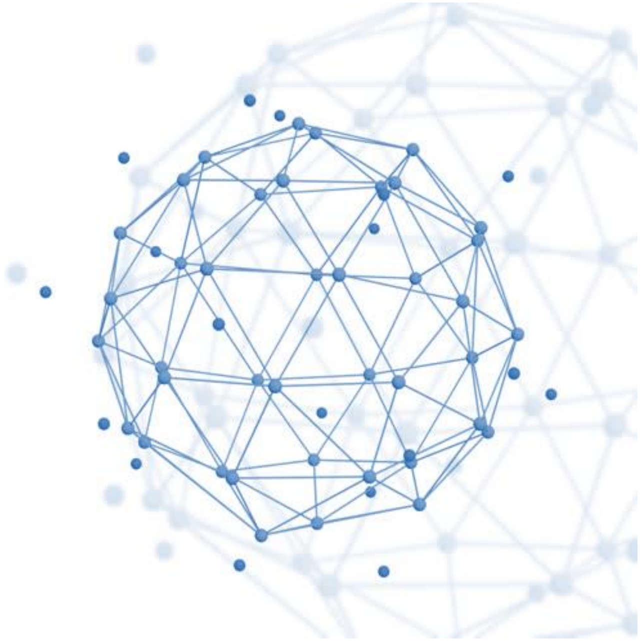
CoAP Programming 3

2020.04.01

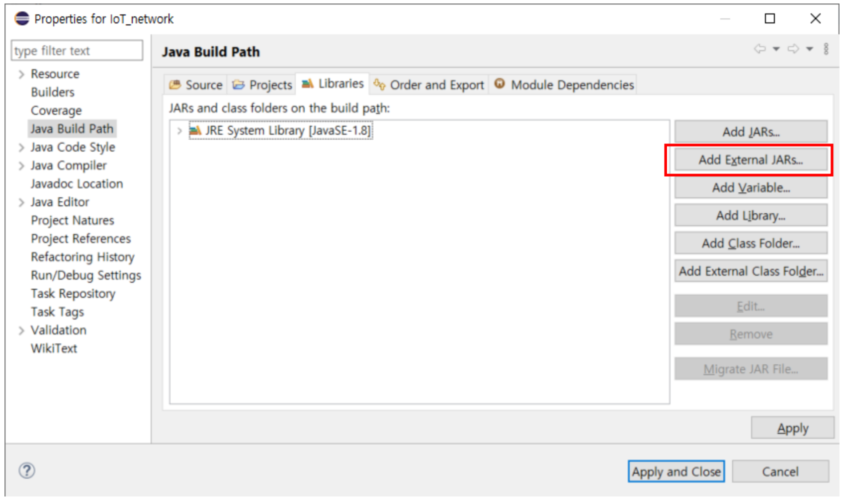
Sang-woo Lee
glutton.leesw@gmail.com



Contents

Jsoup – Bitcoin 가격 파싱

- Jsoup import



Properties for IoT_network

type filter text

- Resource
- Builders
- Coverage
- Java Build Path
- Java Code Style
- Java Compiler
- Javadoc Location
- Java Editor
- Project Natures
- Project References
- Refactoring History
- Run/Debug Settings
- Task Repository
- Task Tags
- Validation
- WikiText

Java Build Path

Source | Projects | Libraries | Order and Export | Module Dependencies

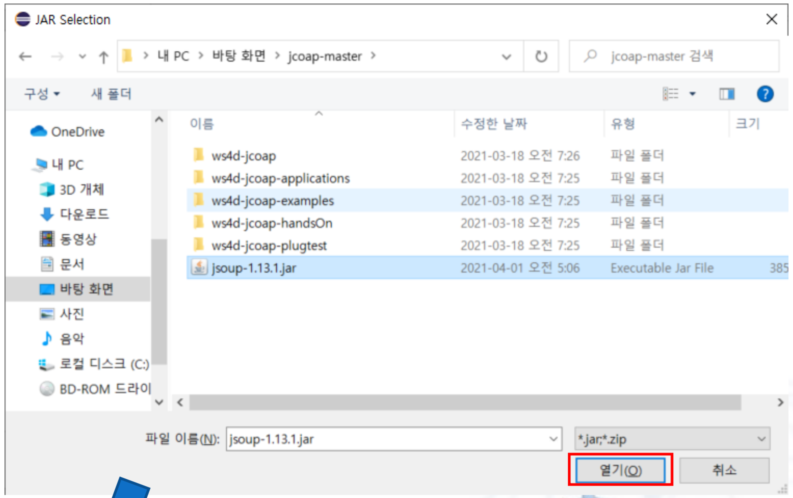
JARs and class folders on the build path:

JRE System Library [JavaSE-1.8]

Add JARs... | Add External JARs... | Add Variable... | Add Library... | Add Class Folder... | Add External Class Folder... | Edit... | Remove | Migrate JAR File...

