



# Ubuntu Packaging

## 1. Make package

심경섭 (<https://kssim.com>)

# 목차

- 개요
- Debian 디렉토리
- 환경 설정
- Debian Packaging
- QnA
- 실습



개요

# Packaging ?

- Application 과 관련된 파일들을 쉽게 배포하고, 설치하기 위한 도구? 절차?
- 통상적으로 하나의 패키지로 만드는 과정과 패키지를 배포하는 과정을 의미
- iOS 나 Android 에서 App Store 나 Play Store 에 App 을 배포하는 것과 유사

# App 배포 프로세스



```
<?xml version="1.0" encoding="utf-8"?>  
  
<manifest>  
  
    <uses-permission />  
    <permission />  
    <permission-tree />  
    <permission-group />  
    <instrumentation />
```

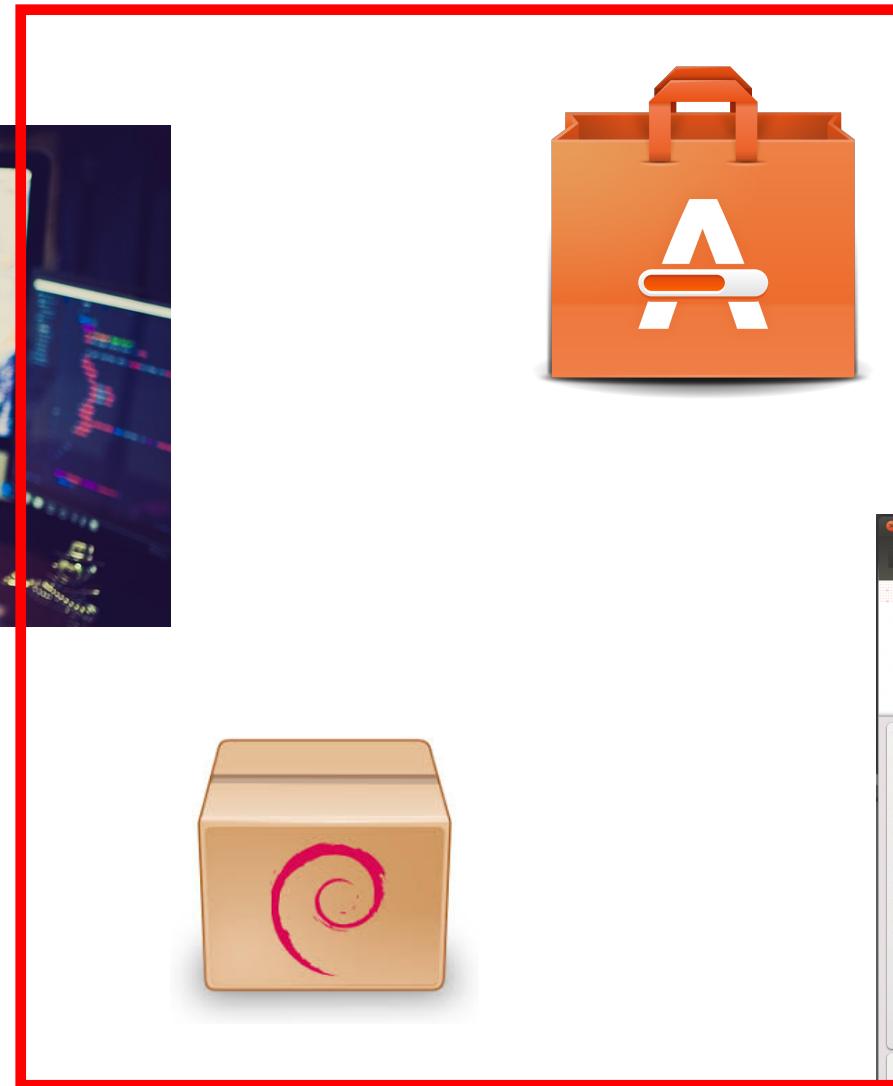


# Ubuntu Package 배포 프로세스





# 실습 내용



# 실습 내용



# 실습 내용



[Overview](#) [Code](#) [Bugs](#) [Blueprints](#) [Translations](#) [Answers](#)

