블록체인 이해와 구조(실습)

실습자료

1. 블록체인 프로그래밍
2. 암호학 실습
3. 네트워크 실습

Development Environment

- 개발에 필요한 유틸리티
  - Cmder : 도스 커맨트 프로그램
  - JAVA : 자바 프로그램
  - Maven: 자바 라이브러리(jar) 관리 및 빌드 환경 제공

Development Environment

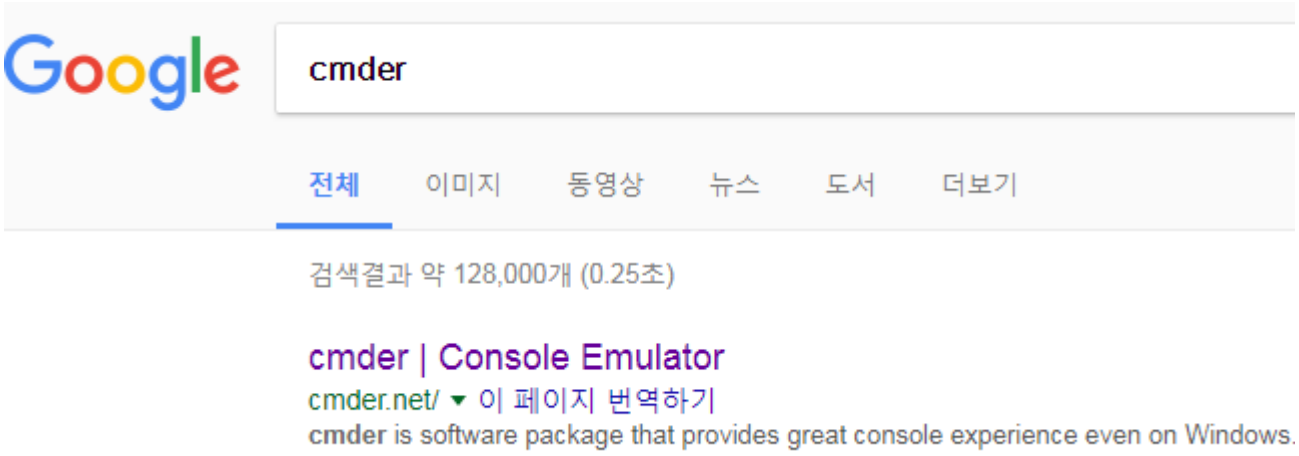
- Maven
  - Java 라이브러리 관리 도구
  - 프로젝트를 진행하게 되면 단순히 자신이 작성한 코드만으로 개발하는 것이 아니라 많은 라이브러리들을 활용해서 개발을 하게 됨
  - 이 때 사용되는 라이브러리들의 수가 수십 개가 훌쩍 넘어버리는 일이 발생해 이 많은 라이브러리들을 관리하는 것이 힘들어지는 경우가 종종 발생하곤 함
  - Maven은 이러한 문제를 해결해 줄 수 있는 도구
  - Maven은 내가 사용할 라이브러리 뿐만 아니라 해당 라이브러리가 작동하는 데에 필요한 다른 라이브러리들까지 관리하여 네트워크를 통해서 자동으로 다운받아 줌

Development Environment

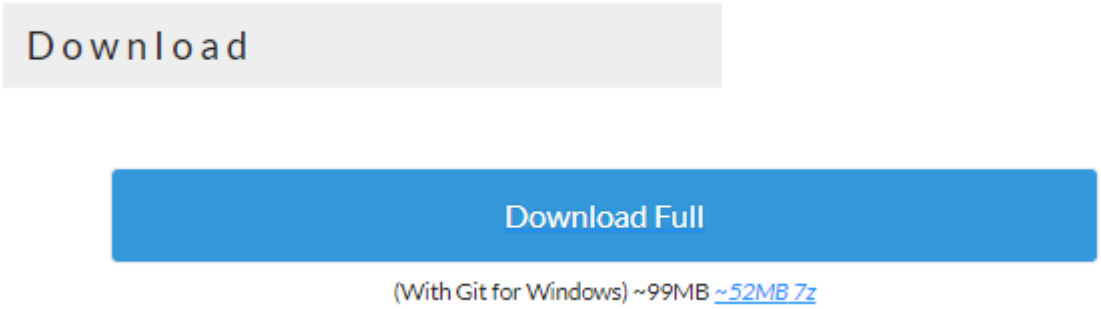
- Maven
  - POM(Project Object Model)
    - Maven 설정 파일
    - 프로젝트 정보 : 프로젝트의 이름, 개발자 목록, 라이센스 등
    - 빌드 설정 : 소스, 리소스, 라이프 사이클별 실행한 플러그인(goal)등 빌드와 관련된 설정
    - 빌드 환경 : 사용자 환경 별로 달라질 수 있는 프로파일 정보
    - POM 연관 정보 : 의존 프로젝트(모듈), 상위 프로젝트, 포함하고 있는 하위 모듈 등

Development Environment

Cmder 설치

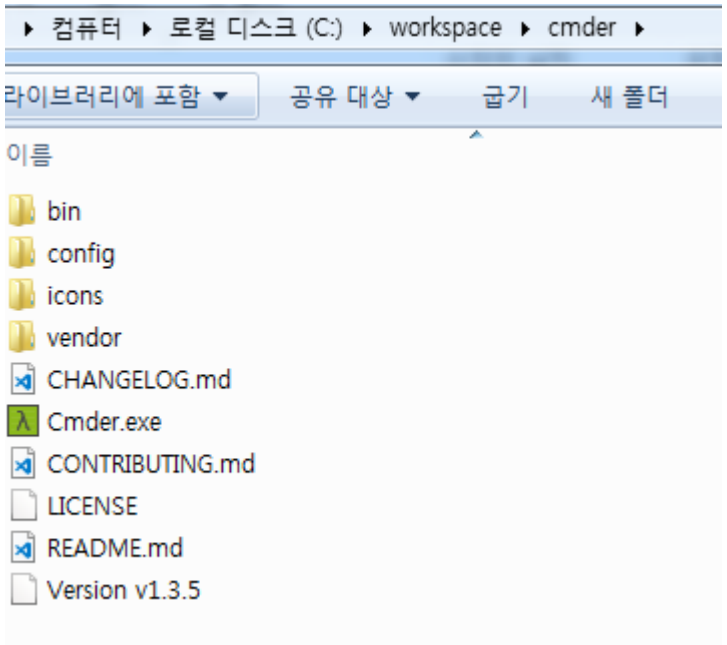


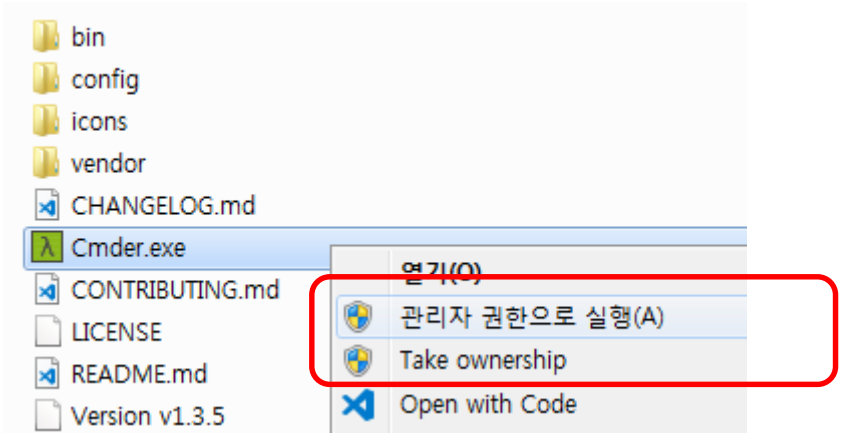
Download > Download Full 선택



Development Environment

- CD₩utils₩cmder₩cmder.zip
- C:₩workspace 폴더 생성 후 압축 풀기
- 관리자 권한으로 Cmder 실행





Development Environment

- Eclipse Setup
  - JAVA JDK 설치
    - CD₩utils₩android₩32 or 64₩jdk-8u101-windows-x64.exe 설치
    - or
    - **http://www.oracle.com/technetwork/java/javase/downloads/index.html**
    - Java Platform(JDK) 선택

Development Environment

- Eclipse Setup
  - JAVA JDK 설치
    - Accept License Agreement 선택
    - 해당 버전 선택

Java SE Development Kit 8u101

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

○ Accept License Agreement ◉ Decline License Agreement

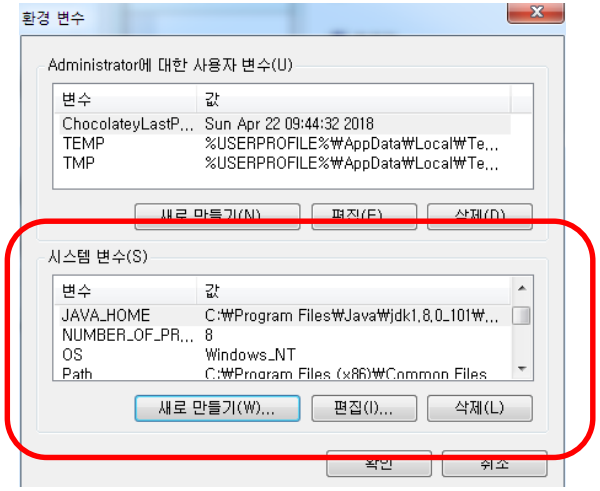
Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.77 MB	jdk-8u101-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.72 MB	jdk-8u101-linux-arm64-vfp-hflt.tar.gz
Linux x86	160.28 MB	jdk-8u101-linux-i586.rpm
Linux x86	174.96 MB	jdk-8u101-linux-i586.tar.gz
Linux x64	158.27 MB	jdk-8u101-linux-x64.rpm
Linux x64	172.95 MB	jdk-8u101-linux-x64.tar.gz
Mac OS X	227.36 MB	jdk-8u101-macosx-x64.dmg
Solaris SPARC 64-bit	139.66 MB	jdk-8u101-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	98.96 MB	jdk-8u101-solaris-sparcv9.tar.gz
Solaris x64	140.33 MB	jdk-8u101-solaris-x64.tar.Z
Solaris x64	96.78 MB	jdk-8u101-solaris-x64.tar.gz
Windows x86	188.32 MB	jdk-8u101-windows-i586.exe
Windows x64	193.68 MB	jdk-8u101-windows-x64.exe

Development Environment

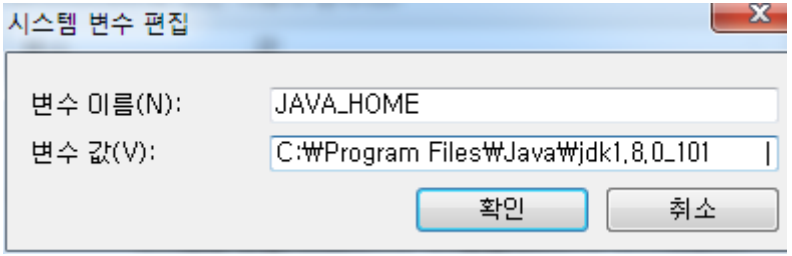
Eclipse Setup

JAVA JDK 설치

- 환경변수 등록
- 내 컴퓨터 > 마우스 오른쪽 버튼 > 속성 > 고급 시스템 설정 > 고급 > 환경 변수
- 시스템 변수 > 새로 만들기 > 변수 이름: **JAVA_HOME**, 변수 값: {your java path 추가}

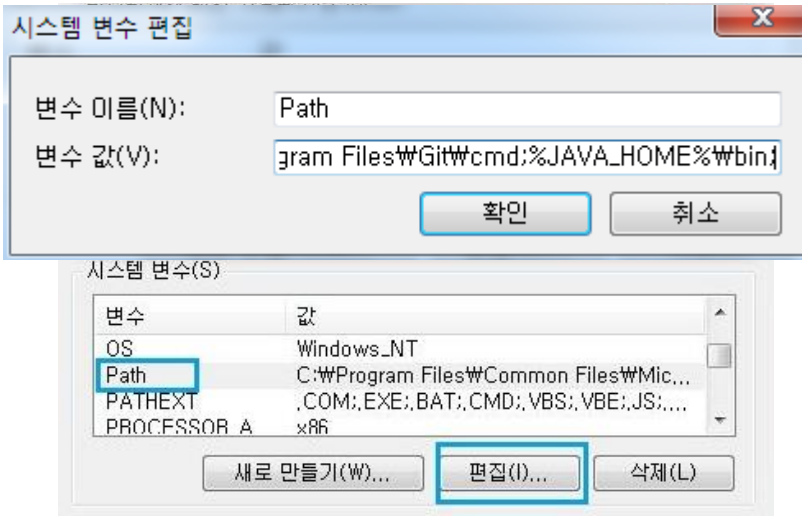


C:₩Program Files₩Java₩jdk1.8.0_101



Development Environment

- Eclipse Setup
  - JAVA JDK 설치
    - 환경변수 등록
    - 시스템 변수 > **PATH** > 편집 > ";%JAVA_HOME%₩bin; " 패스추가



;%JAVA_HOME%₩bin;

Development Environment

- Maven
  - CD₩utils₩maven₩apache-maven-3.5.4.zip 파일
  - or
  - Download
  - http://maven.apache.org/download.cgi

Files

Maven is distributed in several formats for your convenience. Simply pick a ready-made binary distribution archive and follow the installation instructions. Use a source archive if you intend to build Maven yourself.

