|  |  |
| --- | --- |
| **420-W47-SF DSED**  **Pierre-François Léon**  **Automne 2023** | **Examen final**  **3h00** |

Modalité de remise et consignes

La remise consiste en deux artéfacts : ce fichier Word avec vos réponses et la solution Visual Studio. Les deux artéfacts doivent être remis dans une archive au format ZIP par Omnivox **avant la fin du cours**.

**Si vous préférez répondre aux questions de cours de manière manuscrite**, vous pouvez me fournir des **photos lisibles** de vos réponses. Ces photos doivent être jointes dans votre archive sur Omnivox.

**Gestion du temps** : si vous n’avez pas la réponse à une question ou exercice, passez au suivant et revenez sur les points non répondus à la fin. Je vous conseille de ne pas chercher à tester l’exécution de vos méthodes avant d’avoir terminé l’ensemble des questions car cela peut vous prendre du temps à déboguer.

Les deux exercices sont indépendants et peuvent donc être traités dans n’importe quel ordre.

Vous avez le droit à seulement vos notes de cours et vos codes personnels. À l’exclusion des ressources du Cégep**, vous n’avez pas le droit d’accéder à internet**. Vous n’avez pas non plus le droit de communiquer avec les autres étudiants ou une aide externe.

Tout plagiat du cours, d’une correction, d’une autre personne, d’internet, etc. sera traité en respect de la PEA.

|  |  |
| --- | --- |
| **Prénom, Nom :** | Zachary Gagnon |
| **Numéro de matricule :** | 1736814 |

Exercice 1 - Questions (10 points) ~ 40 mins

1. Que sont les « topics » ? À quoi servent-t-ils ? Dans quel(s) cas les utiliser ? (3 pts)

|  |
| --- |
| **Réponse :**  Les topics sont une des façons de séparer ou de filtre les messages. Ils sont utilisés pour publie des messages de façon uniforme pour que le consommateur puisse traiter le bon type de message. Les sujets sont très utiles lorsqu’on a plusieurs consommateurs et qu’on veut qu’il traite diffèrent type de message. Les sujet sont des liste de mot sépare par des point comme vitesse.couleur.espèce donc un topic ressemblerais a lengt.jaune.tortue donc un consommateur pourrait que prendre tout type de vitesse jaune et tous les espèce de cette façon \*.jaune.\*. |

1. Comment choisir entre un échange synchrone ou asynchrone ? (2 pts)

|  |
| --- |
| **Réponse :**  Pour choisir il faut tout d’abord savoir es qu’ont à besoin que le traitement soit complété avant de faire autre chose ou non. Il faut aussi prendre en considération le théorème de CAP qui dit qu’il est impossible de garantir la cohérence, disponibilité et la tolérance au partitionnement. Donc il faut savoir qu’elles sont la priorité puisque si ont choisie synchrone ont choisi la cohérence donc ont ne peut pas être disponible et tolérant au partitionnement. De plus, si on a une préférence de méthode ont peut choisir avec ça puisque si on veut absolument du pub/sub ça fait qu’ont pas le choix d’être asynchrone. |

1. Expliquez les termes suivants dans le contexte d’échanges de données : « chorégraphie » et « orchestration ». Quels sont les avantages/inconvénients de chaque approche ? Quelle approche est la meilleure ? (3 pts)

|  |
| --- |
| **Réponse :**  Les deux termes font référence aux types de collaboration qu’ont retrouvé dans les systèmes d’échange de donne lorsqu’on a plusieurs services. La chorégraphie est un système qui a plusieurs services qui ne sont pas dépendent à un point central et qui envoi des évènements et peuvent s’abonner à n’importe quel évènement envoyer par un autre service. De plus puisqu’ils sont indépendants il n’a pas de point de défaillance unique, il s’intègre bien dans un mode de développement agile et la mise a l’échelle est facile. Par contre, il utilise un mode d’intégration asynchrone, les processus d’affaire ne sont pas visibles et est difficile a debugger. Tandis que le mode orchestration est un type ou les services sont tous dépendent d’un point central ou il publie leur message et s’abonne. Donc il est efficace en mode synchrone, facile a maintenir, processus d’affaire unifier et explicites et peut effectuer des traitement. Par contre, les services sont couple, il y a un point de défaillance unique et il est difficile de mettre a jour les service. Comme toujours, qu’elle est le meilleur à utiliser dépend des besoins de la personne comme, s’il veut absolument du synchrone ou sa lui est égal ou s’il veut quelque chose de facile a debugger et a upgrader ou non. Donc cela dépend de la personne et du contexte |

