MODULE 03

Domotique Zigbee DIY

Gaëtan GEORGES (g2.iris.lla@gmail.com)
Pascal HAGOT (pascal.hagot@ac-creteil.fr)
David IERVESE (d_iervese@yahoo.com)

Lycée Louis Armand
173 Bd de Strasbourg
Nogent sur Marne

MODULE 03
SÉANCE WEB 01
TP D'INFORMATIQUE
Durée 2h30

LA PAGE D'ACCUEIL

UNITÉ CERTIFICATIVE
U6 - VALORISATION DE LA DONNÉE ET CYBERSÉCURITÉ

COMPÉTENCE(S)
C08 - CODER

OBJECTIF PÉDAGOGIQUE
Création de 1 pages HTML contenant la présentation de la domotique ainsi que la mise en page par du CSS

CONNAISSANCES ISSUES DU RÉFÉRENTIEL

• Langages de développement, de description, de création d'API et les IDE associés	Niveau 4
• Chaînes de développements (ordinateur, embarqué, cross compilation)	Niveau 3

CONNAISSANCES OPÉRATIONNALISÉES

• Codage d'une page HTML avec une feuille de style CSS	Niveau 2

Gaëtan GEORGES
Pascal HAGOT
David IERVESE

g2.iris.lla@gmail.com
pascal.hagot@ac-creteil.fr
d_iervese@yahoo.com

Lycée Louis Armand
173 bd de Strasbourg
Nogent sur Marne

TP
Mise en place du versionning avec git

Prérequis

Installation de Visual Studio Code

Installation de Git pour Windows

Installation de SSHFS-Win Manager pour Windows

> Si ces outils ne sont pas installés sur votre poste de travail, les installer.

Préparation des outils de travail pour le versionning

> Connecter votre poste windows à votre serveur web en utilisant SSHFS-win Manager
>
> Créer un répertoire nommé M03SW sur votre serveur web
>
> Ouvrir ce répertoire avec Visual Studio Code (Clic droit, Ouvrir avec VS Code)
>
> Créer un fichier README contenant le texte suivant :

```
Module 3 Web
Gestion de modules domotique.
```

> Aller sur l'outil git puis cliquer sur le bouton initialiser un dépôt pour créer un dépôt local (le répertoire (.git) permettra de gérer le versionning du projet. Par défaut, une branche « master » a été créée.
>
> Ajouter le fichier README aux fichiers indexés.
>
> Puis effectuer votre premier commit sur la branche master en validant tous les changements (ceci crée un snapshot de l'état actuel de votre projet). Vous pouvez précisez le texte suivant : « premier commit avec le fichier README ».
>
> Créer une branche SW01

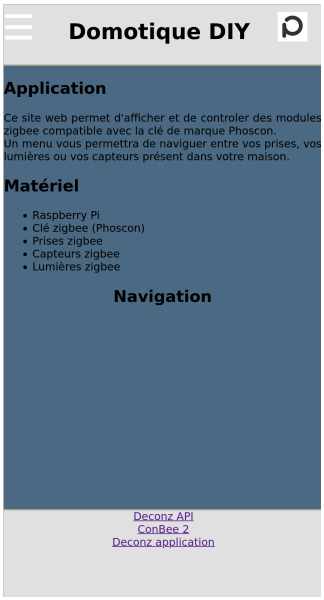
C'est dans la branche SW01 que nous travaillerons tout au long de cette séance.

Défi 1
Création de la page de présentation

Objectif

L'objectif de cette partie est d'écrire la page de présentation en HTML.

Le résultat final ressemblera à ceci :



Première partie : le code HTML de la page d'accueil

Créer un fichier nommé index.html dans votre répertoire de travail.

Créer une structure de base d'un fichier HTML comme suit :

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Deconz</title>
</head>
<body>
    <nav id="nav">

    </nav>
    <header id="header">

    </header>

    <section id="section">

    </section>

    <footer id="footer">

    </footer>
</body>
</html>
```

Compléter les différentes balises du code précédent afin de faire apparaître le contenu du résultat final. Chaque élément de la barre de navigation sera dans une balise « div » ayant une classe « nav », le text, quand à lui, sera dans une balise span.

Compléter les images du header avec des identifiants ayant comme valeur « imageHamburger » et «imageDomoticz».

[GIT] Indexer le fichier index.html sur la branche SW01.

[GIT] Faire un commit avec le message : « Le code HTML de la page d'accueil»

Deuxième partie : le code CSS de la page d'accueil

Créer le fichier style.css dans votre répertoire de travail

Compléter le fichier style.css en respectant, entre autres, les paramètres suivants

- **body** : alignementdu text central, largeur de « 100vw »
- **header** : largeur de « 100% », une hauteur de « 10vh »
- **images du header** : flottant gauche ou droite, marges extérieures de 10px
- **nav** : hauteur de « 100vh », affichage desactivé, largeur de « 50% », position absolue, un « z-index » de 10
- **section** : hauteur de « 75vh », une largeur de « 100% »
- **footer** : hauteur de « 15vh », une largeur de « 100% », une position fixe, une valeur basse de 0

Vérifier la mise en page en faisant apparaître la barre de navigation en modifiant son affichage en « block »

[GIT] Indexer le fichier style.css sur la branche SW01.

[GIT] Faire un commit avec le message : « Le code HTML+CSS de la page d'accueil»

Git de fin de séance : merger la branche SW01 sur la branche master

[GIT] Se placer sur la branche master et faire un « Merge Branch » de la branche SW01

La branche master contient alors maintenant le code de la branche SW01

[GIT] Publier la branche master sur GitHub

MODULE 03
SÉANCE WEB 02
TP D'INFORMATIQUE
Durée 2h30

Initiation à l'AJAX

UNITÉ CERTIFICATIVE
U6 - VALORISATION DE LA DONNÉE ET CYBERSÉCURITÉ

COMPÉTENCE(S)
C08 - CODER

OBJECTIF PÉDAGOGIQUE
Comprendre et tester l'échange d'une requête HTTP par le javascript

CONNAISSANCES ISSUES DU RÉFÉRENTIEL

• Langages de développement, de description, de création d'API et les IDE associés	Niveau 4
• Chaînes de développements (ordinateur, embarqué, cross compilation)	Niveau 3
• Outils de documentation	Niveau 3

CONNAISSANCES OPÉRATIONNALISÉES

Gaëtan GEORGES
Pascal HAGOT
David IERVESE

g2.iris.lla@gmail.com
pascal.hagot@ac-creteil.fr
d_iervese@yahoo.com

Lycée Louis Armand
173 bd de Strasbourg
Nogent sur Marne

TP
Gestion du hamburger

Objectif :

L'objectif de cette partie est de comprendre la gestion des évènements en JS. Un évènement est déclenché généralement suite à une action de l'utilisateur : un clic sur un bouton, une saisie d'un texte, etc. Lorsqu'un évènement est déclenché, il est possible d'appeler une fonction.

Preparation GIT

> [GIT] Créer la branche SW02

Gestion des évènements

En JS, la gestion d'un évènement se fait en 2 étapes :

1. On récupère l'objet sur lequel l'utilisateur interagit
2. On indique quel évènement de cet objet nous intéresse et la fonction qui sera exécutée.

Un exemple

Supposons que nous souhaitons afficher un message popup lorsque la souris de l'utilisateur survole le footer de notre page. La première étape sera de récupérer l'objet (notre footer) avec la méthode getElementById() comme dans l'exemple suivant (p_footer est l'id du footer) :

