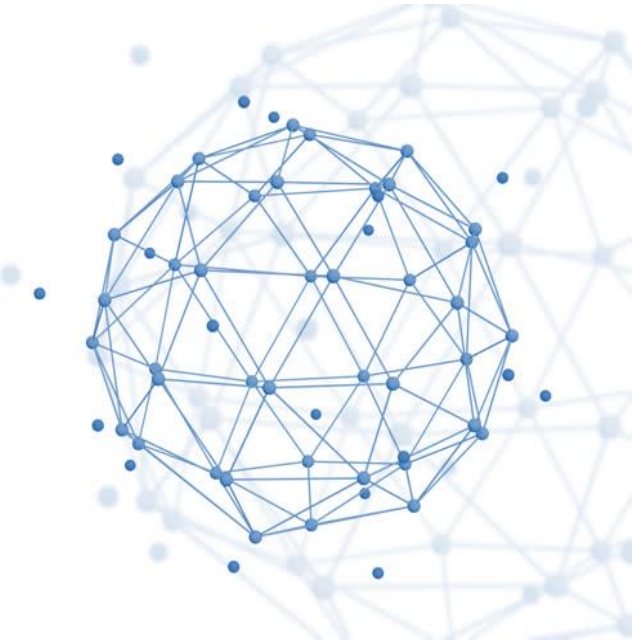
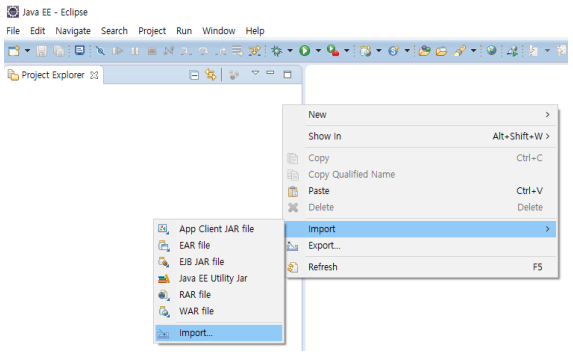
Contents

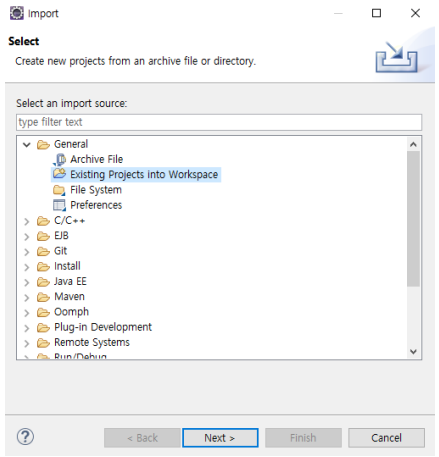
- **Implementation**
  - **jCoAP Server**
  - **jCoAP Client**
    - ✓ **Resource Discovery**
    - ✓ **GET**
    - ✓ **POST**
    - ✓ **Observe**
    - ✓ **UI**
- **Conclusion**



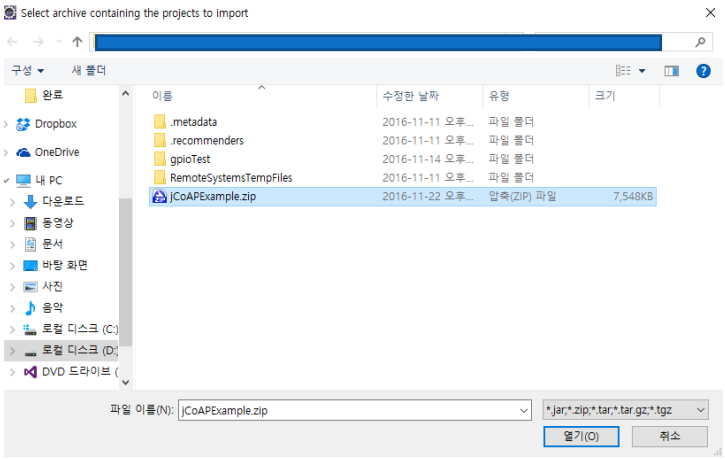
jCoAP project import(1/2)

