



PRÉPARATION

MESSAGES
QUEUES
& SERVICE BUS


Savoir-Lire &
Preuves de concept

5%

Vous avez droit à deux JOKERS :



Vous pouvez placer un JOKER à une sous-question dont vous ne savez pas la réponse sans perdre de points.



PRÉPARATION MESSAGES QUEUES

Savoir-Lire &
Preuves de concept

TECHNOLOGIES explorées

- RabbitMQ
- ActiveMQ de Camel
- ZeroMQ

(ZeroMQ peut être changé)

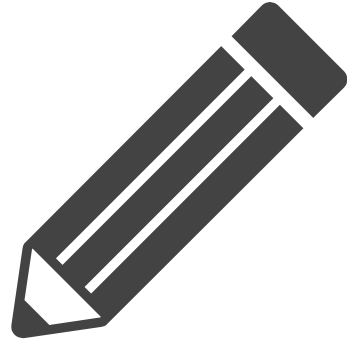
Doit être complété
AVANT la journée
DEV CAMP à la fin de
la session.

1. SAVOIR LIRE

Recherche sur les
queues de messages

Doit être complété
AVANT la journée
DEV CAMP à la fin de
la session.

SAVOIR- LIRE



Effectuer une recherche sur les queues de messages. Dupliquer le document. Répondre aux questions posées directement dans le document.

SAVOIR-LIRE - MESSAGES QUEUES

NIVEAU 1) EXPLORATION INITIATIQUE 🛸🛸🛸

QUESTIONS GÉNÉRALES

En utilisant les sources suivantes :

<https://www.cloudamqp.com/blog/what-is-message-queuing.html>

<https://aws.amazon.com/message-queue/>

<https://community.automationedge.com/t/activemq-working/2154>

Citer un texte qui explique ce qu'est le message queue ? Message Queues (cloudamqp.com)

A Message Queue provides an **asynchronous communications protocol**, which is a system that puts a message onto a Message Queue and does not require an immediate response to continuing processing. Email is probably the best example of asynchronous communication. When an email is sent, the sender continues to process other things without needing an immediate response from the receiver. This way of handling messages decouples the producer from the consumer so that they do not need to interact with the message queue at the same time.

Expliquer ensuite dans vos mots ce qu'est le message queue pour vous ?

Un message Queue est un protocole asynchrone.

Cela signifie qu'un message peut être envoyé sans attendre de réponse immédiate, permettant ainsi au système d'envoyer le message et de continuer à fonctionner.

SAVOIR-LIRE - MESSAGES QUEUES

NIVEAU 2) ANALYSE des qualités et défauts 👍❤️👎

QUESTIONS GÉNÉRALES

En utilisant les sources suivantes :

<https://www.ibm.com/cloud/learn/message-queues>

https://en.wikipedia.org/wiki/Message_queue

<https://blog.iron.io/top-10-uses-for-message-queue/>

<https://medium.com/singhal-labs/messaging-queue-d6dcd6995775>

Quelles sont les bénéfices des messages queues ?

- fiabilité de la livraison des messages
- connexion entre applications
- polyvalence
- résilience
- sécurité renforcée
- transferts de fichiers intégrés

SAVOIR-LIRE - MESSAGES QUEUES

individuel

NIVEAU 3) LE NIVEAU APPLICATIONS 🎀🎀🎀

QUESTIONS d'ÉTUDES de CAS

En utilisant les sources suivantes :

<https://gamedevelopment.tutsplus.com/tutorials/how-to-implement-and-use-a-message-queue-in-your-game--cms-25407>

<https://medium.com/curai-tech/to-queue-or-not-to-queue-simplifying-our-messaging-architecture-with-socketio-30bb14ff0165>

<https://www.cloudamqp.com/blog/why-message-queues-for-iot-projects.html>

<https://www.helpsystems.com/robot/resources/articles/two-way-message-response-robot>

Expliquer dans **quelles applications** des messages queues pourraient être utilisées et pour faire quoi ?
Inventez les logiciels. Soyez précis et détaillé.

dans une application de commerce en ligne pour acheter des vêtements, chaussures et autres.

ils sont utilisés pour des microservices indépendants tels que gestion de commande, paiements, livraison, notification et autres.

Donnez 1 exemple en **détaillant l'info** échangée par message.

