



## MODULE 03

### Contrôle domotique

# MODULE 03

## SÉANCE SYSTÈME 01

### TP D'INFORMATIQUE

Durée 2h30

## PRISE EN MAIN DE L'API REST DECONZ ET ANALYSE WIRESHARK

### BLOC DE COMPÉTENCES

U6 - VALORISATION DE LA DONNÉE ET CYBERSÉCURITÉ

### COMPÉTENCE(S)

C10 EXPLOITER UN RÉSEAU INFORMATIQUE

### OBJECTIF PÉDAGOGIQUE

Étude de la configuration d'une installation domotique : utilisation d'une API HTTP (REST). Analyse Wireshark des données.

### CONNAISSANCES ISSUES DU RÉFÉRENTIEL

- |   |          |
|---|----------|
| • Réseaux IoT   | Niveau 2 |
| • Réseaux informatiques (protocoles, équipements et outils usuels et industriels) | Niveau 3 |
| • Outils logiciels d'évaluation, de traçabilité de l'information                  | Niveau 3 |

### CONNAISSANCES OPÉRATIONNALISÉES

- |   |          |
|---|----------|
| • Utilisation d'une API REST à partir de sa documentation | Niveau 3 |
| • Analyser une communication réseau avec Wireshark        | Niveau 2 |

## TD

# Comprendre l'API REST

### Qu'est ce que le REST ?

Une API, ou interface de programme d'application, est un ensemble de règles qui définissent comment les applications ou les unités peuvent se connecter et communiquer entre eux. Une API REST est une API conforme aux principes de conception du style architectural REST ou Representational State Transfer. C'est pour cette raison que les API REST sont parfois appelées API RESTful .

Les API REST communiquent **via des requêtes HTTP** pour exécuter des fonctions de base de données standard telles que la **création, la lecture, la mise à jour et la suppression** d'enregistrements dans une ressource.

La représentation de la ressource peuvent être fournies à un client dans pratiquement n'importe quel format, y compris JavaScript Object Notation (JSON), HTML, XML, PHP ou un texte en clair

### Structure d'une requête



Figure 1: Structure d'une requête HTTP

Dans une requête HTTP, nous aurons plusieurs paramètres dont la méthode et l'URL qui nous intéressent ici. D'abord intervient la méthode puis la ressource via l'URL.

#### A quoi sert la méthode ?

Permettre de faire une requete a une adresse souhaité

#### A quoi sert la ressource (URL) ?

C'est l'endroit ou la ressource est demandé, par exemple des informations sur un appareil

## Méthodes et ressource dans une requête

Les méthodes de requête HTTP permettent de définir l'action que l'on souhaite réaliser sur la ressource indiquée. Il en existe plusieurs mais nous allons nous focaliser sur les suivantes GET, POST, PUT, DELETE. Quand à elle, la ressource permet de définir la demande de l'utilisateur.

Définir dans quel cas chaque méthode est utilisée dans une API REST :

**GET** : Permet d'obtenir une ressource

**POST** : Créer une nouvelle ressource

**PUT** : Modifie une ressource existante

**DELETE** : Supprime une ressource

Quels caractères sont utilisés comme séparateurs dans une trame HTTP ? Indiquer les 4 caractères de fin de trame, comment les écrire dans un code en langage C ?

LEs : 200 OK

## Format JSON

Les données sont envoyées au format JSON.

Qu'est ce que le format JSON ?

c'est des structures de textes

Donner un exemple de données au format JSON :

"elements":{ objets: }

## TP

# L'API deconz

### Objectif

Comprendre l'API utilisée par deconz ([https://dresden-elektronik.github.io/deconz-rest-doc/about\\_rest/](https://dresden-elektronik.github.io/deconz-rest-doc/about_rest/))

### Préparation

Dans cette partie nous aurons besoin d'un client REST permettant de tester les requêtes HTTP.

| Installer une extension REST Client pour Firefox ou Postman ou Boomrang pour Chrome ou Edge

### Les bases de l'API

Selon le lien précédent, donner les informations suivantes :

| Quelle ressource faut ajouter à l'URL pour avoir des informations sur tous les lumières ?

/lights

| Quelle ressource faut ajouter à l'URL pour avoir des informations sur la lumière numéro 1 ?

/lights/1

| Quelle ressource faut ajouter à l'URL pour avoir des informations sur tous les capteurs ?

/sensors

| Que doivent faire les clients avant de pouvoir interroger le l'API ?

posséder une clé api

| Quels sont les avantages à utiliser l'API REST ?

Accessibilité, compatible avec une multitude de langage, intelligible

## Trouver l'adresse IP de la passerelle domotique

Pour scanner un réseau, vous pouvez utiliser le logiciel « Advanced IP Scanner »

Installer le logiciel précédent sur votre poste

Scanner le réseau 172.20.21.0-254/24

Trouver le module ayant comme site web gatewayCIELIR

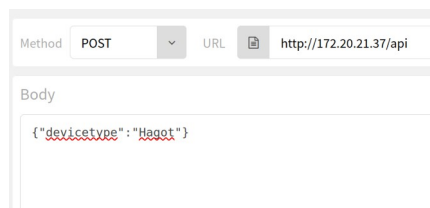
## Obtention d'une clé API

Donner les requêtes HTTP permettant de trouver l'adresse ip de la passerelle

Donner les requêtes HTTP permettant récupérer une clé pour l'API en utilisant votre nom

Pour obtenir une clé de l'API, il faut demander à l'enseignant d'activer la demande de clé.

Ouvrir le client rest de votre navigateur, le compléter avec la requête précédente afin d'obtenir une clé d'utilisation. Exemple de client rest :



The screenshot shows a REST client interface. The 'Method' is set to 'POST' and the 'URL' is 'http://172.20.21.37/api'. The 'Body' is a JSON object: {"device\_type": "Hagot"}.