- import - import … - General - Existing Project into Workspace



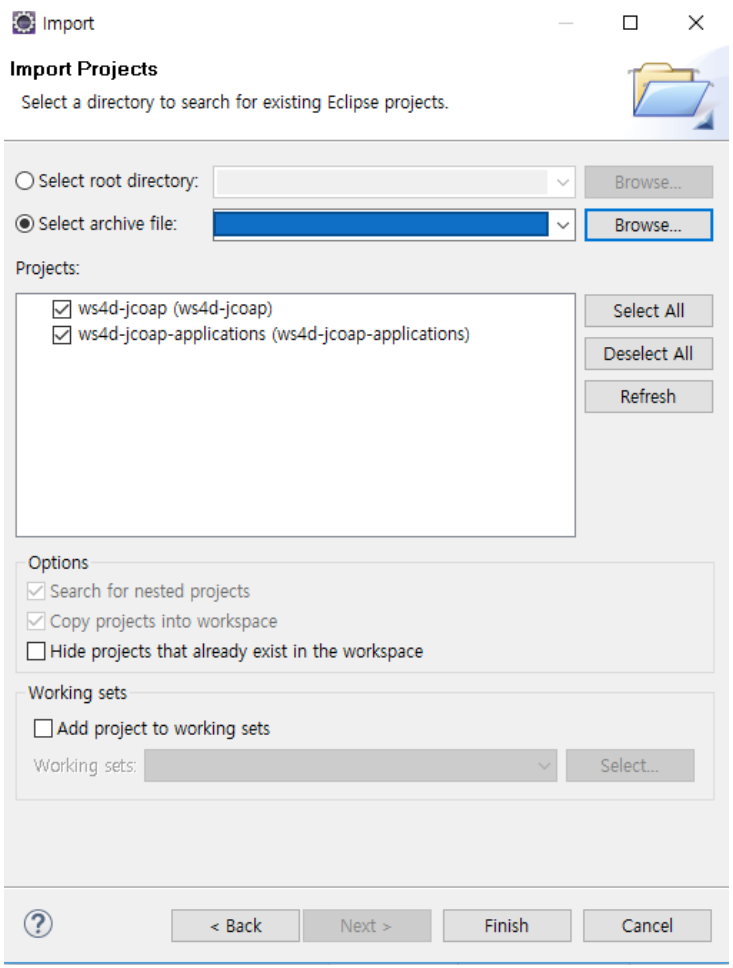


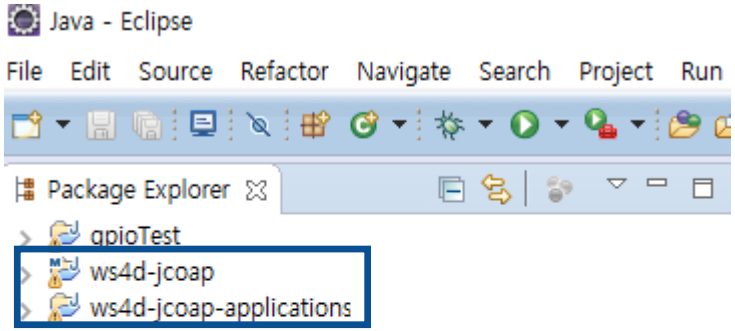
- Select archive file



jCoAP project import(2/2)

- Error : eclipse maven plugin installation





CoAP Introduction

No.	C	U	N	R	Name	Format	Length	Default
1	x			x	If-Match	opaque	0-8	(none)
3	x	x	-		Uri-Host	string	1-255	(see below)
4				x	ETag	opaque	1-8	(none)
5	x				If-None-Match	empty	0	(none)
7	x	x	-		Uri-Port	uint	0-2	(see below)
8				x	Location-Path	string	0-255	(none)
11	x	x	-	x	Uri-Path	string	0-255	(none)
12					Content-Format	uint	0-2	(none)
14		x	-		Max-Age	uint	0-4	60
15	x	x	-	x	Uri-Query	string	0-255	(none)
17	x				Accept	uint	0-2	(none)
20				x	Location-Query	string	0-255	(none)
35	x	x	-		Proxy-Uri	string	1-1034	(none)
39	x	x	-		Proxy-Scheme	string	1-255	(none)
60			x		Size1	uint	0-4	(none)

C=Critical, U=Unsafe, N=NoCacheKey, R=Repeatable

Table 4: Options

RFC 7252 (Option code)

```
public static CoapHeaderOptionType parse(int optionTypeValue){
    switch(optionTypeValue){
    case 1: return If_Match;
    case 3: return Uri_Host;
    case 4: return Etag;
    case 5: return If_None_Match;
    case 6: return Observe;
    case 7: return Uri_Port;
    case 8: return Location_Path;
    case 11:return Uri_Path;
    case 12:return Content_Format;
    case 14:return Max_Age;
    case 15:return Uri_Query;
    case 17:return Accept;
    case 20:return Location_Query;
    case 23:return Block2;
    case 27:return Block1;
    case 28:return Size;
    case 35:return Proxy_Uri;
    case 39:return Proxy_Scheme;
    case 60:return Size1;
        default: return UNKNOWN;
    }
}
```

