DPI6 : Architecture / design document

by: Bart Kneepkens
lecturer: Michael Franssen
26 april 2016

Introduction

This document will contain architectural information of the systems that are to be implemented, named MovieRater.

According to the research, also done for DPI6, the message queue that is to be used for the implementation is OpenMQ. The application container will be Glassfish 4.1, and all code will be written in Java.

The system is described as follows:

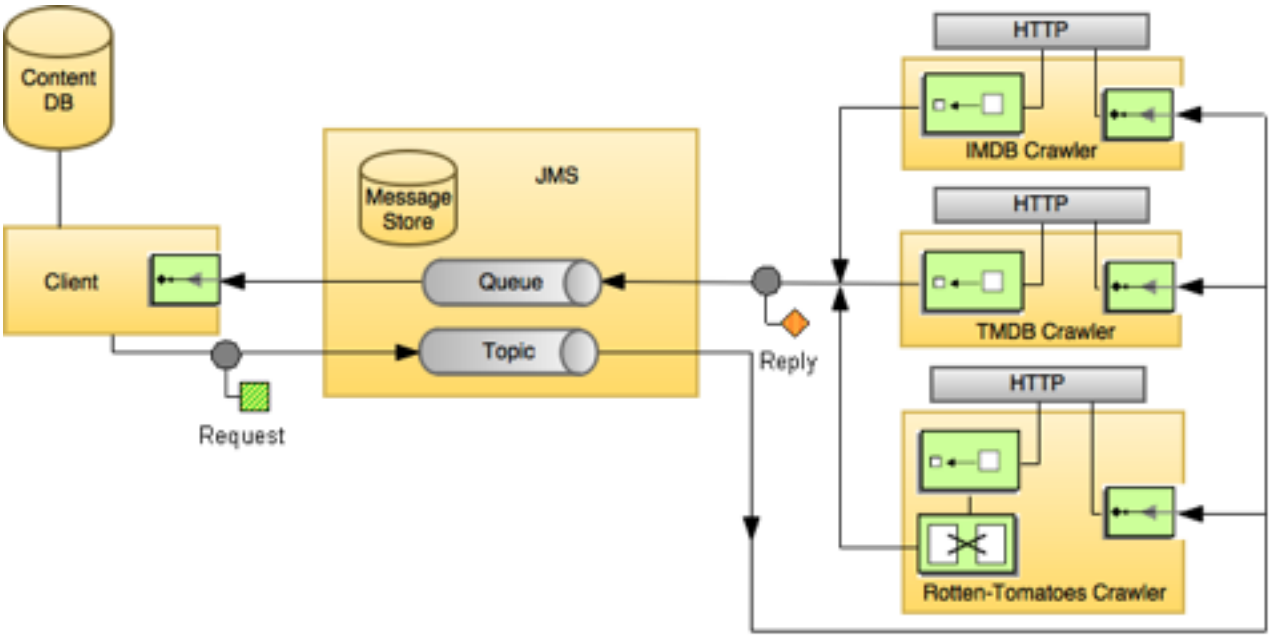
MovieRater is a web-application where the user can query multiple movie databases for the ratings of a movie or series. The user will be presented the results of this query, and be able to enrich them. Also, the user can set (up to three) genres of interest, and it will receive a message if a new movie is added to one of these genres.

Requirements

The system will be implemented to have the following basic requirements:

- R1: The user should be able to enter a query in the system, and retrieve the ratings that represent those on imdb.com, themoviedb.org, and rottentomatoes.com. These ratings should also contain the weight, in other words, the amount of ratings that make the average.
- R2: The user should be able to enrich said ratings with its own, locally.
- R3: The user should be able to enter up to three (3) genres for which it wants to receive notifications, when a new movie gets added to these genres.

System Architecture



Within these systems there will be two sorts of messages; one for requests and one for replies. These messages will have the following attributes:

Messages could for example look like this:

RequestMessage

```
{
    "query": "silence of the lambs"
}
```

Attribute	Description
query	The search query that the user has entered, in String format. The words should be separated by spaces.

ReplyMessage

```
{
    "crawlerID": "YYYYYYYY",
    "rating": "7.8",
    "weight": "1626"
}
```

Attribute	Description
crawlerID	The unique identifier of the web crawler. This is a String that has a static value. See “Static Value Reference” for values.
rating	The rating that the website displays for this movie or series. This will be a Double, and will always be a number between 1 and 10. If the site uses */5 ratings, it will be converted to */10 inside the crawler.
weight	The amount of votes that make the rating attribute. This will be a Integer (32 bit).

