

Ajax & services web RESTFul

c 'est quoi un « service web »

- des ressources applicatives : données, fonctionnalités
- exposées et accessibles à distance
- au travers d'une **api** construite au dessus de http (i.e. utilise des urls)
- **pourquoi faire ?**
- Fabriquer des applications web dans lesquelles le serveur se contente de fournir des données et le rendu est réalisée sur le client
 - rend les applications/services programmables et composables
 - modèle d'architecture d'applications web

une api « service web »

- des urls dédiées
- utilisée par des applications (et non un navigateur)
 - app cliente js, app mobile, autre serveur
- retourne des données dans un format exploitable par n'importe quelle application
 - xml, json, ~~HTML~~
- facile d'emploi et d'utilisation par les programmeurs

une api REST

- **Principe** : interagir avec un service web en échangeant des **états** de **ressources**, en se basant sur le **protocole HTTP**
- Les URLs désignent des **ressources** :
 - une ressource = une donnée accessible par l'api
 - désignée par une **uri** : **http://photobox.net/photos/1234**
- ou des **collections** de ressources :
 - ensemble de ressources de même type
 - désignée par une uri : **http://photobox.net/photos/**
- les opérations sur les ressources sont spécifiées par la méthode HTTP : GET, POST, PUT DELETE

une requête REST

- une action :
 - GET, POST, PUT, DELETE
- une RESSOURCE ou une COLLECTION de ressources

```
GET /photos/1234 HTTP/1.1
host: photobox.net
accept: application/json
```

```
POST /photos/1234/comments HTTP/1.1
host: photobox.net
content-type: application/json

{ "pseudo": "mic34" , "content": "nec mergitur" }
```

- des headers précisant la requête
- un body pour transférer des données (POST, PUT)

accéder à des ressources : GET

accéder
une
ressource

GET /photos/3 HTTP/1.1
host: photobox.net
accept: application/json

HTTP/1.1 200 OK
content-type: application/json
{ "id" :38 , "titre" : "montagne",.. }

A
P
I

R
E
S
T
F
U
I

accéder
une
collection
de
ressources

GET /photos/ HTTP/1.1
host: photobox.net
accept: application/json

HTTP/1.1 200 OK
content-type: application/json
[{ } , { } ,]

les réponses

- 1 réponse REST = réponse HTTP
 - **status line + headers + content**
- 1 **status** pour indiquer le succès ou les erreurs
 - 200 OK, 201 Created,
 - 403 – Forbidden, 404 Not Found, 500 Internal Server Error
- utilise les **headers** pour indiquer le type des données :
 - Content-type : application/json
 - Content-type : application/xml
- contenu de la réponse : représentation de la ressource demandée

les cas d'erreurs

- status indiquant l'erreur + Content-type +
- un message « verbeux » dans la réponse, à destination des utilisateurs

HTTP/1.1 404 NOT FOUND
Content-type : application/json

```
{  
  "userMessage" : "la ressource demandée n'a pas été trouvée ; Essayez  
avec une autre valeur, et faites plus attention à ce que vous faites !" ,  
  "errorCode" : "123456",  
  "devMessage" : "RTFM"  
}
```


contenu des reponses

- le **body** de la réponse http contient la description de la ressource demandée ou la collection de ressource demandée
 - en json ou en xml

1 ressource : ../photos/108

```
{
  "photo": {
    "id": 108,
    "titre": "kittens-555822_1280",
    "file": "img_586d38f829ea8.jpg",
    "descr": "Laboriosam et cupiditate accusantium voluptas necessita",
    "format": "JPEG",
    "type": "image/jpeg",
    "size": 483078,
    "width": 1280,
    "height": 711,
    "url": {
      "href": "/photobox/img/large/img_586d38f829ea8.jpg"
    }
  },
  "links": {
    "categorie": {
      "href": "/photobox/photos/108/categorie"
    },
    "comments": {
      "href": "/photobox/photos/108/comments"
    }
  }
}
```