```
var p_footer = document.getElementById("p_footer");
```

Ensuite, nous allons indiquer quel évènement sur cet objet nous intéresse et quelle fonction sera appelée :

```
p_footer.addEventListener('mouseover', mon_popup);
```

Il reste maintenant à créer la fonction mon_popup() comme suit :

```
function mon_popup() {
  alert("Gestion de l'évènement 'mouse over' sur mon footer");
}
```

Afficher le menu

> Donner le code permettant de gérer l'évènement 'click' sur l'image du hamburger (son id est 'imageHamburger') et exécuter la fonction AfficherMenu().

```

```

> Créer un fichier fonction.js
> Créer la fonction AfficherMenu(). Compléter le code de cette fonction qui changera le style de l'affichage de la barre de navigation en « block »
> Ajouter le code gérant l'événement du click sur le hamburger après la fonction

En JS, la méthode *getElementById()* permet de récupérer un objet du corps de la page connaissant son identifiant. La méthode *addEventListener()* permet de gérer les évènements d'un objet.

En JS, le mot-clé `function` permet de créer une nouvelle fonction avec un nom et si nécessaires des arguments. Ensuite, la définition de la fonction se fait entre les accolades ouvrante et fermante.

Fermer le menu

Donner le code permettant degérer l'évènement 'click' sur le texte « Fermer » du menu (son id est 'fermer') et exécuter la fonction FermerMenu().

```
function FermerMenu(){
   console.log('MenuFermer');
   document.getElementById("nav").style.display = "none";
}
```

Créer la fonction FermerMenu(). Compléter le code de cette fonction qui changera le style de l'affichage de la barre de navigation en « none »

Ajouter le code gérant l'événement du click sur le texte « Fermer » après la fonction

[GIT] Indexer les fichiers sur la branche SW02.

[GIT] Faire un commit avec le message : « Gestion du hamburger»

Préparation de la suite

Pour les données météorologiques, nous allons utiliser openweathermap

Se créer un compte sur openweathermap

Vérifier de bien avoir une clé « API Key » dans votre profil vous permettant de récupérer la météo. Cela peut prendre du temps alors passer à la suite en attendant.

TD
Comprendre l'AJAX

Dans cette partie nous verrons comment modifier une partie d'un site en utilisant le javascript et l'AJAX à partir des données chargées depuis le serveur au format JSON.

Définitions

Que signifie l'acronyme AJAX ?

Asynchronous JavaScript and XML

Quelle est la différence entre un processus synchrone et asynchrone ?

Synchrone: lit tout dans un même temps
Asynchrone: lit tout dans différent temps tels que des threads

Citer 3 différents formats de données pouvant être échangé entre un serveur et le client

CSV XML JSON

Analyse de code

Voici un exemple de fonction AJAX en javascript permettant d'interroger le serveur :

```
function ajax() {
  const xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      var reponse=this.reponseText;
      console.log(reponse);
    }
  };
  xhttp.open(method, url);
  xhttp.send();
}
```

Quatre principales étapes sont nécessaires pour l'échange d'information :

1. Initialiser un objet XMLHttpRequest
2. Spécifier la méthode et l'URL ciblé
3. Envoyer la demande
4. Action à faire lorsque la réponse est reçue

Situer la ou les lignes de codes pour chaque étape précédente.

1 const xhttp
2. xhttp.open
3 xhttp.send
4. console.log(reponse)

A quoi sert l'objet « readyState » ? Quelles sont les différentes valeurs possibles ? Donner leurs significations ?

C'est un proprieté qui permet de suivre le temps de chargement d'un document ou d'une requete
Il y a 5 valeur possible allant de 0 a 4: unsent openned headers_received loading et done
Unsent : client créé. opened : open() a été appelé. HEaders_received send() a été appelé.
Loading : le fichier se télécharge. Done : opération complete

A quoi sert l'objet «status» ? Donner la signification des valeurs 200, 403, 404.

Status permet de determiner le résultat de la requete d'un client a un serveur.
200: La requete est un succès.
403 : L'acces au fichier demandé est interdit
404: Le fichier demandé est introuvable

Quelles sont les différentes valeurs possibles de la variable « method » ?

GET POST PUT DELETE

TD
Comprendre le JSON

Structure JSON

Que signifie l'acronyme JSON ?

Javascript object notation

Par quel caractère doit commencer un structure de données JSON ?

{ }

Par quel caractère doit commencer un tableau de données JSON ?

[ ]

Quels sont les différents type de valeurs que peut prendre un objet JSON ?

chaien de cractere nombre objet tableau et boleen

Donner un exemple de structure JSON

```
{
  "nom": "Jean",
  "ville": "Paris"
}
```

Exploitation du JSON

Quelle fonction javascript permet de convertir une chaîne de caractère JSON en un objet JSON ?

JSON.parse ()

Exemple de réponse JSON du serveur openweathermap à une requête du client :

```
{
  "coord": {
    "lon": 2.3488,
    "lat": 48.8534
  },
  "weather": [
    {
      "id": 801,
      "main": "Clouds",
      "description": "peu nuageux",
      "icon": "02d"
    }
  ],
  "base": "stations",
  "main": {
    "temp": 22.23,
```

```
    "feels_like": 22.34,
    "temp_min": 21.34,
    "temp_max": 23.32,
    "pressure": 1010,
    "humidity": 70
  },
  "visibility": 10000,
  "wind": {
    "speed": 4.63,
    "deg": 170
  },
  "clouds": {
    "all": 20
  },
  "dt": 1655974695,
  "sys": {
    "type": 2,
    "id": 2041230,
    "country": "FR",
    "sunrise": 1655956045,
    "sunset": 1656014283
  },
  "timezone": 7200,
  "id": 2988507,
  "name": "Paris",
  "cod": 200
}
```


La réponse précédente sera stockée dans une variable JS « reponseMeteo ». Pour pourvoir utiliser les données d'un format JSON, il faut les convertir en un objet javascript. Cette conversion sera stockée dans une variable « reponseMeteoObject ».

Donner le code permettant de convertir la chaîne de caractère JSON précédente en un objet JS « reponseMeteoObject »

```
toString()
```

En utilisant la variable reponseMeteoObject , Donner le code permettant de :

Connaître le nom de la ville

```
reponseMeteoObject(city)
```

Connaître le pays

```
reponseMeteoObject(contry)
```

Connaître la température

```
reponseMeteoObject(deg)
```

Connaître la distance de visibilité

```
reponseMeteoObject(visibility)
```

Quel est l'unité de la vitesse du vent ?

```
En km/h
```

Quel est l'unité de la température ?

```
Fareinheit
```

TP
Récupérer des données météorologiques

Openweathermap API

Nous utiliserons dans ce module la version 2.5 de openweathermap

En vous aidant de la documetation (https://openweathermap.org/current), quelle requête HTTP faut-il envoyer pour récupérer la météo actuelle au format JSON, en français, aux unités métriques, de la ville de Paris.

```
https://api.openweathermap.org/data/2.5/weather?
lat=48.86&lon=2.34&appid=5e3224bd40937b9f365f31fef4fa61c9
```

Pour tester la requête HTTP, nous allons utiliser une extension d'un navigateur.

Ouvrez votre navigateur et installer une extension « client rest »
Tester la requête précédente dans un navigateur en y intégrant votre clé API.

Quel paramètre nous permet d'extraire la valeur de la température, la vitesse du vent, la visibilité et l'icone ?

```
console.log(test.main.temp)
test.main.speed
test.visibility
```

Quelles sont les unités des valeurs précédentes ?

```
temp : kelvin
speed : m/s
visibility : valeur maximal de visibilité sur 10km
```

Quelle requête faut il envoyer pour récupérer l'icône correspondante ?

```
console.log(test.weather['0'].icon)
```

Index.html et style.css

Ajouter en bas de la barre de navigation une division avec les attibuts id=meteo et class=meteo
Dans la division précédente, il faut ajouter des « span » pour la ville, la température et l'icone
Créer des attributs « id » et « class » pour les divisions précédentes (ville_meteo, temperature, icone_meteo). Nous utiliserons l'id avec le javascript et la classe pour la mise en page
Modifier le design (style.css) pour avoir un affichage convenable.

Javascript

Créer le fichier ajax.js
Créer la fonction « RecupererMeteo » en y intégrant le code de base AJAX pour une requête HTTP
Modifier dans la fonction précédente la méthode et l'URL afin de récupérer la météo actuelle de Paris
Appeler la fonction « RecupererMeteo » lorsque l'on fait afficher la barre de navigation
Vérifier le contenu de la réponse dans la console de votre navigateur

[GIT] Indexer les fichiers sur la branche SW02.
[GIT] Faire un commit avec le message : « Initialisation ajax»

Git de fin de séance : merger la branche SW02 sur la branche master

[GIT] Se placer sur la branche master et faire un « Merge Branch » de la branche SW02

La branche master contient alors maintenant le code de la branche SW02

[GIT] Publier la branche master sur GitHub

MODULE 03
SÉANCE WEB 03
TP D'INFORMATIQUE
Durée 2h30

Exploitation des données météorologiques

UNITÉ CERTIFICATIVE
U6 - VALORISATION DE LA DONNÉE ET CYBERSÉCURITÉ

COMPÉTENCE(S)
C06 – VALIDER UN SYSTÈME INFORMATIQUES
C09 – INSTALLER UN RÉSEAU INFORMATIQUE

OBJECTIF PÉDAGOGIQUE
Analyser et exploiter des données d'une API openweathermap afind 'afficher la météo sur le site web.

CONNAISSANCES ISSUES DU RÉFÉRENTIEL
- Langages de développement, de description, de création d'API et les IDE associés — Niveau 4
- Chaînes de développements (ordinateur, embarqué, cross compilation) — Niveau 3
- Outils de documentation — Niveau 3

CONNAISSANCES OPÉRATIONNALISÉES
- Analyse Wireshark d'une communication HTTP — Niveau 2

Gaëtan GEORGES
Pascal HAGOT
David IERVESE

g2.iris.lla@gmail.com
pascal.hagot@ac-creteil.fr
d_iervese@yahoo.com

Lycée Louis Armand
173 bd de Strasbourg
Nogent sur Marne

TD
Extraction de données JSON

Préparation GIT

[GIT] Créer la branche SW03

Compréhension

Exemple de réponse JSON du serveur openweathermap à une requête du client :