1. Imaginez que vous deviez défendre l’utilisation de SignalR dans un projet. Comment allez-vous convaincre votre équipe de l’utiliser ? (2 pts)

|  |
| --- |
| **Réponse :**  SignalR est un très bon outil pour gère les connexion et le fait de façon automatique il permet aussi de faire des appel RPC en le client et le serveur dans les deux sens. Il peut aussi effecteur le même appel RPC sur tous les clients ou il peut cibler un client en particulier ou un groupe de client. C’est aussi une solution dimensionnable et est facile à installer puisqu’il est déjà intégrer dans .Net 6 et plus |

Exercice 2 – Gestion des entrées de journaux applicatifs (25 points) ~ 2h

Votre équipe est mandatée pour créer un système maison de gestion de la journalisation des applications de l’entreprise : chaque application de l’entreprise doit pouvoir émettre des entrées de journalisation. Une entrée de journal contient : un identifiant unique, une date, une sévérité et un message. La sévérité du message peut être soit « critique », soit « avertissement » ou soit « information ». D’autres sévérités doivent pouvoir être utilisées sans avoir d’impact sur les applications déjà développées et déployées.

Les entrées de journaux sont envoyées directement dans l’outil d’échanges RabbitMQ. Vous devez utiliser les « topics » afin de diriger (« router ») les messages vers les bonnes applications (i.e. celles qui se sont abonnées).

Les applications qui envoient leurs entrées n’attendent pas de résultat autre que l’entrée est prise en compte. Des applications peuvent s’abonner afin de traiter les messages. Les applications qui s’abonnent à ce genre de messages le font généralement pour : soit faire des statistiques, soit analyser les entrées et lever des alertes de sécurité, ou soit garder une trace des entrées en les stockant dans des dépôts de données (fichiers, SGBD, NoSQL, etc.).

Dans votre solution :

* Les applications « DSED\_Producteur1 » et « DSED\_Producteur2 » simulent des entrées de journaux afin que vous puissiez faire vos tests. Elles envoient des messages à l’échange « ex2-bus-entreprise » avec un sujet qui contient la sévérité, le nom de l’application et le mot clef « journal ». Afin de ne pas répéter trop de code les deux programmes utilisent une bibliothèque de classes qui contient le code utile. Les deux applications ne font qu’instancier la classe « ProducteurEntreesJournal » en lui passant le nom de l’échange et le nom du programme. La méthode « Executer » de cette classe est à compléter (Voir questions plus bas). Son rôle est de déclarer l’échange, de générer des messages à chaque appui de la touche « retour chariot »
* L’application « DSED\_Examen2\_Web » est une application ASP.Net MVC. Elle consomme les messages qui concernent les entrées de journaux de toute sévérité. L’application propose une page web qui affiche en temps « réel » le nombre d’entrées de chaque sévérité reçues depuis le début de la journée. Le projet est configuré pour que la page de statistiques soit la page par défaut au démarrage de votre application (contrôleur et action). *Pour simplifier le code, au démarrage de la page, les statistiques ne sont pas affichées. Il faut attendre la première entrée après la connexion d’un client pour que les statistiques commencent à s’afficher (i.e. il ne se passe rien à la connexion).*
* L’application « DSED\_Examen2\_CopierToutesLesEntreesCritiquesVersFichiers » récupère tous les messages qui contiennent des données de journalisation de sévérité « critique » et les enregistre dans un fichier texte. Le nom du fichier contient minimalement l’identifiant du message.
* L’application « DSED\_Examen2\_BusEntrepriseProxy » est une application ASP.Net MVC. Elle permet d’envoyer des entrées de journaux à partir du protocole REST. Elle reçoit en paramètres toutes les informations d’une entrée de journal avec en plus le nom de l’application et la sévérité

**Contenu de la solution VS et résumé :**

* Les applications de l’entreprise vont produire des entrées de journalisation. Dans votre solution, vous avez deux applications qui produisent des entrées : « DSED\_Producteur1 » et « DSED\_Producteur2 »
* Des applications peuvent s’abonner aux entrées de journalisation. Dans votre solution, vous avez l’application « DSED\_Examen2\_Web ». Vous aurez aussi à créer les applications « DSED\_Examen2\_CopierToutesLesEntreesCritiquesVersFichiers » et « DSED\_Examen2\_BusEntrepriseProxy »
* L’application « DSED\_Examen2\_Web » propose une page qui affiche le nombre d’entrées de chaque sévérité pour la journée actuelle. Elle contient aussi le hub « TableauDeBordHub » qui est exposé à l’adresse « /tableauDeBordHub »
* Les entrées de journaux sont envoyées dans l’outil d’échanges *RabbitMQ* avec un sujet spécifique
* Chaque application / projet est à compléter ou à créer
* Les données sont manipulées à travers un dépôt et stockées dans une simple liste afin de faciliter l’exercice.

