

Web Services & Application Répartie

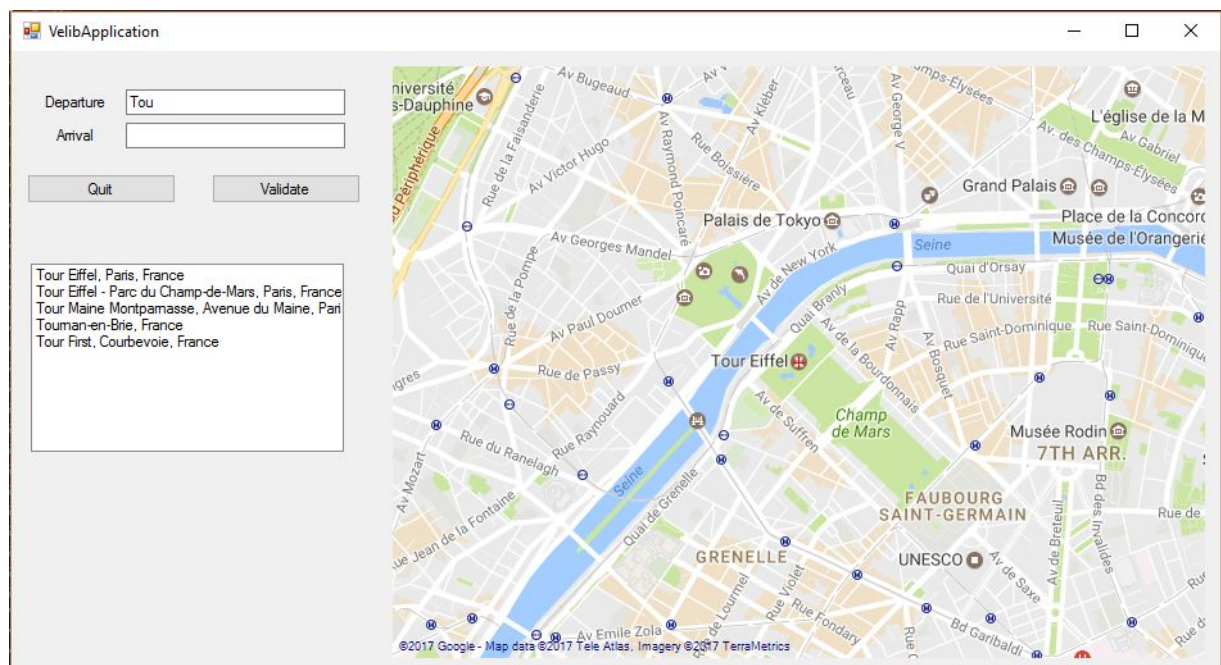
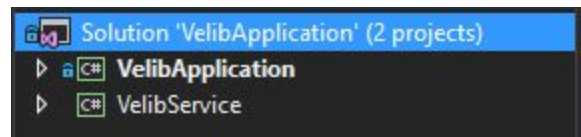
Maxime Carlier - Yasin Eroglu



Question 1 : Optimisation de l'utilisation des Vélib's de la ville de Paris

Détail de la solution

La Solution que nous livrons pour la question 1 est composée de deux Projets.



VelibApplication

Le projet VelibApplication est un projet WinForm proposant une interface minimaliste destinée au client qui lui permet de saisir facilement un point de départ et d'arrivée à l'aide de champ de texte. L'application affiche une liste de suggestion pour la saisie utilisateur afin de faciliter la sélection des point de départ et d'arrivée.

VelibService

Le projet VelibService est un projet WCF (SOAP en v1) qui expose une unique opération **GetJourney**. Celle-ci renvoie un contrat de donnée **Journey** qui permet à partir d'une adresse de départ et d'arrivée, d'obtenir quatre points GPS représentant :

- Le point de départ
- La station de départ
- La station d'arrivée
- Le point d'arrivée

Ces points GPS permettent ainsi à quiconque utilise notre Web Service de savoir où prendre son vélo (Station la plus proche avec un vélo disponible), à quelle station se rendre en vélo (Station la plus proche du point d'arrivée avec un emplacement de libre) puis de se rendre à la destination finale.

