Balsamiq 3](#)
  - JetBrains 제품군
    - [PyCharm](#)
    - [WebStorm](#)

4. 개발팀 내부 및 회사 내부 공유 서버 연결 후, **CMD로 명령어 진행** (Mac 흔적을 없애기 위해)

## 퇴사하는 직원을 위한 Guide

반드시 필요한 정보를 백업, 보관 후 진행해야 합니다

1. 문서 작성, 인수인계 진행 및 데이터 백업
2. Trello, Slack, Naver Band 탈퇴
3. JetBrains 등, 필요한 라이선스 반환
4. Texclub 계정 탈퇴 및 Mail 계정 삭제
5. 컴퓨터 초기화 (Mac의 경우 iCloud 계정 해제 필수)



# Install Parallels Desktop on Mac

1. Parallels의 윈도우 세팅

2. Parallels의 개발환경 세팅



3. Parallels에서 [Symbolic Link](#)를 이용하여 Windows와 Mac간의 개발 폴더 공유

i. Mac에서는 원하는 작업 구역에 'www'라는 폴더를 만듭니다. 이 폴더는 추후 git Repository의 파일들을 내려받는 공간이 될 것입니다.



ii. Parallels의 윈도우에서는 맥의 'www'폴더가 `\\Mac\Home\Documents\www` 형식으로 인식됩니다. 이는 공유 폴더로 인식하기 때문입니다.



iii. 이 폴더를 [symbolic link](#) 로 c:\에 만들어줍니다. 주소는 본인 컴퓨터에 맞게 작성하여 cmd에 넣으세요. `mklink /d "c:\www" " \\Mac\Home\Documents\www"`



iv. 처음엔 `python manage.py runserver`로, 그 후에는 `apache`로 시뮬레이션을 테스트하여 잘 작동하는지 체크하세요.



- 에러가 생길 수 있는 이유
- `sim.ini` 체크
- `wsgi` 설치 여부 및 `wsgi.py`의 경로 체크
- `setting.py`의 경로 체크
- Apache Log 확인

# WebTex3D

이 Chapter는 WebTex3D 서비스의 내용을 포함하고 있습니다.

전체적으로 개발 환경 및 설치, WebTex3D에 수반된 각종 프로젝트들을 서술합니다.

1. [Specification](#) : WebTex3D의 설치 사양 및 설치되는 소프트웨어의 버전 등을 서술합니다.
2. [How To Install](#) : WebTex3D의 설치 방법에 대하여 알려드립니다.
3. [Youngwoo](#) : 영우씨엔아이 내부에서 제작된 Project에 대한 각각의 설명입니다.
4. [Yuka-Alpha](#) : Yuka-Alpha의 계약을 통해 제작된 Project에 대한 각각의 설명입니다.
5. [Elite](#) : 형지Elite와의 계약을 통해 제작된 Project의 설명입니다.

# Specification

## WebTex3D를 설치하기 위한 조건

- Windows 2008 버전 이상의 Server 제품군
- 최소 2GB 이상의 RAM
- 최소 100GB 이상의 HDD
- 항상 Online 연결

## WebTex3D를 설치하기 위해 필요한 소프트웨어

- Python 2.7.X
- Django 1.4.X
- PostgreSQL 9.2 이상
- south 0.8.4
- psycopg2-2.6.1.win32-py2.7-pg9.4.4
- Pillow-2.7.0.win32-pypy2.4-py2.7

최신 버전의 경우 문제가 발생할 수 있습니다. 반드시 버전을 지켜 설치를 부탁드립니다. 각 버전별 다운로드 링크는 *Backup*된 내용 혹은 [How To Install](#) 문서를 통해 참조하세요.

# How To Install

이가현 사원의 문서를 참고하여 작성하였습니다. 다운로드 링크가 존재하지 않는 경우 Backup된 설치 파일 들을 이용해주세요.

## 설치 방향

### 1. python 2.7.3 설치

- python-2.7.3.msi
- [Download](#)
- python27폴더를 속성(내컴퓨터) -> 고급, 환경변수 -> path에 추가 `C:\Python27\;` (<- 환경변수 입력할때 \;잊지말것!)

### 2. postgresql(DB) 설치

- postgresql-9.2.3-1-windows.exe
- <http://www.postgresql.org/download/windows/> <-Download 경로
- `C:\Program Files (x86)\PostgreSQL\9.2;C:\Program Files (x86)\PostgreSQL\9.2\bin;` 를 환경변수 path에 추가
- 설치시 기본 유저 : postgres
- 패스워드 : 00000000
- 기본 포트 : 00000000

### 3. django 1.4.7 설치

- Django-1.4.7z
- [Download](#) (구버전이라 사실 다운 불가)
- 압축해제한 뒤, `C:\Django-1.4` 로 디렉토리 이동
- `C:\Django-1.4>python setup.py install` (<-Enter)
- 설치완료되면 `C:\Python27\Lib\site-packages\django` 디렉터리가 생성됨

`C:>python -> python 실행+version 확인`

```
>>> import django 했을때 에러 안나면 ok django.VERSION 결과로 (1, 4, 2, 'final', 0) 등이 나오면 설치 성공
```

- `C:\DjangoProject0>django-admin.py startproject mysite` 명령어로 프로젝트 생성

```
mysite/
manage.py
mysite/
    __init__.py
    settings.py
    urls.py
    wsgi.py
```

- 위와같은 경로로 생성되는데 manage.py는 settings.py와 같은 폴더에 있어야함.

```
mysite/
__init__.py
settings.py
urls.py
wsgi.py
manage.py
```

위와 같이 구성하도록 함.

manage.py, settings.py, wsgi.py 에 'mysite.~' 와 같이 되어 있는 부분들에서 'mysite.'을 삭제.

예) `os.environ.setdefault("DJANGO_SETTINGS_MODULE", "mysite.settings")` 은  
`os.environ.setdefault("DJANGO_SETTINGS_MODULE", "settings")` 로 되어야 합니다.

#### 4. south 설치 South는 django와 postgresql이 연동할 수 있도록 해주는 툴로서 반드시 필요함.

- south-0.8.4.tar.gz
- [Download](#)
- South 최신버전을 다운로드 받아 압축해제 후 폴더에 setup.py 파일이 있는지 확인.
- cmd->압축해제한 디렉토리로 이동 `C:\South-0.8.4>python setup.py install` (<- Enter)
- South 설치완료

#### 5. Psycopg2 설치

- psycopg2-2.6.1.win32-py2.7-pg9.4.4-release.exe
- [Download](#)

최신버전을 다운로드 받아도 무방하지만 반드시 python2.7용을 설치해야 함 주의할 점은 반드시 exe 파일로 설치해야 함. 윈도우에서 tar.gz을 다운로드받아 위의 django 또는 South와 같이 설치할 경우 작동하지 않음. python과 postgresql 연결해주는 adapter

cmd에서 python으로 접속후, `>>> import psycopg2` 했을때 에러 안나면 (아마)ok

## 6. Pillow 설치

- Pillow는 python의 이미지 프로세싱을 하는 라이브러리
- Pillow-2.7.0.win32-pypy2.4-py2.7.zip <-일단 압축해제
- `site-packages\PIL,\Pillow` 디렉터리를 `C:\Python27\Lib\site-packages` 에 복사
- [Download](#)
- 다른 설치방법(현재는 python v3.4와 충돌이 나는 것 같아서 위의 방법대로 함) -> 주의 : 설치파일은 최신버전을 다운로드 받아도 무방하지만 반드시 python2.7용을 설치해야 함
  - i. <http://www.lucypark.kr/blog/2013/02/06/python-and-ipython-on-windows/> 에서 `get-pip.py` 다운로드 `C:\Python27` 경로에 파일 복사한 뒤, cmd에서 `C:\Python27>python get-pip.py` 해서 pip설치  
<http://www.lfd.uci.edu/~gohlke/pythonlibs/>에서 .whl파일을 다운받은 뒤, `pip install whl파일명` 으로 설치
  - ii. <http://www.lfd.uci.edu/~gohlke/pythonlibs/>에서 Pillow-2.3.0-win.....exe를 다운로드 받아 설치

## 초기 DB Setting

- PostgreSQL을 사용하기 위해 아래 내용을 환경변수에 추가 \* `C:\Program Files\PostgreSQL\9.2\bin`

### DB 만들기

webtex3dse db가 이미 있으면 지우고 새로 넣어준다.

1. webtex3d data 복원 목적 ('webtex3dse' = webtex3d db name)
2. cmd에서 아래 명령어를 이용해서 db를 생성할 수도 있고, PG AdminIII로 생성할 수도 있음

```
C:\PostgreSQL\9.2\bin>dropdb -U postgres webtex3dse
```

```
C:\PostgreSQL\9.2\bin>createdb -U postgres webtex3dse
```

### db dump 파일 restore는 아래의 Data 백업\복원 내용 참고

db dump파일을 restore. 이때 dumpfilename은 fullpath `C:\PostgreSQL\9.2\bin>psql -U postgres webtex3dse < dumpfilename`

- 설정이 끝나면 mysite로 이동해서 south와 연동위해 `python manage.py syncdb`

### Data 백업하기(내보내기)

cmd->백업파일을 저장할 디렉토리로 이동

```
C:\> pg_dump -U postgres [DB이름] > [생성할 파일 이름] 즉, C:\> pg_dump -U postgres webtex3dse > dump_20150707
```

## Data 복원하기(가져오기)

- Postgres 안에 이미 데이터베이스가 생성되어 있어야 함 (PG AdminIII로 미리 만들어놓는다)  
(다음 예와 같은 경우에는 webtex3dse 라는 빈 데이터베이스가 미리 생성되어 있어야 복원가능)
- cmd->백업파일을 저장할 디렉토리로 이동

```
C:\> p_sql -U postgres [DB이름] < [백업할 파일 이름] 즉, C:\> p_sql -U postgres
webtex3dse < dump_20150707
```

## DB Setting 완료 후

- data.7z 파일
  - webtex3d source에서 버전관리를 하지 않는 data들이기 때문에 일일이 복사해주어야한다.
  - data.7z 파일을 압축해제한 후, 로컬의 C:\var\_w3d\www\media\data 안에 위치시키면 된다.

## Apache 세팅

/Apache2.2/conf/httpd.conf 에 있는 **httpd.conf** 파일에서

LoadModule wsgi\_module modules/mod\_wsgi.so 추가

WSGIPythonPath "c:/var/www/mysite" 추가

Options Indexes FollowSymLinks 에서 **Indexes** 삭제

#Include conf\extra\httpd-vhosts.conf 에서 #(주석) 삭제

\Apache2.2\conf\extra\httpd-vhosts.conf 에 있는 **httpd-vhosts.conf** 파일에서

첫번째 <VirtualHost \*:80> ~ </VirtualHost> 부분을 다음과 같이 변경

### httpd-vhosts.conf

```
<VirtualHost *:80>
  DocumentRoot "C:/var_w3d/www/mysite"
  ServerName "알맞게 변경"
  WSGIScriptAlias / "c:/var_w3d/www/mysite/wsgi.py"
  <Directory "c:/var_w3d/www/mysite">
    <Files wsgi.py>
      Order deny,allow
      Allow from all
    </Files>
  </Directory>
</VirtualHost>
```

위의 경로 "c:/var\_w3d/www/mysite" 는 var\_w3d폴더가 생성될 위치에 따라 변하게 됨

예를들어 현재 d:/var\_w3d로 되어 있다면 위의 설정에서도 "d:/var\_w3d/www/mysite" 가 되어야 함.

설정완료 후 ApacheMonitor에서 apache를 **restart** 해야함.



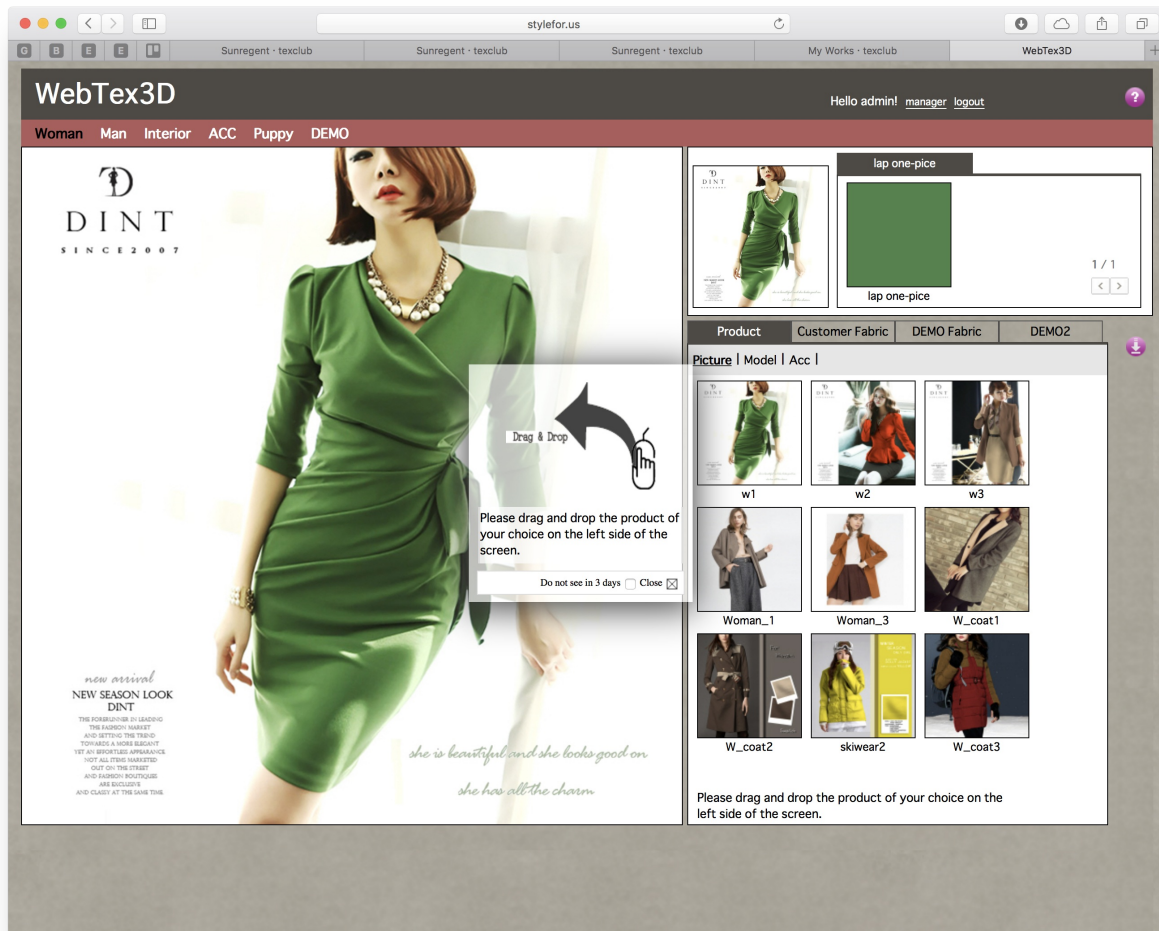
# Youngwoo

영우씨엔아이에서 내부적으로 진행된 WebTex3D 프로젝트를 다루는 Chapter입니다.

3, 4번을 제외한 항목은 김찬울 사원이 직접적으로 참여한 프로젝트가 아닙니다. 따라서 작업한 프로젝트가 아닌 경우 아는 지식 선에서 서술하였습니다.

1. [Stylefor.us](#) : WebTex3D Base
2. [Fabric.stylefor.us](#) : WebTex3D Base에서 GridGroup에 따라 다른 Fabric들을 보여주기 위한 Sample Project
3. [Shirts.stylefor.us](#) : WebTex3D Base와는 다르게 설계된, 셔츠의 Mapping을 위해 만들어진 Project ~~또다른 국책과제라 카더라 시판을 위해 제작되었지만 느린 속도로 인해 납품수-0~~
4. [Mobile.stylefor.us](#) : WebTex3D Base에서 국책과제 (App)를 서비스하기 위해 만들어진 Project
5. [Uniform](#) : WebTex3D Base에서 Uniform을 구현하기 위해 제작된 Sample Project

# stylefor.us



- 링크 : <http://stylefor.us>
- 참여자 (Web 한정) : 김영광, 정영수, 이가현, 김찬울

## 개요

내부 프로젝트의 일환으로 기획된 Stylefor.us 를 개발한 자료입니다.

- WebTex3D 모듈을 이용하여 2번째로 작성된 프로젝트.
- 원래는 Flash의 신 버전인 Flex를 이용하여 작성 중 시대에 맞지 않아 폐기 후 HTML 기반으로 재작성
- 영업용 Demo로도 사용됩니다.
- 가장 기초적인 기능만 담고 있어 다른 프로젝트를 시작하기에 앞서 Clone 되는 프로젝트.
- WebTex3D Base가 바로 이 프로젝트입니다.

# 개념원리

Win32 응용프로그램인 WebTex3D 모듈을 이용하여 Grid Group에 원하는 Fabric을 Mapping할 수 있다. 처리방식에 대한 자세한 내용은 `개발서버/server/current/Document/04. 프로젝트/33.`

`WebTex3D/20151202_WebTex3D Server 구조 및 Simulation 처리 방식.odt` 를 참조합니다.

Tex3DW를 이용해 디자이너가 작업 내용을 Upload 후, Web상에서 Manager 페이지를 이용해 관리할 수 있다.

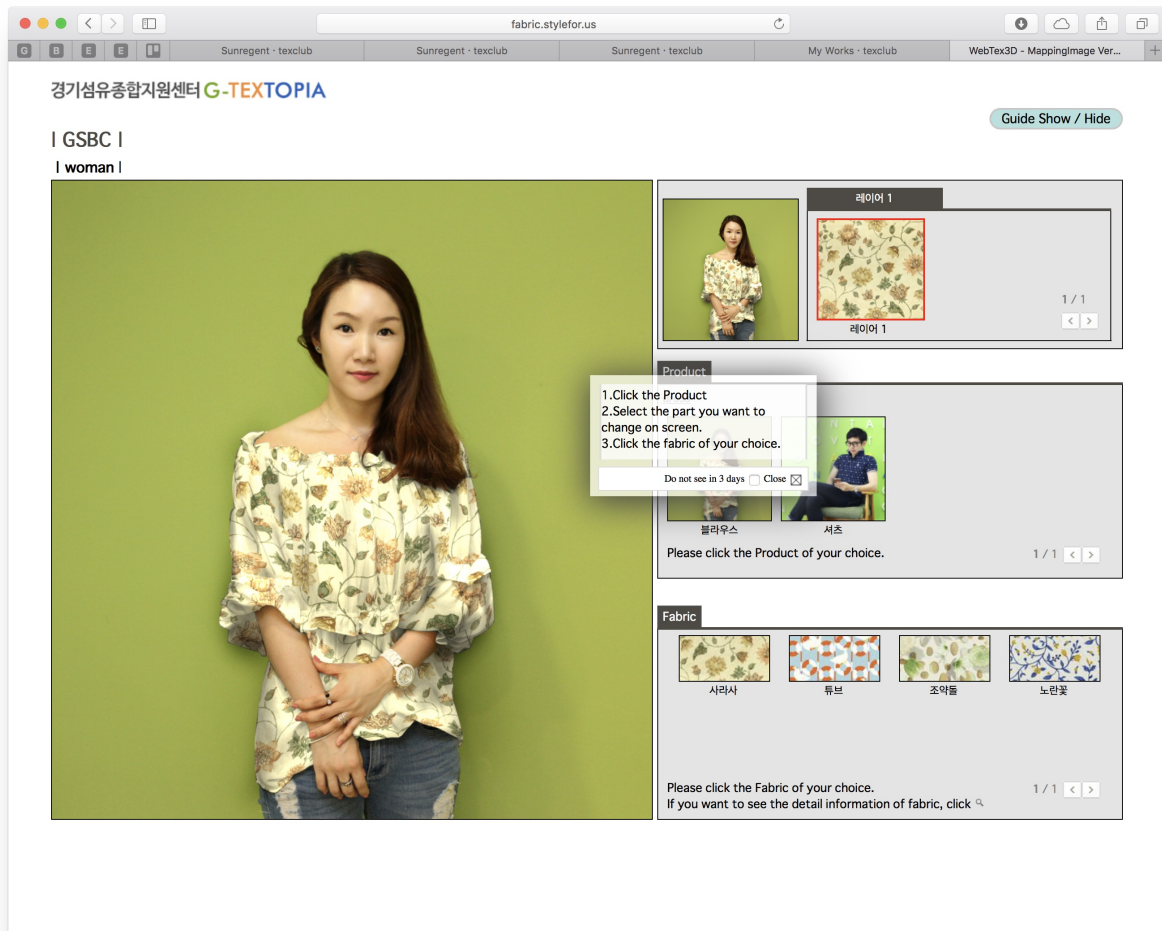
## 주요 기능

- Viewer 페이지에서 이루어지는 기능
  - Product Parse
  - 선택된 Product의 Layer 및 Grid Group, Mask Image 등의 표현
  - Fabric Parse
  - Fabric을 Drag를 이용하여 Mapping 및 자세한 정보 확인 가능
  - Fabric의 UserUpload기능
  - 로그인 기능
- Manager 페이지에서 이루어지는 기능
  - Product의 관리
    - 특정 회원 그룹에게만 노출, 혹은 비공개 기능
    - 각 Fabric 별 상태 수정 (Laundry Tag 등을 비롯하여)
    - 카테고리 설정 (왜 그런지 의도는 모르지만, Level1은 서비스 명을, Level2는 Top Menu Category, Level3는 Product Category로 제작되어 있습니다. 참고하세요.)
  - Fabric의 관리 및 추가
    - 특정 회원 그룹에게만 노출, 혹은 비공개 기능
    - 각 Fabric 별 상태 수정 (Laundry Tag 등을 비롯하여)
    - 카테고리 설정 (Product는 Level3내용을 한 Tab에 몰아 보여주는 반면, Fabric은 Level1을 탭에, Level2, 3을 세부카테고리로 설정)
    - Manager 페이지에서의 신규 Fabric Upload
  - 회원 관리 및 추가
    - Admin, Uploader, Guest 등급 조절 가능
  - 회원 그룹 관리 및 추가
    - 이를 통해 Product, Fabric의 노출 대상 설정 가능
  - 시뮬레이션 리셋 기능
    - Database와 실제 저장된 Data를 제거
- 그외 기능
  - Tex3DW에서의 업로드가 이루어 지는 과정 Support

## 위험요인 및 피드백

- 최신 내용으로, 16년 2월 26일자로 Simulation Failed에 대한 수정이 이루어졌습니다. 자세한 내용은 Commit된 `66b47d1` 내용을 확인해주세요.
- 최신 내용으로, 16년 1월 14일자로 보안 및 OrderNumber의 Bug 및 Logic에 대한 수정이 이루어졌습니다. 자세한 내용은 Commit된 `c0e10a9` 내용을 확인해주세요.
- 레이아웃 및 사용법이 모바일 디바이스에 최적화되지 않았습니다. (Hover, Drag를 제거해야 하며, Responsive Web 방식을 이용하여 콘텐츠를 디바이스에 최적화 될 수 있도록 고쳐야 합니다. ~우선 디자인이 좋지 않습니다~)
- 너무 많은 Data를 가지고 있고, 사용이 많았기 때문에 Database 내부적으로 꼬인 부분들이 있을 수 있습니다. 정기적으로 Simulation Data를 초기화 해주는 것이 좋습니다. 또한 꼬인 부분을 토대로 내부 Logic이 수정되어야 합니다.
- Code가 간결화 되어 있지 않으며, Restful API형태로 개발되지 않아 이를 Base로 삼기 어렵습니다.
  - 추후 버전업이 이루어질 경우, 필요한 기능만 걸러내어 Restful 하게 개발해야 합니다. 이는 추후 다른 프로젝트 (mobile.stylefor.us 등) 처럼 다른 플랫폼 혹은 서비스에 WebTex3D를 제공할 경우 꼭 필요합니다.
  - Javascript의 경우 잘 짜여진 부분과 아닌 부분이 명확히 보입니다. 코드 검수를 통해서 간결화해야 합니다.
  - 코드는 한번 망가지면 지속적으로 망가집니다.
- Mail 보내기 기능 등 더 이상 쓰지 않는 기능들이 있습니다.
- 최신 Django, PIL 등으로 설치시 동작하지 않습니다. 안정성 및 개발 효율을 위해 향후 1년 내에 반드시 업데이트 할 수 있도록 해야 합니다.

# fabric.stylefor.us



- 링크 : <http://fabric.stylefor.us>
- 참여자 (Web 한정) : 정영수, 이가현

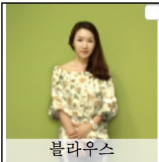
## 개요

내부 프로젝트의 일환으로 기획된 fabric.stylefor.us 를 개발한 자료입니다.

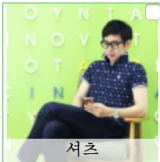
- 영업용 Demo로도 사용됩니다.
- WebTex3D Base에서 Grid Group별로 다른 Fabric을 고를 수 있도록 설계된 프로젝트입니다.

## WebTex3D Base 에서 달라진 부분


Product List




블라우스



셔츠




IMG-6886



여자

1. Product 선택

Product Information



Access

☒ All

☐ Specific group...

There is no Client Groups.

Name

Style No.

Url

Publish ☒


GridsGroup

☒ 레이어 1 ☐ 레이어 1


2. 설정을 원하는 GridGroup 선택

3. 해당 GridGroup에 선택가능하게 하고싶은 Fabric을 드래그하여 우측으로 이동


Fabrics



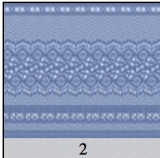
별



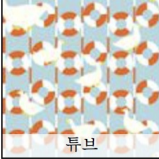
1




사라사




2



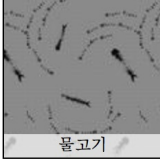
튜브



조약돌




노란꽃




물고기

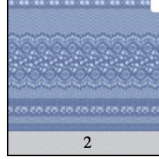
레이어 1



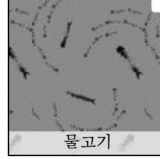
별



사라사



2



물고기

Modify

edit name

4. 해당 GridGroup을 Mapping할때 선택 가능한 Fabric을 보출

Category(level1)

test1

Category(level2)

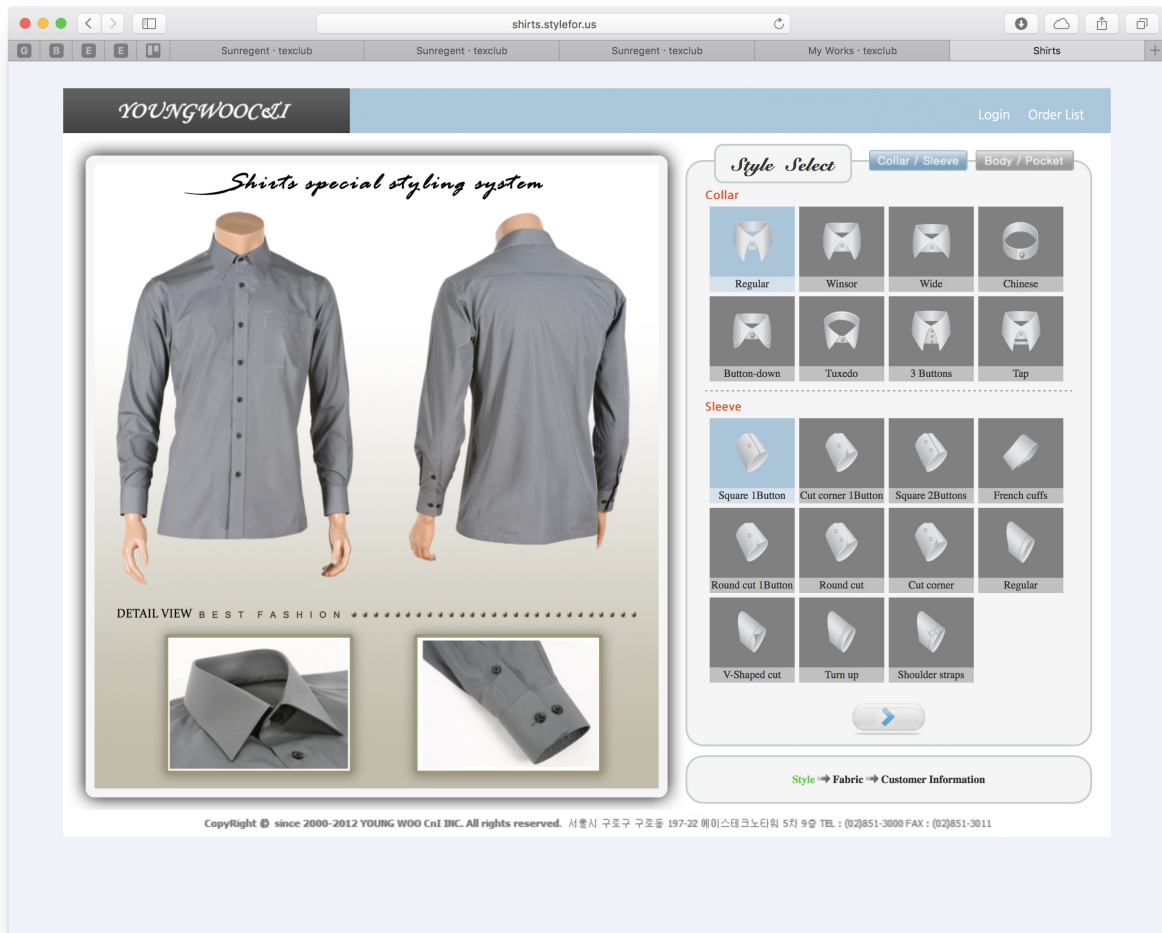
test2

Category(level3)

test3

- Fabric Group의 개념 정리
  - 원하는 Grid Group에 Fabric을 넣을 수 있도록 합니다.
  - 상단 이미지를 통해 구조를 참고해주세요.