1. Donnez la nomenclature que vous choisissez pour former le sujet en tenant compte qu’elle doit contenir la sévérité de l’entrée, le mot clef « journal » ainsi que le nom de l’application (sans caractère spéciaux, accent ou blanc). Exemple de nomenclature donnée en cours : « <vitesse>.<couleur>.<type\_animal> ». (1 pt)

|  |
| --- |
| **Réponse :** |

1. Expliquez le choix de votre nomenclature (ex. ordre des termes par rapport aux filtrages) (3 pts)

|  |
| --- |
| **Réponse :** |

1. Complétez le diagramme suivant (ex. en copiant / collant le dessin dans « Paint » et en faisant l’inverse pour le réinsérer ici, ou un fichier contenant une photo du dessin refait à la main) (2 pts) :



1. Ajoutez les propriétés nécessaires pour représenter une entrée de journal dans la classe « EntreeJournal » (Projet « DSED\_Commun ») (0,5 pt)
2. Ajoutez les instructions nécessaires dans la méthode « Executer » de la classe « ProducteurEntreesJournal » (Projet « DSED\_Commun ») pour qu’elle : (4 pts)
   1. Crée l’échange « ex2-bus-entreprise » s’il n’existe pas déjà
   2. Génère la chaine « sujet » qui contient le sujet dans lequel diriger le message
   3. Crée un objet de type « EntreeJournal » avec des données aléatoires. Une des applications ayant besoin d’afficher le nombre d’entrées par sévérité de la journée active, il est conseillé de générer des dates qui sont aussi bien du jour actuel et antérieur (Rappels : date/heure actuelles : DateTime.Now, méthodes de manipulations de la date : « AddXYZ »)
   4. Envoyez le message dans l’échange
3. Créez le projet «  DSED\_Examen2\_CopierToutesLesEntreesCritiquesVersFichiers » de type « Console ». Le programme doit : (2,5 pts)
   1. Crée l’échange « ex2-bus-entreprise » s’il n’existe pas déjà
   2. Crée la / les files nécessaires si elles n’existe pas déjà
   3. Associe l’échange avec la / les files et les sujets
   4. Enregistre le contenu du message dans un fichier
4. Complétez les méthodes 3 méthodes « ObtenirEntrees<Sévérité>Du » de la classe « DepotJournaux » (Projet « DSED\_Examen2\_Web ») (1 pt)
5. Ajoutez les instructions nécessaires dans la méthode « MettreConsommateurJournauxEnPlace » de la classe « ConsommateurFileJournalisation » (Projet « DSED\_Examen2\_Web ») pour qu’elle : (5 pts)
   1. Crée l’échange « ex2-bus-entreprise » s’il n’existe pas déjà
   2. Crée la / les files nécessaires si elles n’existe pas déjà
   3. Associe l’échange avec la / les files et les sujets
   4. Ajoute les entrées de journaux reçues dans le dépôt de données
   5. Utilise l’objet contenu dans la donnée membre « \_tableauDeBordHub » pour demander aux clients du hub SignalR d’afficher le nombre d’entrées pour chaque sévérité de la date courante. Pour cela, créez un objet de type « Statistiques » et renseigne-le.
6. Modifiez le code de la vue « TableauDeBord/Index.cshtml » pour qu’elle affiche les données sur la page. Pour vous faciliter votre code, vous devez utiliser la fonction javascript « modifierLaVue » qui prend un objet de type « Statistiques » et affichage chaque statistique dans la page. (2 pts)
7. Créez le projet « DSED\_Examen2\_BusEntrepriseProxy » de type « ASP.Net MVC ». Le programme doit : (4 pts)
   1. Crée l’échange « ex2-bus-entreprise » s’il n’existe pas déjà
   2. Crée une API REST qui reçoit une entrée de journal en paramètres avec en plus le nom de l’application et la sévérité et qui envoie un message dans l’échange « ex2-bus-entreprise » (Quel verbe utiliser ? Quel est le type du paramètre ?)