UI Client

Afin de rendre le tout le plus visuellement satisfaisant lors d'une utilisation depuis notre application client, une fois que les quatre points GPS sont obtenus à l'aide du **WS VelibService**, le client effectue quelques requêtes supplémentaires sur l'API **Google Directions** afin d'afficher un tracé de l'itinéraire.

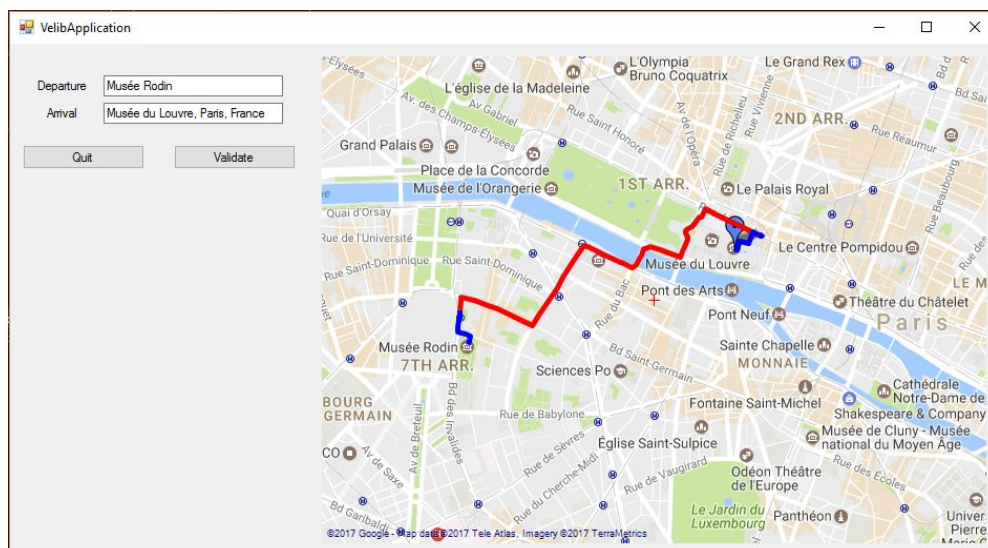
```
[ServiceContract]
public interface IJourneyService
{
    [OperationContract]
    Journey GetJourney(string departure, string arrival);
}

[DataContract]
public class Journey {
    [DataMember]
    public Coordinates OriginCoordinates;

    [DataMember]
    public Coordinates StartStationCoordinates;

    [DataMember]
    public Coordinates EndStationCoordinates;

    [DataMember]
    public Coordinates DestinationCoordinates;
```



On remarque en bleu le trajet piéton pour se rendre aux stations de l'itinéraire et en rouge le trajet en Vélib's.

Question 2 : Tarification des services

Services Web	STANDARD	PREMIUM
Google Maps Directions API Google Maps Distance Matrix API ⁴ Google Maps Elevation API Google Maps Geocoding API Google Maps Geolocation API Google Maps Roads API Google Maps Time Zone API	Gratuit jusqu'à 2 500 requêtes par jour. 0,50 USD pour 1 000 requêtes supplémentaires, jusqu'à 100 000 par jour, si la facturation est activée.	Tarif basé sur le volume souhaité. Fonctionnalités améliorées du plan Premium : <ul style="list-style-type: none">• Extensions pour les requêtes par seconde.• Fonctionnalités améliorées de Distance Matrix API et de Roads API.² Pour plus d'informations, voir Taux et limites d'utilisation de Premium Plan .
Google Places API Web Service	150 000 requêtes gratuites par jour (après validation de la carte de crédit).	Tarif basé sur le volume souhaité. Pour plus d'informations, voir Taux et limites d'utilisation de Premium Plan .