1. Lorsqu'un client valide son panier, un message est placé dans une file appelée OrderQueue.
2. Un service dédié lit les messages dans OrderQueue, vérifie les informations de paiement, puis envoie un message dans PaymentQueue une fois le paiement validé.
3. Une fois le paiement réussi, un message est transmis à ShippingQueue contenant l'adresse d'expédition. Le

SAVOIR-LIRE - MESSAGES QUEUES

individuel

NIVEAU 4) LE NIVEAU PROGRAMMEUR

POUR LA TECHNOLOGIE X (RabbitMQ)

1. Quelle est l'url pour télécharger les librairies de développement ?
(<https://www.rabbitmq.com/download.html>)(<https://www.rabbitmq.com/client-libraries/devtools.htmlrabbitmq.com>)
2. Quelle est la licence de cette technologie ? Quel organisme ou cie est l'auteur ?
(**Licence du serveur : Mozilla Public License 2.0**
Licence des clients : La plupart sont en double licence MPL 2.0 / Apache 2.0
Auteur : Développé à l'origine par Pivotal Software, RabbitMQ est désormais maintenu par VMware et Broadcom.)
3. Est-ce possible de l'utiliser en Java ? en JavaScript ? en C++ ?
(**oui pour les 3**)
4. Quels sont les avantages dont se vante la technologie sur son site web ?
(**fiabilité, polyvalence, open source, extensibilité et interopérabilité**) sont les avantages mis en avant sur le site officiel
5. Comment cela fonctionne-t-il, dans vos mots si possible ?
(**RabbitMQ est un broker de messages qui facilite la communication entre différentes applications ou services.**)
6. Les utilisateurs & la popularité :

Sources

Tutoriels officiels RabbitMQ : <https://www.rabbitmq.com/tutorials>

Guide pour débutants - CloudAMQP : <https://www.cloudamqp.com/blog/part1-rabbitmq-for-beginners-what-is-rabbitmq.html>

Tutoriel RabbitMQ sur TutorialsPoint : <https://www.tutorialspoint.com/rabbitmq/index.htm>

Playlist YouTube - RabbitMQ de débutant à avancé : <https://www.youtube.com/playlist?list=PLzewa6pjbr3J5SNo7-hLK6ChB8l1g68B7>

Guide d'installation de RabbitMQ sur Windows :

<https://www.svix.com/resources/guides/rabbitmq-windows-install-guide/CloudAMQP+1CloudAMQP+1TutorialsPointYouTubesvix.com>

Échantillon de code producteur

```
import com.rabbitmq.client.Channel;
import com.rabbitmq.client.Connection;
import com.rabbitmq.client.ConnectionFactory;

public class Sender {
    private final static String QUEUE_NAME = "hello";

    public static void main(String[] argv) throws Exception {
        ConnectionFactory factory = new ConnectionFactory();
        factory.setHost("localhost");
        try (Connection connection = factory.newConnection();
            Channel channel = connection.createChannel()) {
            channel.queueDeclare(QUEUE_NAME, false, false, false, null);
            String message = "Bonjour RabbitMQ!";
            channel.basicPublish("", QUEUE_NAME, null, message.getBytes());
            System.out.println(" [x] Envoyé " + message + "");
        }
    }
}
```

Échantillon de code consommateur

```
import com.rabbitmq.client.*;

public class Receiver {
    private final static String QUEUE_NAME = "hello";

    public static void main(String[] argv) throws Exception {
        ConnectionFactory factory = new ConnectionFactory();
        factory.setHost("localhost");
        try (Connection connection = factory.newConnection();
            Channel channel = connection.createChannel()) {
            channel.queueDeclare(QUEUE_NAME, false, false, false, null);
            System.out.println(" [*] En attente de messages. Pour quitter, appuyez sur CTRL+C");

            DeliverCallback deliverCallback = (consumerTag, delivery) -> {
                String message = new String(delivery.getBody(), "UTF-8");
                System.out.println(" [x] Reçu '" + message + "'");
            };
            channel.basicConsume(QUEUE_NAME, true, deliverCallback, consumerTag -> { });
        }
    }
}
```

SAVOIR-LIRE - MESSAGES QUEUES

NIVEAU 4) LE NIVEAU PROGRAMMEUR

Fournissez 5 LIENS de votre cru pour en apprendre plus sur les messages QUEUES ?

<https://www.youtube.com/playlist?list=PLzewa6pjbr3J5SNo7-hLK6ChB8l1g68B7>

<https://www.cloudamqp.com/blog/part1-rabbitmq-for-beginners-what-is-rabbitmq.html>

<https://kafka.apache.org/intro>

<https://www.rabbitmq.com/getstarted.html>

<https://www.ibm.com/fr-fr/topics/message-queue>

Fournissez quelques liens sur des applications ou librairies qui vous intéressent ?