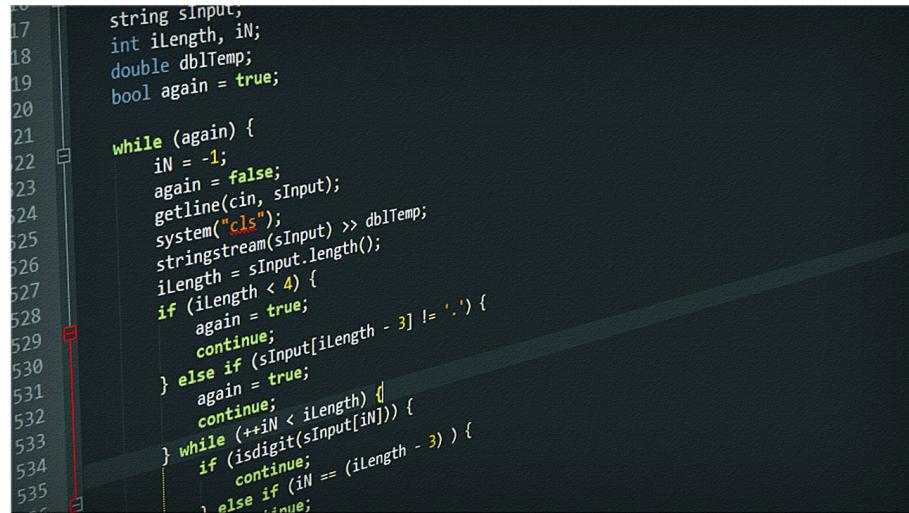
Personal Package Archives for Ubuntu



# 패키지 만들기



# 패키지 만들기



```
17 string sInput;
18 int iLength, iN;
19 double dblTemp;
20 bool again = true;
21
22 while (again) {
23     iN = -1;
24     again = false;
25     getline(cin, sInput);
26     system("cls");
27     stringstream(sInput) >> dblTemp;
28     iLength = sInput.length();
29     if (iLength < 4) {
30         again = true;
31         continue;
32     } else if (sInput[iLength - 3] != '.') {
33         again = true;
34         continue;
35     } while (++iN < iLength) {
36         if (isdigit(sInput[iN])) {
37             continue;
38         } else if (iN == (iLength - 3)) {
39             continue;
40         }
41     }
42 }
```



```
<?xml version="1.0" encoding="utf-8"?>
<manifest>
    <uses-permission />
    <permission />
    <permission-tree />
    <permission-group />
    <instrumentation />
```

# Package 의 Manifest



## Debian Kernel Source Code

arch	fs	MAINTAINERS	security
block	include	Makefile	sound
COPYING	init	mm	tools
CREDITS	ipc	net	usr
crypto	Kbuild	README	virt
<b>debian</b>	Kconfig	REPORTING-BUGS	
Documentation	kernel	samples	
drivers	lib	scripts	

# Ubuntu Package 의 Manifest

```
abi  
bin  
changelog  
compat  
config  
config.defines.dump  
control  
control.md5sum  
copyright  
installer  
lib  
patches  
po  
README.Debian  
README.source  
rules  
rules.defs  
rules.gen  
rules.real  
source  
templates  
watch
```

## Debian Kernel Source Code

arch	fs	MAINTAINERS	security
block	include	Makefile	sound
COPYING	init	mm	tools
CREDITS	ipc	net	usr
crypto	Kbuild	README	virt
debian	Kconfig	REPORTING-BUGS	
Documentation	kernel	samples	
drivers	lib	scripts	



# Debian 디렉토리

# Debian 디렉토리

- Debian pacakge 를 구성하기 위한 패키지 정보를 담은 디렉토리
- changelog, control, copyright, rules 파일이 가장 중요
- 그 외, compat, README, docs, source/format 등 여러 파일들이 있음

# changelog

- Debian 패키지의 변경 기록을 적은 파일
- 패키지 명과 버전, release 버전, 변경 내역등이 작성됨
- 버전은 "<upstream 버전>-<debian 버전><우분투 버전>" 으로 작성
  - Ex) 1.0.0-1ubuntu-1
  - Debian 에 없는 패키지인 경우, debian 버전을 0으로 표기함

# changelog

- 변경 내역에 Launchpad 의 이슈가 있을 경우, "LP: #<number>" 로 작성
- 핵심 변경 사항은 "\*" 를 사용해서 표기
- 핵심 변경 사항과 관련된 하위 변경 사항은 "-" 을 사용해서 표기
- 작성 일자는 RFC 5322 형식에 맞춰서 작성

# changelog

```
1 hello (1.0.0-0ubuntu1) xenial; urgency=medium
2
3   [ kssim ]
4   * Initial Release.
5   * Major Point
6     - Minor Point
7
8   [ test ]
9   * major
10
11 -- KyeongSeob Sim <ksub0912@gmail.com> Sat, 24 Mar 2018 08:17:56 +0000
```

# control

- 패키지의 메타 정보를 기록한 파일
  - Source 패키지의 정보와 생성할 debian 패키지의 정보를 작성
- Source 패키지 정보
  - Source 이름, section, maintainer, build-depends 등을 기록
- Debian 패키지 정보
  - Package 이름, architecture, depends, description 등을 기록
- Maintainer
  - Ubuntu 에서는 maintainer 를 Ubuntu Developers 로 설정

