**COMPTE RENDU TP1   
EXERCICE 1 - EXERCICE 2 - EXERCICE 3**

**Réalisé par : Imane Dahbi**

**Groupe : 4IIR – G8**

**EMSI Les Orangers**

**Exercice 1 : (Première servlet)**

**L’exécution**

la méthode doGet()

Une image contenant texte, capture d’écran, Police, logiciel

Le contenu généré par l’IA peut être incorrect.

Modifiez l’annotation adéquate de sorte que votre servlet s’exécute lors d’une demande d’accès à la ressource nommée bonjour.

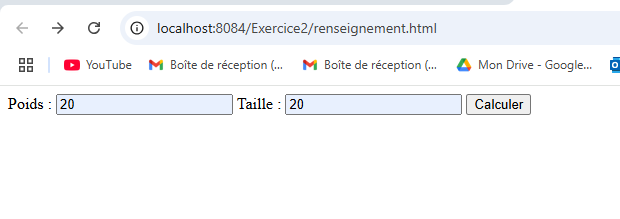
Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

**Exercice 2 : (Récupérer des données transmises par le client)**

**L’exécution avec la methode get**

Créez une servlet nommée CalculDeMonImc qui affichera l’indice de masse corporelle sur l’écran du navigateur lorsque l’url suivante est invoquée http://VotreIP:LeBonPort/CalculDeMonImc?poids=94&taille=1.86

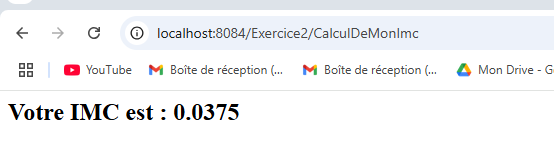


Une image contenant texte, capture d’écran, Police, ligne

Le contenu généré par l’IA peut être incorrect.

**L’exécution avec la methode post**

**Une image contenant texte, capture d’écran, Police, ligne

Le contenu généré par l’IA peut être incorrect.**

**5. Différences entre HTTP GET et HTTP POST**

* **GET** :
  + Les données sont envoyées dans l'URL, visibles dans la barre d'adresse du navigateur.
  + Limité en taille, car la longueur de l'URL est restreinte.
  + Utilisé pour des requêtes qui ne modifient pas les ressources (lecture).
* **POST** :
  + Les données sont envoyées dans le corps de la requête HTTP, invisibles pour l'utilisateur.
  + Pas de limite stricte de taille, peut envoyer de grandes quantités de données.
  + Utilisé pour des actions qui modifient l'état du serveur (création, mise à jour, etc.).

**La méthode GET est adaptée pour récupérer des informations sans modification, tandis que la méthode POST est plus sécurisée et permet d'envoyer des informations sensibles ou volumineuses.**

**Exercice 3 : (Une classe “métier” dans notre application web)**

**Classe java nommée Imc.**

**Une image contenant texte, capture d’écran, affichage, logiciel

Le contenu généré par l’IA peut être incorrect.**

**Une image contenant texte, capture d’écran, ligne, Police

Le contenu généré par l’IA peut être incorrect.**

**Une image contenant texte, capture d’écran, Police, ligne

Le contenu généré par l’IA peut être incorrect.**

**(a) Où avez-vous déclaré votre objet ?**

L'objet de la classe Imc a été déclaré dans la méthode doPost de la servlet, à l'intérieur de cette méthode :

Imc imc = new Imc(taille, poids);

**(b) Dans quelle méthode l’avez-vous construit ?**

L'objet de la classe Imc a été construit dans la méthode doPost de la servlet, après avoir validé les entrées poids et taille. Voici la ligne de code où l'objet est créé :

Imc imc = new Imc(taille, poids);

**(c) Quelle est la portée de cet objet ?**

La portée de l'objet imc est **locale à la méthode doPost**. Cela signifie qu'il est créé et utilisé uniquement dans cette méthode, et n'est pas accessible en dehors de celle-ci. Une fois la méthode doPost terminée, l'objet imc est détruit (puisque sa portée est limitée à la méthode).

En résumé :

* **Déclaration** : dans doPost
* **Construction** : dans doPost après la validation des entrées
* **Portée** : locale à la méthode doPost