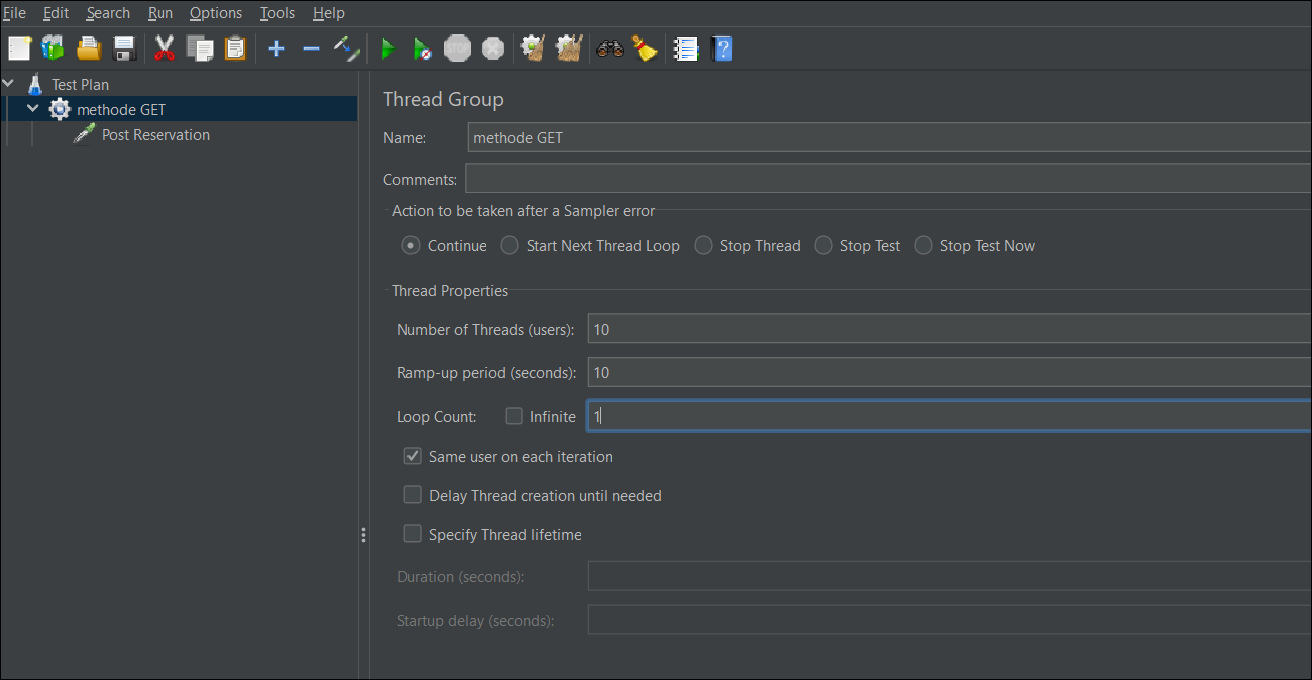
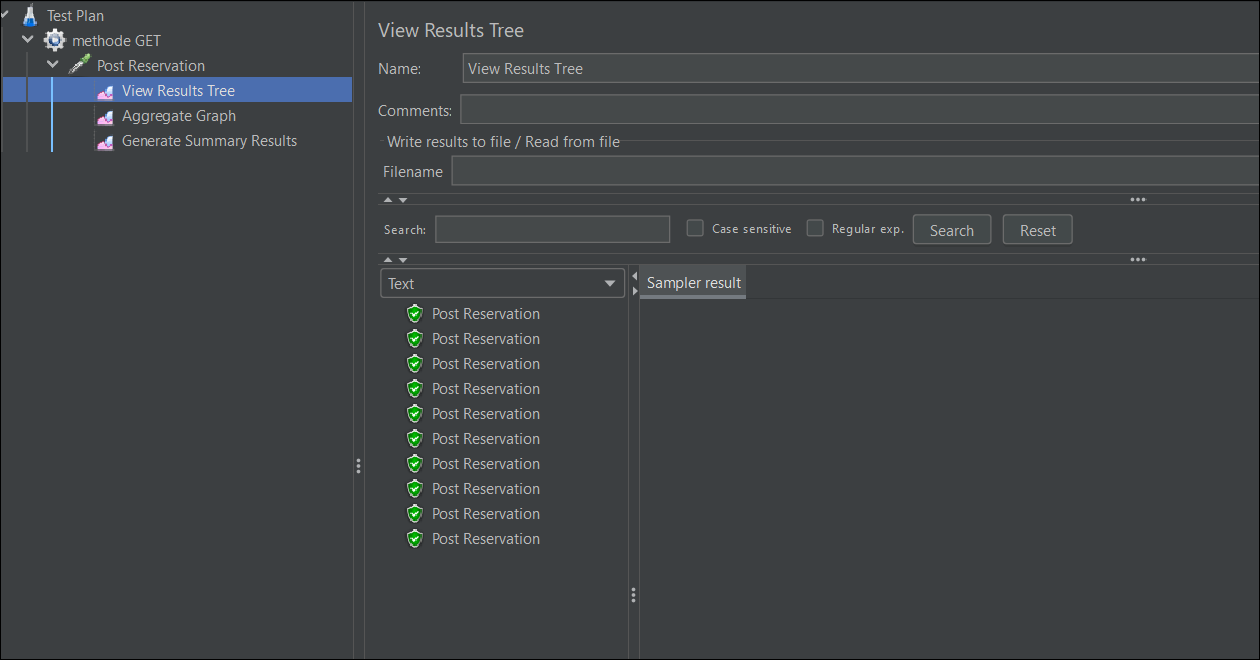
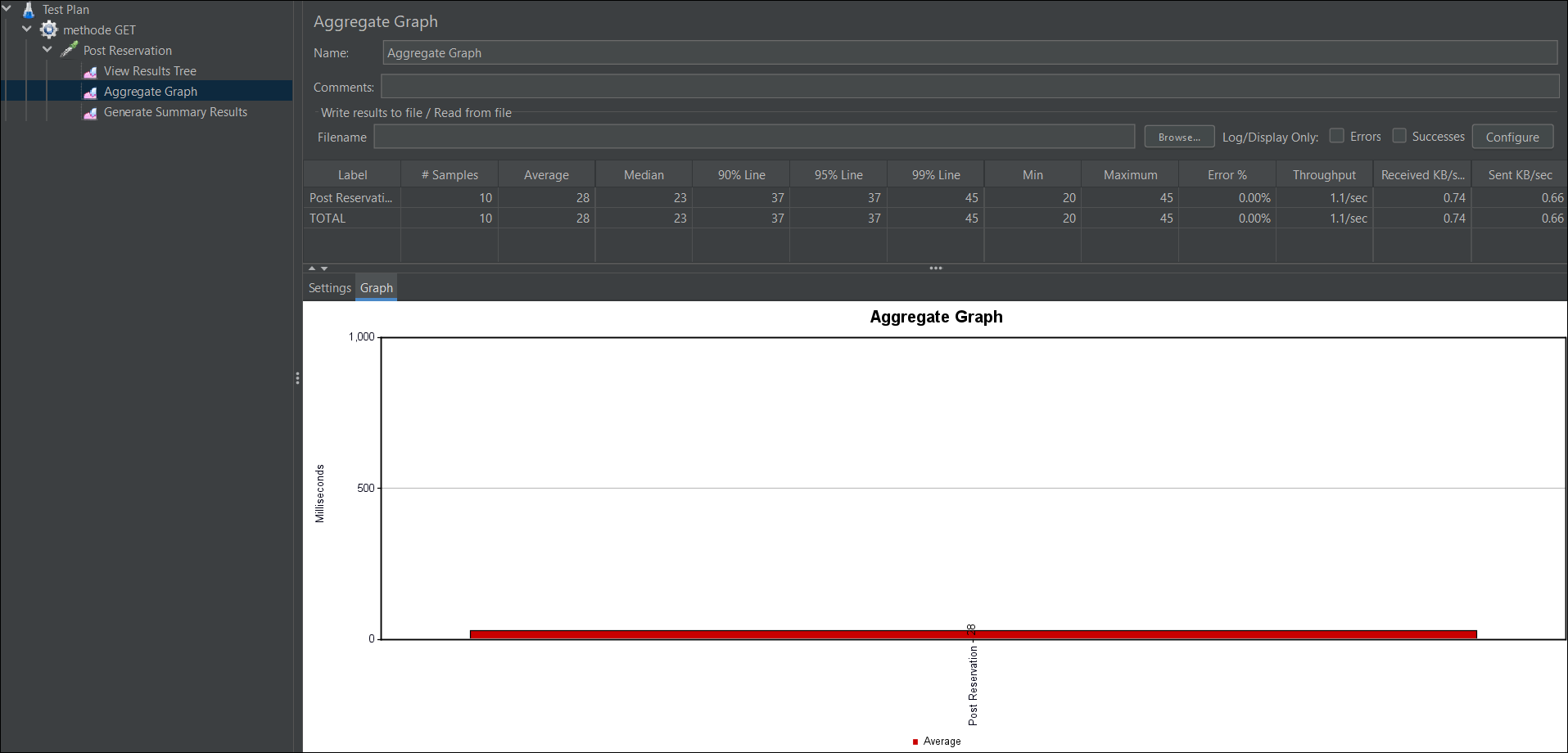
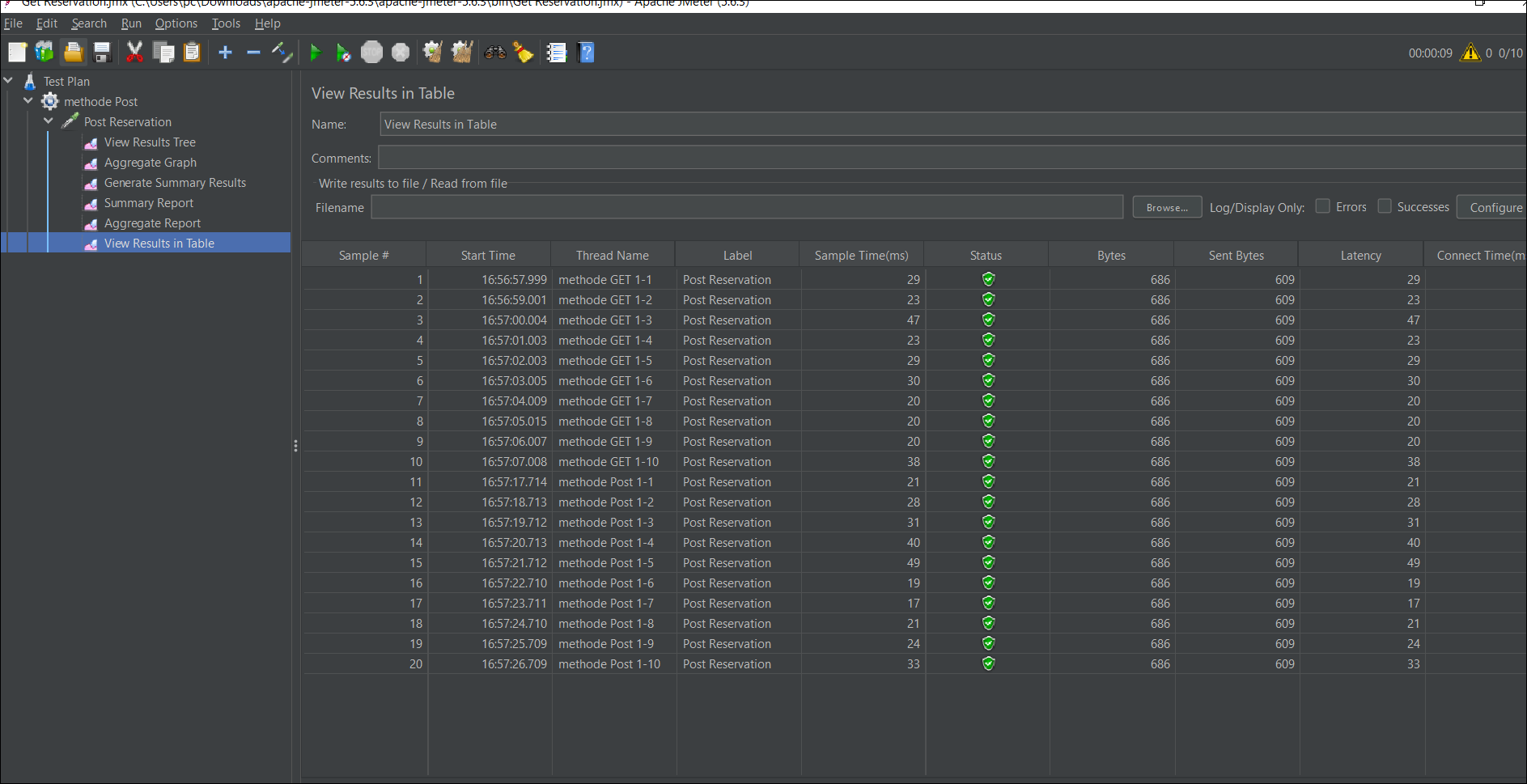
|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Étude de Cas : Système de Gestion des Réservations d’hôtel    ***Réalisé par*** :  Mlle. Latifa didi alaoui  Mlle. Majida  Mlle. Hiba  ***Source Code*** :    https://github.com/latifadidialaoui/Syst-me\_de\_Gestion\_des\_R-servations\_d-H-tels.git | | | |
|  | | | |
| ***Étape 1: Préparation de l'environnement***   1. **Configuration du service SOAP:**  * Création un service SOAP avec **JAX-WS** et **Spring Boot**      1. **Configuration du SoapUI pour les tests fonctionnels :**  * Importation du fichier WSDL dans SoapUI      * Test de chaque opération   Créer une reservation  Consulter une reservation  Modifier une réservation  Supprimer une réservation | | | |
|  | | | |