Apply

Apply and Close | Cancel



JAR Selection

내 PC › 바탕 화면 › jcoap-master ›

jcoap-master 검색

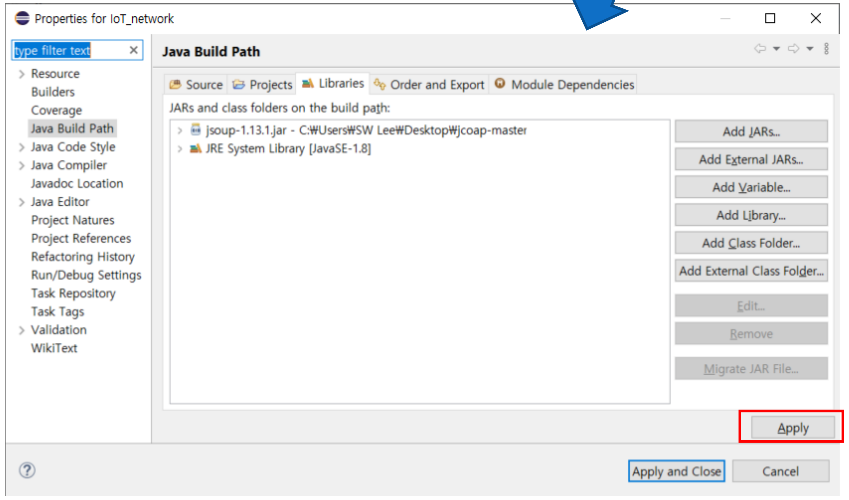
구성 ▾ 새 폴더

OneDrive, 내 PC, 3D 개체, 다운로드, 동영상, 문서, 바탕 화면, 사진, 음악, 로컬 디스크 (C:), BD-ROM 드라이

이름	수정한 날짜	유형	크기
ws4d-jcoap	2021-03-18 오전 7:26	파일 폴더	
ws4d-jcoap-applications	2021-03-18 오전 7:25	파일 폴더	
ws4d-jcoap-examples	2021-03-18 오전 7:25	파일 폴더	
ws4d-jcoap-handsOn	2021-03-18 오전 7:25	파일 폴더	
ws4d-jcoap-plugtest	2021-03-18 오전 7:25	파일 폴더	
jsoup-1.13.1.jar	2021-04-01 오전 5:06	Executable Jar File	385

파일 이름(N): jsoup-1.13.1.jar   *.jar;*.zip

열기(O) | 취소



Properties for IoT_network

type filter text

- Resource
- Builders
- Coverage
- Java Build Path
- Java Code Style
- Java Compiler
- Javadoc Location
- Java Editor
- Project Natures
- Project References
- Refactoring History
- Run/Debug Settings
- Task Repository
- Task Tags
- Validation
- WikiText

Java Build Path

Source | Projects | Libraries | Order and Export | Module Dependencies

JARs and class folders on the build path:

jsoup-1.13.1.jar - C:₩Users₩SW Lee₩Desktop₩jcoap-master
JRE System Library [JavaSE-1.8]

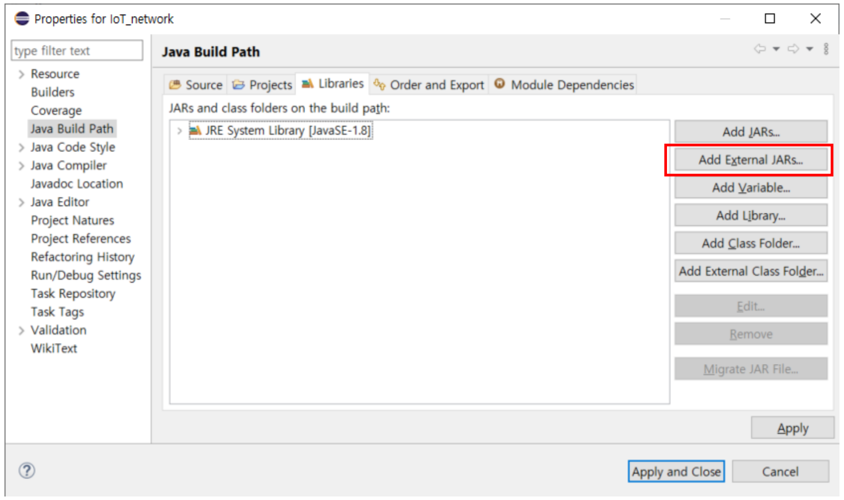
Add JARs... | Add External JARs... | Add Variable... | Add Library... | Add Class Folder... | Add External Class Folder... | Edit... | Remove | Migrate JAR File...

