

UNIVERSITÉ DE MONTPELLIER

M1 AIGLE

Rapport : TP REST

Étudiants :

Denis BEAUGET, Hayaat HEBIRET

Encadrant :

M. Abdelhak SERAI

Année : 2020 – 2021

Lien du projet :

<https://github.com/Beauget/TPServiceWebREST>



Microsoft
.NET



UNIVERSITÉ
DE MONTPELLIER

Table des matières

1 Objectifs	2
2 Exercice 1 :	2
2.1 Postman :	2
3 Exercice 2 :	4
4 Question 1 :	4
4.1 UML version distribuée :	4
4.2 Explication :	4
4.3 Résultat	6
5 Question 3 :	8
5.1 UML : Comparateur	8
5.2 Explications :	8
5.3 Résultat	9
Extrait de codes	10

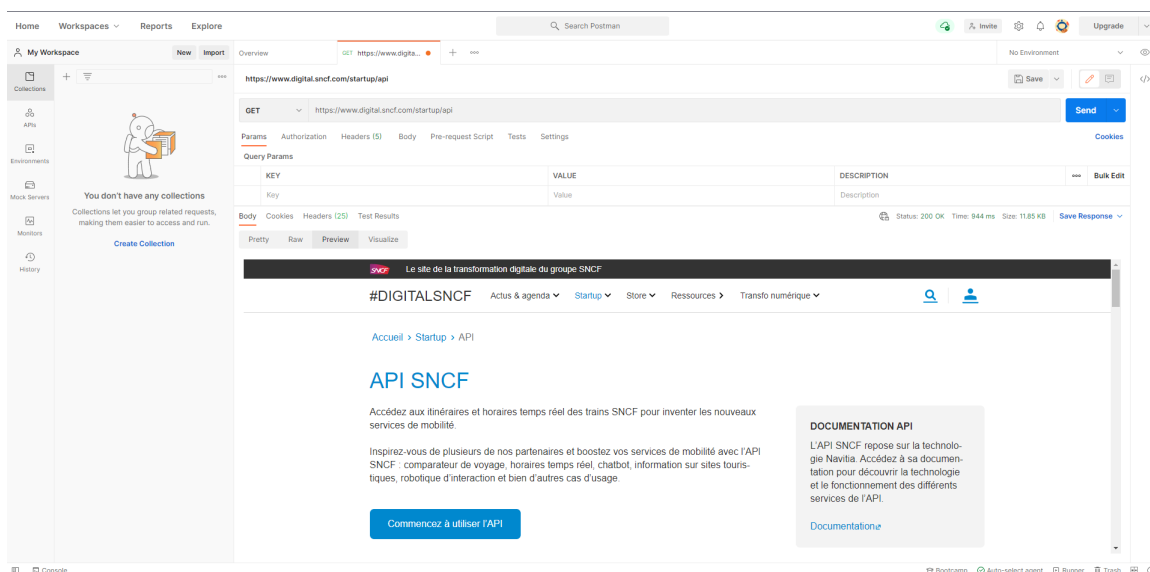
1 Objectifs

L'objectif de ce TP est de reprendre le thème de l'application de gestion d'hôtels du TP SOAP mais d'utiliser les services REST (representational state transfer) plutôt que les services SOAP. Dans un premier temps à travers des exemples de services REST puis en implémentant plusieurs services pour notre application.

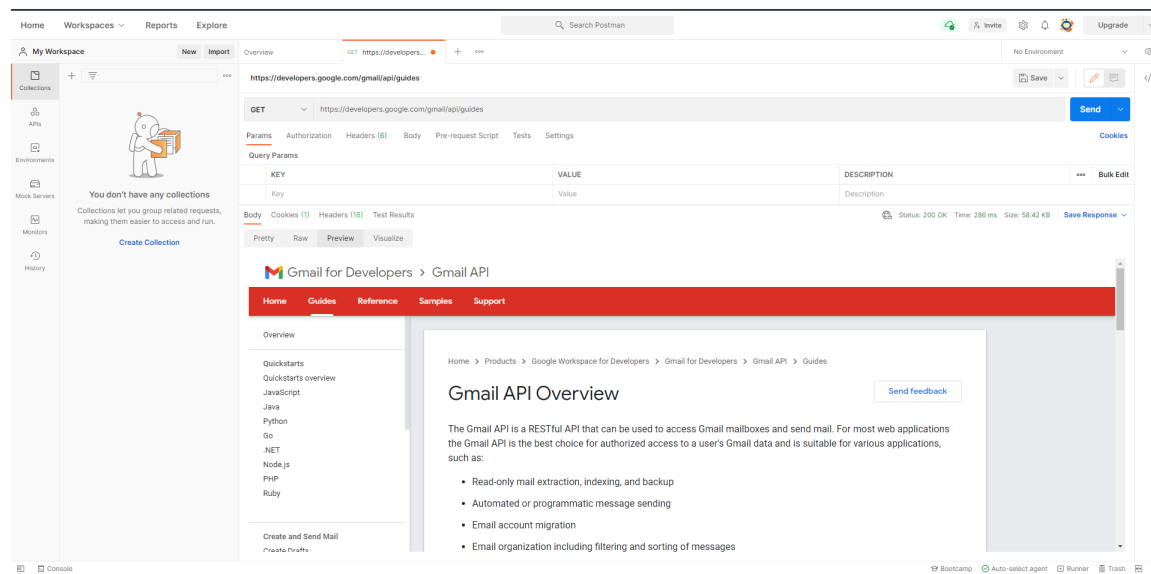
2 Exercice 1 :

2.1 Postman :

Postman est un outil permettant de tester des API Web, nous avons donc tester 2 services dans le cadre de cet exercice, celui de la SNCF et celui de Gmail.



