

TP - AJAX

Introduction

Nous vous demanderons dans ce TP de créer une application web permettant un affichage interactif des lignes de métro/bus de Tisseo.

Pour cela vous allez récupérer des données disponibles sur les lignes Tisseo via l'API de Tisseo puis utiliser ces données via l'API Google map dans l'objectif de les visualiser sur une carte interactive.

1. Lister les lignes

- Créez une page web contenant un bouton.
- Mettez en place le code permettant d'envoyer, lorsque l'utilisateur clique sur le bouton, une requête Ajax vers l'API Tisseo. Une fois les données reçues par le client, elles devront être affichées dans la page sous la forme d'une liste donnant les lignes métro/bus proposées par Tisseo. L'url du service de Tisseo à interroger pour obtenir cette liste¹ est

<https://api.tisseo.fr/v1/lines.json?key=a3732a1074e2403ce364ad6e71eb998cb>

L'illustration 1 donne un échantillon des données retournées par l'API Tisseo au format JSON. L'illustration 2 montre la page html ayant réagi au clic sur le bouton.

```
{
  expirationDate: "2015-04-04 03:45",
  lines: [
    line: {
      color: "(255,0,0)",
      bgXmlColor: "#ff0000",
      fgXmlColor: "#FFFFFF",
      id: "11821949021891694",
      name: "Basso Cambo / Balma - Gramont",
      shortName: "A",
      network: "Tisséo",
      transportMode: {
        id: "13792273858822586",
        article: "le",
        name: "métro"
      }
    },
    line: {
      color: "(255,255,0)",
      bgXmlColor: "#ffff00",
      fgXmlColor: "#000000",
      id: "11821949021892004",
      name: "Borderouge / Ramonville",
      shortName: "B",
      network: "Tisséo",
      transportMode: {
        id: "13792273858822586",
        article: "le",
        name: "métro"
      }
    }
  ]
}
```

Illustration 1: Format des données

TP AJAX

Init lines

- A - Basso Cambo / Balma - Gramont
- B - Borderouge / Ramonville
- T1 - Aéroconstellation / Palais de Justice
- T2 - Aéroport / Palais de Justice
- AERO - Navette Aéroport
- CIMTR - Navette Cimetières
- NOCT - Noctambus
- VILLE - Navette Centre Ville
- L16 - Sept Deniers / Gymnase de l'Hers
- 1 - Grand Rond / Compans - Caffarelli
- 2 - Cours Dillon / Université Paul Sabatier
- 2s - Cours Dillon / Université Paul Sabatier
- 3 - St Cyprien - République / Oncopole
- 8 - Basso Cambo / Cité Scolaire Rive-Gauche
- 10 - Cours Dillon / Malepère
- 10s - Cours Dillon / Malepère
- 11 - Empalot / Basso Cambo
- 12 - Basso Cambo / Cours Dillon

Illustration 2: Rendu attendu

¹ Veuillez à utiliser exactement cette url.

c) Complétez le code pour que chaque ligne soit affichée dans la couleur (bgXmlColor) renvoyée pour la ligne dans API. Réalisez cette question sans puis avec JQuery.

TP AJAX

- Init lines
- A - Basso Cambo / Balma - Gramont
 - B - Borderouge / Ramonville
 - T1 - Aéroconstellation / Palais de Justice
 - T2 - Aéroport / Palais de Justice
 - AERO - Navette Aéroport
 - CIMTR - Navette Cimetières
 - NOCT - Noctambus
 - VILLE - Navette Centre Ville
 - L16 - Sept Deniers / Gymnase de l'Hers
 - 1 - Grand Rond / Compans - Caffarelli
 - 2 - Cours Dillon / Université Paul Sabatier
 - 2s - Cours Dillon / Université Paul Sabatier
 - 3 - St Cyprien - République / Oncopole
 - 8 - Basso Cambo / Cité Scolaire Rive-Gauche
 - 10 - Cours Dillon / Malepère
 - 10s - Cours Dillon / Malepère
 - 11 - Empalot / Basso Cambo
 - 12 - Basso Cambo / Cours Dillon
 - 12s - Cours Dillon / Basso Cambo

Illustration 3: Aperçu : coloration ligne

2. Affichage des arrêts d'une ligne

Nous souhaitons maintenant afficher sur la page les arrêts d'une ligne spécifique.

Pour cela, mettez en place une requête Ajax récupérant et affichant la liste de tous les arrêts d'une ligne (nom, longitude, latitude) lors du clic sur le nom de cette ligne (récupéré et affiché précédemment) en interrogeant le service suivant :

https://api.tisseo.fr/v1/stop_points.json?key=a3732a1074e2403ce364ad6e71eb998cb&displayCoordXY=1&lineId=idLigne