AbstractCoapMessage.java(Option code)

CoAP Introduction

Code	Name	Reference
0.01	GET	[RFC7252]
0.02	POST	[RFC7252]
0.03	PUT	[RFC7252]
0.04	DELETE	[RFC7252]

Table 5: CoAP Method Codes

RFC 7252 (Method code)

```
public enum CoapRequestCode {
    GET(1),
    POST(2),
    PUT(3),
    DELETE(4);
```

CoaprequestCode.java(Method code)

CoAP Introduction

Code	Description	Reference
2.01	Created	[RFC7252]
2.02	Deleted	[RFC7252]
2.03	Valid	[RFC7252]
2.04	Changed	[RFC7252]
2.05	Content	[RFC7252]
4.00	Bad Request	[RFC7252]
4.01	Unauthorized	[RFC7252]
4.02	Bad Option	[RFC7252]
4.03	Forbidden	[RFC7252]
4.04	Not Found	[RFC7252]
4.05	Method Not Allowed	[RFC7252]
4.06	Not Acceptable	[RFC7252]
4.12	Precondition Failed	[RFC7252]
4.13	Request Entity Too Large	[RFC7252]
4.15	Unsupported Content-Format	[RFC7252]
5.00	Internal Server Error	[RFC7252]
5.01	Not Implemented	[RFC7252]
5.02	Bad Gateway	[RFC7252]
5.03	Service Unavailable	[RFC7252]
5.04	Gateway Timeout	[RFC7252]
5.05	Proxying Not Supported	[RFC7252]

Table 6: CoAP Response Codes

RFC 7252 (Response code)

```java
public static CoapResponseCode parseResponseCode(int codeValue) {
    switch (codeValue) {
    /* 32..63: reserved */
    //Success 2.xx
    case 65:  return Created_201;
    case 66:  return Deleted_202;
    case 67:  return Valid_203;
    case 68:  return Changed_204;
    case 69:  return Content_205;
    case 95:  return Continue_231;

    //Client Error 4.xx
    case 128: return Bad_Request_400;
    case 129: return Unauthorized_401;
    case 130: return Bad_Option_402;
    case 131: return Forbidden_403;
    case 132: return Not_Found_404;
    case 133: return Method_Not_Allowed_405;
    case 134: return Not_Acceptable_406;
    case 140: return Precondition_Failed_412;
    case 141: return Request_Entity_To_Large_413;
    case 143: return Unsupported_Media_Type_415;

    //Server Error 5.xx
    case 160: return Internal_Server_Error_500;
    case 161: return Not_Implemented_501;
    case 162: return Bad_Gateway_502;
    case 163: return Service_Unavailable_503;
    case 164: return Gateway_Timeout_504;
    case 165: return Proxying_Not_Supported_505;
    default:
```

CoapResponseCode.java(Response code)

jCoAP Server

- ws4d-jcoap-applications - org.ws4d.coap.server – CoapSampleResourceServer.java

```
private void run() throws IOException
{
    if (resourceServer != null)
        resourceServer.stop();
    resourceServer = new CoapResourceServer();

    /* Show detailed logging of Resource Server*/
    Logger resourceLogger = Logger.getLogger(CoapResourceServer.class.getName());
    resourceLogger.setLevel(Level.ALL);

    /* add resources */
    BasicCoapResource light = new BasicCoapResource("/test/light", "light Actuator".getBytes(), CoapMediaType.text_plain);
    light.registerResourceHandler(new ResourceHandler() {
        @Override
        public void onPost(byte[] data) {
            System.out.println("Post to /test/light");
            light.setValue(data);
            light.changed();
        }
    });
    light.setObservable(false);
    resourceServer.createResource(light);
    try {
        resourceServer.start();
    } catch (Exception e) {
        e.printStackTrace();
    }

    //observe example
    /*
    int counter = 0;
    while(true){
        try {
            Thread.sleep(5000);
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        counter++;
        light.setValue(((String)"Message #" + counter).getBytes());
        light.changed();
    }
    */
}
```

jCoAP Client

- ws4d-jcoap-applications - org.ws4d.coap.client – BasicCoapClient.java

```
public static void main(String[] args)
{
    System.out.println("Start CoAP Client");
    //String serverIp = "127.0.0.1";
    String serverIp = "raspberrypi.mshome.net";
    //Constants.COAP_DEFAULT_PORT (5683)
    BasicCoapClient client = new BasicCoapClient(serverIp, Constants.COAP_DEFAULT_PORT);
    client.channelManager = BasicCoapChannelManager.getInstance();

    //UI
    client.clientUi();

    //example
    //client.resourceDiscoveryExample();
    //client.getExample();
    //client.postExample();
    client.observeExample();
}
```