Résultat : Requête GET sur l'api de la SNCF

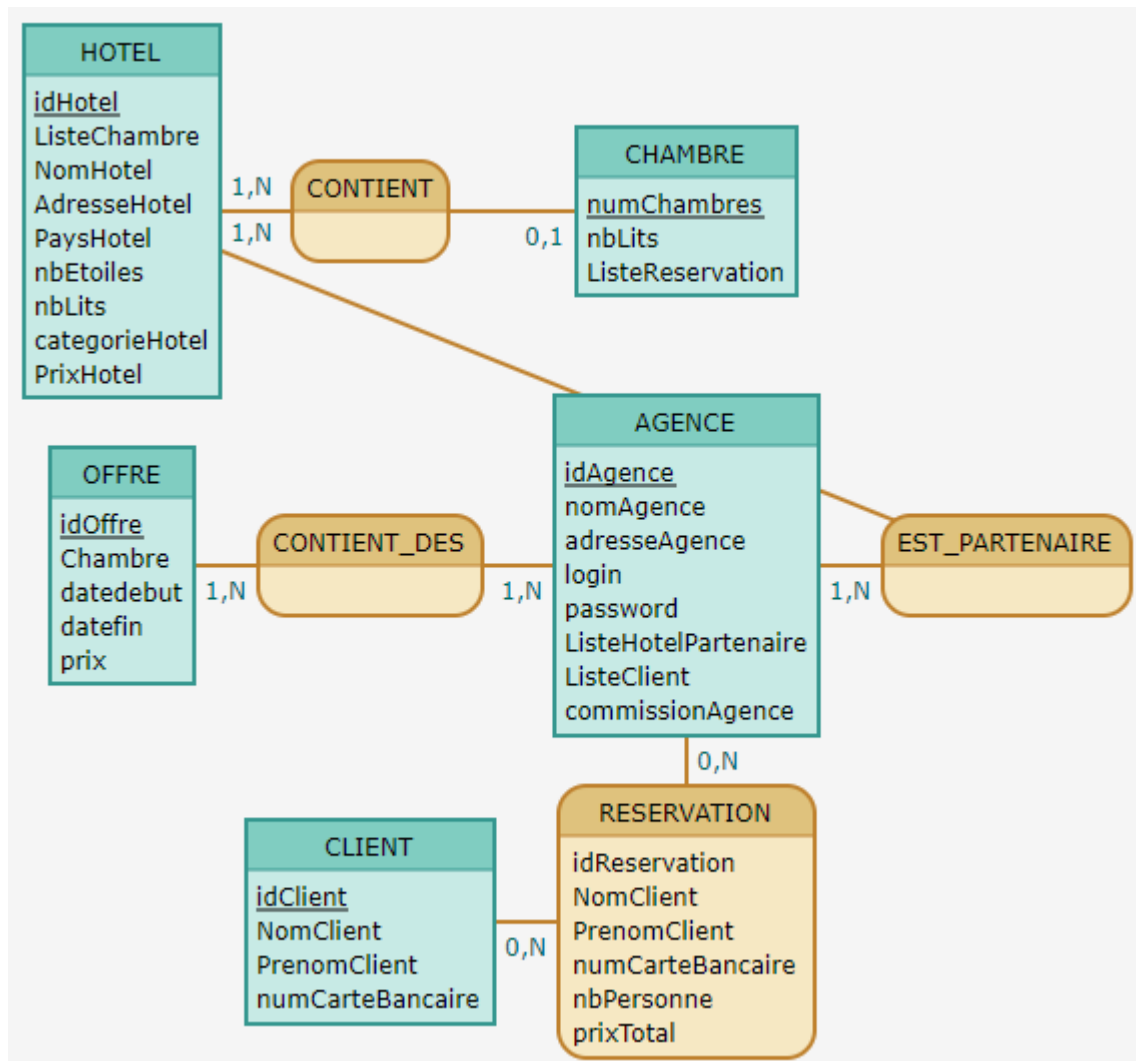


Résultat : Requête GET sur l'api Gmail

3 Exercice 2 :

4 Question 1 :

4.1 UML version distribuée :



4.2 Explication :

Pour cette exercice nous avons décidé de reprendre l'UML de la version distribuée du TP SOAP et de réutiliser la logique métier lié à celle-ci.