# control

```
1 Source: hello
2 Section: misc
3 Priority: optional
4 Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
5 XSBC-Original-Maintainer: KyeongSeob Sim <ksub0912@gmail.com>
6 Build-Depends: debhelper (>=9)
7 Standards-Version: 3.9.6
8 Homepage: https://kssim.com
9 #Vcs-Git: git://anonscm.debian.org/collab-maint/hello.git
10 #Vcs-Browser: https://anonscm.debian.org/cgit/collab-maint/hello.git
11
12 Package: hello
13 Architecture: any
14 Depends: ${shlibs:Depends}, ${misc:Depends}
15 Description: sample package.
16 ubuntu packaging workshop sample package.
17
18 Package: hello-dev
19 Architecture: any
20 Depends: ${shlibs:Depends}, ${misc:Depends}
21 Description: developer library for sample package.
22 ubuntu packaging workshop sample package.
```

# copyright

- 패키지 내부 파일들의 라이센스를 명시
- 파일별, 디렉토리 별로 나눠서 작성할 수 있음

# copyright

```
1 Format: https://www.debian.org/doc/packaging-manuals/copyright-format/1.0/
2 Upstream-Name: hello
3 Source: <url://example.com>
4
5 Files: *
6 Copyright: <years> <put author's name and email here>
7           <years> <likewise for another author>
8 License: MIT
9
10 Files: debian/*
11 Copyright: 2018 KyeongSeob Sim <ksub0912@gmail.com>
12 License: MIT
13
14 License: MIT
15 Permission is hereby granted, free of charge, to any person obtaining a
16 copy of this software and associated documentation files (the "Software"),
17 to deal in the Software without restriction, including without limitation
18 the rights to use, copy, modify, merge, publish, distribute, sublicense,
19 and/or sell copies of the Software, and to permit persons to whom the
20 Software is furnished to do so, subject to the following conditions:
21 .
22 The above copyright notice and this permission notice shall be included
23 in all copies or substantial portions of the Software.
24 .
25 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
26 OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
27 MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
28 IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
29 CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
30 TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
31 SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

# README, doc

- 패키지를 사용하기 위한 정보를 담은 파일
- README 나 doc 파일을 만들어 내용을 작성
  - doc 파일은 upstream source 에 대한 문서
  - README 파일은 비표준적인 특징이 있는 경우 작성

# source/format

- 초기 생성된 내용을 그대로 유지
- 소스 패키지의 버전을 명시
  - 1.0 : 기본 형식 (Default 값)
  - 3.0 (quilt) : upstream 과 분리된
  - 3.0 (native) : debian native 패키지 (upstream 이 없음)

# rules

- Makefile 같이 패키지의 빌드를 위한 파일
- debhelper 에 의해서 많은 부분이 자동화 되어있음
  - dh 명령을 사용해서 빌드를 수행함
  - 설정을 커스터마이징 할 수 있음
- 패키징 로그는 “debian/package.debhelper.log” 파일에 기록됨

# rules

```
1 #!/usr/bin/make -f
2 # See debhelper(7) (uncomment to enable)
3 # output every command that modifies files on the build system.
4 #export DH_VERBOSE = 1
5
6
7 # see FEATURE AREAS in dpkg-buildflags(1)
8 #export DEB_BUILD_MAINT_OPTIONS = hardening=+all
9
10 # see ENVIRONMENT in dpkg-buildflags(1)
11 # package maintainers to append CFLAGS
12 #export DEB_CFLAGS_MAINT_APPEND = -Wall -pedantic
13 # package maintainers to append LDFLAGS
14 #export DEB_LDFLAGS_MAINT_APPEND = -Wl,--as-needed
15
16
17 %:
18     dh $@
19
20
21 # dh_make generated override targets
22 # This is example for Cmake (See https://bugs.debian.org/641051 )
23 #override_dh_auto_configure:
24 #     dh_auto_configure -- # -DCMAKE_LIBRARY_PATH=$(DEB_HOST_MULTIARCH)
```

# rules