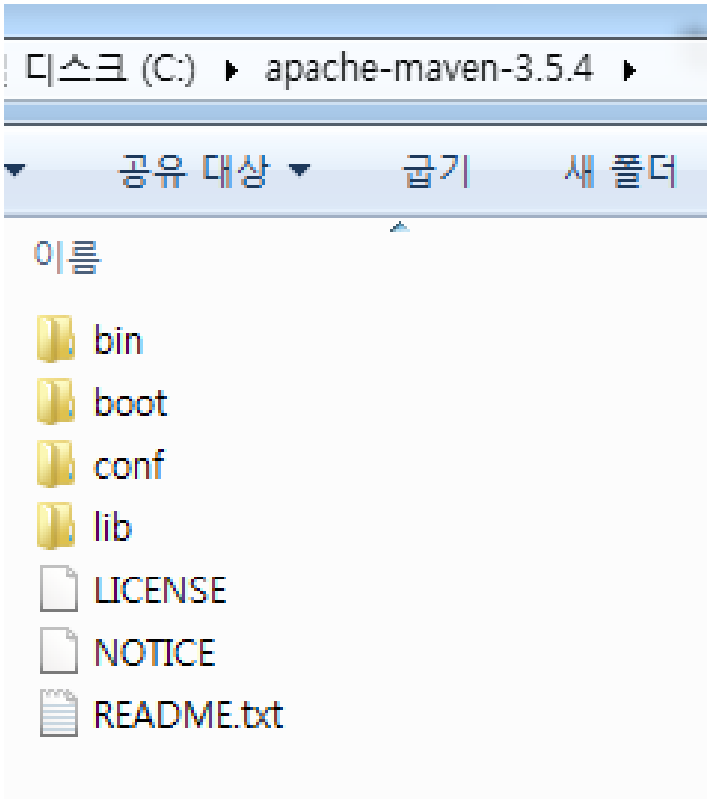
In order to guard against corrupted downloads/installations, it is highly recommended to verify the signature of the release bundles against the public KEYS used by the Apache Maven developers.

	Link	Checksums	Signature
Binary tar.gz archive	apache-maven-3.5.4-bin.tar.gz	apache-maven-3.5.4-bin.tar.gz.sha1	apache-maven-3.5.4-bin.tar.gz.asc
Binary zip archive	apache-maven-3.5.4-bin.zip	apache-maven-3.5.4-bin.zip.sha1	apache-maven-3.5.4-bin.zip.asc
Source tar.gz archive	apache-maven-3.5.4-src.tar.gz	apache-maven-3.5.4-src.tar.gz.sha1	apache-maven-3.5.4-src.tar.gz.asc
Source zip archive	apache-maven-3.5.4-src.zip	apache-maven-3.5.4-src.zip.sha1	apache-maven-3.5.4-src.zip.asc

Development Environment

Maven

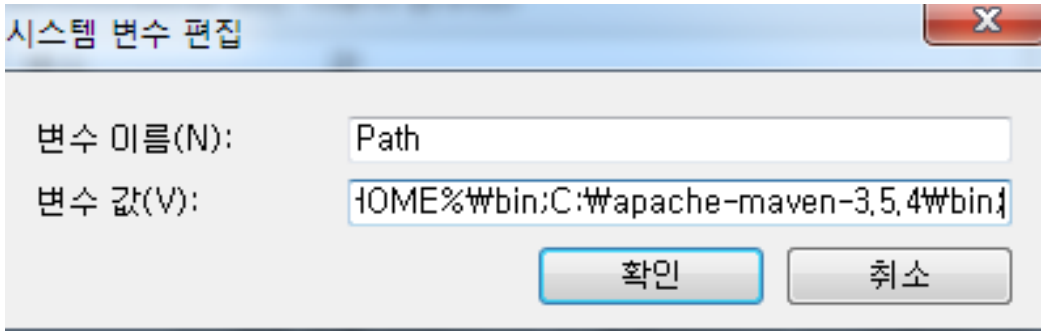
- CD₩utils₩maven₩apache-maven-3.5.4.zip 파일을 C:₩apache-maven-3.5.4 폴더로 압축 파일 풀기



Development Environment

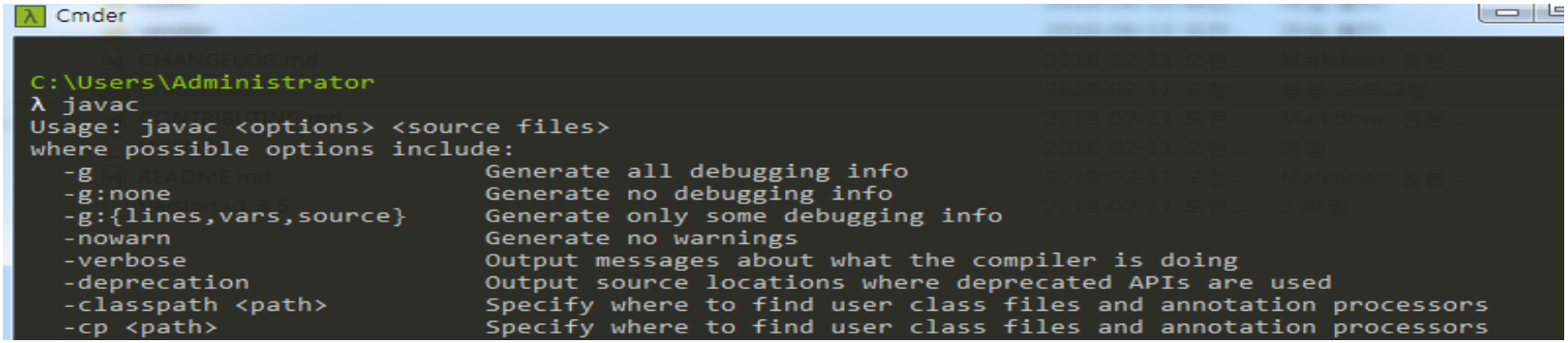
Maven

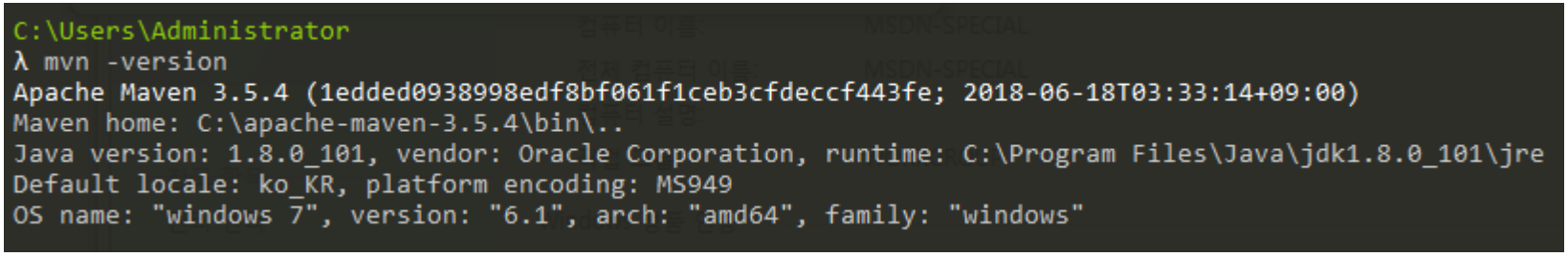
- 내 컴퓨터 > 마우스 오른쪽 버튼 > 속성 > 고급 시스템 설정 > 고급 > 환경 변수
- 시스템 변수 > **PATH** > 편집 > "C:₩apache-maven-3.5.4₩bin;" 패스추가



Development Environment

- 프로그램 버전 확인
  - cmder 실행
  - cmd> javac
  - cmd> mvn -version



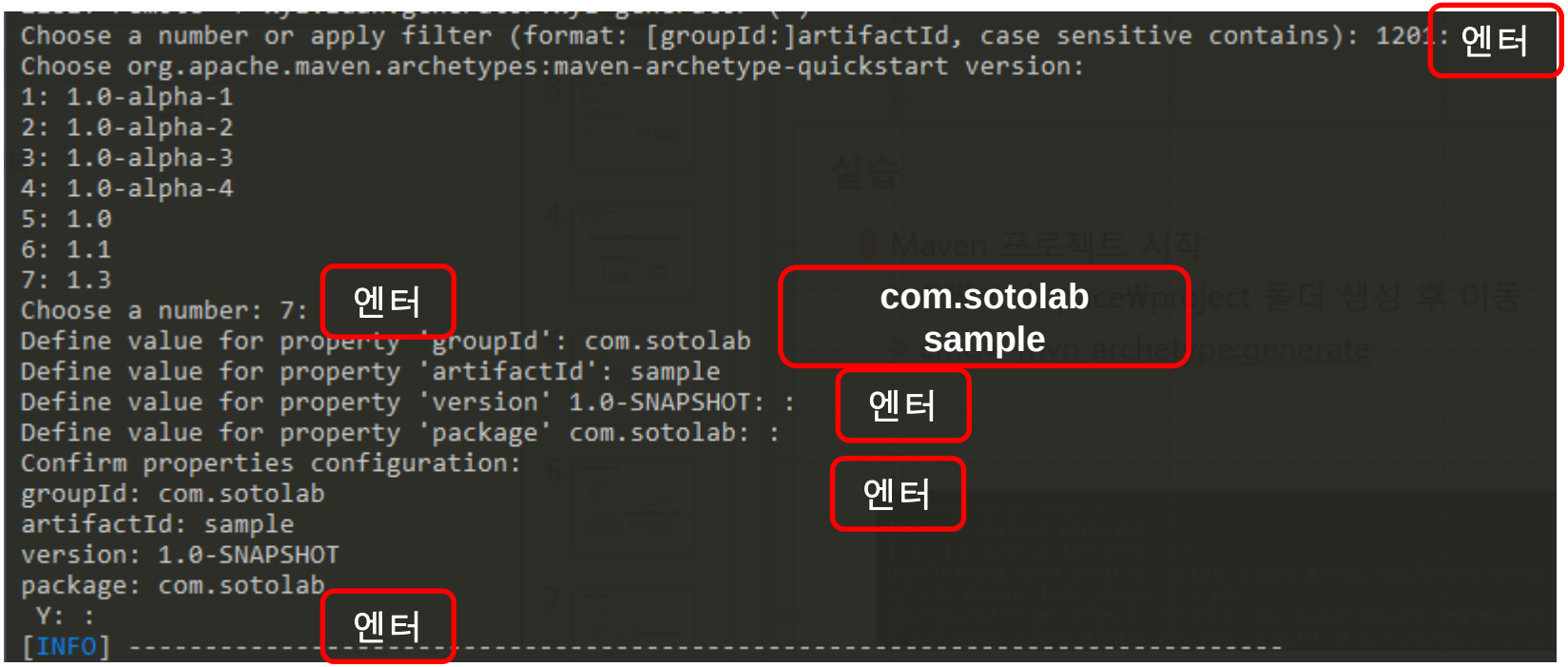


실습

- Maven 프로젝트 시작
  - Workspace 폴더 이동하고, project 폴더 생성 후 이동
  - cmd> cd c:₩workspace
  - cmd> mkdir project
  - cmd> cd project

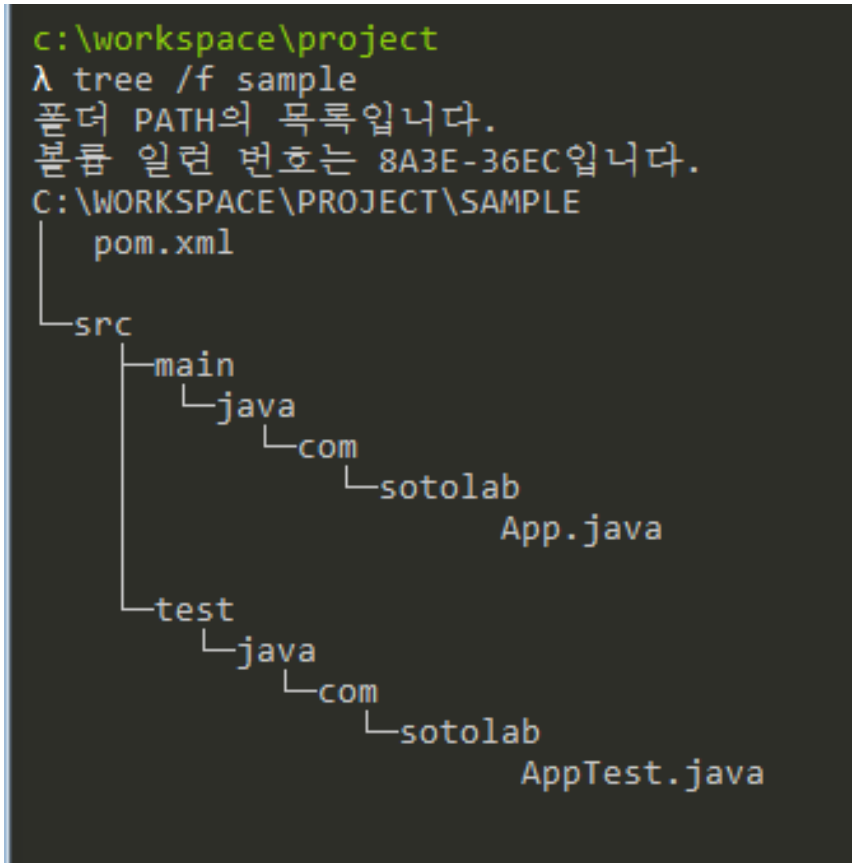
실습

- Maven 프로젝트 시작
  - 프로젝트 생성
  - cmd> mvn archetype:generate



실습

- Maven 프로젝트 시작
  - 폴더 및 파일 확인
  - cmd> tree /a sample



실습

- Maven 프로젝트 시작
  - Sample 폴더 이동 후 package 실행
  - cmd> cd sample
  - cmd> mvn package

```
c:\workspace\project\sample
λ mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------< com.sotolab:sample >------------------------
[INFO] Building sample 1.0-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- maven-resources-plugin:3.0.2:resources (default-resources) @ sample ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory c:\workspace\project\sample\src\main\resources
```

실습

- Maven 프로젝트 시작
  - 프로그램 실행
  - cmd> java –cp target₩sample-1.0-SNAPSHOT.jar com.sotolab.App

```
c:\workspace\project\sample
λ mjava -cp target\sample-1.0-SNAPSHOT.jar com.sotolab.App
Hello World!
```

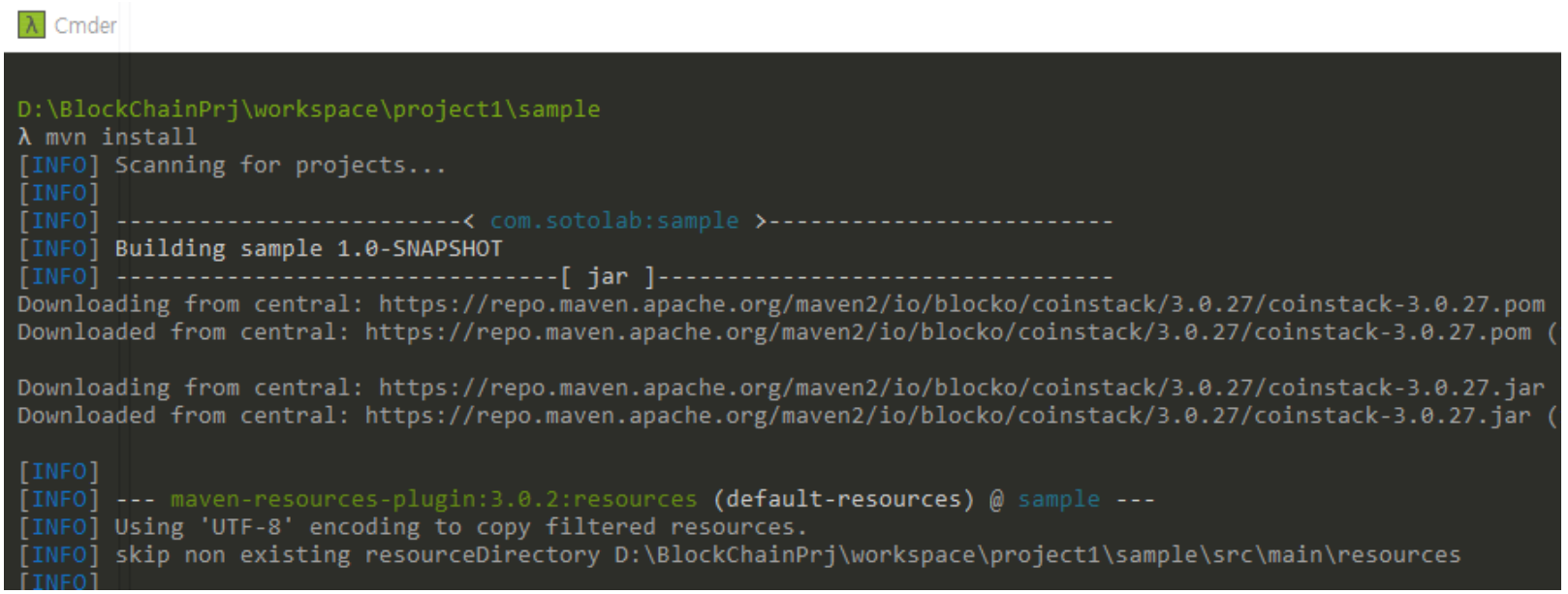
실습

CoinStack 라이브러리 설치

- pom.xml 파일 수정

```
<dependency>
    <groupId>io.blocko</groupId>
    <artifactId>coinstack</artifactId>
    <version>3.0.27</version>
</dependency>
```

- mvn install (라이브러리 다운로드)



CoinStack 라이브러리 설치

- CoinStack 라이브러리 다운로드 확인
  - C:₩Users₩[사용자이름]₩.m2₩repository₩io₩blocko₩coinstack₩3.0.27

```
C:\Users\in4ki\.m2\repository\io\blocko\coinstack\3.0.27
λ dir
 C 드라이브의 볼륨에는 이름이 없습니다.
 볼륨 일련 번호: 2698-6AD3

 C:\Users\in4ki\.m2\repository\io\blocko\coinstack\3.0.27 디렉터리

2018-09-05  오후 02:24    <DIR>          .
2018-09-05  오후 02:24    <DIR>          ..
2018-09-05  오후 02:24        11,618,880 coinstack-3.0.27.jar
2018-09-05  오후 02:24                40 coinstack-3.0.27.jar.sha1
2018-09-05  오후 02:24            11,699 coinstack-3.0.27.pom
2018-09-05  오후 02:24                40 coinstack-3.0.27.pom.sha1
2018-09-05  오후 02:24               204 _remote.repositories
               5개 파일          11,630,863 바이트
               2개 디렉터리  51,098,730,496 바이트 남음
```

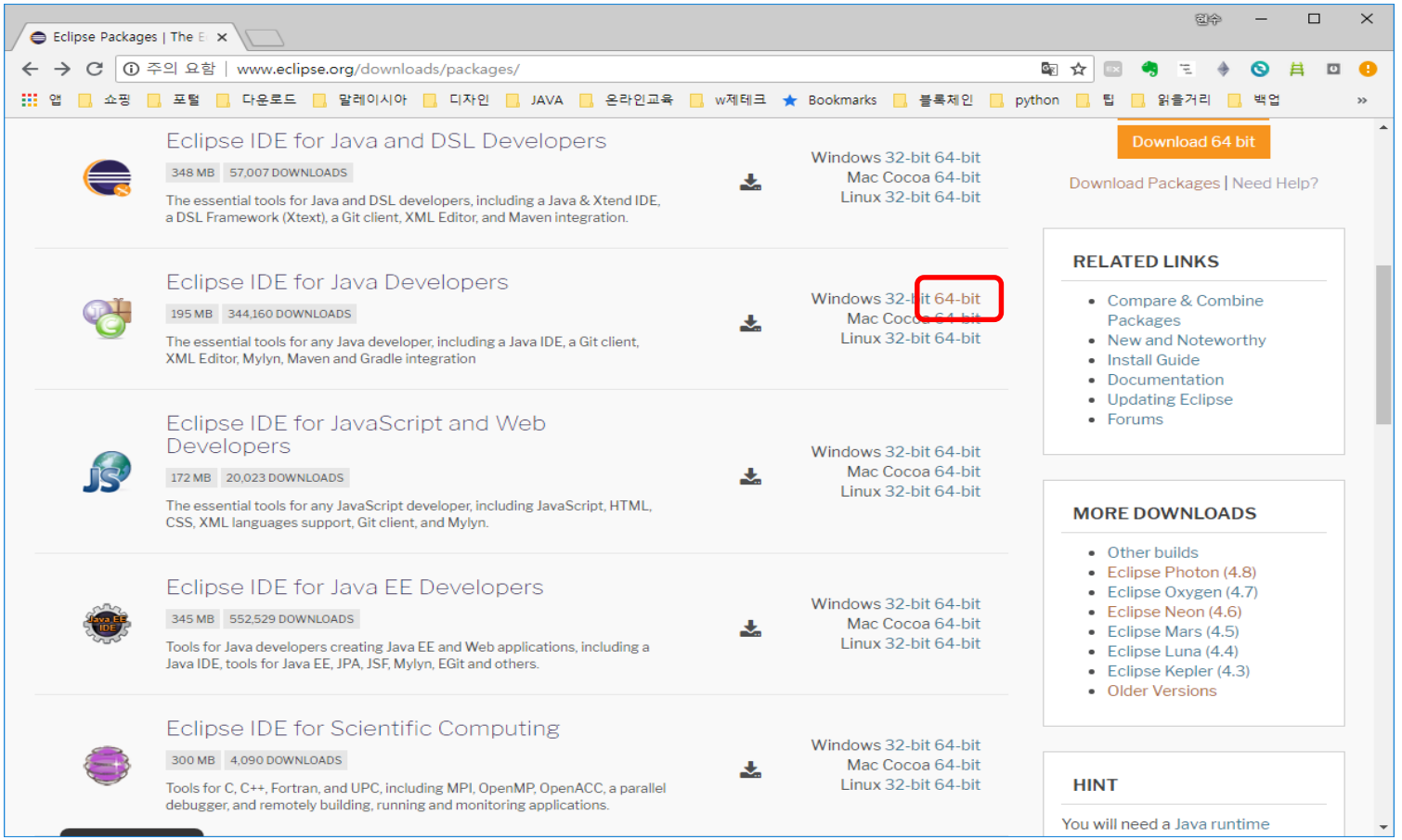
부록

- Install
  - Java JDK
  - Eclipse
  - Maven
  - Coinstack SDK

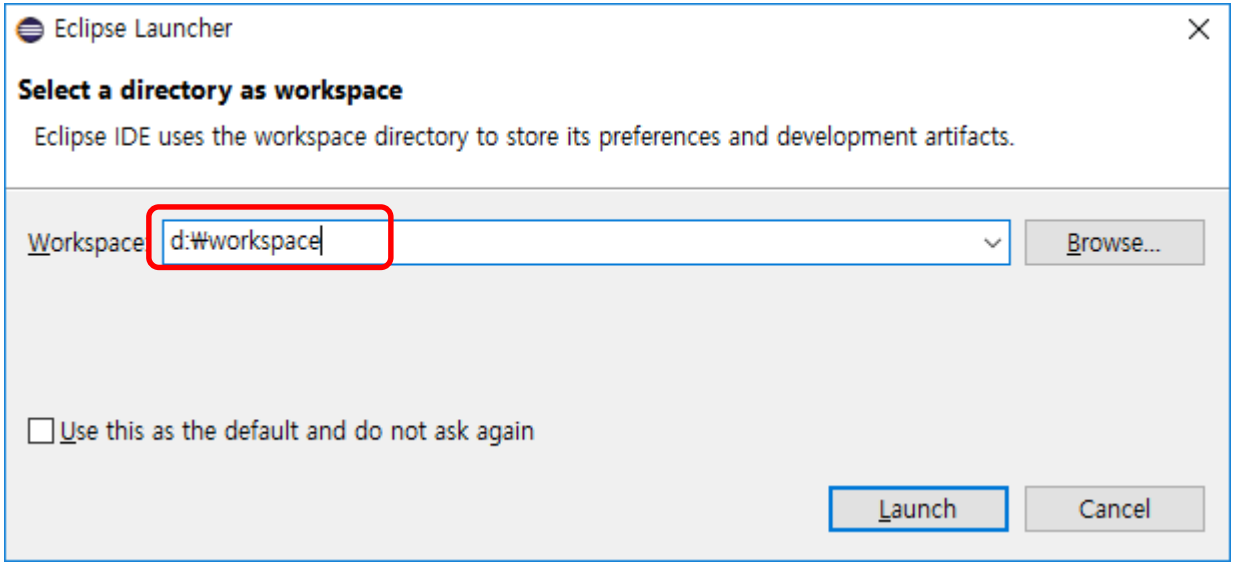
이클립스 Maven 개발환경 구축

이클립스 다운로드

http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/photon/R/eclipse-java-photon-R-win32-x86_64.zip