Pour cette version et pour ne pas juste reproduire le schéma du TP SOAP nous avons créer 2 services REST, chacun représentant un hôtel (1 : HotelPasCher , 2 : HotelPlusCher) avec chacun sa propre liste d'offres (notamment pour anticiper le futur comparateur).

Nous avons toujours 2 solutions, la première représentant le "serveur" avec la logique métier et les services web (représentant les hôtels) et de l'autre une solution console envoyant des requêtes au services web et représentant l'interface IHM pour les agences.

La différence majeur entre les 2 méthodes est la façon de "requêter" les services, en REST chaque méthoqe porte une annotation (GET/PUT/POST/DELETE..) et on appelle les méthodes directement via des requêtes HTTP dont on stock le résultat dans des variables (des String la plupart du temps pour nous). Pour cela, tout au long du TP nous avons utilisé une méthode "GetReleases" permettant d'exécuté la requête et de parser le résultat.

```

1 public static String GetReleases(String url)
2     {
3         var request = (HttpWebRequest)WebRequest.Create(url);
4
5         request.Method = "GET";
6         requestAutomaticDecompression = DecompressionMethods.Deflate |
7         DecompressionMethods.GZip;
8
9         var content = string.Empty;
10        using (var response = (HttpWebResponse)request.GetResponse())
11        {
12            using (var stream = response.GetResponseStream())
13            {
14                using (var sr = new StreamReader(stream))
15                {
16                    content = sr.ReadToEnd();
17                }
18            }
19        }
20
21        String resultat = content.ToString();
22        return resultat.Replace("\n", "");
23    }

```

Méthode : GetReleases

```

1 string offres = GetReleases(url + "HotelPasCher/AfficherOffreDisponible?
2 login=" + login +
3 "&password=" + password +
4 "&dateArrive=" + dateD +
5 "&dateDepart=" + dateF +
6 "&nbPersonne=" + nbPersonnes);

```

Exemple : Requête via la méthode GetReleases() pour afficher les offres

Commandes disponibles :

- Choisir entre les 2 hôtels partenaire :

L'utilisateur commence par choisir l'un de nos 2 hôtels avant d'afficher les offres (qui dépendent de l'hôtel choisis).

- Afficher les offres (sans images) :

L'utilisateur peut afficher toute les offres de notre hôtel sans images pour faire son choix.

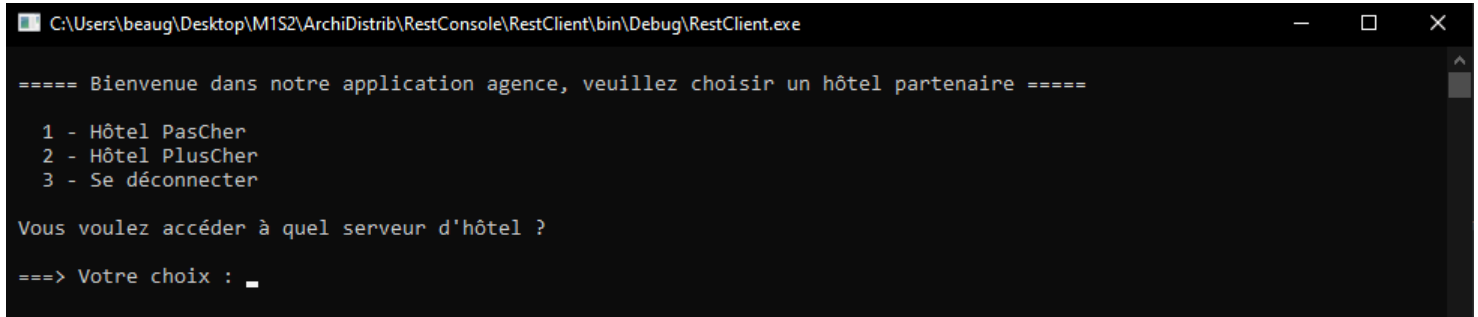
- Afficher les offres (avec images et GUI) :

A l'image du TP SOAP, l'utilisateur peut afficher chacune des offres avec une images de la chambre à chaque fois et la possibilité de réserver en direct via le GUI.

- Faire une réservation :

L'utilisateur peut réserver l'offre qui l'intéresse avec l'ID de celle-ci et ses informations client.