[ACTIVER LA FACTURATION](#)
(pour déverrouiller les quotas)
[Voir les conditions de service des plans Standard](#)

[CONTACTER LE SERVICE COMMERCIAL](#)
(pour en savoir plus sur les plans Premium)
[Voir les conditions de service des plans Premium](#)

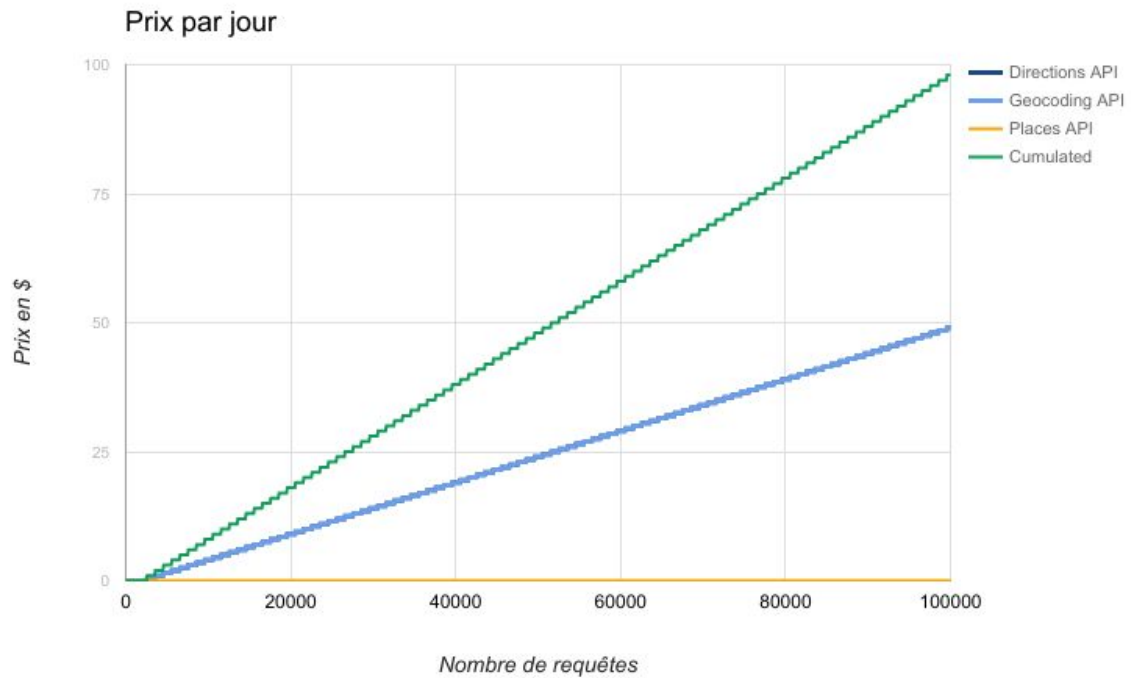
Dans cette partie nous allons donner une évaluation des tarifs pour une utilisation de notre service. Nous utilisons exclusivement des APIs de Google comme :

- Google Direction API
- Google Geocoding API
- Google Places API

Pour chacun de ces APIs, on peut constater sur le tableau ci-dessus que nous avons un nombre de requête gratuit par jour. Nous avons alors pour les APIs Direction et Geocoding 2500 requêtes gratuites par jour et pour l'API Places 150 000 requêtes gratuites par jour (à condition de fournir un numéro de Carte Bancaire valide) sinon cette limite tombe à 1 000 requêtes journalières.

Mais pour une utilisation professionnelle, nous ne pouvons pas nous limiter à cela en ce qui concerne les APIs Direction et Geocoding. Il nous faudra donc nous attendre à dépasser les quotas gratuits journalière, et payer en conséquence.