Ouvrez wireshark et vérifiez l'envoi de la requête HTTP et la réception de la réponse. Quelle est la durée de la transaction (entre l'envoi de la requête HTTP et la réponse du serveur) ? Pour faciliter la mesure, vous pouvez définir le premier élément comme étant la référence temporelle : menu éditer → Fixer le temps de référence.

Noter ici votre clé

F276DD7951

## Interroger un élément

Quelle requête permet d'obtenir la liste des adresses mac de tous les éléments connectés à la passerelle ?

/devices

Exécuter la requête et noter les deux premières valeurs

"00:15:8d:00:08:a9:f1:ce" , "00:21:2e:ff:ff:06:dd:aa"

Quelle requête permet d'obtenir les informations d'un élément selon son adresse mac ?

/devices/adresseMac

Exécuter la requête et noter le nom du fabricant du 3<sup>ème</sup> élément

LEDVANCE

Exécuter la requête permettant d'obtenir la liste des différentes lumières présentes. Donner le nombre d'élément présent

3

Constatez vous un ou plusieurs éléments particuliers ? Si oui pourquoi ?

Oui il y en, comme les parametre de chaque lumieres

Quel paramètre permet de faire la différence entre les éléments ?

Les parametre d'etat

Quel est sa valeur pour les lumières ?

Cela depends des appareil, mais c'est soit true soit false pour : on

Exécuter la requête permettant d'obtenir la liste des différents capteurs présents. Donner le nombre d'élément présent

Il y a environ 18 appareils

Constatez vous un ou plusieurs éléments particuliers ? Si oui pourquoi ?

Ils ont des proprietes qui different

## Étude de la réponse JSON

Notepad++ dispose d'une extension permettant de rendre plus lisible les données JSON. Ouvrir Notepad++ -> ALT+P -> Plugin Manager -> JSON Viewer -> Install. Redémarrer Notepad++. Vous pouvez maintenant utiliser le raccourci : CTRL + ALT +SHIFT + M ou bien ALT+P -> Plugin Manager -> JSON Viewer -> Format JSON. Appliquer ce formatage à la liste des capteurs.

## Modifier un élément

Parmi les données JSON d'une lumière, quelle paramètre permet d'avoir son état ?

State

Donner la requête permettant d'éteindre la première prise de votre salle

```
/api/apikey/lights/1/state  
{ "on": false }
```

Exécuter la requête précédente et vérifier le changement d'état.

Donner la requête permettant d'éteindre la lumière de votre salle

```
/api/apikey/lights/3  
{ "on": false }
```

Exécuter la requête précédente et vérifier le changement d'état.



# MODULE 03

## SÉANCE SYSTÈME 02

### TP D'INFORMATIQUE

Durée 2h30

## CLIENT HTTP EN C++ : ENVOI REQUÊTE REST

### BLOC DE COMPÉTENCES

U6 - VALORISATION DE LA DONNÉE ET CYBERSÉCURITÉ

### COMPÉTENCE(S)

C08 - CODER

### OBJECTIFS PÉDAGOGIQUES

Analyse UML de l'installation. Codage d'un client HTTP permettant de demander la liste du matériel domotique.

### CONNAISSANCES ISSUES DU RÉFÉRENTIEL

- |  |          |
|--|----------|
| • Langages de modélisation                                       | Niveau 2 |
| • Programmation orientée objet                                   | Niveau 3 |
| • Programmation réseau   | Niveau 2 |
| • Outils logiciels d'évaluation, de traçabilité de l'information | Niveau 3 |

### CONNAISSANCES OPÉRATIONNALISÉES

- |  |          |
|--|----------|
| • Utiliser une classe C++                          | Niveau 3 |
| • Envoyer une trame HTTP                           | Niveau 3 |
| • Analyser une communication réseau avec Wireshark | Niveau 2 |
| • Versionner un code                               | Niveau 2 |

## TP

# Envoi de la requête GET permettant d'obtenir la liste des appareils

### Copie de la requête GET

En utilisant un client REST, envoyer à nouveau la requête GET permettant d'obtenir la liste des capteurs. Analyser les échanges sous Wireshark et copier la requête complète dans un bloc-note. Chaque fin de ligne doit être complétée par `\r\n`, les 4 caractères `\r\n\r\n` doivent terminer la trame HTTP.

La requête complète :

```
22 1.707956 172.20.21.32 172.20.21.5 HTTP 381 GET /api/F276DD7951/sensors HTTP/1.1
```

Quel filtre Wireshark avez-vous utiliser ?

```
ip.addr == 172.20.21.5 && http
```

Quels sont les protocoles sur lesquels HTTP est basé ?

Il s'appuie sur le TCP

Copier de la même manière la liste des lampes.