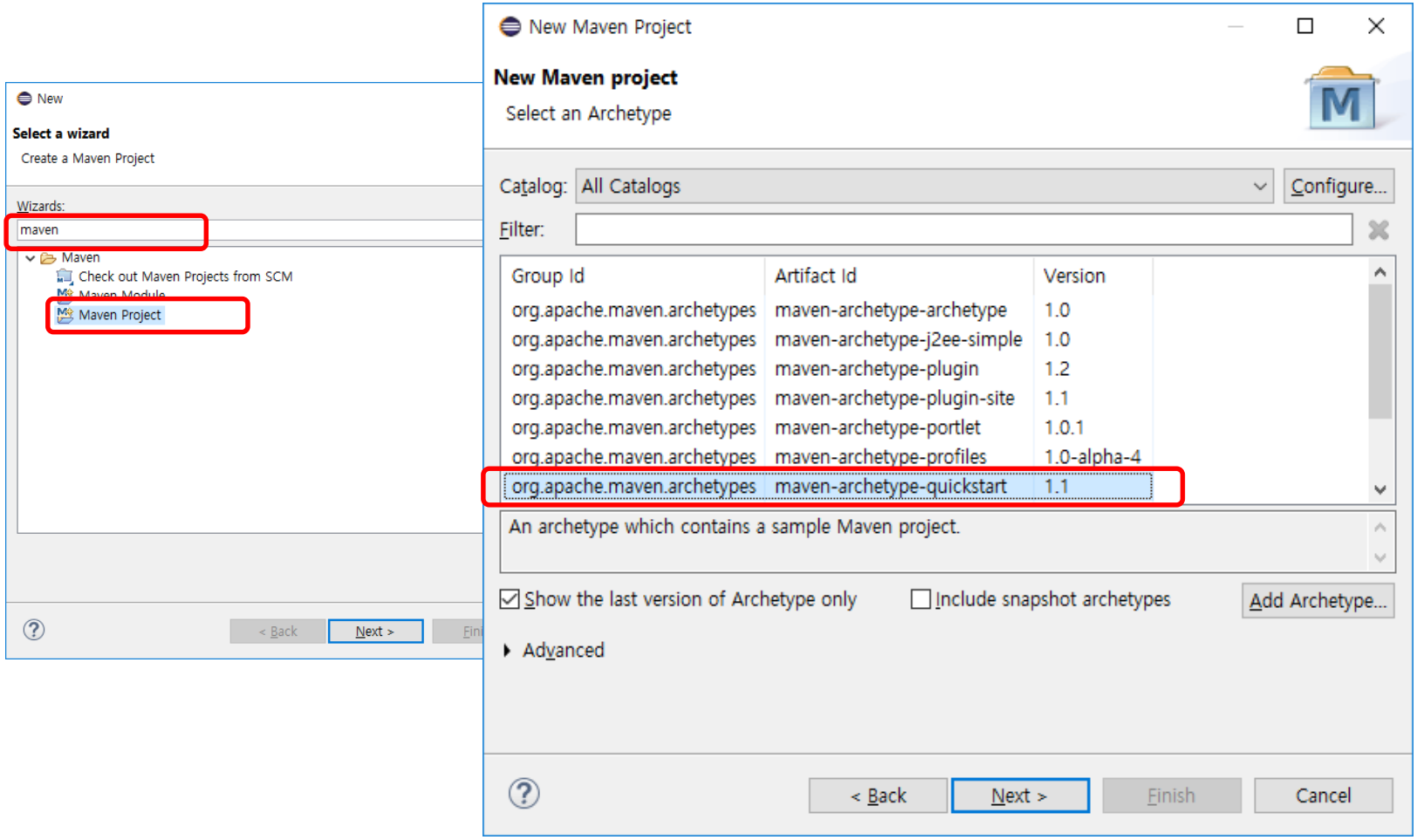
- 압축 풀고 실행
  - c:₩eclipse 폴더에 압축 풀고 실행



새 프로젝트 생성

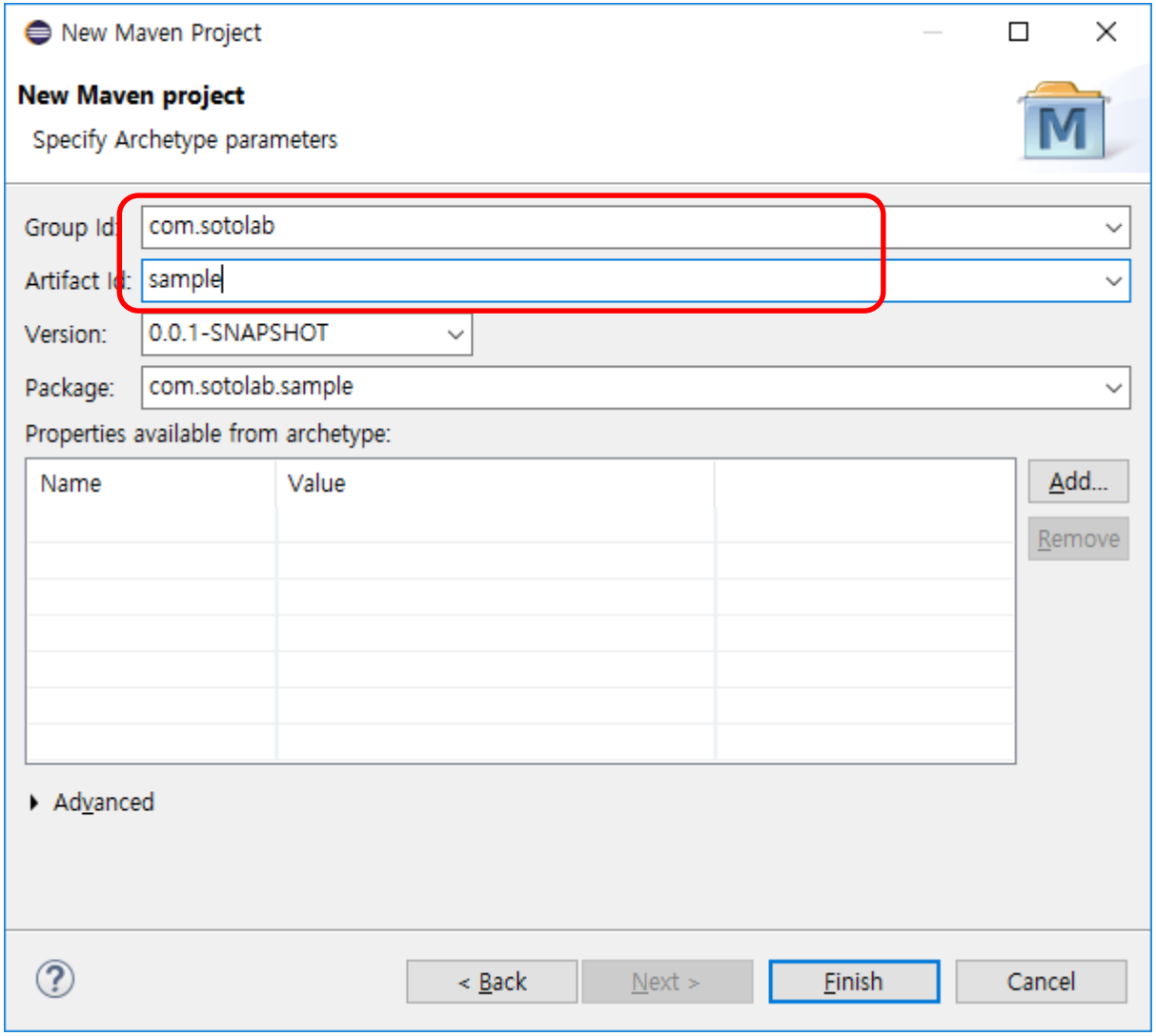
Maven Project 생성

- Ctrl + N => maven 입력 => Maven Project 선택



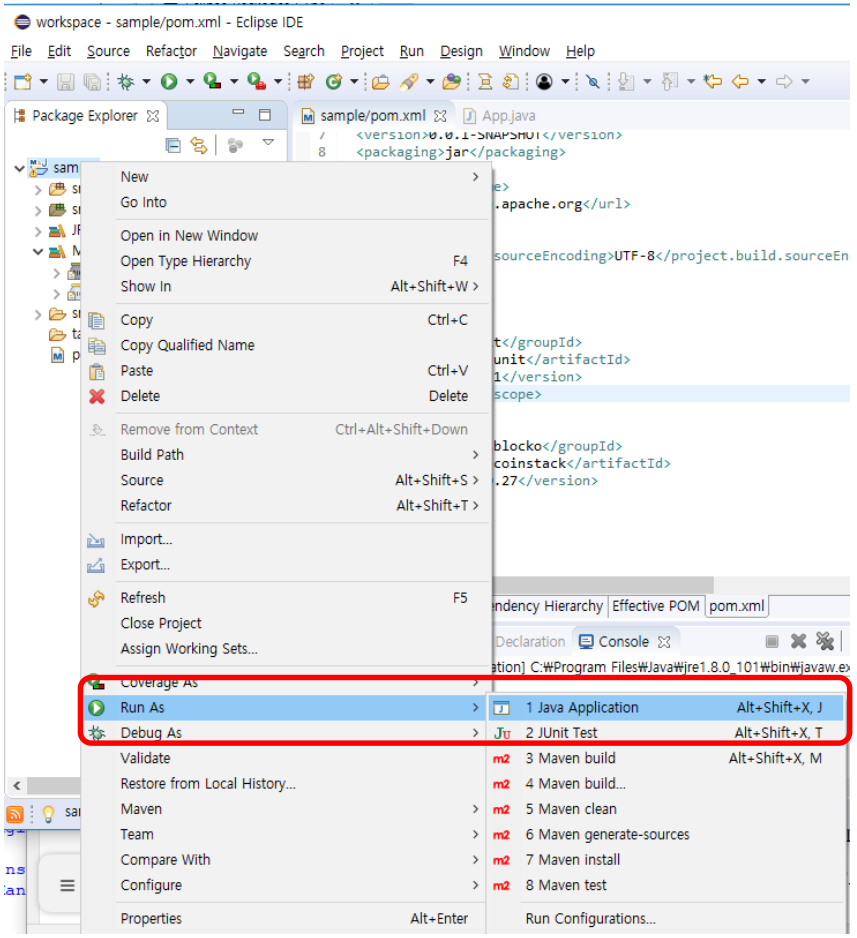
패키지명 입력

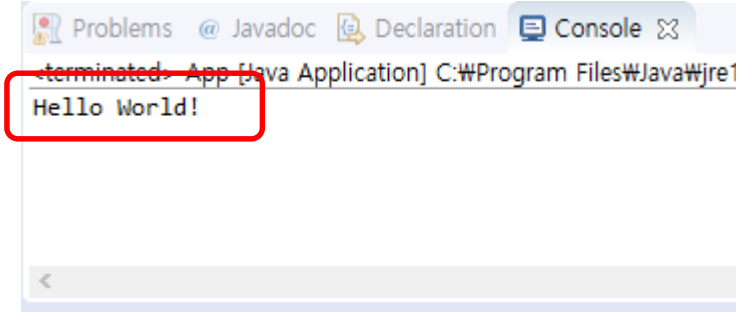
- Group Id와 Artifact Id 입력



샘플 코드 실행

- App.java 컴파일 및 실행
  - 오른쪽 마우스 -> Run As -> Java Application 선택

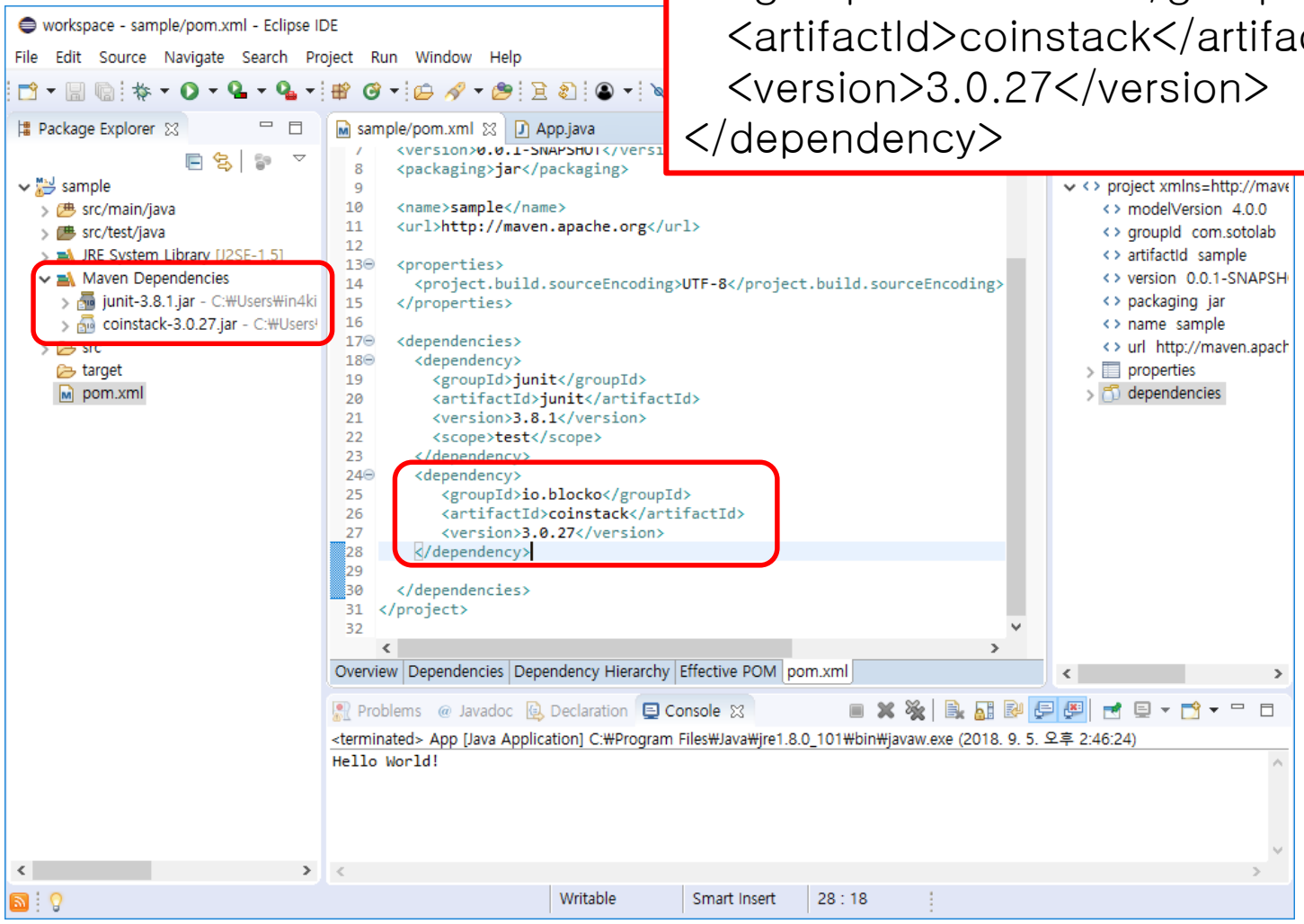




CoinStack 라이브러리 추가

- Pom.xml 파일 수정

```
<dependency>
   <groupId>io.blocko</groupId>
   <artifactId>coinstack</artifactId>
   <version>3.0.27</version>
</dependency>
```

