



# TP2 Architecture en pipeline

---

## Architecture pipeline

---

Dans la première partie de ce TP nous allons mettre en place une architecture en pipeline simple. L'objectif est de d'afficher sur une carte du monde tous les tremblements de terre recensés lors des deux derniers jours.

### Afficher les derniers tremblements de terre !

Pour vous faciliter la tâche nous allons utiliser Node-Red.

1. Lancez Node-Red à l'aide de Docker :  
<https://hub.docker.com/r/nodered/node-red>  
(si vous n'avez pas encore installé docker vous pouvez installer Node-Red directement)
2. Vous pouvez maintenant utiliser Node-Red en vous connectant à <http://localhost:1880>
3. Node-Red est une plateforme de programmation pour l'IoT. Avec Node-Red les applications prennent la forme d'un ensemble de nœuds (nodes) assemblés par de liens (wire). Les communications entre les liens se font sous la forme de messages, ce qui rend la plateforme Node-Red adaptée pour mettre en place des pipelines pour traiter des flux de données. Les nœuds sont implémentés en JavaScript (Node-Red tourne au-dessus de Node.js). De nombreux nœuds existent et peuvent être installés automatiquement. Essayez d'utiliser Node-red en suivant le tutoriel disponible ici : <http://developer.opto22.com/nodered/general/getting-started/node-red-hello-world/>.

Nous allons maintenant créer une application permettant d'afficher sur une carte du monde les tremblements de terre recensé lors des 2 derniers jours. L'affichage est mis à jour toutes les 5 minutes. Nous utiliserons une petite architecture en pipeline pour construire cette application.

Commencez par consulter la page suivante : [http://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/2.5\\_day.csv](http://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/2.5_day.csv)

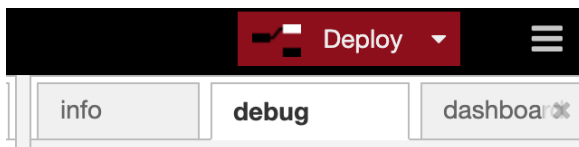
Il s'agit de notre source de données. Les données sont au format CSV. Nous afficherons les tremblements de terre grâce au composant `worldmap` de `node-red`. Ce dernier prend en entrée un JSON au format suivant :

```
{ "name": "Bob", "lat": 51.05, "lon": -1.35, "radius": 30 }
```

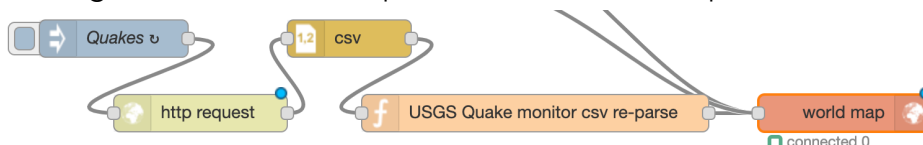
4. Proposez une architecture pour votre application

Nous allons maintenant implémenter notre application. Dans node-red :

- Ajoutez un nœud « `http request` » à votre application. Configurez le nœud pour qu'il fasse une requête `http GET` à l'url suivante : [http://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/2.5\\_day.csv](http://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/2.5_day.csv)
- Ajoutez un nœud de type « `inject` » que vous appellerez « `Quakes` ». Ces nœuds sont utilisés pour déclencher une action en envoyant un message. En utilisant la propriété « `Repeat` » faites en sorte qu'il déclenche une action toute les 5 minutes. Reliez ce nœud au nœud « `http request` ».
- Ajoutez un nœud de type « `debug` » (un nœud qui permet d'afficher dans l'onglet « `debug` » le contenu des messages qu'il reçoit). Reliez la sortie du nœud « `http request` » au nœud `debug`.



- Ça y est, vous avez construit votre premier pipeline ! Toutes les cinq minutes vous aller chercher les informations sur les derniers tremblements de terre et vous affichez ces informations ! Pour tester votre pipeline cliquez sur le bouton à côté du nœud « `Quakes` ». Dans votre pipeline le nœud « `http request` » joue le rôle de data source. Nous allons maintenant compléter notre pipeline pour : (i) formater et extraire uniquement les informations pertinentes et (ii) afficher les tremblements de terre sur la carte du monde.
- Installez le nœud « `worldmap` » (dans le menu en haut à droite de l'écran cliquez sur `Manage Palette` puis `Install`). Pour afficher une information sur la carte il suffit de lui transmettre un message contenant des informations du type : `msg.payload = { "name":"Jason", "lat":51.05, "lon":-1.35, "radius": 30 }` (notez que le `radius` est une valeur fixe pour chaque message)  
Ajoutez un nœud « `worldmap` » à votre application.
- Utilisez les nœuds « `csv` » et « `function` » pour récupérer les informations du nœud « `http request` » et transmettre uniquement les informations pertinentes au nœud « `worldmap` ». La figure ci-dessous présente un exemple d'assemblage final.



11. Vous pouvez visualiser le résultat de votre application à l'url : <http://127.0.0.1:1880/worldmap>.

Attention, pour tester, il faut ouvrir la carte avant d'appuyer sur le bouton à côté du nœud « Inject ». On pourra obtenir le résultat suivant.

