

zigbee proxy intro

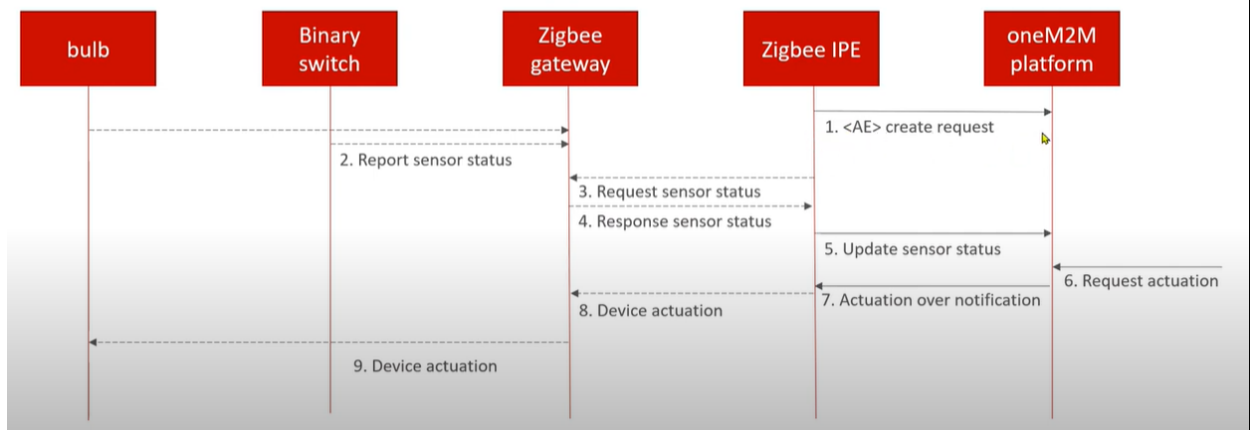
deconz-rest-plugin/websocket_server.cpp : webserver

deconz-rest-plugin/gateway.cpp : http 요청을 zigbee 요청으로 변환 및 반환

deconz-rest-plugin/de_web_plugin_private.h : http request와 response를 정의한다.
따라서 해당 부분을 참고해서 oneM2M request를 deconz에서 원하는 방식으로 바꿔서 요청하면 zigbee response를 http response로 받을 수 있고 이를 수정하면 oneM2M 방식으로 바꿀 수 있을 것 같음. 아마도?

목표

• Simple scenario



1. Zigbee IPE가 oneM2M의 request를 곧바로 zigbee protocol로 변환해야 한다.
2. 예를 들어, 기존 코드에서 oneM2M platform의 GET 요청에 대한 response를 얻기 위해, deconz로 보낸 GET의 response를 oneM2M platform에 보내는 코드를 찾아서 이를 수정해야 한다.
3. 수정해야 할 사항: oneM2M platform의 GET 요청을 곧바로 deconz 코드를 참고하여 zigbee로 요청해야 한다.

Mobius GET method(Zigbee IPE/DeviceData.js)

```

110  this.get_device_temperature = function() {
111      setInterval(function(){
112          for (const [key, value] of Object.entries(resourceConfig)){
113              for(let i = 0; i < value.length; i++){
114                  for(let j = 0; j < value[i].deconzResource.state.length; j++) {
115                      if(value[i].deconzResource.state[j] === "temperature") {
116                          let temperature_value = deconz.sensordata(GATEWAY_ADDRESS, value[i].deconzId, value[i].deconzResource.state[j]);
117                          if (Last_temperature_level.get(value[i].oneM2MResource.deviceResourceName) != temperature_value) {
118                              Last_temperature_level.set(value[i].oneM2MResource.deviceResourceName, temperature_value);
119                              let device_temperature_value_path = "/" + CSE_NAME + "/" + IPE_RESOURCE_NAME + "/" + value[i].deconzResource.state[j];
120                              let device_temperature_value_resp = KETIMobius.put_fcmt(device_temperature_value_path, temperature_value);
121                              console.log(device_temperature_value_resp);
122                          }
123                      }
124                  }
125              }
126          }
127      }, ipeConfig.zigbeeGateway.sensingInterval);
128  }

```

116번째 temperature_value는 deconz.sensordata를 통해 deconz로 GET 요청을 통해 얻은 response이다. 따라서 deconz에서 이에 해당하는 부분을 찾아 수정하면 oneM2M의 GET 요청을 곧바로 Zigbee 요청으로 변경할 수 있다.

deconz GET method (deconz-rest-plugin/rest_devices.cpp)

```

*/
int RestDevices::handleApi(const ApiRequest &req, ApiResponse &rsp)
{
    // GET /api/<apikey>/devices
    if (req.hdr.pathComponentsCount() == 3 && req.hdr.httpMethod() == HttpGet)
    {
        return getAllDevices(req, rsp);
    }
    // GET /api/<apikey>/devices/<uniqueid>
    else if (req.hdr.pathComponentsCount() == 4 && req.hdr.httpMethod() == HttpGet)
    {
        return getDevice(req, rsp);
    }
    // PUT /api/<apikey>/devices/<uniqueid>/ddf/reload
    else if (req.path.size() == 6 && req.hdr.method() == QLatin1String("PUT") && req.path[4] == QLatin1String("ddf/reload"))
    {
        return putDeviceReloadDDF(req, rsp);
    }
    // GET /api/<apikey>/devices/<uniqueid>/ddf
    else if (req.hdr.pathComponentsCount() == 5 && req.hdr.httpMethod() == HttpGet && req.hdr.pathAt(4) == QLatin1String("ddf"))
    {
        return getDeviceDDF(req, rsp);
    }
}

```

deconz에서 GET 요청을 받으면 이를 getDevice를 통해 Zigbee 통신으로 하는 것을 알 수 있다.

가장 핵심: ApiRequest, ApiResponse

```
1012     */
1013     class ApiRequest
1014     {
1015     public:
1016         ApiRequest(const QHttpRequestHeader &h, const QStringList &p, QTcpSocket *s, const QString &c);
1017         QString apiKey() const;
1018         ApiVersion apiVersion() const { return version; }
1019
1020         const QHttpRequestHeader &hdr;
1021         const QStringList &path;
1022         QTcpSocket *sock;
1023         QString content;
1024         ApiVersion version;
1025         ApiAuthorisation auth;
1026         ApiMode mode;
1027     };
1028
1029     /*! \class ApiResponse
1030
1031         Helper to simplify HTTP REST request handling.
1032     */
1033     class ApiResponse
1034     {
```