

(IoT Network) Practice -5-

CoAP Method Option – 2



Index

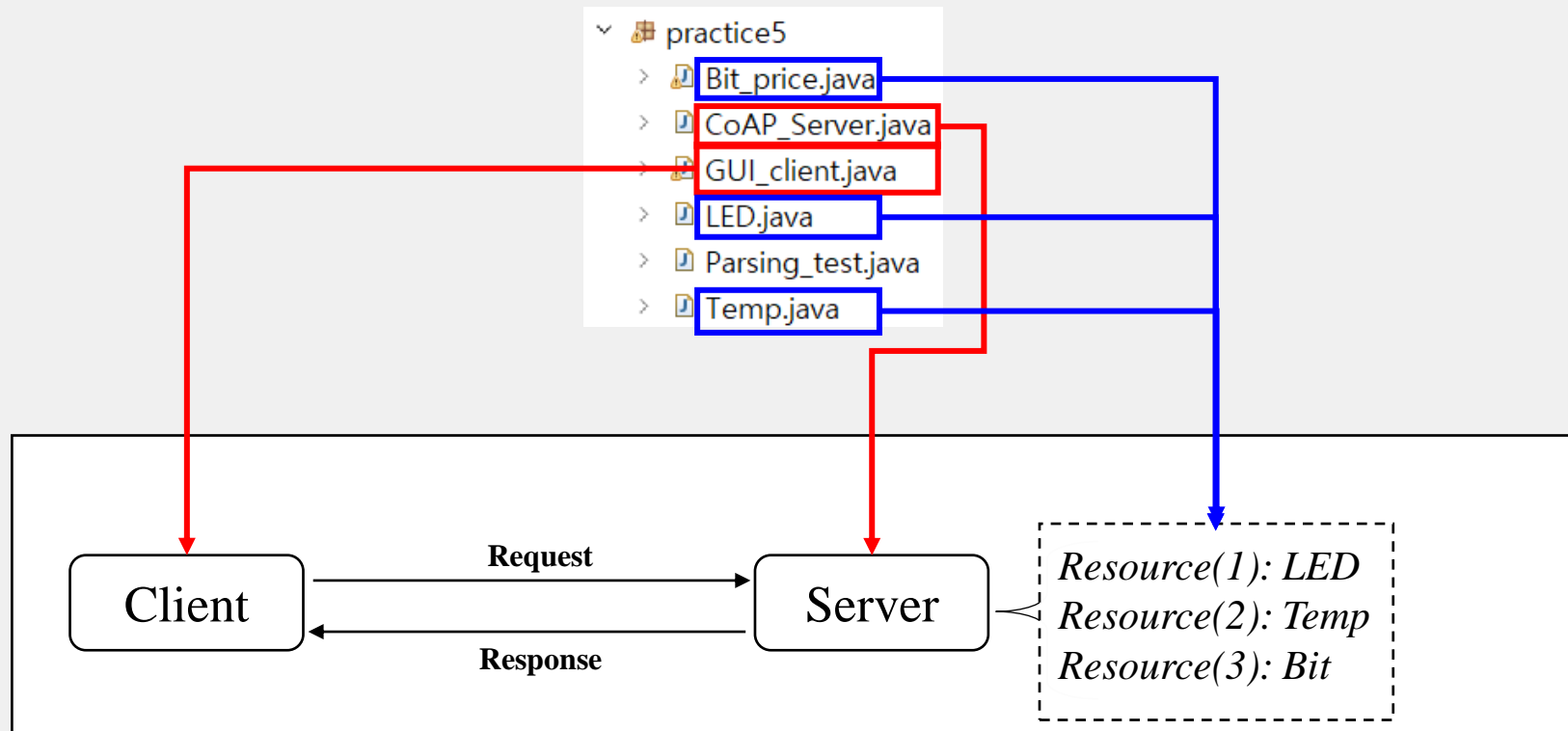
- I. Practice Overview
- II. Bitcoin Price Parsing
- III. Bitcoin Price Resource
- IV. Revising BasicCoapResource
- V. Add Bitcoin Price Resource to Server