# shirts.stylefor.us



- 링크 : <http://shirts.stylefor.us>

## 개요

Yuka 측에서 2016년 1월 중에 요청되어, 2월 18일 부터 6월 2일까지 제작된 Project인, Sunregent 를 개발한 자료입니다.

## 요구사항 및 WebTex3D Base 에서 달라진 부분

- 웹에서 셔츠의 각 파트 이미지의 패턴을 변경해서 시뮬레이션 하는 솔루션
- 셔츠 등 패턴조각이 달라질 수 있는 의류에 대하여 사용 가능
- 이에 따라 따로 디자이너용 윈도우 디자인 프로그램 Tex3DW\_Shirts이 존재함

## 개념원리

아래 내용은 개발서버/server/current/Document/04. 프로젝트/33.

WebTex3D/20140819\_WebTex3D\_Shirt 비교문서.hwp 를 기준으로 작성되었습니다.

- WebTex3d\_Shirts 모듈
  - 패턴 조각 변경에 따라 .lay파일을 불러옴
  - 각 lay파일에 fabric Image Simulation을 차례로 진행
  - 각 lay파일을 통해 나온 결과를 통째로 result로 전송

이런 부분 때문에 기존 WebTex3D 모듈에 비해 3~4배 이상 속도가 느려집니다.

## 위험요인 및 피드백

1. 느린 시뮬레이션 속도
  - i. 각 lay파일을 open, read하는 것이 굉장히 느리다고 알고있습니다.
  - ii. 각 lay파일에 fabric mapping을 하는 것은 stylefor.us처럼 한 product에 원단을 입히는 것과 같은 과정입니다.
  - iii. 각 lay파일에 fabric mapping 후 결과를 모두 합치는 과정 또한 느립니다.
2. 납품수 0
  - i. 느린 시뮬레이션 속도
  - ii. 샘플 디자인 파일 제작의 어려움



# mobile.stylefor.us

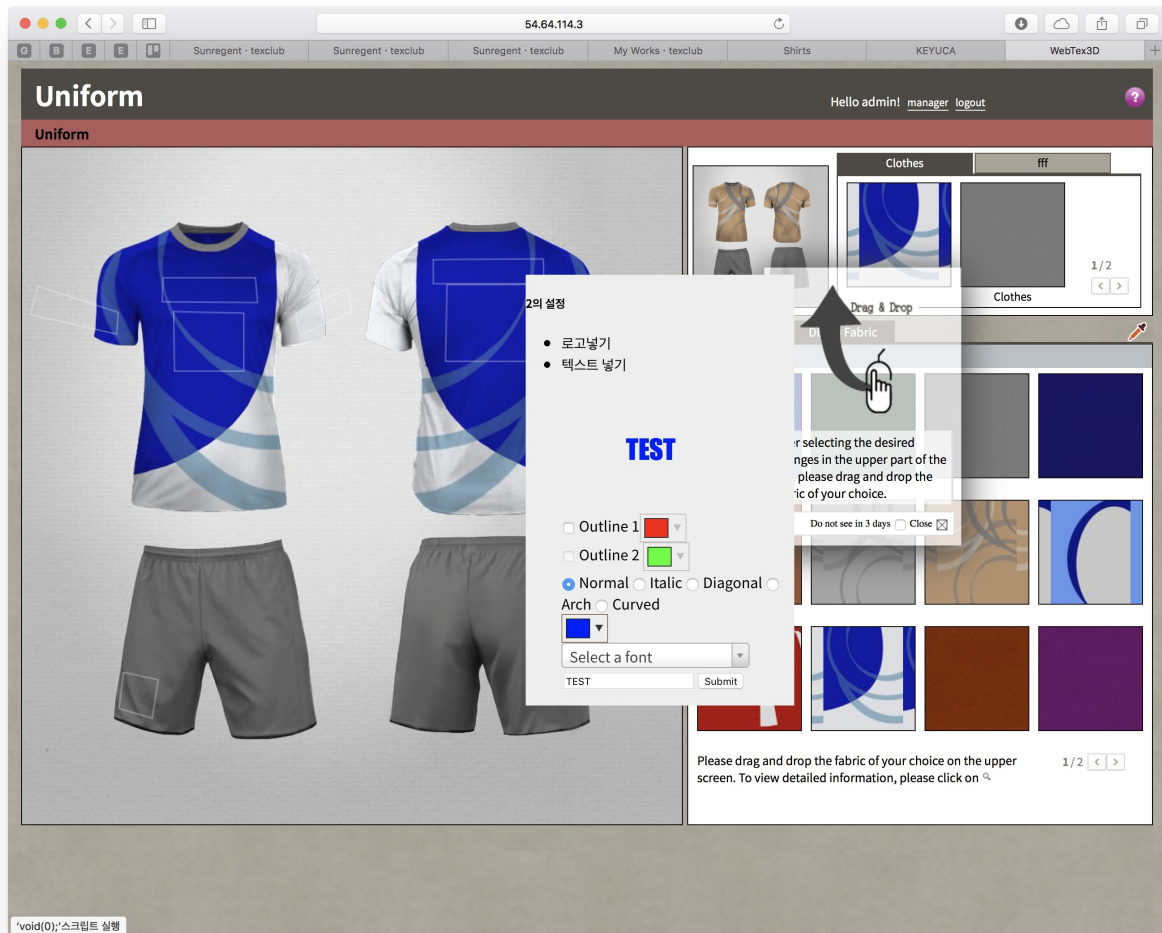
- 링크 : <http://mobile.stylefor.us>
- 참여자 (Web 한정) : PJ, KCW

## 개요

국책과제 App 개발을 위해 stylefor.us를 clone하여 만들어진 프로젝트입니다. App의 필수 기능 동작(시뮬레이션 등)을 위해 API화 되어 개발한 테스트 사이트입니다.

자세한 내용은 박진님께 확인해주세요.

# Uniform



- 참여자 (Web 한정) : 김찬울, 이하영

## 개요

Yuka 측에서 2016년 1월 중에 유니폼 관련으로 요청되어, Youngwoo에서 내부적으로 2016년 1월 18일부터 2016년 7월 7일까지 진행된, 그리고 현재는 중단된 프로젝트입니다.

Uniform 시뮬레이션을 진행하기 위해 WebTex3D 모듈을 사용하여 작업을 진행했으며, 레퍼런스 조사에 대한 자료는 다음 문서를 참조하세요.

개발 서버/server/current/Document/04. 프로젝트/33. WebTex3D/07. 일본Yuka/유니폼자료조사

## 요구사항 및 WebTex3D Base 에서 달라진 부분

WebTex3D를 크게 수정하지 않는 선에서 Uniform을 시뮬레이션하기 위한 필수 기능들 (로고 텍스트 삽입 등)을 작업하였다.

1. 특정 구역에 텍스트 및 이미지의 Mapping (스티커 기능)
  - i. 화면에 그리드 보여주기
  - ii. 텍스트의 폰트 설정
  - iii. 이미지 업로드
  - iv. 특정 Grid Group에 Mapping하기
2. ColorChip

## 위험요인 및 피드백

1. Uniform 개발에 현재의 WebTex3D는 절대적으로 적용하기에 좋지 않습니다.
2. 다양한 스타일의 변화를 위해서는 각 스타일별로 txd파일을 만들거나, webtex3d-shirts와 같은 방식으로 진행해야 합니다. 이는 매우 비효율적이며 경쟁사인 i-Designer에 비해 매우 밀립니다.

# Yuka-Alpha

Yuka-Alpha에서 계약된 WebTex3D 프로젝트를 다루는 Chapter입니다.

1. **KEYUCA** : 일본 가구 업체 Keyuca에서 계약된 Project, 2개 종류로 판매됨.
2. **Evegarden** : 일본 의류 업체 Evegarden에서 계약된 Project, 처음으로 JoyCART 연계 기능이 제작됨. 또한 최초로 상용 불가됨
3. **Fukushin Dress** : 일본 의류 업체 Evegarden에서 계약된 Project.
4. **Sunregent** : 일본 가구 업체 Sunregent에서 계약된 Project, 김찬울 사원이 본격적으로 참여함.
5. **Royal** : 일본 유니폼 업체 Royal에서 주문된 Project, 계획..

관련 문서로 `개발서버/server/current/Document/04. 프로젝트/33. WebTex3D/07. 일본Yuka` 폴더를 참고하십시오.

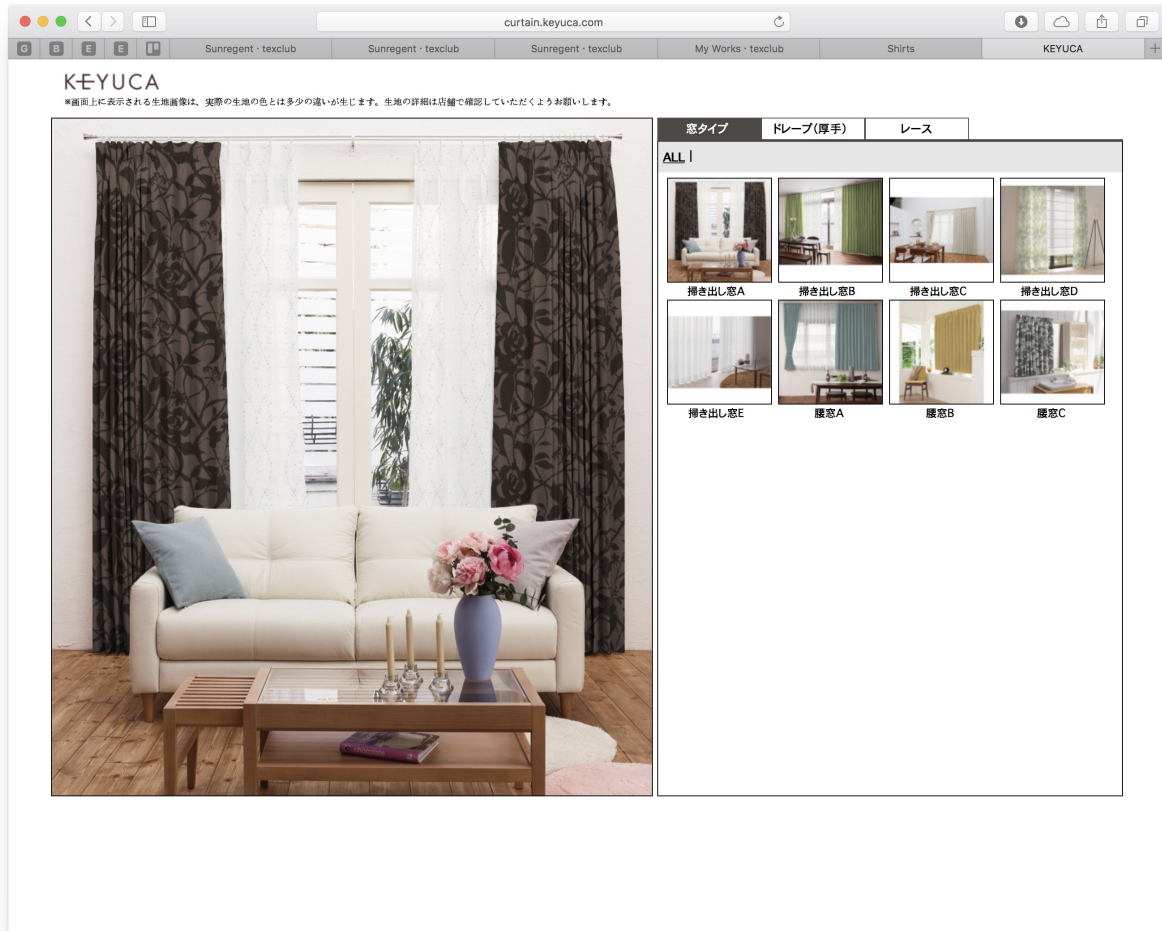
## 외부 인물

2,3,4,5번 인물의 경우 주로 LINE과 Mail로 의사소통을 진행을 합니다. LINE을 항상 커둠으로서 바로 응답할 수 있도록 합니다.

1. Hori : Yuka-Alpha 대표
2. Ohmori : Yuka-Alpha 개발자, 각종 프로젝트 설치 및 수정 진행.
3. Fueki : Yuka-Alpha 영업, 간단한 코드 수정까지는 가능. **KEYUCA**와 **Evegarden**, **Fukushin Dress**를 진행.
4. Sakakibara : Yuka-Alpha 영업. **Sunregent** 진행.
5. Takaki : Yuka-Alpha 영업. (원래는 개발자였다고 함) **Royal** 진행.

메일링시 반드시 참조(기존 메일들을 참고하세요)와 함께 일어번역을 추가해야 하며, 영업적인 논의가 필요한 부분은 반드시 류성훈 부장님과 함께 상의 후 전달되어야 합니다.

# KEYUCA



- Curtain 버전 : <http://curtain.keyuca.com>
- Furniture 버전 : <http://furniture.keyuca.com>
- 참여자 : 정영수(책임), 김찬울, 이가현

## 개요

Yuka 측에서 2014년 9월 5일 중에 요청되어, 2015년 2월 4일 납품된 Project인, 그리고 지속적으로 수정요청이 들어와 10월 중순까지 계속된 KEYUCA 를 개발한 자료입니다.

KEYUCA의 경우 Curtain 버전과 Furniture 버전이 따로 작성되었으며 두 서비스는 검색 항목 및 레이아웃의 차이가 다소 있습니다.

## 클라이언트 요구사항 및 WebTex3D Base 에서 달라진 부분

1. Mapping될 Grid Group에 따른 원단 카테고리 변경 (특정 원단은 특정 그룹에만 Mapping)
2. 카테고리 복사 기능
3. 검색 기능 (Curtain 버전) 및 검색 조건 설정 기능
4. 레이아웃의 변경

김찬울 사원은 3, 4번을 진행하였습니다.

## 각 기능 세부 설명 및 사용된 플러그인

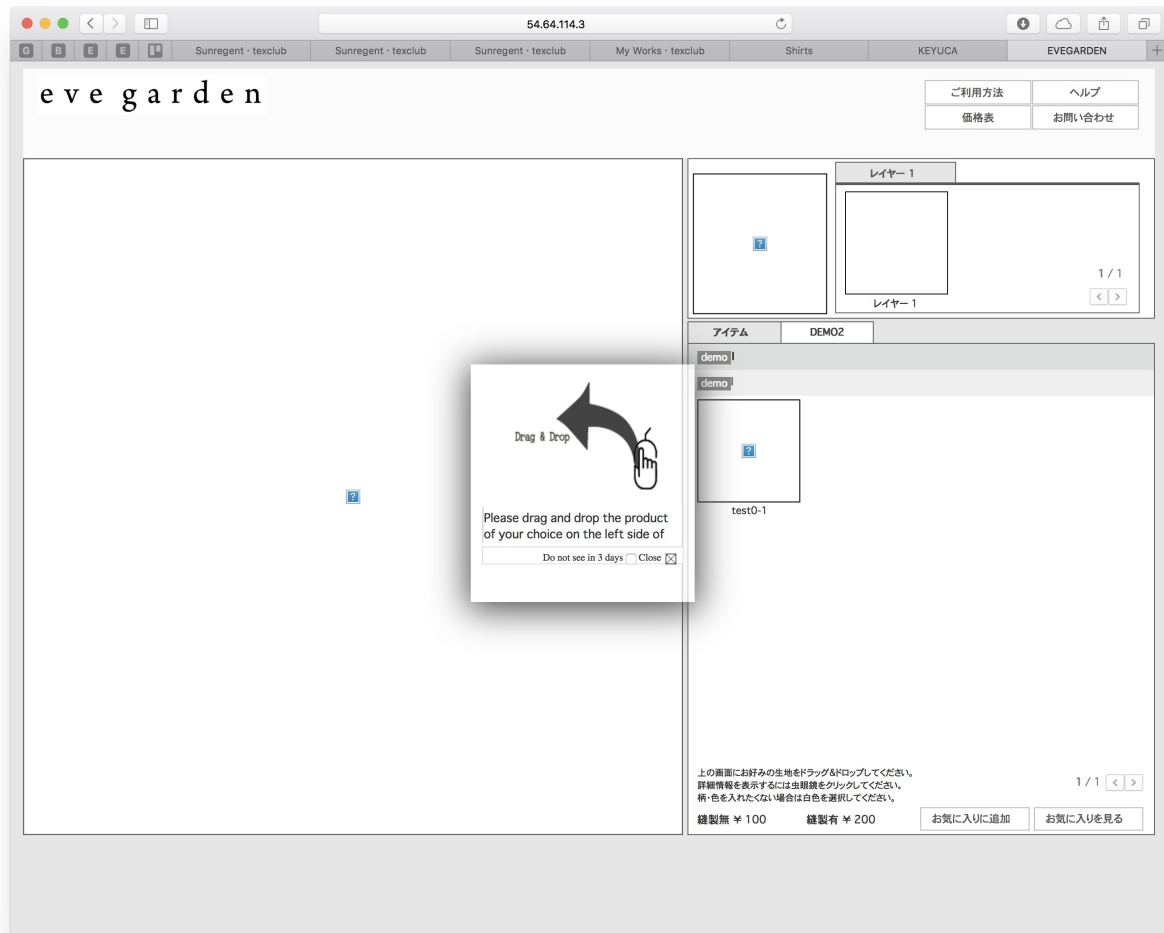
아래 링크들을 참조 부탁드립니다.

1. Mapping될 Grid Group에 따른 원단 카테고리 변경 (특정 원단은 특정 그룹에만 Mapping)
  - 정영수씨 인수인계 보고서 참고하세요
  - 참고로 [Fabric.stylefor.us](http://Fabric.stylefor.us)와 Keyuca의 방식은 좀 다릅니다. 써보시면 아실듯
2. 카테고리 복사 기능
  - 이가현씨 인수인계 보고서 참고하세요 진짜 핵급 쓰레기를 만들었다 카더라
  - 개발서버/server/current/Document/04. 프로젝트/33. WebTex3D/10. 개발문서/copy기능 매뉴얼 작성중 변경사항/copy 기능 관련 매뉴얼.odt 참조
3. 검색 기능 (Curtain 버전만) 및 검색 조건 설정 기능

## 위험요인 및 피드백

1. 입사 후 첫 작업인 만큼 검색 조건 부분의 코드가 간결하지 못함, 취약함.
2. 기존 WebTex3D의 코드를 간결하게 하지 못하고 시작한 커스텀 버전
3. Keyuca 본서버 설치 중 Manager에서 Fabric 등록사 UserItem으로 등록되는 문제 (2016년 1월 13일자로 해결, 커밋기록 0800501 확인바람)
4. CMYK Fabric 등록사 등록되지 않는 문제 (2016년 1월 11일자로 해결, 커밋기록 e84215e 확인바람)
5. 현재 Yuka-Alpha WebTex3D 수출 이후 처음이자 마지막으로 상용되고 있는 서비스 (다른 프로젝트들은 납품 후에도 실 서비스에 도입이 지연되거나 취소됨)

# Evegarden



- 참여자 : 김찬울(책임), 이하영, 정영수, 이가현

## 개요

Yuka 측에서 2015년 9월 중에 요청되어, 15년 11월 3일 부터 15년 11월 30일까지 1차 개발, 추후 Stand Alone System을 위해 16년 2월 5일부터 3월 18일까지 제작된 Project인, Evegarden 를 개발한 자료입니다.

## 클라이언트 요구사항 및 WebTex3D Base 에서 달라진 부분

1. My Favorite 기능을 만들어 원하는 시뮬레이션 디자인을 담는 기능

2. 제품 별 가격 등록 기능
3. My Favorite 창에서 담겨진 디자인 물품들의 수량, 옵션등을 토대로 전체 가격을 산정하는 기능
4. My Favorite 창에 담겨진 정보들을 외부 결제 화면으로 전달하는 기능
5. 4번의 기능이 이루어질 때 서버의 특정 경로에 해당 데이터들을 파일로 저장하는 기능
6. 인터넷 연결이 종료되어도 ~~Stand Alone~~ 한 상태로 기능이 작동되게 하는 기능 (도중에 요청된 부분으로, 가능하다고 하였으나 00월 00일자로 불가능을 판정)

## 각 기능 세부 설명 및 사용된 플러그인

아래 링크들을 참조 부탁드립니다.

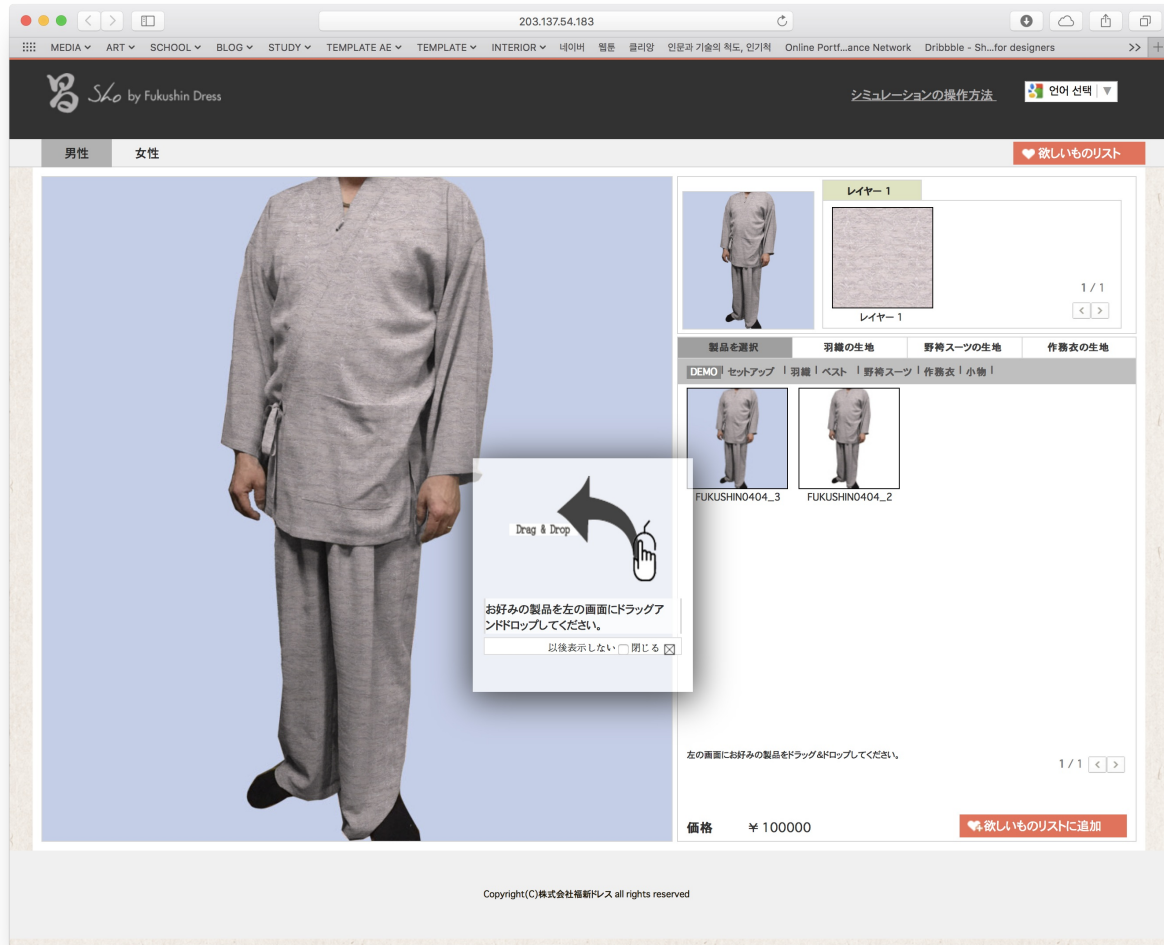
1. [Modal My Favorite](#)
2. [Check Total Cost](#)
3. [Save Simulation Data on CSV](#)
4. [Transfer Data to JoyCART](#)
5. [Stand Alone System \(+ restart BAT file\)](#)

## 위험요인 및 피드백

1. 납품은 되었지만 실제로 쓰이지 않게 됨 (그래서 Stand Alone 기술을 부탁하였으나 불가능함을 전달, 자세한 것은 문서 참조)
2. 4번의 사유로 사실은 ~~흑역사 프로젝트~~



# Fukushin Dress



- 참여자 : 김찬울(책임), 이하영

## 개요

Yuka 측에서 2015년 9월 중에 요청되어, 2016년 1월 26일 부터 2월 23일까지 제작된 Project인, Fukushin Dress를 개발한 자료입니다.

## 클라이언트 요구사항 및 WebTex3D Base 에서 달라진 부분

1. My Favorite 기능을 만들어 원하는 시뮬레이션 디자인을 담는 기능
2. 제품 별 가격 등록 기능

3. My Favorite 창에서 담겨진 디자인 물품들의 수량, 옵션등을 토대로 전체 가격을 산정하는 기능
4. My Favorite 창에 담겨진 정보들을 외부 결제 화면으로 전달하는 기능

## 각 기능 세부 설명 및 사용된 플러그인

아래 링크들을 참조 부탁드립니다.

1. [Modal My Favorite](#)
2. [Check Total Cost](#)
3. [Transfer Data to JoyCART](#)

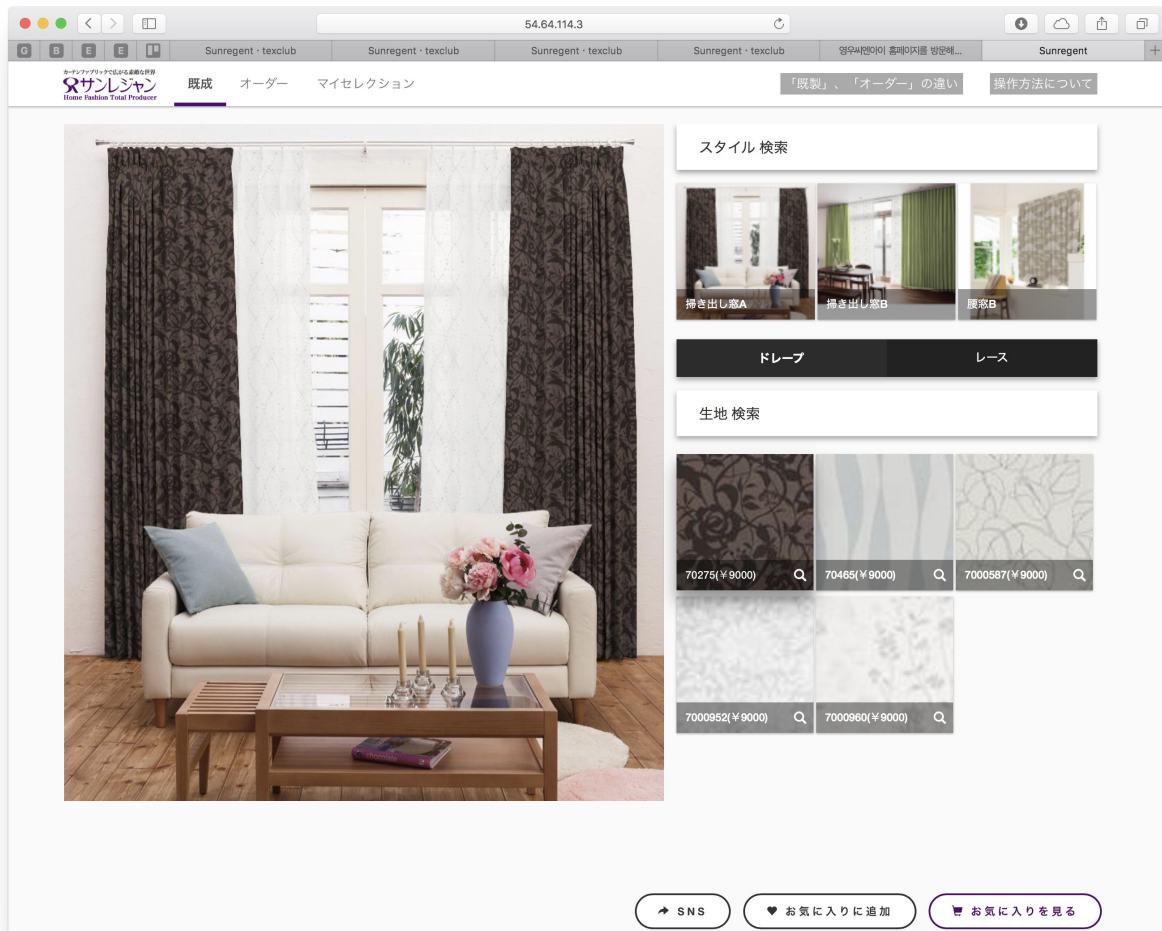
## 위험요인 및 피드백

1. JoyCART 결제 API의 난해함

JoyCart의 경우 API 가이드라인이나, 도움말 등이 확실치 않아 작업이 어려웠습니다. 한국의 쇼핑몰 API를 연결하는 것과는 차원이 다르므로 유의하시길 바랍니다.