```
#!/usr/bin/make -f
%:
    dh $@

override_dh_strip:
    dh_strip -Xfoo
```

```
#!/usr/bin/make -f
%:
    dh $@ --with python2
```

```
#!/usr/bin/make -f
%:
    dh $@ --buildsystem=perl_build
```

```
#!/usr/bin/make -f
%:
    dh $@ --parallel
```

```
#!/usr/bin/make -f
%:
    dh $@

override_dh_auto_build-indep:
    $(MAKE) -C docs

# No tests needed for docs
override_dh_auto_test-indep:

override_dh_auto_install-indep:
    $(MAKE) -C docs install
```

# install

- dh\_install 에 의해서 옮겨지는 파일 목록
- 생성되는 패키지가 하나면, 파일 명을 “install”로 설정
- 여러개의 패키지가 생성되면 “<패키지명>.install”로 설정

# watch

- upstream 패키지를 debian 패키지로 만들 때, 선택적으로 사용
- upstream 패키지의 업데이트 상태를 확인



# 환경 구축

# 패키징을 위한 패키지

- ubuntu-dev-tools
  - 패키징을 쉽게 도와주는 **tool** 들을 가지고 있는 패키지
  - devscripts, dpkg-dev, binutils 같은 패키지가 포함되어 있음
- devscripts
  - Debian 패키지 관리를 도와주는 **script** 들을 가지고 있는 패키지
  - Dch, debclean, debi, debpkg, debuild 등의 패키지가 포함되어 있음
- dh-make
  - Debian 패키지를 만들기 위한 debian 디렉토리를 생성해주는 툴
  - Upstream source code 를 사용하거나 native debian package 를 만들 수 있음

# 패키징 환경을 위한 패키지

- pbuilder
  - 시스템과 분리된 독립된 환경에서 패키지를 빌드할 수 있도록 해주는 tool
  - debootstrap 을 이용해서, chroot 환경을 만들어줌
- gnupg
  - 전자 서명을 위한 도구
  - Launchpad 에 업로드할 패키지를 서명하기 위한 tool
- haveged
  - 난수를 생성하는 도구

# GPG Key 설정

- “gpg –gen-key” 를 사용해서 키를 생성
  - 키 종류 : RSA or DSA
    - Default : RSA
  - 키 길이 : 2048 이상
    - Default : 2048
  - 키 유효기간 : 0 (없음)
    - Default : 0 (없음)
  - 이름 & 이메일 입력
- “gpg –send-keys” 를 사용해서 키를 등록
  - “gpg –send-keys –keyserver keyserver.ubuntu.com <KEY ID>”

# Committer 정보 등록

- Commit log 에 자동으로 작성자와 작성자 메일이 등록되게 설정
- ~/.bashrc 나 ~/.zshrc 같은 환경 설정 파일에 등록
- Debian 과 Ubuntu 에서 사용 가능
  - `export DEBFULLNAME="KyeongSeob Sim"`
  - `export DEBEMAIL="ksub0912@gmail.com"`
- Ubuntu 에서만 사용 가능
  - `Export UBUMAIL="KyeongSeob Sim <ksub0912@gmail.com>"`

# Committer 정보 등록

```
97 export DEBFULLNAME="KyeongSeob Sim"
98 export DEBMAIL="ksub0912@gmail.com"
99
100 export UBUMAIL="KyeongSeob Sim <ksub0912@gmail.com>"
```

# GPG Key 생성

```
kssim@dev: ~$ gpg --gen-key
gpg (GnuPG) 1.4.20; Copyright (C) 2015 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: directory `/home/kssim/.gnupg' created
gpg: new configuration file `/home/kssim/.gnupg/gpg.conf' created
gpg: WARNING: options in `/home/kssim/.gnupg/gpg.conf' are not yet active during this run
gpg: keyring `/home/kssim/.gnupg/secring.gpg' created
gpg: keyring `/home/kssim/.gnupg/pubring.gpg' created
Please select what kind of key you want:
 (1) RSA and RSA (default)
 (2) DSA and Elgamal
 (3) DSA (sign only)
 (4) RSA (sign only)
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048) 2048
Requested keysize is 2048 bits
Please specify how long the key should be valid.
      0 = key does not expire
      <n> = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
      <n>y = key expires in n years
Key is valid for? (0) 0
Key does not expire at all
Is this correct? (y/N) y
```

# GPG Key 생성

```
You need a user ID to identify your key; the software constructs the user ID
from the Real Name, Comment and Email Address in this form:
  "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"
```

```
Real name: KyeongSeob Sim
Email address: ksub0912@gmail.com
```

Comment:

You selected this USER-ID:  
 "KyeongSeob Sim <ksub0912@gmail.com>"

```
Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? o
You need a Passphrase to protect your secret key.
```

```
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
```

# GPG Key 생성

```
gpg: /home/kssim/.gnupg/trustdb.gpg: trustdb created
gpg: key 9FD41E43 marked as ultimately trusted
public and secret key created and signed.

gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0  valid:  1  signed:  0  trust: 0-, 0q, 0n, 0m, 0f, 1u
pub  2048R/9FD41E43 2018-03-24
      Key fingerprint = 0940 EE77 A7BA 47A8 3D00  5C0B 5B7E E948 9FD4 1E43
uid            KyeongSeob Sim <ksub0912@gmail.com>
sub  2048R/012DE7A5 2018-03-24
```

# GPG Key 등록

```
kssim@dev:~$ gpg --send-keys --keyserver keyserver.ubuntu.com 9FD41E43
gpg: sending key 9FD41E43 to hkp server keyserver.ubuntu.com
```

# Lanchpad 에 SSH Key 등록

- “ssh-keygen” 을 사용해서, ssh key 생성
- public key 를 Lanchpad 에 등록
  - <https://launchpad.net/~/+editsshkeys>

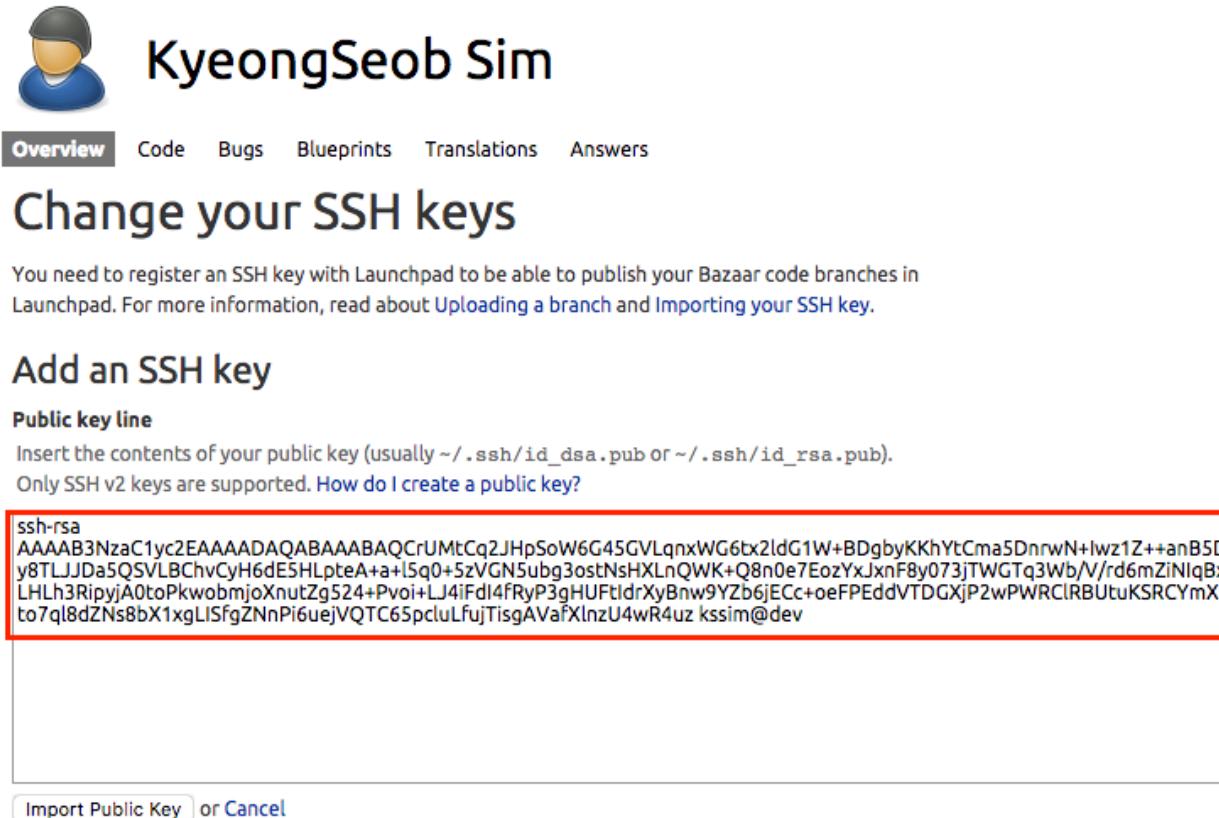
# SSH Key 생성

```
kssim@dev: ~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/kssim/.ssh/id_rsa):
Created directory '/home/kssim/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/kssim/.ssh/id_rsa.
Your public key has been saved in /home/kssim/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:aa+VK/LYUUTBUFFT8XMiZQEhMt892RWq6Mvb8ii61LQ kssim@dev
The key's randomart image is:
+---[RSA 2048]---+
|       o .oB**=|
|       + .+.++|
|       . +.=o=|
|       .... o+|
|       .S...     |
|       o.o. o    |
|       . E o=    |
|       . .+*+.   |
|       oo +B*o   |
+---[SHA256]---+
```

# SSH Key 생성

