

Ma Météo



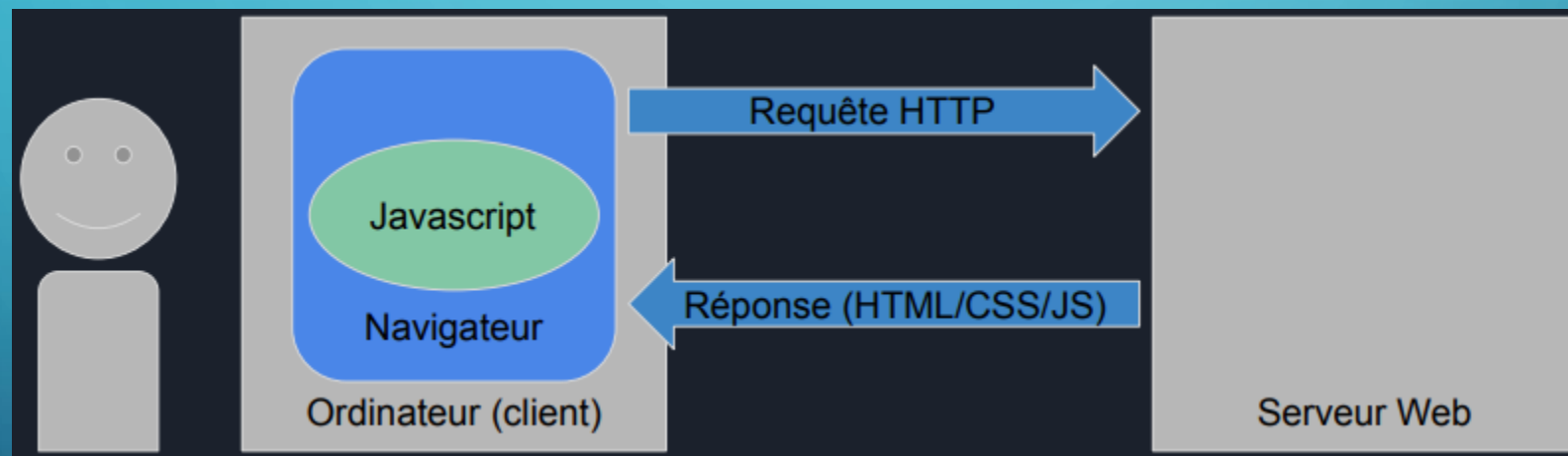
**ADRAR
DIGIT@L ACADEMY**

PÔLE NUMERIQUE DU CENTRE DE FORMATION ADRAR
> SUPPORT, ADMINISTRATION SYSTEMES & RESEAUX
> DEVELOPPEMENT D'APPLICATIONS WEB & MOBILES
> TRANSFORMATION NUMERIQUE DES ENTREPRISES

<http://www.adrar-numerique.com>

Ma Météo

Notion du modèle client - serveur



Ma Météo

Le protocole HTTP

- HyperText Transfer Protocol
- Protocole de communication
- À la base du Web
- HTTPS (chiffrement TLS)
- Actuellement à la version 3

Composition d'une réponse HTTP :

- Version
- Code de status:
 - 20x : OK
 - 30x : redirections
 - 40x : erreurs client (dont la fameuse 404)
 - 50x : erreurs serveur ○ ...
- En-tête
- Corps

Ma Météo

JSON_(rappel)

- Javascript Object Notation,
- Format normé de structuration de données:
 - clé – valeur,
 - types de données: object, number, boolean, string, null, array ,
- Très proche des objets Javascript,
- Utilisé dans tous les langages de programmation,
- Pratique pour les échanges de données client / serveur
 - sérialisation / désérialisation d'objets,
- Peut être utilisé dans le corps des requêtes et réponses HTTP,

Ma Météo

EXEMPLE DE JSON

```
{
  "prenom": "Jonathan",
  "nom": "Gruson",
  "modules": [
    {"titre": "JAVA", "heures": 70},
    {"titre": "PHP", "heures": 40}
  ],
  "referent": true,
  "computer_use": {
    "processeur": "i7",
    "carte-graphique": "2080RTX",
    "ram": "16GB"
  }
}
```

Ma Météo

Autres formats de structuration de données

- XML
- YAML
- Protobuf
- ...

