

Rapport E5

1. Présentation de l'application :

Application de surveillance du trafic routier de la métropole rennaise basée sur flask.

2. Présentation du dispositif de monitoring :

Afin de surveiller l'état de santé de l'application , deux éléments sont mis en place :

- Monitoring du serveur d'application
- Journalisation

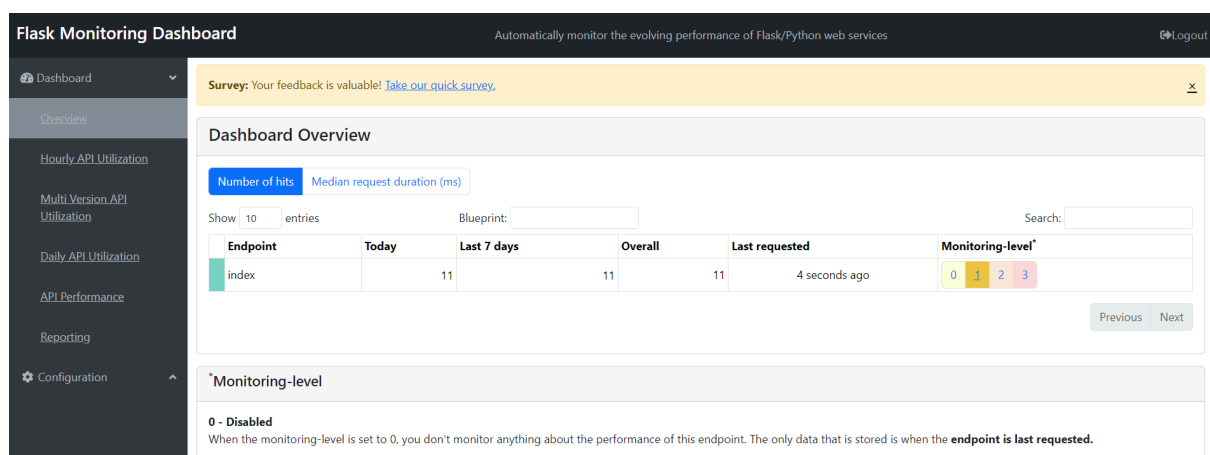
1. Monitoring

La solution de monitoring choisie est flask-monitoring.

Pour des raisons de :

- simplicité de développement
- "légèreté" de l'application pas une contrainte on peut donc la rendre plus monolithique en intégrant le monitoring à celle-ci
- seuils d'alertes prédéfinis

Cette solution (à la manière du panel admin de Django) ajoute [une route à la localisation /dashboard](#) , sécurisable en [username/password](#), qui va venir collecter les métriques voulues et les stocker dans une [base de données type "SQLite"](#).



- lien documentation :
<https://flask-monitoringdashboard.readthedocs.io/en/latest/index.html>
- lien répo :
<https://github.com/flask-dashboard/Flask-MonitoringDashboard>

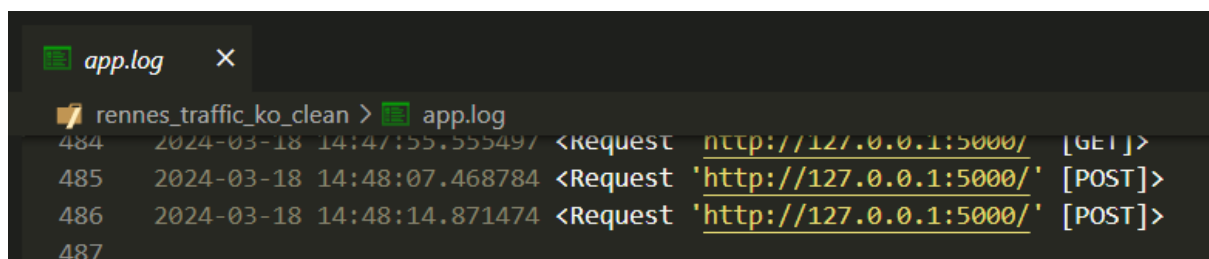
2. Journalisation

Pour celle-ci , les événements sont **stockés et horodatés** dans un fichier.
Cette fonctionnalité est directement intégrée au code de l'application.

On choisit cette solution parce que:

- simplicité d'implémentation
- alourdissement de l'application qui n'est pas une contrainte
- accès facile au conteneur pour récupérer ledit fichier

dépendances (librairies) : Flask , Logging



```

app.log
rennes_traffic_ko_clean > app.log
484 2024-03-18 14:47:55.555497 <Request http://127.0.0.1:5000/ [GET]>
485 2024-03-18 14:48:07.468784 <Request 'http://127.0.0.1:5000/' [POST]>
486 2024-03-18 14:48:14.871474 <Request 'http://127.0.0.1:5000/' [POST]>
487

```

exemple du fichier

3. Présentation de l'incident technique :

Mise en situation:

Vous avez à votre charge , un nouvel alternant fraîchement arrivé qui ne comprend pas pourquoi son application bug. Votre devoir en tant que tuteur est donc de retrouver et corriger ses erreurs.

Erreur observée :

Remontée utilisateur :

La carte passée ne s'affiche pas



Sur le monitoring : Aucune erreur remontée

Log associé :

```
10 2024-03-18 16:16:55.509140 <Request http://127.0.0.1:5000/ [GET]>
11 2024-03-18 16:17:47.251695 <Request 'http://127.0.0.1:5000/' [GET]>
```

Aucune erreur

4. Diagnostic :

L'erreur ne semble donc pas venir du serveur d'application, ni du code mais de l'affichage de celle-ci.

Il faut donc aller vérifier le code HTML

5. Méthodologie de la résolution :

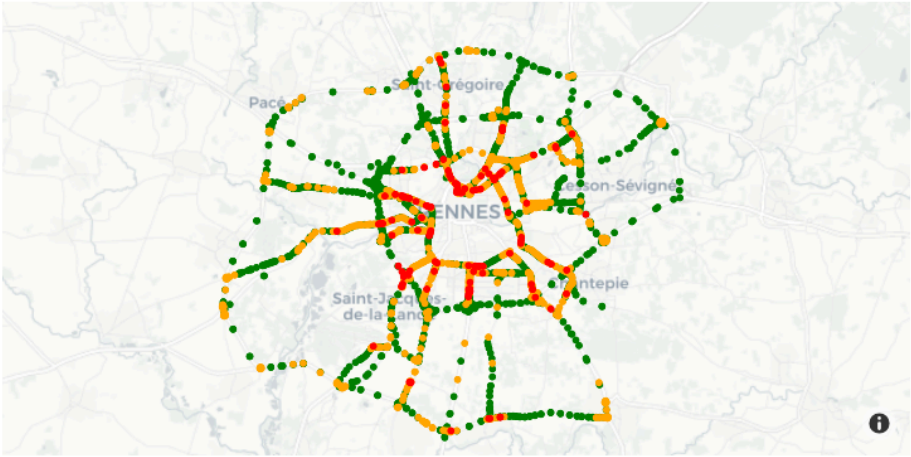
Solution : Une balise HTML non fermée

```
34 <script>
35     var graphJSON = {{ graph_json | safe }};
36     Plotly.newPlot('scatter-plot', graphJSON.data, graphJSON.layout);
37 <script>
38 </body>
```

Vérification :

Traffic Rennes

Traffic en temps réel



- traffic
- freeFlow
 - heavy
 - congested