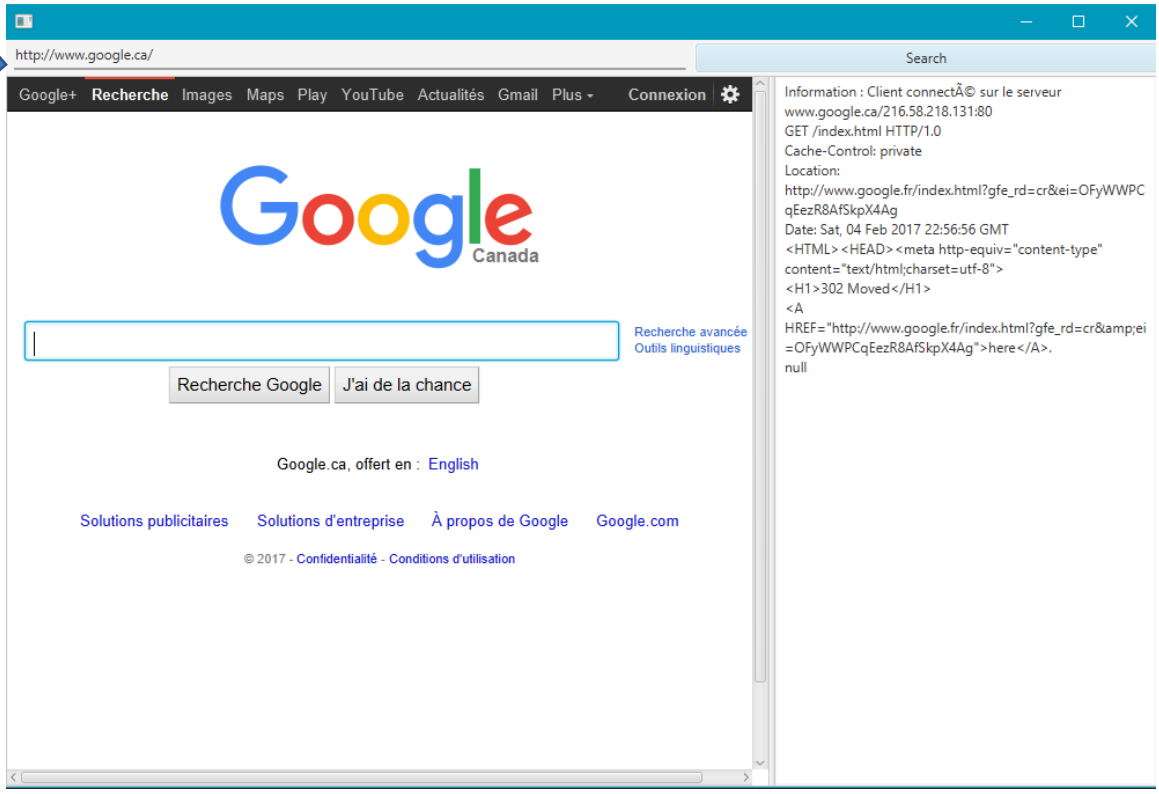
Golfier Maxime, Lathaud Quentin, Majeune Nicolas

