

Entretien technique

On dispose d'une structure de donnée au format JSON :

```
events : {
  deviceId1: "event description",
  deviceId2: "other event",
  ...
}

devices : {
  deviceId1 : {
    location: "Paris",
    name: "deviceName"
  },
  otherDeviceId : { ... },
  ...
}
```

À la racine se trouvent deux objets `events` et `devices`, avec chacun des propriétés `deviceId` qui se retrouvent dans les 2 structures. Dans `events` on stocke un ensemble d'évènements (ici de simples chaînes de caractères), chacun associé à un *device*. On peut retrouver le détail de chaque *device* dans la partie `devices` en utilisant le même `deviceId`.

Pour accéder à ces données, on dispose d'une fonction `async fetch(path)` qui permet de récupérer un sous-objet en spécifiant son chemin dans la hiérarchie. Ce chemin est une chaîne de caractères composée des clefs des objets, séparés par `/` :

```
fetch("devices") ->
  { "deviceId1" :
```

```
        { location: "Paris", name: "deviceName" },
    "otherDeviceId" :
        { location: "Brest", name: "otherName" },
    ... }
```

```
fetch("devices/otherDeviceId") ->
    { location: "Brest", name: "someName" }
```

```
fetch("devices/otherDeviceId/location") -> "Brest"
```

En pratique, les requêtes `fetch` se feraient vers un serveur. Pour cet exercice, on fournit une méthode `fetch` qui simule ce comportement avec un objet JSON local.

Le but de l'exercice est d'écrire une `function report()` qui renvoie un objet JS contenant tout le contenu présent dans la partie `events` de la base, **groupé par lieu**. Le lieu d'un événement est disponible dans la propriété `location` du *device* correspondant dans la partie `devices` de la base.

Exemple de résultat :

```
{
  "Brest" : ["event", "other event", "more..."],
  "Paris" : ["event description", "un autre texte"],
  ...
}
```

Le fichier `exo.js` fournit la méthode `fetch` qui permet de récupérer le contenu de la donnée JSON locale (que vous ne devez pas utiliser directement). À vous d'écrire la méthode `report()`. Vous pouvez exécuter votre code en tapant `node exo.js` dans un terminal (ou en utilisant la console d'un navigateur internet)