2. 결제 정보 전달의 취급 (Backend에서 데이터 검증을 지키지 않아, 충분히 데이터가 해킹되거나 변경되어 전달될 가능성이 있음)

# Sunregent



- 참여자 : 김찬울(책임), 이하영, 박진

## 개요

Yuka 측에서 2016년 1월 중에 요청되어, 2월 18일 부터 6월 2일까지 제작된 Project인, Sunregent 를 개발한 자료입니다.

## 클라이언트 요구사항 및 WebTex3D Base 에서 달라진 부분

초기 개발 기획과, 추가 개발 요청에 따라 다음과 같은 요구사항이 있었습니다. (김찬울 사원 본인이 제작한 기능은 3~6번입니다.)

1. 카테고리의 변경 및 카테고리별 각기 다른 기능 적용
2. 뷰어에서의 매우 세부적인 검색기능
3. CSV로 가격표가 출력되고, 이미지 갤러리가 추가된 Fabric Information 페이지
4. SNS 공유기능
5. 즐겨찾기 등록 및 인쇄와 PDF 출력이 가능한 즐겨찾기 페이지
6. 스마트폰 레이아웃 지원 (반응형 레이아웃)

이후의 내용들은 김찬울 사원 본인이 제작한 기능만 설명합니다.

## 각 기능 세부 설명 및 사용된 플러그인

아래 링크들을 참조 부탁드립니다.

1. [Modal Favorite](#)
2. [CSV Price Table](#)
3. [Image Gallery on FabricInformation](#)
4. [SNS Feed](#)
5. [AJAX Loading bar](#)
6. [Not Hover on Mobile Device](#)
7. [Modal \(My Favorite\)](#)
8. [Print](#)
9. [Export to PDF](#)
10. [Responsive Layout](#)

## 위험요인 및 피드백

- 16년 7월 15일 추가적인 문의가 들어왔습니다. 이에 대한 정보는 Commit 정보 `35ef529`, `cb3510d`, `913ceac` 참조하세요.
- CSV 예외처리가 이루어지지 않았습니다.
- 이미지 갤러리는 항상 정사각형의 형태로 늘려져 보여지게 됩니다.
- 이미지 갤러리의 저해상도에 대한 예외처리가 되지 않았습니다.
- 공유하는 기능의 SID 예외처리가 되지 않았습니다. [SNS Feed](#)를 참조하세요.
- `Hover` 이벤트를 가진 `Element`의 `touch event` 오동작이 일어납니다. (두번 클릭해야함) 수정되었습니다. [Not Hover on Mobile Device](#) 를 참고하세요.
- PDF출력 기능이 불안정합니다. (속도가 느린 부분은 데이터 변환을 이유로 어쩔수 없습니다)
- Chrome이나 IE의 경우 스크롤 할 내용이 생기게 되면 가로 해상도가 변화되는데, 이 사이에서 계속 가로 해상도가 작아졌다 커졌다 하는 부분에서 레이아웃이 잘못 보여지는 문제. (1920~2400px 이상의 가로 해상도에서 문제가 생길수 있습니다.)



# Royal

- 참여자 : 김찬울(책임), 이하영, 박진

## 개요

Yuka 측에서 2016년 6월 중에 요청되어, 7월 7일까지 제작된 7월 12일자로 개발 중지된 Project 인, Royal Uniform 에 관련된 자료입니다.

## 클라이언트 요구사항 및 WebTex3D Base 에서 달라진 부분

1. WebTex3D에서의 유니폼

## 각 기능 세부 설명 및 사용된 플러그인

아래 링크들을 참조 부탁드립니다.

1. [Multi-part Simulation](#)
2. 특정 구역에 텍스트 및 이미지의 Mapping (스티커 기능)
  - i. [화면에 그리드 보여주기](#)
  - ii. [텍스트의 폰트 설정](#)
  - iii. [이미지 업로드](#)
  - iv. [특정 Grid Group에 Mapping하기](#)
3. ColorChip

## 위험요인 및 피드백

1. Yuka-Alpha와 서비스 진행 이래 최초의 계약 파토
2. **WebTex3D는 절대적으로 Uniform 시스템을 개발하기에 적절하지 않습니다.**

유니폼별 스타일을 모두 사진 촬영하기도, 포토샵으로 진행하기에도 무리가 있습니다.

또한 각 그리드를 잡기에도, 최근에는 세밀한 혹은 복잡한 작업을 요하는 fabric들이 많아 이를 일일이 색을 변경하는 과정을 진행하기도 어렵습니다.



# Elite

형지 Elite에서 계약된 WebTex3D 프로젝트를 다루는 Chapter입니다.

1. **Specification** : WebTex3D의 설치 사양 및 설치되는 소프트웨어의 버전 등을 서술합니다.
2. **업무분담** : 참여하는 회사별 담당자와 작업 구역에 대하여 서술합니다.
3. **Works** : 작업한 부분에 대한 내용을 서술합니다.

**관련 문서로** `개발서버/server/current/Document/04. 프로젝트/33. WebTex3D/11. 엘리트베이직` 폴더를  
참고하십시오.



# Specification

## WebTex3D를 설치하기 위한 조건

- Apache, Python이 작동하는 모든 OS군
- 최소 2GB 이상의 RAM
- 최소 100GB 이상의 HDD
- 항상 Online 연결

## WebTex3D를 설치하기 위해 필요한 소프트웨어

이전부터 계속 지적되었던 Django의 버전업을 이번 프로젝트를 통해 임시적으로 진행하였습니다. 하지만 1.8.X 버전을 사용하면서 South를 사용해야 하는 불편함은 여전합니다.

- Python 2.7.X
- Django 1.8.X
- PostgreSQL 9.2 이상
- south 0.8.4
- psycopg2-2.6.1.win32-py2.7-pg9.4.4
- Pillow-2.7.0.win32-pypy2.4-py2.7
- [SASS](#)를 지원하는 코드 에디터

더 효율적인 CSS 관리를 위하여 SASS를 사용하게 되었습니다. [외부 문서](#)를 통해 SASS를 사용하는 방법을 확인하세요. 기존에는 Sublime Text의 Plugin을 통해 Compiler했지만, JetBrains사의 코드 에디터를 통해 자동 컴파일을 네이티브로 진행할 수 있습니다.

설치법은 다음과 같습니다.

1. Ruby를 설치합니다. (Mac은 기본적으로 내장이 되어있으니 업데이트 후 사용하십시오)
2. Ruby의 gem (잼~ (°\_°)ㅎ) 를 이용하여 sass를 설치합니다.
3. (Sublime Text를 사용한다면) SublimeText3에서 Install Package를 이용하여 sass builder, sublimeonsavebuild 설치, 후에 프로그램 재시작
4. 샘플 css 파일을 만들어 저장 후 save, 이때 Sublime에서 Tools>BuildSystem>SASS로 체크되어있어야함.

## 업무분담

### Youngwoo CNI

- 총 담당 : H상무님
- (개발 및) C파트 담당 : LJ사원
- Web 파트 담당 : PJ사원 (KCW에서 변경)
- 디자인 파트 담당 : C팀장님

### 형지Elite

- 프로젝트 총괄 매니저 : BNY

### Di9

- 매니저 : LSH
  - 메인 페이지 기획 및 디자인, 퍼블리싱 담당

# Works

- 참여자 : KCW(retired), LHY, PJ(담당자)

## 개요

형지 Elite 측에서 2015년 10월 중에 요청되어, 2016년 4월 10일부터 10월 30일까지 개발 예정된 Project인, 형지 Elite 교복 시뮬레이션 시스템에 관련된 자료입니다.

## 클라이언트 요구사항 및 WebTex3D Base 에서 달라진 부분

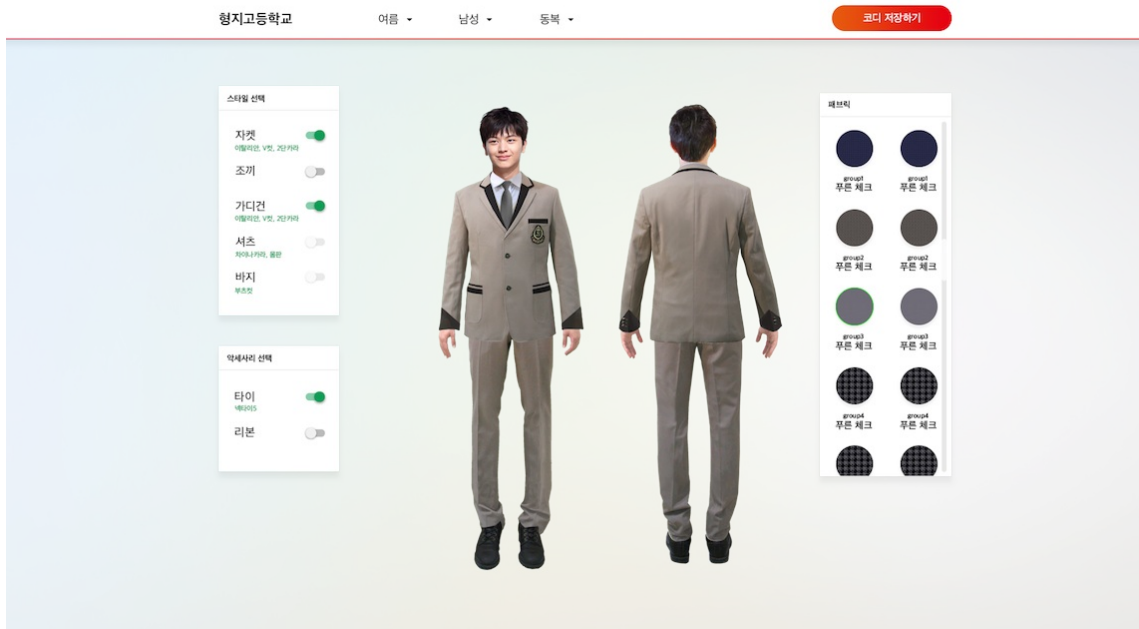
- Web에서 WebTex3D 모듈을 사용하지 않습니다.
- 시스템을 설계하는 수준의 프로젝트입니다.
- 사양이 변경되었습니다.
- Tex3DWe가 새로 제작되었습니다.

해당 프로젝트를 위해 새로운 개념으로 디자이너가 작업할 수 있는 Tex3DWe가 제작되었습니다. 폴더별 레이어 관리, Fabric Group의 개념 등장 등, 추가 개발이 C팀에서 진행되었고, 이에 따라 Upload하는 방법도 달라졌습니다.

## 각 기능 세부 설명

1. 메인 페이지
  - i. 로그인
  - ii. 엘리트 디자인 제안
    - i. 유형 선택
    - ii. 학교 유형
    - iii. 스타일 유형
    - iv. 검색 화면
    - v. 상세 페이지
  - iii. 전국 디자인 현황
    - i. 검색 화면
    - ii. 상세 페이지
  - iv. 마이 페이지
    - i. 학교별 제안
      - i. 학교 선택

- i. 엘리트 제안
      - i. 아이템 화면
      - ii. 상세 페이지
    - ii. 학교 관심 디자인
      - i. 아이템 화면
      - ii. 상세 페이지
    - iii. 채택 디자인
      - i. 아이템 화면
      - ii. 상세 페이지
  - ii. 상담 리스트
    - i. 상담내용 보기
- 2. 시뮬레이션 뷰어 페이지
  - i. 상세 페이지를 통한 시뮬레이션 페이지 진입



- ii. User 조작
  - i. Style의 Detail 선택에 따른 Data (이미지, Fabric Group 등)변경

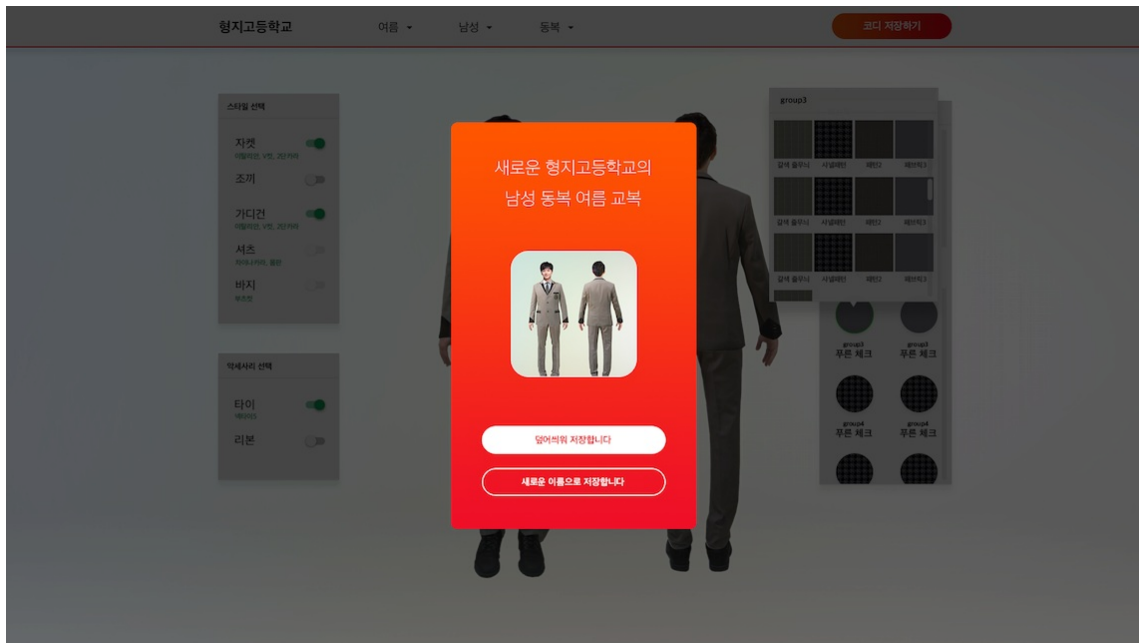


i. Fabric Group의 Fabric 선택에 따른 Data (이미지, 등) 변경



i. 남녀 공학인 경우 성별을 바꿔 불러올 수 있는 기능 (대신 작업하던 Data는 날아감)

iii. Save기능



- i. 새로운 이름으로 <학교 관심 디자인>에 저장하기
  - ii. <학교 관심 디자인>에서 불러온 경우 덮어 저장하기
- 3. 매니저 페이지
  - i. 회원 관리
    - i. 기본 기능 (관리자-총판-대리점, 디자이너)
    - ii. 추가 기능 (보안 강화)
  - ii. 메인 페이지의 조작 기능
    - i. 엘리트 디자인 제안의 아이템 설정
    - ii. 전국 디자인 현황 세팅
    - iii. 대리점의 담당 학교 설정
    - iv. 기타
- 4. Tex3DWe 연동
  - i. 업로드

## 위험요인 및 피드백

1. 의사소통의 어려움 (BNY과)
2. 의사소통의 어려움 (각 팀과)
3. 의사소통의 어려움 (팀 내부끼리)

# Texclub

이 Chapter는 Texclub 서비스의 내용을 포함하고 있습니다.

1. [Mail](#) : 영우씨엔아이의 메일 관리 지침 및 메일 서버 관리에 대한 내용을 서술합니다.
2. [Homepage](#) : 영우씨엔아이의 홈페이지 중 팝업창과 CMS2에서 제작했던 PHP파일에 대한 내용을 서술합니다.
3. [Install](#) : 인스톨 서버에 대한 내용을 서술합니다.

Texclub 서버는 LAMP (Linux + Apache + PHP + MySQL) 기반입니다.

[리눅스 서버에 대한 폴더 체계](#)를 참고하셔야, SFTP를 사용하시기에 수월할 것 같습니다.

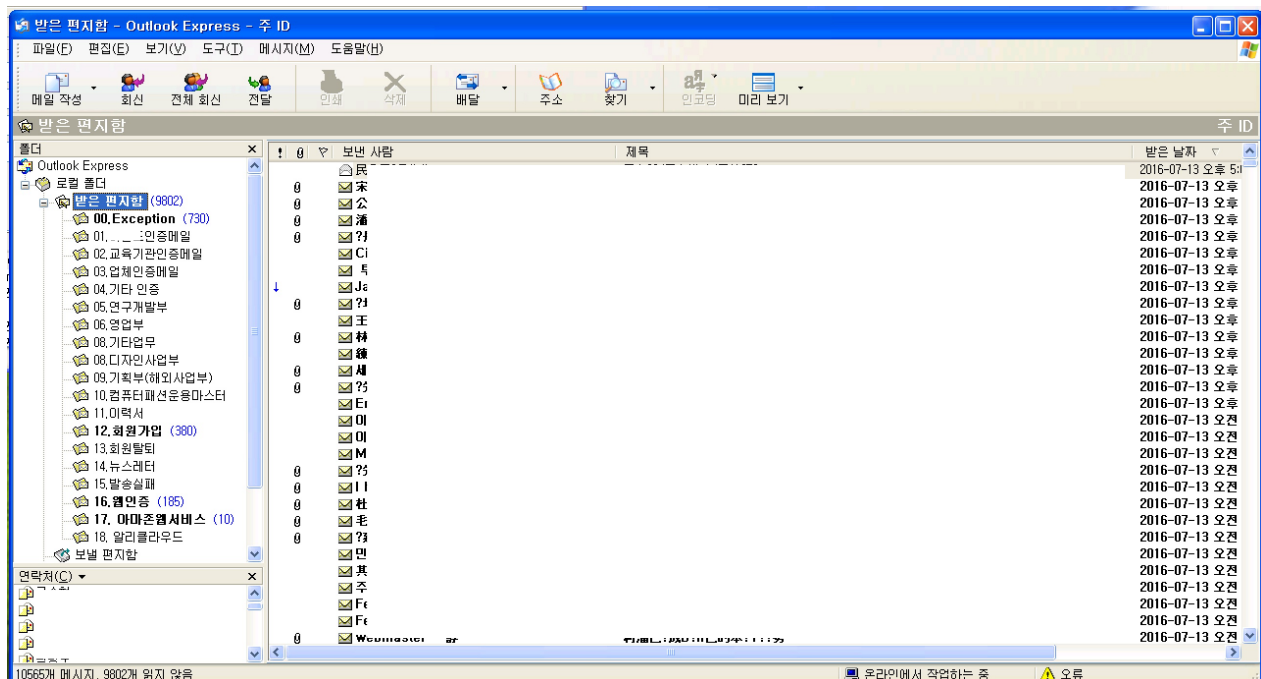
# Mail

이 문서는 Mail의 계정관리에 대하여 다루고 있습니다.

개발서버/server/current/Document/11. 인수인계/20. 개발 담당자 폴더/07. 박상훈/Server/TexClub.com  
메일서버관리.odt 문서와 함께 읽어주세요.

## webmaster@texclub.com 관리

Install 서버에 접속하여 Webmaster의 계정을 관리할 수 있습니다. Webmaster의 계정은 중요한 문서들이 자주 오기 때문에 매일 아침 10시, 저녁 6시 사이로 2번정도의 확인을 부탁드립니다.



중요한 내용을 포함한 메일은 다음과 같습니다.

- 전체보관함에 수신되는 메일 중 일반 사용자의 문의
- 폴더 1, 2, 3, 4 등의 인증메일 폴더

메일을 받게 되면 해당 담당자에게 메일의 내용을 전달하고 처리 방법을 물어야 합니다.

- 세금 계산 및 계산서 발행 관련 문의 : 김도영 과장님
- 프로그램 구매 문의 : 한승완 상무님
- 프로그램 사용 방법 문의 : 개발팀, 디자인팀
- 인증 문의 : 한승완 상무님과 확인 후 인증 과정 진행

스팸이 자주 오기 때문에 정리를 잘 해 주시고, 특히 악성 스팸 메일로 생각되어지는 메일은 읽지 말고 버려주세요. 메일 계정의 휴지통은 1주일에 한번 정도씩 비워줍니다.

## 용량 관리



간혹 Mail폴더의 용량이 증가하면서 메일 송수신에 문제가 생기는 경우가 있습니다. 잔여 공간을 만들기 위해서는 서버의 접속이 필요합니다.

서버로의 접속을 위해서는 SFTP 클라이언트를 사용해야 합니다. 접속 후 메일 용량을 관리하기 위해 개발서버/server/current/Document/11. 인수인계/20. 개발 담당자 폴더/09. 정영수/서버, 메일, 홈페이지 관리.odt 를 참조해 주세요.

## 향후 방침

현재 Mail 서버는 굉장히 불안정합니다. 또한 전체적인 연락처 관리, 송수신 관리 등이 어렵습니다. 마지막으로 메일의 백업 및 복원과 외부 클라이언트로의 접근도 어렵습니다. (제가 봤을때는 3~6개월 정도로 크리티컬한 이슈가 생길 것 같습니다. 용량 및 안정성을 위해 서버 담당자와 연락한 적이 있지만 제대로 전달이 안된 것 같습니다. 앞으로도 그럴 것 같구요.)

따라서 추후 문제가 생기기 전에 상급자에게 다음과 같은 제안을 부탁드립니다.

1. [Naver Works](#)로의 Migration (무료)
2. [Google for Work](#)로의 Migration (유료)

두 서비스 모두 큰 기업에서 진행하는 서비스다 보니 다음과 같은 장점이 있습니다.

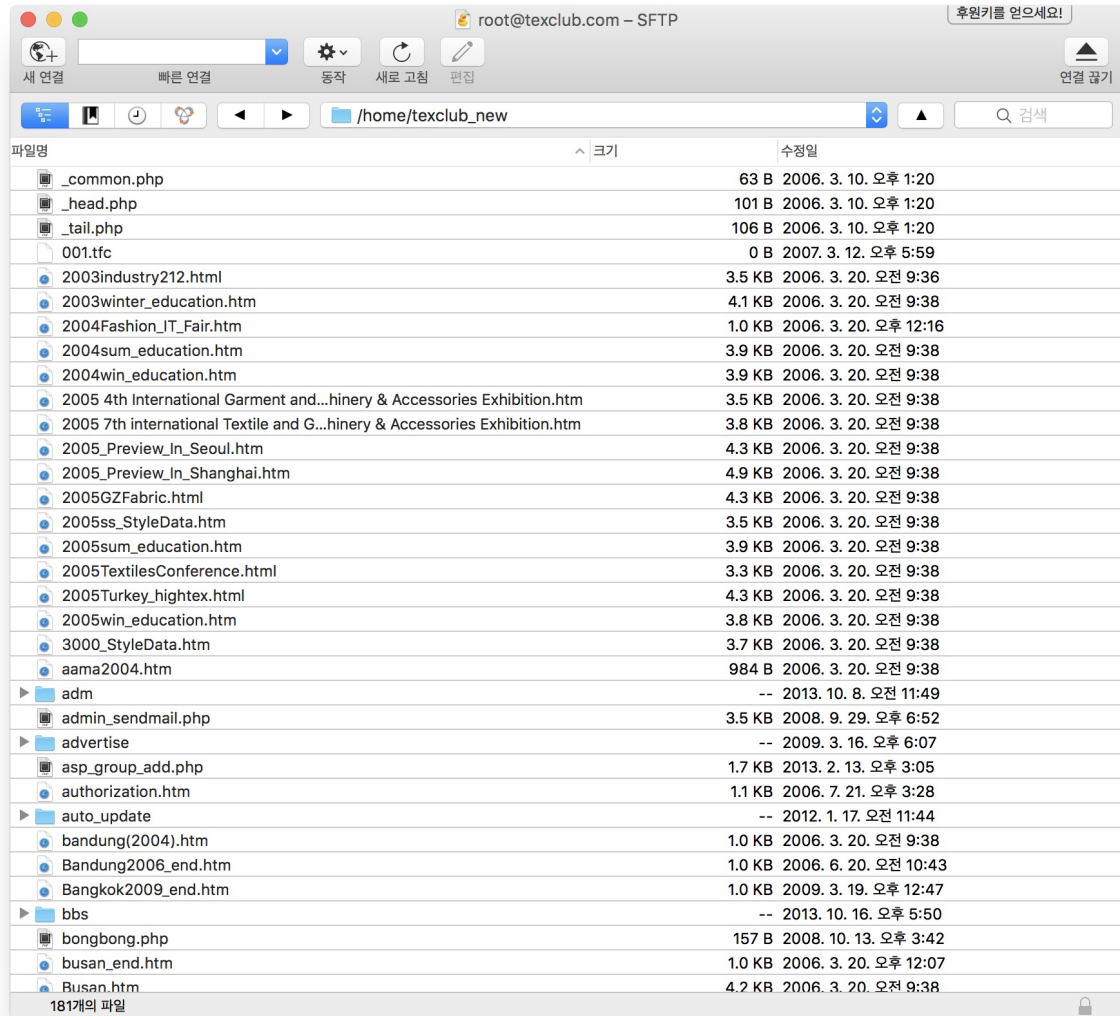
1. 안정성
2. 여러 서비스와의 연계
3. 외부 클라이언트에서의 접속
4. 백업 및 복원
5. 연락처 및 대용량 첨부 등 메일 기능의 장점
6. 다양한 외부 서비스(Naver Memo, N Drive, Google Word, Sheet, Google Drive)와의 연계로 인한 효율성 극대화

이 부분들을 재고하여 상급자 분들을 설득하고, 추후 위 시스템으로의 이주가 진행되었으면 합니다.

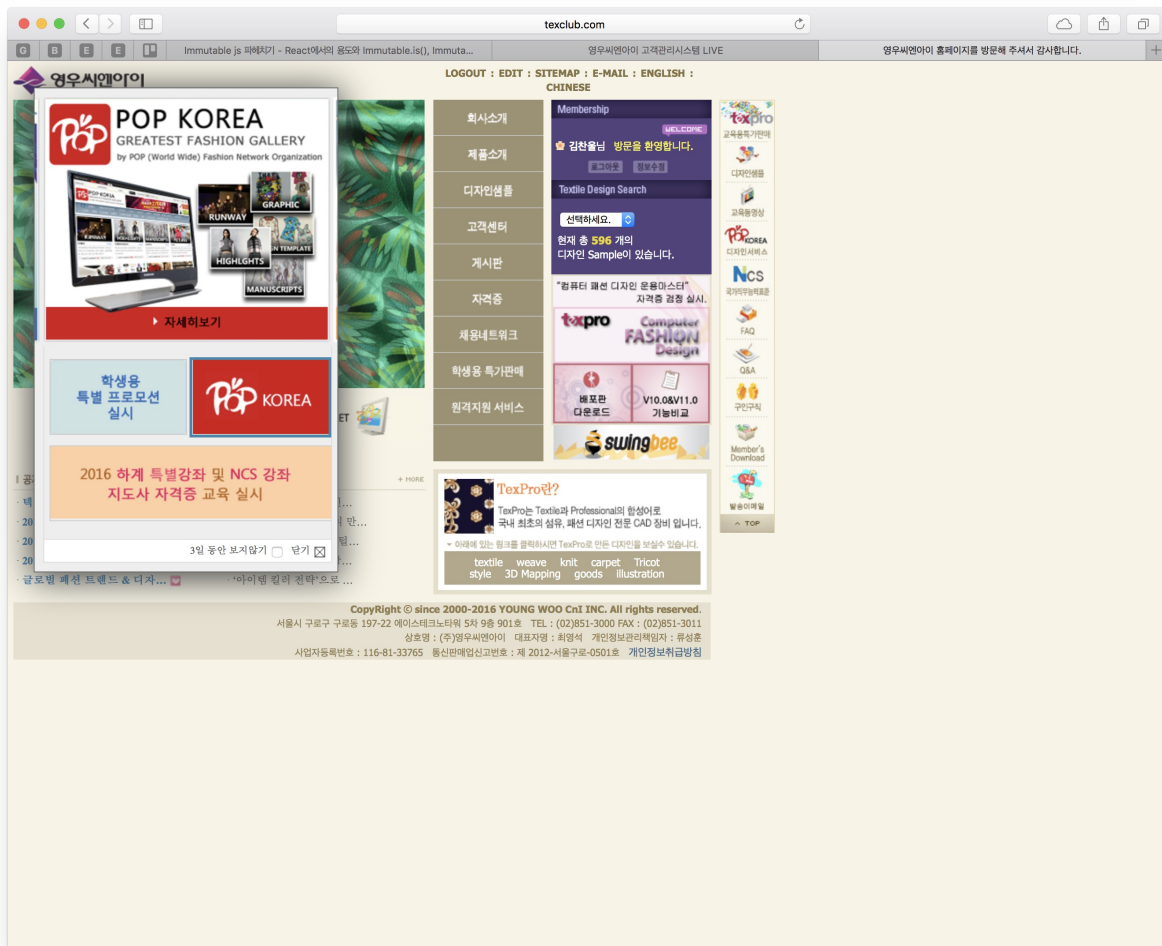
# Homepage

## Main Page

홈페이지 메인페이지입니다. 크게 제가 진행했던 부분은 없지만, 아는 부분 내에서 알려드립니다.