Apply

Apply and Close | Cancel

Jsoup – Bitcoin 가격 파싱

- Jsoup import



Properties for IoT_network

type filter text

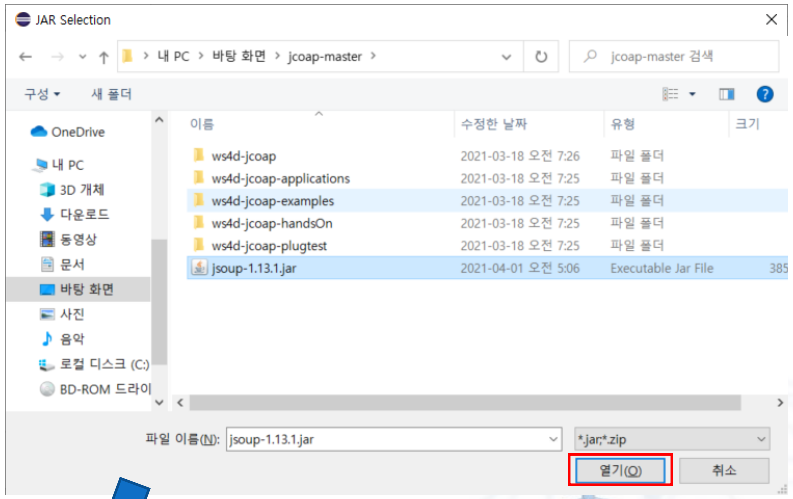
- Resource
- Builders
- Coverage
- Java Build Path
- Java Code Style
- Java Compiler
- Javadoc Location
- Java Editor
- Project Natures
- Project References
- Refactoring History
- Run/Debug Settings
- Task Repository
- Task Tags
- Validation
- WikiText

Java Build Path

Source | Projects | Libraries | Order and Export | Module Dependencies

JARs and class folders on the build path:

- JRE System Library [JavaSE-1.8]



JAR Selection

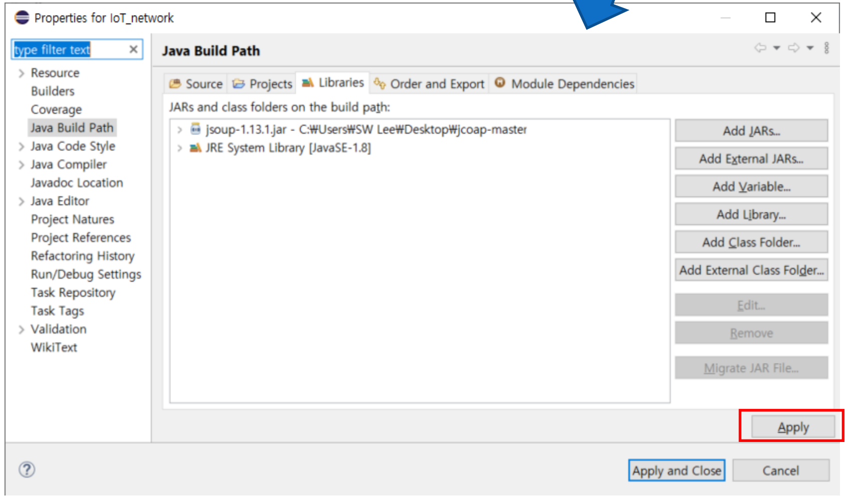
내 PC › 바탕 화면 › jcoap-master ›

jcoap-master 검색

이름	수정한 날짜	유형	크기
ws4d-jcoap	2021-03-18 오전 7:26	파일 폴더	
ws4d-jcoap-applications	2021-03-18 오전 7:25	파일 폴더	
ws4d-jcoap-examples	2021-03-18 오전 7:25	파일 폴더	
ws4d-jcoap-handsOn	2021-03-18 오전 7:25	파일 폴더	
ws4d-jcoap-plugtest	2021-03-18 오전 7:25	파일 폴더	
jsoup-1.13.1.jar	2021-04-01 오전 5:06	Executable Jar File	385

파일 이름(N): jsoup-1.13.1.jar

.jar;.zip



Properties for IoT_network

type filter text

- Resource
- Builders
- Coverage
- Java Build Path
- Java Code Style
- Java Compiler
- Javadoc Location
- Java Editor
- Project Natures
- Project References
- Refactoring History
- Run/Debug Settings
- Task Repository
- Task Tags
- Validation
- WikiText