For the systems async capabilities (see requirement R3), the following message will be sent by the crawlers and received by the client:

The body could for example look like this:

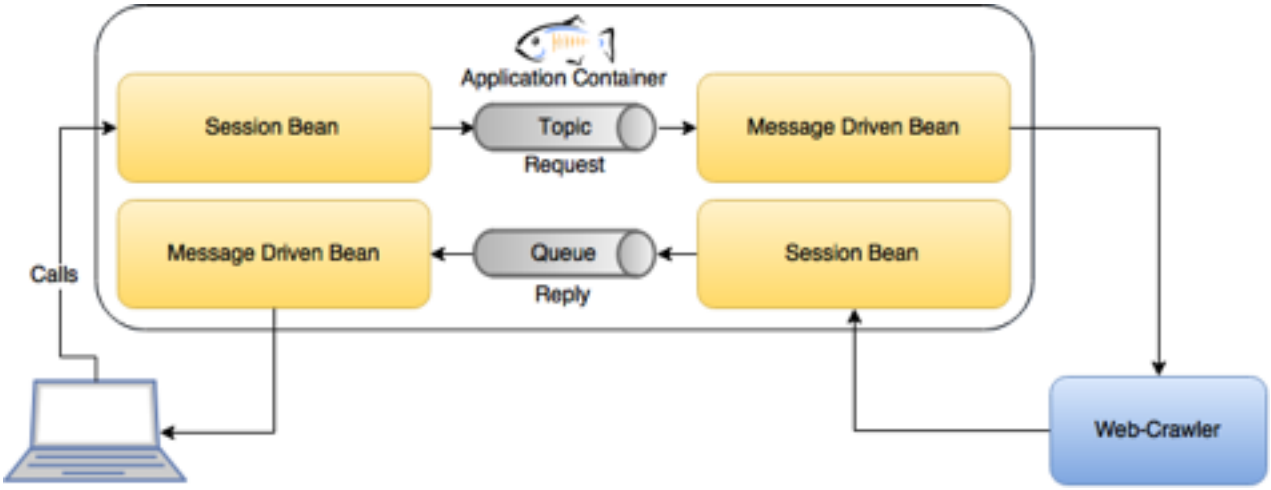
InterestMessage

```
{
        "crawlerID": "YYYYYYYY",
        “description”: “Batman vs. Superman : Dawn of Justice”,
        “genre”: “action”
}
```

Attribute	Description
crawlerID	The unique identifier of the web crawler. This is a String that has a static value. See “Static Value Reference” for values.
description	The description of the movie or series that has been found. This is a String with no assigned length.
genre	The genre in which said movie or series has been added. This is a String with a value, that the user has entered.

Flow diagram

The following diagram describes the steps that are taken whenever a user enters a query.



1. When the user enters a query, its session bean is called.
2. Said session bean posts a Message to the request queue.
3. The web-crawler's MDB (message driven bean) receives the request message.
4. The web-crawler processes the message, and searches the website for the query.
5. The web-crawler calls its session bean.
6. Said session bean posts a Message to the reply queue.
7. The client's MDB (message driven bean) receives the request message.
8. The client processes the message.

General flow per system

The total system can be divided in roughly two (2) types of systems:

- The client
- The crawlers

Each of these systems will have different responsibilities and a different working. In general:

Client

The client will be a web-application, which will be connected to the application server through a java session bean, and JSF. For a better layout of the product, the library PrimeFaces will be used.

The client interact with the controls, which interact with the session bean, which interacts with the application container.

The client also has a local DB where it saves (own generated) content. The crawled content won't be saved here; because that content might become invalid with time.

Crawlers

The crawlers will take the following approach to gathering information:

1. Receive the query
2. Search the website using the query
3. Grab the first result that is returned in the search
4. Search the site for the element that is described in "Static Value Reference".
5. Return the value. If not found, return an empty String.

Static value reference

CrawlerID's

Crawler website	CrawlerID
IMDB	IM
TMDB	TM
Rotten Tomatoes	RT

Element ID's for ratings

Crawler website	Rating value	Rating Weight
IMDB	8,2	10.152
TMDB	7.3	4,117
Rotten Tomatoes	Average Rating:	User Ratings: