HALLYM UNIVERSITY

CIC Convergence Information & Communications

(IoT Network) Practice -7-

CoAP Blockwise Transfer

Index

I. Practice Overview

II. Adding “Large Resource”

III. Revising Jcoap

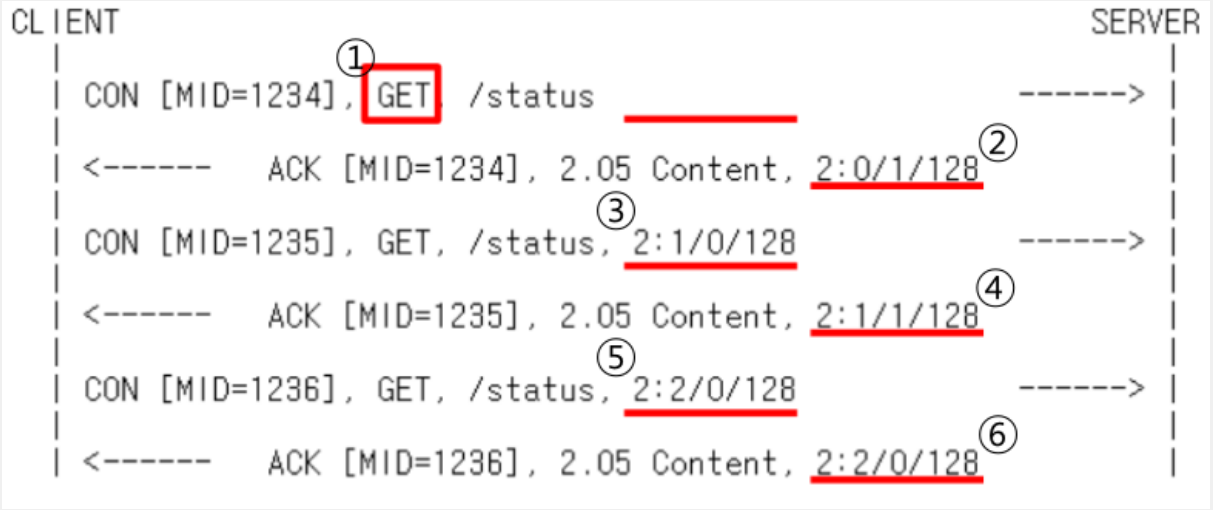
IV. Revising GUI Client

- Blockwise Transfer(BT) Button

V. Printing Blockwise Transfer Process

Practice Overview

- 실습목표
  - Blockwise Transfer Option 2 구현
    - ① (Client → Server) Bockwise Transfer Option2 요청
    - ② (Server → Client) Piggybacking으로 0번 블록 전달 **[전송하는 블록번호, 남은 블록(Y/N), 블록 사이즈]**
    - ③ (Client → Server) 1번 블록 요청 **[요청하는 블록번호, 0 (고정), 블록 사이즈]**
    - ④ (Server → Client) Piggybacking으로 1번 블록 전달 **[1, 1 (Yes), 128]**
    - ⑤ (Client → Server) 2번 블록 요청 **[2, 0, 128]**
    - ⑥ (Server → Client) Piggybacking으로 세 번째 블록 전달 **[2, 0 (No), 128]**



Multicast practice scenario

Adding “Large Resource”

- **Large.java**
  - LED Resource “copy and paste” → 클래스 이름 및 생성자 수정

```
public class Large  extends BasicCoapResource{
    private String state = "largeData";


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

    public Large() {
        this("/large", "largeData", CoapMediaType.text_plain);
        //랜덤 글자 생성
        int leftLimit = 97; // letter 'a'
        int rightLimit = 122; // letter 'z'
        int targetStringLength = 100; // 글자 수
        Random random = new Random();
        StringBuilder buffer = new StringBuilder(targetStringLength);
        for (int i = 0; i < targetStringLength; i++) {
            int randomLimitedInt = leftLimit + (int)
              (random.nextFloat() * (rightLimit - leftLimit + 1));
            buffer.append((char) randomLimitedInt);
        }
        String generatedString = buffer.toString();
        this.setValue(Encoder.StringToByte(generatedString));
    }
```

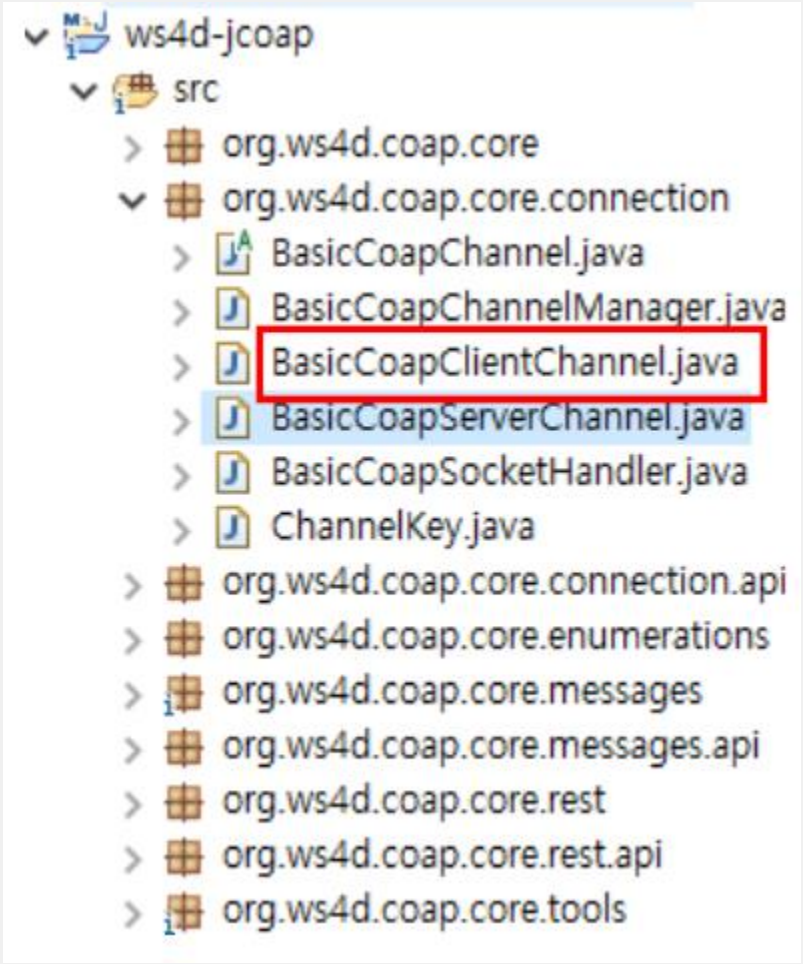
Adding “Large Resource”

- **CoAP_Server.java**

```
    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();
        Large large = new Large();

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



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

    }
}
```

Revising Jcoap

- **BasicCoapClientChannel.java**



Revising Jcoap

- **BasicCoapClientChannel.java**

기존

```
140             /* blockwise transfer finished */
141
142             message.setPayload(new CoapData(this.blockContext.getPayload(), message.getContentType()));
143             this.blockContext = null;
144         }
145
146         /* normal or separate response */
147         this.client.onResponse(this, (BasicCoapResponse) message);
148     }
```

수정

```
140             /* blockwise transfer finished */
141
142             message.setPayload(new CoapData(this.blockContext.getPayload(), this.lastRequest.getContentType()));
143             this.blockContext = null;
144         }
145
146         /* normal or separate response */
147         this.client.onResponse(this, (BasicCoapResponse) message);
148     }
```

Revising GUI Client

- GUI_client.java

```
public class GUI_client extends JFrame implements CoapClient{
    private static final boolean exitAfterResponse = false;
    JButton btn_get = new JButton("GET");
    JButton btn_post = new JButton("POST");
    JButton btn_put = new JButton("PUT");
    JButton btn_delete = new JButton("DELETE");
    JButton btn_obsget = new JButton("Observe GET");
    JButton btn_BTget = new JButton("BT GET");
```

```
//btn
btn_get.setBounds(20, 670, 100, 50);
btn_put.setBounds(130, 670, 100, 50);
btn_post.setBounds(240, 670, 100, 50);
btn_delete.setBounds(350, 670, 100, 50);
btn_obsget.setBounds(460,  670, 130, 50);
btn_BTget.setBounds(600, 670, 100, 50);
```

```
this.add(btn_get);
this.add(btn_post);
this.add(btn_put);
this.add(btn_delete);
this.add(btn_obsget);
this.add(btn_BTget);
this.add(path_text);
this.add(path_label);
this.add(payload_label);
this.add(payload_text);
this.add(display_text_jp);
this.add(display_label);
```

Revising GUI Client

- **GUI_client.java**

```
btn_BTget.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        String path = path_text.getText();
        String payload = payload_text.getText();
        CoapRequest request = clientChannel.createRequest(CoapRequestCode.GET, path, true);
        clientChannel.setMaxReceiveBlocksize(CoapBlockSize.BLOCK_16);
        CoapBlockOption block2 = new CoapBlockOption(0, true, clientChannel.getMaxReceiveBlocksize());
        request.setBlock2(block2);
        request.setContentType(CoapMediaType.text_plain);
        displayRequest(request);
        clientChannel.sendMessage(request);
    }
});
```

Printing Blockwise Transfer Process

- GUI_client.java

```
btn_BTget.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        String path = path_text.getText();
        String payload = payload_text.getText();
        CoapRequest request = clientChannel.createRequest(CoapRequestCode.GET, path, true);
        clientChannel.setMaxReceiveBlocksize(CoapBlockSize.BLOCK_16);
        CoapBlockOption block2 = new CoapBlockOption(0, true, clientChannel.getMaxReceiveBlocksize());
        request.setBlock2(block2);
        request.setContentType(CoapMediaType.text_plain);
        displayRequest(request);
        System.out.println(request);
        System.out.printf("  Block Options: %d / %b / %d\n", block2.getNumber(), !block2.isLast(), block2.getBlockSize().getSize());
        clientChannel.sendMessage(request);
    }
});
```

Printing Blockwise Transfer Process

- GUI_client.java

```
public void onResponse(CoapClientChannel channel, CoapResponse response) {
    if (response.getPayload() != null) {
        display_text.append(
                "Response: " + response.toString() + " payload: " + Encoder.ByteToString(response.getPayload()));
        display_text.setCaretPosition(display_text.getDocument().getLength());
    } else {
        display_text.append("Response: " + response.toString());
        display_text.setCaretPosition(display_text.getDocument().getLength());
    }
    if (GUI_client.exitAfterResponse) {
        display_text.append("===END===");
        System.exit(0);
    }
    if (response.getBlock2()!= null) {
        System.out.println(response);
        System.out.printf("  Block Options: %d/%b/%d%n", response.getBlock2().getNumber(), !response.getBlock2().isLast(), response.getBlock2().getBlockSize().getSize());
        System.out.println("  Payload: " + Encoder.ByteToString(response.getPayload()));
    }
    display_text.append(System.lineSeparator());
    display_text.append("*");
    display_text.append(System.lineSeparator());
}
```

Printing Blockwise Transfer Process

- **BasicCoapClientChannel.java**

```
126         /* create a new request for the next block */
127         BasicCoapRequest request = new BasicCoapRequest(this.lastRequest.getPacketType(),
128                 this.lastRequest.getRequestCode(), this.getSocketHandler().getChannelManager().getNewMessageID());
129         request.copyHeaderOptions((BasicCoapRequest) this.blockContext.getFirstRequest());
130         request.setToken(this.blockContext.getFirstRequest().getToken());
131         if (request.getRequestCode() == CoapRequestCode.GET) {
132             request.setBlock2(newBlock);
133         } else {
134             request.setBlock1(newBlock);
135             request.setPayload(this.blockContext.getNextPayload(newBlock));
136         }
137         System.out.println(message);
138         System.out.printf("  Block Options: %d/%b/%d%n", message.getBlock2().getNumber(), !message.getBlock2().isLast(), message.getBlock2().getBlockSize().getSize());
139         System.out.println("  Payload: " + Encoder.ByteToString(message.getPayload()));
140         System.out.println("-------------------------------------------------------");
141         System.out.println(request);
142         System.out.printf("  Block Options: %d/%b/%d%n", request.getBlock2().getNumber(), !request.getBlock2().isLast(), request.getBlock2().getBlockSize().getSize());
143         sendMessage(request);
144     }
145     return;
146 }
```