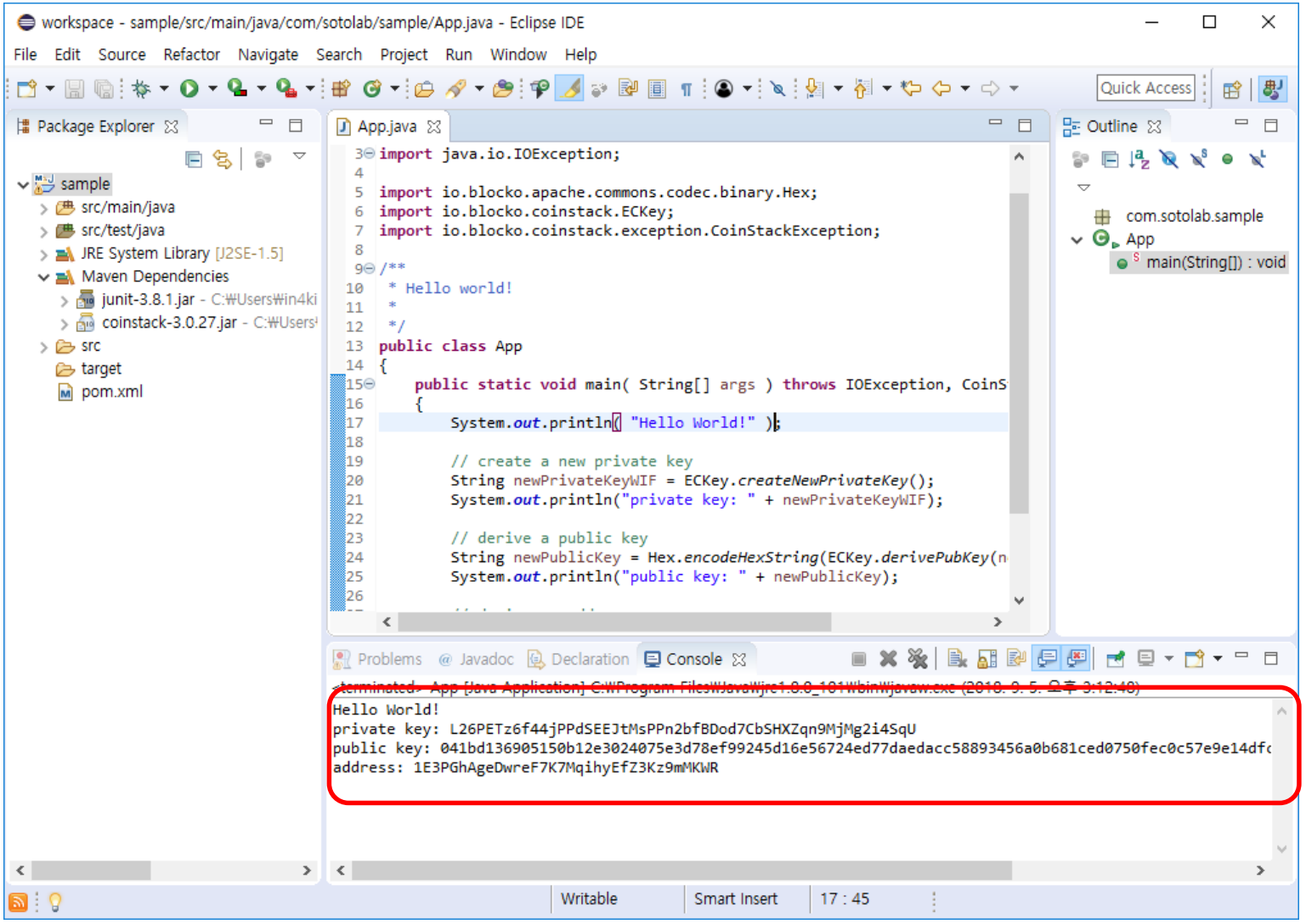


CoinStack Test 코드 입력

Private Key, PublicKey, Wallet Address 생성 실습

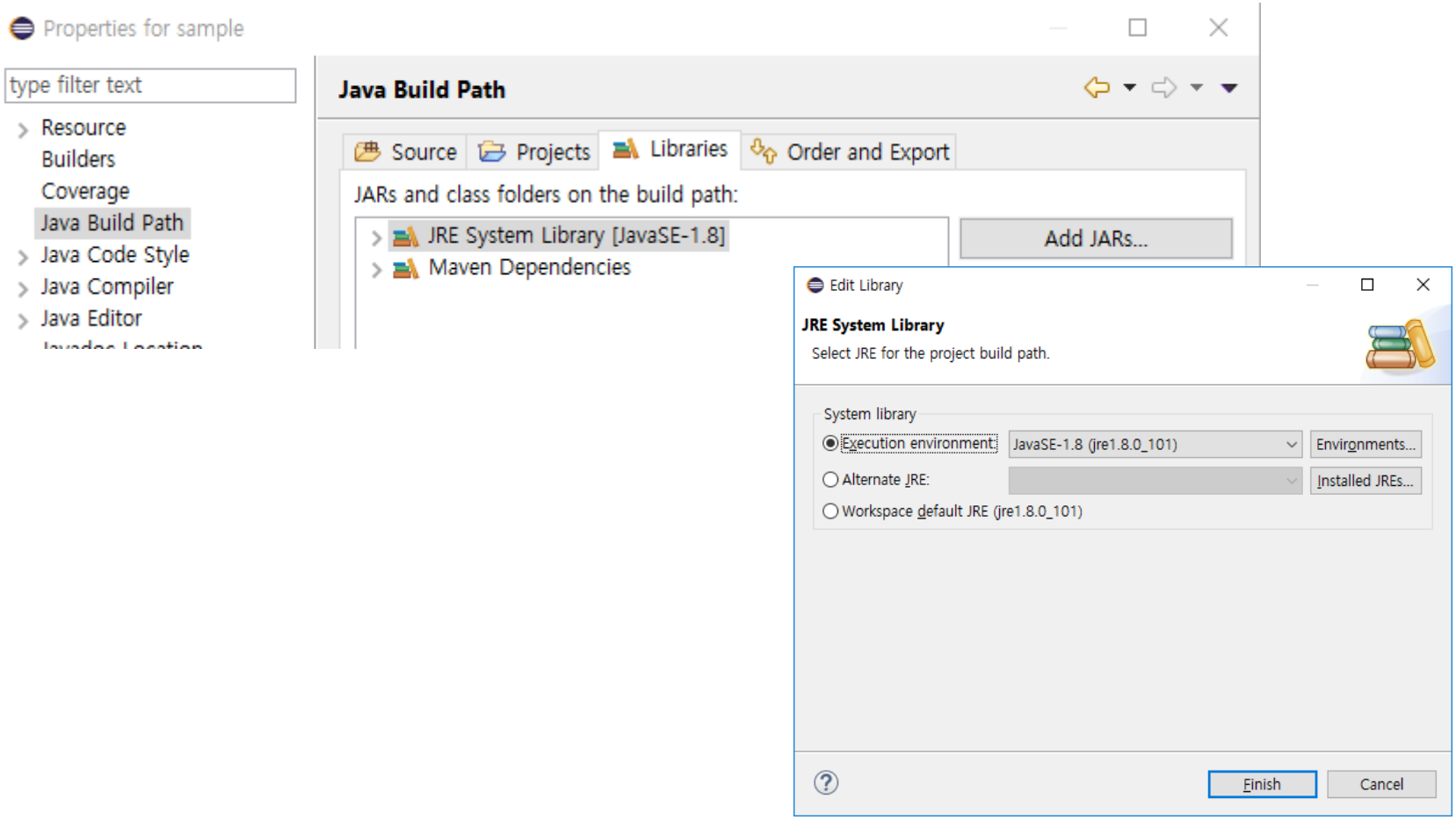
```
import io.blocko.apache.commons.codec.binary.Hex;
import io.blocko.coinstack.ECKey;
import io.blocko.coinstack.exception.CoinStackException;

public class App
{
    public static void main( String[] args ) throws IOException, CoinStackException
    {
        System.out.println( "Hello World!" );

        // create a new private key
        String newPrivateKeyWIF = ECKey.createNewPrivateKey();
        System.out.println("private key: " + newPrivateKeyWIF);

        // derive a public key
        String newPublicKey = Hex.encodeHexString(ECKey.derivePubKey(newPrivateKeyWIF, false));
        System.out.println("public key: " + newPublicKey);

        // derive an address
        String your_wallet_address = ECKey.deriveAddress(newPrivateKeyWIF);
        System.out.println("address: " + your_wallet_address);
    }
}
```

CoinStack 동작 확인



JRE 버전 확인

- 프로젝트 -> 오른쪽 마우스 클릭 -> Property -> Java Build Path -> Libraries



부록

- Private Key 생성
- Public Key 생성
- Wallet address 생성
- Block Chain State 확인
  - Main net : Block Height, Block Hash
  - Test net : Block Height, Block Hash
- Get Balance 확인
  - My wallet address
  - My BitCoin

부록

App.java

```
package com.test.sample;

import java.io.IOException;
import java.security.PublicKey;
import java.text.SimpleDateFormat;
import java.util.Date;
import io.blocko.apache.commons.codec.binary.Hex;
import io.blocko.coinstack.AbstractEndpoint;
import io.blocko.coinstack.CoinStackClient;
import io.blocko.coinstack.ECKey;
import io.blocko.coinstack.Endpoint;
import io.blocko.coinstack.exception.CoinStackException;
import io.blocko.coinstack.model.BlockchainStatus;
import io.blocko.coinstack.model.CredentialsProvider;
           ….
```

부록

App.java

```
public class App
{
public String newPrivateKeyWIF = "";
public String newPublicKey = "";
public String your_wallet_address = "";

public void createKey() throws IOException, CoinStackException {
     // create a new private key
     newPrivateKeyWIF = ECKey.createNewPrivateKey();
     System.out.println("private key: " + newPrivateKeyWIF);

     // derive a public key
     newPublicKey = Hex.encodeHexString(ECKey.derivePubKey(newPrivateKeyWIF));
     System.out.println("public key: " + newPublicKey);

     // derive an address
     your_wallet_address = ECKey.deriveAddress(newPrivateKeyWIF);
     System.out.println("address: " + your_wallet_address);
     System.out.println(" ");
}
```

부록

App.java

```
    public void BlockchainStatus(CoinStackClient Client) throws IOException,
CoinStackException {
      BlockchainStatus testStatus = Client.getBlockchainStatus();
      System.out.println("best Height: " + testStatus.getBestHeight());
      System.out.println("best BlockHash: " + testStatus.getBestBlockHash());
      System.out.println(" ");
    }


    public void getBalance(CoinStackClient Client) throws IOException,
CoinStackException {
      // get a remaining balance
      System.out.println("My wallet address: " + your_wallet_address);
      long balance = Client.getBalance(your_wallet_address);
      //System.out.println("balance: " + balance);
      double result  = balance /10000000;
      System.out.println("My BitCoin: " + result);
    }
```

부록

App.java

```
 public static void main( String[] args ) throws IOException, CoinStackException
    {
        System.out.println( "Hello World!" );

   CoinStackClient mainnetClient = new CoinStackClient(new CredentialsProvider() {
  @Override
  public String getAccessKey() {
    return "c7dbfacbdf1510889b38c01b8440b1";
  }
  @Override
  public String getSecretKey() {
    return "10e88e9904f29c98356fd2d12b26de";
  }
}, Endpoint.MAINNET);

CoinStackClient testnetClient = new CoinStackClient(new CredentialsProvider() {
@Override
public String getAccessKey() {
return "0";
}
```

부록

App.java

```
  @Override
  public String getSecretKey() {
    return "0";
  }
}, new AbstractEndpoint() {
   public boolean mainnet() {
       return true;
   }

   public PublicKey getPublicKey() {
        return null;
   }

   public String endpoint() {
        return "http://testchain.blocko.io";
   }
 });
```

부록

App.java

```
        // Create a instance
         App myTest = new App();

         Date dt = new Date();
         SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd, hh:mm:ss a");

         //create keys - newPrivateKeyWIF, newPublicKey, your_wallet_address
         System.out.println(" ");
         System.out.println(" == Create newPrivateKeyWIF, newPublicKey,
your_wallet_address ==");
         myTest.createKey();

         System.out.println(" == Main Net BlockchainStatus == " +
sdf.format(dt).toString() );
         myTest.BlockchainStatus(mainnetClient);
```

부록

App.java

```
        System.out.println(" == Test Net BlockchainStatus == " +
sdf.format(dt).toString() );
        myTest.BlockchainStatus(testnetClient);

        // Get a remaining balance
        System.out.println(" == Get Balance == " + sdf.format(dt).toString() );
        myTest.getBalance(testnetClient);

        mainnetClient.close();
        testnetClient.close();
    }
}
```

Run > Run

```
Hello World!

 == Create newPrivateKeyWIF, newPublicKey, your_wallet_address ==
private key: KwJ1x2QWsYuWHRQhZAbr4Yk12XqtCS24HtD4Xc7DjpWTMNfio5LA
public key: 02d3bd085654c2fa6226e6e8b8865fd57566717bb35e3a6cac96f2d15e50d4de75
address: 1DSsHJnEDii88pK7hSDXL7GcvQY27Ef7N6

 == Main Net BlockchainStatus == 2017-03-18, 02:24:30 오후
best Height: 457780
best BlockHash: 000000000000000011500596cd3d4faf8e8cc18f6e0db8ed073aada7c06fb8c

 == Test Net BlockchainStatus == 2017-03-18, 02:24:30 오후
best Height: 235219
best BlockHash: 00000ef81505fbd9d1001f4621ab4eeb1fc49840b8c2621b9f445a76c3e08153

 == Get Balance == 2017-03-18, 02:24:30 오후
My wallet address: 1DSsHJnEDii88pK7hSDXL7GcvQY27Ef7N6
My BitCoin: 0.0
```

실습 – QR code

- QRcode programming
  - CD\Day01\qrcode 폴더를 c:\workspase 폴더에 복사
  - $ cd c:\workspace\qrcode

실습 - QR code

$ QRcode.java

```
import java.io.File;
…

public class QRCode {

 public static void main(String[] args) throws WriterException,
IOException, NotFoundException {
        String qrCodeData = "Hello World!";
        String filePath = "QRCode.png";
        String charset = "UTF-8"; // or "ISO-8859-1"
        Map<EncodeHintType, ErrorCorrectionLevel> hintMap = new
HashMap<EncodeHintType, ErrorCorrectionLevel>();
        hintMap.put(EncodeHintType.ERROR_CORRECTION,
ErrorCorrectionLevel.L);

        createQRCode(qrCodeData, filePath, charset, hintMap, 200, 200);
        System.out.println("QR Code image created successfully!");

        System.out.println("Data read from QR Code: "
                + readQRCode(filePath, charset, hintMap));

    }
…
```

실습 - QR code

$ QRcode.java

```
…

    public static void createQRCode(String qrCodeData, String filePath,
      String charset, Map hintMap, int qrCodeheight, int qrCodewidth)
throws WriterException, IOException {
        BitMatrix matrix = new MultiFormatWriter().encode(
          new String(qrCodeData.getBytes(charset), charset),
          BarcodeFormat.QR_CODE, qrCodewidth, qrCodeheight, hintMap);
        MatrixToImageWriter.writeToFile(matrix,
filePath.substring(filePath.lastIndexOf('.') + 1), new File(filePath));
    }

    public static String readQRCode(String filePath, String charset, Map
hintMap) throws FileNotFoundException, IOException, NotFoundException {
     BinaryBitmap binaryBitmap = new BinaryBitmap(new
HybridBinarizer(  new BufferedImageLuminanceSource(
    ImageIO.read(new FileInputStream(filePath)))));
    Result qrCodeResult = new MultiFormatReader().decode(binaryBitmap,
 hintMap);
        return qrCodeResult.getText();
    }
}
```