```
{
  "coord": {
    "lon": 2.3488,
    "lat": 48.8534
  },
  "weather": [
    {
      "id": 801,
      "main": "Clouds",
      "description": "peu nuageux",
      "icon": "02d"
    }
  ],
  "base": "stations",
  "main": {
    "temp": 22.23,
    "feels_like": 22.34,
    "temp_min": 21.34,
    "temp_max": 23.32,
    "pressure": 1010,
    "humidity": 70
  },
  "visibility": 10000,
  "wind": {
    "speed": 4.63,
    "deg": 170
  },
  "clouds": {
    "all": 20
  },
  "dt": 1655974695,
  "sys": {
    "type": 2,
    "id": 2041230,
    "country": "FR",
    "sunrise": 1655956045,
    "sunset": 1656014283
  },
  "timezone": 7200,
  "id": 2988507,
  "name": "Paris",
  "cod": 200
}
```

Pour récupérer la longitude et la latitude, il faut alors utiliser le code suivant :

```
var latitude = reponseMeteoObject.coord.lat;
```

Écrire le code permettant de récupérer la longitude, la température, la vitesse du vent, la visibilité et le taux d'humidité ainsi que le nom de la ville.

```
var longitude = reponseMeteoObject.coord.lon;var ten
```

Certaines valeurs peuvent être stockées dans un tableau d'objet. Ainsi pour la description de la météo, il faut utiliser le code suivant :

```
var description = reponseMeteoObject.weather[0].description;
```

Écrire le code permettant de récupérer l'icône de la météo

```
var icone = reponseMeteoObject.weather['0'].icon
```

Afficher les données du site

En vous aidant du TD précédent compléter la fonction ajaxMeteo() avec le code JS permettant d'extraire les données d'un format JSON et de les enregistrer dans les variables « temperature », « icone », « longitude » et « latitude » et « ville ».

Compléter le code en affichant les valeurs de la ville, la température,

Ajouter l'icône en modifiant l'image de fond de votre division « meteo »

Modifier le design de votre site si besoin

[GIT] Indexer les fichiers sur la branche SW03.

[GIT] Faire un commit avec le message : « Affichage des données météorologiques »

Git de fin de séance : merger la branche SW02 sur la branche master

[GIT] Se placer sur la branche master et faire un « Merge Branch » de la branche SW03

La branche master contient alors maintenant le code de la branche SW03

[GIT] Publier la branche master sur GitHub

MODULE 03
SÉANCE WEB 04
TP D'INFORMATIQUE
Durée 2h30

Analyser l'API deconz

BLOC DE COMPÉTENCES
U6 – VALORISATION DE LA DONNÉE ET CYBERSÉCURITÉ

COMPÉTENCE(S)
C08 – CODER

OBJECTIF PÉDAGOGIQUE
Etudier la documentation de l'API permettant d'interroger la passerelle zigbee utilisant le logiciel deconz.

CONNAISSANCES ISSUES DU RÉFÉRENTIEL

• Langages de développement, de description, de création d'API et les IDE associés	Niveau 4
• Chaînes de développements (ordinateur, embarqué, cross compilation)	Niveau 3

CONNAISSANCES OPÉRATIONNALISÉES

• Codage d'un formulaire HTML adapté à un code PHP donné	Niveau 2
• Analyse d'un code PHP de récupération des input HTML	Niveau 2

Gaëtan GEORGES
Pascal HAGOT
David IERVESE

g2.iris.lla@gmail.com
pascal.hagot@ac-creteil.fr
d_iervese@yahoo.com

Lycée Louis Armand
173 bd de Strasbourg
Nogent sur Marne

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 ?

La methode get permet de demander une informaton au serveur

A quoi sert la ressource (URL) ?

La ressource url permet de rechercher l'info demandé a l'url

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.

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

GET : Pour lire des ressources sur le serveur REST

POST : Pour creer des nouvelles ressources sur le serveur rest

PUT pour mettre a jours des ressourcesd sur le sevreur rest

DELETE pour supprimer des ressources sur le serveur rest

Quand à elle, la ressource permet de définir la demande de l'utilisateur.

TD
L'API deconz

Objectif

Comprendre l'API utilisée par deconz
(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 ?

ils doivent acceder a l'api local ou a la telecommande depuis le réseau

Quels sont les avantages à utiliser l'API REST ?

Accessible depuis n'importe quel appareil et langage de prog. Le format des requetes est intelligibleL'

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 172.20.21.57

Obtention d'une clé API

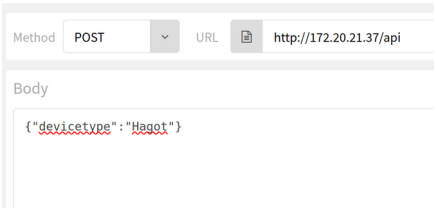
Donner les requêtes HTTP permettant de trouver l'adresse ip de la passerelle

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

```
http://172.20.21.57/api                    devicetype :
```

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 :



Ouvrez wireshark et vérifier l'envoie de la requête HTTP et la réception de la réponse. Quelle est la durée de la transaction (entre le l'envoie 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 editer → Fixer le temps de référence

```
Quasiment inst
```

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 ?

