

Défi 1 : consolider vos compétences en JS et en API REST

Dans le cadre de ce premier défi, vous allez développer une application Web à partir des données publiées sur l'open data de Tisseo et accessibles à partir d'une API REST.

Attention, pour ce défi :

- chacun des membres du projet doit déposer son code sur IRIS à la fin de la séance
- le chef de projet doit organiser la séance (répartition des tâches, mise en commun, discussion, implication de tous ...)
- le secrétaire doit déposer le compte rendu de la séance contenant :
 - Les membres du groupe, et qui était le chef de projet, qui était le secrétaire
 - vos réponses au défi 1.1
 - les parties du défi restant à terminer
 - les notions qui ne sont pas claires et sur lesquelles vous auriez besoin d'un cours)
 - qui joueront les rôles de secrétaire et de chef de projet à la prochaine séance

Défi 1.1 : Un open data

1. Visualisez la vidéo suivante présentant l'open data <https://www.dailymotion.com/video/x11p37n>

2. Indiquez dans le document à remettre en fin de séance ce qu'est un open data ?

3. Recherchez sur le Web les réponses aux questions suivantes :

i. Qu'est-ce qu'une licence associée à une ressource ?

ii. Une licence est-elle uniquement liée à un logiciel ?

iii. Sous quelle licence sont disponibles les données de Tisseo sur data.toulouse-metropole.fr ?

Vous pouvez trouver la réponse dans les documents :

- **Description de l'open data proposé par Tisseo** : <https://data.toulouse-metropole.fr/explore/dataset/api-temps-reel-tisseo/information/>
- **Documentation décrivant le fonctionnement de l'API proposée par Tisseo** : <https://data.opendatasoft.com/explore/dataset/api-temps-reel-tisseo@toulouse-metropole/files/72b8051dbf567762260cfa66baf8c333/download/>

iv. Quel droit octroie la licence ouverte d'etalab sur les données auxquelles elle est associée (<https://www.etalab.gouv.fr/wp-content/uploads/2017/04/ETALAB-Licence-Ouverte-v2.0.pdf>) ?

Défi 1.2 : Application Web - version 1

Vous devez créer une application Web permettant d'afficher la **liste des lignes de transports** proposées par Tisseo en utilisant l'API REST.

1. Indiquez dans le document à remettre ce qu'est une API REST.

2. Accéder à la documentation de l'API REST de Tisseo

(<https://data.opendatasoft.com/explore/dataset/api-temps-reel-tisseo@toulouse-metropole/files/72b8051dbf567762260cfa66baf8c333/download/>) et retrouver dans la documentation :

- l'objectif et le fonctionnement de l'API de Tisseo (page 4 et page 5)
- la description du service permettant de récupérer l'ensemble des lignes disponibles.

Attention, vous avez pu noter que l'utilisation de l'API de Tisseo nécessite d'avoir au préalable un clé délivrée par Tisseo. Nous en avons fait la demande et la clé est

key=a3732a1074e2403ce364ad6e71eb998cb

Cette clé doit être intégrée dans l'url d'accès aux différents services proposée par Tisseo (page 5 de la documentation).

3. Créez une page HTML contenant un bouton.

4. Mettez en place le code JS permettant d'envoyer, lorsque l'utilisateur clique sur le bouton, une requête Ajax vers l'API Tisseo afin de récupérer l'ensemble des lignes de transports proposées par Tisseo. Une fois les données reçues par le client Web, elles devront être affichées dans la page sous la forme d'une liste à puce.

Petite aide :

- l'URL du service à interroger est

<https://api.tisseo.fr/v2/lines.json?key=a3732a1074e2403ce364ad6e71eb998cb>

- les données retournées par ce service sont sous la forme d'un objet JSON et peuvent être visualisées dans un navigateur¹.

Défi 1.3 : Application Web - version 2

Dans cette nouvelle version, vous devez au clic sur le nom d'une ligne afficher dans la page l'ensemble de ses arrêts.

1. Lisez la documentation de l'API et accédez dans la documentation à la description du service permettant de récupérer l'ensemble des arrêts pour une ligne.

2. Mettez en place le code permettant d'envoyer, lorsque l'utilisateur clique sur le nom d'une ligne, une requête Ajax vers l'API Tisseo afin de récupérer l'ensemble des lignes de transports proposées par Tisseo. Une fois les données reçues par le client Web, elles devront être affichées dans la page sous la forme que vous souhaitez.

Petite aide :

- l'URL du service à interroger est [https://api.tisseo.fr/v2/stop_points.json?](https://api.tisseo.fr/v2/stop_points.json?key=a3732a1074e2403ce364ad6e71eb998cb&lineId=IDLIGNE)

[key=a3732a1074e2403ce364ad6e71eb998cb&lineId=IDLIGNE](https://api.tisseo.fr/v2/stop_points.json?key=a3732a1074e2403ce364ad6e71eb998cb&lineId=IDLIGNE) où IDLIGNE doit correspondre à l'identifiant de la ligne pour laquelle les arrêts doivent être affichés.

Cette identifiant est fourni dans la réponse envoyée par le service utilisé au défi 1.2

La capture ci-dessous reprend les données renvoyées par le service utilisé au défi 1.2 :

```
expirationDate: "2023-01-20 15:12"
lines:
  line:
    0:
      bgXmlColor: "#db001b"
      color: "(219,0,27)"
      fgXmlColor: "#ffffff"
      id: "line:61"
      name: "Basso Cambo / Balma - Gramont"
      network: "Tisséo"
      reservationMandatory: "0"
      shortName: "A"
      transportMode:
        article: "le"
        id: "commercial_mode:1"
        name: "métro"
    1:
      bgXmlColor: "#ffdd00"
      color: "(255,221,0)"
      fgXmlColor: "#000000"
      id: "line:69"
      name: "Borderouge / Ramonville"
      network: "Tisséo"
      reservationMandatory: "0"
      shortName: "B"
      transportMode:
        article: "le"
        id: "commercial_mode:1"
        name: "métro"
    2:
      bgXmlColor: "#004687"
      color: "(0,70,135)"
      fgXmlColor: "#ffffff"
      id: "line:68"
```

Par exemple, pour accéder aux arrêts de la ligne « A », l'URL qui devra être utilisé par le script répondant au défi 1.3 sera :

¹ Il est possible d'intégrer une extension au navigateur tel que JSON VIEWER <https://chromewebstore.google.com/detail/json-viewer/>

[https://api.tisseo.fr/v2/stop_points.json?
key=a3732a1074e2403ce364ad6e71eb998cb&lineId=line:61](https://api.tisseo.fr/v2/stop_points.json?key=a3732a1074e2403ce364ad6e71eb998cb&lineId=line:61)

Vous pourrez donc noter que l'url à utiliser pour récupérer les arrêts d'une ligne dépendra de la ligne cliquée pour obtenir les arrêts. Vous devez donc écrire un script qui sera en mesure d'accéder à l'id de la ligne cliquée pour formuler la requête adaptée

Défi 1.4 : Application Web - version 3

Vous avez pu noter que l'application affiche 2 fois le même arrêt pour une ligne. Modifiez l'application pour que chaque arrêt ne soit affiché qu'une seule fois.