4.3 Résultat



```
C:\Users\beaug\Desktop\MIS2\ArchiDistrib\RestConsole\RestClient\bin\Debug\RestClient.exe

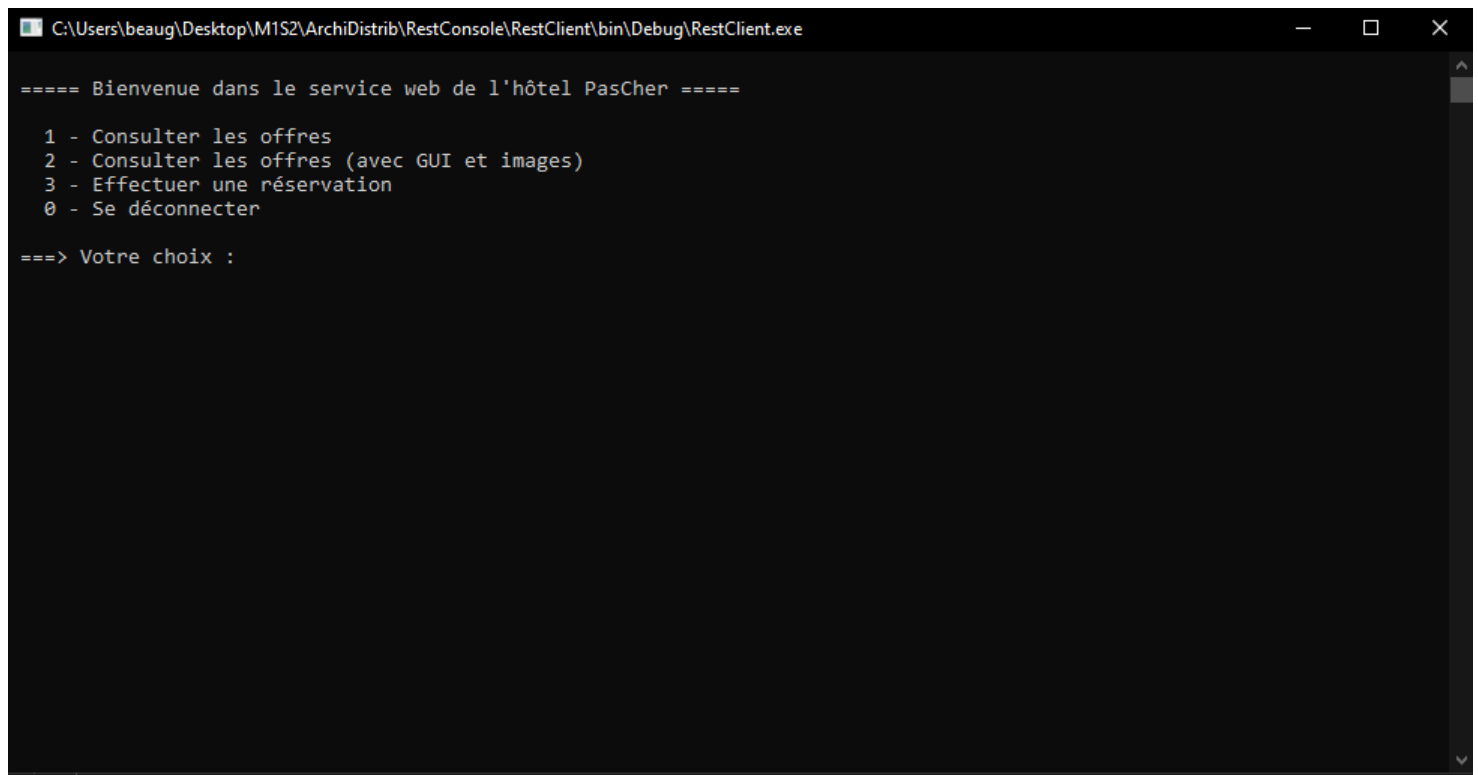
===== Bienvenue dans notre application agence, veuillez choisir un hôtel partenaire =====

1 - Hôtel PasCher
2 - Hôtel PlusCher
3 - Se déconnecter

Vous voulez accéder à quel serveur d'hôtel ?

==> Votre choix : █
```

Figure 1 : Choix de l'hôtel



```
C:\Users\beaug\Desktop\MIS2\ArchiDistrib\RestConsole\RestClient\bin\Debug\RestClient.exe

===== Bienvenue dans le service web de l'hôtel PasCher =====

1 - Consulter les offres
2 - Consulter les offres (avec GUI et images)
3 - Effectuer une réservation
0 - Se déconnecter

==> Votre choix :
```

Figure 2 : Menu de l'application

```

C:\Users\beaug\Desktop\M1S2\ArchiDistrib\RestConsole\RestClient\bin\Debug\RestClient.exe
Veuillez entrer les informations suivantes : login de l'agence, mot de passe de l'agence, date d'arrivée et de départ et
nombre de personnes
loginAgence1
admin1
Entrez la date de debut au format dd/mm/yyyy :
03/04/2021
Entrez la date de fin au format dd/mm/yyyy :
10/04/2021
2
- Identifiant      : IbisBudget-1
- Numéro de chambre : 2
- Nombre de lit     : 0
- Prix de l'offre   : 294

- Identifiant      : IbisBudget-2
- Numéro de chambre : 1
- Nombre de lit     : 1
- Prix de l'offre   : 168

- Identifiant      : IbisBudget-3
- Numéro de chambre : 3
- Nombre de lit     : 2
- Prix de l'offre   : 126

- Identifiant      : IbisBudget-4
- Numéro de chambre : 4
- Nombre de lit     : 3
- Prix de l'offre   : 168

```

Figure 3 : Afficher les offres sans GUI

```

C:\Users\beaug\Desktop\M1S2\ArchiDistrib\RestConsole\RestClient\bin\Debug\RestClient.exe
===== Service : Réservation d'une offre partenaire =====
Veuillez entrer le login de l'agence et son mot de passe, l'ID de l'offre choisis, votre nom, votre prénom, le nombre de
personne et votre numéro de carte bancaire pour la réservation
loginAgence1
admin1
IbisBudget-1
beauget
denis
2
12334
- Nom Client      : beauget
- Prenom Client   : denis
- Prix total      : 40
- Référence carte : 12334
RESERVATION PRISE EN COMPTE, MERCI DE VOTRE VISITE

Appuyer sur une touche pour revenir en arrière...

