(IoT Network) Practice -5-

CoAP Method Option – 2

Index

I. Practice Overview

II. Bitcoin Price Parsing

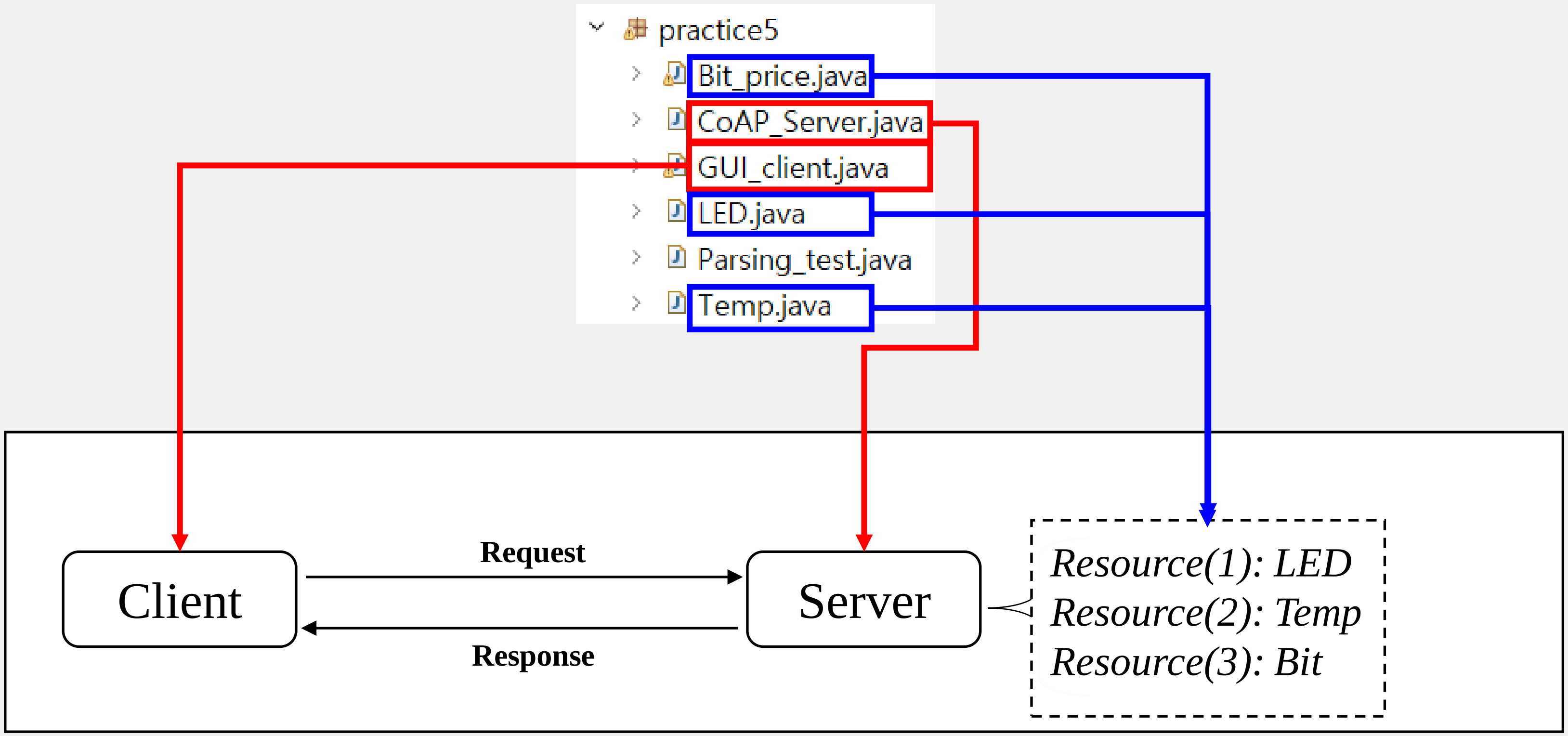
III. Bitcoin Price Resource

IV. Revising BasicCoapResource

V. Add Bitcoin Price Resource to Server

Practice Overview

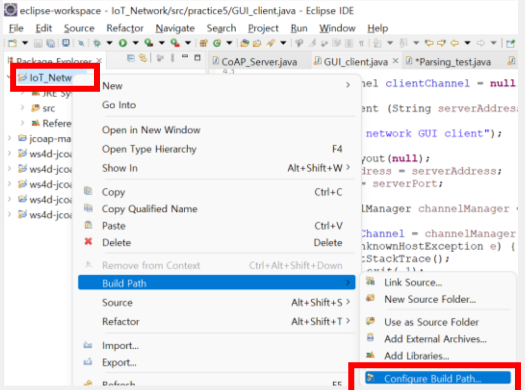
- **실습목표**
  1. Web Parsing을 사용해서 “**Bitcoin 가격**” Resource를 정의한다.
  2. “**Bitcoin 가격**” Resource의 Observe 기능을 구현한다.
  3. “**Bitcoin 가격**” Resource의 Observe는 Bitcoin의 가격이 변했을 때에만 Client에게 메시지를 전송한다.