Java Build Path

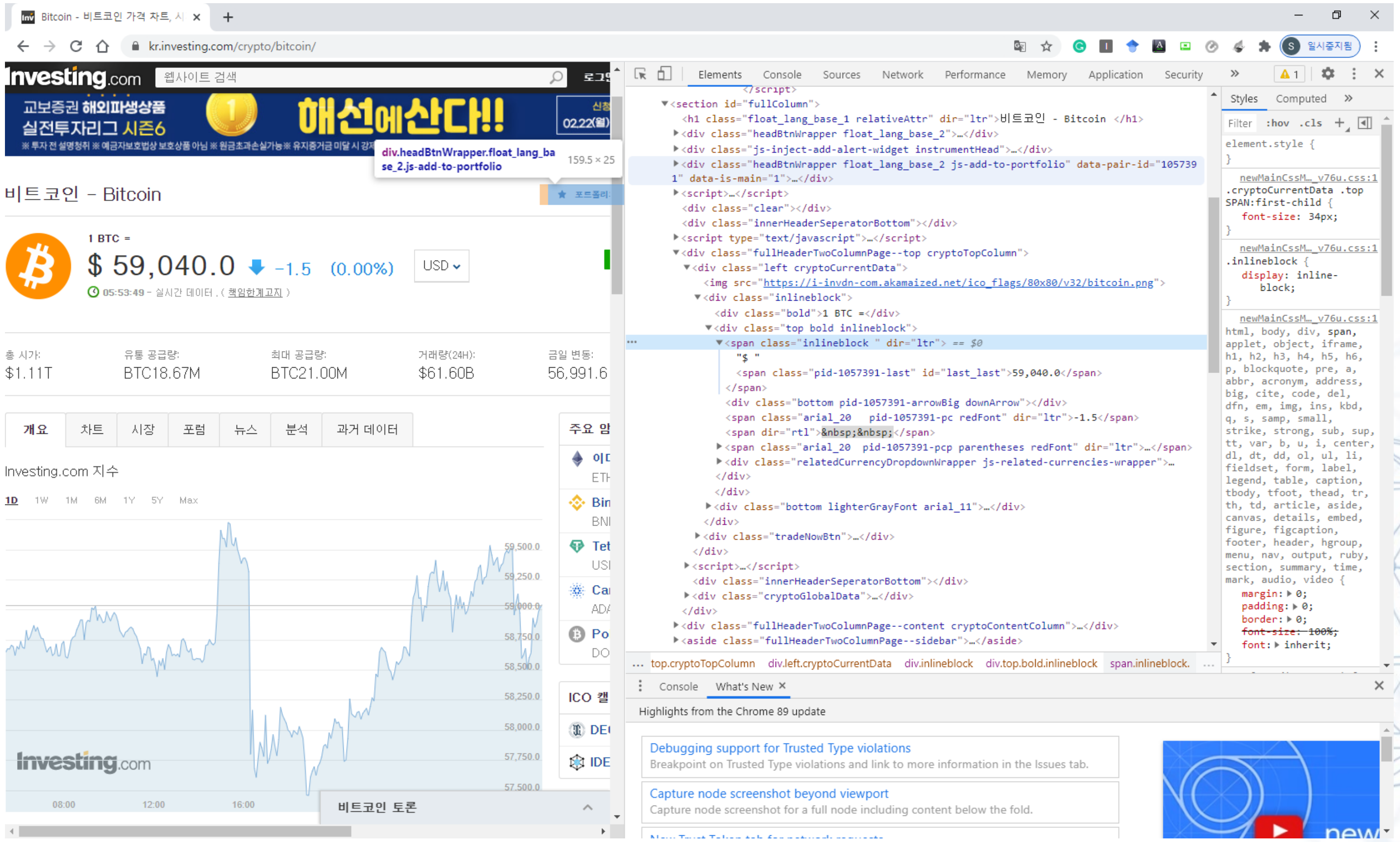
Source | Projects | Libraries | Order and Export | Module Dependencies

JARs and class folders on the build path:

- jsoup-1.13.1.jar - C:₩Users₩SW Lee₩Desktop₩jcoap-master
- JRE System Library [JavaSE-1.8]