Practice Overview

• 실습목표

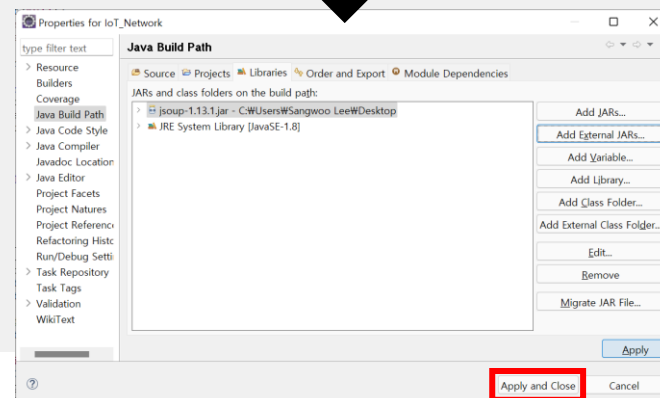
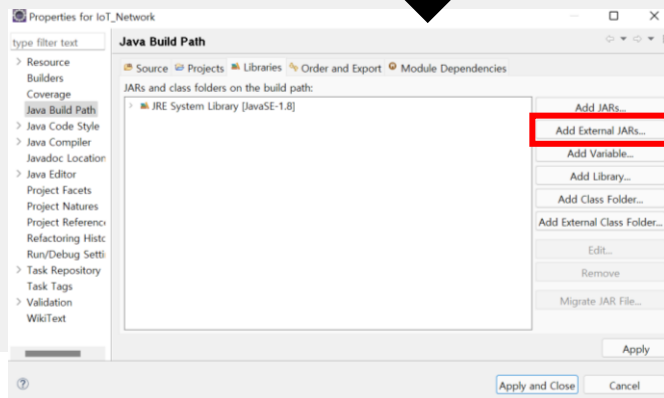
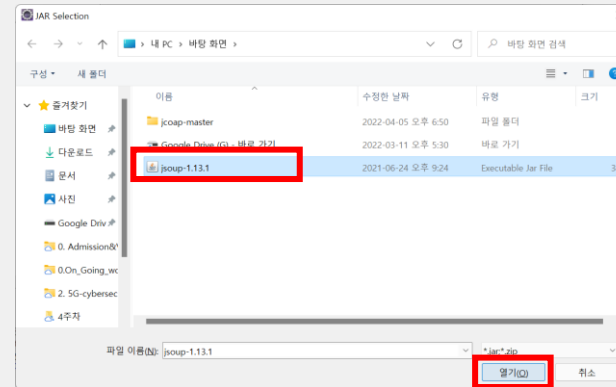
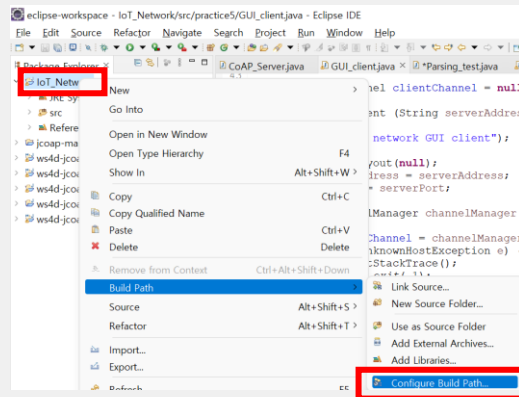
1. Web Parsing을 사용해서 “**Bitcoin 가격**” Resource를 정의한다.
2. “**Bitcoin 가격**” Resource의 Observe 기능을 구현한다.
3. “**Bitcoin 가격**” Resource의 Observe는 Bitcoin의 가격이 변했을 때에만 Client에게 메시지를 전송한다.