```

Figure 4 : Effectuer une réservation

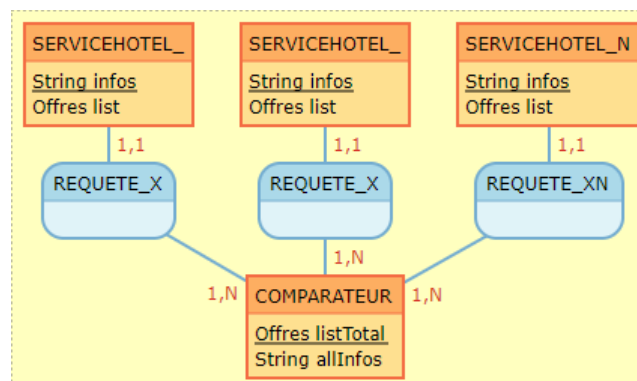
The screenshot shows a web application window titled 'Agence'. It displays details for an offer 'IbisBudget-4'. The 'Informations' section lists: Nom de l'hôtel : IbisBudget, Adresse : 230 Avenue des roses, Etoiles : 2, Prix nuit/personne : 35, and Prix total de l'offre : 168€. It also shows the date de disponibilité (Arriver : 09/04/2021, Départ : 13/04/2021). A photo of a hotel room with two beds is shown. Below the photo, it says 'Photo de la chambre (ci-dessous) et Nombre de lits : 4 lits'. On the right, a 'Réservation' form asks 'Cette offre vous plaît ? Remplissez notre formulaire de réservation :'. The form includes fields for 'IdOffre' (filled with 'IbisBudget-4'), 'Nom', 'Prénom', 'Nb Personnes', and 'Numéro CB'. There are 'Réserver' and 'Reset' buttons. At the bottom right, there is an 'Offre suivante' button.

Figure 5 : GUI de l'application

5 Question 3 :

L'objectif de cette question était de créer une petite application "comparateur" permettant de comparer les offres de plusieurs hôtel en fonction des informations passé en paramètre.

5.1 UML : Comparateur



UML : Comparateur

5.2 Explications :

Nous avons créé un service supplémentaire : Comparateur. Celui-ci s'occupe de créer les requêtes pour les hôtels à partir des informations fournis par l'utilisateur, il récupère les informations sur les offres renvoyé par les hôtels et les renvoie à l'interface IHM.

Commandes disponibles :

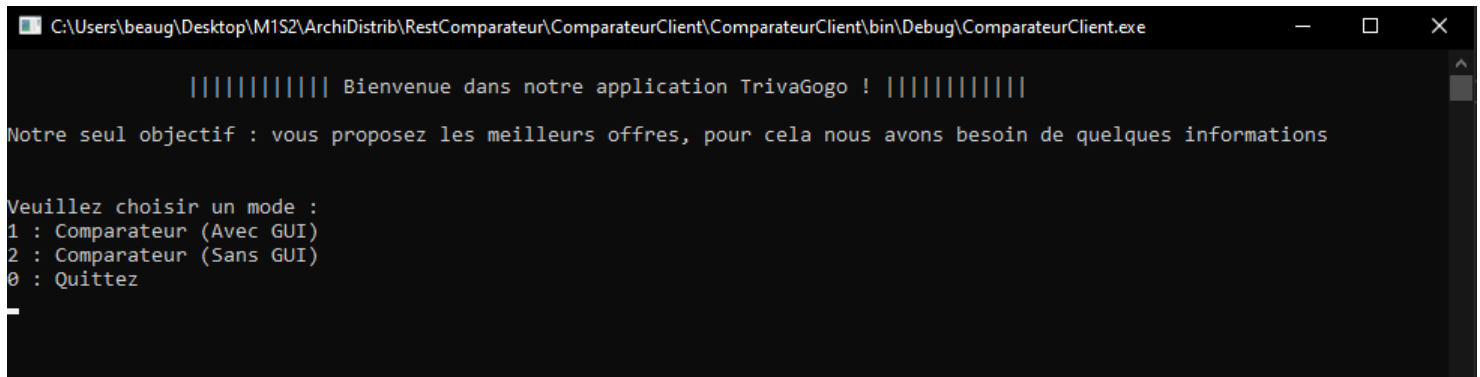
- Afficher les offres (sans GUI) :

L'utilisateur commence par fournir les différentes informations nécessaires et l'application renvoie les offres (fictive) à l'utilisateur.