```
GET  /api/cléapi/de
```

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 ?

```
GET /api/cléapi/devices/adr
```

Exécuter la requête et noter le nom du fabriquant du 3ème élément

LEDVANCE

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

4 elements

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

Certains ont des parametres differents

Quel paramètre permet de faire la différence entre les éléments ?

l'etat de l'element

Quel est sa valeur pour les lumières ?

Cela depend de la la lumiere, mais ça peut etre alumé ou encore voir la couleur de celle ci

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

19

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

Oui car certains on plusieurs parametre en plus que les autres

Modifier un élément

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

GET /api/<apikey>/lights/<id>

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

PUT /api/<apikey>/lights/<id>/state

{"on":"false"}

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

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

```
http://172.20.21.100/api/F276DD7951/lights/00:17:88:01:0b:df:d8:f6-0b/state
{"on":false}
```

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

MODULE 03
SÉANCE WEB 05
TP D'INFORMATIQUE
Durée 2h30

Gestion de lumières connectées

UNITÉ CERTIFICATIVE
U6 – VALORISATION DE LA DONNÉE ET CYBERSÉCURITÉ

COMPÉTENCE(S)
C08 – CODER

OBJECTIF PÉDAGOGIQUE
Coder la page web permettant de récupérer et de commander de lumières zigbee.

CONNAISSANCES ISSUES DU RÉFÉRENTIEL

• Langages de développement, de description, de création d'API et les IDE associés	Niveau 4
• Chaînes de développements (ordinateur, embarqué, cross compilation)	Niveau 3

CONNAISSANCES OPÉRATIONNALISÉES

• Mise en page d'une page web avec les grilles CSS.	Niveau 2

Gaëtan GEORGES
Pascal HAGOT
David IERVESE

g2.iris.lla@gmail.com
pascal.hagot@ac-creteil.fr
d_iervese@yahoo.com

Lycée Louis Armand
173 bd de Strasbourg
Nogent sur Marne

TD
Analyser la réponse de l'API deconz

Objectif

Décortiquer et comprendre le contenu de la réponse de l'API deconz des lumières

Exemple de réponse

Voici un exemple de réponse à la requête

```
GET http://[ip]/api/[apikey]/lights
```


```
{
    "1": {
        "etag": "026bcfe544ad76c7534e5ca8ed39047c",
        "hascolor": true,
        "manufacturer": "dresden elektronik",
        "modelid": "FLS-PP3",
        "name": "Light 1",
        "pointsymbol": {},
        "state": {
            "alert": "none",
            "bri": 111,
            "colormode": "ct",
            "ct": 307,
            "effect": "none",
            "hue": 7998,
            "on": false,
            "reachable": true,
            "sat": 172,
            "xy": [ 0.421253, 0.39921 ]
        },
        "swversion": "020C.201000A0",
        "type": "Extended color light",
        "uniqueid": "00:21:2E:FF:FF:00:73:9F-0A"
    },

    "2": {
        "etag": "026bcfe544ad76c7534e5ca8ed39047c",
        "hascolor": false,
        "manufacturer": "dresden elektronik",
        "modelid": "FLS-PP3 White",
        "name": "Light 2",
        "pointsymbol": {},
        "state": {
            "alert": "none",
            "bri": 1,
            "effect": "none",
            "on": false,
            "reachable": true
        },
        "swversion": "020C.201000A0",
        "type": "Dimmable light",
        "uniqueid": "00:21:2E:FF:FF:00:73:9F-0B"
    }
}
```

Chaque élément de notre maison connecté est répertorié, dans un format JSON, par un numéro : ici « 1 » et « 2 » car il y a deux éléments.

Supposons que la chaine de caractères précédente soit stockée dans la variable « reponseLumiere ». Cette chaîne de caractère est convertir en un objet javascript nommé « objetLumiere ».
Voici le code que nous utiliserons