<https://github.com/rabbitmq/rabbitmq-java-client>

<https://docs.bullmq.io/>

<https://activemq.apache.org/>

<https://nats.io/>

<https://zeromq.org/>

COMPAREZ les MESSAGES QUEUES

NOM	RABBIT MQ	ACTIVE MQ	ZERO MQ
Fiabilité de livraison	Messages persistants, accusés de réception, gestion des duplications	Support des transactions, redelivery configurable	Pas de persistance native, fiabilité à gérer manuellement
Compatibilité protocolaire	AMQP, MQTT, STOMP, HTTP, WebSocket	OpenWire, AMQP, STOMP, MQTT, REST	ATCP, IPC, inproc, multicast, mais pas de protocoles de messagerie standard
Performance	Haute performance, clustering	Bonne performance, adapté aux systèmes Java	Très haute performance, ultra-rapide, faible latence
Facilité d'intégration	Clients pour Java, Python, .NET, Go, JS, PHP, etc.	Excellente compatibilité JMS, plugins pour Spring	Intégration manuelle, mais disponible dans de nombreux langages
Simplicité	Interface web de gestion, plugins	Console de gestion,	Pas d'interface : tout



PRÉPARATION SERVICES BUS

Savoir-Lire &
Preuves de concept

TECHNOLOGIES explorées

- Camel
- Mule
- WSO2

(WSO2 ou Mule
peut être changé)

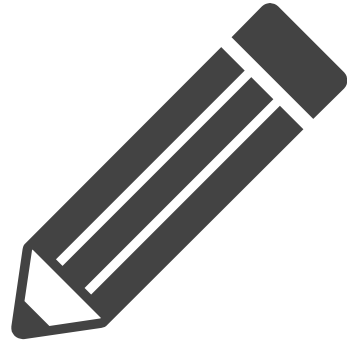
Doit être complété
AVANT la journée
DEV CAMP à la fin de
la session.

1. SAVOIR LIRE

Recherche sur les
Services Bus

Doit être complété
AVANT la journée
DEV CAMP à la fin de
la session.

SAVOIR- LIRE



Effectuer une recherche sur les services bus. Dupliquer ce document. Répondre aux questions posées directement dans le document.

SAVOIR-LIRE - SERVICES BUS

NIVEAU 1) EXPLORATION INITIATIQUE 🚀🚀🚀

QUESTIONS GÉNÉRALES

En utilisant les sources suivantes :

<https://www.hcltech.com/blogs/everything-you-need-know-about-enterprise-service-bus-esb>

<https://wso2.com/what-is-an-enterprise-service-bus/>

https://en.wikipedia.org/wiki/Enterprise_service_bus

<https://searcharchitecture.techtarget.com/definition/Enterprise-Service-Bus-ESB>

<https://www.rootstack.com/en/blog/enterprise-service-bus-dummies-heres-what-you-need-know>

(section What does an ESB do)

Citer un texte qui explique ce qu'est le service bus ?

« Un Enterprise Service Bus (ESB) est un modèle d'architecture qui permet à différentes applications de communiquer entre elles. L'ESB agit comme une couche middleware qui standardise la communication, la transformation des données et le routage des messages entre des systèmes disparates. Cela découple les systèmes les uns des autres, permettant une plus grande flexibilité et évolutivité dans l'intégration d'entreprise.

>>

Expliquer ensuite **dans vos mots** ce qu'est le service bus pour vous ?

le service bus cest comme un chef d'orchestre numérique un peu, sans chef d'orchestre ca jouerait en désaccord et donc ca ne marche pas.

SAVOIR-LIRE - SERVICE BUS

NIVEAU 1) EXPLORATION INITIATIQUE 🛸🛸🛸

QUESTIONS GÉNÉRALES sur la TERMINOLOGIE

En utilisant les sources suivantes :

https://www.tutorialspoint.com/apache_camel/apache_camel_components.htm

https://www.tutorialspoint.com/apache_camel/apache_camel_endpoints.htm

<https://stackoverflow.com/a/19706238/5981056>

<https://camel.apache.org/manual/latest/book-getting-started.html>

<https://docs.mulesoft.com/connectors/>

Tentez de clarifier les éléments suivants, **dans vos mots en français**.

