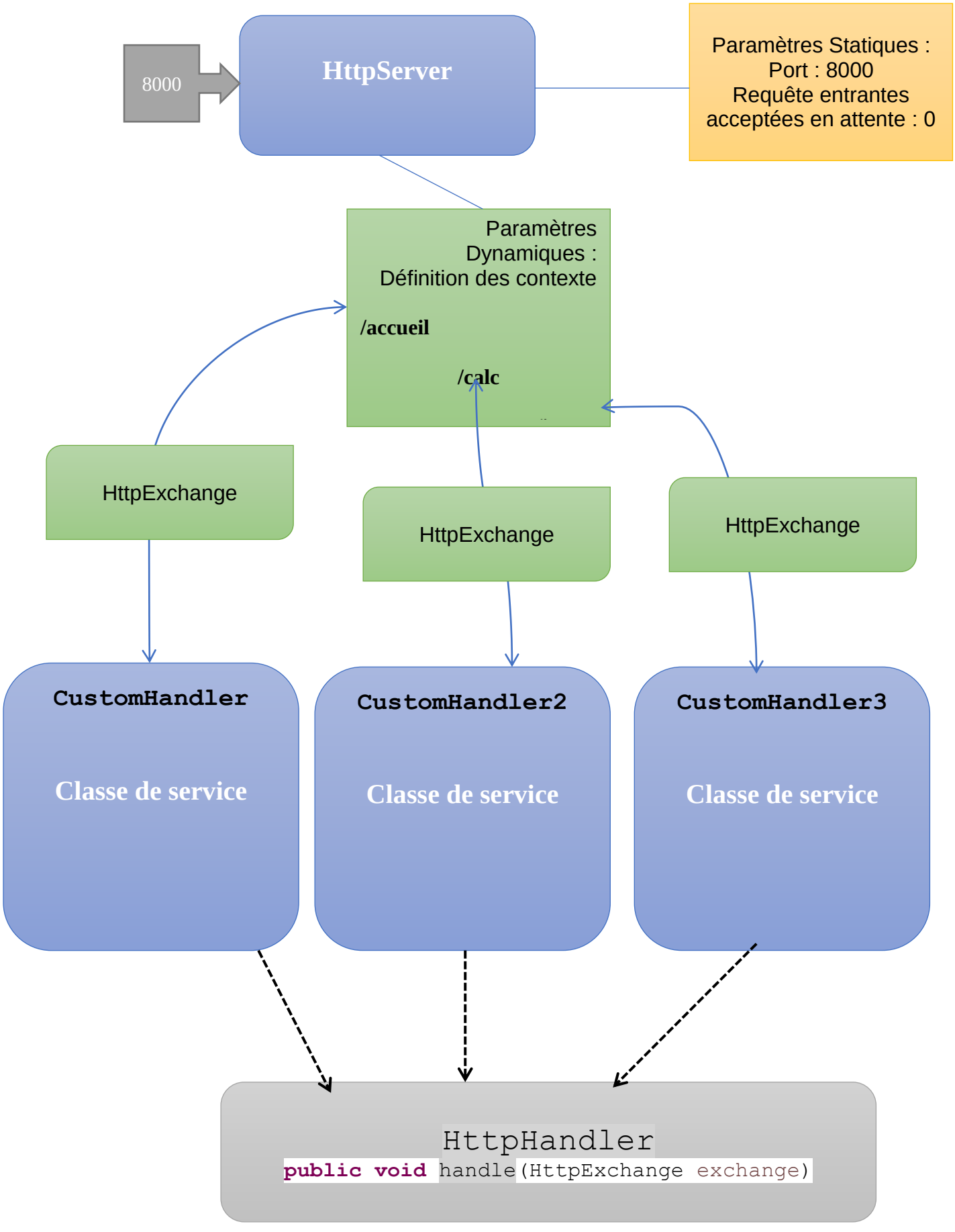
Simple Serveur http en Javva : La classe HttpServer

Vue globale :