<Practice5 소스코드 구조>

Bitcoin Price Parsing

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



Bitcoin Price Parsing

- Bitcoin 실시간 가격 Webpage

- <https://kr.investing.com/crypto/bitcoin>

The screenshot displays the Investing.com website for Bitcoin prices. The main content area shows the current price of 1 BTC as \$46,572.0, with a green upward arrow indicating a price increase of +\$662.0 (+1.44%). Below the price, there are tabs for '개요' (Overview), '차트' (Charts), '시장' (Markets), '포럼' (Forums), '뉴스' (News), '분석' (Analysis), and '과거 데이터' (Historical Data). A line chart is visible under the '차트' tab, showing the price movement of Bitcoin over time. On the right side, there is a sidebar with '주요 암호화폐' (Major Cryptocurrencies) listed, including Ethereum (ETH), Tether (USDT), BNB, USD Coin (USDC), and Solana (SOL).

On the right side of the image, the Chrome DevTools 'Elements' panel is open, showing the DOM structure of the page. The selected element is a `span` with the class `pid-1057391-last` and the id `last_last`, containing the text `46,572.0`. The DOM tree shows the hierarchy of the page, including the `head`, `body`, and various `div` and `span` elements used for layout and styling.

Bitcoin Price Parsing

- Bitcoin 가격 Parsing

```
1 package practice5;
2
3 import org.jsoup.Jsoup;
4
5
6
7 public class Parsing_test {
8     public static void main(String[] args) {
9         String url = "https://kr.investing.com/crypto/bitcoin";
10        Document doc = null;
11        try {
12            doc = Jsoup.connect(url).get();
13        } catch (Exception e) {
14            // TODO: handle exception
15            System.out.println(e);
16        }
17        Elements element = doc.select("#last_last");
18        System.out.println(element.text());
19    }
20 }
```

Bitcoin Price Resource

- Bit_price.java

```
1 package practice5;
2
3 import java.util.ArrayList;
17
18 public class Bit_price extends BasicCoapResource{
19     private String value = "0";
20
21
22     private Bit_price(String path, String value, CoapMediaType mediaType) {
23         super(path, value, mediaType);
24     }
25
26     public Bit_price() {
27         this("/bitcoin", "0", CoapMediaType.TEXT_PLAIN);
28     }
29
30     @Override
31     public synchronized CoapData get(List<String> query, List<CoapMediaType> mediaTypesAccepted) {
32         return get(mediaTypesAccepted);
33     }
34
35     @Override
36     public synchronized CoapData get(List<CoapMediaType> mediaTypesAccepted) {
37         String url = "https://kr.investing.com/crypto/bitcoin";
38         Document doc = null;
39         try {
40             doc = Jsoup.connect(url).get();
41         } catch (Exception e) {
42             // TODO: handle exception
43             System.out.println(e);
44         }
45         Elements element = doc.select("#last_last");
46
47         this.value = element.text();
48         return new CoapData(Encoder.StringToByte(this.value), CoapMediaType.TEXT_PLAIN);
49     }
50
51     public synchronized void optional_changed() {
52         String existing_value = this.value;
53
54         List<CoapMediaType> accepted_data = new ArrayList<CoapMediaType>();
55         accepted_data.add(CoapMediaType.TEXT_PLAIN);
56         CoapData new_value = this.get(accepted_data);
57
58         if(this.value.equals(existing_value)) {
59             System.out.println("Price has not changed.");
60         } else {
61             this.changed(new_value); 실습 뒤에 부분을 진행하면 오류가 사라짐.
62         }
63     }
```

```
64
65     @Override
66     public synchronized boolean setValue(byte[] value) {
67         this.value = Encoder.ByteToString(value);
68         return true;
69     }
70
71     @Override
72     public synchronized boolean post(byte[] data, CoapMediaType type) {
73         return this.setValue(data);
74     }
75
76     @Override
77     public synchronized boolean put(byte[] data, CoapMediaType type) {
78         return this.setValue(data);
79     }
80
81     @Override
82     public synchronized String getResourceType() {
83         return "Temperature";
84     }
85
86 }
```