Ma Météo

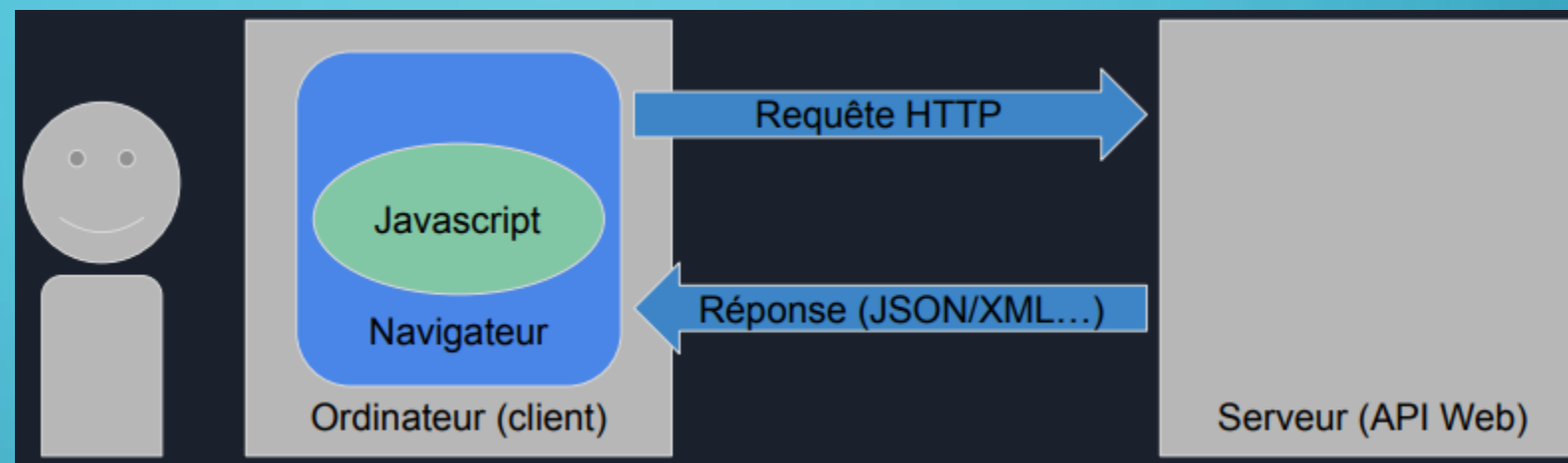
API

Application Programming Interface

- Terme générique, interface permettant à deux systèmes différents d'interagir. Un logiciel offre ses services à un autre via l'API qu'il lui présente
- Un serveur web peut présenter une API, avec laquelle il est possible d'interagir via des requêtes HTTP. Elle peut-être publique, et est dans ce cas documentée
- Exemple: l'API web de Twitter, qui permet par exemple:
 - de rechercher et récupérer des tweets pour pouvoir effectuer un traitement dans sa propre application
 - de publier des tweets
 - de gérer des campagnes de publicité sur Twitter ...
 - documentation: <https://developer.twitter.com>

Ma Météo

API WEB



Javascript: Fetch

- Documentation
- Successeur de XMLHttpRequest
- Permet d'effectuer des requêtes HTTP en Javascript

Ma Météo

TP sur une API météo.

- Site météo
- Exemple= Données météo pour Toulouse: <http://www.prevision-meteo.ch/services/json/toulouse>
- Conseil pour bien visualiser la structure: copiez le JSON dans un fichier, sauvegardez le avec l'extension .json, ouvrez le avec VSCode et utilisez l'auto-formatage

Ma Météo

TP: Faire une requête sur une API web (ici celle de la météo)

1_ créer un fichier `meteo.html` avec ce contenu :

```
<!DOCTYPE html>
<html Lang="en">
  <head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>API météo</title>
  </head>
  <body>
    <h1>API Météo</h1>
    <form>
      <label for="city">Ville:</label><input name="city" type="text">
      <input type="submit" value="Envoyer">
    </form>
    <div id="display"></div>
    <script src="meteo.js"></script>
  </body>
</html>
```

Ma Météo

2_ créer un fichier `meteo.js` avec ce contenu :

```
// je crée une variable dans laquelle je stocke la base de l'url de l'api
const url = "https://www.prevision-meteo.ch/services/json/";

// je fais une fonction de requête à l'api
function requestApi(event) {
    event.preventDefault();
    // je récupère l'input de la ville
    const city = document.querySelector("form input[name='city']");
    console.log(city.value); // à commenter plus tard
    // je fais l'envoi de la requête en concaténant la base url avec la valeur de city
    fetch(`${url}${city.value}`)
        // quand j'obtiens une réponse je mets le body en json
        .then(response => response.json())
        // puis je traite les données reçues
        .then(data => {
            console.log('success:', data);
            // todo plus tard
        })
        // en cas d'erreur je lève une exception et j'affiche l'erreur
        .catch((error) => {
            console.error('Error:', error);
        });
}
```

Ma Météo

3_ Modifiez meteo.js pour faire en sorte d'utiliser le JSON reçu pour créer une interface présentant la météo de la ville (captures ci-dessous)

API Meteo

Ville:

Météo pour Toulouse, France

15.12.2021, 12:00

Conditions actuelles :




Température: 10°C.
Vent: 3 Km/h, E.
Faibles passages nuageux.

Jeudi, 16.12.2021



Ensoleillé.
Température minimale: 4°C, température maximale: 14.

Vendredi, 17.12.2021




Ensoleillé.
Température minimale: 3°C, température maximale: 12.

Samedi, 18.12.2021



Ensoleillé.
Température minimale: 3°C, température maximale: 12.

Dimanche, 19.12.2021



Ensoleillé.
Température minimale: 2°C, température maximale: 13.

Ma Météo

4_ Ajoutez la météo heure par heure pour chaque jour:

Tips:

- l'attribut contenant les données de chaque heure dans le JSON commence par un chiffre (exemple: 7H00). Cela empêche d'utiliser l'opérateur point pour y accéder. À la place, vous aurez besoin de l'opérateur crochet (exemple: `data.fsct_day_1.hourly_data['7H00']`) ([Doc](#))
- Il y a 24 heures dans un jour. Il peut-être judicieux d'essayer d'utiliser une boucle for pour accéder aux attributs 0H00 à 23H00 (avec un compteur i de 0 à 23).

API Meteo

Ville:

Météo pour Toulouse, France

15.12.2021, 13:00

Conditions actuelles :



Température: 12°C.
Vent: 5 Km/h, SE.
Faibles passages nuageux.

Évolution de la journée heure par heure:

0H00 Nuit nuageuse
1H00 Nuit claire
2H00 Nuit claire
3H00 Nuit claire
4H00 Nuit claire

5H00 Nuit claire
6H00 Nuit claire
7H00 Nuit claire
8H00 Nuit claire
9H00 Ensoleillé
10H00 Ensoleillé
11H00 Faibles passages nuageux
12H00 Faibles passages nuageux
13H00 Faibles passages nuageux
14H00 Ensoleillé