Pensez à changer dans l'URL ci-dessus la valeur « idLigne » par l'id de la ligne dont vous souhaitez récupérer les arrêts. La valeur 1 associée à l'attribut « displayCoordXY » dans l'url permet de demander dans la réponse les coordonnées longitude/latitude de chaque arrêt de la ligne.

```
{
  expirationDate: "2015-04-04 03:45",
  physicalStops: {
    physicalStop: [
      {
        id: "3377699720880977",
        name: "Basso Cambo",
        handicappedCompliance: "1",
        x: "1.3922690047953872",
        y: "43.570110000025252",
        operatorCodes: [
          {
            operatorCode: {
              value: "40001",
              network: "Tisséo"
            }
          }
        ]
      },
      {
        id: "3377699720881446",
        name: "Bellefontaine",
        handicappedCompliance: "1",
        x: "1.3992620047927491",
        y: "43.565895000024319",
        operatorCodes: [
          {
            operatorCode: {
              value: "40010",
              network: "Tisséo"
            }
          }
        ]
      }
    ]
  },
  stopArea: {
    cityName: "TOULOUSE",
    id: "1970324837184808",
    name: "Basso Cambo",
    x: "1.392284",
    y: "43.570008"
  },
  {
    id: "3377699720881446",
    name: "Bellefontaine",
    handicappedCompliance: "1",
    x: "1.3992620047927491",
    y: "43.565895000024319",
    operatorCodes: [
      {
        operatorCode: {
          value: "40010",
          network: "Tisséo"
        }
      }
    ]
  },
  stopArea: {
    cityName: "TOULOUSE",
    id: "1970324837185011",
    name: "Bellefontaine",
    x: "1.399326",
    y: "43.565778"
  },
}
```

Illustration 4: Format des données

L'illustration 3 montre un échantillon des données retournées par le service. L'illustration 4 donne un exemple de rendu attendu lorsque l'utilisateur clique sur la ligne A.

TP AJAX

- Init lines
- A - Basso Cambo / Balma - Gramont
 - B - Borderouge / Ramonville
 - T1 - Aéroconstellation / Palais de Justice
 - T2 - Aéroport / Palais de Justice
 - AERO - Navette Aéroport
 - CIMTR - Navette Cimetières
 - NOCT - Noctambus
 - VILLE - Navette Centre Ville
 - L16 - Sept Deniers / Gymnase de l'Hers
 - 1 - Grand Rond / Compans - Caffarelli
 - 2 - Cours Dillon / Université Paul Sabatier
 - 2s - Cours Dillon / Université Paul Sabatier
 - 3 - St Cyprien - République / Oncopole
 - 8 - Basso Cambo / Cité Scolaire Rive-Gauche
 - 10 - Cours Dillon / Malepère
 - 10s - Cours Dillon / Malepère
 - 11 - Empalot / Basso Cambo
 - 12 - Basso Cambo / Cours Dillon
 - 12s - Cours Dillon / Basso Cambo
 - 14 - Basso Cambo / Marengo - SNCF
 - 15 - Jeanne d'Arc / États-Unis Fondéyre
 - 17 - Andromède - Lycée / Mondonville Foyer Rural
 - 19 - Borderouge / Place de l'Indépendance
 - 20 - Balma - Gramont / Beaupuy Mairie
 - 21 - Basso Cambo / Colomiers Airbus

Arrets de la ligne VILLE - Navette C

- Carmes (1.445645004779992 - 43.598465000027154)
- Cours Dillon (1.4373890047827482 - 43.598909000027561)
- Jean Jaurès (1.4484720047796444 - 43.605975000028138)
- Pont Neuf (1.4404250047817777 - 43.599302000027514)
- Salin-Parlement (1.4443220047801173 - 43.594490000026632)
- Bellegarde (1.4455010047807628 - 43.607731000028522)
- Rémusat (1.4437960047811291 - 43.605236000028228)
- Brunière (1.4432460047805349 - 43.595290000026779)
- Féral (1.4464270047795373 - 43.595954000026772)
- Trois Banquets (1.4480860047792912 - 43.599815000027256)
- St Sermin (1.4418590047820476 - 43.608806000028828)
- Embarthe (1.4383210047831543 - 43.608040000028893)
- Quai St Pierre (1.4342540047841796 - 43.603979000028438)
- Gambetta (1.44216800478152 - 43.603353000028015)
- Quai de la Daurade (1.4401780047819175 - 43.600055000027638)

Illustration 5: Résultat attendu

3. Visualisation des arrêts sur une carte

Vous devez maintenant reprendre le code produit pour la question précédente de façon à ne plus afficher les informations sur les arrêts d'une ligne mais à visualiser ces arrêts sur une carte interactive.

Les coordonnées longitude/latitude de chaque arrêt de la ligne retournées par le script de tisseo correspondent aux points à afficher sur la carte.

Pour faciliter la génération de la carte dynamique (avec l'API Google) à la réception des données renvoyées par l'API Tisseo, une fonction vous est fournie. Pour l'utiliser il vous faut intégrer deux scripts en ajoutant le code suivant dans le head de votre page html :

```
<script src="http://amarger.murloc.fr/web2/exam/jquery-1.11.2.min.js"></script>
<script src="http://amarger.murloc.fr/web2/exam/app.js"></script>
```

La fonction (définie dans app.js) que vous devez utiliser est :

```
init_map(idDiv, markers);
```

Cette fonction permet de générer une carte dynamiquement dans un div. A ce div intégré dans la page doit être associé un identifiant à donner en premier paramètre lors de l'appel à la fonction (idDiv). Le deuxième paramètre (markers) de la fonction contient les données des différents markers (points) à afficher sur la carte. La variable « markers » doit être un objet javascript correspondant à l'illustration 6. Dans cet objet, l'attribut « color » sert à définir la couleur qui sera utilisée pour les markers. L'attribut « stops » est un tableau contenant chaque arrêt de la ligne avec la latitude, la longitude et le nom de l'arrêt.

```
{
  color: "#ff0000",
  stops: [
    {
      lat: "43.570110000025252",
      long: "1.3922690047953872",
      name: "Basso Cambo"
    },
    {
      lat: "43.565846000024308",
      long: "1.3992390047927525",
      name: "Bellefontaine"
    }
  ]
}
```

Illustration 6: Format des données (markers)

L'illustration 7 donne un exemple de rendu attendu.



Illustration 7: Rendu attendu