## Popup



홈페이지의 팝업창을 관리할 수 있도록 하는 방법입니다.

/home/texclub\_new/index.php

```
<!------- 팝업 시작 ----->
<?
/*
<팝업 수정시>
texclub.com 서버 /home/texclub_new/html/popup/new/ 안에 팝업용 슬라이드이미지(큰 이미지)
썸네일이미지(작은 이미지)를 넣고
밑에 팝업 추가용 예시 태그 1번, 2번에 맞춰서 링크주소, 이미지 주소를 알맞게 넣어주면 css js 추가
수정없이도 애니메이션 그대로 작동합니다.
위에 setcookie getcookie 함수에 들어가는 인수(날짜)도 바꿔줘야 기존에 '3일 동안 보지 않기' 체크
했던 유저도 새 팝업창을 볼 수 있습니다.
*/
/* $URL_1 = "http://book.naver.com/bookdb/book_detail.nhn?bid=7463472"; 이전 주소*/
$URL_1 = "/html/asp/intro.php";
$URL_2 = "http://www.texclub.com/bbs/board.php?bo_table=bbs_notice&wr_id=185";
$URL_3 = "http://www.texclub.com/bbs/board.php?bo_table=bbs_notice&wr_id=188";
?>
```

php문서를 읽다보니 다음과 같이 팝업을 시작하는 부분과 함께 주석으로 처리된 부분들이 있습니다. (친절하다)

이 부분에서 `URL_*` 에 대한 URL을 각각 할당해 줍시다.

```
<div class="popupLayer">
  <div class="popupContainer">
    <div class="popupMain">
      <a id="URL1" href="<?=$URL_1?>" target="_blank">

        
      </a>
      <a id="URL2" href="<?=$URL_2?>" target="_blank">
        
      </a>
      <a id="URL3" href="<?=$URL_3?>" target="_blank">
        
      </a>
```

Popup에서 상단 부분, 큰 thumbnail의 부분입니다. 각각 주소와 href, id를 잘 설정해주세요.

```
</div>
<div class="popupSub">
  <div class="popupThumb" id="thumb1">
    <div style="position:absolute;">
      <a href="<?=$URL_1?>" target="_blank">
        
      </a>
    </div>
  </div>
  <div class="popupThumb" id="thumb2">
    <div style="position:absolute;">
      <a href="<?=$URL_2?>" target="_blank">
        
      </a>
    </div>
  </div>
  <div class="popupThumb" id="thumb3">
    <div style="position:absolute;">
      <a href="<?=$URL_3?>" target="_blank">
        
      </a>
    </div>
  </div>
  ...
<!------- 팝업 끝 ----->
```

이 부분은 아랫 부분, 작은 thumbnail을 설정하는 부분입니다. 이 부분도 url과 img src를 적절하게 잘 세팅해줍니다. 새로고침을 진행하면, 해당 thumbnail의 selector 이동과 함께 상단의 콘텐츠 이미지도 변경되는 것을 볼 수 있습니다.

다만, 아직 한가지가 더 남았을 수도 있습니다. 가로가 긴 배너를 두거나, 혹은 일반 사이즈이거나.

### /home/texclub\_new/js/popup.js

```
function selMove(num) { // 선택터 이동시키기
    if(num == 3) $("#popupSelector").animate({width:314},50); // 3-1.jpg 이미지 두 배 크기일 경우 width:314}
    else $("#popupSelector").animate({width:155},50);

    if(num % 2) { // 홀수 - 1열일 경우
        $("#popupSelector").animate({left:3, top:13+((num/2).toFixed(0)-1)*selHeight},300);
    }
    else { // 짝수 - 2열일 경우
        $("#popupSelector").animate({left:165, top:13+(num/2-1)*selHeight},300);
    }
    // 선택터 URL에 현재 썸네일의 URL을 넣어줌
    $("#popupSelector a").attr('href', $('#URL'+num).attr('href'));
}
```

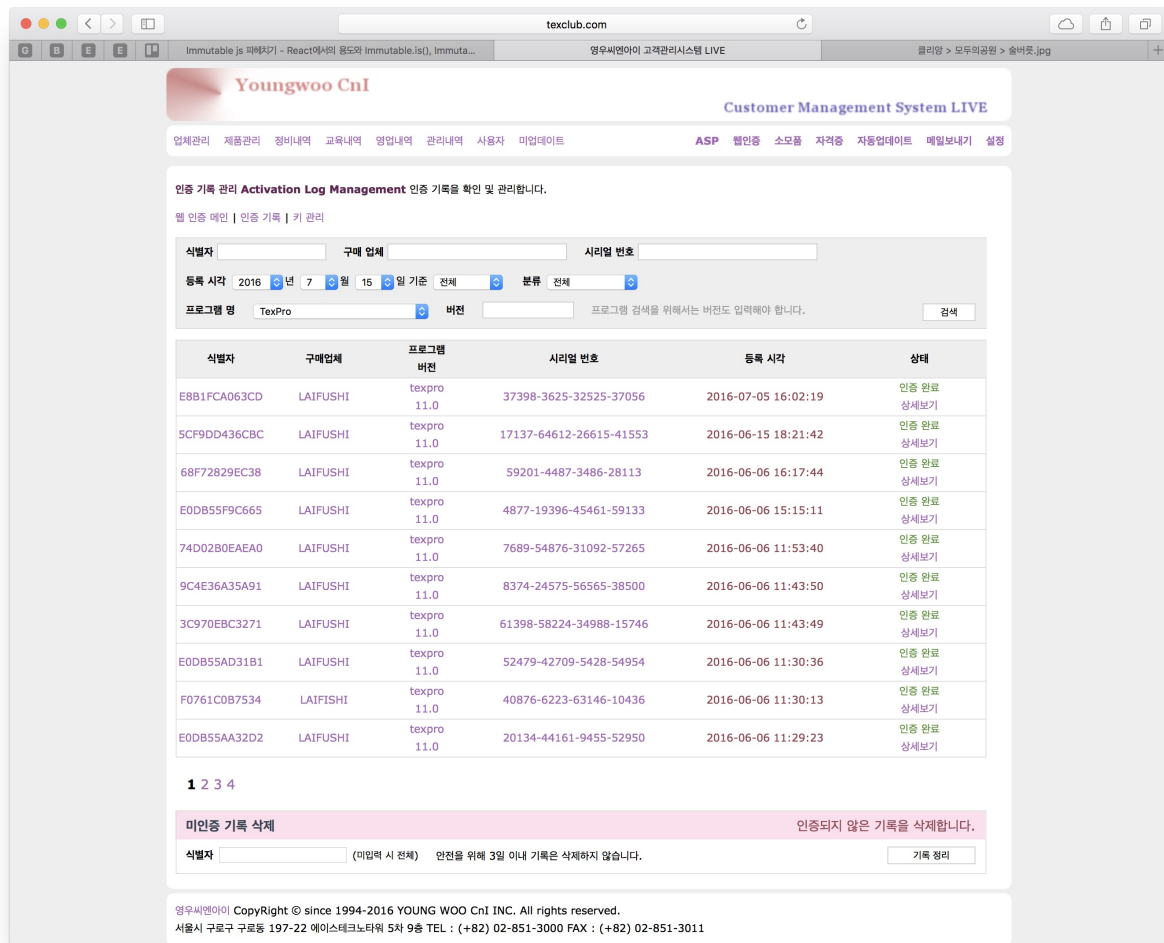
이를 해결할 수 있는 부분은 위 코드입니다. 움직이는 selector는 작은 thumbnail의 크기에 맞춰져 크기가 바뀌어야 합니다만, thumbnail의 너비가 2배사이즈일 경우 상단 코드의 2번째 줄 처럼, 따로 num을 지정하여 width:314px 로 지정해 주어야 합니다.

## CMS2

CMS2를 이용하여 여러가지 인증 및 고객정보 파악이 가능합니다. 사용 방법은 개발서버/server/current/Document/04. 프로젝트/09. CMS 고객관리/CMS2/20120105\_UserManual(Upgrade).odt를 참고해 주세요.

## 지아나 Hasp Web 인증

지아나 Hasp의 경우, 중국의 납품 프로그램 사용처에서 offline으로 사용하는 경우가 많아 특정 협약 하에 한번 online 인증 후, offline으로 진행할 수 있도록 제작된 인증 페이지입니다.

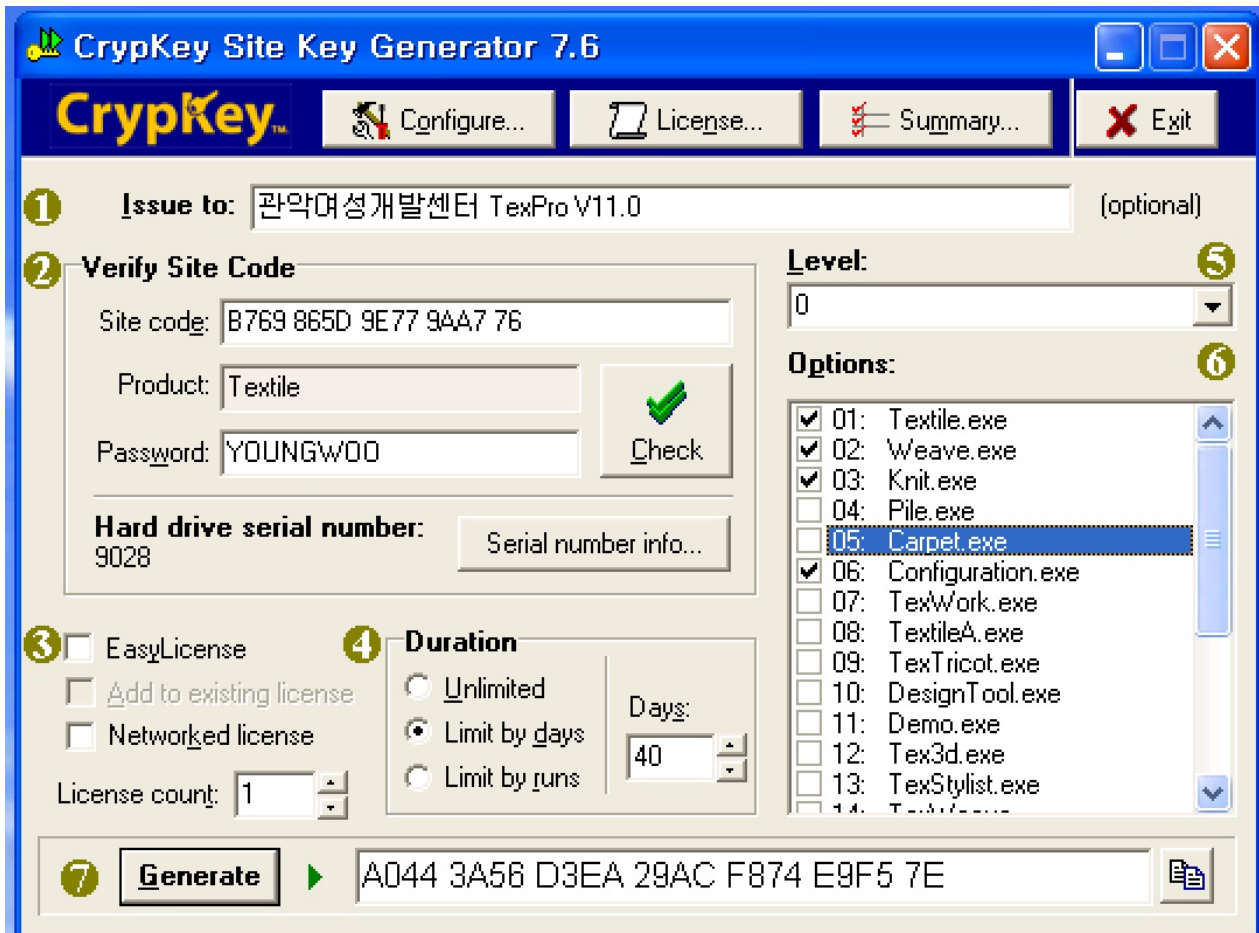


- activation\_web\_menu.php : 지아나 Hasp Web 인증의 모든 페이지에 include 되는 항목입니다. 각 페이지의 링크를 가리키고 있습니다.
- activation\_web.php : 지아나 Hasp Web인증의 메인페이지입니다.
  - activation\_web\_log.php : 지아나 Hasp Web인증의 메인페이지에 나타나는 로그를 삭제할 수 있습니다.
- activation\_web\_log.php : 지아나 Hasp Web인증의 log를 파악할 수 있는 부분입니다.
  - activation\_web\_confirm.php
  - activation\_web\_delete.php
  - activation\_web\_modify.php
- activation\_web\_key.php : 지아나 Hasp Web인증을 위하여 키를 수동으로 발급할 수 있습니다.
  - activation\_web\_submit.php : Key를 발급할 경우 진행되는 페이지입니다. 실제적으로 Database에 추가가 됩니다.

# Install

Youngwoo CNI에서 만든 프로그램을 Web인증할 때 사용합니다. 지아나 Web 인증의 경우 [Homepage 문서](#)를 참고해주세요.

## Web Autholization



1. 사용처나 기타 메모사항을 작성할 수 있습니다.
2. 사용처에서 발급 받은 코드를 붙여두는 공간입니다. 작성 후 Check버튼을 누릅니다.
3. 그대로 두시면 됩니다.
4. 인증의 기간을 입력합니다.
  - i. 보통 Limit by days로 날짜 기준 인증을 진행합니다. (30일정도)
  - ii. Unlimited로 진행하게 되면 1000Days로 찍히며, 프로그램은 이를 무제한 인증으로 이해합니다.
5. 그대로 두시면 됩니다.
6. 인증을 하려는 프로그램을 선택합니다. Textile, Weave, Configuration을 기본값으로 갖습니다.
7. Generate 버튼을 누르면 인증키가 발급됩니다. 복사하여 소비자에게 알려주시면 됩니다.





# Indexes

이 Chapter는 개발했던 기능들에 대한 내용을 포함하고 있습니다. 모든 문서는 다른 문서 (프로젝트 문서 등)로 부터 꼬리를 물고 있는 문서입니다.

## SNS Feed



트위터, 라인, 페이스북에 대한 공유기능입니다.

## 사용된 Open Source, 기술 URL

- [Python Module Requests](#)

## 사용 예시

코드 설명에 앞서서 SNS 공유 방법에 대하여 설명드립니다. 다음 내용을 유의하여 봐주세요.

1. SNS 공유의 경우 대부분의 SNS는 자체 API를 제공하고 있습니다.
2. SNS 공유를 하기위해서는 OG TAG라는 것을 이용하여 프리뷰 이미지, 프리뷰 타이틀과 콘텐츠를 작성할 수 있습니다.
3. SNS 공유를 위해 OG TAG를 사용할 때, 이미 서버쪽에서 해당 값이 렌더링된 상태여야 합니다.

3번의 경우 서버쪽에서 렌더링 된 상태라는 뜻은 SNS 회사의 검색 Robot들이 해당 페이지를 읽어 들일때, 이미 우리의 서버쪽에서 페이지가 완성되어있어야 한다는 뜻입니다. 프론트엔드쪽에서 JS 나 기타 꼼수를 활용하여 meta태그를 변경하여도, Robot은 이를 무시합니다. 따라서 Python-Django에서 해당 값들이 렌더링되어야 합니다.

Preview.html에서 렌더링을 위해 서술된 metatag 는 다음과 같습니다.

### templates/pages/preview.html

```
<title>{{ productName }}과 {{fabricNames}}의 조합으로 만든 작품</title>
<meta property="og:title" content="WebTex3D Simulation" />
<meta property="og:url" content="http://{{currentUrl}}/preview/{{simid}}" />
<meta property="og:type" content="website" />
<meta property="og:description" content="{{ productName }}과 {{fabricNames}}의 조합으로
만든 작품" />
<meta property="og:image" content="http://{{currentUrl}}{{imageUrl}}" />
<meta name="twitter:card" content="summary_large_image" />
<meta name="twitter:title" content="WebTex3D Simulation" />
<meta name="twitter:description" content="{{ productName }}과 {{fabricNames}}의 조합으로
만든 작품" />
<meta name="twitter:url" content="http://{{currentUrl}}/preview/{{simid}}" />
<meta name="twitter:image" content="http://{{currentUrl}}{{imageUrl}}" />
<meta name="twitter:site" content="@sunregent_jp">
```

이 프리뷰 페이지는 <http://.../preview/{simulationid}>로 접근이 가능합니다. simulationid로 접근시, 해당 시뮬레이션의 정보를 {{ }} 로 서술된 코드에 집어넣는 것입니다. 이 부분은 서버에서 일어나므로, 해당 로봇이 특정 sid의 프리뷰 페이지에 접근했을 때 정확한 OGTAG의 내용을 받아 올 수 있습니다.

simulation의 runPreview 함수에서는 다음과 같이 값을 html에 넘겨줍니다.

### simulation/views.py

```
def runPreview(request, sim_id):
    try:
        infoSimulation = Simulation.objects.get(id = sim_id)
        cache_str = infoSimulation.cache_str
        imageUrl = infoSimulation.image.url
        pid, fids = cache_str.split()
        fids = fids.split(',')
        fabricNames=[]
        productName = Product.objects.get(id = pid).name
        for fid in fids:
            fabricNames.append(Fabric.objects.get(id = fid).name)
    except:
        return HttpResponseRedirect('/viewer')

    fabricNames = ','.join(fabricNames)

    variables = {'simid': sim_id, 'productName':productName, 'fabricNames':fabricNames,
'imageUrl':imageUrl, 'currentUrl':request.META['HTTP_HOST']}

    return render_to_response('pages/preview.html', variables)
```

이를 위해 `import requests`라는 구문을 이용하여 `requests`라는 python api를 사용합니다. 리퀘스트된 `sid`에 따라 다른 페이지를 보여주기 위함입니다. 다른 프로젝트에서 이러한 기능을 사용하거나, 새로운 서버에 이 프로젝트를 설치하고 싶다면, 꼭 **requests**를 깔아주세요.

다시 한번 `Preview.html`을 봅시다.

`preview.html`에서는 다음과 같이 페이스북, 트위터, 라인으로의 공유기능을 스크립트로 작성하였습니다.

`preview.html`

```

<div id="fb-root"></div>

<script>
  (function (d, s, id) {
    var js, fjs = d.getElementsByTagName(s)[0];
    if (d.getElementById(id)) return;
    js = d.createElement(s);
    js.id = id;
    js.src = "//connect.facebook.net/ko_KR/sdk.js#xfbml=1&version=v2.5&appId=1551049345196475";
    fjs.parentNode.insertBefore(js, fjs);
  }(document, 'script', 'facebook-jssdk'));
</script>
<div id="sectionSnsButtons" style="text-align: center">
  <!-- Your share button code -->

  <a href="https://twitter.com/share" data-url="http://{{currentUrl}}/preview/{{simid}}" class="twitter-share-button" style="display: inline-block;">Tweet</a>

  <script>
    ! function (d, s, id) {
      var js, fjs = d.getElementsByTagName(s)[0],
          p = /^http:/.test(d.location) ? 'http' : 'https';
      if (!d.getElementById(id)) {
        js = d.createElement(s);
        js.id = id;
        js.src = p + '://platform.twitter.com/widgets.js';
        fjs.parentNode.insertBefore(js, fjs);
      }
    }(document, 'script', 'twitter-wjs');
  </script>
  <script>
    var url = 'http://{{currentUrl}}/preview/{{simid}}';
    url = encodeURIComponent(url);
    document.getElementById('sectionSnsButtons').innerHTML += ('<a target = "_blank" href="http://line.me/R/msg/text/?" + url + '" style="display: inline-block;"></a>');
  </script>
  <div class="fb-share-button" data-href="http://{{currentUrl}}/preview/{{simid}}" data-layout="button_count" style="display: inline-block; top: -4px !important;"></div>

</div>

```

위 스크립트는 `<body>` 에 들어가는 태그며, 각 SNS에서 제공하는 api를 사용하였습니다. 10번째 줄의 **appId**는 Sunregent의 고유 [FB AppID](#)입니다.

OG TAG가 잘 설정되었는지 확인하기 위해 Facebook에서 제공하는 [Sharing Debugger](#)를 해보세요!

## SNS API

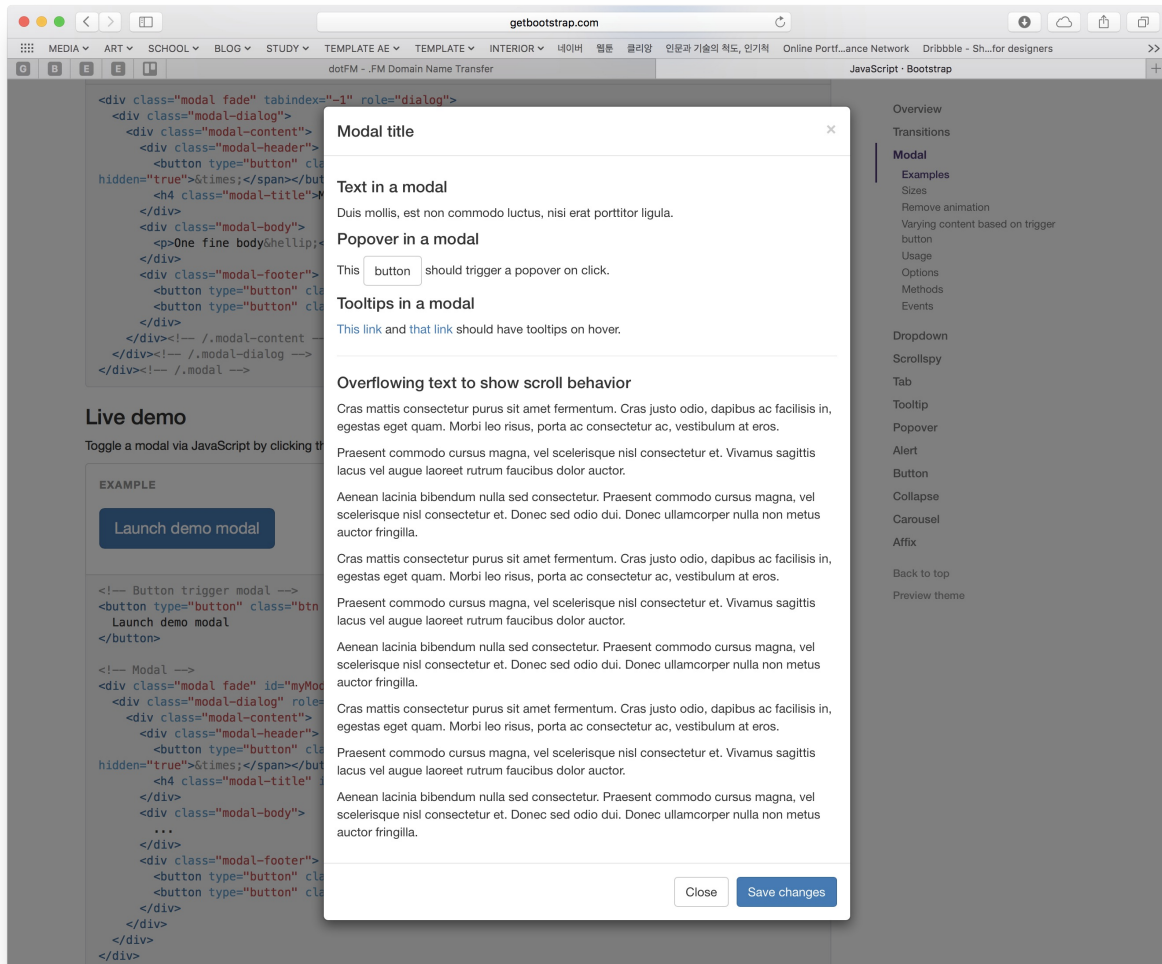
1. [Facebook for developers](#)
2. [Twitter Developers](#)
3. [Line for Media Operators](#)

## 위험 요소

1. Requests를 이용한 preview.html에 대하여 예외처리가 진행되지 않았습니다.
2. simulation id가 삭제되면 1번과 같은 오류가 발생될 수 있습니다.

# Modal (Bootstrap)

새창을 띄우지 않고 Popup을 띄우기 위하여 사용되었습니다. 이 부분은 Modal, Dialogue, Popup 등으로 불려집니다.



## 사용된 Open Source, 기술 URL

- [Bootstrap3](#)

## 사용 예시

### Basic

Modal을 사용하기 위해서는 기본적으로 Modal을 해당 페이지에 마크업 해야 합니다. Modal은 id로 호출이 되며, 다음 코드의 첫 줄에서 modal의 id를 설정하게 됩니다. 여기서는 myModal이라고 이름 짓겠습니다.

### viewer.html

```
<!-- Modal -->
<div class="modal fade" id="myModal" tabindex="-1" role="dialog" aria-labelledby="myModalLabel">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal" aria-label="Close"><span aria-hidden="true">&times;</span></button>
        <h4 class="modal-title" id="myModalLabel">Modal title</h4> /Modal의 타이틀입니다.
        해당 콘텐츠의 제목을 넣습니다.
      </div>
      <div class="modal-body">
        //Modal의 콘텐츠가 들어갑니다.
      </div>
      <div class="modal-footer"> //Modal의 하단입니다. 여기서는 액션을 취할수 있도록 버튼이 들어가네요.
        <button type="button" class="btn btn-default" data-dismiss="modal">Close</button>
        <button type="button" class="btn btn-primary">Save changes</button>
      </div>
    </div>
  </div>
</div>
```

그 다음엔, Modal을 불러올 수 있도록 버튼을 만들고, 액션을 넣어야 겠지요? data-toggle 을 통해 어떤 형태를 불러올 것인지, data-target 이라는 attr를 이용해서 어떤 id를 가진 modal을 불러올 것인지 설정합니다.

```
<!-- Button trigger modal -->
<button type="button" class="btn btn-primary btn-lg" data-toggle="modal" data-target="#myModal">
  Modal을 불러오는 버튼
</button>
```

이렇게 제작을 하게 되면, 눌렀을 때 #myModal 이라는 Modal을 불러오는 버튼이 보여지게 됩니다.

## Option

### 1. Modal의 사이즈 조정



기본적으로 Bootstrap은 3가지의 Modal 사이즈를 가지고 있습니다.

Modal의 Markup을 진행할때 modal-dialog 부분에 class를 추가함으로써 너비를 조절할 수 있습니다.

- `class="modal-dialog"` : **Default**, width가 598px인 modal
- `class="modal-dialog modal-sm"` : width가 298px인 modal
- `class="modal-dialog modal-lg"` : width가 898px인 modal

## 2. Modal의 Action 설정

Modal을 Markup하면서 Static한 Contents가 아니라 Dynamic한 정보를 삽입하고 싶다면 다음 방법을 이용합니다. Modal에게 어떤 행동을 시킬 때 여러가지 이벤트를 진행시킬 수 있도록 설계되었습니다. 간단하게 다음 5가지가 있습니다.