jCoAP Client

- Resource Discovery

```
public void resourceDiscoveryExample()
{
    try {
        clientChannel = (BasicCoapClientChannel) channelManager.connect(this, InetAddress.getByName(SERVER_ADDRESS), PORT);
        CoapRequest coapRequest = clientChannel.createRequest(true, CoapRequestCode.GET);
        byte [] token = generateRequestToken(3);
        coapRequest.setUriPath("/.well-known/core");
        coapRequest.setToken(token);
        clientChannel.sendMessage(coapRequest);
    } catch (UnknownHostException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }
}
```

- GET

```
public void getExample()
{
    try {
        clientChannel = (BasicCoapClientChannel) channelManager.connect(this, InetAddress.getByName(SERVER_ADDRESS), PORT);
        CoapRequest coapRequest = clientChannel.createRequest(true, CoapRequestCode.GET);
        byte [] token = generateRequestToken(3);
        coapRequest.setUriPath("/test/light");
        coapRequest.setToken(token);
        clientChannel.sendMessage(coapRequest);
    } catch (UnknownHostException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }
}
```

jCoAP Client

- POST

```
public void postExample()
{
    try {
        clientChannel = (BasicCoapClientChannel) channelManager.connect(this, InetAddress.getByName(SERVER_ADDRESS), PORT);
        CoapRequest coapRequest = clientChannel.createRequest(true, CoapRequestCode.POST);
        byte [] token = generateRequestToken(3);
        coapRequest.setUriPath("/test/light");
        coapRequest.setToken(token);
        coapRequest.setPayload("lightTest".getBytes());
        clientChannel.sendMessage(coapRequest);
    } catch (UnknownHostException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }
}
```

- Observe

```
public void observeExample()
{
    try {
        clientChannel = (BasicCoapClientChannel) channelManager.connect(this, InetAddress.getByName(SERVER_ADDRESS), PORT);
        CoapRequest coapRequest = clientChannel.createRequest(true, CoapRequestCode.GET);
        byte [] token = generateRequestToken(3);
        coapRequest.setUriPath("/test/light");
        coapRequest.setToken(token);
        coapRequest.setObserveOption(1);
        clientChannel.sendMessage(coapRequest);
    } catch (UnknownHostException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }

}
```