## Codage du programme principal permettant d'envoyer la requête et d'afficher la réponse

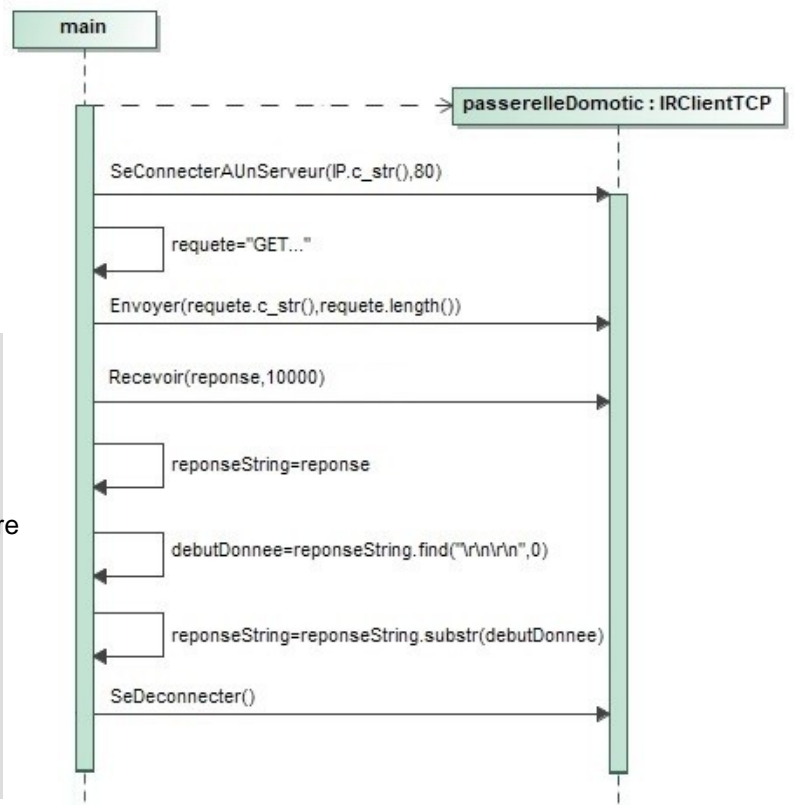
Mettre en place le code correspondant au diagramme de séquence ci-contre.

Quel doit être la classe de l'objet requete, sachant que cette classe contient les méthodes `c_str()` et `length()` ? A quoi servent ces 2 méthodes ?

`length` permet de dire la taille d'une chaîne de caractère

`c_str()` permet de convertir une chaîne de caractère en tableau de char

Requete est un string



Donner la signification des arguments de la méthode `find`. Que renvoie cette méthode ? A quelle classe appartient-elle ?

il y a `"\r\n\r\n"` et `0`. Elle renvoie le contenu de la trame http. Elle appartient à `IRClientTCP`

Même question pour la méthode `substr` ?

Permet de substituer la string et de la transposer dans une variable

Ajouter l'affichage de la réponse.

Versionner le code complet.

# MODULE 03

## SÉANCE SYSTÈME 03

### TP D'INFORMATIQUE

Durée 2h30

## ENVOI D'UNE REQUÊTE GET EN C++ EXTRACTION DE DONNÉES JSON

### BLOC DE COMPÉTENCES

U6 - VALORISATION DE LA DONNÉE ET CYBERSÉCURITÉ

### COMPÉTENCE(S)

C08 - CODER

### OBJECTIFS PÉDAGOGIQUES

Codage de l'envoi d'une requête GET permettant de demander l'état d'un capteur de température. Extraction de données JSON : affichage de la température, de l'humidité et de la pression atmosphérique.

### CONNAISSANCES ISSUES DU RÉFÉRENTIEL

- |  |          |
|--|----------|
| • Langages de modélisation                                       | Niveau 2 |
| • Programmation orientée objet                                   | Niveau 3 |
| • Programmation réseau   | Niveau 2 |
| • Outils logiciels d'évaluation, de traçabilité de l'information | Niveau 3 |

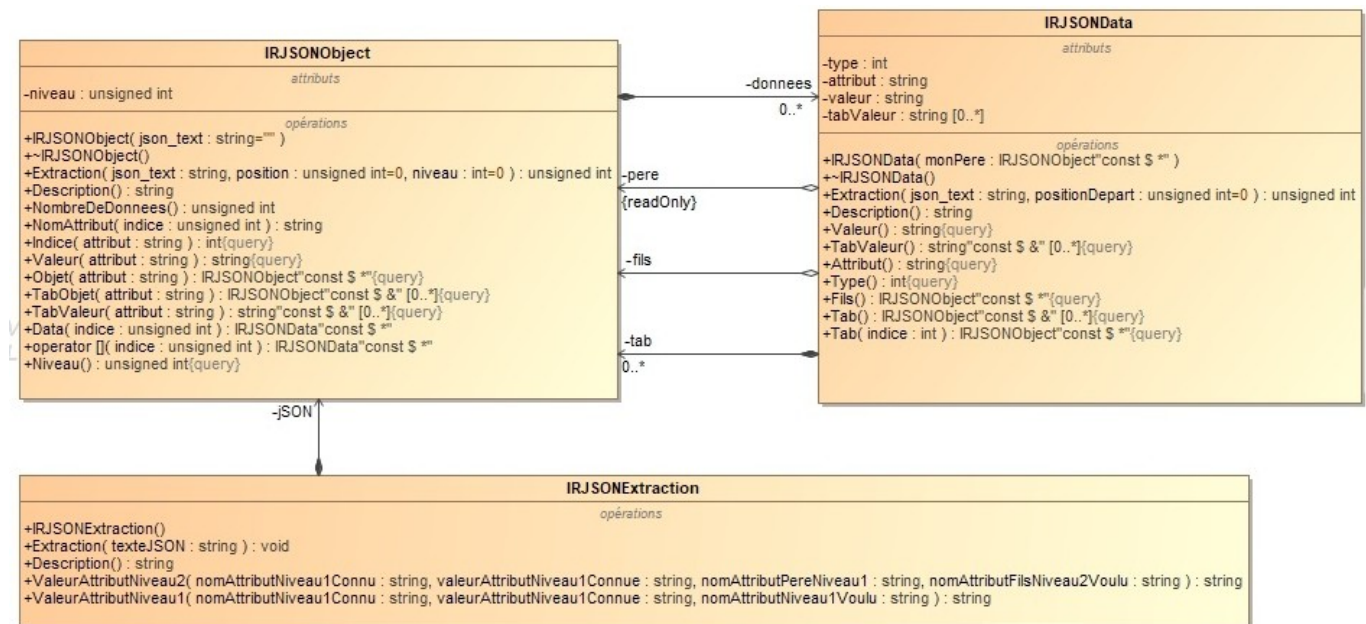
### CONNAISSANCES OPÉRATIONNALISÉES

- |  |          |
|--|----------|
| • Utiliser une classe C++                          | Niveau 3 |
| • Envoyer une trame GET en HTTP                    | Niveau 3 |
| • Analyser une communication réseau avec Wireshark | Niveau 2 |
| • Versionner un code                               | Niveau 2 |