- `show.bs.modal` : Modal을 부르는 버튼을 클릭했을 때 (호출되었을 때)
- `shown.bs.modal` : Modal이 보여졌을 때 (css상으로 모두 드러났을 때)
- `hide.bs.modal` : Modal을 종료하는 버튼을 눌렀을 때 (종료되도록 시행되었을 때)
- `hidden.bs.modal` : Modal이 확실히 사라졌을 때 (css상으로 확연히 안보여 지게 되었을 때)
- `loaded.bs.modal` : Modal의 Contents가 모두 Load되었을 때 (자세한 사항은 Bootstrap Modal 문서 확인)

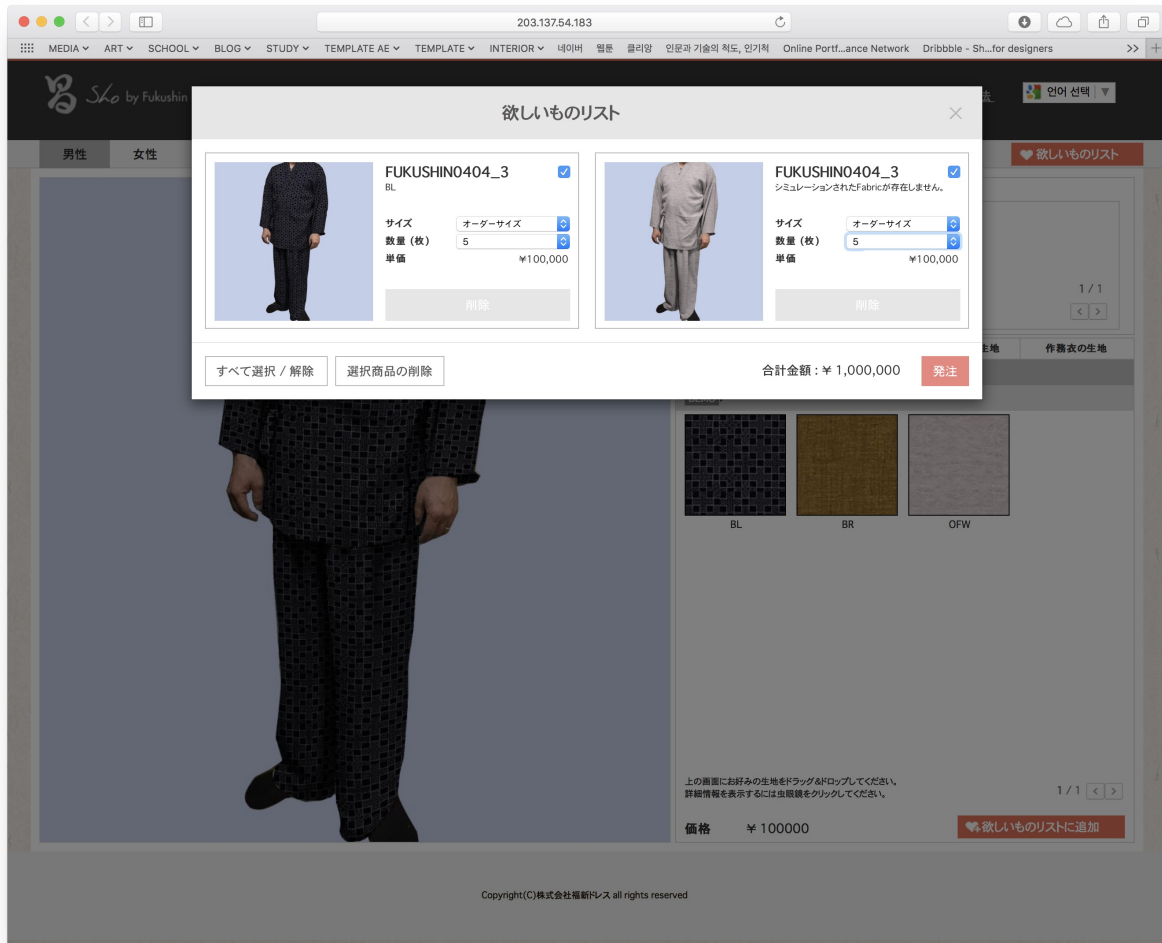
JS 코드는 다음과 같은 방법으로 사용됩니다.

```
$('#myModal').on('hidden.bs.modal', function (e) {
  // 실행될 이벤트 작성
})
```

이를 통해서, Modal이 show될 때 `modal-body` 에 아이템 리스트를 로드하여 출력한다던지, hidden되었을 때 modal의 내용을 비우도록 할 수도 있습니다.

# My Favorite

원하는 상품을 담아 두고, 확인하거나 세부 정보를 조절하여 다른 단계로 전송하는 기능을 포함하고 있습니다.



## 사용된 Open Source, 기술 URL

- [Modal](#) ([Bootstrap3](#)의 기능이 Add-on, My Favorite 문서를 읽기 전 확실히 참고 부탁드립니다)

## 사용 예시

### 기본 기능

## 1. Cookie에 저장된 아이템들을 불러오거나, 초기화하기

```
//product cookie의 존재를 확인하고 없으면 초기화합니다.
if (typeof ($.cookie('url')) == 'undefined') {
    ww.dataInCart = [];
} else {
    ww.dataInCart = JSON.parse($.cookie('url'));
}
```

## 2. 아이템 추가하기

```
// 30개 이상이면 안받아요, 왜냐하면 쿠키는 용량이 한정되 있거든요!
if (ww.dataInCart.length < 30) {
    var id = ww.dataInCart.length + 1;
    var found = ww.dataInCart.some(function (el) {
        return el.url === imgtodrag.attr('src');
    });
    if (!found) {
        ww.dataInCart.push({
            'url': imgtodrag.attr('src'), //url은 simulation image src, 이게 제일 식별
            'quantity': '1',
            'sewing': 'no',
            ...
        });
    } else {
        ...
    }
    $.cookie('url', JSON.stringify(ww.dataInCart), {
        expires: 14 //쿠키 14일 제한
    });
}
```

## 3. 담아둔 아이템 출력 기능

## 4. 아이템별 세부 설정

## 5. 삭제

### i. 아이템 삭제

### ii. 체크된 아이템 삭제

### iii. 모든 아이템 삭제

## 세부 기능

아래 항목들은 문서가 길어져 분리되었습니다. 해당 문서를 확인하십시오.

### 1. [아이템 전체의 금액 계산](#)

### 2. [CSV, txt 등의 데이터로 저장](#)

### 3. [PDF로 출력](#)

4. [Print Page](#) 출력
5. [JoyCART](#)로의 데이터 전송

# Check Total Cost

[My Favorite](#)에 담긴 각각의 Item을 합산하고, 앞에 일본 원화 (엔)표시를 위해서 다음과 같은 과정들이 진행되었습니다.

## 사용된 Open Source, 기술 URL

- [autoNumeric](#)

## 사용 방법

[Autonumeric](#)이라는, 강력한 숫자 표현 플러그인을 알려드립니다.

주로 다음과 같은 명령어가 사용되었습니다.

### 1. **init**

- 해당 input 박스에 `autonumeric`을 사용하기 위해 쓰이는 명령어입니다.
- `init`을 선언하면 그 후 이 DOM은 `Autonumeric`의 영향을 받습니다. 물론 값을 수정하기 위해서도 `autonumeric`을 거쳐야 합니다.
- `$(selector).autoNumeric('init', {options});`

### 2. **set**

- `autonumeric` 선언을 한 input에 값을 **설정**하기 위해서 사용합니다.
- `$(selector).autoNumeric('set', value);`

**fh.viewer.js**

```
function createSummary() {
    var orderCost = 0;

    $('.item-list').each(function () {
        quantity = $(this).find('select[name="quantity"]').val();
        sewing = $(this).find('select[name="sewing"]').val();
        price = $(this).find('input[name="price"]').val();
        console.log(price);
        $(this).find('.span-price').autoNumeric('init').autoNumeric('set', price)
        checked = $(this).find('input[name="product"]').prop('checked');
        total = (quantity * price * checked);
        orderCost += total;
    });

    console.log(orderCost);
    $("#currentCost").autoNumeric('init',{ mDec: '0', wEmpty: 'zero'}).autoNumeric('set', orderCost);
}
```

각 아이템의 가격을 표기하기 위해 `$('.span-price')` DOM에 `autoNumeric('init')` 으로 `autonumeric`을 사용함을 선언합니다. 또한 `autoNumeric('set', price)` 으로 각각의 가격을 표기합니다.

마지막으로 각 아이템의 `sewing`, `quantity`등의 option과 `price`를 종합하여 가격을 체크하여 `#currentCost` 에 가격을 표기합니다. 이때, `init`하면서 `mDec`을 이용하여 소수점 자리가 나타나지 않도록하고, 기본값으로 0을 표기하도록 설정합니다.

**fh.viewer.js** 또한 markup에서도 다음과 같은 과정을 요구합니다.

```
<input class="span-price" type="text" name ="confirmedPrice"
    data-a-sign="₩"
    data-m-dec="0"
    data-w-empty="zero"
    style="...."
    value="0"
    readonly>
```

각 attr는 다음 뜻을 의미합니다.

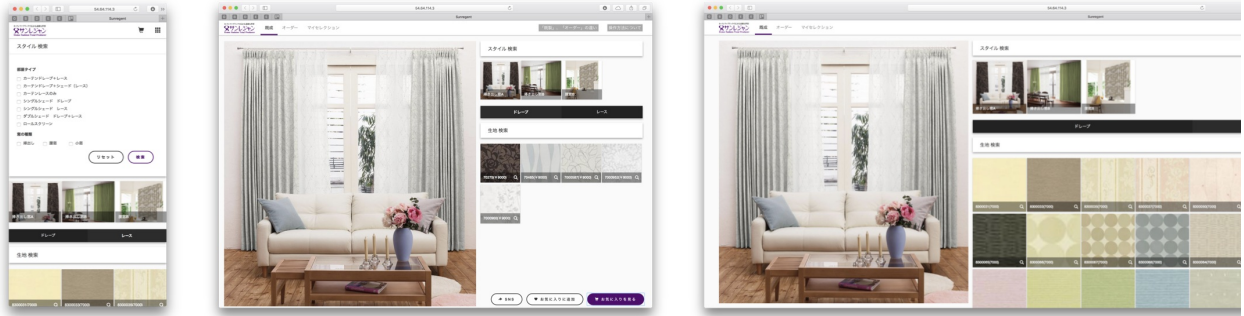
- **data-a-sign** : autonumeric이 사용하는 attr입니다. 통화를 표기하기 위해 쓰이며, 원하는 기호를 value로 사용합니다.
- **data-m-dec** : autonumeric이 사용하는 attr입니다. 소수점 단위 표기를 방지하기 위해 value로 0을 넣었습니다.
- **data-w-empty** : autonumeric이 사용하는 attr입니다. 값이 null일 경우 기본값을 어떻게 채워 넣을지 쓰는 attr이며 value로 기본값 0 을 나타내기 위해 'zero'를 사용하였습니다.
- **readonly** : 사용자가 임의로 가격 input을 수정하는 것을 막기 위해 `readonly` markup으로 수

정을 불가능하게 합니다.

## 문제점

1. 가격 등 중요한 정보는 반드시 서버에서도 체크해야 합니다.
  - Front-end단에서만 가격의 유효성을 체크하는 것은 매우 어렵습니다.
  -

# Responsive Web



Device별 해상도에 맞춘 레이아웃을 보여주기 위해 반응형 웹 기술을 사용합니다.

## 사용된 Open Source, 기술 URL

- [Web Responsive 개념](#)
- [Media Query](#)
- [Bootstrap Grid System on Media Query](#)

## 사용 예시

다음과 같이 브라우저의 가로 해상도를 이용하여 레이아웃을 다르게 뿌려줍니다.

### theme.css

```
/* Small devices (tablets, 768px and up) */

@media (min-width: 768px) {
    ...
}
```

theme.css에서 CSS의 media query인 min width를 이용하여 원하는 레이아웃을 짤수 있습니다.

이 때, CSS는 작성된 순서대로 Overwriting이 됩니다. 예를 들어 10번째 줄에 width:100px을 하고 11번째 줄에서 width:200px를 하면, 최종적으로는 200px이 표현됩니다. 이에 따라 media query min width를 작성할때는 다음과 같이 작성하세요.



```
.div {  
    //모바일, PC공통으로 보여지는 스타일  
    width:100px;  
    background-color:#000;  
}  
@media (min-width: 992px) {  
    div{  
        //992px이상에서 앞에 서술한 div 에 덮어쓸 css  
        width:200px;  
    }  
}  
@media (min-width: 1200px) {  
    div{  
        //1200px이상에서 앞에 서술한 div 에 덮어쓸 css  
        background-color:#fff;  
    }  
}
```

- 이 경우 모바일에서는 width 100px, 검은 배경,
- 992px이상의 가로 해상도에서는 width 200px, 검은 배경,
- 1200px이상의 가로 해상도에서는 width 200px, 하얀 배경의 div를 보여줍니다.

이 외에도 [Bootstrap](#)을 사용하고 있는 경우, 자체적으로 내장된 element가 기본적으로 responsive하게 설정되어 있으므로 쉽게 반응형 웹페이지를 구현할 수 있습니다.

하지만, 개발이 선행되기 전에 각 레이아웃별 디자인과 구성 요소가 확실하게 정해져 있어야 합니다.

# Print Cost Table from CSV Data

서버에 저장된 CSV Data를 Web에 보여주기 위해 다음과 같이 진행하였습니다.

## 사용된 Open Source, 기술 URL

- [Papa Parse](#)

## 사용 예시

CSV파일이 다음과 같이 있다면

```
,巾既製・1.5倍ﾀﾞｲ,,,  
丈,,100 c m(2枚),150 c m (2枚) ,200 c m (1枚)  
,105cm,"\\4,180",,  
,110cm,"\\4,180",,  
,120cm,"\\4,380",,  
,135cm,"\\4,580","\\6,880",  
,150cm,"\\4,780", ,  
,178cm,"\\4,980","\\7,480","\\4,980"  
,185cm,"\\5,130",,  
,190cm,"\\5,280",,  
,195cm,"\\5,430",,  
,200cm,"\\5,580","\\8,380","\\5,580"  
,205cm,"\\5,730", ,  
,210cm,"\\5,880",,  
,215cm,"\\5,980","\\8,980","\\5,980"  
,225cm,"\\6,230",,
```

위 데이터는 다음과 같이 변경되어 출력됩니다.

## Cost Table

丈	巾 既製・1.5倍ビタ			
		100 c m(2枚)	150 c m (2枚)	200 c m (1枚)
	105cm	\4,180		
	110cm	\4,180		
	120cm	\4,380		
	135cm	\4,580	\6,880	
	150cm	\4,780		
	178cm	\4,980	\7,480	\4,980
	185cm	\5,130		
	190cm	\5,280		
	195cm	\5,430		
	200cm	\5,580	\8,380	\5,580
	205cm	\5,730		
	210cm	\5,880		
	215cm	\5,980	\8,980	\5,980
	225cm	\6,230		

이를 구현하기 위해서는 Fabric 등록시 CSV 관련 값을 받아 저장해야합니다.

`fabricManager.html` 에서 `csvUploader`를 통해 다음과 같이 전송합니다.

**fabricManager.html**

```

var csvUploader = new qq.FileUploaderBasic({
  params: {
    csv : true
  },
  allowedExtensions: ['csv'],
  element: $('#detail_input .qqUploader')[0],
  button: $('#change_csv')[0],
  text: '',
  action: '/manager/fabric/uploadtempfabric',
  onComplete: function (id, filename, response) {
    if (response['success']) {
      $("#a_fabricCSV")
        .data('src', response['filepath'])
        .attr('href', response['filepath'])
        .show();
    } else {
      switch (response['reason']) {
        case 'FileSizeExceeded':
          alert('File size can not exceed 10MB.');
```

`createManagerFabric` 에서 다음과 같이 Python File Read를 이용하여 django를 통해 Save합니다.

### **fabric/views.py**

```

if request.POST["csv"]:
    CSV_PATH = media_url2path(csv)
    CSV_TYPE = imghdr.what(CSV_PATH)
    file = open(CSV_PATH)
    djangoFile = File(file)
    fabric.csv.save('data/fabric_csv/%s/%s.csv' % (str(fabric.id/10000), str(fabric.id
    )), djangoFile)
```

여기서 CSV는 `data/fabric_csv/0/fabricID.csv` 로 저장되며, DB에서는 해당 csv의 주소값이 저장됩니다. 이는 추후 Viewer에서 Fabric의 CSV를 불러올 때 사용됩니다.

`fh.viewwr.js` 에서 다음과 같이 Table을 만들고 CSV로 이루어진 가격표를 구성합니다. `.table-csv` 는 fabric Information 페이지에 속한 DOM의 Classname이며, CSV성분 중 `,` 와 `\n` 에 따라 테이블을 만들어 나갑니다.

이 과정에는 좋은 CSV Parser인 Papa parse를 이용합니다. [문서](#)를 참고해주세요!

### fh.viewer.js

```
//table-csv의 구역을 깨끗히 해줘요~
$(".table-csv").html('');
//Papa라는 좋은 CSV Parser를 이용합니다.
Papa.parse($('csv', detailInfo).text(), {
  download: true,
  //데이터가 모두 로드되었을때 표를 만들거예요!
  complete: function(results) {
    if (results.errors.length == 0) {
      $.each(results.data, function(i, el) {
        var row;
        //1번째 줄인 경우엔 테이블 헤더를 의미하겠죠? 컴공에게 0은 1이죠!
        if (i == 0) {
          row = $("<thead>");
          $.each(el, function(j, cell) {
            if (j == 0) {
              row.append("<th style='width:50px;'>" + cell + "</th>");
            } else if (cell !== "")
              row.append("<th colspan='4' />").text(cell));
          });
          // 이런 부분은 switch문으로 작업해도 될텐데요!
        } else if (i == 1) {
          row = $("<tr/>");
          $.each(el, function(j, cell) {
            row.append("<th/>").text(cell));
          });
          // 3번째 줄에 와서야 thead를 닫아줘요. 이제는 컨텐츠가 들어갈 거예요!
        } else if (i == 2) {
          row = $("<tr/></thead>");
          $.each(el, function(j, cell) {
            if (j == 0) {
              row.append("<th/>").text(cell));
            } else {
              row.append("<td/>").text(cell));
            }
          });
          // 이제 컨텐츠 구문만 작성될 거예요!
        } else {
          row = $("<tr/>");
          $.each(el, function(j, cell) {
            if (j == 0) {
              row.append("<th/>").text(cell));
            } else {
              row.append("<td/>").text(cell));
            }
          });
        }
      });
    }
  }
});
```

```
        });  
    }  
    $(".table-csv").append(row);  
    });  
} else {  
    // 데이터 없으면 출력을 이렇게 합니다.  
    $(".table-csv").append('<h4>There Is No Price Data.</h4>');  
}  
}  
});
```

# Stand Alone System (+ restart BAT file)

새창을 띄우지 않고 Popup을 띄우기 위하여 사용되었습니다. 이 부분은 Modal, Dialogue, Popup 등으로 불러집니다.

## 사용된 Open Source, 기술 URL

- [Curl](#)

## 사용 예시

### Local Use jQuery, JS plugin

jQuery를 CDN 참조가 아닌 Local에서 사용하도록 하였습니다. (jQuery 외에도 CDN으로 사용하던 부분들을 Local 파일 참조로 변경하였습니다)

## BAT 사용방법

Stand Alone System을 사용함에 따라 간헐적으로 Simulation Error가 생겼고, 이를 타개하기 위해 WebTex3D Simulation 프로세스를 강제종료하는 방법이 필요하게 되었습니다.

Apache Server와 WebTex3D Engine을 재시작하는 실행파일을 소개합니다.

1. curl.exe를 c:\Windows\system32\에 이동하십시오.
2. rebootEngineApache.bat을 c:\에 이동하십시오.
3. rebootEngineApache.bat을 notepad로 여십시오.
4. 2,3번째 line의 apache경로를 설치하려는 PC의 apache경로로 수정하십시오.
5. 4번째 line의 URL을 webtex3d의 URL로 수정하십시오.
6. 수정된 rebootEngineApache.bat을 저장하고 notepad를 종료하세요.
7. rebootEngineApache.bat의 바로가기를 만듭니다.
8. 만들어진 바로가기를 오른쪽 클릭하여 <속성>으로 들어갑니다.
9. 속성에서 <관리자 모드로 항상 실행> 을 체크하고 저장합니다.
10. 이제 만들어진 바로가기를 바탕화면 등, 원하는 위치에 두어 사용하면 됩니다.

바로가기의 원본은 다른 곳으로 이동되면 안됩니다.

## BAT 내용

**rebootEngineApache.bat**

```
taskkill /f /im TEX3D_SIM.EXE
"C:\Program Files (x86)\Apache Software Foundation\Apache2.2\bin\httpd" -k stop
"C:\Program Files (x86)\Apache Software Foundation\Apache2.2\bin\httpd" -k start
curl -k http://localhost:8002/runtex3dsim
```

우선 하나씩 살펴보자면,

```
taskkill /f /im TEX3D_SIM.EXE
```

TEX3D\_SIM.EXE 의 프로세스를 모두 강제 종료합니다. 가끔 WebTex3D가 동시에 여러개 서비스 되고 있는 서버가 있는데, TEX3D\_SIM 프로세스가 여러개 켜지게 되다보니 모두 강제 종료하게끔 하였습니다.

```
"C:\Program Files (x86)\Apache Software Foundation\Apache2.2\bin\httpd" -k stop
"C:\Program Files (x86)\Apache Software Foundation\Apache2.2\bin\httpd" -k start
```

그 후 Apache Tomcat에 종료 및 시작을 명령합니다. Apache Restart도 고려하였으나, stop and start가 가장 안정적으로 작동하였습니다.

```
curl -k http://localhost:8002/runtex3dsim
```

마지막으로 curl을 이용해 TEX3D\_SIM.EXE 를 강제로 실행시키도록 합니다. 해당 url은 접속시 다음과 같은 행동을 합니다.

**evegarden/simulation/views.py**

```
def runTex3dSim(request):
    try:
        cmd = 'h:' + settings.MEDIA_ROOT + 'data/webtex3d/Tex3D_Sim.exe'
        args = shlex.split(cmd)
        proc = subprocess.Popen(args)
        variables = {'success': True}
    except:
        cmd = settings.MEDIA_ROOT + 'data/webtex3d/Tex3D_Sim.exe'
        args = shlex.split(cmd)
        proc = subprocess.Popen(args)
        variables = {'success': False}

    return HttpResponse(json.dumps(variables), mimetype='application/json')
```

Simulation 자체를 진행하지는 않지만 TEX3D\_SIM.EXE 가 있는 경로를 찾아 Apache의 SubProcess (프로세스에서 프로세스를 호출, 자식처럼 물리게 됨)로서 실행을 합니다.



## 문제점

### 1. jQuery Local화에 따른 오동작

기존에는 jQuery 1.7 이하의 버전을 사용하였는데, Local화를 하면서 가장 최신 버전인 1.11버전 이상을 사용하면서 문제가 발생하였습니다. 1.7과 최신 버전은 특정 문법에 있어서 다소 차이가 있습니다.

가장 대표적인 것으로 attr과 prop, checked 문법 등입니다. 자세한 부분은 구글링을 통하여 검색하세요. 오동작을 픽스하기 위해 2016년 3월 10일자 Commit `c2c3dff` 를 진행하였습니다.

### 2. WebTex3D 모듈의 특성상 Stand Alone System 불가능

애초에 WebTex3D는 다음과 같은 방식으로 인증을 진행합니다.

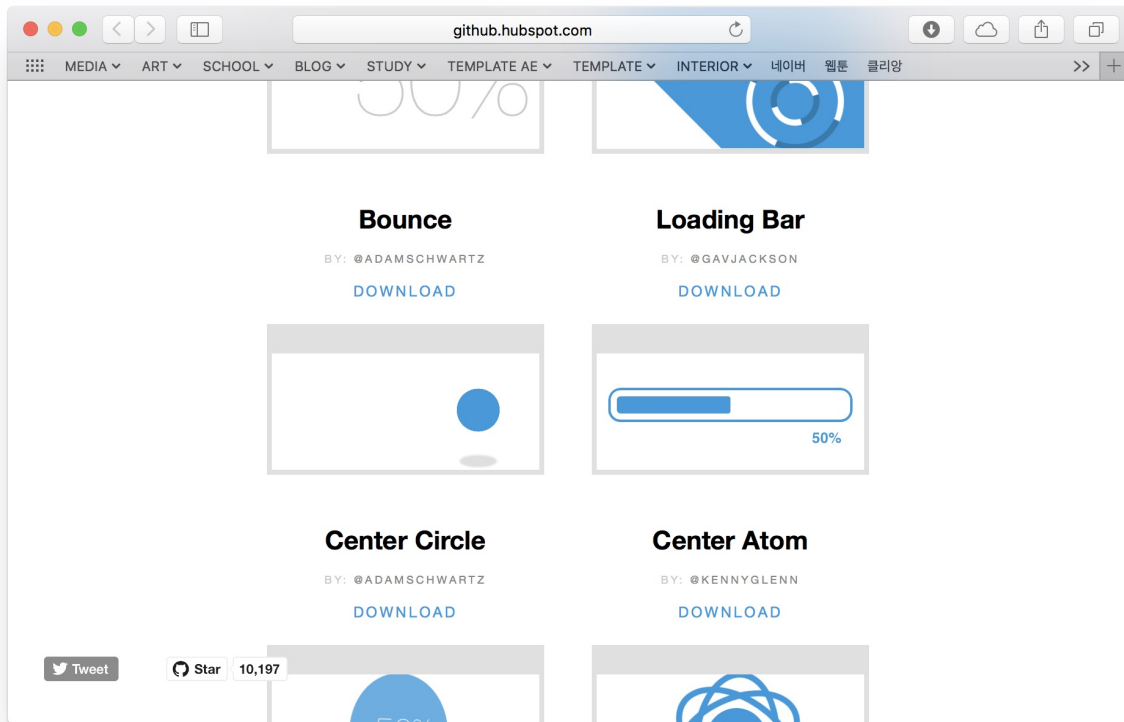
- i. 최초 인증시 웹인증을 통하여 Online상태로 인증
- ii. 이후 모듈 재실행시 Online상태라는 가정 하에, 서버와의 Status 체크 후 추가 인증 없이 바로 실행.
- iii. Online상태가 아닌 경우 재실행시, 서버와 체크 불가로 오류.
- iv. 모듈 실행 중, 주기적인 서버와의 인증 체크 진행

때문에 Offline상태로 테스트를 진행하게되면, 초기 인증을 Online으로 무사히 받았더라도 주기적인 서버와의 인증 체크 과정에서 모듈이 풀리게 되며, 정상적인 시뮬레이션이 불가능해짐에 따라 많은 Error가 났었던 것입니다. 이를 위해 BAT 파일을 만들어 배포했지만, 원론적으로는 해결할 수 없습니다.

Module을 무인증 파일로 제작할 수는 있지만, 불법 복제가 가능함에 따라 이 부분을 포기하게 되었고, WebTex3D는 Evegarden에 비정상적으로 설치될 수 밖에 없었습니다. **절대 WebTex3D는 Module이 교체되지 않으면 Stand Alone 하게 사용할 수 없음**을 유의해 주세요.

# AJAX Loading bar

[Pace.js](#) 를 사용하여 사용자의 인터넷 송수신 현황을 그래픽으로 나타냅니다.



## 사용된 Open Source, 기술 URL

- [Pace.js](#)

## 사용 예시

간편하게 원하는 디자인을 프로젝트 홈페이지에서 골라 Customized된 CSS와 JS를 우리의 프로젝트에 설치해 주면 됩니다. [관련문서](#)를 참조해 주세요.

기본 값으로 접속할 때에 노출되지만, 우리의 경우 클릭에 따라 data의 Ajax통신이 일어날때도 보여져야 하므로 다음과 같이 설정합니다.

```
$(document).ajaxStart(function() {
    Pace.restart();
});
```



# Export to PDF

화면에 보여지는 내용을 PDF로 렌더링하기 위해 다음과 같이 진행하였습니다.



## 사용된 Open Source, 기술 URL

- [pdfMAKE](#)

## 사용 예시

수시로 프론트쪽에서 변경되는 즐겨찾기 아이템을 서버에서 렌더링을 할 수도 없는 노릇이고, 프론트쪽에서 PDF로 출력하는 부분은 고도화된 플러그인이 존재하지 않았습니다.

따라서 [pdfMAKE](#)라는 조은 오픈 소스를 이용하여 작업을 진행합니다. 이를 통해 HTML의 DOM을 PDF로 출력할 수 있습니다. 다만 다음과 같은 애로사항이 존재합니다.

1. CJK, 즉 아시아권 언어가 지원되지 않는다.
2. CSS 스타일링이 적용되지 않는다.
3. 이미지의 경우 파일이 아니라 **Base64**형식으로 전달해야 한다.
4. 1회의 PDF출력이 일어난 후에, 브라우저의 메모리 관리 등을 이유로 다시 PDF 출력을 진행하면 정상적으로 동작하지 않는다.
5. pdfmake가 지원하는 **json**형식으로 출력할 데이터를 넘겨줘야 한다.

