

# Mini-projet Programmation web client riche

Le but de ce travail est de réaliser une page html qui permettra à un utilisateur de rechercher des photos relatives à des villes françaises sur le site **Flickr** <https://www.flickr.com/>. Dans le but de rechercher des photos, vous utiliserez un des services web que propose ce site. Deux solutions s'offrent à vous. Soit vous utilisez le service web accessible sans authentification (voir exemple du cours), soit vous avez un compte sur **Flickr** (vous pouvez d'ailleurs vous en créer un) et vous disposez de services qui demandent votre clé d'authentification pour récupérer des données. Pour plus d'information voir: <https://www.flickr.com/services/api/>

Vous aurez besoin aussi d'un service web qui vous permettra de récupérer le nom des communes françaises pour réaliser de l'auto-complétion d'un champs de saisie utilisateur. Je vous en propose un à l'adresse : <http://infoweb-ens/~jacquinc/codePostal/commune.php>. Il faut lui passer via la méthode GET les données sous ce format : "commune=nantes&maxRows=10" par exemple. Si vous lui passez "commune="nan", il récupérera toutes les communes de la base qui commencent par "nan". Il retourne des données au format JSON (le champs maxRows n'est pas obligatoire dans les données passées dans la requête)

Vous pouvez aussi utiliser un service web sur internet qui réalise la même chose (à chercher et il faut qu'il accepte les requêtes *ajax* inter-domaines !)

## Travail demandé :

Vous allez construire une page html (+ css et script(s) javascript associé(s))

- elle comprend un champs de saisie utilisateur où est demandée la saisie d'un nom de commune française. L'auto-complétion relative à ce champs est mise en place.
- une liste ou autre chose qui permettra à l'utilisateur de saisir le nombre maximal de photos qu'il veut afficher. Lorsque l'utilisateur clique sur une icône de votre choix (<https://api.jqueryui.com/theming/icons/>), la recherche de photos est lancée.

Vous mettrez en place 2 onglets (*UI tabs*) qui permettront de visualiser les données de 2 manières différentes: vue "photo" et vue "tableau".

### a) **Vue photo**

- les photos s'affichent les unes en dessous des autres sans information spécifique. Si aucune photo n'est disponible pour cette commune, une fenêtre

modale le stipule (*UI Dialog*)

- lorsque l'utilisateur clique sur une photo, des informations relatives à la photo (le nom de la photo, l'heure de prise de vue, l'identifiant du photographe,...) s'affichent dans une fenêtre modale (*UI Dialog*)

## b) **Vue table**

- les photos s'affichent dans une table avec dans chaque ligne, une vignette de la photo + dans les autres cellules de la table, les informations relatives à la photo ( le nom de la photo, l'heure de prise de vue, l'identifiant du photographe, ...). Utiliser le plugin *DataTables* pour gérer la table. Vous pourrez aussi accéder aux différentes pages de la table. L'utilisateur pourra aussi choisir le nombre de lignes que contiennent les pages de la table. Il faudra mettre en place un tri sur toutes les colonnes de la table (à l'aide du plugin *DataTables*)
- L'utilisateur pourra aussi demander à visualiser les photos qui ont été prises à partir d'une certaine date. L'utilisateur pourra entrer la date soit manuellement soit via un calendrier (à rajouter au début de la page) (*UI DatePicker*)

**Bonus:** pour chaque photo, si l'information de géolocalisation est disponible, la position géographique de sa prise de vue pourra être affichée sur une carte Google Maps (*plugin gmaps par exemple*).

Vous pouvez vous aider de ce lien pour connaître comment récupérer les coordonnées de géolocalisation d'une photo sur **Flickr** si elles sont disponibles: <http://kylerush.net/blog/tutorial-flickr-api-javascript-jquery-ajax-json-build-detailed-photo-wall/>

Ce mini-projet est à réaliser en jQuery exclusivement (pas d'autres frameworks ou librairies type AngularJS, ...). Vous utiliserez des UI ou plugins jQuery. Le rendu visuel de la page sera aussi pris en compte dans la notation.

## **RENDU ATTENDU**

Le projet est à réaliser en binôme. Il faudra le rendre au plus tard le dimanche **3 avril à minuit** sous *madoc*.

Le mini-projet sera stocké dans un répertoire qui sera nommé *nom1\_nom2* et qui sera ensuite compressé et nommé *nom1\_nom2.tar.gz* (*nom1* et *nom2* sont les noms de famille des étudiants du binôme). Vous prendrez garde que le correcteur puisse exécuter votre code sans problème (pas de chemin absolu...). Il faudra rendre dans une unique archive. :

- un rapport en **pdf** qui donnera l'état d'avancement de chaque point demandé et explicitera ce qui n'a pas été réalisé le cas échéant.
- le code javascript (jquery) commenté