SCÉNARIO DU TP1

Exercice 1

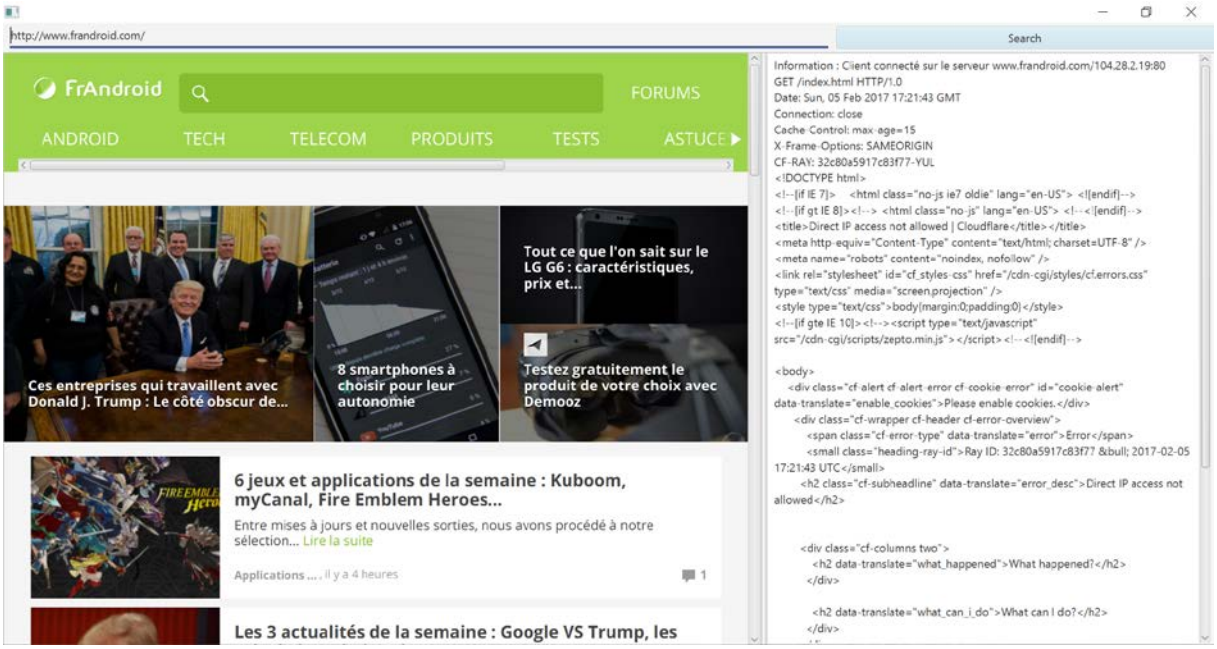
Zone de recherche : au lancemement, elle est initialisé sur google.ca . Nous pouvons faire des reherches manuellement de deux manières :
1 – en tapant le domaine ➔ ex : www.lemonde.fr ou lemonde.fr
2- en tapant la requette en entier ➔ http://www.lemonde.fr

Le bouton search est lié à la touche entrée du clavier, ainsi l'utilisateur naturellement va effectuer ses recherches



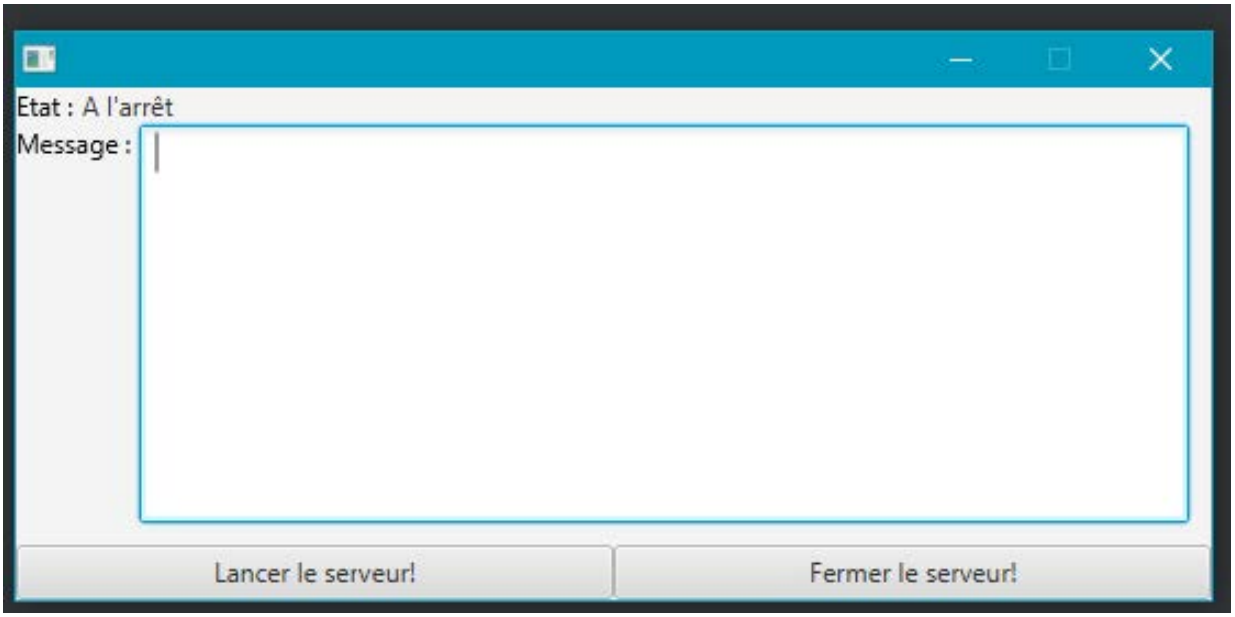
Ici, nous disposons d'une webview qui utilise grâce au framwork java (pattern observateur) d'obtenir en tant réel sur la search bar du nom du site

