Отчет по лабораторной работе №2

Згрундо А.Д

студентка 272301

Java Message Service – это Java API, которые позволяют приложениям создавать, отправлять, получать и читать сообщения. JMS API определяет общий набор интерфейсов, который позволяет программам на Java взаимодействовать с другими реализациями messaging.

JMS приложение состоит из следующих частей:

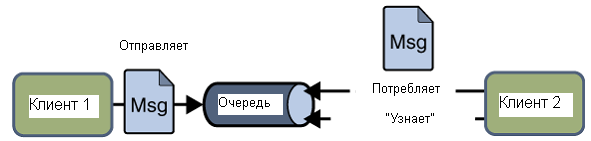
* JMS провайдер – это система, управляющая сообщениями, которая реализовывает JMS API и обеспечивает исполнение административных и управленческих задач. JMS провайдер включен в реализацию платформы Java EE.
* JMS клиенты – это программы или компоненты, написанные на Java, которые производят и потребляют сообщения. Любой компонент приложения Java EE может выступать в качестве клиента JMS.
* Сообщения – это объекты, являющиеся носителями информации.
* Администрируемые объекты – это отконфигурированные JMS объекты, создаваемые администратором для использования клиентами. Пункты назначения (*destinations*) и фабрики соединений (*connection* *factories*) – это администрируемые объекты..

**Point-to-point**.

Подход PTP базируется на концепции очередей сообщений, отправителей и получателей. Каждое сообщение адресуется в определенную очередь, и получающие их клиенты извлекают их из соответствующих очередей. Очереди содержат все отправленные в них сообщения, пока они не будут забраны, либо пока не истечет срок хранения.

Характеристики PTP messaging следующие (рис.2.):

* Каждое сообщение имеет только одного потребителя.
* Отправитель и получатель сообщения не зависят от времени. Получатель может достать сообщение независимо от того, был ли он запущен во время отправки сообщения.
* Получатель узнает об успешной обработке сообщения.

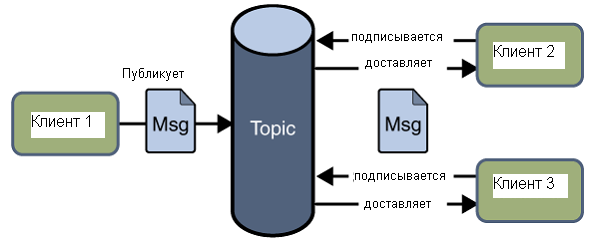


**Publish/subscribe**

В этом случае клиент адресует сообщение в *topic* (набор тем), функция которого напоминает доски объявлений. Публикаторы и подписчики обычно анонимны и могут динамически публиковать и подписываться на сообщения в пределах иерархии содержания топика. Система заботится о распределении сообщений, прибывших от множества публикаторов, множеству подписчиков. Топики содержат сообщения столько времени, сколько требуется на распределение их подписчикам.

* Каждое сообщение может иметь множество потребителей.
* Публикаторы и подписчики зависят от времени. Клиент, который подписывается на топик, может потребить только то сообщение, которое пришло после его подписки. Для потребления сообщений подписчик должен оставаться активным.

JMS API ослабляет эту временную зависимость, предусматривая создание подписчиками длительных подписок (*durable* *subscriptions*), которые позволяют получать сообщения, отправленные в то время, когда подписчики были не активны.



Бин, управляемый сообщениями (далее – *MDB*), – это бин, который позволяет приложению Java EE управлять сообщениями асинхронно. Он обычно выступает в роли слушателя JMS сообщений. Сообщения могут быть отправлены любым Java EE компонентом. MDB может обрабатывать JMS сообщения или другие виды сообщений.

Синхронный запрос - это запрос с ожиданием ответа. Т.е. скрипт создал объект, а объект послал запрос на сервер, это сразу, мгновенно не происходит. И пока этот запрос обрабатывается на сервере скрипт "висит", скрипт просто "висит", он ждет ответ с сервера. Дальше скрипт не выполняется, браузер ничего не делает, он ждет ответ с сервера. Если сервер работает долго или связь с сервером медленная, то это "зависание" может быть даже заметно на глаз. И только тогда, когда сервер все наконец-то пришлет, только тогда скрипт пойдет выполняться дальше. Т.е. скрипт в момент обмена данными с сервером он "висит".

Асинхронный запрос - это запрос без ожидания ответа от сервера. Т.е. скрипт создал объект, а объект послал запрос на сервер. При этом скрипт продолжает выполняться дальше и не ждет ответа от сервера. В какой-то момент данные вернуться.

Использование MDB позволяет параллельно обрабатывать сообщения, не заботясь о сложности асинхронной обработки и дополнительного кода для подписки на очередь сообщений.

Асинхронная обработка реализуется через пул объектов, из которых на обработку сообщения сервер выделят объекты при необходимости.