## TP

## EXTRACTION DES MESURES D'UN CAPTEUR DE TEMPÉRATURE

## Extraction des données JSON : la classe IRJSONExtraction



Créer un objet JSON de la classe IRJSONExtraction dont le code est donné. Appeler la méthode Extraction, puis afficher la liste des capteurs à l'aide de la méthode Description. Votre code :

```

IRJSONExtraction json;
monJson.Extraction(reponse);
std::cout << monJson.Description();
  
```

## Extraction des données du capteur de température

Dans la donnée JSON correspondant à la liste des capteurs, à quoi correspond l'attribut de niveau 0 ?

A l'identifiant du capteur

Quelle est le nom de l'attribut de niveau 1 permettant d'identifier un capteur de température, quelle est sa valeur pour un des capteurs de température ?

"name" Temp A7-1

Quel est le nom de l'attribut de niveau 2 correspondant à la température ?

"temperature"

Quel est le nom de son père de niveau 1 ?

"state"

Ajouter à votre code l'appel de la méthode ValeurAttributNiveau2 permettant d'afficher la température. Votre code :

```
std::string temp = JSON.ValeurAttributNiveau2("name","Temp A7-1","state","temperature");
```

Afficher maintenant l'humidité et la pression atmosphérique. Votre code :

```
std::string press = JSON.ValeurAttributNiveau2("name","Temp A6-1","state","pressure");  
std::string humid = JSON.ValeurAttributNiveau2("name", "Temp A6-1", "state", "humidity");
```

Afficher la valeur de l'attribut « type », en utilisant la méthode ValeurAttributNiveau1. Votre code :

```
std::string type = JSON.ValeurAttributNiveau1("name","Temp A7-1", "type");
```

Versionner le code complet.

# **MODULE 03**

## **SÉANCE SYSTÈME 04**

### **TP D'INFORMATIQUE**

**Durée 2h30**

## **ENVOI D'UNE REQUÊTE PUT EN C++**

### **BLOC DE COMPÉTENCES**

U6 - VALORISATION DE LA DONNÉE ET CYBERSÉCURITÉ

### **COMPÉTENCE(S)**

C08 - CODER

### **OBJECTIFS PÉDAGOGIQUES**

Codage de l'envoi d'une requête PUT permettant de modifier l'état d'une lampe multicolore. Mise en forme de la donnée au format JSON.

### **CONNAISSANCES ISSUES DU RÉFÉRENTIEL**

- |  |          |
|--|----------|
| • Langages de modélisation                                       | Niveau 2 |
| • Programmation orientée objet                                   | Niveau 3 |
| • Programmation réseau   | Niveau 2 |
| • Outils logiciels d'évaluation, de traçabilité de l'information | Niveau 3 |

### **CONNAISSANCES OPÉRATIONNALISÉES**

- |  |          |
|--|----------|
| • Utiliser une classe C++                          | Niveau 3 |
| • Envoyer une trame PUT en HTTP                    | Niveau 3 |
| • Analyser une communication réseau avec Wireshark | Niveau 2 |
| • Versionner un code                               | Niveau 2 |

## TP

## ENVOI DE LA REQUÊTE PUT PERMETTANT DE PILOTER UNE LAMPE

### Copie de la requête PUT

En utilisant un client REST, envoyer la requête PUT permettant d'allumer ou d'éteindre une lampe. Analyser les échanges sous Wireshark et copier la requête complète dans un bloc-note. Chaque fin de ligne doit être complétée par `\r\n`, les 4 caractères `\r\n\r\n` doivent terminer la trame HTTP.

La requête complète sans la donnée JSON :

```
{"on": false}
```

http://172.20.21.5/api/F276DD7951/lights/3/state

Frame 73: 471 bytes on wire (3768 bits), 471 bytes captured (3768 bits) on interface \Device\NPF\_{BD012B80-6367-4ADB-95C4-60CCDAAF2EC1}, id 0Ethern

La donnée JSON permettant d'allumer la lampe (elle sera placée à la suite de la requête HTTP) :

```
std::string requeteLampe ="PUT /api/F276DD7951/lights/3/state HTTP/1.1\r\nHost: 172.20.21.5\r\nConnection: keep-alive\r\n\r\n";
```

```
{"on", true}
```

La donnée JSON permettant de modifier la couleur pour du rouge :

```
{"hue", 360}
```

La donnée JSON permettant de fixer l'intensité maximum :

```
{"bri", 255}
```



## Envoi de la requête PUT

En vous basant sur le code d'envoi de la requête GET, coder l'envoi de la requête PUT, la donnée JSON doit être placée à la suite de la requête (concaténation). Votre code :

```
char[1000];  
std::string requeteLampe ="PUT /api/F276DD7951/lights/3/state HTTP/1.1\r\nHost: 172.20.21.5\r\nConnection: keep-alive\r\n\r\n{\"on\": false}";
```

Ajouter une boucle permettant à l'utilisateur d'allumer la lampe en saisissant « true », ou bien de l'éteindre en saisissant « false ».

Ajouter le pilotage de la couleur et de l'intensité lumineuse.

Versionner le code complet.

# **MODULE 03**

## **SÉANCE SYSTÈME 05**

### **TP D'INFORMATIQUE**

**Durée 2h30**

## **CODAGE DES CLASSES CAPTEUR ET LAMPE EN C++**

### **BLOC DE COMPÉTENCES**

U6 - VALORISATION DE LA DONNÉE ET CYBERSÉCURITÉ

### **COMPÉTENCE(S)**

C08 - CODER

### **OBJECTIFS PÉDAGOGIQUES**

Codage d'une classe Capteur intégrant l'état d'un capteur et la génération de la requête GET. Codage d'une classe Lampe intégrant l'état de la lampe et la génération de la requête PUT.

### **CONNAISSANCES ISSUES DU RÉFÉRENTIEL**

- |  |          |
|--|----------|
| • Langages de modélisation d'application | Niveau 3 |
| • Programmation orientée objet           | Niveau 3 |
| • Programmation réseau                   | Niveau 3 |

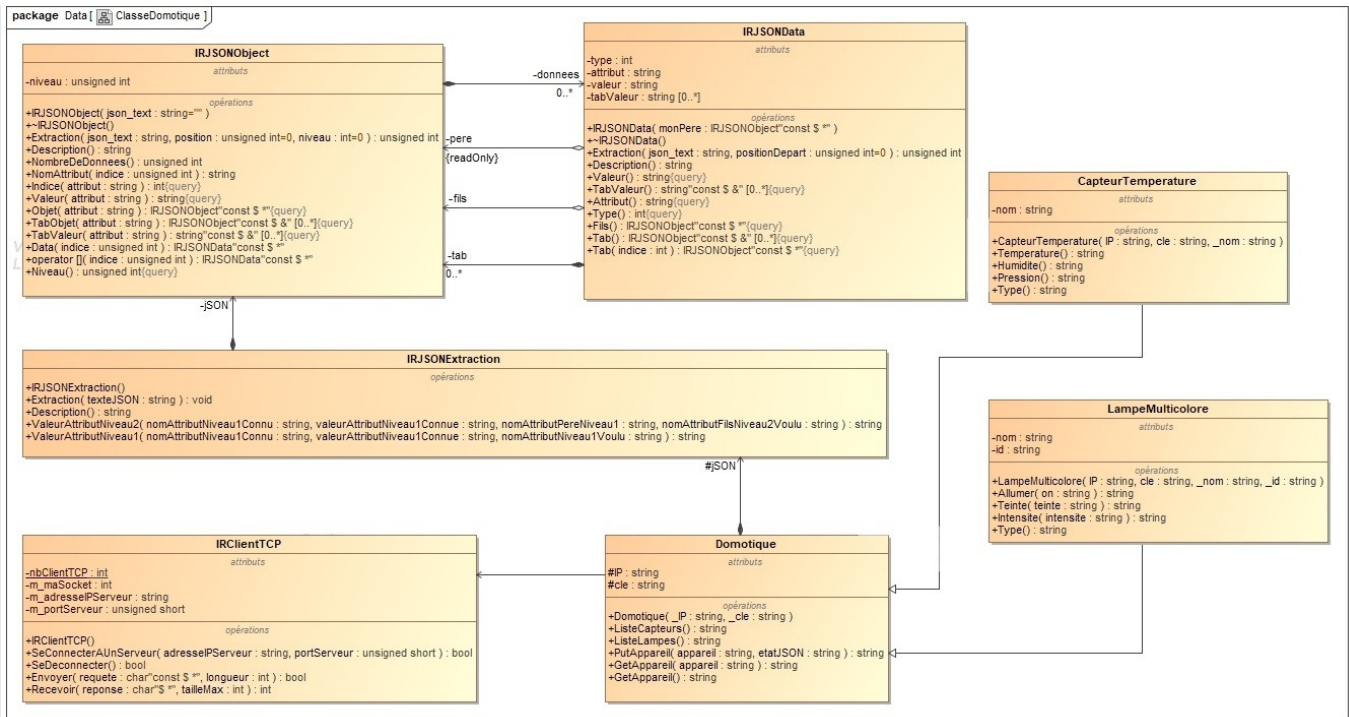
### **CONNAISSANCES OPÉRATIONNALISÉES**

- |  |          |
|--|----------|
| • Écrire une classe C++ à partir d'un diagramme de classes | Niveau 2 |
| • Versionner un code                                       | Niveau 2 |

## TP

## codage de la classe Domotique

Le code doit maintenant être réorganiser sous forme de classes. La classe Domotique permet de gérer les requêtes REST quelque soit l'appareil. Cette classe se spécialise ensuite en 2 classes filles : CapteurTemperature et LampeMulicore. Voici le diagramme de classe complet :



Qu'est ce qu'une composition ? Dans le diagramme ci-dessus, donner un exemple de composition.

Qu'est ce qu'un héritage ? Dans le diagramme ci-dessus, donner un exemple d'héritage.

Coder et tester la classe Domotique (la dernière méthode `std::string GetAppareil()` ne sera pas codée. Exemple de programme principal :

```
int main()
{
    Domotique domotique("172.20.21.4", "4FA1C71BD7");
    std::cout<<domotique.ListeCapteurs()<<std::endl<<std::endl;
    std::cout<<domotique.ListeLampes()<<std::endl<<std::endl;
    system("PAUSE");
    return 0;
}
```

Le code de la déclaration de votre classe :

## TP

### codage des classes

## CapteurTemperature et LampeMulticolore

Comment se code un héritage dans la déclaration d'une classe ?