Dans le côté droit, nous avons les informations de la page index.html du site. Ici nous apprenons que la page de google a été moved (302). Nous avons réitéré l'expérience sur un autre site (frandroid.com) et nous obtenons la page HTML qui s'affiche sur cette même sortie standard le texte HTML reçu.

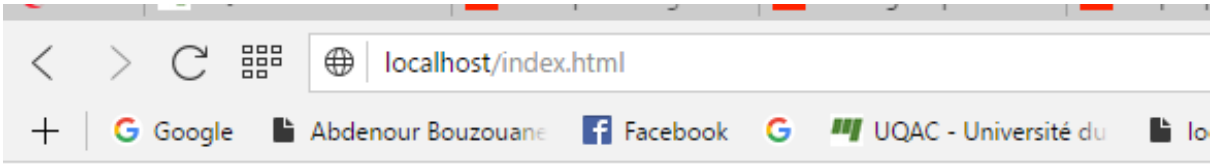


EXERCICE 2 :

Au lancement :

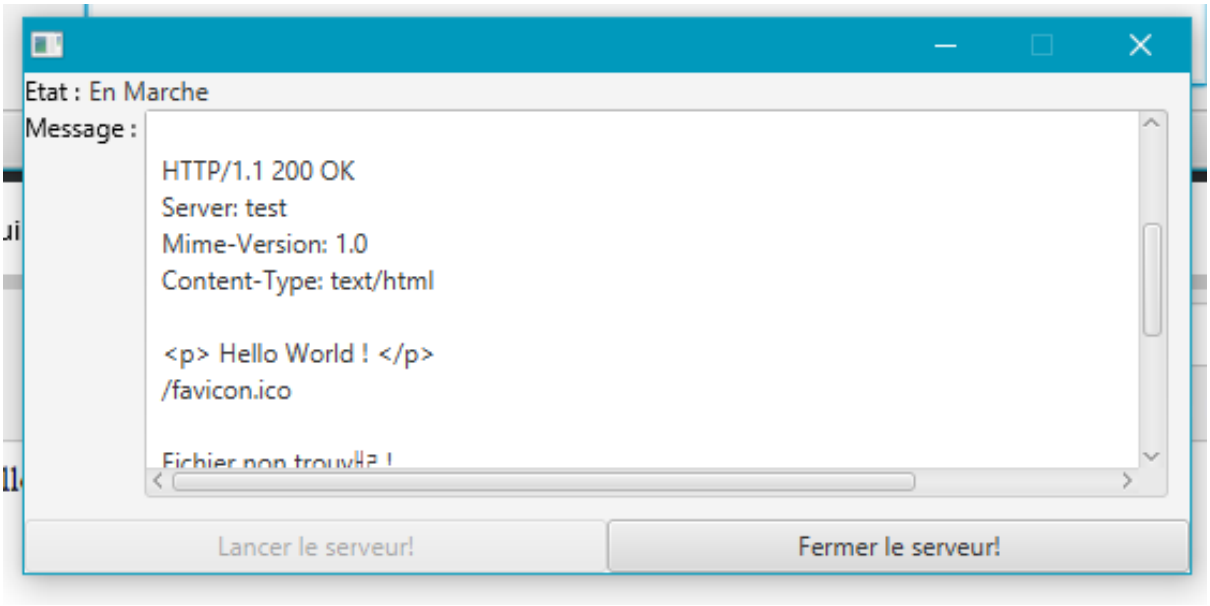


Ce qui s'affiche CÔTÉ CLIENT après avoir lancé le serveur et demandé une page sur le client : (Concrètement : après avoir attendu une connexion sur un port 80, notre programme est capable de recevoir des requêtes HTTP, et de renvoyer une réponse http

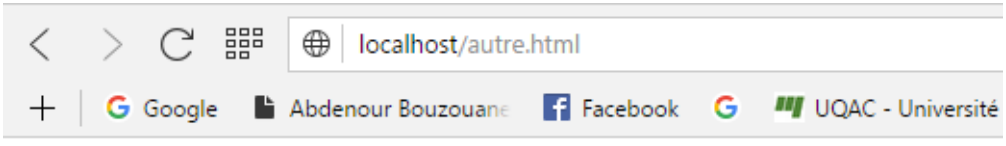