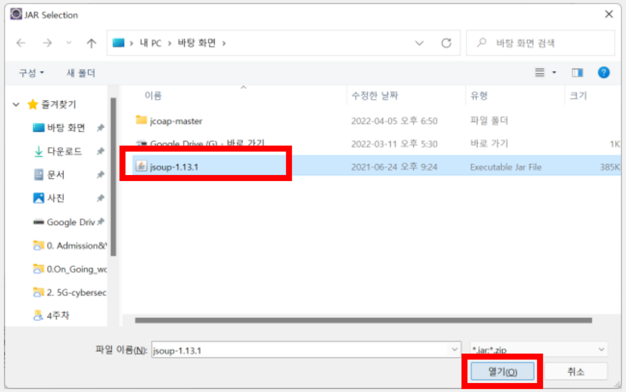
<Practice5 소스코드 구조>

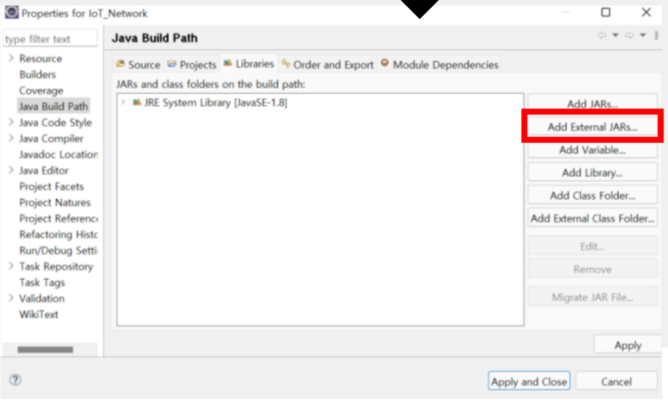
Bitcoin Price Parsing

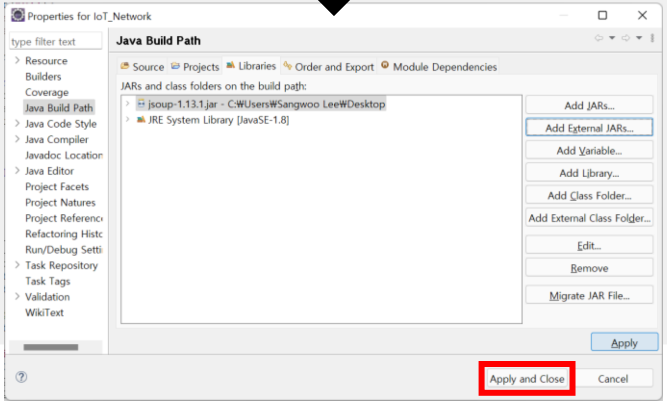
- **Jsoup**
  - Java HTML Parser
  - 즉, 웹 크롤링을 위한 자바 라이브러리
- **Jsoup Import**



```
nel clientChannel = null
ent (String serverAddres
network GUI client");
yout(null);
dress = serverAddress;
= serverPort;
lManager channelManager
Channel = channelManager
nknownHostException e) {
tStackTrace();
```