```
var lumiere='' ;
for(num in objetLumiere){
        let uniqueid=objetLumiere[num].uniqueid;
        let etat=objetLumiere[num].state.on;
        let type=objetLumiere[num].type;
        lumiere=lumiere+'<div id="'+uniqueid+'" >';
        lumiere+="</div>";
}
```

Donner les valeurs de chaque variable en utilisant les données précédentes

uniqueid = 00:21:2E:FF:FF:00:73:9F-0A

etat = false

type = Extended color light

2

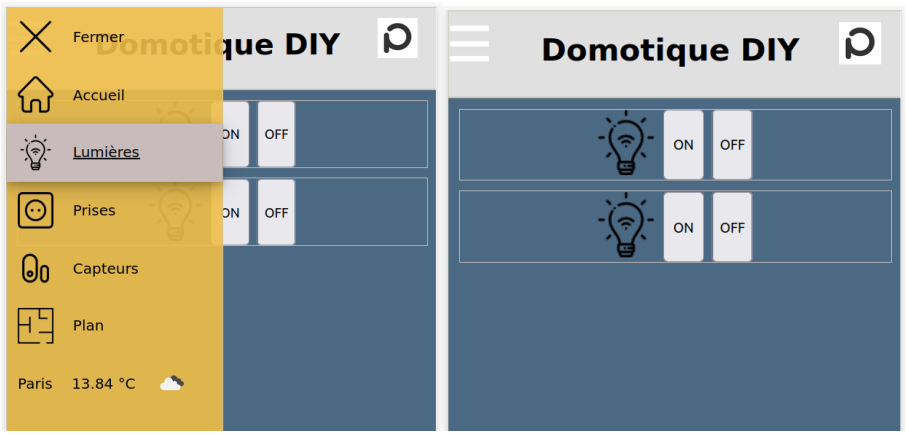
unique id = 00:21:2E:FF:FF:00:73:9F-0B

etat= false

type = Diammable light

TP
Afficher la liste des lumières

Objectifs



Préparation GIT

[GIT] Créer la branche SW05

Menu de navigation

Ajouter un identifiant à chaque division de votre menu de navigation : lumiereNAV, capteurNAV, priseNAV, planNAV

Modifier le fichier fonction.js afin de rendre les liens du menu cliquable vers les fonctions : AfficherLumiere, AfficherPrise, AfficherCapteur

Structure de base des lumières

Créer une fonction JS AfficherLumiere()

Ajouter le code AJAX de base en modifiant la méthode et la requête afin d'interroger les lumières

Vérifier le bon fonctionnement en affichant la réponse dans la console du navigateur.

[GIT] Indexer tous les fichiers modifiés sur la branche SW05.

[GIT] Faire un commit avec le message : « Mise en place de la fonction AfficherLumiere »

Afficher les lumières

La suite va nous permettre de compléter la fonction AfficherLumiere() avec les données de la domotique. La suite sera placée dans zone après la condition « if (this.readyState == 4 && this.status == 200) »

Créer une variable « section » contenant les informations de l'élément ayant comme identifiant « section »

Compléter avec l'initialisation de la variable lumière à une chaine de caractère vide.

Ajouter le code permettant de faire une boucle sur les différents éléments (voir TD précédent)

Compléter la variable « lumiere » avec la structure suivante :

```
lumiere= lumiere+'<img src="./icones/lightbulb.png" alt="lumiere"><button id="on">ON</button><button id="off">OFF</button>';
```

La variable précédente contient alors une structure HTML. Pour insérer un contenu HTML à partir d'un code JS nous utiliserons l'objet innerHTML.
Exemple de code :

```
document.getElementById('division').innerHTML='<p> Un paragraphe <p>';
ou
var param='<p> Un paragraphe <p>';
document.getElementById('division').innerHTML=param;
```

Expliquer le code précédent ?

on rajoute un contenu html de type paragraphe et on le met dans la div avec l'id division

Donner le code permettant d'afficher le contenu de la variable « lumiere » dans l'identifiant section de la page web

document.getElemeentbyId('section').innerHTML = lumiere;

Compléter votre fichier javascript avec le code précédent
Modifier le code CSS si besoin

Pour n'afficher que les lumières, nous allons ajouter une condition sur la valeur de type.

Ajouter dans la boucle for la condition permettant de ne visualiser que les lumières

[GIT] Indexer tous les fichiers modifiés sur la branche SW05.
[GIT] Faire un commit avec le message : « Affichage des lumières »

Interagir avec les lumières

Compléter les boutons « on » et « off » avec les attributs « onclick » ciblant les fonctions AllumerLumiere(num) et EteindreLumiere(num). Attention « num » est la variable de la boucle for à intégrer dans les balises des boutons.
Créer les fonctions précédentes
Les compléter avec le code de base AJAX en adaptant la méthode et l'URL afin d'allumer ou d'éteindre la lumière

[GIT] Indexer tous les fichiers modifiés sur la branche SW05.
[GIT] Faire un commit avec le message : «Interaction avec les lumières »

Git de fin de séance : merger la branche SW05 sur la branche master

[GIT] Se placer sur la branche master et faire un « Merge Branch » de la branche SW05
La branche master contient maintenant le code de la branche SW05.
[GIT] Publier la branche master sur GitHub

MODULE 03
SÉANCE WEB 05
TP D'INFORMATIQUE
Durée 2h30

Gestion de prises connectées

UNITÉ CERTIFICATIVE
U6 – VALORISATION DE LA DONNÉE ET CYBERSÉCURITÉ

COMPÉTENCE(S)
C08 – CODER

OBJECTIF PÉDAGOGIQUE
Contrôler des prises domotiques en interagissant avec une API REST

CONNAISSANCES ISSUES DU RÉFÉRENTIEL

• Langages de développement, de description, de création d'API et les IDE associés	Niveau 4
• Chaînes de développements (ordinateur, embarqué, cross compilation)	Niveau 3

CONNAISSANCES OPÉRATIONNALISÉES
•

Gaëtan GEORGES
Pascal HAGOT
David IERVESE

g2.iris.lla@gmail.com
pascal.hagot@ac-creteil.fr
d_iervese@yahoo.com

Lycée Louis Armand
173 bd de Strasbourg
Nogent sur Marne

TP
Afficher la liste des prises

Objectifs

Créer un lien de menu permettant d'afficher la liste des prises

Préparation GIT

[GIT] Créer la branche SW06

Structure de base des prises

Créer une fonction JS AfficherPrise()
Ajouter le code AJAX de base en modifiant la méthode et la requête afin d'interroger les prises
Vérifier le bon fonctionnement en affichant la réponse dans la console du navigateur.

[GIT] Indexer tous les fichiers modifiés sur la branche SW06.
[GIT] Faire un commit avec le message : « Mise en place de la fonction AfficherPrise»

Afficher les prises

La suite va nous permettre de compléter la fonction AfficherPrise() avec les données de la domotique. La suite sera placée dans zone après la condition « if (this.readyState == 4 && this.status == 200) »

Créer une variable « section » contenant les informations de l'élément ayant comme identifiant « section »
Compléter avec l'initialisation de la variable prise à une chaine de caractère vide.
Ajouter le code permettant de faire une boucle sur les différents éléments (voir TD précédent)
Compléter la variable «prise» avec la structure suivante :