Hello World !

Ce qui s'affiche CÔTÉ SERVEUR après avoir lancé le serveur et demandé une page sur le client :



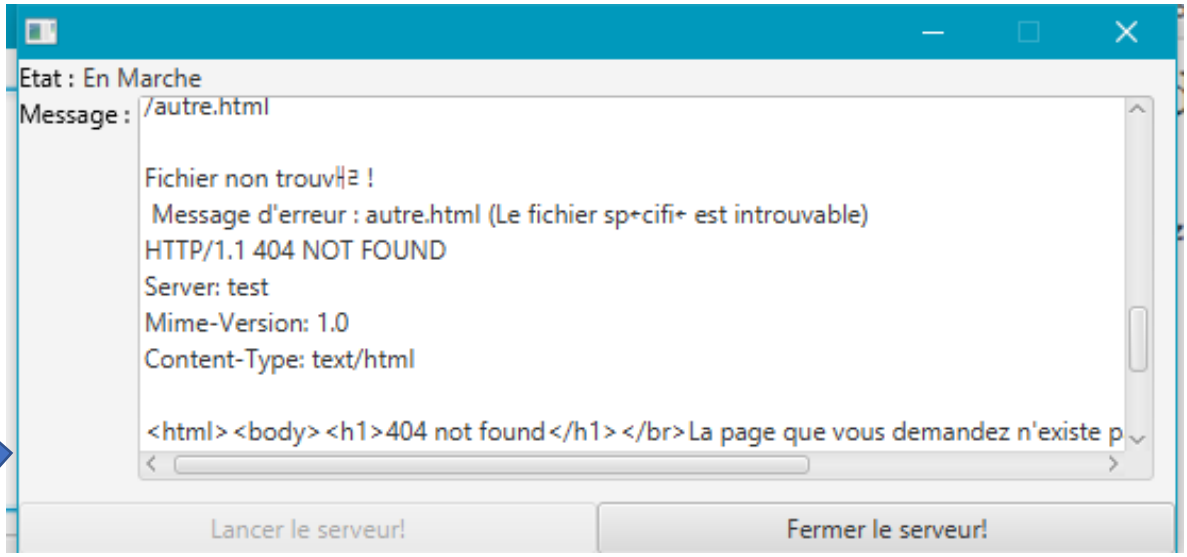
Ce qui s'affiche CÔTÉ CLIENT quand la ressource demandée n'a pas été trouvée/n'existe pas :



404 not found

La page que vous demandez n'existe pas !

Ce qui s'affiche CÔTÉ SERVEUR quand la ressource demandée n'a pas été trouvée/n'existe pas :



Nous avons ici une interface qui permet d'informer l'administrateur système de l'état du serveur et de ce qu'il a envoyé la dernière fois.
Concrètement sur le code : nous avons notre serveur qui marche en multithreading afin de desservir plusieurs clients simultanément !