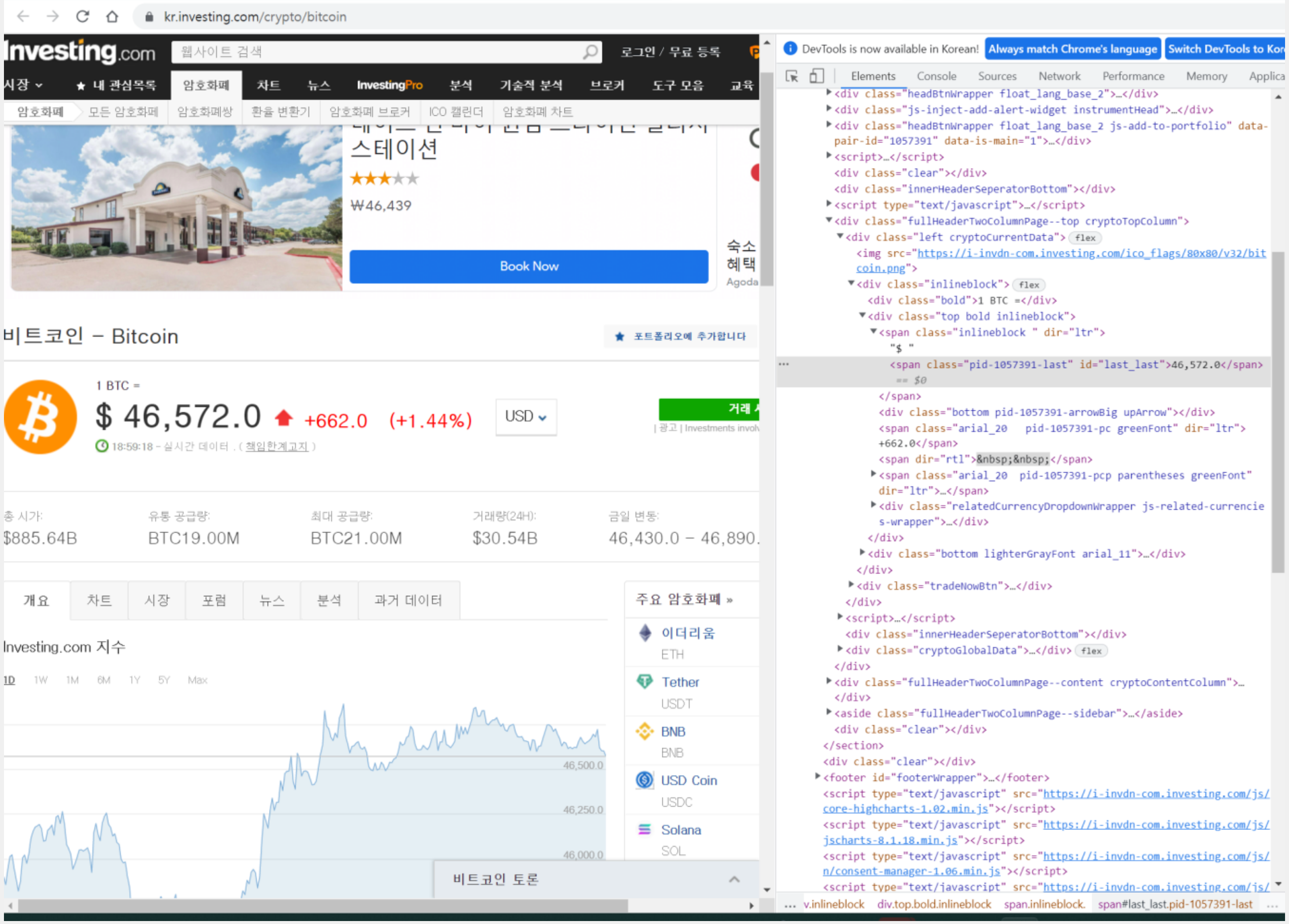






Bitcoin Price Parsing

- **Bitcoin 실시간 가격 Webpage**
  - https://kr.investing.com/crypto/bitcoin



비트코인 – Bitcoin

1 BTC =

$ 46,572.0 +662.0 (+1.44%)

18:59:18 - 실시간 데이터 . (책임한계고지)

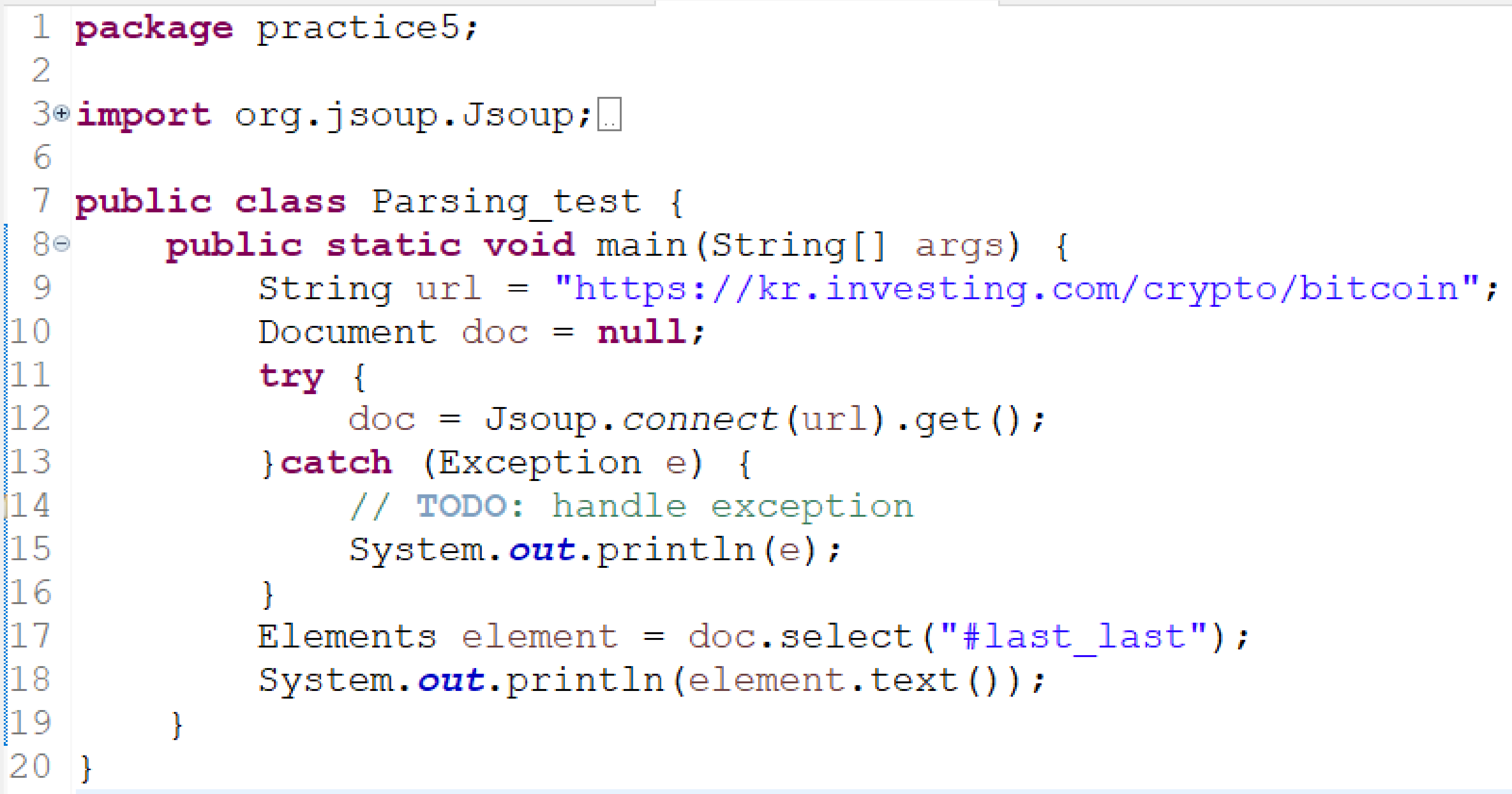
총 시가:	유통 공급량:	최대 공급량:	거래량(24H):	금일 변동:
$885.64B	BTC19.00M	BTC21.00M	$30.54B	46,430.0 – 46,890.

```
▸<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="1057391" 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"> flex
     <img src="https://i-invdn-com.investing.com/ico_flags/80x80/v32/bitcoin.png">
    ▾<div class="inlineblock"> flex
       <div class="bold">1 BTC =</div>