Jsoup – Bitcoin 가격 파싱

- Bitcoin 가격 확인 사이트



kr.investing.com/crypto/bitcoin/

비트코인 – Bitcoin

1 BTC =
$ 59,040.0 -1.5 (0.00%) USD
05:53:49 - 실시간 데이터 . (책임한계고지)

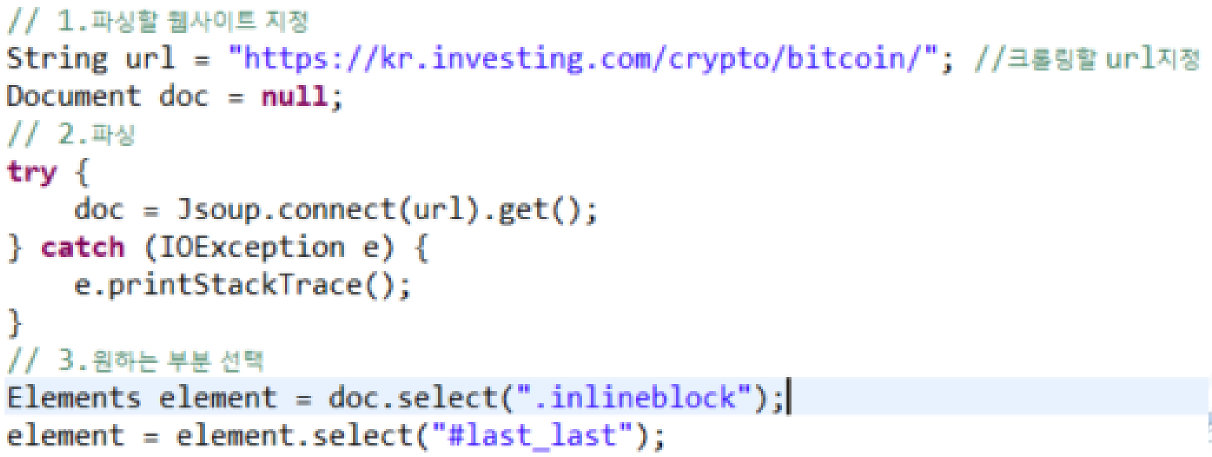
총 시가:	유통 공급량:	최대 공급량:	거래량(24H):	금일 변동:
$1.11T	BTC18.67M	BTC21.00M	$61.60B	56,991.6

```
</script>
▼<section id="fullColumn">
    <h1 class="float_lang_base_1 relativeAttr" dir="ltr">비트코인 - Bitcoin </h1>
  ▶<div class="headBtnWrapper float_lang_base_2">…</div>
  ▶<div class="js-inject-add-alert-widget instrumentHead">…</div>
  ▶<div class="headBtnWrapper float_lang_base_2 js-add-to-portfolio" data-pair-id="105739
  1" data-is-main="1">…</div>
  ▶<script>…</script>
    <div class="clear"></div>
    <div class="innerHeaderSeperatorBottom"></div>
  ▶<script type="text/javascript">…</script>
  ▼<div class="fullHeaderTwoColumnPage--top cryptoTopColumn">
    ▼<div class="left cryptoCurrentData">
        <img src="https://i-invdn-com.akamaized.net/ico_flags/80x80/v32/bitcoin.png">
      ▼<div class="inlineblock">
          <div class="bold">1 BTC =</div>
        ▼<div class="top bold inlineblock">
          ▼<span class="inlineblock " dir="ltr"> == $0
              "$ "
              <span class="pid-1057391-last" id="last_last">59,040.0</span>
            </span>
            <div class="bottom pid-1057391-arrowBig downArrow"></div>
            <span class="arial_20   pid-1057391-pc redFont" dir="ltr">-1.5</span>
            <span dir="rtl">&nbsp;&nbsp;</span>
          ▶<span class="arial_20  pid-1057391-pcp parentheses redFont" dir="ltr">…</span>
          ▶<div class="relatedCurrencyDropdownWrapper js-related-currencies-wrapper">…
          </div>
          </div>
        ▶<div class="bottom lighterGrayFont arial_11">…</div>
        </div>
      ▶<div class="tradeNowBtn">…</div>
      </div>
    ▶<script>…</script>
      <div class="innerHeaderSeperatorBottom"></div>
    ▶<div class="cryptoGlobalData">…</div>
    </div>
  ▶<div class="fullHeaderTwoColumnPage--content cryptoContentColumn">…</div>
  ▶<aside class="fullHeaderTwoColumnPage--sidebar">…</aside>
```

```
element.style {
}
.cryptoCurrentData .top
SPAN:first-child {
  font-size: 34px;
}
.inlineblock {
  display: inline-
    block;
}
```