```
prise= prise+'<img src="./icones/plugin.png" alt="prise"><button
id="on">ON</button><button id="off">OFF</button>';
```

Donner le code permettant d'afficher le contenu de la variable «prise» dans l'identifiant section de la page web
Compléter votre fichier javascript avec le code précédent
Modifier le code CSS si besoin

Pour n'afficher que les prise, nous allons ajouter une condition sur la valeur de type.

Ajouter dans la boucle for la condition permettant de ne visualiser que les prises

[GIT] Indexer tous les fichiers modifiés sur la branche SW06.
[GIT] Faire un commit avec le message : « Affichage des prises»

Interagir avec les prises

Compléter les boutons « on » et « off » avec les attributs « onclick » ciblant les fonctions AllumerPrise(num) et EteindrePrise(num). Attention « num » est la variable de la boucle for à intégrer dans les balises des boutons.

Créer les fonctions précédentes

Les compléter avec le code de base AJAX en adaptant la méthode et l'URL afin d'allumer ou d'éteindre la prise

[GIT] Indexer tous les fichiers modifiés sur la branche SW06.

[GIT] Faire un commit avec le message : «Interaction avec les prises »

Git de fin de séance : merger la branche SW06 sur la branche master

[GIT] Se placer sur la branche master et faire un « Merge Branch » de la branche SW06

La branche master contient maintenant le code de la branche SW06.

[GIT] Publier la branche master sur GitHub

MODULE 03
SÉANCE WEB 07
TP D'INFORMATIQUE
Durée 2h30

Installation et configuration d'une raspberry pi

UNITÉ CERTIFICATIVE
U5 – EXPLOITATION ET MAINTENANCE DE RÉSEAUX INFORMATIQUES

COMPÉTENCE(S)
C09 – INSTALLER UN RÉSEAU INFORMATIQUE

OBJECTIF PÉDAGOGIQUE
Installation de la Raspberry : Raspberry PI OS / Paramétrage réseau / Création des utilisateurs et gestions des droits sudo

CONNAISSANCES ISSUES DU RÉFÉRENTIEL
- Systèmes d'exploitations (Windows, UNIX, Virtualisation) Niveau 3

CONNAISSANCES OPÉRATIONNALISÉES
- Installer un OS Linux Niveau 2

Gaëtan GEORGES
Pascal HAGOT
David IERVESE

g2.iris.lla@gmail.com
pascal.hagot@ac-creteil.fr
d_iervese@yahoo.com

Lycée Louis Armand
173 bd de Strasbourg
Nogent sur Marne

Préparation

Objectif

Créer une passerelle zigbee dans le cadre de la domotique.

Pré-requis

La passerelle zigbee sera crée à l'aide d'une raspberry et d'une clé Conbee II de la marque Phoscon

A cela nous allons ajouter des objets connectés compatible avec le protocole zigbee
En voici la liste

- Capteur de température AQARA
- Capteur de présence AQARA
- Capteur d'ouverture de porte AQARA
- Smart+ plug Ledvance
- Smart+ Ledvance Gardenpole
- Raspberry pi

Préparation

Rechercher et télécharger les documentations ou guides d'installation des éléments précédents

Rechercher la procédure d'installation de la clé Conbee II sur une raspberry

Faire un document résumant les étapes d'installation de la raspberry en précisant si besoin les logiciels utilisés ainsi que la procédure de réinitialisation des Gardenpole qui sera remise au client.

1. Télécharger l'image du système d'exploitation voulu, alias Rasbian Buster Desktop sur le site de conbee.

2 Rendre la carte SD bootable pour installer l'os sur la Raspberry via le logiciel balenaEtcher fourni par conbee, en mettant l'os préalablement installé

3. Insérer la carte SD dans la Raspberry avec les périphériques branché et mettre sous alimentation la Raspberry.

Pour réinitialisera les GardenPole il suffit de le débrancher pendant 5 secondes puis de le brancher pendant 5 secondes tout cela 5 fois

Installation OS Deconz

Installer un l'OS Rasbian pour faire de la domotique en utilisant l'application Deconz de phosocon

Installation de l'OS sur une carte SD

Suivez le tutoriel disponible sur la page dédiée du site officiel de la Raspberry pour installer le système d'exploitation : https://www.phoscon.de/en/conbee2/sdcard

Une fois la carte SD correctement installée, insérer la dans votre Raspberry. Ajouter un écran, une souris et un clavier, puis démarrer la Raspberry en l'alimentant.

Configuration du réseau

Une carte réseau peut être configurée de 2 façons :

1. par une configuration automatique, c'est la configuration dynamique,
2. par une configuration manuelle, c'est la configuration statique.

Par défaut, la Raspberry est en mode dynamique.

Quelle est l'adresse IP de votre Raspberry ?

172.20.21.39

Modifier la carte réseau afin de fixer l'adresse IP à 172.20.21.1xx en remplacant xx par la date du jour.

Modifier la passerelle, le masque et le DNS

Test de premier démarrage

Par défaut les services SSH et VNC sont activé par défaut.

Qu'est ce que le SSH ?

Secure Shell, est un moyen d'entretenir une connexion sécurisé entre deux appareils

Quels sont les identifiants et les mots de passe par défaut pour le SSH ?

celui du pc auquel on tente de se connecter

A quoi sert le VNC?

Vitural network Computingest un système de visualisation et de contrôle de l'environnement de bureau d'un ordinateur distant.

Quels sont les identifiants et les mots de passe par défaut du VNC?

L'identifiant et le mot de passe sont le meme que celui de la session

Utilisation

Faire une mise à jour du système

Créer un nouvel utilisateur avec votre nom

Connecter vous en SSH au nouvel utilisateur
Modifier les droits d'accès au « /home » de l'utilisateur pour lui accorder les droits 755
Créer un dossier public_html
Installer les services apache2, php et mysql
Créer un fichier de base info.php contenant le code suivant dans le dossier « html » :