      ▾<div class="top bold inlineblock">
        ▾<span class="inlineblock " dir="ltr">
           "$ "
...        <span class="pid-1057391-last" id="last_last">46,572.0</span>
            == $0
         </span>
         <div class="bottom pid-1057391-arrowBig upArrow"></div>
         <span class="arial_20   pid-1057391-pc greenFont" dir="ltr">
         +662.0</span>
         <span dir="rtl">&nbsp;&nbsp;</span>
        ▸<span class="arial_20  pid-1057391-pcp parentheses greenFont"
         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> flex
 </div>
▸<div class="fullHeaderTwoColumnPage--content cryptoContentColumn">…
 </div>
▸<aside class="fullHeaderTwoColumnPage--sidebar">…</aside>
 <div class="clear"></div>
</section>
<div class="clear"></div>
▸<footer id="footerWrapper">…</footer>
<script type="text/javascript" src="https://i-invdn-com.investing.com/js/core-highcharts-1.02.min.js"></script>
<script type="text/javascript" src="https://i-invdn-com.investing.com/js/jscharts-8.1.18.min.js"></script>
<script type="text/javascript" src="https://i-invdn-com.investing.com/js/n/consent-manager-1.06.min.js"></script>
<script type="text/javascript" src="https://i-invdn-com.investing.com/js/
```

```
… v.inlineblock  div.top.bold.inlineblock  span.inlineblock.  span#last_last.pid-1057391-last  …
```