Jsoup – Bitcoin 가격 파싱

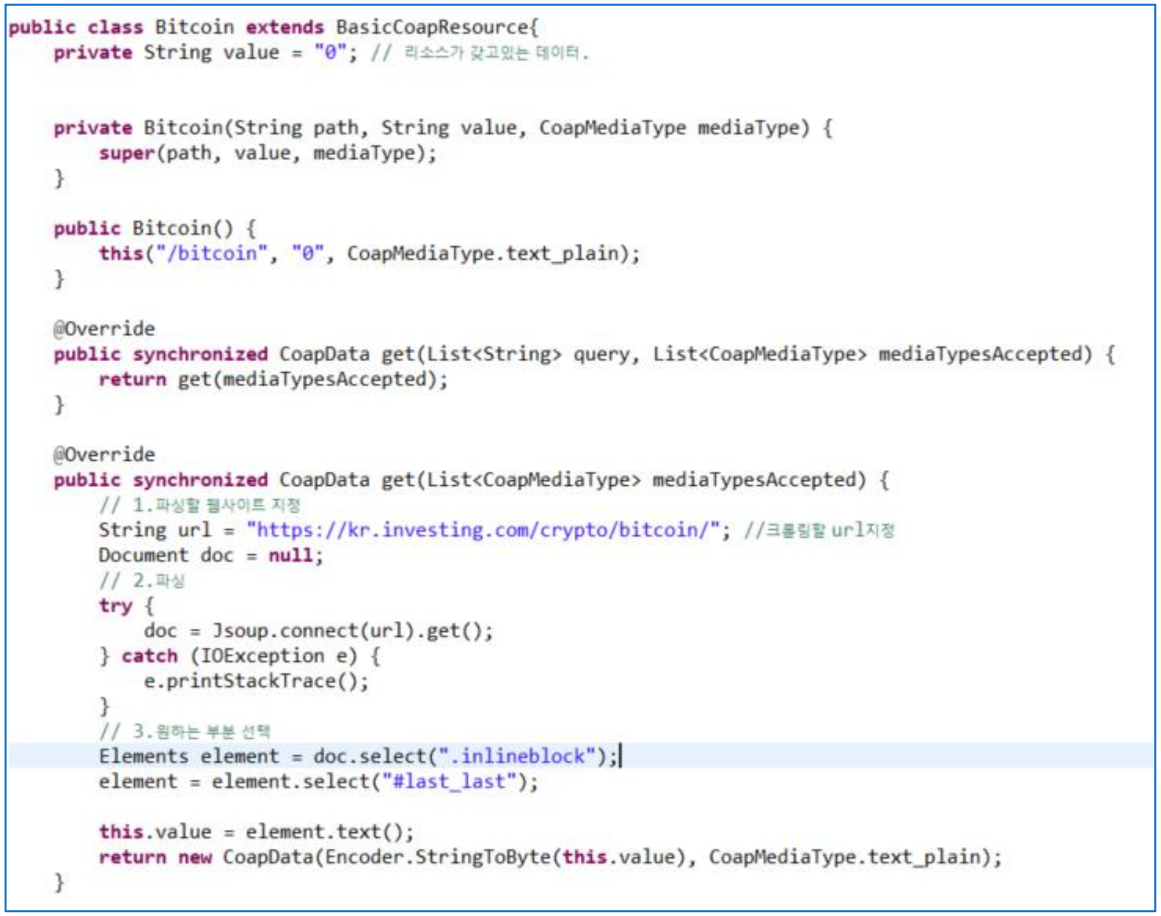
- Bitcoin 가격 확인 사이트



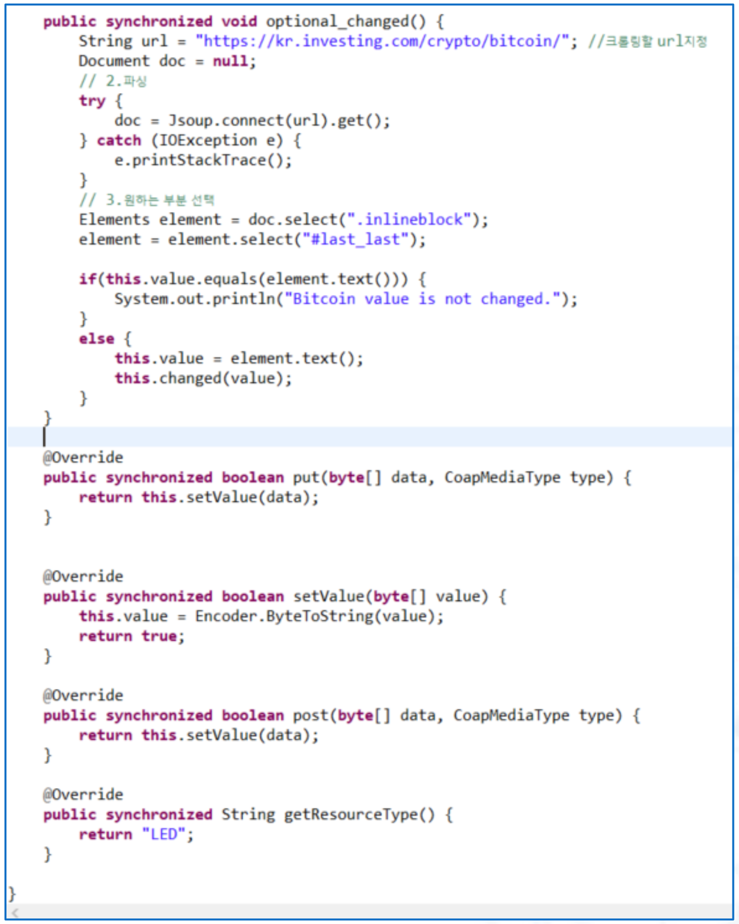
```java
// 1.파싱할 웹사이트 지정
String url = "https://kr.investing.com/crypto/bitcoin/"; //크롤링할 url지정
Document doc = null;
// 2.파싱
try {
    doc = Jsoup.connect(url).get();
} catch (IOException e) {
    e.printStackTrace();
}
// 3.원하는 부분 선택
Elements element = doc.select(".inlineblock");
element = element.select("#last_last");
```

Bitcoin Resource

- Bitcoin Resource



```java
public class Bitcoin extends BasicCoapResource{
    private String value = "0"; // 리소스가 갖고있는 데이터.


    private Bitcoin(String path, String value, CoapMediaType mediaType) {
        super(path, value, mediaType);
    }

    public Bitcoin() {
        this("/bitcoin", "0", CoapMediaType.text_plain);
    }

    @Override
    public synchronized CoapData get(List<String> query, List<CoapMediaType> mediaTypesAccepted) {
        return get(mediaTypesAccepted);
    }

    @Override
    public synchronized CoapData get(List<CoapMediaType> mediaTypesAccepted) {
        // 1.파싱할 웹사이트 지정
        String url = "https://kr.investing.com/crypto/bitcoin/"; //크롤링할 url지정
        Document doc = null;
        // 2.파싱
        try {
            doc = Jsoup.connect(url).get();
        } catch (IOException e) {
            e.printStackTrace();
        }
        // 3.원하는 부분 선택
        Elements element = doc.select(".inlineblock");
        element = element.select("#last_last");

        this.value = element.text();
        return new CoapData(Encoder.StringToByte(this.value), CoapMediaType.text_plain);
    }
```