실습 – QR code

- QRcode programming
  - $ javac -cp “lib*;.” QRCode.java
  - $ java -cp “lib*;.” QRCode

```
C:\workspace\qrcode
λ java -cp "lib/*;." QRCode
QR Code image created successfully!
Data read from QR Code: Hello World!
```

Note

실습자료

1. 블록체인 프로그래밍
2. 암호학 실습
3. 네트워크 실습

실습

DES.java 소스 수정

```
 ….
 try{
 KeyGenerator keygenerator = KeyGenerator.getInstance("DES");
 SecretKey myDesKey = keygenerator.generateKey();

Cipher desCipher;

// Create the cipher
desCipher = Cipher.getInstance("DES/ECB/PKCS5Padding");

// Initialize the cipher for encryption
desCipher.init(Cipher.ENCRYPT_MODE, myDesKey);

//sensitive information
byte[] text = "No body can see me".getBytes();
```

< DES.java >

실습

DES.java 소스 수정

```
    ….
//sensitive information
 byte[] text = "No body can see me".getBytes();

 System.out.println("Text [Byte Format] : " + text);
 System.out.println("Text : " + new String(text));

 // Encrypt the text
 byte[] textEncrypted = desCipher.doFinal(text);

 System.out.println("Text Encryted : " + textEncrypted);

 // Initialize the same cipher for decryption
 desCipher.init(Cipher.DECRYPT_MODE, myDesKey);

 // Decrypt the text
 byte[] textDecrypted = desCipher.doFinal(textEncrypted);
```

< DES.java >

실습

DES

- $ cd c:₩workspace₩security
- $ javac DES.java
- $ java DES

```
c:\workspace\security
λ javac DES.java

c:\workspace\security
λ java DES
Text [Byte Format] : [B@2fc14f68
Text : No body can see me
Text Encryted : [B@591f989e
Text Decryted : No body can see me
```

Note

실습자료

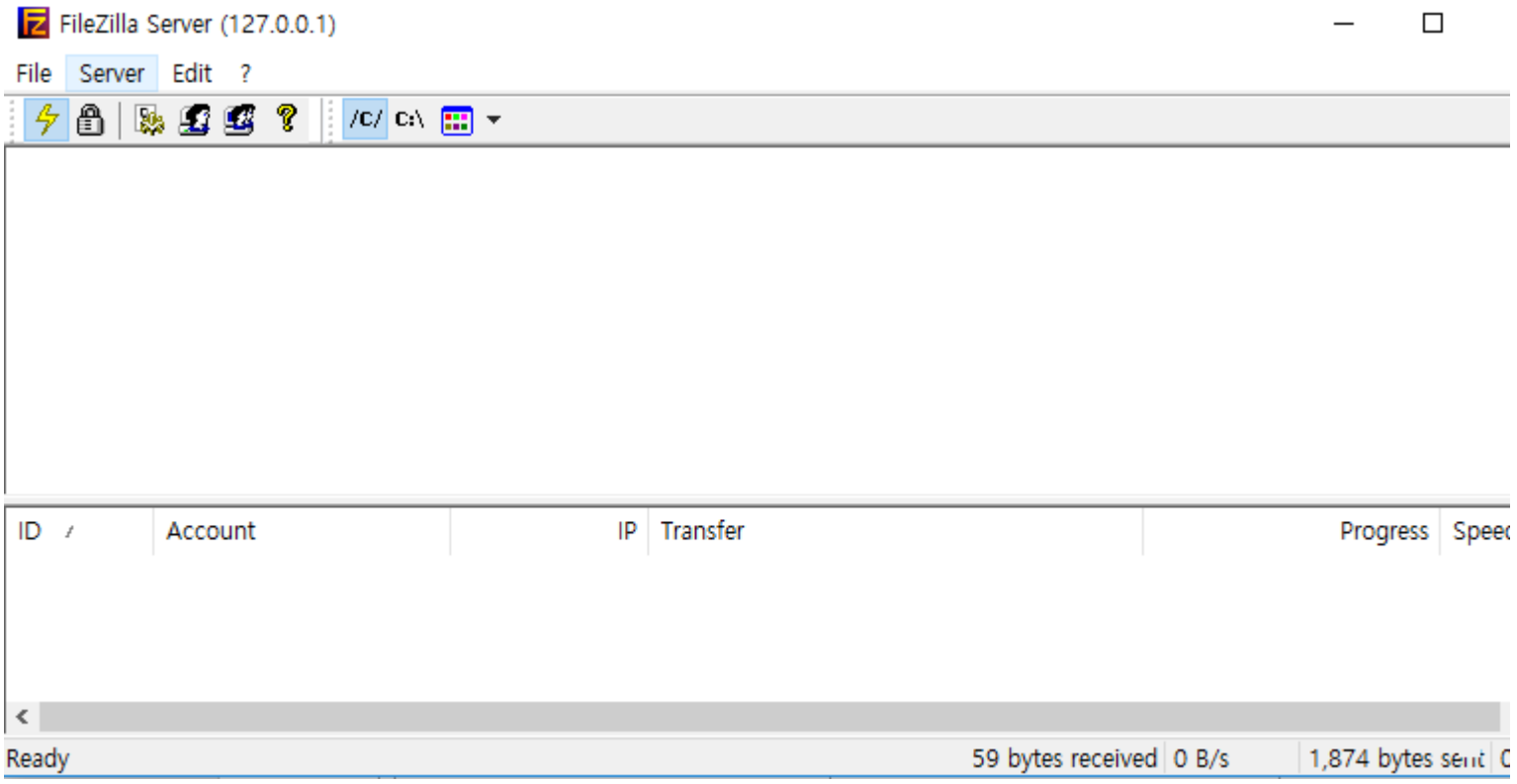
1. 블록체인 프로그래밍
2. 암호학 실습
3. 네트워크 실습

실습

- FTP Server와 Client 설치
  - CD₩utils₩ftp₩FileZilla_Server-0_9_60_2.exe 설치
  - CD₩utils₩ftp₩FileZilla_3.8.1_win32-setup.exe 설치