1번의 경우 Custom Font를 변환하여 탑재하면 한글 및 일본어가 정상적으로 동작한다고 파악하였습니다. 그래서 다음과 같은 방법으로 폰트를 **vfs형태**(자바스크립트의 일종)로 바꾸어 탑재하였습니다.

사용된 폰트는 **spoqa hans**이며, 모든 사용에 대하여 저작권 **Free**입니다. 하지만, 폰트 파일인 **vfs**인 경우 4MB를 넘는 고용량 파일이기 때문에 **viewer**를 로딩할때 같이 부르기에는 부담이 컸습니다. (초기 로딩이 느려지기 때문) 따라서 다음과 같이 PDF 인쇄 호출이 일어날 때만 불러와 지도록 제작했습니다.

### fh.viewer.js

```
$('#btnPDF').click(function() {
  if (checkIsDeleted) {
    alert('PDF製作前に、ショッピングカート画面を入れ直してください。');
    $('#modalWishlist').modal('hide');
    location.reload();
    return false;
  }
  $.getScript("/media/js/vfs_fonts.js", function() {
    ...
  })
});
```

`.click()` 이벤트가 일어났을때 `$getScript` 를 통해 **vfs**파일의 로딩이 일어나며, 로딩이 모두 완료되었을때야 PDF출력 기능을 실행합니다.

`docDefinition`은 `makePrintData`의 데이터와, 스타일링을 보유하고 있습니다.

```

$('#btnPDF').click(function() {
  if (checkIsDeleted) {
    alert('PDF製作前に、ショッピングカート画面を入れ直してください。');
    $('#modalWishlist').modal('hide');
    location.reload();
    return false;
  }
  $.getScript("/media/js/vfs_fonts.js", function() {
    var printData = makePrintData();
    var totalCost = '合計金額 : ¥ ' + $('#currentCost').text() + ' ';
    var docDefinition = {
      content: [{
        text: 'お気に入り',
        style: 'header',
        alignment: 'center'
      },
      printData
    ],
    styles: {
      header: {
        fontSize: 13,
        bold: true,
        margin: [0, 5, 0, 5]
      },
      ...
    }
  },
  defaultStyle: {
    columnGap: 20,
    font: 'myFont'
  },
  footer: [{
    text: 'Sunregent',
    margin: [30, 0, 0, 0]
  }, {
    text: totalCost,
    alignment: 'right',
    margin: [0, 0, 20, 100]
  }]
}
console.log(docDefinition);
pdfMake.createPdf(docDefinition).download('sample.pdf');
});

```

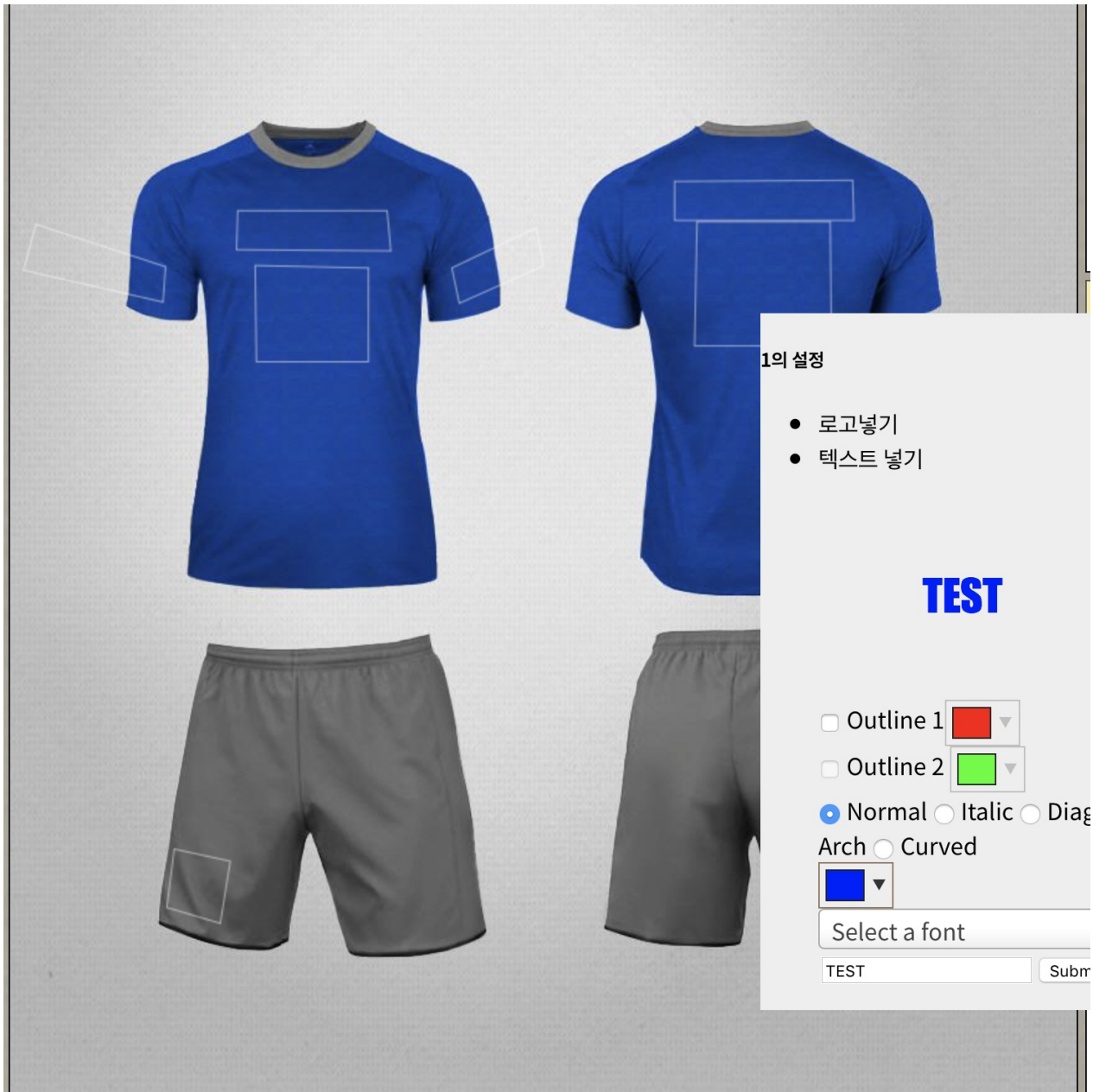
PDF 출력기능이 시작되면, `makePrintData`를 통해 출력할 아이템 데이터를 가지고 옵니다. 이 도중에 `Image`의 `Path`를 읽게 되면, 이를 `getBase64FromImageUrl` 를 통해 `Base64 (string)`형태로 저장하여 `Data`에 포함시킵니다. 이 과정이 끝나게 되면

`pdfMake.createPdf(docDefinition).download` 를 통해 인쇄가 진행됩니다.

```
function getBase64FromImageUrl(url) {  
    var img = new Image();  
    img.setAttribute('crossOrigin', 'anonymous');  
    img.onload = function() {  
        var canvas = document.createElement("canvas");  
        canvas.width = this.width;  
        canvas.height = this.height;  
  
        var ctx = canvas.getContext("2d");  
        ctx.drawImage(this, 0, 0);  
  
        var dataURL = canvas.toDataURL("image/png");  
        // alert(dataURL.replace(/^data:image\/(png|jpg);base64/, ""));  
        return dataURL;  
    };  
  
    img.src = url;  
}
```

스타일링은 HTML의 CSS로는 불가능하기 때문에,(pdfMake는 CSS2 이상 수준의 스타일링을 받아들이지 못합니다.) 대신에, pdfMake의 [document](#)를 보면서 [playground](#)에 따라 직접 스타일링 했습니다.

## Render Grid via Image Map



## 사용된 Open Source, 기술 URL

- [jQuery Maphilight](#)

## 사용 예시

위에 소개된 Maphilight는 이미지 위에 Image Map을 뿌려주는 역할을 합니다.



유니폼 등의 프로젝트에서, 특정 구역을 잡기 위해서는 .txd파일의 grid 구역을 체크하여 뿌려주고 이를 클릭하여 액션을 취할 수 있도록 하는 것이 사용 방법에 편의를 도울 수 있다고 판단했기 때문에 Image Map을 사용하였습니다.

다음 코드는 Grid를 Image Map으로 나타내는 부분을 보여줍니다.

### fh.viewer.sticker.js

```
wvs.setProductGrids = function(xml) {
    var dataGrid = [];
    var canvas = '<map name="gridForSticker">';
    var ratiox = 0.86;
    var ratioy = 0.84
    $.ajax({
        type: "GET",
        url: xml,
        dataType: "xml",
        async: false,
        success: function(response) {
```

grid 데이터를 저장하는 data, 최종결과물로 html에 삽입될 canvas, 그리고 원본 이미지보다 작게 표현되는 WebTex3D base를 위해 이 비율을 가로, 세로로 맞춰주는 ratio로 구성되어 있습니다.

데이터 수신에 성공하면 다음 코드가 실행됩니다.

```
var a = xmlToJson(response);
var maxLayer = 0;

for (var i = 0; i < a.canvas.groups.group.length; i++) {
    if (maxLayer < a.canvas.groups.group[i]['@attributes'].layer) {
        maxLayer = a.canvas.groups.group[i]['@attributes'].layer;
    }
}

for (var i = 0; i < a.canvas.groups.group.length; i++) {
    if (a.canvas.groups.group[i]['@attributes'].layer == maxLayer) {
        dataGrid.push(a.canvas.groups.group[i].grids.grid);
    }
}
if (dataGrid.length == 0) {
    return false;
}
```

xml을 JSON형태로 바꾸어 개발의 편의를 돕도록 합니다.

기본적으로 maxLayer를 0으로 잡은 상태에서,

1. groups.group만큼 for문을 돌면서, 최상단 레이어 (로고, 텍스트 맵핑을 할 레이어)가 몇 번째 레이어인지 파악합니다.

2. groups.group만큼 for문을 돌면서, 최상단 레이어가 나타나면 해당 레이어의 그리드 정보들을 dataGrid에 집어넣습니다.
3. 만약 dataGrid에 0개의 grid정보가 입력되었다면 render Grid를 포기합니다.

```
for (var i = dataGrid[0]['@attributes'].index; i < dataGrid.length; i++) {
    var dataPoint = '',
        destPoint = ['p0', 'p1', 'p2', 'p3'],
        dataWidth = 0,
        dataHeight = 0;
```

- dataGrid의 첫 gridGroup부터 마지막까지 for문을 돌면서, 각 grid의 꼭지점을 체크하여 dataPoint에 넣습니다. - dataWidth와 dataHeight는 각 grid구역의 가로와 세로를 뜻합니다.
- destPoint는 각 grid에 있는 정보 p0, p1, p2, p3을 위해 미리 상용구로 지정해 둔 것입니다.

이제 분기점이 두가지로 결정됩니다. grid Group에 grid가 하나만 있을 때, 혹은 팔 부분(앞에서 왼쪽 팔에 로고를 넣으면 뒤에서 오른쪽 팔에도 Mapping되어야 함)처럼 grid Group에 grid가 2개 있을 때를 체크하여 줍니다.

1. gridGroup에 grid가 하나만 있을 때

```
if (!$.isArray(dataGrid[i])) {
    for (point in destPoint) {
        var pointOrder = destPoint[point];

        dataPoint += (dataGrid[i][pointOrder]['@attributes'].x) * ratiox + ', ';
        dataPoint += (dataGrid[i][pointOrder]['@attributes'].y) * ratioy;

        if (point < (destPoint.length - 1)) {
            dataPoint += ', ';
        }
    }
    dataWidth = Math.sqrt(Math.pow(dataGrid[i]['p0']['@attributes'].x - dataGrid[i]['p1']['@attributes'].x, 2) + Math.pow(dataGrid[i]['p0']['@attributes'].y - dataGrid[i]['p1']['@attributes'].y, 2));
    dataHeight = Math.sqrt(Math.pow(dataGrid[i]['p0']['@attributes'].x - dataGrid[i]['p3']['@attributes'].x, 2) + Math.pow(dataGrid[i]['p0']['@attributes'].y - dataGrid[i]['p3']['@attributes'].y, 2));
}
```

.isArray() 를 통해서 dataGrid[i] 가 배열인지 아닌지를 파악합니다. 그 후 각 point마다 x, y 좌표를 체크합니다. 마지막으로 가로와 세로를 p0-p1과 p0-p3를 기준으로 계산하여 저장합니다.

2. gridGroup에서 grid가 **2개 이상** 있을 때

```

        else {
            var firstGrid = dataGrid[i][0];
            for (point in destPoint) {
                var pointOrder = destPoint[point];
                dataPoint += (firstGrid[pointOrder]['@attributes'].x) * ratio
x + ', ';
                dataPoint += (firstGrid[pointOrder]['@attributes'].y) * ratio
y;
                if (point < (destPoint.length - 1)) {
                    dataPoint += ', ';
                }
            }
            dataWidth = Math.sqrt(Math.pow(dataGrid[i][0]['p0']['@attributes'].x - dataGrid[i][0]['p1']['@attributes'].x, 2) + Math.pow(dataGrid[i][0]['p0']['@attributes'].y - dataGrid[i][0]['p1']['@attributes'].y, 2));
            dataHeight = Math.sqrt(Math.pow(dataGrid[i][0]['p0']['@attributes'].x - dataGrid[i][0]['p3']['@attributes'].x, 2) + Math.pow(dataGrid[i][0]['p0']['@attributes'].y - dataGrid[i][0]['p3']['@attributes'].y, 2));
        }
    }

```

무조건 gridGroup의 첫 번째 grid를 기준으로 체크하게 됩니다. 그 외에는 1번 항목과 같은 방식으로 좌표를 체크합니다.

```

        temp = '<area shape="poly" href="javascript:void(0);" data-width=' + dataWidth + ' data-height=' + dataHeight + ' coords="' + dataPoint + '" alt="' + i + '">';
        canvas += temp;
    }
});
canvas += "</map>";

$('map').remove();
$(canvas).appendTo('body');

$('area').bind("click", function(ev) {
    wvs.clickGrid(ev, $(this).attr('alt'), $(this).data('width'), $(this).data('height'));
});
}

```

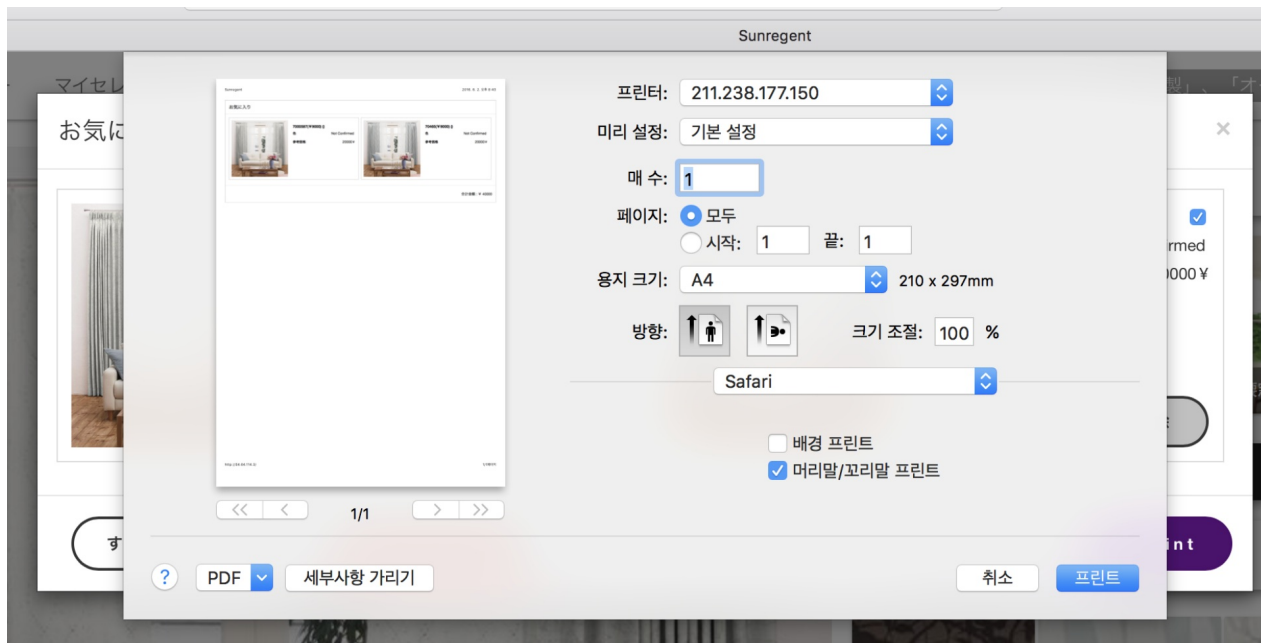
각 grid data들이 체크되면, temp변수를 통해 해당 grid를 area tag dom으로 표현합니다. 이 부분은 곧 image map 위에 grid를 나타내는 공간으로 보여질 것입니다. 마지막으로 계산이 완료된 canvas는 body에 삽입됩니다.

주의하세요. **area**는 **css**가 적용되지 않습니다. 대신에 plugin 내부에 설정할 수 있는 [option](#)들이 있습니다.

현재 설정된 option값은 다음과 같습니다.

```
$.fn.maphilight.defaults = {  
  fill: true,  
  fillColor: '000000',  
  fillOpacity: 0,  
  stroke: true,  
  strokeColor: 'ffffff',  
  strokeOpacity: 0.5,  
  strokeWidth: 1,  
  fade: true,  
  alwaysOn: true,  
  neverOn: false,  
  groupBy: false,  
  wrapClass: true,  
  shadow: false,  
  shadowX: 0,  
  shadowY: 0,  
  shadowRadius: 6,  
  shadowColor: '000000',  
  shadowOpacity: 0.8,  
  shadowPosition: 'outside',  
  shadowFrom: false  
}
```

# Print



즐거찾기에 등록된 아이템을 인쇄물로 출력할 수 있습니다.

## 사용된 Open Source, 기술 URL

CSS Print

## 사용 예시

출력을 이상적으로 하기 위해서는 전체 페이지가 아니라, 즐겨찾기의 아이템 목록, 전체 가격만 출력이 되어야 합니다. 이를 위해서 다음과 같이 JS와 CSS에서 @print 라는 미디어 쿼리를 이용합니다.

fh.viewer.js에서 프린트할 섹션을 만듭니다.

**fh.viewer.js**

```
document.getElementById("btnPrint").onclick = function() {
    printElement(document.getElementById("formCart"));
    window.print();
};

function printElement(elem, append, delimiter) {
    var domClone = elem.cloneNode(true);

    var $printSection = document.getElementById("printSection");

    if (!$printSection) {
        var $printSection = document.createElement("div");
        $printSection.id = "printSection";
        document.body.appendChild($printSection);
    }

    if (append !== true) {
        $printSection.innerHTML = "";
    } else if (append === true) {
        if (typeof(delimiter) === "string") {
            $printSection.innerHTML += delimiter;
        } else if (typeof(delimiter) === "object") {
            $printSection.appendChild(delimiter);
        }
    }

    $printSection.appendChild(domClone);
}
```

theme.css에서 media query 인 screen , page , print 를 이용하여 인쇄될 영역을 아름답게 꾸며줍니다.

### theme.css

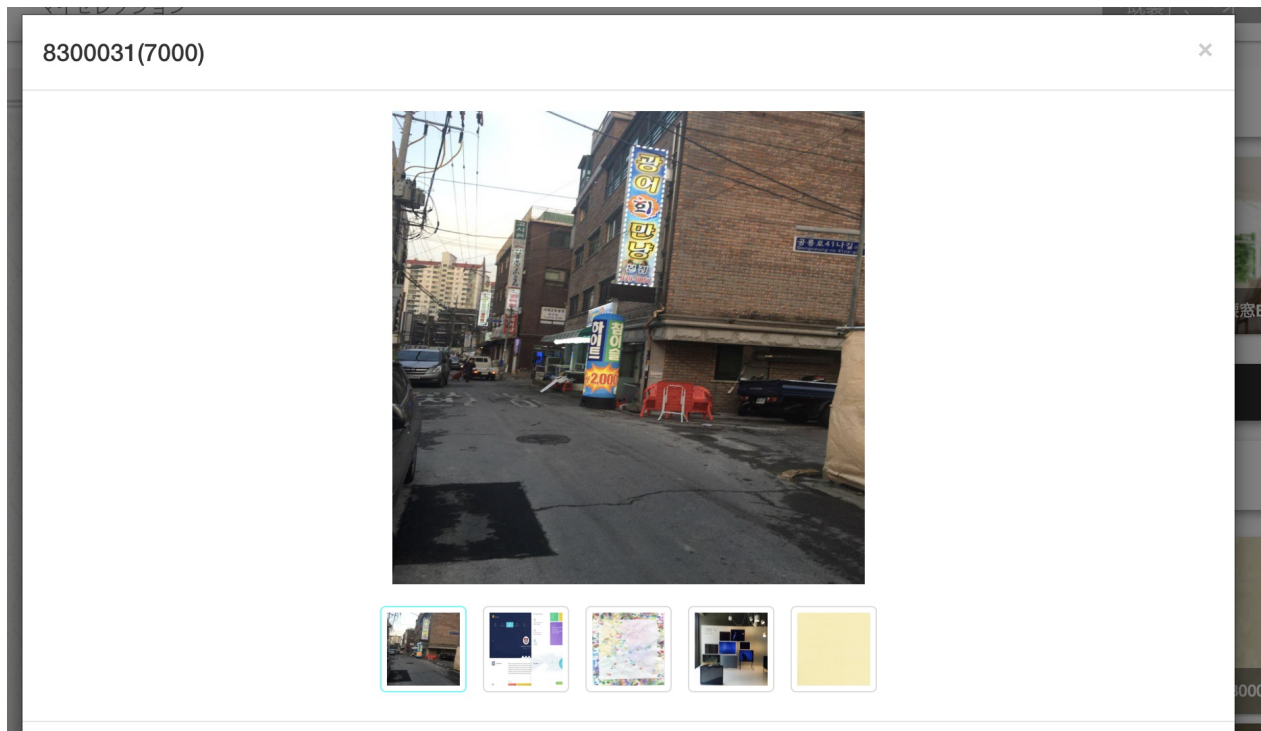
```
@media screen {  
    #printSection {  
        display: none;  
    }  
}  
  
@page {  
    size: A4;  
    margin: 1.2cm 0.8cm;  
}  
  
@media print {  
    body * {  
        visibility: hidden;  
    }  
    .container-webtex3d {  
        display: none;  
    }  
    html,  
    body {  
        width: 1100px !important;  
        height: 29.7cm;  
    }  
    ....  
}
```

screen, 즉 일반 PC에서는 `#printSection` 을 보이지 않게 합니다.

인쇄될 영역과 사이즈는 `page` 로 지정합니다. 기본값인 A4로 진행하게 되었네요. 또한 인쇄될 당시에만 적용되는 CSS인 `@media print` 를 이용해서, 인쇄 당시의 스타일을 잡아줍니다.

다른 부분들은 안보이게 하고, `#printSection` 만 잘 보일 수 있도록, 레이아웃이 무너지지 않도록 스타일을 잡습니다.

# Image Gallery



Fabric Information의 이미지 갤러리 기능입니다. 기존에 있었던 갤러리 기능을 제거하고, 최대 8장까지 표현되는 갤러리를 제작하였습니다.

## 사용된 Open Source, 기술 URL

### 사용 예시

fh.viewer.js의 `wv.showMaterial` 에서 다음과 같이 Fabric에 속한 이미지들을 xml 파싱으로 불러옵니다.

#### fh.viewer.js

```
$('detailimage', FabricInfoXML).each(function() {
  var detailImg = $(this);
  var src = $('image_url', detailImg).text();
  detailImage.push(src);
});
```

또한 다음과 같이 DOM을 만들어 빈 이미지들을 넣어줍니다. `.click()` 이벤트도 만들어줍니다.



```
if (detailImage.length == 0) {
    $('.img-detail-large').attr('src', $('image_url', onerepeatImg).text());
} else {
    $('.img-detail-large').attr('src', detailImage[0]);
    $('.container-detail-image').html('');

    for (var i = 0; i < detailImage.length; i++) {
        $('.container-detail-image').append("<img class='img-thumbnail' id='detailImage" + i + "' src='" + detailImage[i] + "' />");
    }

    $('.img-thumbnail').click(function() {
        $('.img-detail-large').attr('src', $(this).attr('src'));
        $('.active', modal).removeClass('active');
        $(this).addClass('active');
    })
    $('#detailImage0').addClass('active');
}
```

# Not Using Hover on Mobile Device

## 사용된 Open Source, 기술 URL

- [How to deal with :hover on touch screen devices](#)

## 사용 예시

Hover기능은 PC에서는 재미있는 기능이지만, Mobile에서 CSS의 `:hover` 에 속하는 요소들의 hover이벤트 발생을 막아야 합니다. 터치를 두번해야 click기능이 되기 때문입니다.

CSS의 요소 중 `:hover` 가 있는 요소는 모바일 디바이스 사용자에게 오동작을 보여지게 됩니다.

따라서, 우선 js로 접속 브라우저가 모바일 기기로부터 온 것인지 판단하여, flag인 `.no-touch` 를 추가합니다.

### viewer.html

```
if ("ontouchstart" in document.documentElement) {} else {
    $('body').addClass("no-touch");
}
if (/Android|webOS|iPhone|iPad|iPod|BlackBerry|IEMobile|Opera Mini/i.test(navigator.userAgent)) {
    $('.hide-mobile').hide();
    $('body').removeClass("no-touch");
} else {
    $('body').addClass("no-touch");
}
```

그 후, css에서는 `.no-touch` 가 입혀진 element에 hover의 스타일링을 입힙니다.

### theme.css

```
.navbar-nav a {  
    transition: all 0.2s ease-in-out;  
}  
  
.no-touch .navbar-nav a:hover {  
    background-color: #fff !important;  
    box-shadow: inset 8px 0px 0px #76328c;  
    z-index: 10;  
    font-weight: bold;  
}
```

# Save Simulation Data on CSV

My Favorite에 담긴 item들에 대한 정보를 CSV, txt, image 파일 등 으로 저장하여 서버 내에 위치하는 방법에 대하여 서술합니다.

## 사용된 Open Source, 기술 URL

- [Python CSV](#)

## 사용 예시

이 기능 또한 당연히도 js (Front-end)부분과 py (Back-end)부분으로 나누어 개발이 진행 됩니다. js에서는 간단하게, 우리의 Product data를 json형식으로 넘겨주면 됩니다.

### fh.viewer.js

```
$('.item-list').each(function () {  
    var currentList = $(this).attr('id');  
    if ($('#input:checkbox[name="product"]', this).is(":checked")) {  
        jsonData.push(JSON.stringify($('#'+currentList+' :input').serializeObject()));  
    }  
});
```

`.item-list` 아이템 내부의 체크박스가 활성화 되어있을 때, 이 항목에 대한 data를 Json 형식으로 임의의 변수 `jsonData` 에 배열 형식으로 저장합니다.

```
$.ajax({
  url: '/createcart',
  type: 'POST',
  cache: false,
  data: {
    productList: jsonData
  },
  dataType: 'json',
  async: false,
  success: function (response) {
    console.log(response);
    if (response['success']) {
      alert('発注に成功しました。');
    } else {
      ...
    }
  },
  ...
});
```

그 후 이 data를 간단하게 createcart라는, 곧 보여드릴 코드의 부분으로 보내줍니다.