I- Paramétrage du Serveur :

```
HttpServer server = HttpServer.create(new InetSocketAddress(8000), 0);

server.createContext("/accueil", new CustomHandler());
server.createContext("/calc", new CustomHandler2());
server.createContext("/img", new CustomHandler3());

   server.start();
```

1) Serveur paramétré pour répondre sur le port 80 (localhost)
2) Trois contexts définis
3) 3 classes implémentant l'interface HttpHandler pour répondre sur chaque contexte

II- Flow d'exécution :

a. Requête entrante du client http://localhost:8000/accueil
b. HttpServer accept la request car il a bien un contexte défini
c. HttpServer crée un objet HttpExchange avec les infomations provenant du client :
    i. Date
    ii. Navigateur
    iii. Type de request
    iv. Paramètres de la request
    v. ….
d. HttpServeur rechecher l'implémentation du Handler associé au contexte de request : /accueil ⇔ CustomHandler
e. HttpServeur appelle la méthode handle de l'objet CustomHandler en lui fournissant l'objet exchange en paramètre
f. Le développeur peut implémenter la logique « métier » dans cette méthode pour répondre au besoins du client.
g. Il utilisera l'objet exchange pour réponse au client via HttpServer

III- Code :

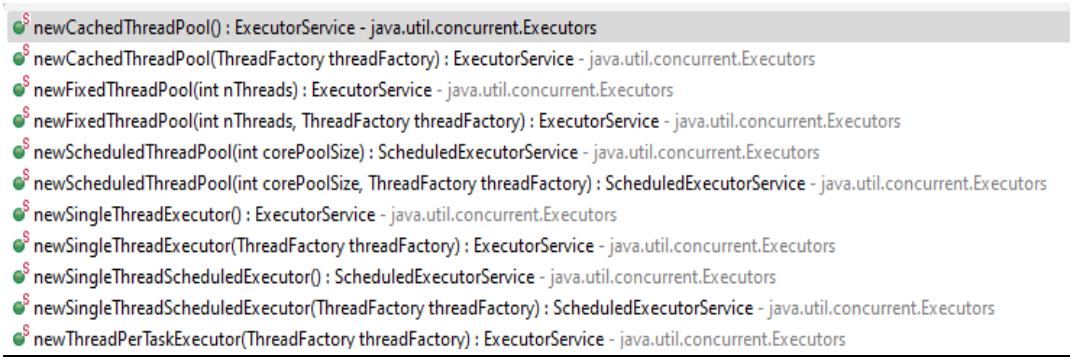
Etudiez le code des applications clientserveur4 et clientserveur42 qui met en place ce petit framework.
Voyez comment automatiser les tâches répétitives et définir un petit système de template.

IV- Extentions & Limitation

V-

Ce petit serveur, en l'état ne gère pas le muttithreading que l'on a mis en place dans le tp consacré au servlet mais, on peut changer cela en selectionnant l'implémentation de son système de gestion.
`server.setExecutor(Executors….(à découvrir))`
java propose des implémentationsdifférentes en fonction des besoins.

Exemples d'Executors :



On voit ici une implémentation avec un système de cache, une autre avec gestion de Threads ect ect….

Malgré cela, et bien qu'il puisse couvrir beaucoup de besoins, il reste un outil de bas niveau sur lequel beaucoup de services d'entreprises manquent.

- Gestion de la sécurité
- Gestion des sessions utilisateurs
- Système de gestion de connexions à des systèmes externes
- Système de monitoring des applications
- ….

Les serveurs d'applications, sont l'extension et l'adaptation de ces solutions techniques vers le monde des entreprises.
Ce sont des solutions logicielles embarquant un grand nombre de librairies et autres outils pour simplifier la vie du développeur et réduire la redondance de code.
Cet ensemble d'outils ainsi que la manière « imposée » de développer dans un cadre (comme ici, la définition des contextes et des handlers par exemple) définissent un cadre de travail normé. D'où l'appellation Framework.

Prochaine étape, le serveur d'application Tomcat, où ces concepts vont se retrouver et être bien entendu étendus.