description de
la ressource

url image

liens pour naviguer à
partir de cette
ressource

1 liste : .../photos

```
{
  "photos": [
    {
      "photo": {
        "id": 108,
        "titre": "kittens-555822_1280",
        "file": "img_586d38f829ea8.jpg",
        "thumbnail": {
          "href": "/www/canals5/photobox/img/small/img_586d38f829ea8.jpg"
        },
        "original": {
          "href": "/www/canals5/photobox/img/large/img_586d38f829ea8.jpg"
        }
      },
      "links": {
        "self": {
          "href": "/www/canals5/photobox/photos/108"
        }
      }
    },
    {
      "photo": {
        "id": 107,
        "titre": "kitten-227009_1280",
        "file": "img_586d38f7c6607.jpg",
        "thumbnail": {
          "href": "/www/canals5/photobox/img/small/img_586d38f7c6607.jpg"
        },
        "original": {
          "href": "/www/canals5/photobox/img/large/img_586d38f7c6607.jpg"
        }
      },
      "links": {
```

la photo 108

url image
vignette

url image
originale

url pour accéder
à la ressource

la photo 107

récupérer une ressource en jQuery

```
loadPhotoDetails : function ( uri ) {
```

```
    var pr = $.ajax( uri, {  
        type: 'GET',  
        dataType: 'json',  
        xhrFields: {  
            withCredentials: true  
        }  
    } );
```

GET pour accéder

données réponses en json

pour traverser l'auth. http
en X-domain

```
    pr.fail(function(d, s, x){  
        alert(' ajax call failed : code '+ s);  
    });
```

alerte en cas d'erreur

```
    return pr;
```

renvoyer une promise

```
},
```

```
loadPhotoDetails( photo_uri )  
    .done( update_lightbox );
```

appeler la fonction de chargement
transmettre les données au callback

créer, modifier, supprimer des ressources REST

- On utilise les mêmes URI
- on change la méthode HTTP :
 - **POST** : pour créer ou modifier une ressource dont on ne connaît pas l'ID
URI : collection → POST /photos
POST /photos/42/comments
 - **PUT** : pour créer ou remplacer une ressource dont on connaît l'ID
URI : ressource → PUT /photos/128
 - **DELETE** : pour supprimer une ressource
URI : ressource → DELETE /photos/42

METHOD	/photos/23/comments	/photos/23
POST create	Crée une nouveau commentaire pour la photo 23 – les données sont dans le body	ERREUR
PUT update	ERREUR	remplace la photo 23 si existe, sinon création de la photo 23 si autorisé sinon erreur (les données sont dans la requête)
DELETE delete	Supprime la liste complète des commentaires de la photo 23 (à éviter)	Supprime photo 23

créer
une
ressource

POST /photos/23/comments HTTP/1.1
host: photobox.net
accept: application/json
content-type: application/json

{ "titre"="yo", "pseudo"="geg23" ... }

HTTP/1.1 **201 CREATED**
content-type: application/json
Location: /comments/104

{ "id" :104 , "titre" : "yo",.. }

A
P
I

R
E
S
T
F
U
I

modifier
une
ressource

PUT /plcommentsats/104 HTTP/1.1
host: photobox.net
content-type:: application/json

{ "titre"="yo les gars",
 "pseudo"="geg23" ... }

HTTP/1.1 **204 NO CONTENT**
content-type: application/json

exemple dans le projet

- Créer un commentaire pour une photo :
 - POST
 - url : /photobox/photos/<id>/comments
 - données :
 - pseudo : pseudo de l'auteur ajoutant le commentaire
 - titre : titre du commentaire
 - content : contenu du commentaire

POST /photos/42/comments HTTP/1.1
host: api.prolunch.net
content-type: application/x-www-form-urlencoded
pseudo=mich28&titre=sed%20fluctuat&content=nec

créer
une
commande

HTTP/1.1 **201 CREATED**

Access-Control-Allow-Credentials: true
Access-Control-Allow-Origin: chrome-extension:...
Content-Type: application/json; charset=utf-8

```
{
  "comment": {
    "titre": "sed fluctuat",
    "pseudo": "mich28",
    "content": "nec mergitur, ...",
    "id": 332,
    "date": "03/03/2017"
  }
}
```

A
P
I

P
R
O
L
U
N
C
H

poster des données avec jQuery

```
postComment : function ( uri, comment) {

    var pr = $.ajax( uri, {
        type: 'POST',
        dataType: 'json',
        xhrFields: {
            withCredentials: true
        },
        data : comment
    });

    pr.fail(function(d,s,x) {
        alert(' ajax call failed : code ' + s);
    });

    return pr;
},
```

Données à transmettre sous la forme d'un objet JS converti en string urlencodée par jQuery

[illegible]