***Étape2 : Test des performances***

1. ***Test de latence et débit :***

* **Outil: Apache JMeter**



1. **Analyse de l'utilisation des ressources :**

* **Outil :** VisualVM ou Prometheus
* Mesurez l'utilisation du CPU, de la mémoire et des threads pendant les tests de charge.

**Étape 3 : Évaluation de la scalabilité**

1. **Simuler une charge croissante :**
   * **Outil :** Apache JMeter ou Gatling.
   * Étapes :
     + Augmentez progressivement le nombre d'utilisateurs (par exemple : de 10 à 500).
     + Mesurez les temps de réponse et vérifiez les erreurs.
2. **Tester avec un répartiteur de charge :**
   * Utilisez **NGINX** ou **HAProxy** pour équilibrer la charge entre plusieurs instances.
   * Exemple de configuration NGINX :

**Étape 4 : Évaluation de la simplicité d'implémentation**

1. **Documenter l’implémentation SOAP :**
   * Utilisez **WSDL** pour fournir la définition des services.
   * Utilisez des outils comme SoapUI pour générer des clients SOAP.
2. **Évaluer l'effort d'intégration client :**
   * Utilisez des outils tels que **wsimport** pour générer des classes clientes.
   * Commande :

**Étape 5 : Évaluation de la sécurité**

1. **Configurer la sécurité WS-Security :**
   * Ajoutez une couche de sécurité SOAP avec **username/password token** ou **chiffrement des messages**.
   * Exemple de configuration avec Spring Boot et CXF :
2. **Tester les vulnérabilités :**
   * Utilisez des outils comme **OWASP ZAP** ou **Burp Suite** pour tester les failles dans votre service SOAP.

**Étape 6 : Évaluation de la flexibilité**

1. **Vérifiez l'interopérabilité :**
   * Testez le service SOAP avec des clients écrits dans différents langages (Java, Python, C#).
   * Exemple en Python (avec zeep) :
2. **Ajouter des extensions :**
   * Testez des fonctionnalités avancées comme la gestion des erreurs SOAP (SOAP Faults) ou l’ajout de nouveaux services sans casser les anciens clients.

**Étape 7 : Documentation et recommandations**

1. **Analyse des résultats :**
   * Comparez les métriques obtenues (latence, débit, consommation de ressources, scalabilité) avec les besoins du projet.
   * Documentez les observations dans un rapport structuré.
2. **Proposer des recommandations :**
   * Précisez les cas d’usage idéaux pour SOAP (interopérabilité, sécurité forte, transactions complexes).
   * Fournissez des suggestions pour améliorer les performances ou la sécurité, si nécessaire.

***Tableaux Résultats:***

*1. Performances : Temps de Réponse (Latence)*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Taille du Message (KB)** | **Opération** | **REST (ms)** | **SOAP (ms)** | **GraphQL (ms)** | **GRPC (ms)** |
| **1KB** | Créer |  | **15** |  |  |
|  | Consulter |  | **15** |  |  |
|  | Modifier |  | **22** |  |  |
|  | Supprimer |  | **15** |  |  |
| **10KB** | Créer |  | **29** |  |  |
|  | Consulter |  | **15** |  |  |
|  | Modifier |  | **30** |  |  |
|  | Supprimer |  | **15** |  |  |
| **100KB** | Créer |  | **40** |  |  |
|  | Consulter |  | **15** |  |  |
|  | Modifier |  | **50** |  |  |
|  | Supprimer |  | **15** |  |  |

2. Performances : Débit (Throughput)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre de RequêtesSimultanées | *SOAP* |  |  |  |
| *10* | *9.1/min* |  |  |  |
| *100* | *10.3/min* |  |  |  |
| *500* | *45.1/min* |  |  |  |
| *1000* | *56.2/min* |  |  |  |

5. Sécurité

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Critère de sécurité** | *REST* | *SOAP* | *graphQL* | *grpc* |
| Support TLS/SSL | *oui* | *oui* | *oui* | *oui* |
| Gestion de l'authentification (OAuth2, JWT) | *oui* | *Non* | *oui* | *oui* |
| Résistance aux attaques (SQL Injection, CSRF) | *oui* | *oui* | *oui* | *oui* |