



**Objectifs** A l'issue de ce projet, vous serez capables de

- Décrire une architecture REST pour un SOA
- Choisir les modules NodeJS adaptés pour construire une application REST
- Construire une application REST

## Partie TP : le classique gestionnaire de tâches

Nous allons manipuler une collection de tâches ("Todos"). Chaque tâche est une entité avec les attributs suivants:

Attribut	Description
id	ID unique de la tâche. Cette donnée sera générée automatiquement, pour références
title	Titre de la tâche (exemple : "Finir le projet JEE")
dateBegin	Date de début de la tâche (exemple : 20/03/2018)
dateEnd	Date d'échéance de la tâche (exemple : 11/04/2018)
statut	Degré d'avancement ("non précisé", "une tâche est requise", "En cours", "Achevée", "Annulé")
tags	Liste de catégories ("tags", simples chaines de caractères) rattachés à la tâche (exemple : "travail"+"web"+"youpi"). Les catégories sont libres.

Dans un premier temps, il faut pouvoir

- récupérer un objet tâche en fonction de son identifiant
- créer un objet tâche dans le système en passant un JSON au système
- modifier un objet tâche dans le système en passant un JSON au système
- détruire un objet tâche dans le système

Pour cela, vous aller proposer un service de gestion de tâche selon l'architecture REST (en utilisant NodeJS et le routage de Express).

Vous aller suivre les étapes suivantes:

1. Sous format texte, proposez une description de l'implémentation du service (les trois sommets du triangle) et soumettez le à votre encadrant. En particulier, vous identifierez la ressource servie, les opérations HTTP liées aux actions et les formats de livraisons. Vous y décrirez comment vous implémenterez le contrôleur et la vue. Pour le modèle, nous allons faire un développement en deux temps. Prévoyez donc un mécanisme pour pouvoir basculer facilement d'une implémentation à une autre.
2. Après validation, implémentez votre service. Dans un premier temps, vous utiliserez une simple liste dynamique comme modèle, sans stockage permanent. (implémentation d'un "Mock").
3. Testez votre système en utilisant des requêtes HTTP forgées par un plugin de votre navigateur (RESTClient pour firefox par exemple). Ne pas utiliser d'AJAX.

## Partie Projet (en monome : le gestionnaire de tâche Premium)

Proposez une interface graphique pour gérer vos tâches. Vous forgerez les requêtes avec l'objet javascript "httpRequest".

Par la suite, nous voulons améliorer le service en récupérant les tâches non achevées et non annulées qui ne sont pas arrivées à échéance.

- Faut il créer un nouveau service (c'est à dire une nouvelle ressource)?
- Modifiez votre code (Contrôleur et Vue) pour cette nouvelle fonctionnalité.

Par la suite, nous voulons améliorer le service en récupérant les tâches en fonction des catégories.

- Faut il créer un nouveau service (c'est à dire une nouvelle ressource)?
- Modifiez votre code (Contrôleur et Vue) pour cette nouvelle fonctionnalité.

Enfin, vous utiliserez une base de données de votre choix ("MySQL" ou "REDIS") pour implémenter la partie Modèle. Bonus au choix (soit l'un, soit l'autre):

- Proposez une modification pour assigner des personnes à une tâche. Faut il faire intervenir un nouveau service?
- Proposez un gestionnaire de tâche comme un plugin firefox.

## Comment faire pour...

### ...servir du JSON?

Express permet d'envoyer au format JSON sans passer par:

```
res.send(JSON.stringify( { type: 'history', data: history} ));
```

Il suffit de faire:

```
res.json( jsonObject );
```