- Afficher les offres (avec GUI) :

L'utilisateur commence par remplir un formulaire et après avoir cliquer sur le bouton "Comparer" il recevra en retour une liste d'offres et pourra appuyer sur "Reset" pour effectuer une nouvelle recherche.

5.3 Résultat



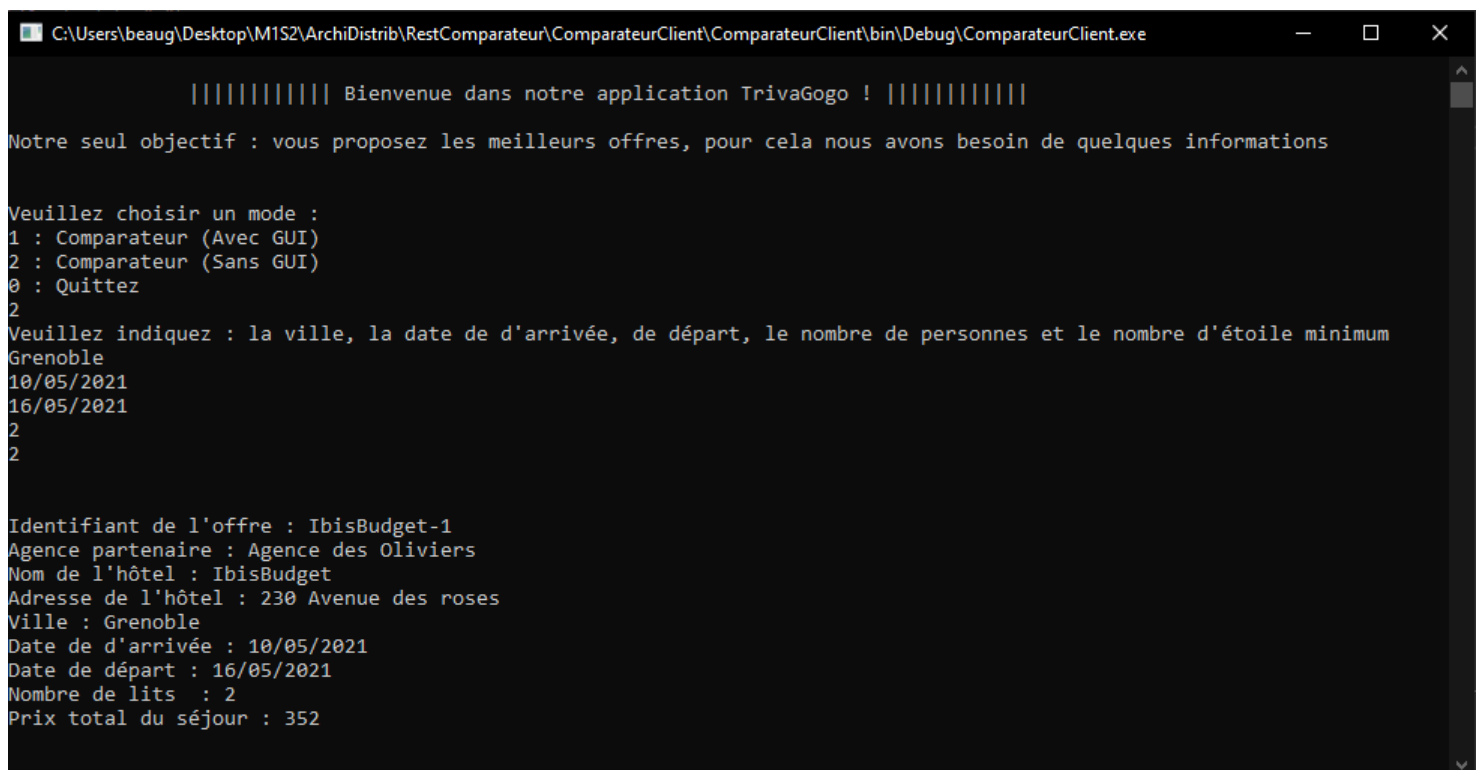
```
C:\Users\beaug\Desktop\M1S2\ArchiDistrib\RestComparateur\ComparateurClient\ComparateurClient\bin\Debug\ComparateurClient.exe

||||| Bienvenue dans notre application TrivaGogo ! |||||

Notre seul objectif : vous proposez les meilleurs offres, pour cela nous avons besoin de quelques informations

Veuillez choisir un mode :
1 : Comparateur (Avec GUI)
2 : Comparateur (Sans GUI)
0 : Quittez
```

Figure 1 : Choix de l'option



```
C:\Users\beaug\Desktop\M1S2\ArchiDistrib\RestComparateur\ComparateurClient\ComparateurClient\bin\Debug\ComparateurClient.exe

||||| Bienvenue dans notre application TrivaGogo ! |||||

Notre seul objectif : vous proposez les meilleurs offres, pour cela nous avons besoin de quelques informations

Veuillez choisir un mode :
1 : Comparateur (Avec GUI)
2 : Comparateur (Sans GUI)
0 : Quittez
2
Veuillez indiquer : la ville, la date de d'arrivée, de départ, le nombre de personnes et le nombre d'étoile minimum
Grenoble
10/05/2021
16/05/2021
2
2

Identifiant de l'offre : IbisBudget-1
Agence partenaire : Agence des Oliviers
Nom de l'hôtel : IbisBudget
Adresse de l'hôtel : 230 Avenue des roses
Ville : Grenoble
Date de d'arrivée : 10/05/2021
Date de départ : 16/05/2021
Nombre de lits : 2
Prix total du séjour : 352
```

