Développement d'Applications Web

Cours 5 : Exercises

Vincent Simonet, 2015-2016

Exercice 1 : RATP

Dans cet exercice, nous supposons que la RATP met à disposition du public une API en ligne permettant de connaître l'horaire de passage des prochains métro. En appel à cette API se fait en faisant une requête HTTP GET vers une adresse de la forme :

```
http://api.ratp.fr/nexttrain?line=<line>
                            &station=<station>
                            &direction=<direction>
                            &callback=<callback>
```

La réponse est un texte brut de la forme :

```
<callback>({line: <line>, station: '<station>',
            direction: '<direction>',
            date: '<date>', time: '<heure>'});
```

Les fragments entre <...> sont à remplacer par des valeurs réelles. La date est au format yyyy-mm-dd (toujours 10 caractères), et l'heure au format hh:mm (toujours 5 caractères, sur 24 heures). Par exemple, l'appel

```
http://api.ratp.fr/nexttrain?line=7
                            &station=Jussieu
                            &direction=Villejuif
                            &callback=mycallback
```

va retourner

```
mycallback({line: 7, station: 'Jussieu', direction: 'Villejuif',
            date: '2014-10-20', time: '15:45'})
```

si le prochain métro sur la ligne 7 à la station Jussieu en direction de Villejuif va passer le lundi 20 octobre 2014 à 15:45. (L'API est très tolérante sur la manière d'écrire le nom des stations, ne vous préoccupez donc pas de l'orthographe exacte dans la suite.)

Dans les prochaines questions, vous êtes en charge de développer un script JavaScript à ajouter à la page d'accueil du site `www.upmc.fr` de manière à afficher les horaires de prochain métros au départ de la station Jussieu. Nous supposerons que l'administrateur du site a ajouté le code HTML suivant à la page d'accueil :

```
<html>
  <head>
    ...
    <script src="metro.js" type="application/javascript"></script>
  </head>
  <body onload="showMetro();">
    ...
    <div id="metro"></div>
    ...
```

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

Votre script devant ajouter les horaires dans le div "metro".

Question 1.1

Un(e) ami(e) qui vient de suivre un cours sur le développement d'applications Web vous suggère d'utiliser un appel AJAX pour réaliser les appels à l'API de la RATP dans ce script. Est-ce une bonne idée ? Pourquoi ?

Pour la suite de l'exercice, l'administrateur du site `www.upmc.fr` vous demande explicitement d'utiliser le protocole JSONP.

Question 1.2

Écrivez un script qui affiche sur la page un message du type "Le prochain train pour Boulogne sur la ligne 10 passe à hh:mm" (où hh:mm est remplacé par l'horaire exact).

Question 1.3

Étendez votre script de manière à afficher deux messages successifs :

"Le prochain train pour Boulogne sur la ligne 10 passe à hh:mm"
"Le prochain train pour Austerlitz sur la ligne 10 passe à hh:mm"

(Il n'est pas demandé que les messages soient affichés dans un ordre particulier.)

Question 1.4

Comme vous le savez, la ligne 7 dispose de deux branches vers le sud, l'une en direction de Villejuif et l'autre en direction d'Ivry. L'administrateur du site souhaite afficher un message indiquant le terminus du prochain train sur la ligne 7, comme :

"Le prochain train vers le sud sur la ligne 7 a pour terminus Ivry"

Écrivez un script qui affiche un tel message. (Notez que vous pouvez comparer les dates et heures directement sous forme de chaînes de caractères puisque l'ordre lexicographique est le même que l'ordre chronologique avec le format utilisé.)

Exercice 2: Client Web et RPC

Dans cet exercise, nous nous intéressons à la partie client d'une petite application web permettant d'obtenir le statut (en vol ou arrivée) et l'heure d'arrivée d'un vol aérien en fonction du numéro de vol. Vous trouverez en annexe le code source de la page HTML et du script JavaScript. Toutes les questions suivantes de l'exercise se réfèrent à ce code.

Question 2.1 Sélecteurs CSS

Indiquez les numéros de ligne des éléments HTML auxquels les sélecteurs CSS suivants se réfèrent. Pour chaque sélecteur, il peut y a voir aucun, 1 ou plusieurs éléments HTML (écrivez explicitement "aucun" dans la colonne de réponse pour distinguer d'une absence de réponse). Pour les éléments qui prennent plusieurs lignes, indiquez le numéro de ligne du tag d'ouverture.

	Sélecteur CSS	Numéro(s) de ligne des sélecteurs
a.	`#flight-number`	
b.	`.field`	
c.	`#status .field`	
d.	`.field > #flight-number`	
e.	`#status #flight-number`	
f.	`#status > #flight-number`	
g.	`span.form input`	
h.	`input[type="button"]`	
i.	`body #status > .field span`	
j.	`.form .field, #status .field`	

Question 2.2: Fonction submitForm

Expliquez le principe de la fonction submitForm. Quel est le nom de la technique / du protocole utilisé ? Comment cela fonctionne-t-il ? Pourquoi le développeur n'a-t-il pas choisi de faire un appel AJAX ?

Question 2.3: Réponse

Donnez un exemple de réponse (en-têtes et corps) retournée par la requête HTTP:

```
GET http://airservice.com/?q=AF007 HTTP/1.1
```

Question 2.4: AJAX (2 points)

Supposant que le problème mentionné à la question 2.2 a été résolu, réécrivez le code JavaScript de cette application en utilisant un appel AJAX, et en supposant que le serveur retourne un document, selon votre préférence, XML ou JSON. Vous veillerez à traiter correctement les cas d'erreur.

Code source exercise 2

Page: http://www.flightstatus.com/index.html

```
1   <html>
2     <head>
3       <title>Flight Status</title>
4       <script type="text/javascript" src="script.js"></script>
5     </head>
6     <body onload="init();">
7       <h1>Flight Status</h1>
8       <div class="form">
9         <span class="field">Enter your flight number
10          <input type="text" id="query"/>
11       </span>
12        <input type="button" id="submit-button" value="Submit"/>
13      </div>
14      <div id="status">
15        <div class="field">Flight number:
16          <span id="flight-number"></span>
17        </div>
18        <div class="field">Current status:
19          <span id="current-status"></span>
20        </div>
21        <div class="field">Expected arrival time:
22          <span id="arrival-time"></span>
23        </div>
24      </div>
25    </body>
26  </html>
```

Script: http://www.flightstatus.com/script.js

```
1   function init() {
2     var submit_button = document.getElementById('submit-button');
3     submit_button.addEventListener('click', submitForm, false);
4   }
5
6   function submitForm(event) {
7     var query = document.getElementById('query');
8     var script = document.createElement('script');
9     script.setAttribute('src', 'http://airservice.com?q=' + query.value);
10    document.head.appendChild(script);
11  }
12
13  function callback(response) {
14    document.getElementById('flight-number').textContent =
15    response.flight_number;
16    document.getElementById('current-status').textContent =
17     response.current_status;
18    document.getElementById('arrival-time').textContent =
19     response.arrival_time;
20  }
```