CONNECTOR : il agit comme une interface entre camel et des systemes ecternes

COMPONENT : permet de connecter des routes à divers systemes externe et services

ROUTE : une séquence d'étapes de traitement appliquées à un messages lors de son parcours d'une source à une destination

ENDPOINT : un point de terminaison qui représente une source ou une destination pour les message

SAVOIR-LIRE - SERVICE BUS

NIVEAU 2) ANALYSE des qualités et défauts 👍❤️👎

QUESTIONS GÉNÉRALES sur les AVANTAGES

En utilisant les sources suivantes :

<https://searcharchitecture.techtarget.com/definition/Enterprise-Service-Bus-ESB>

<https://www.mulesoft.com/resources/esb/why-use-esb>

<https://www.arcesb.com/blog/integration/20200114-modern-esb>

<https://www.hcltech.com/blogs/everything-you-need-know-about-enterprise-service-bus-esb>

https://en.wikipedia.org/wiki/Enterprise_service_bus#Key_benefits

Quelles sont les bénéfices des services bus ?

intégration facilitée des systèmes hétérogènes

réduction du couplage entre les applications

centralisation de la gestion des intégrations

renforcement de la sécurité

réduction des coûts de développement

évolutivité

soutien des transformations de données

SAVOIR-LIRE - SERVICE BUS

NIVEAU 3) LE NIVEAU APPLICATIONS 🎀🎀🎀

QUESTIONS d'ÉTUDES de CAS

En utilisant les sources suivantes :

<https://docs.mulesoft.com/connectors/> & <https://www.mulesoft.com/platform/cloud-connectors>

<https://camel.apache.org/components/latest/>

<https://docs.wso2.com/display/ESBCONNECTORS/> & <https://store.wso2.com/store/assets/esbconnector/list>

Nous avons parlé la semaine passée des files de messages (messages queue). C'est la partie des ESB qui est devenue super-populaire. Quels sont les autres connecteurs ou composants ou end-point que l'on peut exploiter avec des vrais ESB ?

- Listez en au moins 12 tout logiciel confondu.

Salesforce Connector, SAP Connector, Amazon S3 Connector, Workday Connector, Kafka Connector, ActiveMQ Component, File Component, HTTP Component, JMS Component, Twitter Connector, Salesforce Connector(WSO2) et Gmail Connector

- Mentionnez pour au moins 3 d'entre eux (un par participant) quel est l'usage que vous en feriez dans un vrai projet logiciel ou situation d'entreprise.

1. Salesforce Connector (MuleSoft) – Projet CRM pour une entreprise de services financiers

2. File Component (Apache Camel) – Automatisation du traitement des commandes dans une entreprise de e-commerce

SAVOIR-LIRE - SERVICE BUS

individuel

NIVEAU 4) LE NIVEAU PROGRAMMEUR

POUR LA TECHNOLOGIE X (remplacer X, une fiche par participant)

1. Quelle est l'url pour télécharger les librairies de développement
<https://www.rabbitmq.com/download.html>
2. Quelle est la licence de cette technologie ? Quel organisme ou cie est l'auteur ?
MPL 2.0 (Mozilla Public License), Initialement développé par LShift et CohesiveFT, puis acquis par VMware. Aujourd'hui maintenu par VMware Tanzu.
3. Est-ce possible de l'utiliser en Java ? en JavaScript ? en C++ ? oui
4. Quels sont les avantages dont se vante la technologie sur son site web ?
(Fiabilité, interopérabilité, Haute disponibilité, Extensible et Performance)
5. Comment cela fonctionne-t-il, dans vos mots si possible ?
il agit comme une boîte aux lettres entre les applications
6. Les utilisateurs & la popularité :
 - a. Est-ce qu'une compagnie est connue pour utiliser cette technologie ? Si oui qui ? NASA, Mozilla, Reddit
 - b. Est-ce qu'on a une estimation du nombre d'utilisateurs selon la compagnie ? Plus de 35 000 entreprises dans le monde
7. Donnez un échantillon de code pour votre technologie (dans une diapo séparée)
8. Donnez au moins 5 sources ou tutoriels débutants pour tester votre technologie.

Echantillon de code

```
import com.rabbitmq.client.Channel;
import com.rabbitmq.client.Connection;
import com.rabbitmq.client.ConnectionFactory;

public class Send {
    private final static String QUEUE_NAME = "hello";

    public static void main(String[] argv) throws Exception {
        ConnectionFactory factory = new ConnectionFactory();
        factory.setHost("localhost");
        try (Connection connection = factory.newConnection();
            Channel channel = connection.createChannel()) {
            channel.queueDeclare(QUEUE_NAME, false, false, false, null);
            String message = "Bonjour RabbitMQ!";
            channel.basicPublish("", QUEUE_NAME, null, message.getBytes());
            System.out.println(" [x] Sent " + message + "");
        }
    }
}
```

1. <https://www.rabbitmq.com/getstarted.html>
2. <https://spring.io/guides/gs/messaging-rabbitmq>
3. <https://www.rabbitmq.com/tutorials/tutorial-one-javascript.html>
4. <https://www.tutorialspoint.com/rabbitmq/index.htm>
5. <https://www.digitalocean.com/community/tutorials/how-to-install-and-manage-rabbitmq>

SAVOIR-LIRE - MESSAGES QUEUES

NIVEAU 4) LE NIVEAU PROGRAMMEUR

Fournissez 5 LIENS de votre cru pour en apprendre plus sur les services BUS ?