```
kssim@dev:~$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCrUMtCq2JHpSoW6G45GVLqnWG6tx21dG1
W+BDgbyKKhYtCma5DnrwN+Iwz1Z++anB5Dy8TLJJDa5QSVLBChvCyH6dE5HLpteA+a+l5q0+
5zVGN5ubg3ostNsHXLnQWk+Q8n0e7EozYxJxnF8y073jTWGTq3Wb/V/rd6mZiNIqBxLHLh3R
ipyjA0toPkwobmjoxnutZg524+Pvoi+LJ4iFdI4fRyP3gHUftIdrXyBnw9YZb6jECc+oeFPE
ddVTDGXjP2wPWRClRBUtukSRCYmX0to7ql8dZNs8bX1xgLISfgZNnPi6uejVQTC65pcluLfu
jTisgAVafXlnzU4wR4uz kssim@dev
```

# Launchpad 에 SSH Key 등록



KyeongSeob Sim

Overview    Code    Bugs    Blueprints    Translations    Answers

## Change your SSH keys

You need to register an SSH key with Launchpad to be able to publish your Bazaar code branches in Launchpad. For more information, read about [Uploading a branch](#) and [Importing your SSH key](#).

### Add an SSH key

**Public key line**

Insert the contents of your public key (usually `~/.ssh/id_dsa.pub` or `~/.ssh/id_rsa.pub`).  
Only SSH v2 keys are supported. [How do I create a public key?](#)

```
ssh-rsa AAAAAB3NzaC1yc2EAAAQABAAQCrUMtCq2JHpSoW6G45GVLqnxFW6tx2ldG1W+BDgbyKKhYtCma5DnrwN+lwz1Z++anB5Dy8TLJJDa5QSVLBChvCyH6dE5HLpteA+a+l5q0+5zVGNSubg3ostNsHXLnQWK+Q8n0e7EozYxJxnF8y073jTWGTq3Wb/V/rd6mZlNIqBxLHLh3RipyjA0toPkwobmjoXnutZg524+Pvoi+LJ4iFdI4fRyP3gHUFtldrXyBnw9YZb6jECc+oeFPEddVTDGXjP2wPWRClRBUtuKSRCYmX0to7ql8dZNs8bX1xgLISfgZNnPi6uejVQTC65pcluLfujTisgAVafXlnzU4wR4uz kssim@dev
```

or [Cancel](#)

# pbuild 환경 설정

- 빌드를 위한 깔끔한 환경을 제공해줌
- 시스템의 설정을 수정하지 않고, 다양한 빌드 환경을 만들어줌
- Ubuntu 나 Debian 의 다양한 릴리즈 환경을 지원
- “pbuilder-dist <release> create” 명령을 사용
  - Ex) pbuilder-dist xenial create
  - Ex) pbuilder-dist zesty create

# pbuild 환경 설정

```
kssim@dev:~$ pbuilder-dist xenial create
W: /home/kssim/.pbuilderrc does not exist
I: Logging to /home/kssim/pbuilder/xenial_result/last_operation.log
I: Distribution is xenial.
I: Current time: Sat Mar 24 06:02:29 UTC 2018
I: pbuilder-time-stamp: 1521871349
I: Building the build environment
I: running debootstrap
/usr/sbin/debootstrap
I: Retrieving InRelease
I: Checking Release signature
I: Valid Release signature (key id 790BC7277767219C42C86F933B4FE6ACC0B21F32)
I: Retrieving Packages
I: Validating Packages
I: Resolving dependencies of required packages...
I: Resolving dependencies of base packages...
```

# pbuild 환경 설정

```
I: new cache content 'libc6_2.23-0ubuntu3_amd64.deb' added
I: new cache content 'libgcc-5-dev_5.4.0-6ubuntu1~16.04.9_amd64.deb' added
I: new cache content 'libasan2_5.3.1-14ubuntu2_amd64.deb' added
I: new cache content 'linux-libc-dev_4.4.0-117.141_amd64.deb' added
I: new cache content 'diffutils_1%3a3.3-3_amd64.deb' added
I: new cache content 'debconf_1.5.58ubuntu1_all.deb' added
I: new cache content 'libquadmath0_5.4.0-6ubuntu1~16.04.9_amd64.deb' added
I: new cache content 'libprocps4_2%3a3.3.10-4ubuntu2.3_amd64.deb' added
I: new cache content 'ncurses-bin_6.0+20160213-1ubuntu1_amd64.deb' added
I: new cache content 'cpp-5_5.3.1-14ubuntu2_amd64.deb' added
I: unmounting dev/pts filesystem
I: unmounting run/shm filesystem
I: unmounting proc filesystem
I: creating base tarball [/home/kssim/pbuilder/xenial-base.tgz]
I: cleaning the build env
I: removing directory /var/cache/pbuilder/build/1751 and its subdirectories
```



# Debian Packaging

# debian 디렉토리 생성

- “dh\_make” 명령을 사용해서 debian 디렉토리 생성
- 불필요한 템플릿 파일을 제거
- 주요 파일에 대한 내용 작성

# debian 디렉토리 생성

```
kssim@dev:~/hello$ ls
src
kssim@dev:~/hello$ dh_make --native -p hello_1.0.0 -c mit
Type of package: (single, indep, library, python)
[s/i/l/p]?
Email-Address      : ksub0912@gmail.com
License            : mit
Package Name       : hello
Maintainer Name    : KyeongSeob Sim
Version            : 1.0.0
Package Type       : single
Date               : Sat, 24 Mar 2018 08:17:56 +0000
Are the details correct? [Y/n/q]
Currently there is not top level Makefile. This mayrequire additional tuning
Done. Please edit the files in the debian/ subdirectory now.