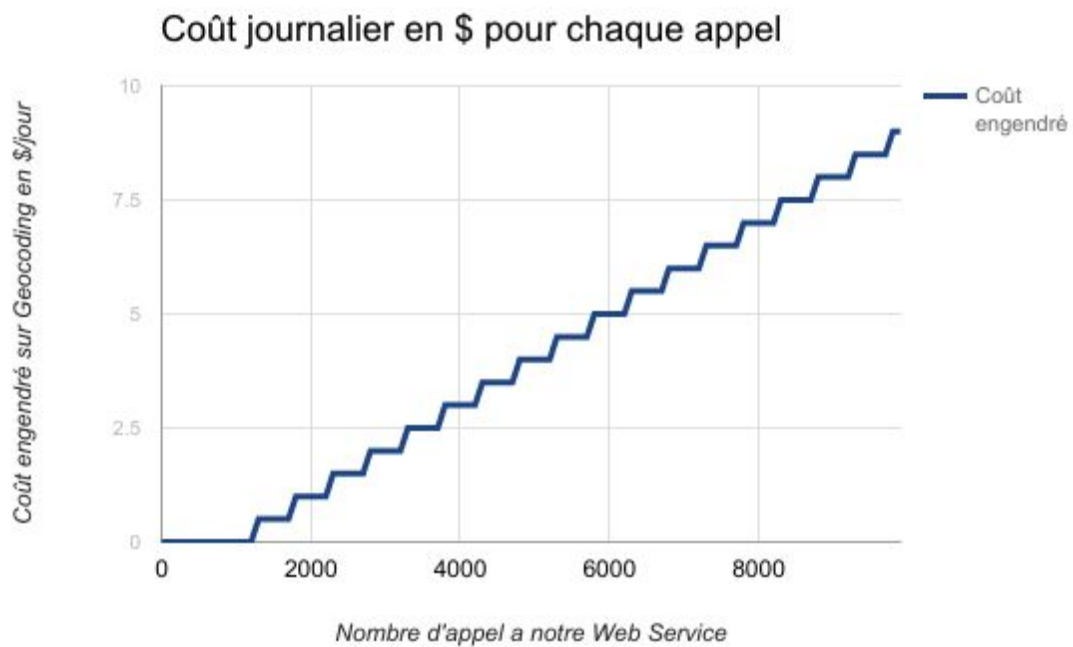
Pour ceux la, il nous faut nous attendre à payer pour chaque 1 000 requêtes supplémentaire, qui nous seront facturé 0.50\$ jusqu'à un seuil maximum de 100 000 requête journalières (soit un total de 48.5\$/jour par services, donc 97\$ pour les deux services par jour).



Le graphe ci-dessus résume le pricing appliqué par Google pour les API concerné par notre Projet. On notera que Geocoding et Directions partage le même pricing.

Rentabilité du Web Service uniquement

Nous allons à présent nous pencher sur le prix de chaque appel à notre Web Service **uniquement** en nous basant sur le pricing ci-dessus. A chaque appel de notre Web Service, est associé 2 appel à l'API Geocoding de map. Ce qui nous permet d'obtenir la courbe de prix suivant :



Ainsi, le seuil de rentabilité de notre Web Service est de 0\$ si chaque jour notre Web Service est appelé moins de 1250 fois.

Sinon pour chaque 500 appel à notre Web Service, notre seuil de rentabilité augmente de 0,5\$, pour un maximum de 50.000 appel journalier coûtant 49\$.

Coût d'utilisation du Web Service facturé au client

Afin de donner un coût final de l'utilisation de notre Web Service, on suppose que de la même manière que Google, nous sommes en mesure de tenir compte du nombre de requête effectué par chaque Clients.

Comme nous ne souhaitons pas avoir de pertes, nous n'appliquons pas le système des X première requêtes gratuite, et comme nous souhaitons offrir un prix intéressant, nous adoptons le même système de seuil avec une marge commerciale de 40%.

Ainsi la grille tarifaire est la suivante :

Nombre de requête journalière du client	Tarif (\$)
0 - 1 250	0,7\$
1 250 - 1 750	1,4\$
1 750 - 2 250	2.1 \$

et ainsi de suite jusqu'à un maximum de 50 000 requête journalière. Au delà un devis sera nécessaire.

