

Création d'une ressource RESTful

- `$resource` :
 - Méthode nous permettant de créer une « ressource RESTful »
- Syntaxe de la création (ressource standard) :
 - `var maRessource = $resource('/chemin/vers/la/ressource');`
- Syntaxe de l'appel :
 - `maRessource.get();` // GET qui renvoie un objet
 - `maRessource.query();` // GET qui renvoie un tableau
 - `maRessource.save();` // POST
 - `maRessource.remove();` ou `maRessource.delete();` // DELETE

Création d'une ressource personnalisé

- Syntaxe de la création :

- ```
var maRessource = $resource("/ma/ressource",
 {...}, // paramètres par défaut
 {
 "methode1" : { method: "GET", isArray: true },
 "methode2" : { method: "GET", isArray: false, url: "/ressource/:id" },
 "methode3" : { method: "PUT", url: "/ressource2" }
 }
);
```

- Syntaxe de l'appel :

- `maRessource.methode1();` // GET /ma/ressource (renvoie un tableau)
  - `maRessource.methode2({ id: 5 });` // GET /ressource/5 (renvoie un objet)
  - `maRessource.methode3({ ... });` // PUT /ressource2 avec donnée passée en paramètre



# Retours des appels

- Plus de promesses, que des résultats !
  - Objet (par défaut ou si isArray est à false)
  - Tableau (si isArray est à true)
  - Vides tant que la réponse n'est pas arrivée
  - Augmentés de leur contenu dès la réponse
- On peut donc les intégrer directement dans la vue, ils s'afficheront automatiquement dès que la réponse sera arrivée