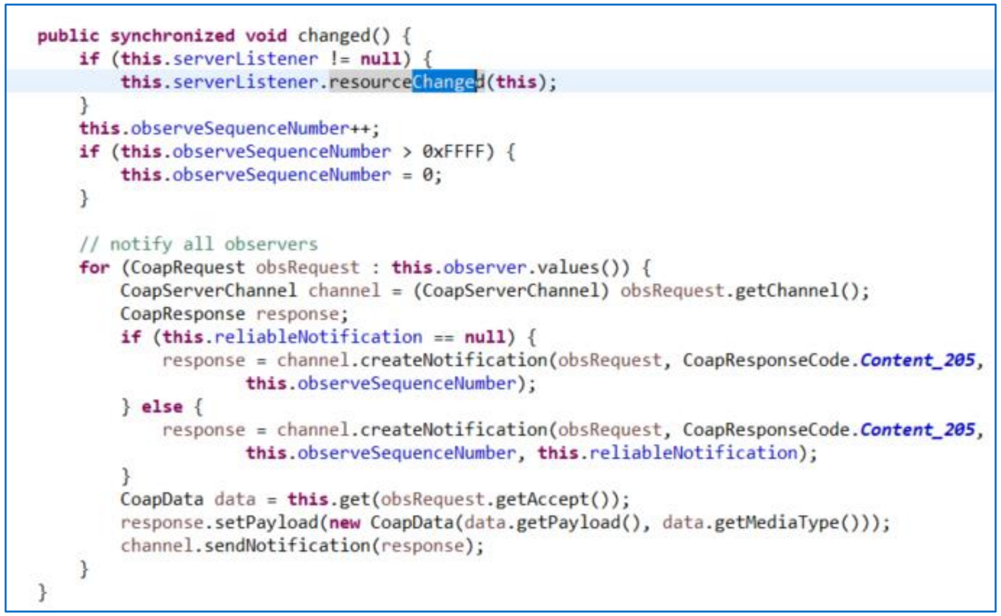


```java
    public synchronized void optional_changed() {
        String url = "https://kr.investing.com/crypto/bitcoin/"; //크롤링할 url지정
        Document doc = null;
        // 2.파싱
        try {
            doc = Jsoup.connect(url).get();
        } catch (IOException e) {
            e.printStackTrace();
        }
        // 3.원하는 부분 선택
        Elements element = doc.select(".inlineblock");
        element = element.select("#last_last");

        if(this.value.equals(element.text())) {
            System.out.println("Bitcoin value is not changed.");
        }
        else {
            this.value = element.text();
            this.changed(value);
        }
    }

    @Override
    public synchronized boolean put(byte[] data, CoapMediaType type) {
        return this.setValue(data);
    }


    @Override
    public synchronized boolean setValue(byte[] value) {
        this.value = Encoder.ByteToString(value);
        return true;
    }

    @Override
    public synchronized boolean post(byte[] data, CoapMediaType type) {
        return this.setValue(data);
    }

    @Override
    public synchronized String getResourceType() {
        return "LED";
    }

}
```

BasicCoapResource

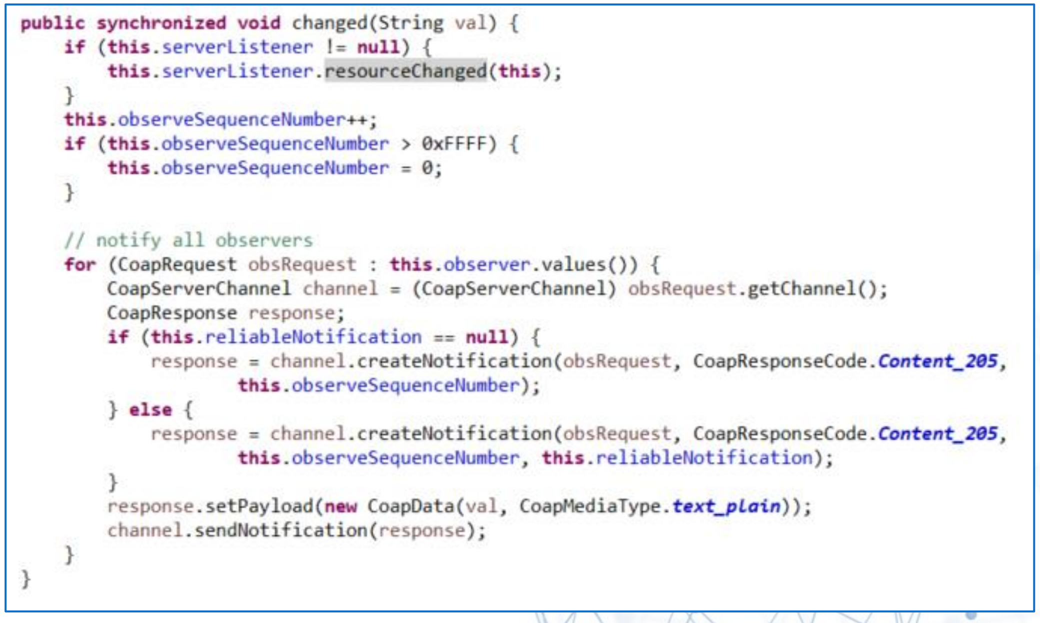
- BasicCoapResource

기존 method



```java
public synchronized void changed() {
    if (this.serverListener != null) {
        this.serverListener.resourceChanged(this);
    }
    this.observeSequenceNumber++;
    if (this.observeSequenceNumber > 0xFFFF) {
        this.observeSequenceNumber = 0;
    }

    // notify all observers
    for (CoapRequest obsRequest : this.observer.values()) {
        CoapServerChannel channel = (CoapServerChannel) obsRequest.getChannel();
        CoapResponse response;
        if (this.reliableNotification == null) {
            response = channel.createNotification(obsRequest, CoapResponseCode.Content_205,
                    this.observeSequenceNumber);
        } else {
            response = channel.createNotification(obsRequest, CoapResponseCode.Content_205,
                    this.observeSequenceNumber, this.reliableNotification);
        }
        CoapData data = this.get(obsRequest.getAccept());
        response.setPayload(new CoapData(data.getPayload(), data.getMediaType()));
        channel.sendNotification(response);
    }
}
```