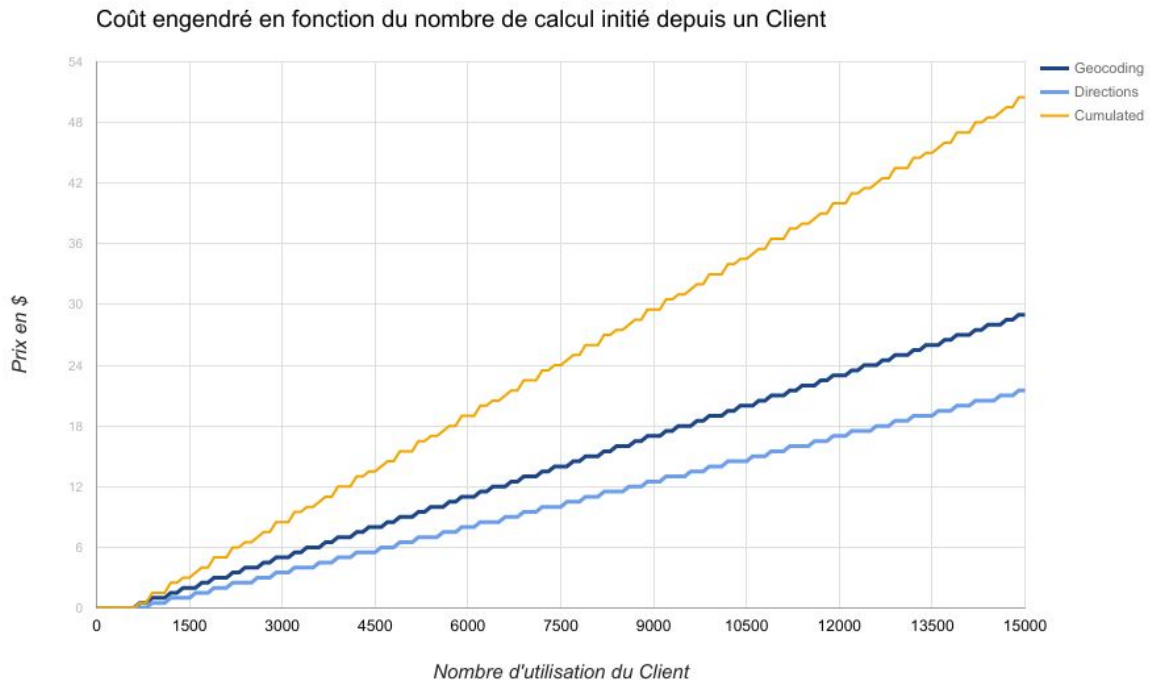
Rentabilité du Web Service + Client

Si on s'intéresse cette fois au coût engendré par le Web Service et le Client, le prix est notablement plus élevé. En effet, notre client fait :

- En moyenne 5 appel a Places pour l'auto Suggestion de la saisie utilisateur par champs (2 champ au total) (CAD 10 appel à Places)
- 2 Appel à Geocoding pour afficher les marqueur sur la carte une fois le point de départ et d'arrivé choisis
- 1 appel à notre Web Service (CAD 2 Appel à Geocoding)
- 3 Appel à Directions pour obtenir les itinéraires nécessaires.

Le facteur limitant est ici Geocoding car au delà de 100 000 appel, un devis commercial est nécessaire.

Cela nous donne donc la courbe du coût suivante :



On remarquera que dans ce modèle, notre solution ne peut être exécutée plus que 25.000 fois par jours pour un coût journalier maximum atteignant les 85,5\$.

Coût d'utilisation de l'application facturée au client

Si on suppose cette fois que nous livrons notre application au client, alors les tarifs seront différent, mais toujours calculé de la même manière (pas gratuit en dessous d'un seuil, marge de 40%).

Nombre d'utilisation journalière de l'application par le client	Tarif (\$)
0 - 2 000	7\$
2 000 - 4 000	16,8\$
4 000 - 6 000	26,6\$

Et ainsi de suite.

Charge fixe

Enfin si on souhaite pousser la question jusque dans ses limites, on notera qu'il reste à ajouter a ces Tarifs, les charges fixe engendré par l'hébergement de notre service sur un service type AWS. Mais pour résoudre cette problématique il faut se baser sur les besoins CPU et Réseau, ce qui a ce stade n'est pas possible pour nous.

Question 3 : Multi-bindings Extranet / Intranet

Non traités car nous n'avons pas réussi à configurer le https depuis App.Config.