kssim@dev:~/hello$ ls
debian  src
```

# debian 디렉토리 생성

```
kssim@dev:~/hello/debian$ ls
README          control           hello.doc-base.EX menu.ex      rules
README.Debian   copyright        init.d.ex       postinst.ex  source
README.source   hello-docs.docs  manpage.1.ex    postrm.ex
changelog       hello.cron.d.ex  manpage.sgml.ex preinst.ex
compat          hello.default.ex manpage.xml.ex  prerm.ex
```

# changelog 작성

- “dch -i” 로 changlog 를 작성
- Launchpad 의 이슈가 있을 경우 “LP: #<number>” 로 작성
- 작성자와 이메일 정보를 확인
- 작성 일자를 확인
  - RFC 5322 형식에 맞춰서 작성

# changelog 작성

```
hello (1.0.0ubuntu1) UNRELEASED; urgency=medium

 *
[- KyeongSeob Sim <ksub0912@gmail.com> Sat, 24 Mar 2018 08:32:32 +0000

hello (1.0.0) unstable; urgency=medium

 * Initial Release.

-- KyeongSeob Sim <ksub0912@gmail.com> Sat, 24 Mar 2018 08:17:56 +0000
```

# changelog 작성

```
hello (1.0.0ubuntu1) xenial; urgency=medium

  * Initial Release.

-- KyeongSeob Sim <ksub0912@gmail.com>  Sat, 24 Mar 2018 08:17:56 +0000
```

# 테스트 빌드

- “debuild -S” 명령을 사용해서, 테스트 빌드를 수행 (소스 코드만 빌드)
- “pbuilder-dist” 명령을 사용해서 빌드 환경 테스트
- “dpkg -l” 옵션으로 테스트로 빌드된 패키지 정보 확인

# 테스트 빌드

```
kssim@dev:~/hello$ debuild -S -d -us -uc
dpkg-buildpackage -rfakeroot -d -us -uc -S
dpkg-buildpackage: source package hello
dpkg-buildpackage: source version 1.0.0ubuntu1
dpkg-buildpackage: source distribution xenial
dpkg-buildpackage: source changed by KyeongSeob Sim <ksub0912@gmail.com>
dpkg-source --before-build hello
fakeroot debian/rules clean
dh clean
  dh_testdir
  dh_auto_clean
  dh_clean
dpkg-source -b hello
dpkg-source: warning: Version number suggests Ubuntu changes, but Maintainer: does not have Ubuntu address
dpkg-source: warning: Version number suggests Ubuntu changes, but there is no XSBC-Original-Maintainer field
dpkg-source: info: using source format '3.0 (native)'
dpkg-source: info: building hello in hello_1.0.0ubuntu1.tar.xz
dpkg-source: info: building hello in hello_1.0.0ubuntu1.dsc
  dpkg-genchanges -S >../hello_1.0.0ubuntu1_source.changes
dpkg-genchanges: including full source code in upload
  dpkg-source --after-build hello
dpkg-buildpackage: source-only upload: Debian-native package
Now running lintian...
W: hello source: out-of-date-standards-version 3.9.6 (current is 3.9.7)
Finished running lintian.
```

# 테스트 빌드

```
kssim@dev:~/hello$ pbuilder-dist xenial build ../hello_1.0.0ubuntu1.dsc
W: /home/kssim/.pbuilderrc does not exist
I: Logging to /home/kssim/pbuilder/xenial_result/hello_1.0.0ubuntu1_amd64.build
I: using fakeroot in build.
I: pbuilder: network access will be disabled during build
I: Current time: Sat Mar 24 08:46:56 UTC 2018
I: pbuilder-time-stamp: 1521881216
I: Building the build Environment
I: extracting base tarball [/home/kssim/pbuilder/xenial-base.tgz]
I: copying local configuration
W: --override-config is not set; not updating apt.conf Read the manpage for details.
I: mounting /proc filesystem
I: mounting /run/shm filesystem
I: mounting /dev/pts filesystem
I: policy-rc.d already exists
I: Obtaining the cached apt archive contents
I: Installing the build-deps
  -> Attempting to satisfy build-dependencies
  -> Creating pbuilder-satisfydepends-dummy package
```

# 테스트 빌드

```
kssim@dev:~/hello$ dpkg -I /home/kssim/pbuilder/xenial_result/hello_1.0.0ubuntu1_amd64.deb
new debian package, version 2.0.
size 2158 bytes: control archive=458 bytes.
  257 bytes,   10 lines      control
  199 bytes,    3 lines      md5sums
Package: hello
Version: 1.0.0ubuntu1
Architecture: amd64
Maintainer: KyeongSeob Sim <ksub0912@gmail.com>
Installed-Size: 10
Section: misc
Priority: optional
Homepage: https://kssim.com
Description: sample package.
  ubuntu packaging workshop sample package.
```

# 테스트 빌드