### evegarden/cart/views.py

```
def createCart(request):
    try:
        if not request.POST['productList[]']:
            raise
        productList = request.POST.getlist('productList[]')
    except Exception as e:
        variables = {'success': False, 'reason': 'RequestError: ' + str(e)}
        return HttpResponse(json.dumps(variables), mimetype='application/json')
```

createCart 에서는 request 된 내용을 체크합니다. 이 때, productList는 배열로 오기 때문에 request.POST.getlist('productList[]') 형식으로 읽어 들입니다.

```

if not os.path.exists("/cart"):
    try:
        os.makedirs("/cart");
    except Exception as e:
        variables = {'success': False, 'reason': 'MakeDirError-cart: ' + str(e)}
        return HttpResponse(json.dumps(variables), mimetype='application/json')

if not os.path.exists("/hacchudata"):
    try:
        os.makedirs("/hacchudata");
    except Exception as e:
        variables = {'success': False, 'reason': 'MakeDirError-hacchudata: ' + str
(e)}

        return HttpResponse(json.dumps(variables), mimetype='application/json')

```

서버 로컬 내부의 폴더 여부에 대해 체크합니다. 있으면 스킵하고, 없으면 폴더를 만드는 형식으로 진행합니다.

```

cart = Cart()
cart.save()
oid = str(cart.id)
time = cart.time + timedelta(hours=9)

```

cart의 모델에 따라 cart를 Database에 저장합니다. time에 timedelta를 붙인 이유는 우리 python server가 현재 지역(한국)에 맞는 시간대로 적용하지 않았기 때문입니다. 이를 해결하기 위해 timedelta(hours=9) 로 보정해 줍니다.

```

if not os.path.exists("/hacchudata/"+time.strftime('%Y%m%d')+ "-" +oid):
    try:
        os.makedirs("/hacchudata/"+time.strftime('%Y%m%d')+ "-" +oid);
    except Exception as e:
        variables = {'success': False, 'reason': 'MakeDirError-datefolder: ' + str
(e)}

        return HttpResponse(json.dumps(variables), mimetype='application/json')

```

똑같은 방법으로 현재 날짜로 된 폴더가 있는지 체크하고, 없다면 만들어 줍니다.

```

#request = [{'product': 'product01:01-01', 'fabric': ['fab-01', 'fabr-01', 'fabri-01'],
'size': 's', 'quantity': '5', 'price': '5000'}, {'product': 'product02:01-01', 'fabric': ['f-01', 'fa-01', 'fab3-01'], 'size': 's', 'quantity': '5', 'price': '5000'}, {'product': 'product03:01-01', 'fabric': ['f-01', 'fab-01', 'fabr-01'], 'size': 's', 'quantity': '5', 'price': '5000'},]
#date = datetime.datetime.now().strftime('%Y-%m-%d - %H%M%S')

```

request에 전달되어야 하는 정보의 convention 규칙은 위와 같습니다. (주석으로 해당 정보의 양식에 대해 작성해 둔 것입니다. 저 양식을 따르지 않으면 오류가 납니다.)

```

returnTxt = createTxt(oid, time, productList)
if not returnTxt['success']:
    return HttpResponse(json.dumps(returnTxt), mimetype='application/json')

returnCsv = createCsv(oid, time.strftime('%Y%m%d'), productList)
if not returnCsv['success']:
    return HttpResponse(json.dumps(returnCsv), mimetype='application/json')

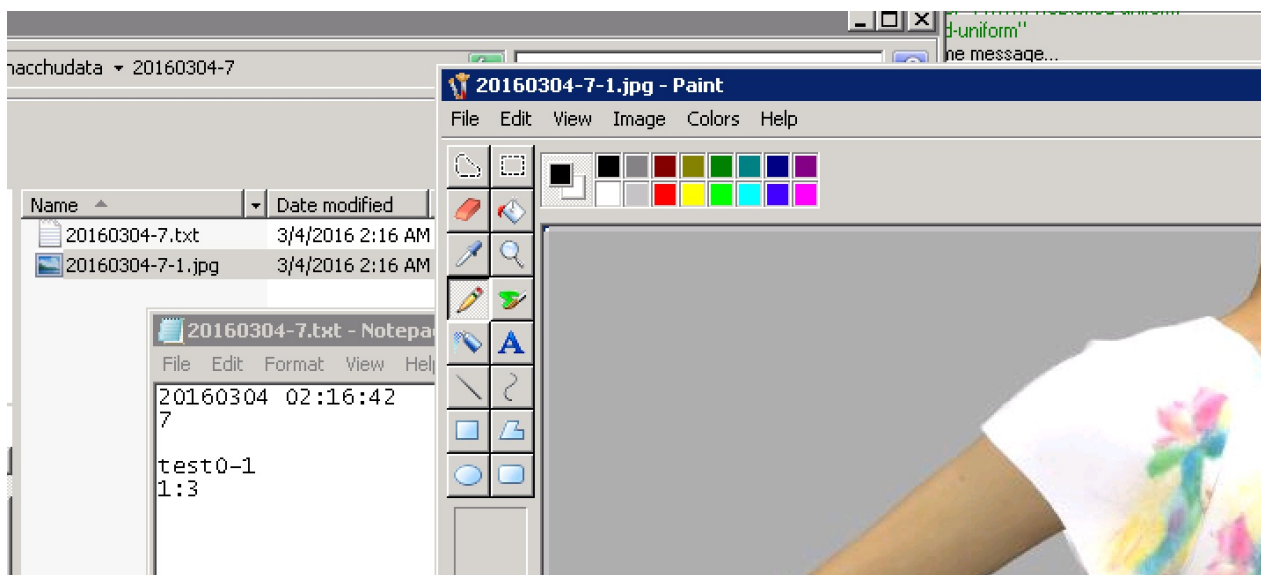
returnJpg = createJpg(oid, time.strftime('%Y%m%d'), productList)
if not returnJpg['success']:
    return HttpResponse(json.dumps(returnJpg), mimetype='application/json')

variables = {'success': True}
return HttpResponse(json.dumps(variables), mimetype='application/json')

```

이제 각각 text, csv, jpg에 대하여 data를 전달하고, 저장할 수 있도록 함수를 불러옵니다.

## createTxt



```

def createTxt(oid, time, productList):
    try:
        date = time.strftime('%Y%m%d')
        file = open("/hacchudata/"+date+"-"+oid+"/"+date+"-"+oid+".txt", "w")
    except Exception as e:
        variables = {'success': False, 'reason': 'MakeTxtError: ' + str(e)}
        return variables

    try:
        file.write(time.strftime('%Y%m%d %H:%M:%S')+"\n")
        file.write(oid+"\n\n")
        for product in productList:
            item = ast.literal_eval(product)
            file.write(item['product']+"\n")
            if 'fabric' in item :
                if type(item['fabric']) is list:
                    for i in range(len(item['fabric'])):
                        file.write(str(i+1)+":"+str(item['fabric'][i])+"\n")
                else:
                    file.write(str(1)+":"+str(item['fabric'])+"\n")
            else :
                file.write("(Default product, no fabric)\n")
            file.write("\n")

    except Exception as e:
        file.close()
        variables = {'success': False, 'reason': 'WriteTxtError: ' + str(e)}
        return variables

    file.close()
    variables = {'success': True}
    return variables

```

productList내부의 Product별로 서술이 들어갑니다. 각각의 product에 대해 서술하면서, fabric이 단일 fabric일지, 0개일지, 여러개의 fabric일 지에 따라 text를 다르게 작성합니다.

file open을 하고, write를 했던 것이 끝나면 close를 통해 file의 IO stream을 닫아줍니다.

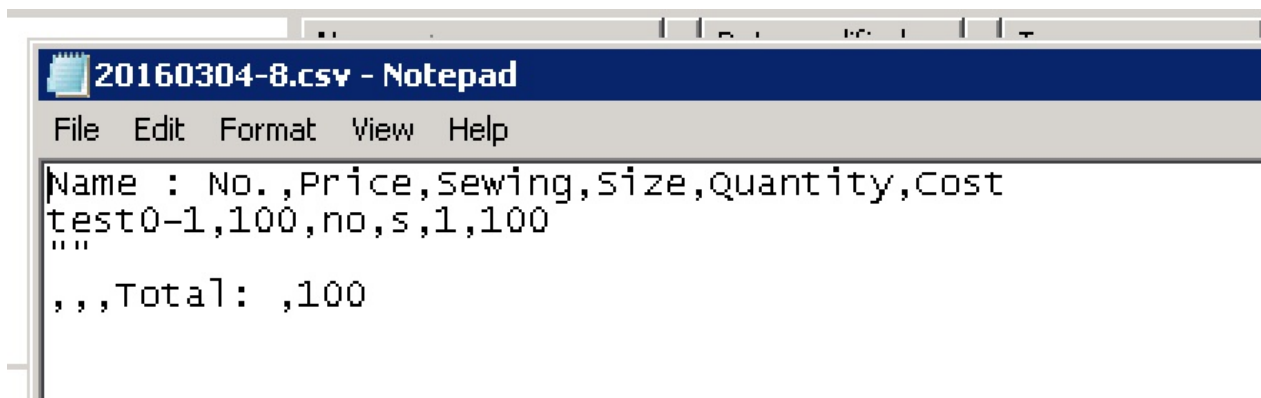
## createJpg



```
def createJpg(oid, time, productList):
    count = 0
    try:
        for product in productList:
            count += 1
            dst = "/hacchudata/"+time+"-"+oid+"/"+time+"-"+oid+"-"+str(count)+".jpg"
            item = ast.literal_eval(product)
            shutil.copy(settings.MEDIA_ROOT+item['url'][7:], dst)
    except Exception as e:
        variables = {'success': False, 'reason': 'MakeJpgError: ' + str(e)}
        return variables
    variables = {'success': True}
    return variables
```

JPG를 저장하는 부분입니다. 각 product별 시뮬레이션 이미지를 url로 체크하는데, 절대 경로가 아닌 상대 경로를 이용하기 위해 [7:] 형식으로 주소를 잘라 복사를 진행합니다.

## createCsv



```

def createCsv(oid, time, productList):
    try:
        f = open("/cart/"+time+"-"+oid+".csv", 'wb')
        file = csv.writer(f)
    except Exception as e:
        variables = {'success': False, 'reason': 'MakeCsvError: ' + str(e)}
        return variables

    sum = 0
    try:
        file.writerow(['Name : No.', 'Price', 'Sewing', 'Size', 'Quantity', 'Cost'])
        for product in productList:
            item = ast.literal_eval(product)
            if item['sewing'] == "yes" :
                price = item['price2']
            else :
                price = item['price']
            if int(price) == "" :
                price = 0
            file.writerow([item['product'], price, item['sewing'], item['size'], item['quantity'], str(int(item['quantity']) * int(price)) ])
            sum += int(item['quantity']) * int(price)
        file.writerow([''])
        file.writerow(['', '', '', 'Total: ', str(sum)])

    except Exception as e:
        f.close()
        variables = {'success': False, 'reason': 'WriteCsvError: ' + str(e)}
        return variables

    f.close()
    variables = {'success': True}
    return variables

```

createTxt와 같은 방식으로 data를 작성합니다. 하지만 CSV는 문서를 작성할 때 줄바꿈 `writerow` 과 컬럼구분을 위한 콤마 , 로 구성이 됩니다. 이를 꼭 **확인**하십시오. 특히 가장 상단에 서의 , 갯수는, 다음 줄에서도 동일해야 합니다.

# Search (KEYUCA)

## 사용된 Open Source, 기술 URL

- [select2.js](#)

## 사용 예시

### DB Table Scheme

본 기능을 구현하기 위해서 추가된 Database Table들입니다. 자세한 내용은 보고서 작성 기준 curtain-keyuca 제품을 확인할 수 있는 AWS을 원격 접속을 통해 pgAdmin에서 확인할 수 있습니다. 모든 Table은 webTex3D 데이터베이스에 포함되어 있습니다. 또한 전체의 데이터베이스를 기술하는 것이 아닌, 추가된 점만 기술하였습니다.

표의 각 row는 Field명 – Type 및 속성 – 간단한 설명으로 구성됩니다.

속성에 사용되는 단어 중 NN은 NOT NULL, PKEY는 PRIMARY KEY, FKEY는 FOREIGN KEY, UKEY는 UNIQUE KEY를 의미합니다. 또한 Type중 BOOL는 실제로는 BOOLEAN입니다.

**fabric\_searchcolor** : fabric\_information에 종속되어 fabric의 searchColor를 나타냅니다.

Field Name	Type 및 속성	Content
ID	INT/NN/PKEY	PRIMARY KEY. AUTO INCREMENT입니다.
fabricinfo_id	INT/NN/FKEY	fabric_fabricinformation-ID에 대한 외부 키입니다.
option 0~8	BOOL/NN	각 색상 옵션에 따라 해당 여부가 기록됩니다.

**fabric\_searchimage** : fabric\_information에 종속되어 fabric의 searchImage를 나타냅니다.

Field Name	Type 및 속성	Content
ID	INT/NN/PKEY	PRIMARY KEY. AUTO INCREMENT입니다.
fabricinfo_id	INT/NN/FKEY	fabric_fabricinformation-ID에 대한 외부 키입니다.
option 0~8	BOOL/NN	각 이미지 옵션에 따라 해당 여부가 기록됩니다.

**fabric\_searchfunction** : fabric\_information에 종속되어 fabric의 searchFunction를 나타냅니다.

Field Name	Type 및 속성	Content
ID	INT/NN/PKEY	PRIMARY KEY. AUTO INCREMENT입니다.
fabricinfo_id	INT/NN/FKEY	fabric_fabricinformation-ID에 대한 외부 키입니다.
option 0~8	BOOL/NN	각 기능 옵션에 따라 해당 여부가 기록됩니다.

이에 맞춰 `mysite/fabric/models.py` 가 수정되었으며, 기본적인 사안 외에는 수정된 점이 없어 따로 기술하지 않습니다. 다만 `views.py`와 기타 파일이 변경된 점은 다음 항목인 관련 문서 설명에서 기술하였습니다.

## 관련 문서 설명

파란색은 자바스크립트 파일의 함수, 초록색은 Python 파일의 함수로 구분되었습니다.

1. viewer (사용자 측면) 자바스크립트 파일은 `media/js/fh.viewer.js` 입니다. Python View 파일은 `mysite/fabric/views.py` 입니다.

- **ww.parseSearchFabrics (\$datastring)**

- 해당 함수는 `ww.parseFabrics` 와 뼈대는 같으나 검색 결과에 맞게 수정되었습니다. 검색버튼이 클릭되어 `#searchPanel` Form 에서 기록된 input의 Value값을 serialize 하였을 때, serialize된 값을 파라미터로 받으며 호출되는 함수입니다.
- ajax에서 data를 기존 코드처럼 보내지 않고 간단하게 `data: datastring` 하는 것으로 마칠 수 있습니다. 데이터는 `views.py` 의 `getSearchList` 로 전달되며 요청 성공시 `parseFabrics` 함수처럼 Fabric을 화면에 뿌려주고 각각의 `fabricItem`의 이벤트를 만듭니다.
- 또한 브라우저에서 Fabric 선택 시 Drag를 통해 인식했던 것과 달리 Click으로 이벤트를 바꾸면서 생긴 문제들에 대한 처리도 담고 있습니다.

터치 디바이스에서 Hover > Click 순으로 이벤트가 넘어가기 때문에 항상 ViewerBtn을 띄우게 하고, Fabric이 클릭되었을 때 Event 대상이 ViewerBtn 일 경우 `runSimulation` 실행이 아닌 FabricInformation창이 뜨도록 하였습니다.

- 마지막으로 선택된 Fabric Category가 특정 검색 항목을 사용하지 않을 경우 (fabricColor 등) 이를 disabled하여 등록되지 못하도록 하는 기능도 존재합니다.

- **Def getSearchList (request)**

- `ww.parseSearchFabrics` 에서 post된 data를 통해 해당되는 아이템을 검색 후 결과를 다시 전송하는 함수입니다. `request.POST["데이터의 항목명"]`을 통해 serialize된 데이터를 불러오며 복수개의 데이터가 올수 있는 항목은 dict를 이용하여 리스트화 합니다. 검색 옵션 중 기능이나 이미지, 컬러의 경우 복수 응답이 올 수 있으므로 `import operator`를 최상단에 선언하여 `operator` 기능을 쓸수 있게 한 후 `reduce`를

통해 리스트에 있는 항목과 DB에 있는 옵션들을 각각 비교합니다. 그 후 기능, 이미지, 컬러 순으로, 유효한 검색조건이 존재하면 이에 따라 fabriclist를 계속 검색해 나갑니다.

- 같은 카테고리 내의 검색 조건은 OR로, 다른 카테고리와의 검색조건은 AND로 묶어 사용자가 원하는 검색 결과가 나오도록 하였습니다. (예를 들어 검은색과 빨간색을 체크하면 검은색 항목, 빨간색 항목의 아이템 모두가 나오게 됩니다. 하지만 검은색과 차광1급 기능을 체크하면 두 조건 모두를 만족시키는 아이템이 나옵니다.)
- getList와 같이 user 등급 체크, item의 valid 체크 등은 동일하게 진행됩니다. 모든 작업 후 결과를 response하면서 끝냅니다. 이 때 전달되는 `xml/fabriclist.xml` 은 검색 옵션에 맞게 수정되어 있습니다.
- 추가된 *Post* 변수
  - i. `categoryId` = 원단인지, 레이스인지 구별하도록 함. (브라우저상에서는 보이지 않으나 `display:none`; 처리되어 있어서이고, `supercategory` 선택할 때마다 변경됩니다.)
  - ii. `price = fabric_price` 옵션 중 하나라도 체크되어 있으면 값을 리스트화 하여 가지고 옵니다. 그 외의 경우에는 False로 처리하여 불필요한 DB 색인 작업을 막습니다.
  - iii. `Color`, `image`, `function` = `price`와 내용이 상동합니다.
  - iv. `Number` : 상품명에 대한 데이터를 저장하고 비교합니다.

2. `manager` (관리자) 자바스크립트 파일은 `media/js/w3d.manager.fabric.js` 입니다. Python View 파일은 `'mysite/fabric/views.py'` 입니다.

`Select2.js`는 `manager` page의 fabric마다 selectbox의 option들을 태그 형식으로 등록할 수 있도록 하기 위해 사용되었습니다.

- **ww.initFabricInfo (\$datastring)**

- 처음, 혹은 Fabric의 FabricInformation을 가지고 올때마다 실행되는 함수입니다. 브라우저 상의 페이지 하단에 있는 input값들을 초기화 시키는 작업인데 select2를 사용한 input들을 초기화 하는 기능을 추가하였습니다.

- **ww.saveFabric (\$saveAs)**

- Fabric을 저장할 때 브라우저상에 체크 된 기능과 이미지, 컬러들을 받아들여 LaundryTags와 유사한 방법으로 `False = 0, True = 1` 로 저장하여 보냅니다.

- **ww.getFabricInfo (\$datastring)**

- Fabric의 FabricInformation을 불러들일 때, 해당 FabricInformation의 ID를 참조하는 Search 관련 테이블이 존재하지 않는다면 이를 참조하는 테이블을 만든 후 모든 option 값을 False로 초기화하는 기능을 갖습니다. 그 후 `Color`, `Image`, `Function`의 값을 불러들여 response 합니다.

- **Def createManagerFabric (request)**

- `ww.saveFabric` 을 통해 호출되면 변경된 정보를 DB에 저장하는 기능입니다.

strLaundryTags와 동일한 원리로 DB에 검색 조건 데이터를 저장합니다. 정보들은 1000111... 과 같이 이진수 번호로 들어오며 순서에 따라 `1 = True, 0 = False` 로 option에 저장합니다.

◦ 추가된 *Post* 변수

1. strFabricFunction = 체크된 fabric\_function 의 input들은 1로, 그 외의 경우에는 0으로 저장되어 도달합니다. FIFO로 하나씩 비교하여 해당 값을 option 컬럼에 저장합니다.
2. strFabricImage, strFabricColor = strFabricFunction와 내용이 상동합니다.

위에서 언급한 부분은 검색 기능의 주요한 부분입니다. 그 외에도 스타일시트와 HTML 마크업이 다소 수정되었으나 언급한 부분에 비해 소스 코드만 보아도 쉽게 이해할 수 있으리라 판단되어 따로 기술하지 않았습니다.

Manager의 Fabric 뷰어에서 셀렉트박스의 다중 선택을 GUI 상으로, 기술적으로도 쉽게 하기 위해 오픈 소스인 select2.js를 사용하였습니다.

직접 개발하지 않아도 충분한 퀄리티를 만들 수 있는 점이 오픈 소스의 장점이지만 플러그인을 너무 복잡하게 사용하거나 중복되는 기능들을 사용하거나, 혹은 [Javascript의 Prototype을 너무 남용하는 오픈소스 사용은 자제](#)해야 합니다. 대신 해당 오픈 소스의 라이선스 범위를 숙지하고 제공하는 기능이 충돌이 없는지 등을 좀더 확인하면 좋을 것입니다.

또한 Python 코드를 수정할 시 몇몇 코드 수정 프로그램 (Bracket 등) 에서 Tab키를 잘못된 값으로 받아들여 정상적인 코드 작성이 불가능했습니다. Python 파일을 수정할 경우 Pycharm과 같은 프로그램을 이용하여 코드 수정을 하시는 것을 권장합니다.

마지막으로, 프론트엔드 쪽에서 Form의 input 값을 보낼 때 일일이 input의 DOM ID나 name을 이용하여 호출하는 것을 기존 코드에서 보았습니다. 사실 이럴 필요 없이 **Form을 Serialize**하여 보내면 훨씬 간단하고 명료하게 코드 작성을 할 수 있습니다. Google에서 [Form Serialize Post Django](#) 로 검색하시면 자세한 정보를 확인 할 수 있습니다.

# Apache

Apache 2.2X버전을 기준으로 작성되었습니다. 2.4의 경우 `httpd-vhost.conf` 의 수정 방법이 달라집니다.

이 내용은 기존에 제작되었던 문서를 기준으로 작성되었습니다. 문서의 경로는 개발서버/server/current/Document/04. 프로젝트/33. WebTex3D/20150115\_Multiple django sites with Apache and mod\_wsgi.rtf 입니다.

반드시 수정 후 **Apache**를 재시작해주세요.

## 포트별로 다른 프로젝트 할당하기

httpd에서는 개방할 Port와 각 프로젝트의 `wsgi.py` 가 위치할 폴더를 지정해 주어야 합니다. 다음 코드를 참고하세요.

### Apache2/conf/httpd.conf

```
...

#Listen 12.34.56.78:80
Listen 80 // 80번 host 개방
Listen 8000 // 8000번 host 개방
Listen 8080 // 8080번 host 개방

...

//WSGI, 즉 wsgi.py가 위치할 경로를 각 프로젝트별로 기술합니다. 이때 구분자는 : 입니다.
WSGIScriptAlias /www/webtex3d-evegarden/evegarden:/www/webtex3d-fukushin/fukushin:/www/webtex3d-sunregent/sunregent:/www/webtex3d-uniform/mysite

...
```

`httpd-vhost.conf` 에서는 개방할 Port와 각 프로젝트에 대한 정보 및 권한 설정이 서술됩니다. 다음 코드를 참고하세요.

### Apache2/conf/extra/httpd-vhost.conf

```
...

NameVirtualHost *:80 // 80번 host 개방
NameVirtualHost *:8000 // 8000번 host 개방
NameVirtualHost *:8080 // 8080번 host 개방
```

```

...

<VirtualHost *:8000> // 8000번 host에 대한 Project 설정
    DocumentRoot "C:/www/webtex3d-evegarden/evegarden" //Document Root Path 설정
    ServerName 09wang.stylefor.us:8000 //의미가 없어요
    WSGIScriptAlias / "c:/www/webtex3d-evegarden/evegarden/wsgi.py" //프로젝트의 wsgi Pa
th
    <Directory "c:/www/webtex3d-evegarden/evegarden"> // 서버에서 작동할 파일들의 디렉토리 Pa
th 설정
        <Files wsgi.py>
            Order deny,allow // 권한설정, 절대 변경하지 마세요.
            Allow from all // 권한설정, 절대 변경하지 마세요.
        </Files>
    </Directory>
</VirtualHost>

<VirtualHost *:8080> // 8080번 host에 대한 Project 설정
    DocumentRoot "C:/www/webtex3d-uniform/mysite"
    ServerName 09wang.stylefor.us:8080
    WSGIScriptAlias / "c:/www/webtex3d-uniform/mysite/wsgi.py"
    <Directory "c:/www/webtex3d-uniform/mysite">
        <Files wsgi.py>
            Order deny,allow
            Allow from all
        </Files>
    </Directory>
</VirtualHost>

<VirtualHost *:80> // 80번 host에 대한 Project 설정
    DocumentRoot "C:/www/webtex3d-sunregent/sunregent"
    ServerName 09wang.stylefor.us:80
    WSGIScriptAlias / "c:/www/webtex3d-sunregent/sunregent/wsgi.py"
    <Directory "c:/www/webtex3d-sunregent/sunregent/">
        <Files wsgi.py>
            Order deny,allow
            Allow from all
        </Files>
    </Directory>
</VirtualHost>

<VirtualHost *:80> //기본값
    ServerAdmin webmaster@dummy-host2.example.com
    DocumentRoot "c:/Apache2/docs/dummy-host2.example.com"
    ServerName dummy-host2.example.com
    ErrorLog "logs/dummy-host2.example.com-error.log"
    CustomLog "logs/dummy-host2.example.com-access.log" common
</VirtualHost>

```

## 서브도메인(Sub-Domain)별로 다른 프로젝트 할당하 기



httpd에서는 각 프로젝트의 `wsgi.py` 가 위치할 폴더를 지정해 주어야 합니다. 서브도메인은 기본 포트 80번으로 접속되니까, 이처럼 포트별로 할당을 해줄 필요는 없습니다.

### Apache2/conf/httpd.conf

```
...

Listen 80 // 포트별로 프로젝트 할당하기와는 다르게 Port는 80번 (기본값)으로만 접속 허용할 것이므로 80
만 적어둡니다

...

//WSGI, 즉 wsgi.py가 위치할 경로를 각 프로젝트별로 기술합니다. 이때 구분자는 : 입니다.
WSGIPythonPath /var_w3d/www/mysite:/var_w3d_keyuca/keyuca:/var_shirt_youngwoo/youngwoo
:/var_w3d_map/map:/www/webtex3d-fabric/fabrics:/www/webtex3d-mobile/mobile

...
```

`httpd-vhost.conf` 에서는 각 프로젝트에 대한 `ServerName` (URL) 및 권한 설정이 서술됩니다.  
다음 코드를 참고하세요.

### Apache2/conf/extra/httpd-vhost.conf

```
...

NameVirtualHost *:80 // 포트별로 프로젝트 할당하기와는 다르게 Port는 80번 (기본값)으로만 접속 허용
할 것이므로 80만 적어둡니다

...

<VirtualHost 0.0.0.0:80> // 80번 포트만!
    ServerName www.stylefor.us // 여기서는 URL에 따라 접속할 프로젝트를 다르게 뿌려줄테니 Server
Name을 알맞게 작성하여야 합니다.
    WSGIScriptAlias / /var_w3d/www/mysite/wsgi.py
    <Directory /var_w3d/www/mysite>
        <Files wsgi.py>
            Order allow,deny
            Allow from all
        </Files>
    </Directory>
</VirtualHost>

<VirtualHost 0.0.0.0:80>
    ServerName keyuca.stylefor.us// Fabric에 대한 프로젝트이므로 위와 ServerName이 달라지겠죠?
    WSGIScriptAlias / /var_w3d_keyuca/keyuca/wsgi.py
    <Directory /var_w3d_keyuca/keyuca>
        <Files wsgi.py>
            Order allow,deny
            Allow from all
        </Files>
    </Directory>
</VirtualHost>

...
```

반드시, ServerName은 가비아의 네임플러스를 통해 등록되어야 합니다. 또한 네임플러스를 통해 추가된 서브도메인은 반영되는데 **24시간~48시간 이상** 걸릴 수 있습니다. (전세계 IDC에 정보들이 업데이트 되어야 하므로)

# Image Upload

Logo 혹은 Text를 Mapping하기 위해 Upload하는 과정을 서술합니다.

## 사용된 Open Source, 기술 URL

- [qqUploader](#)

## 사용 예시

어떠한 연유로 Upload를 진행할 때 [qqUploader](#)를 쓰게 되었는지는 모르겠지만, WebTex3D 내부에서 써 왔던 플러그인이다 보니, 이를 그대로 응용하여 Upload하는 과정을 작성하였습니다.

### fh.viewer.sticker.js

```
wvs.uploadImage = function(flag, src) {
  var uploaderTest = new qq.FileUploaderBasic({
    params: {
      currentGrid: wvs.currentGrid,
      //imgSizeLimit : '10000x10000'
    },
    allowedExtensions: ['jpg', 'jpeg', 'png', 'gif', 'tif', 'tiff'],
    button: $('.popup-upload-logo')[0],
    text: '',
    action: '/upload/logo',
    onComplete: function(id, fileName, response) {
      if (response['success']) {
        var logoId = response['logoId'];
        var logoName = response['name'];
        var thumb_url = response['thumb_url'];
        var image_url = response['image_url'];
        var width = response['imgWidth'];
        var height = response['imgWidth'];
        var userId = response['userId'];
        console.log(response);
      }
    }
  });
  ...
}
```

**wvs.uploadImage**에서는 사용자가 그리드 설정 창에서 원하는 로고나 텍스트를 submit했을 때 flag(logo인지, text인지) 와 src(해당 그리드의 순번)을 parameter로 삼아 upload를 진행합니다.

