

# Interface 2037 Ready for inquiry

## Description du service

---

Interface 2037 est un service Web permettant à un usager de soumettre des questions à un système expert afin d'obtenir une réponse.

Les composants du système :

- **Le serveur de questions** : il mémorise les questions posées par les usagers et les réponses fournies par les systèmes experts.
- **Le client usager** : il permet à un individu de poser une question et de consulter sa réponse une fois qu'elle a été traitée par un système expert.
- **Le client système expert** : il consulte les questions une à une et fournit une réponse.

Vous devez implémenter les composants suivants du service : le serveur de questions, un client simple pour les usagers et le client système expert. On considérera que le système expert sera **simulé** par un programme permettant d'afficher la question en cours sur un terminal et de saisir la réponse à la question.

Les cas d'utilisation du service sont décrits dans les scénarios ci-dessous :

### Un usager pose une question

#### **Scénario :**

Quand l'utilisateur pose une question au serveur

Alors le serveur indique qu'il a enregistré la question

Et il permet à l'utilisateur de localiser la réponse lorsqu'elle sera disponible

### Un système expert consulte la prochaine question

#### **Scénario :**

Étant donné qu'il existe une question en attente de réponse

Quand le système expert demande la prochaine question au serveur

Alors il récupère la question en attente

Et la question suivante devient la question en attente

#### **Scénario :**

Étant donné qu'il n'existe aucune question

Quand le système expert demande la prochaine question au serveur

Alors le serveur indique qu'il n'existe pas de question

Et le système expert se met en veille avant de redemander une question

## Un système expert répond à la question

### **Scénario :**

Étant donné que le système expert a récupéré une question en attente  
Et qu'il a trouvé une réponse  
Quand il fournit la réponse au serveur  
Alors le serveur indique qu'il a enregistré la réponse à la question

## Un usager consulte la réponse à sa question

### **Scénario :**

Étant donné qu'un usager a posé une question  
Et aucun système expert n'a pas encore traité la question  
Quand l'usager demande à consulter la réponse  
Alors le serveur indique que la réponse n'est pas encore disponible

### **Scénario :**

Étant donné qu'un usager a posé une question  
Et un système expert a traité la question  
Quand l'usager demande à consulter la réponse  
Alors le serveur affiche la réponse du système expert

## Un système expert ne peut pas répondre à la question

### **Scénario :**

Étant donné que le système expert a récupéré une question en attente  
Et qu'il ne peut pas fournir la réponse  
Quand il notifie le serveur de son échec  
Alors le serveur enregistre que cette question n'a pas de réponse connue.

## Extension à N usagers

---

Afin d'accepter un grand nombre de clients usagers, le service doit pouvoir gérer efficacement la mise en cache des questions et des réponses. Pour cela, il est recommandé d'utiliser un reverse proxy HTTP. Pour Interface 2037, le serveur Squid (<http://www.squid-cache.org/>) sera utilisé.

Vous devez fournir le fichier de configuration de Squid ainsi qu'une description de la solution d'implémentation choisie dans la réalisation du serveur de questions pour gérer la mise en cache.

## Extension à N systèmes experts

---

Le service doit accepter plusieurs clients systèmes experts en parallèle. Cependant, le service doit garantir qu'**une réponse ne sera fournie que par un seul client système expert**.

Vous devez fournir une description de la solution d'implémentation choisie pour assurer cette contrainte d'intégrité dans le traitement des questions par le service.

## Technologies

---

Les technologies conseillées pour la réalisation de ce projet sont :

Node.js et Express (<http://expressjs.com/>) : Node.js permet de développer des serveurs en Javascript et Express est un framework pour le développement d'application Web.

Java EE et JAX-RS : JAX-RS est l'API de Java EE pour le développement de client et de serveur de services Web RESTful. (<https://wikis.oracle.com/display/Jersey/Overview+of+JAX-RS+1.0+Features>).

Python + Flask (<http://flask.pocoo.org/>) : Flask est un framework pour le développement d'applications Web en Python. Flask incorpore un serveur HTTP de test pour exécuter une application Web.

Choisissez la technologie qui vous convient le mieux pour réaliser le projet. Vous pouvez également utiliser différentes technologies selon les parties du système réalisées (par exemple Node.js et Express pour le serveur de questions et JAX-RS pour le client de système expert).

Pour la mise en place du reverse proxy, le serveur de cache Squid (<http://www.squid-cache.org/>) sera utilisé.

## Notation

---

- 2,5 points pour la documentation du service. La documentation ne sera pas notée « au poids ». Un service bien conçu aura normalement besoin de peu de documentation.
- 2,5 points pour la réalisation du service. Seront pris en compte la lisibilité générale du code ainsi que la simplicité de lancement du service dans un environnement de développement.
- 5 points pour les tests automatisés d'acceptation. Le fonctionnement du service sera indirectement noté à travers l'automatisation des tests d'acceptation.
- 5 points pour la prise en compte des extensions (N usagers et N systèmes experts).
- 5 points pour la soutenance orale