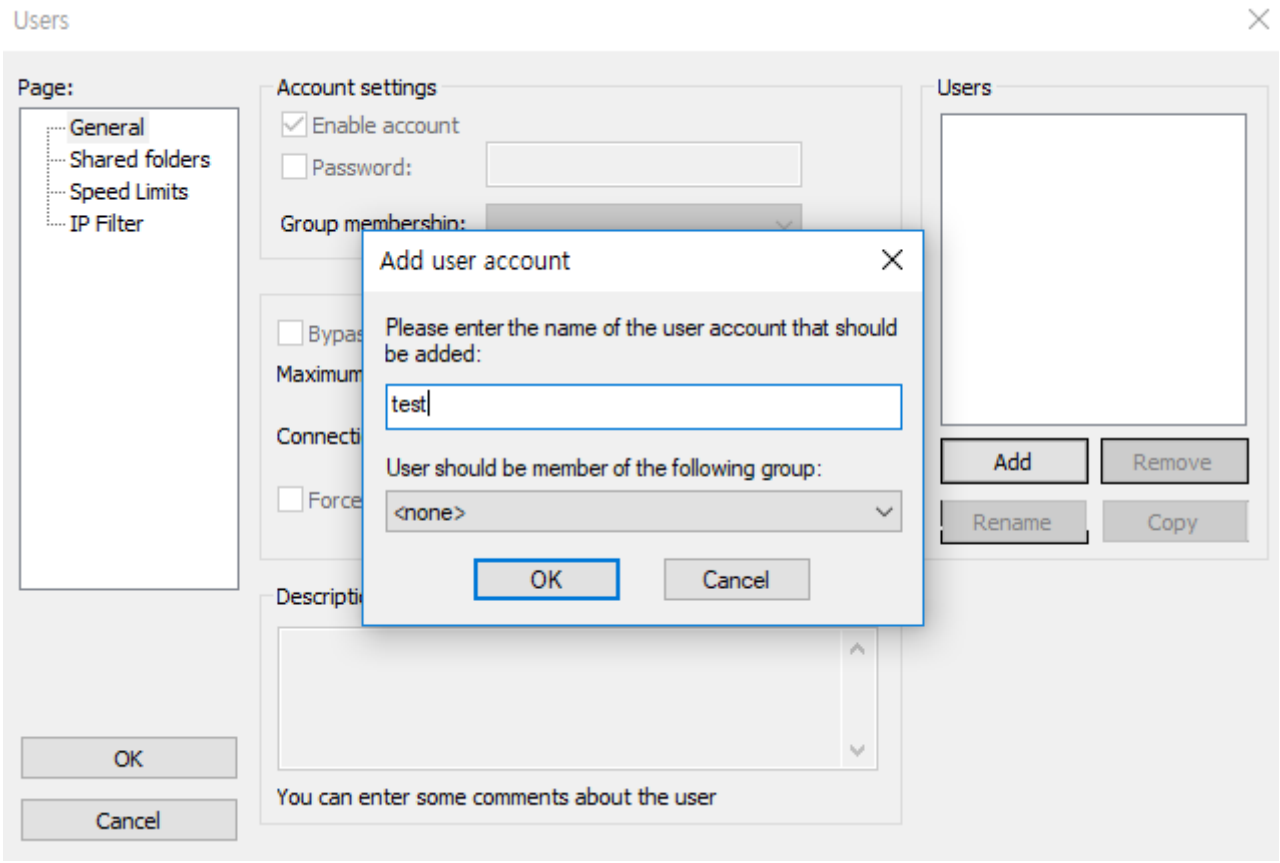
실습

- FTP Server 사용자 추가
  - Edit > Users



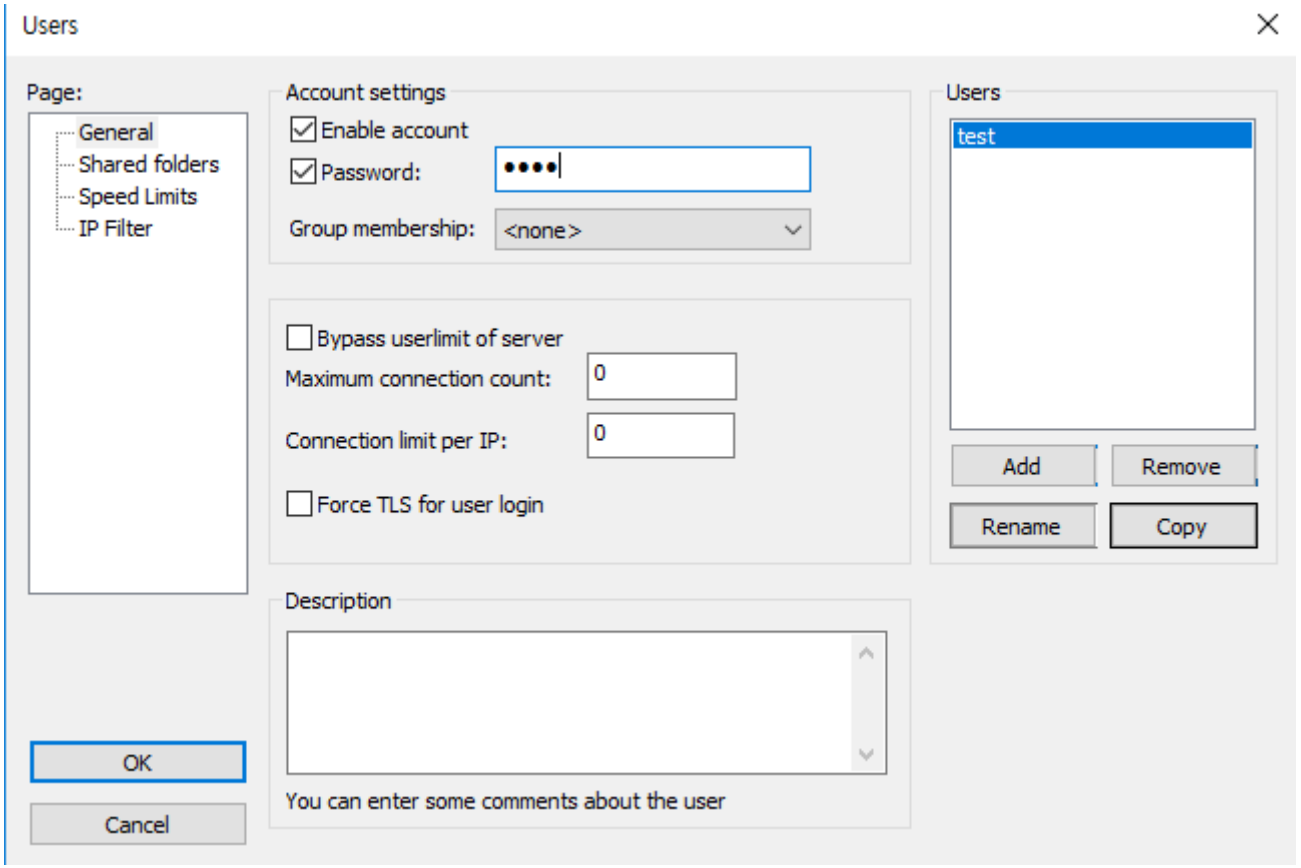
실습

- FTP Server 사용자 추가
  - Users > Add > test > OK



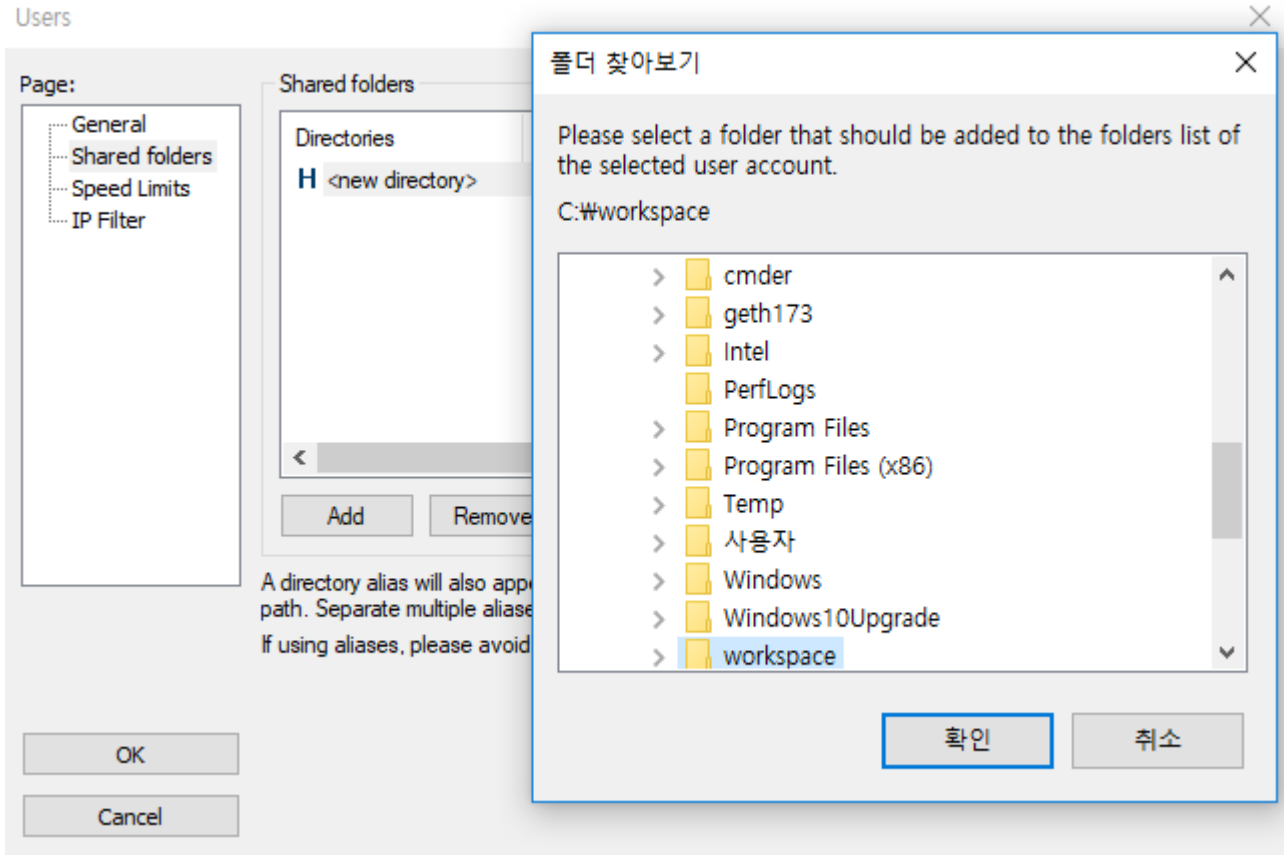
실습

- FTP Server 사용자 추가
  - Password 체크 > ‘1234’ > OK



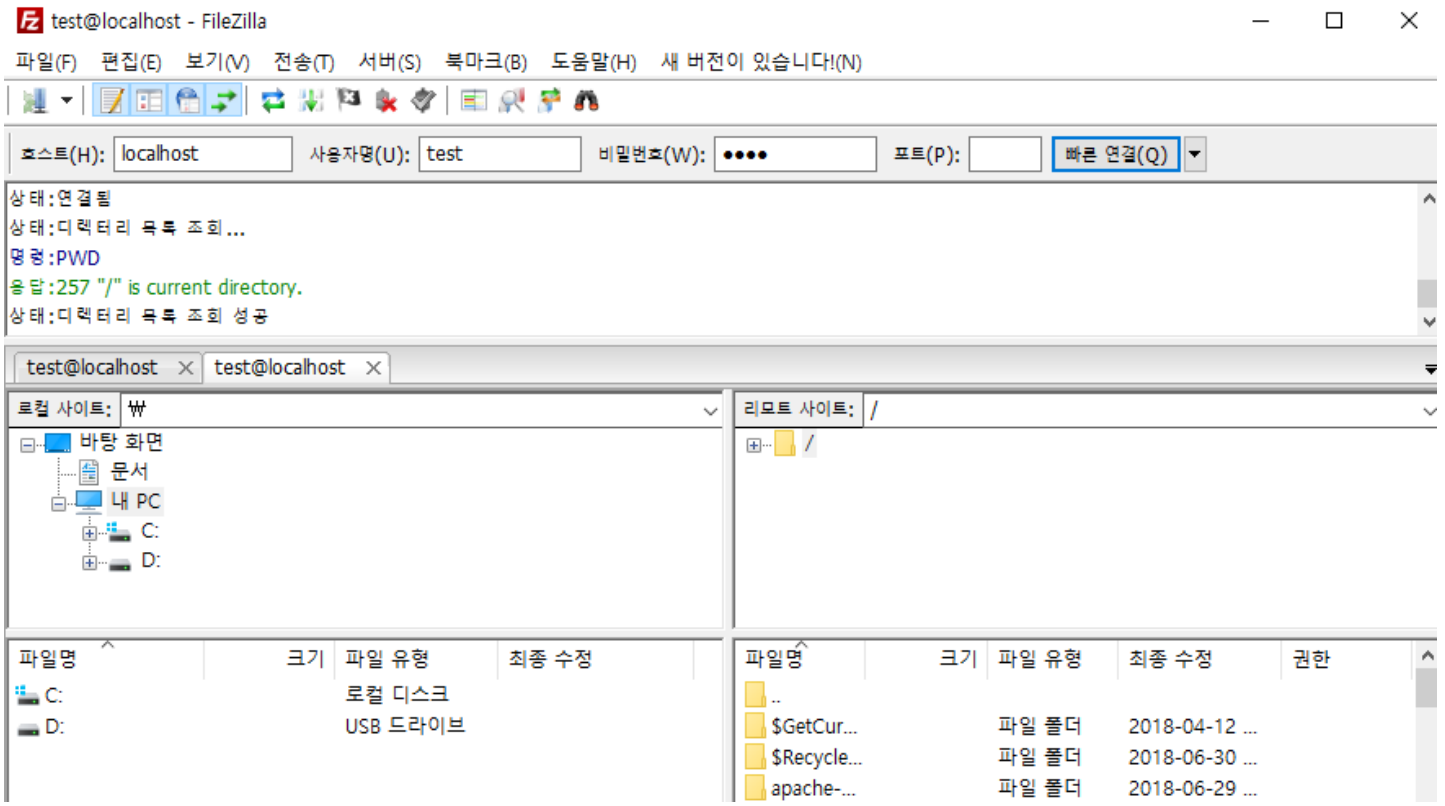
실습

- FTP Server 사용자 추가
  - 공유 폴더 선택 > c:₩workspace > 확인 > OK



실습

- FileZilla Client
  - FileZilla 실행
  - 호스트: localhost, 사용자명: test , 비밀번호: 1234
  - 빠른 연결



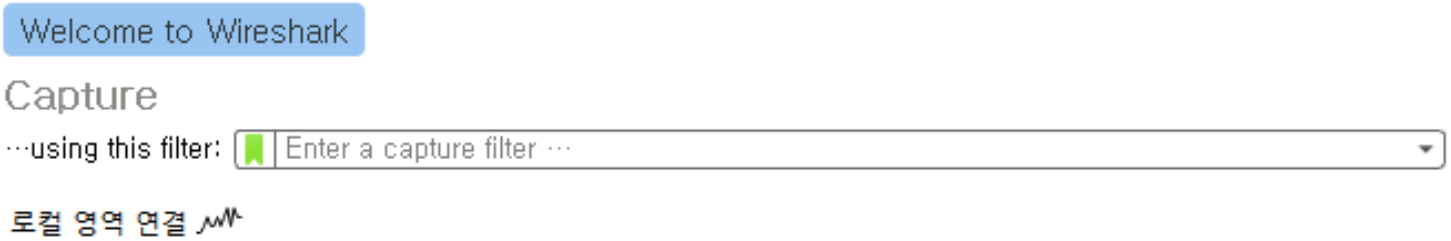
실습

- FileZilla
  - FileZilla 실행
  - 연결 시 상태 명령어와 메시지 확인

```
응답:220-written by Tim Kosse (tim.kosse@filezilla-project.org)
응답:220 Please visit https://filezilla-project.org/
명령:USER test
응답:331 Password required for test
명령:PASS ****
응답:230 Logged on
상태:연결됨
상태:디렉터리 목록 조회...
명령:PWD
응답:257 "/" is current directory.
상태:디렉터리 목록 조회 성공
```

실습

- Wireshark
  - Wireshark 실행
  - "로컬 영역 연결" 을 더블클릭



실습

Wireshark

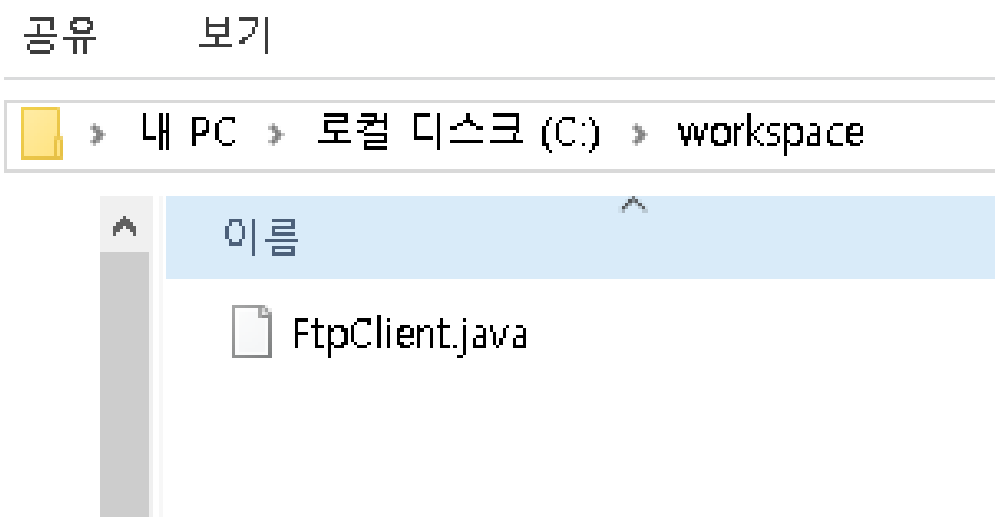
- 필터에 "ftp" 입력

Protocol	Length	Info
FTP	74	Response: 220 (vsFTPd 3.0.2)
FTP	63	Request: USER pi
FTP	88	Response: 331 Please specify the password.
FTP	70	Request: PASS raspberry
FTP	77	Response: 230 Login successful.
FTP	59	Request: PWD
FTP	70	Response: 257 "/home/pi"

```
> Frame 2469: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0
> Ethernet II, Src: Raspberr_8b:d0:4e (b8:27:eb:8b:d0:4e), Dst: HewlettP_6c:ef:69 (c4:34:6b:6c:ef:69)
> Internet Protocol Version 4, Src: 166.104.115.201, Dst: 166.104.115.137
> Transmission Control Protocol, Src Port: 21 (21), Dst Port: 56332 (56332), Seq: 78, Ack: 31, Len: 16
v File Transfer Protocol (FTP)
  v 257 "/home/pi"\r\n
      Response code: PATHNAME created (257)
      Response arg: "/home/pi"
```

실습

- FTP programming
  - C:₩workspace 폴더 생성
  - CD₩day02₩ftp₩FtpClient.java 파일을 c:₩workspace 폴더로 복사



실습

FtpClient.java 소스 수정

```
….
Socket ctrlSocket;//    제어용 소켓
public PrintWriter ctrlOutput;//  제어 출력용 스트림
public BufferedReader ctrlInput;//  제어 입력용 스트림
final int CTRLPORT =  21;          //   ftp 제어용 포트
// openConnection  메소드
// 주소와 포트 번호로부터 소켓을 만들고 제어용 스트림을 작성한다.
public void openConnection(String host)
    throws IOException,UnknownHostException
{
    ctrlSocket = new Socket(host, CTRLPORT);
    ctrlOutput = new PrintWriter(ctrlSocket.getOutputStream());
    ctrlInput
      = new BufferedReader(new InputStreamReader(ctrlSocket.getInputStream()));
}
….
```

실습

FtpClient.java 소스 수정

```
…
try{
        System.out.println("로그인 이름을 입력하세요 : ");
        loginName = lineread.readLine();
        // USER 명령에 의한 로그인
        ctrlOutput.println("USER " + loginName);
        ctrlOutput.flush();
        // PASS 명령에 의한 패스워드의 입력
        System.out.println("패스워드를 입력하세요 : ");
        password = lineread.readLine();
        ctrlOutput.println("PASS " + password);
        ctrlOutput.flush();
….
```

실습

FtpClient.java 소스 수정

```
…
public void doCd()
{
    String dirName = "" ;
    BufferedReader lineread
       = new BufferedReader(new InputStreamReader(System.in)) ;
    try{
         System.out.println("디렉토리 이름을 입력하세요 : ") ;
         dirName = lineread.readLine() ;
      ctrlOutput.println("CWD " + dirName);   // CWD 명령
         ctrlOutput.flush() ;
    }catch(Exception e)
    {
         e.printStackTrace();
         System.exit(1);
    }
}
```

실습

FtpClient.java 소스 수정

```
…
public void doLs()
{
   try{
        int n ;
        byte[] buff = new byte[1024] ;
        // 데이터용 연결(connection)을 만든다.
        Socket dataSocket = dataConnection("LIST");
        //  데이터를 읽어 처리하는 스트림을 사용한다.
        BufferedInputStream dataInput
          = new BufferedInputStream(dataSocket.getInputStream()) ;
        //  디렉토리 정보를 읽고 처리한다.
        while((n = dataInput.read(buff)) > 0){
            System.out.write(buff,0,n)  ;
        }
        dataSocket.close() ;
   }catch(Exception e)
```