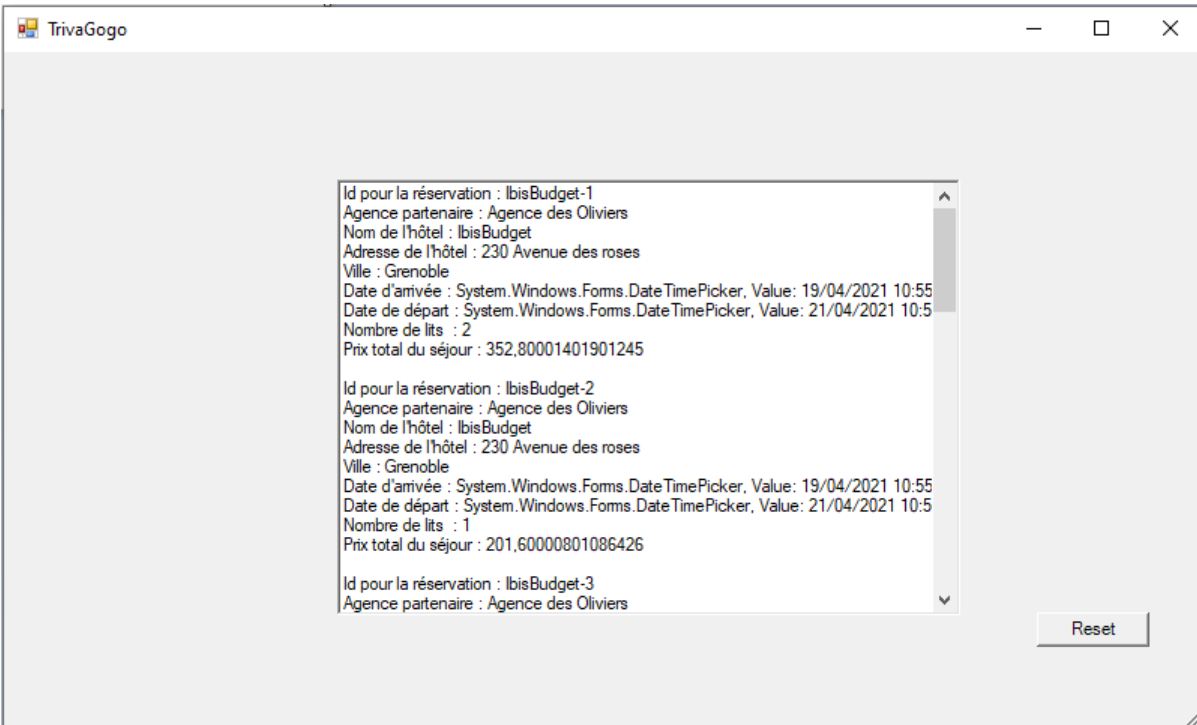
Figure 2 : Résultat du comparateur (sans GUI)



The screenshot shows a window titled 'TrivaGogo' with a form titled 'Vos informations'. The form contains the following fields:

- Ville:
- Date départ:
- Date arrivée:
- Personnes:
- Nombre étoiles:

A 'Comparer' button is located to the right of the 'Nombre étoiles' field.

Figure 3 : Formulaire du GUI

The screenshot shows the same 'TrivaGogo' window, but now displaying search results in a scrollable list. The results are as follows:

- Id pour la réservation : IbisBudget-1
Agence partenaire : Agence des Oliviers
Nom de l'hôtel : IbisBudget
Adresse de l'hôtel : 230 Avenue des roses
Ville : Grenoble
Date d'arrivée : System.Windows.Forms.DateTimePicker, Value: 19/04/2021 10:55
Date de départ : System.Windows.Forms.DateTimePicker, Value: 21/04/2021 10:5
Nombre de lits : 2
Prix total du séjour : 352,80001401901245
- Id pour la réservation : IbisBudget-2
Agence partenaire : Agence des Oliviers
Nom de l'hôtel : IbisBudget
Adresse de l'hôtel : 230 Avenue des roses
Ville : Grenoble
Date d'arrivée : System.Windows.Forms.DateTimePicker, Value: 19/04/2021 10:55
Date de départ : System.Windows.Forms.DateTimePicker, Value: 21/04/2021 10:5
Nombre de lits : 1
Prix total du séjour : 201,60000801086426
- Id pour la réservation : IbisBudget-3
Agence partenaire : Agence des Oliviers

A 'Reset' button is located at the bottom right of the window.

