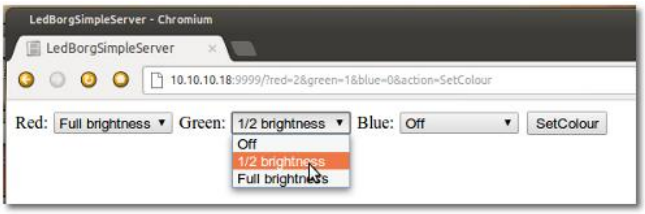
Le code manque de robustesse : nous ne vérifions pas la présence des paramètres rouge, vert et bleu du GET, ni ne validons leurs valeurs. Il n'y a pas de retour dans le HTML envoyé au client, pour dire si l'opération a réussi, quelle couleur a été définie, ou si la LedBorg n'est pas connectée.

Ces ajouts peuvent être un exercice pour le lecteur!




```
// LedBorgSimpleServer.vala

// Les espaces de nom que nous utiliserons
using GLib;
using Soup;

// Notre classe principale
public class LedBorgSimpleServer :
GLib.Object {
  // définit le numéro de port à écouter
  static const int LISTEN_PORT = 9999;

  // Définit le fichier correspondant à la carte
  static const string DEVICE = "/dev/ledborg";

  // la méthode utilisée au démarrage
  public static int main (string[] args)
{
     // crée une instance du serveur
     var server = new
Soup.Server(Soup.SERVER_PORT, LISTEN_PORT);

     // gère les requêtes du client
     server.add_handler("/",default_handler);

     // démarre le serveur
     server.run();

     return 0;
  }

  // gestionnaire http par défaut
  public static void
 default_handler(Soup.Server server,
 Soup.Message msg, string path,
 GLib.HashTable<string, string>? query,
 Soup.ClientContext client)
  {
     // traite une requête
     if(query != null)
     {
        // vérifie les paramètres
        if(query["action"] == "SetColour")
        {
           // récupère les valeurs RVB dans les
paramètres
           string red = query["red"];
           string green = query["green"];
           string blue = query["blue"];

           /* construire notre chaine couleur RVB
           chaque chiffre valant 0, 1 or 2:
           éteint, demi ou pleine brillance */
           string colour = red + green + blue;

           // modifier la couleur
           do_colour_change(colour);
        }
     }
     // construire la chaine html pour le client
     string html = """
   <html>
     <head>
        <title>LedBorg Serveur Simple</title>
     </head>
     <body>
        <form method="get" action="/">
         Rouge:<select name="red">
           <option value="0">Éteint</option>
           <option value="1">Faible</option>
           <option value="2">BrillantFull</option>
         </select>
         Vert:<select name="green">
           <option value="0">Éteint</option>
           <option value="1">Faible</option>
           <option value="2">BrillantFull</option>
         </select>
         Bleu:<select name="blue">
           <option value="0">Éteint</option>
           <option value="1">Faible</option>
           <option value="2">BrillantFull</option>
         </select>
         <input type="submit" name="action"
      value="SetColour" />
        </form>
     </body>
   </html>
       """;

       // renvoie le html au client
       msg.set_status_full(
         Soup.KnownStatusCode.OK, "OK");
       msg.set_response("text/html",
         Soup.MemoryUse.COPY, html.data);
 }

 // changer la couleur
 public static void
   do_colour_change(string colour)
 {
    /* Ici nous utilisons la gestion de fichier
POSIX
    pour écrire dans le fichier, plutôt que la
gestion
    GIO de vala, puisque nous n'avons pas besoin
de la
    sécurité de GIO pour accéder à /dev */
    // ouvre le fichier pour écrire
    Posix.FILE f = Posix.FILE.open(DEVICE, "w");

    // écrit la chaine de couleur dans le fichier
    f.puts(colour);
 }
}
```

Article de Ross Taylor