deconz Get Request → Response

deconz-rest-plugin/rest_gateways.cpp

```
94
95      /*! GET /api/<apikey>/gateways/<id>
96          \return REQ_READY_SEND
97                  REQ_NOT_HANDLED
98       */
99      int DeRestPluginPrivate::getGatewayState(const ApiRequest &req, ApiResponse &rsp)
100     {
101         rsp.httpStatus = HttpStatusOk;
102
103         bool ok;
104         size_t idx = req.path[3].toUInt(&ok);
105
106         if (!ok || idx == 0 || (idx - 1) >= gateways.size())
107         {
108             rsp.list.append(errorToMap(ERR_RESOURCE_NOT_AVAILABLE, QString("/gateways/%1").arg(req.path[3]), QStri
109             rsp.httpStatus = HttpStatusNotFound;
110             return REQ_READY_SEND;
111         }
112
113         idx -= 1;
114
115         gatewayToMap(req, gateways[idx], rsp.map);
116
117         if (rsp.map.isEmpty())
118         {
119             rsp.str = "{}";
120         }
121
122         return REQ_READY_SEND;
123     }
```

GET /api/<apikey>/gateways/<id>에 대하여 gatewayToMap에서 request에 대응하는 response를 받는다.

deconz-rest-plugin/rest_gateways.cpp

```
334 void DeRestPluginPrivate::gatewayToMap(const ApiRequest &req, const Gateway *gw, QVariantMap &map)
335 {
336     Q_UNUSED(req);
337
338     if (!gw)
339     {
340         return;
341     }
342
343     if (!gw->uuid().isEmpty())
344     {
345         map[QLatin1String("uuid")] = gw->uuid();
346     }
347     if (!gw->name().isEmpty())
348     {
349         map[QLatin1String("name")] = gw->name();
350     }
351     map[QLatin1String("ip")] = gw->address().toString();
352     map[QLatin1String("port")] = (double)gw->port();
353     map[QLatin1String("pairing")] = gw->pairingEnabled();
354
355     if (!gw->groups().empty())
356     {
357         QVariantMap groups;
358
359         for (size_t i = 0; i < gw->groups().size(); i++)
360         {
361             const Gateway::Group &g = gw->groups()[i];
362             groups[g.id] = g.name;
363         }
364
365         map[QLatin1String("groups")] = groups;
366     }
367
```

```
Gateway::Gateway(DeRestPluginPrivate *parent) :
    QObject(parent),
    d_ptr(new GatewayPrivate)
{
    Q_D(Gateway);
    d->parent = parent;
    d->pings = 0;
    d->port = 0;
    d->state = Gateway::StateOffline;
    d->pairingEnabled = false;
    d->needSaveDatabase = false;
    d->reply = nullptr;
    d->manager = new QNetworkAccessManager(this);
    connect(d->manager, SIGNAL(finished(QNetworkReply*)), this, SLOT(finished(QNetworkReply*)));
    d->timer = new QTimer(this);
    d->timer->setSingleShot(true);
    d->reqBuffer = new QBuffer(this);
    connect(d->timer, SIGNAL(timeout()), this, SLOT(timerFired()));

    d->startTimer(5000, ActionProcess);
}
```

Qobject::connect에 의하여 TCP/IP 통신에 대한 응답을 수신하여 Gateway가 생성된다. 이를 통해 생성된 Gateway가 있다면 response를 받을 것이고 이를 res에 적어서 나중에 json파일로 파싱한다.

Signals & Slots | Qt Core 6.6.0

An overview of Qt's signals and slots inter-object communication mechanism. Signals and slots are used for communication between objects. The signals and slots mechanism is a central feature of Qt and probably the part that differs most from the features provided by other frameworks. Signals and slots are made possible by

https://doc.qt.io/qt-6/signalsandslots.html