Expliquer le passage d'arguments à la classe mère, dans la définition du constructeur de la classe CapteurTemperature :

```
CapteurTemperature::CapteurTemperature(string IP,string cle,string _nom):Domotique(IP,cle)
{
    nom=_nom;
}
```

Coder et tester les 2 classes. Exemple de programme principal :

```
int main()
{
    CapteurTemperature captTemperature("172.20.21.4","4FA1C71BD7","Temp A7-1");
    std::cout<<captTemperature.Type()<<std::endl;
    std::cout<<captTemperature.Temperature()<<std::endl;
    std::cout<<captTemperature.Humidite()<<std::endl;
    std::cout<<captTemperature.Pression()<<std::endl<<std::endl;
    LampeMulticolore lampe("172.20.21.4","4FA1C71BD7","Extended color light 3","3");
    std::cout<<lampe.Type()<<std::endl;
    while(1)
    {
        std::cout<<"teinte (0-65536), intensite (0-255), allumer (true false) : ";
        std::string teinte,intensite,allumer;
        std::cin>>teinte>>intensite>>allumer;
        lampe.Allumer(allumer);
        lampe.Teinte(teinte);
        lampe.Intensite(intensite);
    }
    return 0;
}
```

Le code de la déclaration de votre classe CapteurTemperature :

Le code de la déclaration de votre classe LampeMulticolore :

[Versionner le code complet.](#)

# **MODULE 03**

## **SÉANCE SYSTÈME 06**

### **TP D'INFORMATIQUE**

**Durée 2h30**

## **INTERFACE GRAPHIQUE DOMOTIQUE EN C++**

### **BLOC DE COMPÉTENCES**

U6 - VALORISATION DE LA DONNÉE ET CYBERSÉCURITÉ

### **COMPÉTENCE(S)**

C08 - CODER

### **OBJECTIFS PÉDAGOGIQUES**

Codage d'une interface graphique permettant d'afficher l'état des capteurs et de piloter les lampes.

### **CONNAISSANCES ISSUES DU RÉFÉRENTIEL**

- Langages de développement, de description, de création d'API et les IDE associés Niveau 3
- Programmation orientée objet Niveau 3

### **CONNAISSANCES OPÉRATIONNALISÉES**

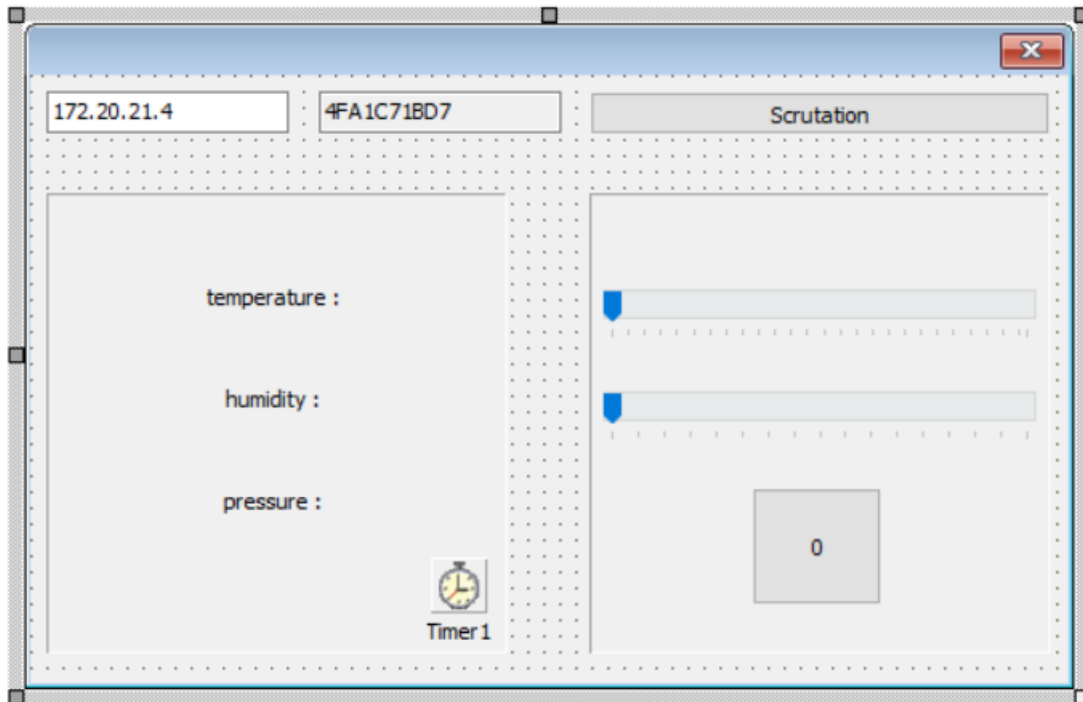
- Gestion des objets graphiques et des événements. Niveau 2
- Versionner un code Niveau 2

## TP

# Intégration dans une interface graphique des classes

## CapteurTemperature et LampeMulticolore

Créer un projet graphique sous C++Builder. Ajouter les éléments suivants :

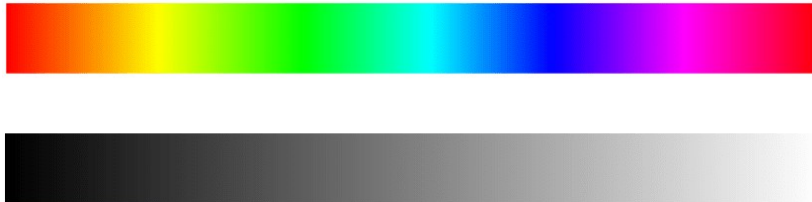




L'appui sur Scrutation lance le timer chargé d'appeler les méthodes de la classe CapteurTemperature. Le bouton « 0 » permet d'allumer la lampe, les 2 trackBars règlent l'intensité et la couleur. Les cadres sont des Bevels.

Coder entièrement l'interface et valider son fonctionnement.