실습

FtpClient.java 소스 수정

```
…
try{
        int n ;
        byte[] buff = new byte[1024] ;
        // 서버상의 파일의 이름을 지정한다.
        System.out.println("파일 이름을 입력하세요 : ") ;
        fileName = lineread.readLine() ;
        // 클라이언트상에 수신용 파일을 준비한다.
        FileOutputStream outfile =  new FileOutputStream(fileName) ;
        // 파일 전송용 데이터 스트림을 작성한다.
        Socket dataSocket = dataConnection("RETR " + fileName);
        BufferedInputStream dataInput
          = new BufferedInputStream(dataSocket.getInputStream()) ;
        // 서버로부터 데이터를 받아 파일로 저장한다.
        while((n = dataInput.read(buff)) > 0){
            outfile.write(buff,0,n) ;
        }
```

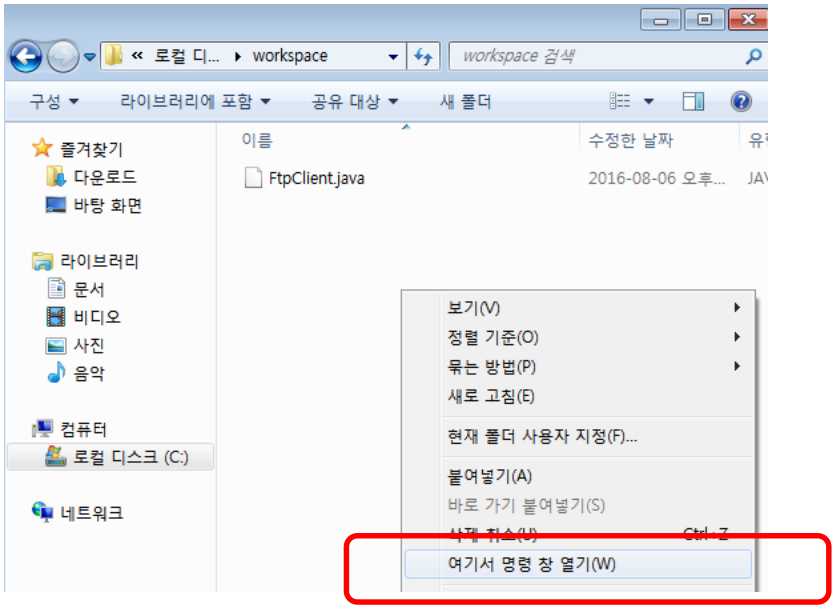
실습

FtpClient.java 소스 수정

```
…
try{
        int n ;
        byte[] buff = new byte[1024] ;
        FileInputStream sendfile = null ;
        // 파일 이름을 지정한다.
        System.out.println("파일명을 입력하세요 : ") ;
        ….
            return ;
        }
        // 전송용 데이터 스트림을 사용한다.
        Socket dataSocket = dataConnection("STOR " + fileName);
        OutputStream outstr =  dataSocket.getOutputStream() ;
        // 파일을 읽어 네트워크를 경유하여 서버로 보낸다.
        …
        }
```

실습

- FTP programming
  - Shift + 마우스 오른쪽 버튼
  - 여기서 명령 창 열기 선택



Shift + 마우스 오른쪽 버튼

실습

- FTP programming
  - 자바 소스 컴파일 및 실행
  - 로그인: test, 패스워드: 1234

```
> javac FtpClient.java
> java  FtpClient
(> java    -cl    .    FtpClient )

로그인 : test
패스워드: 1234
```

```
C:\workspace\ftp
λ java FtpClient
로그인 이름을 입력하세요 :
220-FileZilla Server 0.9.60 beta
220-written by Tim Kosse (tim.kosse@filezilla-project.org)
220 Please visit https://filezilla-project.org/
test
패스워드를 입력하세요 :
331 Password required for test
1234
>Command?
230 Logged on
2 ls    3 cd    4 get    5 put    6 ascii    7 binary    9 quit
```

실습

- UDP programming
  - C:₩workspace 폴더 생성
  - CD₩Day02₩udp 폴더를 c:₩workspace 폴더로 복사

실습

UDPClient.java 소스 수정

```
 ….
 public static void main(String args[]) throws Exception
{
   BufferedReader inFromUser =
      new BufferedReader(new InputStreamReader(System.in));
   DatagramSocket clientSocket = new DatagramSocket();
   InetAddress IPAddress = InetAddress.getByName("127.0.0.1");
   byte[] sendData = new byte[1024];
   byte[] receiveData = new byte[1024];
   String sentence = inFromUser.readLine();
   sendData = sentence.getBytes();
   DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length,
            IPAddress, 9876);
   ….
```

<UDPClient.java>

실습

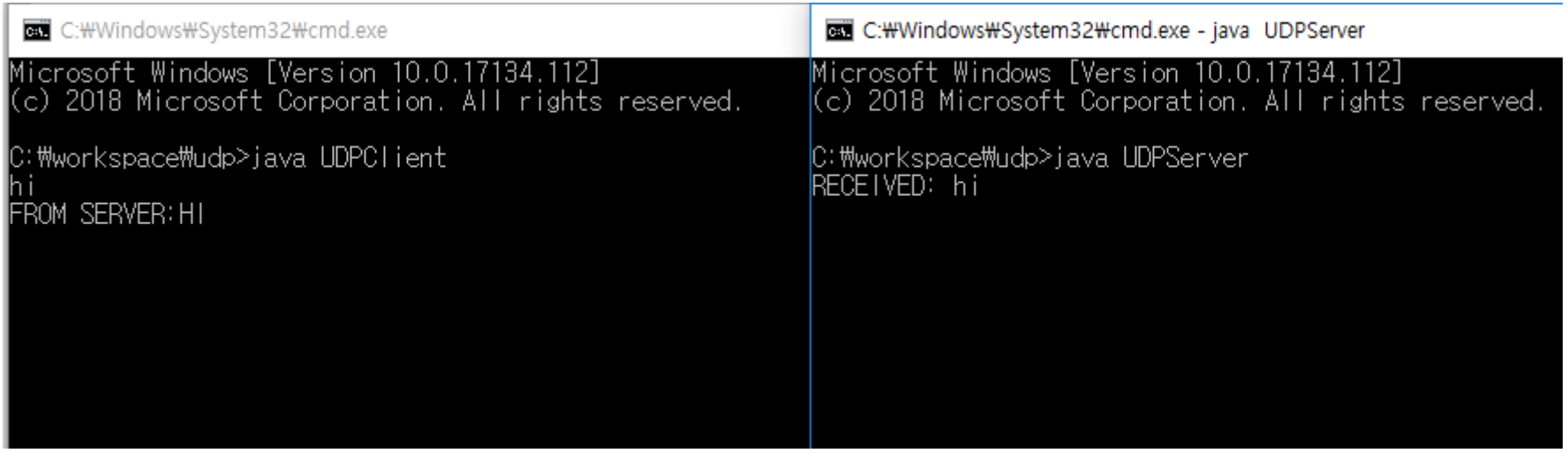
UDPClient.java 소스 수정

```
    ….
      clientSocket.send(sendPacket);
      DatagramPacket receivePacket = new DatagramPacket(receiveData,
               receiveData.length);
      clientSocket.receive(receivePacket);
      String modifiedSentence = new String(receivePacket.getData());
      System.out.println("FROM SERVER:" + modifiedSentence);
      clientSocket.close();
  }
```

<UDPClient.java>

실습

- UDP programming
  - $ cd workspace₩udp
  - $ javac *.java
  - 두 개의 윈도우를 열기
  - cmd1> $ java UDPServer
  - cmd2> $ java UDPClient
  - hi



실습

- TCP programming
  - CD₩Day02₩tcp 폴더를 c:₩workspace 폴더로 복사

실습

TcpClientTest.java 소스 수정

```
….
public class TcpClientTest {
public static void main(String[] args) {
    try {
      String serverIP = “127.0.0.1"; // 127.0.0.1 & localhost 본인
        System.out.println("서버에 연결중입니다. 서버 IP : " + serverIP);
        // 소켓을 생성하여 연결을 요청한다.
      Socket socket = new Socket(serverIP, 5000);
        // 소켓의 입력스트림을 얻는다.
        InputStream in = socket.getInputStream();
      DataInputStream dis = new DataInputStream(in);
        // 소켓으로 부터 받은 데이터를 출력한다.
        System.out.println("서버로부터 받은 메세지 : " + dis.readUTF());
        System.out.println("연결을 종료합니다.");
        // 스트림과 소켓을 닫는다.
        dis.close();
      socket.close();
  ….
```

<TcpClientTest.java>

실습

- TCP programming
  - $ cd workspace₩tcp
  - $ javac *.java
  - $ javac -encoding euc-kr TcpServerTest.java
  - 두 개의 윈도우를 열기
  - cmd1> $ java TCPServerTest
  - cmd2> $ java TCPClientTest
  - hi

실습

- TCP Multiple programming
  - CD₩Day02₩multi 폴더를 c:₩workspace 폴더로 복사

실습

MultichatClient.java 소스 수정

```
….
try {
        String serverIp = “127.0.0.1";
        // 소켓을 생성하여 연결을 요청한다.
        Socket socket = new Socket(serverIp, 7777);
        System.out.println("서버에 연결되었습니다.");
        Thread sender = new Thread(new ClientSender(socket, args[0]));
        Thread receiver = new Thread(new ClientReceiver(socket));
        sender.start();
        receiver.start();
    } catch (ConnectException ce) {      ….
```

<MultichatClient.java>

실습

MultichatClient.java 소스 수정

```
….
static class ClientSender extends Thread {
    Socket socket;
    DataOutputStream out;
    String name;
    ClientSender(Socket socket, String name) {
        this.socket = socket;
        try {
   out = new DataOutputStream(socket.getOutputStream());
            this.name = name;
        } catch (Exception e) {
        }
    }
….
```

<MultichatClient.java>

실습

MultichatClient.java 소스 수정

```
….
static class ClientReceiver extends Thread {
    Socket socket;
    DataInputStream in;
    ClientReceiver(Socket socket) {
        this.socket = socket;
        try {
       in = new DataInputStream(socket.getInputStream());
        } catch (IOException e) {
        }
    }
….
```

<MultichatClient.java>

실습

- TCP Multiple programming
  - $ cd workspace₩multi
  - $ javac *.java
  - $ javac -encoding euc-kr MultichatServer.java
  - 세 개의 윈도우를 열기
  - cmd1> $ java MultichatServer
  - cmd2> $ java MultichatClient good
  - cmd3> $ java MultichatClient nide
  - hi

실습

- Peer to Peer programming
  - CD₩Day02₩p2pchat 폴더를 c:₩workspace 폴더로 복사

실습

- Peer to Peer programming
  - $ cd workspace₩p2pchat
  - $ javac *.java
  - $ java P2PChatConsole good

```
C:\workspace\p2pchat>r.bat

C:\workspace\p2pchat>java  P2PChatConsole test
P2PChat ver 0.2
Listening At: 192.168.145.223,192.168.56.1,192.168.99.1,254.128.0.0.0.0.0.0.101.177.120.66.253.116.84.1
13,254.128.0.0.0.0.0.0.228.97.38.164.85.242.59.17,254.128.0.0.0.0.0.0.125.163.29.86.118.180.168.214:115
81
Connection From: 1 (?) 192.168.145.167:?
Send: HELO 192.168.145.223,192.168.56.1,192.168.99.1,254.128.0.0.0.0.0.0.101.177.120.66.253.116.84.113,
254.128.0.0.0.0.0.0.228.97.38.164.85.242.59.17,254.128.0.0.0.0.0.0.125.163.29.86.118.180.168.214:11581
test
Recv: HELO 192.168.145.167,192.168.56.1,254.128.0.0.0.0.0.0.212.21.44.175.175.129.58.118,254.128.0.0.0.
0.0.0.44.144.171.176.88.72.254.46:11581 good
HELO From: 1 good 192.168.145.167,192.168.56.1,254.128.0.0.0.0.0.0.212.21.44.175.175.129.58.118,254.128
.0.0.0.0.0.0.44.144.171.176.88.72.254.46:11581
Recv: CHAT oo
1 good: oo
Recv: CHAT have  a nice day
1 good: have  a nice day
```

실습

- Peer to Peer programming
  - $ cd workspace₩p2pchat
  - $ java P2PChatConsole nice **192.168.145.223**:11581

```
c:₩workspace₩p2pchat>java P2PChatConsole good 192.168.145.223:11581
P2PChat ver 0.2
Listening At: 192.168.145.167,192.168.56.1,254.128.0.0.0.0.0.0.212.21.44.175.175
.129.58.118,254.128.0.0.0.0.0.0.44.144.171.176.88.72.254.46:11581
Connection From: 1 (?) 192.168.145.223:11581
Send: HELO 192.168.145.167,192.168.56.1,254.128.0.0.0.0.0.0.212.21.44.175.175.12
9.58.118,254.128.0.0.0.0.0.0.44.144.171.176.88.72.254.46:11581 good
Recv: HELO 192.168.145.223,192.168.56.1,192.168.99.1,254.128.0.0.0.0.0.0.101.177
.120.66.253.116.84.113,254.128.0.0.0.0.0.0.228.97.38.164.85.242.59.17,254.128.0.
0.0.0.0.0.125.163.29.86.118.180.168.214:11581 test
HELO From: 1 test 192.168.145.223,192.168.56.1,192.168.99.1,254.128.0.0.0.0.0.0.
101.177.120.66.253.116.84.113,254.128.0.0.0.0.0.0.228.97.38.164.85.242.59.17,254
.128.0.0.0.0.0.0.125.163.29.86.118.180.168.214:11581
oo
Send: CHAT oo
good: oo
have  a nice day
Send: CHAT have  a nice day
good: have  a nice day
```

Note