Figure 4 : Résultat du GUI

Extrait de codes

Ces extraits de code sont des briques logiciel que nous avons souhaité intégrer dans le rapport, elles regroupent souvent plusieurs appels à des fonctions externe. L'objectif est de montrer nos fonctions "synthèse" mais qui sont le "cœur" de notre application.

sans pour autant intégrer l'intégralité du code.

Cela est censé permettre au lecteur une compréhension global de l'application.

```

1  [HttpGet]
2      [Route("/HotelPasCher/AfficherOffreDisponible")]
3      public string AfficherOffreDisponible(string login, string password,
4      string dateArrive, string dateDepart, int nbPersonne)
5      {
6          Create();
7          ServiceHotel1Controller luncher = new ServiceHotel1Controller();
8          DateTime dt1 = DateTime.ParseExact(dateArrive, "dd/MM/yyyy", culture);
9          DateTime dt2 = DateTime.ParseExact(dateDepart, "dd/MM/yyyy", culture);
10         this.agenceChoisis = checkConnexion(login, password);
11         List<Offre> Listres = new List<Offre>();
12         String listRes = "";
13         if (this.agenceChoisis != null)
14         {
15             foreach(Offre i in listTempGUI)
16             {
17                 listRes = listRes + i.idOffre +
18                     "=" + i.numChambre.nbLits +
19                     "=" + i.numChambre.numChambre +
20                     "=" + i.prixTotalOffre +
21                     "=" + i.numChambre.imageURL +
22                     "=" + i.deb +
23                     "=" + i.fin +
24                     "$";
25             }
26             Listres = listTemp;
27         }
28         else
29         {
30             Console.WriteLine("Désoler votre identification a échoué ! ");
31         }
32     }
33     return listRes;
34 }
35

```

Afficher les offres (dans les services hôtel)

```

1  [HttpGet]
2      [Route("/HotelPasCher/Reservation")]
3      public string faireReservation(string login, string password, string idOffre,
4      string nomPersonne, string prenom, int numeroCB, int nbPersonne)
5      {
6
7          if(this.hotelPasCher == null)
8          {
9              Create();
10         }

```

```

11     checkConnexion(login, password);
12     Reservation resFinal = new Reservation();
13     String reserv = "";
14     foreach (Offre x in listTemp)
15     {
16
17
18         if (x.idOffre == idOffre)
19         {
20
21             Reservation res = new Reservation(nomPersonne, prenom, numeroCB, x.deb, x.fin,
22             nbPersonne, x.prixTotalOffre);
23             resFinal = res;
24             Client temp = new Client(nomPersonne, prenom, numeroCB);
25             agenceChoisis.ClientAgence.Add(temp);
26             //recherche la premiere chambre libre et fais la reservation
27             //si elle n'existe pas/la reservation n'a pas pu etre effectuer renvoie null
28             TypeChambre chambre = hotelPasCher.Reserver(res);
29             hotelPasCher.ListChambres.Add(chambre);
30             reserv = reserv + res.nomClient +
31                 "=" + res.prenomClient +
32                 "=" + res.prixTotal +
33                 "=" + res.numCarteBancaire;
34         }
35     }
36     return reserv;
37 }

```

Effectuer une réservation (dans les services hôtel)

```

1  [Route("/Trivago/Compareur")]
2  [HttpGet]
3  public string Get(string ville, string dateDebut, string dateFin,
4  int nombrePersonne, int nombreEtoile)
5  {
6      String url = "https://localhost:44350/";
7      string offresHotelPasCher = GetReleases(url +
8
9      "HotelPasCher/getOffreCompareur?ville=" +
10     ville +
11     "&dateDebut=" + dateDebut +
12     "&dateFin=" + dateFin +
13     "&nombrePersonne=" + nombrePersonne +
14     "&nombreEtoile=" + nombreEtoile);
15     string offreHotelPlusCher = GetReleases(url +
16
17     "HotelPlusCher/getOffreCompareur?ville=" +
18     ville +
19     "&dateDebut=" + dateDebut +
20     "&dateFin=" + dateFin +
21     "&nombrePersonne=" + nombrePersonne +
22     "&nombreEtoile=" + nombreEtoile);
23
24     return offresHotelPasCher + offreHotelPlusCher;
25 }

```

```
26  
27     }
```

Méthode qui rassemble les offres (Service comparateur)

```
1     public String GetReleases(String url)  
2     {  
3         var request = (HttpWebRequest)WebRequest.Create(url);  
4  
5         request.Method = "GET";  
6         request.AutomaticDecompression = DecompressionMethods.Deflate |  
7         DecompressionMethods.GZip;  
8  
9         var content = string.Empty;  
10        using (var response = (HttpWebResponse)request.GetResponse())  
11        {  
12            using (var stream = response.GetResponseStream())  
13            {  
14                using (var sr = new StreamReader(stream))  
15                {  
16                    content = sr.ReadToEnd();  
17                }  
18            }  
19        }  
20  
21        String resultat = content.ToString();  
22        return resultat.Replace("\\"", "");  
23    }
```

Méthode GetReleases() pour les requêtes

‘