Ajouter une image permettant de visualiser les couleurs de la teinte, puis une autre pour l'intensité :



Versionner le code complet.

# **MODULE 03**

## **SÉANCE SYSTÈME 07**

### **TP D'INFORMATIQUE**

**Durée 2h30**

## **ANDROID : UTILISATION DE LA CLASSE CAPTEUR**

### **BLOC DE COMPÉTENCES**

U5 - EXPLOITATION ET MAINTENANCE DE RÉSEAUX INFORMATIQUES

### **COMPÉTENCE(S)**

C06 - VALIDER UN SYSTÈME INFORMATIQUE

### **OBJECTIFS PÉDAGOGIQUES**

Installation Android Studio et test de déploiement sur tablette. Codage de l'application permettant d'afficher l'état des capteurs sur tablette.

### **CONNAISSANCES ISSUES DU RÉFÉRENTIEL**

- |  |          |
|--|----------|
| • Maîtrise des environnements de développement, d'intégration, de déploiement logiciel et des versions logicielles associées | Niveau 2 |
| • Langages de développement, de description, de création d'API et les IDE associés   | Niveau 3 |
| • Tests unitaires et d'intégration   | Niveau 3 |

### **CONNAISSANCES OPÉRATIONNALISÉES**

- |  |          |
|--|----------|
| • Installer un environnement de développement et de test pour appareil mobile sous Android | Niveau 2 |
| • Coder une application pour Android utilisant les sockets                                 |          |
| • Versionner un code   | Niveau 2 |

# Android - Utilisation de la classe Capteur

## Objectif de la séance

L'objectif de cette partie est de créer une application Android qui fonctionnera donc sur tous les smartphones et tablettes possédant l'OS Android. Cette application permettra l'affichage des données des capteurs (SS07) : la température, le taux d'humidité et la pression. Cette application réutilisera les classes que vous avez développées en C++ lors des précédentes séances.

## Création d'un projet sur Android Studio

Android Studio est un environnement de développement créé par Google pour créer des application pour l'OS Android. Cet environnement de développement utilise le Java et maintenant de Kotlin comme langage de développement. Il est cependant possible d'ajouter des fonctions natives en C++ avec le NDK (Native Development Kit) d'Android. C'est ce que nous ferons. Nous pourrons ainsi réutiliser du code que nous avons déjà écrit et testé.

Créer un nouveau projet Android en utilisant le NDK. Suivre le tuto vidéo fourni : [AndroidStudio-CréationDUnProjetNDK.mp4](#).

## Test de compilation et d'installation de l'application sur un émulateur

Par défaut, lors de l'installation d'Android, un AVD a été installé. Un AVD est un Android Virtual Device, c'est-à-dire un appareil virtuel fonctionnant sous Android. L'appareil virtuel est un téléphone Pixel 3A.

Tester si votre nouvelle application se lance bien sur l'appareil virtuel

## Test de compilation et d'installation de l'application sur une tablette

Tester également si votre nouvelle application se lance bien sur la tablette

## Ajout des éléments dans l'IHM (Voir le tuto vidéo)

Voir le tuto : [AndroidStudio-PositionnementDesElementsDansLaFenetre.mp4](#)

- Ajout d'un Button et d'un TextEdit
  - Lien vers un tuto de mise en page : [Use ConstraintLayout to design your Android views](#)
  - Button :
    - id : buttonMAJ
    - text : Mise à jour des capteurs
  - TextEdit :
    - id : textEditTemperature
    - text : La température

En suivant le tuto vidéo, ajouter ces éléments dans l'IHM. Vérifier leur placement.

## Gestion du clic sur le Button et affichage de la température

Voir le tuto : *AndroidStudio-ImporterEtUtiliserDesClasses.mp4*

Ajout des classes dans le projet

Ajout des classes dans le répertoire cpp (.h et .cpp)

Ajout des fichiers sources (.cpp) dans le fichier CMakeLists.txt

- Ajout d'une nouvelle fonction native (tuto video)
  - Ajout de la fonction dans la classe MainActivity
    - `public native String RecupererTemperatureSalleA7();`
  - Ajout automatique de la fonction dans le fichier native-lib.cpp
  - Écriture du code de la fonction en C++ :
    - Création d'un objet `captTemp` de la classe `CapteurTemperature`
    - Récupération de la température dans un objet `strTemp` de la classe `string` en appelant la méthode `Temperature()` avec l'objet `captTemp`.
    - Renvoyer l'objet `strTemp`.
  - Appeler la fonction `RecupererTemperatureSalleA7()` lors du clic sur `buttonMAJ`, récupérer sa valeur de retour et la mettre dans le `textEditTemperature`.

En suivant le tuto vidéo, effectuer l'ensemble de ces étapes.

Faire de même pour les données de pression :

- Ajouter un nouveau `textEdit` dont l'id est `textEditPression`
- Ajouter une nouvelle fonction native : `public native String RecupererPressionSalleA7();`
- Ajouter automatiquement cette fonction dans le fichier `native-lib`
- Écrire le code en C++ de cette fonction
- Appeler la fonction `RecupererPressionSalleA7()` lors du clic sur `buttonMAJ`, récupérer sa valeur de retour et l'afficher dans le `textEditPression`

Faire de même pour les données d'humidité:

- Ajouter un nouveau `textEdit` dont l'id est `textEditHumidite`
- Ajouter une nouvelle fonction native : `public native String RecupererHumiditeSalleA7();`
- Ajouter automatiquement cette fonction dans le fichier `native-lib`
- Écrire le code en C++ de cette fonction
- Appeler la fonction `RecupererHumiditeSalleA7()` lors du clic sur `buttonMAJ`, récupérer sa valeur de retour et l'afficher dans le `textEditHumidite`

### En bonus, des pistes d'amélioration :

- Améliorer l'affichage des données :
  - Ajouter la virgule à la température,
  - Arrondir à l'unité la pression,
  - Ajouter les unités de chacune des mesures.
- Ajouter les capteurs de la salle A6
- Rafraîchir les données toutes les minutes
- Améliorer le design de l'application : un thermomètre, un baromètre, une jauge d'humidité...

# **MODULE 03**

## **SÉANCE SYSTÈME 08**

### **TP D'INFORMATIQUE**

**Durée 2h30**

## **ANDROID : UTILISATION DE LA CLASSE LAMPE**

### **BLOC DE COMPÉTENCES**

U6 - VALORISATION DE LA DONNÉE ET CYBERSÉCURITÉ

### **COMPÉTENCE(S)**

C08 - CODER

### **OBJECTIFS PÉDAGOGIQUES**

Codage d'une application Android permettant de piloter les lampes d'une installation domotique.

### **CONNAISSANCES ISSUES DU RÉFÉRENTIEL**

- Langages de développement, de description, de création d'API et les IDE associés Niveau 3
- Programmation orientée objet Niveau 3

### **CONNAISSANCES OPÉRATIONNALISÉES**

- Concevoir une interface graphique sous Android : gestion des événements java et communication en C++ natif Niveau 2
- Versionner un code Niveau 2

# Android : Utilisation de la classe LampeMulticolore

## Objectif de la séance

L'objectif de cette séance est d'ajouter à l'application Android la gestion des lumières multicolores : la gestion de l'allumage de la lampe, de la gestion des couleurs et de l'intensité.

## Gestion de l'allumage et de l'extinction de la lampe

Dans l'interface graphique (activity\_main.xml) :

Dans l'interface graphique (activity\_main.xml), ajouter 2 boutons dans votre interface : le 1<sup>er</sup> bouton : (id : buttonOn, text : ON) et le 2ème bouton (id:buttonOff, text : OFF)

Dans la classe d'interface MainActivity (MainActivity.java)

Dans la classe d'interface (MainActivity.java), gérer les clics sur ces deux boutons. Lors du clic sur le buttonOn, appeler la fonction AllumerLampeMulticoloreA6(). Lors du clic sur le buttonOff, appeler la fonction EteindreLampeMulticoloreA6().

Tout en bas de la classe d'interface (MainActivity.java), déclarer les 2 fonctions natives : AllumerLampeMulticoloreA6() et EteindreLampeMulticoloreA6().

Dans le fichier native-lib.cpp :

Écrire le code de ces 2 fonctions en C++ dans le fichier native-lib.cpp. Utiliser la classe LampeMulticolore comme dans votre projet en C++.

## Gestion de la brillance

Dans l'interface graphique (activity\_main.xml) :

Ajouter un champ de type texte (Plain Text) permettant de saisir la valeur de la brillance (entre 0 et 65535) : id : editTextBrillance, text : 30000.

Ajouter un bouton (id : buttonBrillance, text : Brillance) permettant d'appliquer cette valeur.

Dans la classe d'interface MainActivity (MainActivity.java)

Dans la classe d'interface MainActivity (MainActivity.java), ajouter la déclaration de la fonction native : ModifierBrillanceLampeMulticoloreA6(String valeur).

Dans le fichier native-lib.cpp :

Écrire le code de cette fonction en C++ dans le fichier native-lib.cpp. Utiliser la classe LampeMulticolore. Vous aurez également besoin de la remarque suivante :

Au début de cette fonction, il sera nécessaire de convertir la chaîne de caractères valeur de type jstring passée en argument en un objet de la classe string. C'est ce que fait l'instruction suivante :

```
string valeurBrillance = env->GetStringUTFChars(valeur, NULL);
```

Pour finir, dans la classe d'interface MainActivity (MainActivity.java) :

Gérer le clic sur le bouton `buttonBrillance` puis : Récupérer la valeur la brillance saisie dans le `editTextBrillance` et appeler la fonction native `ModifierBrillanceLampeMulticoloreA6()` en lui passant en argument la valeur saisie.

## Gestion de l'intensité

Dans l'interface graphique (`activity_main.xml`) :

Ajouter un champ de type texte (Plain Text) permettant de saisir la valeur de l'intensité (entre 0 et 255) : `id : editTextIntensite`, `text : 255`.

Ajouter un bouton (`id : buttonIntensite`, `text : Intensité`) permettant d'appliquer cette valeur.

Dans la classe d'interface `MainActivity` (`MainActivity.java`)

Dans la classe d'interface `MainActivity` (`MainActivity.java`), ajouter la déclaration de la fonction native : `ModifierIntensiteLampeMulticoloreA6(String valeur)`.

Dans le fichier `native-lib.cpp` :

Écrire le code cette fonction en C++ dans le fichier `native-lib.cpp`. Utiliser la classe `LampeMulticolore`. Vous aurez également besoin de la remarque suivante :

Au début de cette fonction, il sera nécessaire de convertir la chaîne de caractères `valeur` de type `jstring` passée en argument en un objet de la classe `string`. C'est ce que fait l'instruction suivante :

```
string valeurIntensite = env->GetStringUTFChars(valeur, NULL);
```

Pour finir, dans la classe d'interface `MainActivity` (`MainActivity.java`) :

Gérer le clic sur le bouton `buttonIntensite` puis : Récupérer la valeur l'intensité saisie dans le `editTextIntensite` et appeler la fonction native `ModifierIntensiteLampeMulticoloreA6()` en lui passant en argument la valeur saisie.

## Bonus

- Changer les champs de saisie texte par des slide bar.
- N'utiliser qu'un seul bouton pour allumer et éteindre la lumière (sa valeur change en fonction de l'état de la lumière)