### /mysite/custom/views.py

```
scaleRatio=4
```

이미지나 텍스트를 고해상도로 출력하기 위하여 다음과 같이 전역변수를 설정하였습니다. 여기서는 4배 높아진 해상도로 출력하려 합니다.

```
def uploadLogo(request):
    IMGFILE_SIZE_LIMIT = 10 * 1024 * 1024
    allowedExt = ['jpg', 'jpeg', 'png', 'gif'] # allowed extension list
    exceptionalExt = ['tif', 'tiff']

    gridHeight = scaleRatio*int(float(request.GET['currentGrid[height]']))
    gridWidth = scaleRatio*int(float(request.GET['currentGrid[width]']))
    name = request.FILES['qqfile'].name
    logoImage = request.FILES['qqfile']

    gridSize = (gridWidth, gridHeight)
    print gridSize
    # if file size is grater than 10MB
    if logoImage.size >= IMGFILE_SIZE_LIMIT:
        try:
            raise UploadedImgFileError('FileSizeExceeded')
        except UploadedImgFileError as e:
            response = {'success': False, 'reason': e.reason}
            return HttpResponse(json.dumps(response))

    IMAGE_EXTENSION = os.path.splitext(name)[-1]
    print 'imgObj'
```

`IMGFILE_SIZE_LIMIT`, `allowedExt`, `exceptionalExt` 는 qqUploader에서 upload되는 파일을 제한하기 위해 사용됩니다.

중요한 것은 `gridHeight`, `gridWidth`인데요, 그리드의 가로 세로의 값에 `scaleRatio`를 곱하여 `canvas`를 지정합니다. 가끔 `currentGrid`값에 소수점이 포함될 때가 있어 `float`으로 받고, 이를 간편히 쓰기 위해 `int`로 항변환합니다.

```
if request.user.is_anonymous():
    logo = upload_logo(name=name)
else:
    logo = upload_logo(user=request.user, name=name)

imgObj = Image.open(logoImage)
imgObj = imgObj.crop(imgObj.getbbox())
imgObj.thumbnail(gridSize, Image.ANTIALIAS)
background = Image.new('RGBA', (gridWidth, gridHeight), (255, 255, 255, 0))
background.paste(imgObj, ((gridSize[0] - imgObj.size[0]) / 2, (gridSize[1] - imgObj
.size[1]) / 2))
background.mode = 'RGBA'
temp_handle = StringIO()
background.save(temp_handle, 'PNG')
temp_handle.seek(0)

logoImage = ContentFile(temp_handle.read())
IMAGE_EXTENSION = '.png'
logo.save()
```

`Image` 라이브러리를 사용하여 `imgObj`라는 객체에 원하는 `Image`를 로드합니다. `Image` 라이브러리에 내장되어 있는 `getbbox`를 통해 로고 캔버스의 여백을 잘라주고, `imgObj`을 아까 설정한 그리드 사이즈에 맞게 `Antialiasing`하면서 제작해 줍니다.

`imgObj`는 투명한 상태 (`RGBA`)로 저장되어야만 합니다. 그래야 `mapping`했을때 바탕에 색 없이 이쁘게 `mapping`될 것입니다. 때문에, `canvas`의 배경인 `background`을 투명하게 만들어 붙여줍니다.

이후 `temp_handle`에 해당 객체를 저장하여 `logoImage`라고 만들어 줍니다. `logoImage`는 원본 이미지를 해당 그리드의 4배 크기 투명한 `Canvas`에 크기를 맞춰 저장한 것이겠네요.

```
image_filename = 'data/logo_img/' + str(logo.id/10000) + '/' + str(logo.id) + IMAGE_EXTENSION
logo.path.save(image_filename, ContentFile(logoImage.read()))

img = Image.open(StringIO(logo.path.read()))

logo.width, logo.height = img.size

THUMBNAIL_SIZE = (115, 115)
IMAGE_TYPE = imghdr.what(logo.path.path)

if not IMAGE_TYPE:
    IMAGE_TYPE = img.format.lower()

img.thumbnail(THUMBNAIL_SIZE, Image.ANTIALIAS)
temp_handle = StringIO()
img.save(temp_handle, IMAGE_TYPE)

temp_handle.seek(0)

thumb_filename = 'data/logo_thumbs/' + str(logo.id/10000) + '/' + str(logo.id) + IMAGE_EXTENSION
logo.thumbnail.save(thumb_filename, ContentFile(temp_handle.read()))

thumb_url = settings.MEDIA_URL + thumb_filename
image_url = settings.MEDIA_URL + image_filename

...
```

마지막입니다. 저장된 파일의 이름을 설정하고, Data와 DB를 저장합니다. 같은 방식으로 이미지의 썸네일도 제작하여 저장합니다. 참 쉽죠?

# Draw Text in Canvas

텍스트를 웹에서 그리기 위하여, HTML5의 Canvas를 이용하여 기술을 구현하였습니다.

## 사용된 Open Source, 기술 URL

- [HTML5 Canvas Text Render](#)
- [Use Webfont on Canvas](#)
- [How to save canvas as png image](#)
- [Canvas 기술에 대한 자세한 명세 및 예제](#)
- [조금은 다른 형태의 arch 형 text 구현](#)
- [outline style 속성 변경](#)
- [캔버스와 관련된 속성](#)

## 사용 예시

주축이 되는 Canvas 기술은 HTML5를 지원하는 모든 웹 브라우저에서 작동합니다.

### Fill and Stroke text



텍스트를 스타일과 함께 입힐 수 있습니다. **stroke**를 이용해 텍스트의 외곽선을 그려주고 **fill**을 이용해 텍스트 본문을 겹쳐 아웃라인이 같이 있는 글자를 만들 수 있습니다. 외곽선의 갯수는 두께에 따라 무한히 그릴 수 있으며 색상 선택도 자유롭습니다.

### Text Outline Style



위와 같은 Outline 문제가 있었는데, lineCap, lineJoin 속성을 이용해 아래와 같이 표현할 수 있습니다.

- lineJoin = round

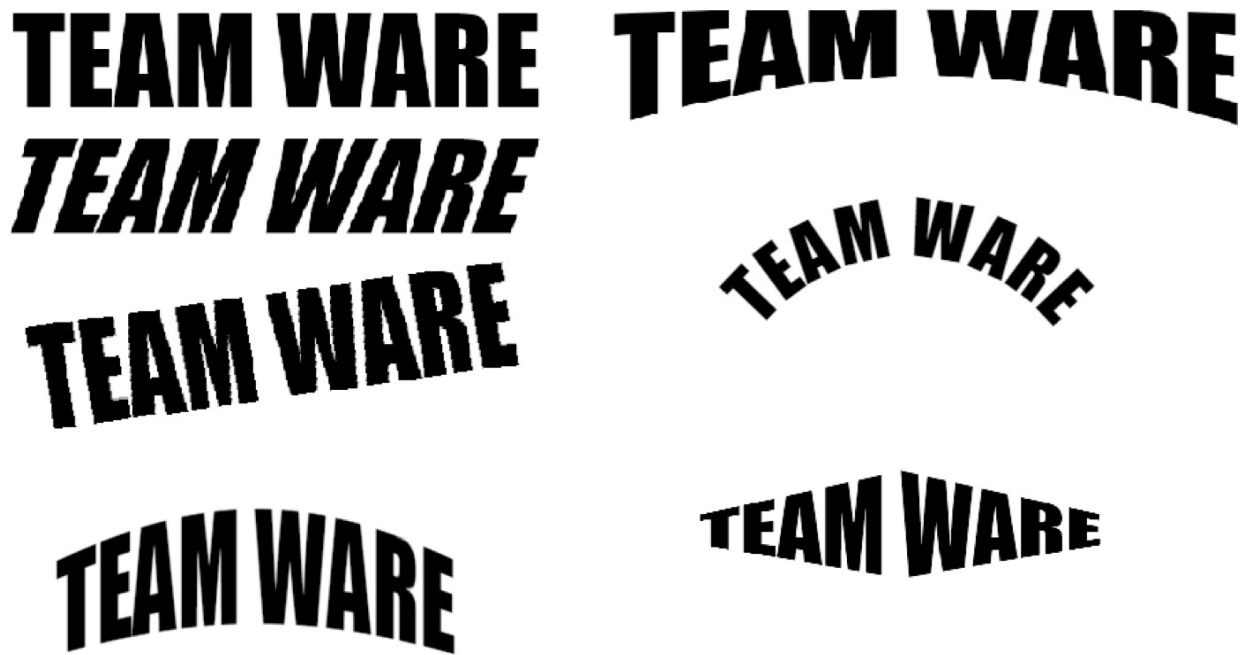


- lineJoin = bevel



## Curve / Shape





만들어진 텍스트를 기울거나, 곡선화 시키거나, 모양을 만들 수 있습니다. 아치 모양, 리본 모양 등 의뢰서에서 요구된 모양을 모두 구현할 수 있었습니다. 하지만 좌측 하단의 Text Warp 구현 부분에 대해서는 조금 더 작업이 필요합니다. 우측 하단은 요청사항은 없었으나 제가 따로 제작해본 모양입니다.

## Canvas 2 Image

다 만들어진 캔버스를 base64를 통해 png 형식으로 저장할 수 있습니다. 이에 대해서는 다음 [How to save canvas as png image](#)를 통해 알 수 있습니다.

## 코드 설명

webtex3d-uniform 프로젝트의 코드를 기준으로 서술합니다.

`fh.viewer.sticker.js`

```
function drawText() {
    textValue = $('#text-value').val();

    console.log(shape);
    var dragging;
    init();

    function init() {
        drawShapes();

        c.addEventListener("mousedown", mouseDownListener, false);
    }
}
```

Text를 canvas에 그리기 위해서, 다음 3가지를 진행합니다.

- 작성될 텍스트의 값 (`textValue`)를 갖고오기
- 사용자가 글씨를 `drag`할때 사용되는 `dragging` 이라는 변수 작성
- 초기화 ( `drawShapes` 라는 함수를 부르면서, `mousedown`에 대하여 `EventListener( mouseDownListener )`를 붙입니다)

```
function mouseDownListener(evt) {
    var bRect = c.getBoundingClientRect();
    mouseX = (evt.clientX - bRect.left) * (c.width / bRect.width);
    mouseY = (evt.clientY - bRect.top) * (c.height / bRect.height);

    if (hitTest(shape, mouseX, mouseY)) {
        dragging = true;
        dragHoldX = mouseX - shape.x;
        dragHoldY = mouseY - shape.y;
    }

    if (dragging) {
        window.addEventListener("mousemove", mouseMoveListener, false);
    }
    c.removeEventListener("mousedown", mouseDownListener, false);
    window.addEventListener("mouseup", mouseUpListener, false);

    if (evt.preventDefault) {
        evt.preventDefault();
    } //standard
    else if (evt.returnValue) {
        evt.returnValue = false;
    } //older IE
    return false;
}
```

`mouseDownListener`에서는 `mouse`의 `x, y` 좌표와 현재 `action` 상태 (`mousedown`, `mouseup`, `mousemove` 등)을 체크하여 액션을 합니다.

```
function mouseUpListener(evt) {
    c.addEventListener("mousedown", mouseDownListener, false);
    window.removeEventListener("mouseup", mouseUpListener, false);
    if (dragging) {
        dragging = false;
        window.removeEventListener("mousemove", mouseMoveListener, false);
    }
}
```

기본적으로 drag는 마우스를 눌렀을 때 작동하고, 움직일 땐 좌표가 변하고, 마우스를 떼었을 때는 끝나는 것이겠지요?

`mouseUpListener` 를 이용해 `mouseUp`상태에서는 `dragging`을 안하는 것으로 간주합니다.

```
function mouseMoveListener(evt) {
    var posX;
    var posY;
    var minX = shape.width;
    var maxX = c.width - shape.width;
    var minY = shape.height;
    var maxY = c.height - shape.height;
    var bRect = c.getBoundingClientRect();
    mouseX = (evt.clientX - bRect.left) * (c.width / bRect.width);
    mouseY = (evt.clientY - bRect.top) * (c.height / bRect.height);

    posX = mouseX - dragHoldX;
    posY = mouseY - dragHoldY;

    shape.x = posX;
    shape.y = posY;

    drawShapes();
}
```

위와 비슷한 방식으로 `mouseMoveListener` 도 설정합니다. `mouseX`와 `mouseY`는 현재 작업이 일어나는 캔버스의 x, y값과 canvas에 그려진 물체 (텍스트)의 크기가 필요합니다.

이 후의 코드는 본격적으로 canvas에 text를 draw하는 방식입니다.

```
function drawShapes() {
    ctx.clearRect(0, 0, wvs.currentGrid.width * scaleRatio, wvs.currentGrid.height
    * scaleRatio);
    var optionValue = $('input[name=inlineRadioOptions]:checked').val();
    var checkOutline1 = $('input[name=outline]:checked').attr('id');
    var finalRatio = wvs.currentGrid.height * textRatio[$('input[name=textSize]:ch
    ecked').val()];
    var textFont = font,
        textSize = scaleRatio * finalRatio + 'px',
        textColor = $('#main-color').spectrum("get").toHexString(),
        textItalic = false,
        textRotate = false,
        offsetX = shape.x,
        offsetY = shape.y,
        textOutline1 = $('#outline1-color').spectrum("get").toHexString(),
        textOutline2 = $('#outline2-color').spectrum("get").toHexString(),
        textBaseline = "top",
        textArc = false,
        textArcUpper = false;
```

ctx(context)를 clearRect를 이용하여 초기화합니다. (지워줍니다.)

그 후에는 canvas draw에 필요한 다양한 option값들을 체크하여 값으로 넣어줍니다.

```
switch (optionValue) {

    case 'option2':
        textItalic = true;
        break;
    case 'option3':
        textItalic = true;
        textRotate = true;
        break;
    case 'option4':
        textArc = true;
        break;
    case 'option5':
        textArcUpper = true;
        // alert('this version have no function CURVED.');
```

optionValue를 통해서 글씨가 뿌려지는 방법에 대하여 체크 할 수 있습니다. Arc형이나 Italic 등에 대해서 말이지요. option5는 ArcUpper Style로 뿌려주는 방법인데, 현재는 outline color 등에 있어서 문제가 있기 때문에 막아 두었습니다.

```
if (textItalic) {
    ctx.font = 'italic ' + textSize + ' ' + textFont;
} else {
    ctx.font = textSize + ' ' + textFont;
}
ctx.textBaseline = textBaseline;
ctx.textAlign = "center";
if (textRotate) {
    ctx.rotate(-13 * Math.PI / 180);
}
if ($('#outline2').prop('checked')) {
    ctx.strokeStyle = textOutline2;
    ctx.lineWidth = 0.8 * finalRatio;
    ctx.lineJoin = 'round';
    if (textArc) {
        ctx.strokeText(textValue, c.width * 0.5, 0);
    } else {
        ctx.strokeText(textValue, offsetX, offsetY);
    }
}
if ($('#outline1').prop('checked')) {
    ctx.strokeStyle = textOutline1;
    ctx.lineWidth = 0.4 * finalRatio;
    ctx.lineJoin = 'round';
    if (textArc) {
        ctx.strokeText(textValue, c.width * 0.5, 0);
    } else {
        ctx.strokeText(textValue, offsetX, offsetY);
    }
}
```

**Italic**, **Rotate** 등에 대하여 스타일을 작성합니다. 또한, **Outline**의 경우 차례차례 외곽선을 그리고 그 위에 다시 그려주는 방식으로 표현을 합니다.

```

ctx.fillStyle = textColor;

if (textArc) {
    ctx.fillText(textValue, c.width * 0.5, 0);
    var i = c.width;
    var angleSteps = 180 / i;

    while (i--) {
        var y = 200 - 140 * Math.sin(i * angleSteps * Math.PI / 180);
        ctx.drawImage(c, i, 0, 1, 80, i, c.height * 0.45 - 10 / 45 * y, 1, y);
    }
    ctx.clearRect(0, 0, c.width, 80);
} else if (textArcUpper) {
    getCircularText(textValue, 600, 0, "center", false, true, textFont, textSize, 5);
} else {
    ctx.fillText(textValue, offsetX, offsetY);
}
if (textRotate) {
    ctx.rotate(13 * Math.PI / 180);
}
}
}

```

이후 `textColor`를 `fillStyle`로 색칠해줍니다. 마지막으로 `textArc`와 `textArcUpper`, `textRotate`의 옵션에 맞춰 이미지를 그려줍니다. `fillText`를 하는 기점에서 실제로 `text`가 `canvas`에 표현되게 됩니다.

## 문제점 및 고찰

1. `textArcUpper` 스타일에 버그가 있습니다.
  - `textArcUpper` 스타일은 기본적으로 유니폼의 텍스트 스타일에 많이 쓰입니다.
  - 하지만 아까 서술한 것 처럼, `outline`처리와 더불어 `canvas` 내부에서의 이동 등에 버그가 있습니다.
  - `getCircularText` 함수에서 버그가 수정되어야 합니다.
2. 각 스타일 옵션 들이 모듈화 되어있지 않습니다.
  - `getCircularText` 함수와 같이, 스타일별로 적절하게 모듈화가 이루어져야 합니다.
  - 모듈화의 장점인 개발 및 관리의 용이성 때문입니다.

# Mapping in Grid Group

Grid Group에 로고, 텍스트를 Mapping하기 위한 방법입니다.

## 사용 예시

webtex3d는 기본적으로 gridGroup의 구역에 mapping을 원하는 fabric을 drag하여 사용하는 방식이었습니다. 하지만 유니폼 프로젝트와 같이 임의로 만든 data (logo, text, colorchip 등)를 mapping해야 하는 방식이 되었습니다.

그에 따라 webtex3d를 크게 고치지 않고서 임의로 만든 data를 mapping하는 방법을 소개합니다.

### fh.viewer.stiker.js

```
wvs.uploadImage = function(flag, src) {
    var uploaderTest = new qq.FileUploaderBasic({
        params: {
            currentGrid: wvs.currentGrid,
            //imgSizeLimit : '10000x10000'
        },
        allowedExtensions: ['jpg', 'jpeg', 'png', 'gif', 'tif', 'tiff'],
        button: $('#.popup-upload-logo')[0],
        text: '',
        action: '/upload/logo',
        onComplete: function(id, fileName, response) {
            if (response['success']) {
                var logoId = response['logoId'];
                var logoName = response['name'];
                var thumb_url = response['thumb_url'];
                var image_url = response['image_url'];
                var width = response['imgWidth'];
                var height = response['imgHeight'];
                var userId = response['userId'];
            }
        }
    });
}
```

wvs.uploadImage (혹은 uploadText) 함수를 보면 업로드가 진행될 때의 과정을 보여주고 있습니다. 성공적으로 업로드가 진행되었을 때 onComplete 에서 response를 받아와 해당 data들을 읽어옵니다.

```

//change the global list
$('.layers').each(function() {
    if ($(this).is(':last-child')) {
        var idList = $(this).data('idList');
        idList[src] = response['image_url'];
        console.log($(this).data('idList'));
    }

});
wv.runSimulation(wv.pid, true);

wvs.thumbLogo(thumb_url, logoName);
wvs.parseLogo();

```

webtex3d는 `wv.runSimulation` 을 할 때 모든 `.layers` 의 DOM을 체크하면서 각 layer의 grid group 설정값 (몇번 grid가 어떤 fabric으로 mapping될 것인지 등)에 대한 정보들을 참조하여 simulation을 진행합니다. (sid, fids를 넘겨주면서요) 그럼 임의의 값을 강제로 적용하면 `wv.runSimulation` 은 이를 그대로 참조하여 simulation을 하겠죠?

`.layers` 의 마지막 DOM은 우리가 mapping을 진행하려는 유니폼의 레이어입니다. (최상단 레이어!) 그렇기 때문에 이 DOM에 원하는 data를 넣습니다.

`wvs.uploadImage` 는 src라는, 몇 번째 grid에 mapping할 것인지를 알 수 있는 parameter가 있습니다. `.layers:last-child` 의 `idList` (grid group의 정보를 갖고 있는 배열)에서, src번째 배열을 찾아 원하는 data를 넣어줍니다. 우리는 `response['image_url']` 의 값을 넣어 줍니다.

그 후 simulation을 진행하면 우리가 넣었던 `response['image_url']` 이 src번째 grid group에 넣어진채로 simulation image가 출력됩니다.



# Transfer Data to JoyCART

The screenshot shows the JoyCART website interface. At the top, there's a navigation bar with links like MEDIA, ART, SCHOOL, BLOG, STUDY, TEMPLATE AE, TEMPLATE, INTERIOR, and others. Below the navigation bar is a header with the logo 'SKO by Fukushima Dress' and the text 'カスタムオーダーの和服スーツ'. The main content area is divided into three steps: 'お客様情報入力' (Customer Information Input), 'ご入力内容の確認' (Check Input Content), and 'ご注文完了' (Order Completed). The first step is active, showing a table for the shopping cart and a form for customer information and size input.

商品	販売価格 (税抜)	数量	変更	小計
カートは空です。				
商品合計				0円

**お客様基本情報の入力**

お名前 (必須)  例) 山田 太郎

フリガナ  例) ヤマダ タロウ

国名  Japan/日本

郵便番号  郵便番号から入力 例) 1000000

都道府県

住所1 (必須)  市区町村、番地等

住所2  アパート・マンション名、部屋番号等

メールアドレス (必須)  携帯電話のメールアドレスをご利用の場合は、[xxx@xxxx.com]からのメールが受信できるようにご設定ください。

電話番号 (必須)  例) 0312345678 ※ 電話番号、または携帯電話のいずれかを入力してください。

FAX番号  例) 0312345678

**お客様サイズ情報の入力**

身長 (必須)  cm

体重 (必須)  kg

上身丈  mm (首の付け根のぐりぐりから胴囲部分まで) [サイズの測り方はこちら](#)

後首まわり  mm (首の横から後ろの首の付け根のぐりぐりまで) [サイズの測り方はこちら](#)

My Favorite에 있는 내용을 JoyCART라는 일본의 결제 페이지로 전달하기 위해서 사용한 방법입니다.

## 사용 예시

일본 클라이언트사에서 JoyCART의 사용을 요구하며 부탁한 조건은 다음과 같습니다.

1. `name="data"` 인 부분을 `textarea`로 하여, 여기에 `data`를 넣어 전송할 것
2. `textarea`로는 `csv`형식으로 `comma`를 이용하여 값을 구분하고, 줄로 `product`를 구별하게 할 것.
3. `name="data"` 외 전송되는 데이터는 무시함.
4. 원하는 url로 `post`할 수 있도록 한다.

이 조건을 가지고 생각해보면, submit버튼을 눌렀을 때 data를 처리하여 보내야 합니다. 처음에는 ajax로 json 형식 data를 제출하려 했으나, joyCART에서는 form에서 post로 submit된 데이터만 받아들이려 했기 때문에 다음과 같이 구성하였습니다.

### viewer.html

```
<form id="formCart" class="modal-content" method="post" action="http://160.16.63.102/cms/xp2016-fuku/" target="_self">
  <div class="modal-header">
    ...
  </div>
  <div class="modal-body">
    <div id="formWishlist">

      </div>
    </div>
    <div class="modal-footer">
      ...
      <button type="submit" class="btn btn-danger">發注</button>
      <textarea id="hiddenData" style="display:none;" name="data"></textarea>
    </div>
</form>
```

viewer의 마크업 중 myFavorite Modal 부분입니다.

.modal-header 부터 .modal-footer 까지 form이 둘러쌓여 있고, submit 버튼을 마크업 해두었습니다. 그 뒷부분을 보면 name="data" 를 가지는 textarea가 마크업 되어 있습니다.

form에는 action을 보낼 곳과 보내는 방법인 method가 post로 설정되어 있네요. 이 form이 submit될 때, name="data" 인 textarea의 value를 설정해서 같이 post하면 좋겠죠?

### fh.viewer.js

```
$('#formCart').submit(function () {
  var checkAmount = $('input[name="product"]:checked').length;
  var checkTotal = $('input[name="product"]').length;
  var jsonData = [];

  $('.item-list').each(function () {
    var currentList = $(this).attr('id');
    if ($('input:checkbox[name="product"]', this).is(":checked")) {
      jsonData.push(JSON.stringify($('#' + currentList + ' :input').serializeObject()));
    }
  });
  jsonData = JSON.stringify(jsonData);
```

그래서 `.submit()`이라는 함수가 있습니다. 대상이 `submit`을 할때, `submit`이전에 일어나는 일들을 처리할 수 있습니다. `.submit`으로 불러오는 `callback`에서 우리가 원하는 데이터들을 정리해서 전달할 수 있도록 준비해 놔야겠지요?

저는 처음에 당연히 JSON형식으로 전달할 수 있을 거라 생각하고 코드를 작성했습니다. (한국 쇼핑몰은 거의 JSON형식으로 `data`를 전달합니다) 그래서 각 아이템들을 돌면서 `checkbox`에 체크가 되어있는지 확인 후, 필요한 정보들을 JSON `data` 형식으로 임시 변수에 저장해 두었습니다.

```
if (jsonData.length == 0) {
    alert('チェックされたアイテムがありません。');
    return false;
} else {
    jsonData = JSON.parse(jsonData);
    var tempString = '';
    for (var i = 0; i < jsonData.length; i++) {
        var temp = JSON.parse(jsonData[i]),
            add1 = 'none',
            add2 = 'none';

        if (typeof temp['add1'] != 'undefined') {
            add1 = temp['add1'];
        }
        if (typeof temp['add2'] != 'undefined') {
            add2 = temp['add2'];
        }
    }
}
```

그렇지만 JSON변수가 아닌 `csv` 형태의 `string`을 전달해야 한다고 하니까, 다음과 같이 모아둔 `jsonData`를 다시 `parse`하여 정리합니다.

```
tempString += temp['sex'] + ',' + temp['product'] + ',' + temp['quantity'] +
    ',' + temp['price'] + ',' + temp['size'] + ',' + temp['unit'] + ',' + add1 + ',' + ad
d2 + ',' + window.location.protocol + '//' + window.location.host + temp['url'] + '\n'
;
    }
    $('#hiddenData').html(tempString);
    return true;
}
});
```

마지막으로 데이터들을 `tempString`, 우리가 전달할 `name="data"` 의 `textarea`에 넣어주면 되겠지요. 그 후 `return true;` 를 통해 함수를 종료하고, `submit`을 실행하게 됩니다.

그러면 `target="_self"` 로 잡혀 있었던 `form`은 `name="data"` 의 `textarea`의 데이터를 전달하면서 `http://160.16.63.102/cms/xp2016-fuku/` 로 이동하겠네요!



# Multi-part Simulation

유니폼에서 각 파트별 시뮬레이션을 진행하기 위해 다음과 같은 연구가 진행되었습니다.

## 원리 및 요구사항



1. 유니폼 사진을 Collar, Top, Bottom, Socks Layer로 나누어 3D 작업을 합니다. (Collar만 우선적으로 부분 변경이 가능하게 하고, 추후 추가개발을 통하여 반팔과 긴팔을 변환할 수 있도록 하려 합니다)

이 때, Detail 변경이 가능한 Collar와 Sleeve는 각 스타일마다 레이어를 생성하여 작업합니다.

2. Web상에 업로드 되어 있는 유니폼을 선택한 후, Detail을 변경할 수 있습니다. 특히, 각 txd파일 별로 mapping을 진행합니다.

이 부분이 가능한 이유는 각 파일이 투명한 바탕의 PNG파일로 나오기 때문입니다. 이를 elite에서 사용하는 것처럼 Web상에서 겹쳐 보여주게 되면, 단벌의 코드로 나오게 됩니다.

이를 위해서는 다음과 같은 개발이 필요합니다.

1. ww.runSimulation 과정의 간소화 및 restfulAPI 변환

시뮬레이션 부분이 현재 WebTex3D와는 다르게 간소화 되어 진행되어야 합니다. 예를 들면 다음과 같이요.

```
//전체적으로 현재 프로덕트들이 어떤 상황인지를 한꺼번에 관리합니다.
wv.state = {
  product: {
    3132: {
      id: 3132
      fids: [1321, 10, 212, 322]
      name: '스포츠 탑 A'
      type: 'jacket'
    }
    3222: {
      id: 3222
      fids: [23, 22, 2]
      name: '스포츠 반바지 A'
      type: 'bottom'
    }
  }
  ...
}
```

```
//fabric이 바뀔 경우 다음과 같이 변경을 합니다.
wv.changeFabric = function(pid, gridOrder, fid){
  wv.state.product[pid].fids[gridOrder]=fid;
  wv.runSimulation(wv.state.product[pid]);
  wv.printImage(pid); //해당 pid의 이미지를 뿌려준다.
}
```

```
//시뮬레이션은 다음처럼 이미지 경로만 나오도록 깔끔하게!
wv.runSimulation = function(obj) {
  $.ajax({
    url: 'simulation',
    type: 'POST',
    cache: false,
    data: obj,
    dataType: 'json',
    async: false,
    success: function(response) {
      return response[obj].src
    },
    error: function() {
      ...
    }
  });
}
```

위 작업은 [React](#)를 사용하면 매우 쉽습니다.

2. 각 복종별 디자이너의 txd파일 제작
3. WebTex3D에 이미지를 겹쳐서 넣을 수 있도록 다중으로 div를 올리고 action을 설정