<https://www.hcltech.com/blogs/everything-you-need-know-about-enterprise-service-bus-esb>

<https://wso2.com/what-is-an-enterprise-service-bus/>

<https://searcharchitecture.techtarget.com/definition/Enterprise-Service-Bus-ESB>

<https://www.rootstack.com/en/blog/enterprise-service-bus-dummies-heres-what-you-need-know>

https://en.wikipedia.org/wiki/Enterprise_service_bus

Fournissez quelques liens sur des applications ou librairies qui vous intéressent ?

<https://camel.apache.org/components/latest/>

<https://wso2.com/integration/>

<https://www.mulesoft.com/platform/enterprise-integration>

COMPAREZ les SERVICE BUS

NOM	CAMEL	MULE	WSO2
Légèreté	Très léger facile a interagir dans une app java	Plus lourd, nécessite la plateforme Anypoint	Léger, surtout en version open source
Simplicité d'utilisation	DSL clair en java ou XML	interface graphique intuitive	Requiert un peu plus de configuration
Intégration multi-protocoles	FTP, JMS, HTTP, Kafka, RABBITMQ.	Très large compatibilité avec APIs et formats	Bonne couverture de protocoles et connecteurs
Open Source et Gratuité	100% open source, apache License	Version open source limitée, version pro payante	100% open source (Apache 2.0)
Flexibilité des routes	Routes programmables dynamiquement	Flow visuel flexible avec transformation	Bonne logique conditionnelle avec médiateurs
Performance et Scalabilité	Excellence en microservices et cloud natif	Bonnes performance dans cloud hybride	Très scalable, clustering intégré

BONNE RECHERCHE !

QUESTIONS dans SLACK ou par
courriel ou par zoom

nadineducegep@gmail.com

PREUVE de concept

”

Une preuve de concept, c'est l'exemple le plus petit que l'on peut faire d'une technologie pour prouver qu'elle fonctionne dans notre environnement.

Cela valide que
notre
environnement
est bien installé

Cela permet de
comparer des
technologies et
choisir.

“

PREUVE DE CONCEPT - MESSAGE BUS

CHAQUE participant fait la preuve de concept de 1 des 3 technologies et aide ses camarades de son équipe. La preuve de concept se réalise dans un langage approuvé (Java, JS, à valider.).

La preuve de concept est d'avoir deux instances de logiciels séparées : AppAmi et AppVoyageur

Le protocole à programmer est le suivant :

1) AppAmi envoie "Qui est là" à AppVoyageur

2) AppVoyageur envoie "Present" à AppAmi

Si possible dans cet ordre.

Vous pouvez tester d'autres comportements selon les capacités et limitations de votre librairie.

=> Donnez le lien vers votre code de preuve de concept (Github public ou zip dans un drive)

LIEN RABBIT MQ : .

LIEN ACTIVE MQ : .

LIEN ZERO MQ : .

PREUVE DE CONCEPT - CAMEL

CHAQUE participant fait la preuve de concept de 1 route camel différente. Vous pouvez travailler dans le même projet ou faire un projet par personne.

La preuve de concept est de partir de l'exemple distribué en classe et d'implémenter une route différente utilisant un différent component ou type de endpoint.

L'objectif est de diversifier nos points de communications possibles pour le jour du DevCamp. Voici des points de contact intéressants.

- ★ **Les fichiers dans un répertoire**
- ★ **Une base de données**
- ★ **Un service de données**
- ★ **Une queue de message externe**

Assurez-vous d'inclure ces chemins dans vos tests :

- ★ **Un formatage de string (variable imprimée dans %s)**
- ★ **Un chemin xpath dans un contenu xml**

LIEN ROUTE 1 : .

LIEN ROUTE 2 : .

LIEN ROUTE 3 : .

BONNE RECHERCHE !

QUESTIONS dans SLACK ou par
courriel ou par zoom

nadineducegep@gmail.com