```
<?php phpinfo(); ?>
```

Tester l'affichage de la page info.php
Heberger votre code du module 2 dans le dossier public_html
Vérifier le fonctionnement et l'affichage de votre page web

MODULE 03
SÉANCE WEB 08
TP D'INFORMATIQUE
Durée 2h30

Installation domotique

UNITÉ CERTIFICATIVE
U6 – VALORISATION DE LA DONNÉE ET CYBERSÉCURITÉ

COMPÉTENCE(S)
C08 – CODER

OBJECTIF PÉDAGOGIQUE
Installation d'un système domotique

CONNAISSANCES ISSUES DU RÉFÉRENTIEL

• Langages de développement, de description, de création d'API et les IDE associés	Niveau 4
• Chaînes de développements (ordinateur, embarqué, cross compilation)	Niveau 3

CONNAISSANCES OPÉRATIONNALISÉES
•

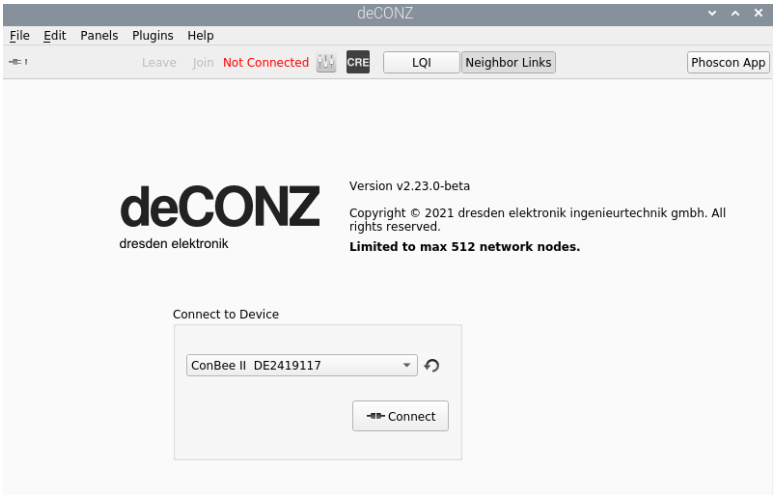
Gaëtan GEORGES
Pascal HAGOT
David IERVESE

g2.iris.lla@gmail.com
pascal.hagot@ac-creteil.fr
d_iervese@yahoo.com

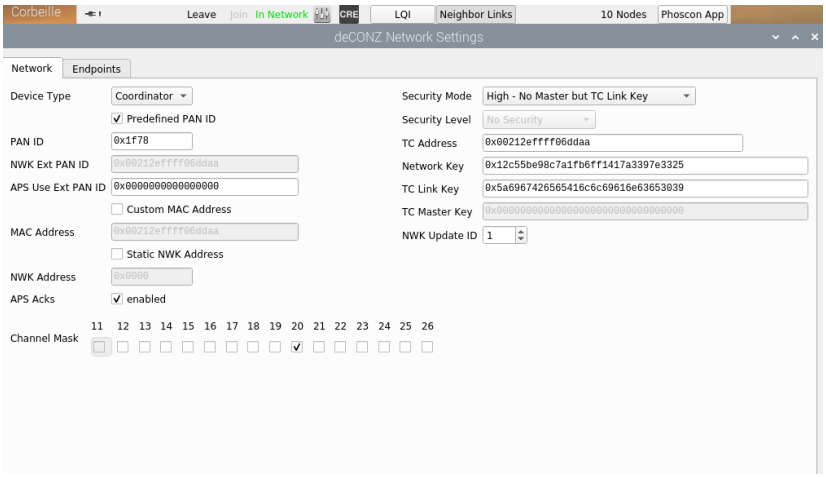
Lycée Louis Armand
173 bd de Strasbourg
Nogent sur Marne

Installation

Démarrer votre raspberry pi installer à la séance précédente avec l'application deconz
Brancher et connecter le dongle Conbee II de phoscon
Ouvrez l'application deconz



Connecter la clé ConBee II
Ouvrer le menu « Network settings » et modifier le canal 11 en 20



Ajouter du matériel

Pré-requis

Voici le matériel mise à votre dispositin :

- Capteur de température AQARA
- Capteur de présence AQARA
- Capteur d'ouverture de porte AQARA
- Smart+ plug Ledvance
- Smart+ Ledvance Gardenpole

Mise en pratique

Ouvrez un navigateur et connecter vous à l'adresse IP de votre raspberry pi

Modifier le nom de la passerelle par : gatewayIR

Modifier le mot de passe par : gatewayIR

Aller dans les paramètres et remettre la langue en français

Aller dans l'onglet « plugs » et cliquer sur le bouton « Add new plug »

Procéder à la manipulation d'appariement de la prise

Aller dans l'onglet « sensors » et cliquer sur le bouton « Add new sensor»

Procéder à la manipulation d'appariement du capteur

Aller dans l'onglet « lights » et cliquer sur le bouton « Add new lights »

Procéder à la manipulation d'appariement de la lumière

Créer un premier groupe « Salon » et y intégrer les différents éléments : prise, capteur d'ouverture de porte et de température

Créer un premier groupe «Jardin» et y intégrer les différents éléments : lumière et capteur d'ouverture de mouvement

Créer un événement qui permet d'allumer les lumières lorsque le capteur de mouvement est activé

Application android

Installer l'application Hue essentiel sur une tablette en téléchargeant l'APK sur internet

Configurer l'application pour la connecter à la passerelle Phoscon

Vérifier que vous êtes capable de contrôler les prises et les lumières