Revising BasicCoapResource

- BasicCoapResource.java

기존에 있던 Changed 메소드

```
135 public synchronized void changed() {
136     if (this.serverListener != null) {
137         this.serverListener.resourceChanged(this);
138     }
139     this.observeSequenceNumber++;
140     if (this.observeSequenceNumber > 0xFFFF) {
141         this.observeSequenceNumber = 0;
142     }
143
144     // notify all observers
145     for (CoapRequest obsRequest : this.observer.values()) {
146         CoapServerChannel channel = (CoapServerChannel) obsRequest.getChannel();
147         CoapResponse response;
148         if (this.reliableNotification == null) {
149             response = channel.createNotification(obsRequest, CoapResponseCode.Content_205,
150                 this.observeSequenceNumber);
151         } else {
152             response = channel.createNotification(obsRequest, CoapResponseCode.Content_205,
153                 this.observeSequenceNumber, this.reliableNotification);
154         }
155         CoapData data = this.get(obsRequest.getAccept());
156         response.setPayload(new CoapData(data.getPayload(), data.getMediaType()));
157         channel.sendNotification(response);
158     }
159 }
```

새로 추가하는 Changed 메소드
(Overloading)

```
161 public synchronized void changed(CoapData data) {
162     if (this.serverListener != null) {
163         this.serverListener.resourceChanged(this);
164     }
165     this.observeSequenceNumber++;
166     if (this.observeSequenceNumber > 0xFFFF) {
167         this.observeSequenceNumber = 0;
168     }
169
170     // notify all observers
171     for (CoapRequest obsRequest : this.observer.values()) {
172         CoapServerChannel channel = (CoapServerChannel) obsRequest.getChannel();
173         CoapResponse response;
174         if (this.reliableNotification == null) {
175             response = channel.createNotification(obsRequest, CoapResponseCode.Content_205,
176                 this.observeSequenceNumber);
177         } else {
178             response = channel.createNotification(obsRequest, CoapResponseCode.Content_205,
179                 this.observeSequenceNumber, this.reliableNotification);
180         }
181         //CoapData data = this.get(obsRequest.getAccept());
182         response.setPayload(new CoapData(data.getPayload(), data.getMediaType()));
183         channel.sendNotification(response);
184     }
185 }
```


Add Bitcoin Price Resource to Server

- CoAP_Server.java

```
3 import org.ws4d.coap.core.rest.CoapResourceServer;
4
5 public class CoAP_Server {
6     private static CoAP_Server coapServer;
7     private CoapResourceServer resourceServer;
8
9     public static void main(String[] args) {
10         coapServer = new CoAP_Server();
11         coapServer.start();
12     }
13
14     public void start() {
15         System.out.println("===Run Test Server ===");
16
17         // create server
18         if (this.resourceServer != null) this.resourceServer.stop();
19         this.resourceServer = new CoapResourceServer();
20
21         // initialize resource
22         LED led = new LED();
23         Temp temp = new Temp();
24         Bit price bit = new Bit price();
25
26         temp.setObservable(true);
27
28         // add resource to server
29         this.resourceServer.createResource(led);
30         this.resourceServer.createResource(temp);
31         this.resourceServer.createResource(bit);
32
33         // run the server
34         try {
35             this.resourceServer.start();
36         } catch (Exception e) {
37             e.printStackTrace();
38         }
39
40         while(true) {
41             try {
42                 Thread.sleep(5000);
43             } catch (Exception e) {
44                 System.out.println(e);
45             }
46             temp.changed(); // temp를 모니터링하는 client에게 temp의 데이터 전송
47             bit.optional changed();
48         }
49     }
50 }
51
52 }
```