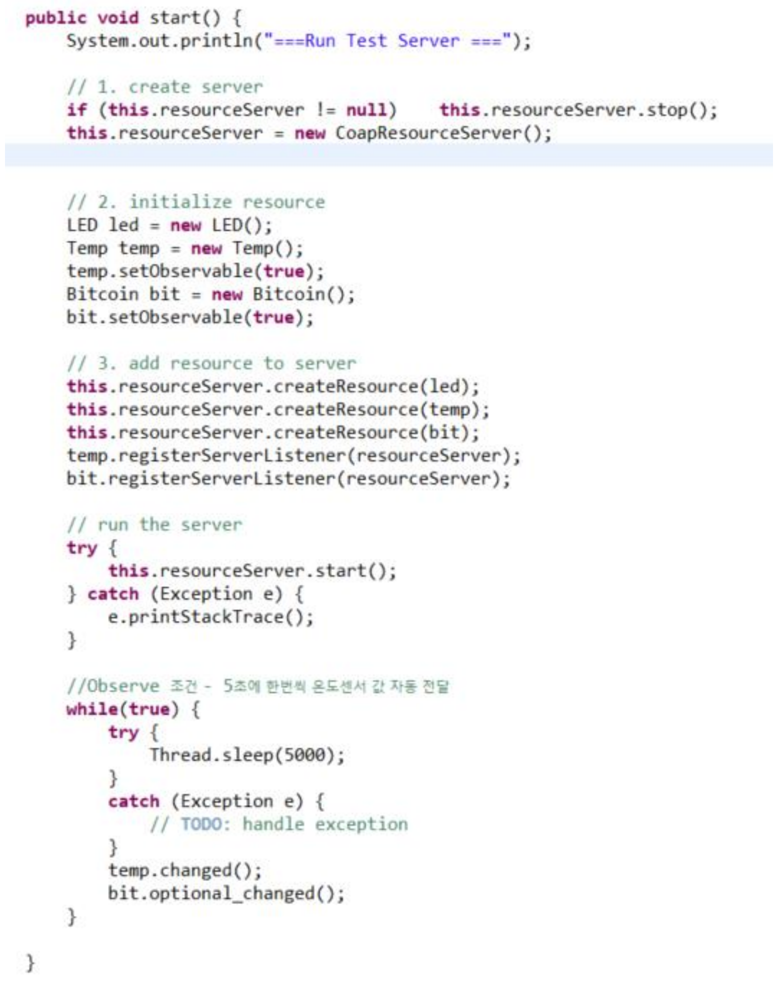
Overloading method



```java
public synchronized void changed(String val) {
    if (this.serverListener != null) {
        this.serverListener.resourceChanged(this);
    }
    this.observeSequenceNumber++;
    if (this.observeSequenceNumber > 0xFFFF) {
        this.observeSequenceNumber = 0;
    }

    // notify all observers
    for (CoapRequest obsRequest : this.observer.values()) {
        CoapServerChannel channel = (CoapServerChannel) obsRequest.getChannel();
        CoapResponse response;
        if (this.reliableNotification == null) {
            response = channel.createNotification(obsRequest, CoapResponseCode.Content_205,
                    this.observeSequenceNumber);
        } else {
            response = channel.createNotification(obsRequest, CoapResponseCode.Content_205,
                    this.observeSequenceNumber, this.reliableNotification);
        }
        response.setPayload(new CoapData(val, CoapMediaType.text_plain));
        channel.sendNotification(response);
    }
}
```

Server

- Server



```java
public void start() {
    System.out.println("===Run Test Server ===");

    // 1. create server
    if (this.resourceServer != null)    this.resourceServer.stop();
    this.resourceServer = new CoapResourceServer();


    // 2. initialize resource
    LED led = new LED();
    Temp temp = new Temp();
    temp.setObservable(true);
    Bitcoin bit = new Bitcoin();
    bit.setObservable(true);

    // 3. add resource to server
    this.resourceServer.createResource(led);
    this.resourceServer.createResource(temp);
    this.resourceServer.createResource(bit);
    temp.registerServerListener(resourceServer);
    bit.registerServerListener(resourceServer);

    // run the server
    try {
        this.resourceServer.start();
    } catch (Exception e) {
        e.printStackTrace();
    }

    //Observe 조건 - 5초에 한번씩 온도센서 값 자동 전달
    while(true) {
        try {
            Thread.sleep(5000);
        }
        catch (Exception e) {
            // TODO: handle exception
        }
        temp.changed();
        bit.optional_changed();
    }

}
```

Thank you