```
dpkg-deb: building package 'hello' in '../hello_1.0.0ubuntu1_amd64.deb'  
dpkg-genchanges >../hello_1.0.0ubuntu1_amd64.changes  
dpkg-genchanges: including full source code in upload  
dpkg-source --after-build hello-1.0.0ubuntu1  
dpkg-buildpackage: full upload; Debian-native package (full source is included)  
I: copying local configuration  
I: Copying back the cached apt archive contents  
I: unmounting dev/pts filesystem  
I: unmounting run/shm filesystem  
I: unmounting proc filesystem  
I: cleaning the build env  
I: removing directory /var/cache/pbuilder/build/20600 and its subdirectories  
I: Current time: Sat Mar 24 08:47:14 UTC 2018  
I: pbuilder-time-stamp: 1521881234
```

# Debian package 정적 분석

- “lintian” 툴을 사용해서 debian package 를 정적 분석
- 패키지 내부의 버그나 정책 위반을 분석
  - Error 와 Warning 으로 구분
- dsc 파일과 deb 파일을 분석

# Debian package 정적 분석

```
kssim@dev:~$ lintian hello_1.0.0ubuntu1.dsc
W: hello source: out-of-date-standards-version 3.9.6 (current is 3.9.7)
kssim@dev:~$
kssim@dev:~$ lintian hello_1.0.0ubuntu1_amd64.deb
E: hello: helper-templates-in-copyright
W: hello: copyright-has-url-from-dh_make-boilerplate
E: hello: copyright-contains-dh_make-todo-boilerplate
W: hello: readme-debian-contains-debmake-template
```

# Debian package 정적 분석

```
kssim@dev:~$ lintian --info hello_1.0.0ubuntu1_amd64.deb
E: hello: helper-templates-in-copyright
N:
N:   The /usr/share/doc/<pkg>/copyright file still contains template markers
N:   from a packaging helper. Please fill in the actual license, upstream
N:   copyright holders, and download information about the package and remove
N:   any remaining templates generated by the packaging helper.
N:
N:   Severity: important, Certainty: certain
N:
N:   Check: copyright-file, Type: binary
N:
W: hello: copyright-has-url-from-dh_make-boilerplate
N:
N:   There is "url://example.com" in your copyright file. This was most
N:   likely a remnant from the dh_make template.
N:
N:   Make sure you include the real location where you obtained the upstream
N:   sources (if any).
N:
N:   Refer to Debian Policy Manual section 12.5 (Copyright information) for
N:   details.
N:
N:   Severity: normal, Certainty: certain
N:
N:   Check: copyright-file, Type: binary
```

# 빌드

- “debuild” 옵션을 사용해서 빌드
  - -F : Full Packaging
  - -S : Source Code Packaging
  - -b : Binary Pacakging

빌드

```
kssim@dev:~/hello$ debuild -i
dpkg-buildpackage -rfakeroot -D -us -uc -i
dpkg-buildpackage: source package hello
dpkg-buildpackage: source version 1.0.0ubuntu1
dpkg-buildpackage: source distribution xenial
dpkg-buildpackage: source changed by KyeongSeob Sim <cksub0912@gmail.com>
dpkg-source -i --before-build hello
dpkg-buildpackage: host architecture amd64
fakeroot debian/rules clean
dh clean
dh_testdir
dh_auto_clean
dh_clean
dpkg-source -i -b hello
```



```
Finished running lintian.  
Now signing changes and any dsc files...  
signfile hello_1.0.0ubuntu1.dsc KyeongSeob Sim <ksub0912@gmail.com>  
  
You need a passphrase to unlock the secret key for  
user: "KyeongSeob Sim <ksub0912@gmail.com>"  
2048-bit RSA key, ID 9FD41E43, created 2018-03-24  
  
gpg: gpg-agent is not available in this session  
  
signfile hello_1.0.0ubuntu1_amd64.changes KyeongSeob Sim <ksub0912@gmail.com>  
  
You need a passphrase to unlock the secret key for  
user: "KyeongSeob Sim <ksub0912@gmail.com>"  
2048-bit RSA key, ID 9FD41E43, created 2018-03-24  
  
gpg: gpg-agent is not available in this session  
  
Successfully signed dsc and changes files
```



# QnA

# 실습

# 실습 과제

1. 빌드 환경을 구축한다.
2. Sample Code 를 다운받는다.
3. Debian directory 를 만들고, 설정한다.
4. Debian 패키지를 만든다.
5. 패키지를 검증한다.