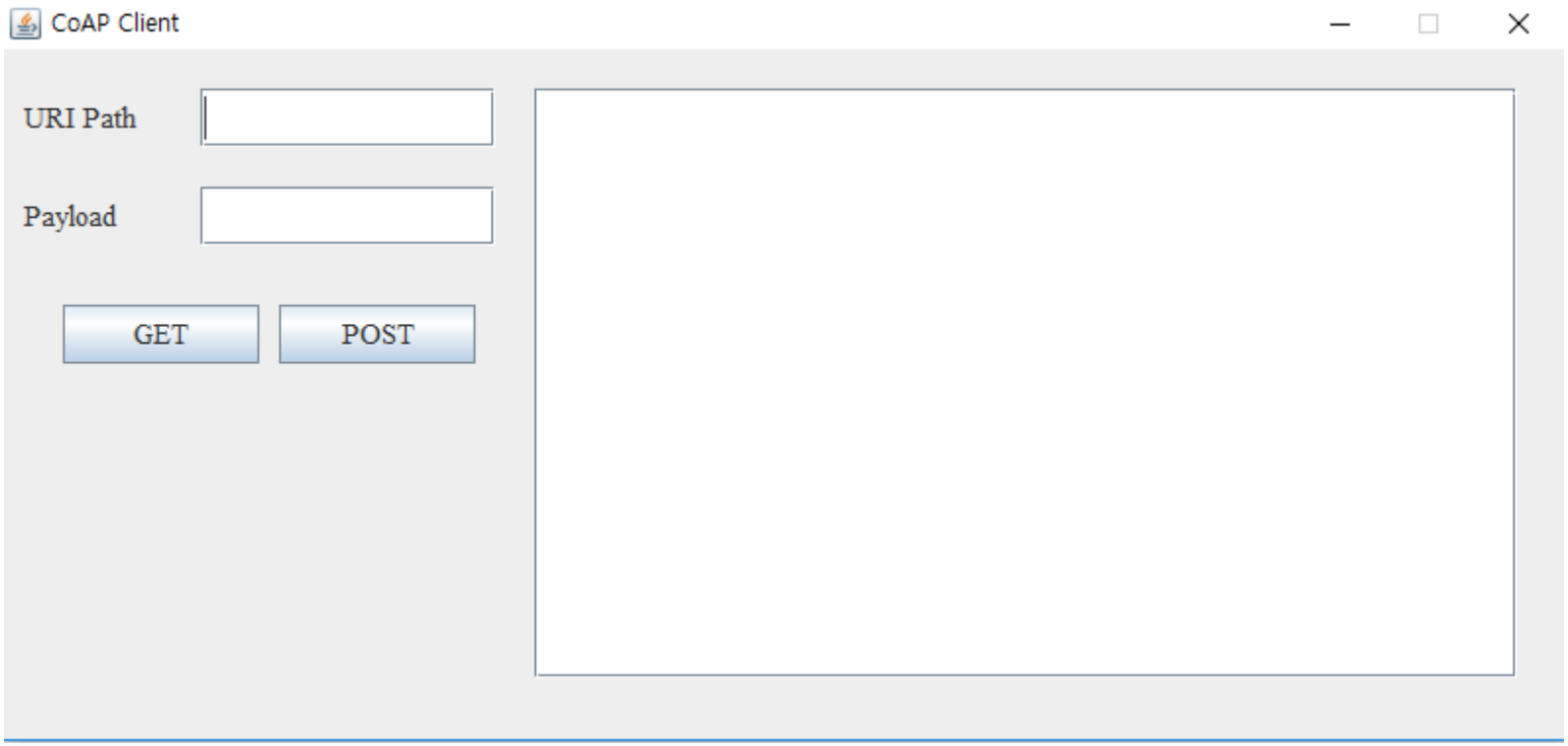
jCoAP Client

- Observe
  - CoapSampleResourceServer.java

```
//observe example

int counter = 0;
while(true){
    try {
        Thread.sleep(5000);
    } catch (InterruptedException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    counter++;
    light.setValue(((String)"Message #" + counter).getBytes());
    light.changed();
}
```

jCoAP Client

- ClientUI() – setVisible(false)
- ClientUI() – setVisible(true) ← Change



Request process (1/2)

- CoapResourceServer.java
  - GET, POST, PUT, DELETE requests are processed
- Request(POST)

```
case POST:
    if( resource != coreResource )
    {
        if (resource != null){
            resource.post(request.getPayload());
            response = channel.createResponse(request, CoapResponseCode.Changed_204);
        } else {
            /* if the resource does not exist, a new resource will be created */
            createResource(parseRequest(request));
            response = channel.createResponse(request, CoapResponseCode.Created_201);
        }
        response.setPayload("success".getBytes());
    } else {
        response = channel.createResponse(request, CoapResponseCode.Forbidden_403);
    }

    break;
```

  - Post() call

Request process (2/2)

- BasicCoapResource.java
  - Handler(onPost()) call

```
@Override
public void post(byte[] data) {
    if (resourceHandler != null) {
        resourceHandler.onPost(data);
    }
    return;
}
```

- CoapSampleResourceServer.java

```
light.registerResourceHandler(new ResourceHandler() {
    @Override
    public void onPost(byte[] data) {
        System.out.println("Post to /test/light");
        light.setValue(data);
        light.changed();
    }
});
```

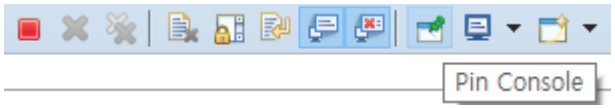
Response process

- BasicCoapClient.java

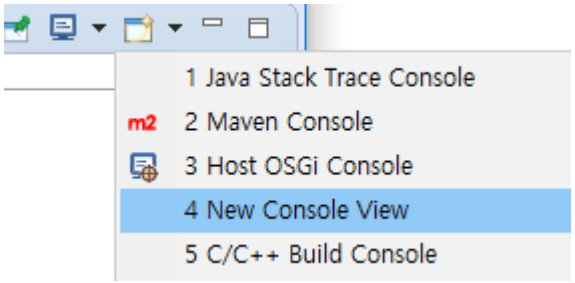
```
@Override
public void onResponse(CoapClientChannel channel, CoapResponse response) {
    System.out.println("Received response");

}
```

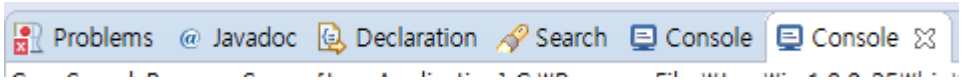
Tip

- Tip
  - Run “CoapSampleResourceServer.java”
    Click “Pin Console”



    Click “New Console view”





  - Run BasicCoapClient.java
    In “New Console view”, Click “BasicCoapClient.java” in “Display selected” and Click “Pin Console”