



FICHE DE CONNAISSANCES

Internet des Objet « IdO »

Internet des Objet « IdO »

En anglais : Internet of Things « IoT »

Objectifs :

Ce module a pour objectif la découverte de l'**Internet des Objets IdO** à travers des exemples concrets.

"Objet connecté" implique dans un premier temps de le connecter à l'internet, se sera donc une première étape.

Ensuite, que peut-on faire ?

- lire des données sur des serveurs web pour les communiquer ou les exploiter,
- envoyer des données pour qu'elles soient consultables à distance,
- envoyer des données
- rendre un système autonome d'un point de vue de ses décisions en lui faisant lire des données sur un serveur et les utiliser pour réaliser des actions. adaptées....

Le module proposé se compose de trois parties :

1. **Introduction de quelques sigles et acronymes afin de se familiariser avec les plus répandus**
2. **Connexion d'un "objet" au réseau WiFi**
3. **Création d'objets connectés**
4. **Mini-projet**

I. Introduction de quelques sigles et acronymes

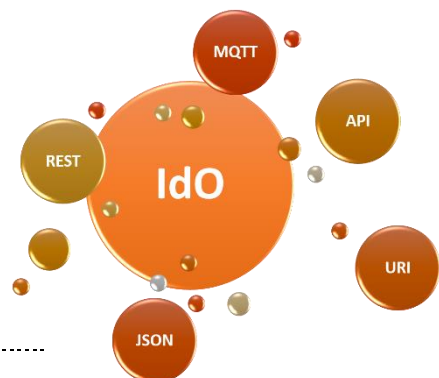
Cette partie apporte les bases nécessaires à la réalisation d'objets connectés.

Il convient donc de commencer par se familiariser aux nombreux sigles ou acronymes en lien avec l'IdO (Internet des Objets).

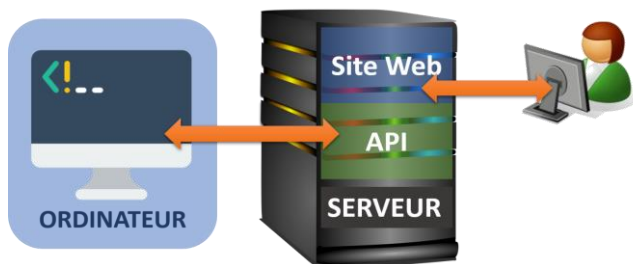
A. API - Application Programming Interface

Définition : Interface de programmation applicative

- une API est un ensemble normalisé de classes, de méthodes ou de fonctions regroupées dans une bibliothèque ou un service Web,
- Elle est **offerte par une bibliothèque logicielle ou un service web**, le plus souvent accompagnée d'une description qui spécifie comment des programmes consommateurs peuvent se servir des fonctionnalités du programme fournisseur.



- Une API est un outil qui rend les données d'un site web "utilisables" par un ordinateur "**client**". Grâce à elle, un client peut voir et modifier des données, exactement comme une personne peut charger des pages et soumettre un formulaire.



- * **Client** : Un programme qui échange des données avec un serveur, au moyen d'une API
- * **Serveur** : Un ordinateur puissant qui exécute un programme nommé API
- * **API** : La partie "cachée" d'un site web qui est destinée à être consommée par les ordinateurs

Il existe deux grands protocoles de communication sur lesquels s'adosent les API :

- **Simple Object Access Protocol (SOAP)** et **Representational State Transfer (REST)**.

REST s'est désormais largement imposé face au SOAP car il est plus flexible.

Il a donné naissance aux API dites **REST ou RESTful (voir plus loin)**.

Exemple du site www.infoclimat.fr



Ce site propose une API

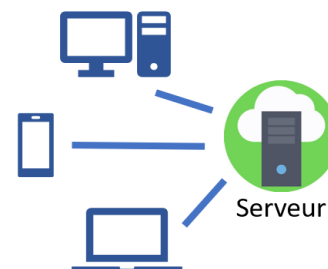
Un onglet API est présent dans la rubrique Prévision. Si on clique sur cet onglet, le site indique le mode d'emploi de son API, c'est à dire comment récupérer les données et dans quel format. Les règles d'utilisation des données sont également indiquées. (voir dans l'activité pratique proposée).

B. Architecture REST - Protocole Client Serveur

Le Web fonctionne sur le mode client/serveur.

- Le client fait une demande au serveur, il s'agit d'une requête.
- Le serveur retourne l'information demandée.

Le client va garder un état en mémoire de ce qui s'est passé, le serveur mémorise rien, il passe aussitôt à la demande suivante.



ne

Définition : REST - Representational State Transfer

REST (Representational State Transfer) ou RESTful est un style d'architecture permettant de construire des applications (Web, Intranet, Web Service).

Il s'agit d'un **ensemble de conventions** et de **bonnes pratiques** à respecter et non d'une technologie à part entière.

L'architecture REST utilise les spécifications originelles du protocole HTTP.

Dans la rubrique "Pour aller plus loin" vous trouverez une description des 5 règles du protocole REST.

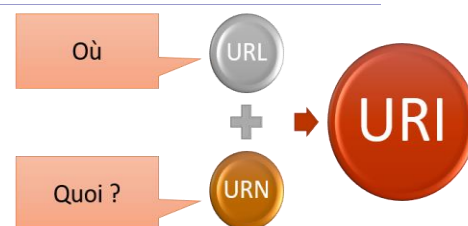
C. URI et URL

Définition : Uniform Resource Identifier

Un URI, de l'anglais **Uniform Resource Identifier** peut se traduire par **Identifiant Uniforme de Ressource**.

Il est composé d'une URL (locator) et d'une URN (name).

- L'URL indique au navigateur Web le chemin à suivre pour accéder aux ressources.
- L'URN est l'identifiant de la ressource. Par exemple le numéro ISBN d'un livre est une URN.



D. JSON



Acronyme de Java Script Object Notation

Se prononce jay-zon avec l'accent anglais.

C'est un format conçu pour transporter des données pour les pages Web.

Il est très utilisé pour l'internet des objets, car il est **compact et simple à utiliser**.

En JSON on trouve **deux types de données** de base :

- les chaînes de caractères qui sont entre "
- et nombres qui ne sont pas entre " et qui contiennent principalement des chiffres.

Par exemple "2018" est une chaîne de caractères et 2018 est un nombre.

On dispose aussi de **deux types structures** :

- **les listes de paires** qui se composent de **deux champs** :
 - le **mot clé** qui est une chaîne de caractères
 - les clés représentent un **attribut de l'objet** que l'on décrit
 - **une valeur** qui peut être n'importe quoi.

Par exemple, voici le json d'un élève de STI2D : { "Identifiant_de_la_classe": 604, "Spécialité_1": "IT", "Spécialité_2": "I2D" }

Les listes sont délimitées par des accolades.

- **les tableaux** : un tableau est **délimité par des crochets droits []** et les éléments sont séparés par des virgules.

Exemple :

[604,"IT","I2D"]

Exemple : Exemple API du site www.infoclimat.fr

```
1  {"request_state":200,"request_key":"fd543c77e33d6c8a5e218e948a19e487","message":  
2  "OK","model_run":"01","source":"internal:GFS:1",  
3  "2018-11-03 04:00:00":{"temperature":{"2m":277.3,"sol":279.2,"500hPa":-  
4  0.1,"850hPa":-0.1},  
5  "pression":{"niveau_de_la_mer":102760},  
6  "pluie":0.1,  
7  "pluie_convective":0,  
8  "humidite":{"2m":94.5},  
9  "vent_moyen":{"10m":8.5},  
10 "vent_rafales":{"10m":26.1},  
11 "vent_direction":{"10m":409},  
12 "iso_zero":2355,  
13 "risque_neige":"non",  
    "cape":0,  
    "nebulosite":{"haute":100,"moyenne":0,"basse":96,"totale":100}},
```

On a ici un exemple de json comportant des listes et **des listes de listes**.

Nous voyons par exemple que pour la clé "pression" une clé "niveau_de_la_mer" a été ajoutée.

La valeur de cette clé est un autre ensemble de clés/valeurs qui nous renseigne sur la pression.

C'est un tableau associatif (associative array), **un objet imbriqué dans un autre objet**.

II. Module ESP 8266 – ESP32

Objectifs

L'objectif de l'activité proposée en activité pratique est la **création d'un objet connecté par la mise en place d'un ESP8266** ou d'un **ESP32**.

Il existe de nombreuses solutions pour réaliser un objet connecté. Le choix de présenter le module **ESP8266** est lié à la facilité de mise en œuvre pour des utilisateurs habitués à l'environnement Arduino.

- Ce circuit intégré comporte un microcontrôleur et un module WiFi.
- Le module WiFi intégré le rend communicant. On peut donc imaginer des **connexions avec un serveur Web, une API...**
- Il dispose d'entrées et de sorties ce qui permet de l'associer à un capteur, un servomoteur...
- Son prix est d'environ 10 euros.
-

L'ESP8266 se programme de la même façon qu'une carte Arduino mais fonctionne en 3.3V

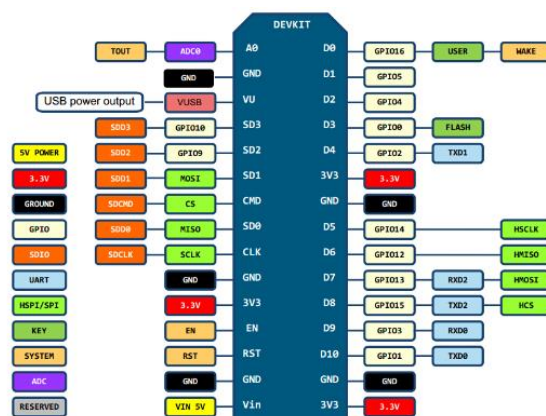
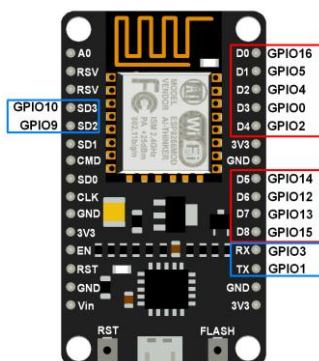
- on retrouve les fonctions habituelles : pinMode, digitalWrite, digitalWrite.
- Par exemple pour **lire GPIO2** la commande est **digitalRead(2)**.



Les numéros de broches Arduino correspondent au numéro de broches du GPIO de l'ESP8266.

- Les broches numériques D0 à D15 peuvent être configurées en : INPUT, OUTPUT ou INPUT_PULLUP
- La broche numérique D16 peut être configurée en : INPUT, OUTPUT ou INPUT_PULLDOWN

Par défaut, au démarrage les broches sont configurées en INPUT.



Caractéristiques des composants de la chaîne d'acquisition :

L'ESP32 :

Fréquence : 240 MHz

Mémoire SRAM : 512 kb

Mémoire Flash : 4 Mb

10 E/S digitales compatibles PWM

Interfaces : I2C, SPI, UART, DAC, ADC

Interface Wifi 802.11 b/g/n 2,4 GHz

Bluetooth : Classique / BLE

Antenne intégrée

Température de service : -40 à 125 °C

Dimensions : 48 x 26 x 11,5 mm