Bitcoin Price Parsing

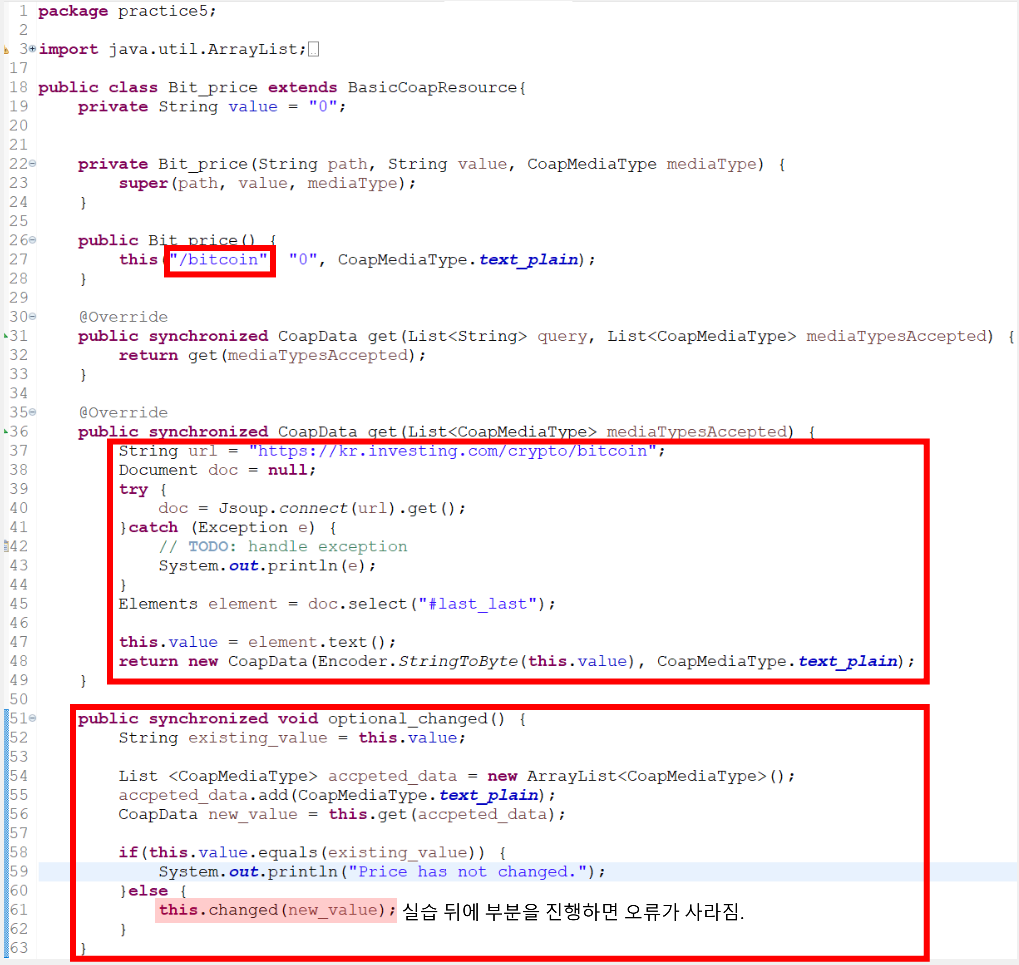
- **Bitcoin 가격 Parsing**



```java
package practice5;

import org.jsoup.Jsoup;

public class Parsing_test {
    public static void main(String[] args) {
        String url = "https://kr.investing.com/crypto/bitcoin";
        Document doc = null;
        try {
            doc = Jsoup.connect(url).get();
        }catch (Exception e) {
            // TODO: handle exception
            System.out.println(e);
        }
        Elements element = doc.select("#last_last");
        System.out.println(element.text());
    }
}
```

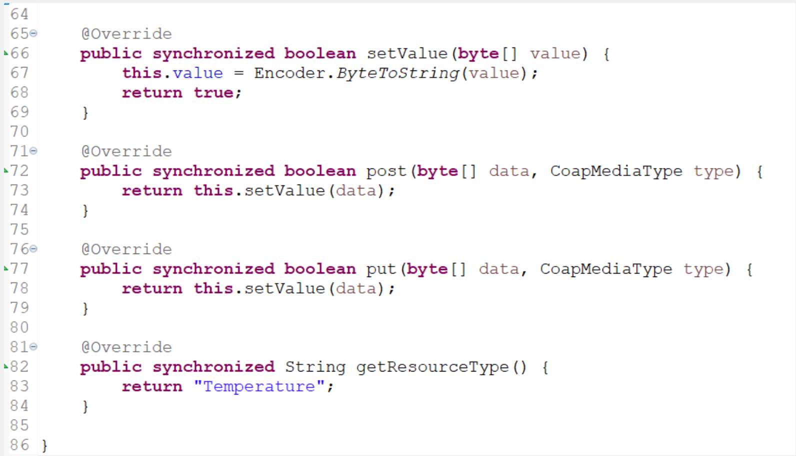
Bitcoin Price Resource

- **Bit_price.java**



```java
package practice5;

import java.util.ArrayList;□

public class Bit_price extends BasicCoapResource{
    private String value = "0";


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

    public Bit_price() {
        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) {
        String url = "https://kr.investing.com/crypto/bitcoin";
        Document doc = null;
        try {
            doc = Jsoup.connect(url).get();
        }catch (Exception e) {
            // TODO: handle exception
            System.out.println(e);
        }
        Elements element = doc.select("#last_last");

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

    public synchronized void optional_changed() {
        String existing_value = this.value;

        List <CoapMediaType> accpeted_data = new ArrayList<CoapMediaType>();
        accpeted_data.add(CoapMediaType.text_plain);
        CoapData new_value = this.get(accpeted_data);

        if(this.value.equals(existing_value)) {
            System.out.println("Price has not changed.");
        }else {
            this.changed(new_value);
        }
    }
```

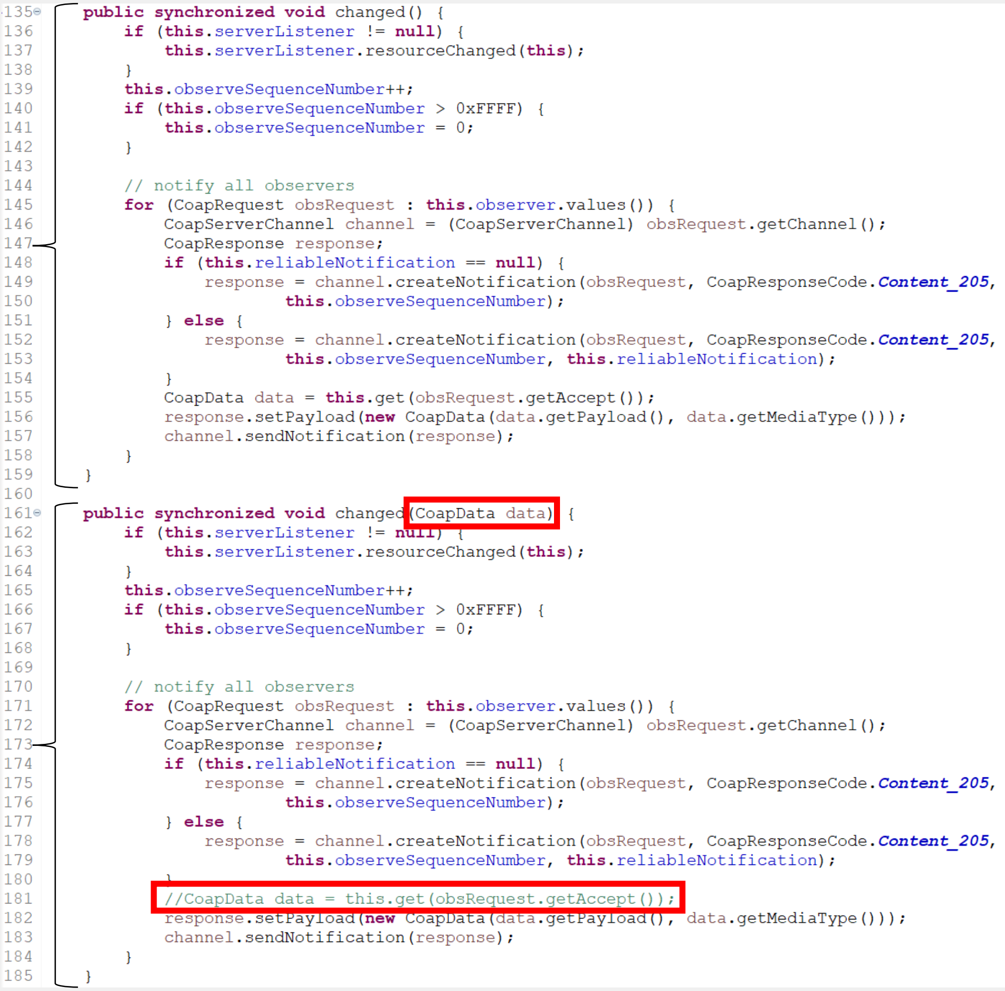
실습 뒤에 부분을 진행하면 오류가 사라짐.



```java

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

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

}
```

Revising BasicCoapResource

- **BasicCoapResource.java**



```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);
    }
}
```

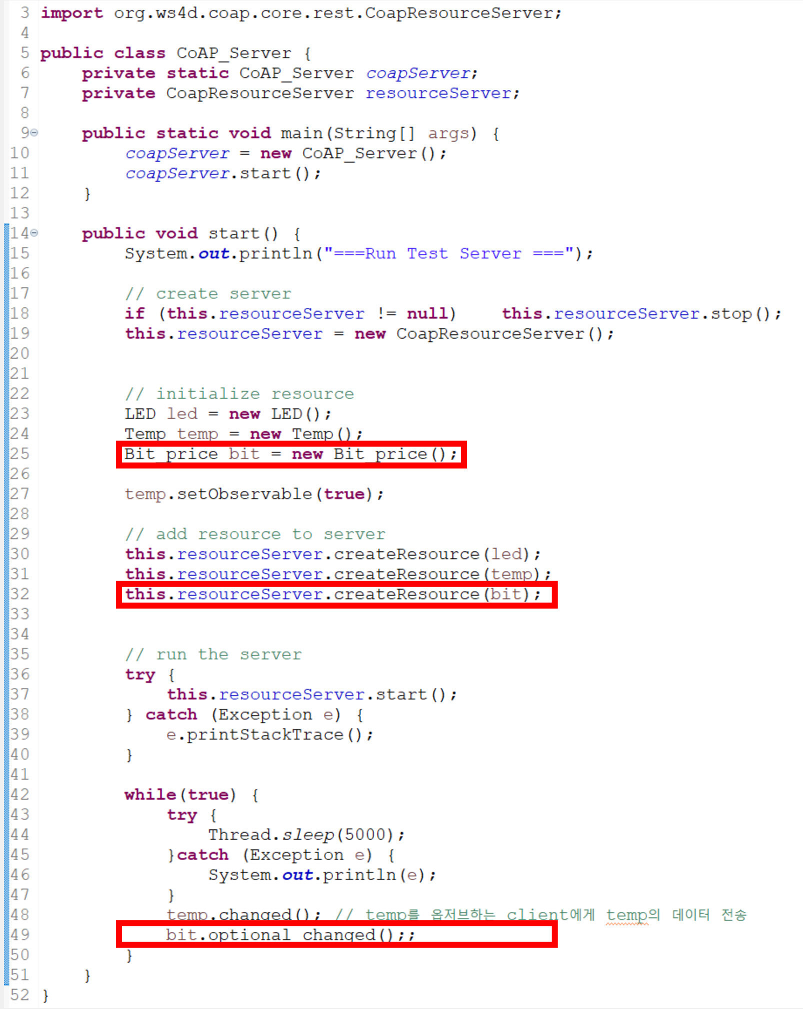
기존에 있던 Changed 메소드

```java
public synchronized void changed(CoapData data) {
    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);
    }
}
```

새로 추가하는 Changed 메소드
(Overloading)

Add Bitcoin Price Resource to Server

- **CoAP_Server.java**



```java
import org.ws4d.coap.core.rest.CoapResourceServer;

public class CoAP_Server {
    private static CoAP_Server coapServer;
    private CoapResourceServer resourceServer;

    public static void main(String[] args) {
        coapServer = new CoAP_Server();
        coapServer.start();
    }

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

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


        // initialize resource
        LED led = new LED();
        Temp temp = new Temp();
        Bit_price bit = new Bit_price();

        temp.setObservable(true);

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


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

        while(true) {
            try {
                Thread.sleep(5000);
            }catch (Exception e) {
                System.out.println(e);
            }
            temp.changed(); // temp를 옵저브하는 client에게 temp의 데이터 전송
            bit.optional_changed();;
        }
    }
}
```