Project 5 Report

Team 7: Zixi Liu (000996162) Mengqi Wang (000995512)

[Task 1 2](#_Toc516500155)

[How did you use connection pooling? 2](#_Toc516500156)

[File name, line numbers as in Github & Snapshots showing use in your code 2](#_Toc516500157)

[*context.xml* 5](#_Toc516500158)

[How did you use Prepared Statements? 6](#_Toc516500159)

[File name, line numbers as in Github & Snapshots showing use in your code 6](#_Toc516500160)

[Task 2 9](#_Toc516500161)

[Address of AWS and Google instances 9](#_Toc516500162)

[Have you verified that they are accessible? Does Fablix site get opened both on Google’s 80 port and AWS’ 8080 port? 9](#_Toc516500163)

[Explain how connection pooling works with two backend SQL (in your code)? 9](#_Toc516500164)

[File name, line numbers as in Github & snapshots 10](#_Toc516500165)

[How read/write requests were routed? 10](#_Toc516500166)

[File name, line numbers as in Github & Snapshots 11](#_Toc516500167)

[Task 3 13](#_Toc516500168)

[Have you uploaded the log files to Github? Where is it located? 13](#_Toc516500169)

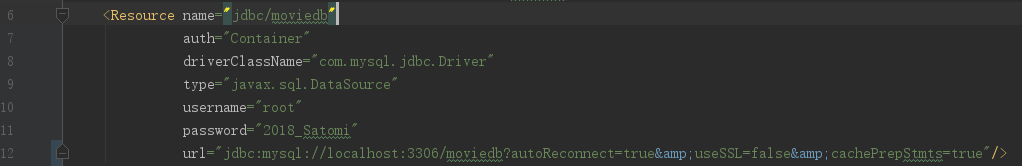
[Have you uploaded the HTML file (with all sections including analysis, written up) to Github? Where is it located? 13](#_Toc516500170)

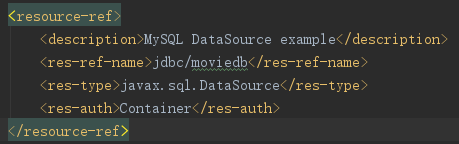
[Have you uploaded the script  to Github? Where is it located? 13](#_Toc516500171)

[Have you uploaded the WAR file and README  to Github? Where is it located? 13](#_Toc516500172)

# Task 1

## How did you use connection pooling?

Add a Resource tag into <context> in context.xml:  


Add a resource-ref tag into <web-app> in web.xml:  


Set a DataSource in the class, using the @Resource injection:



Get Connection from that DataSource:



### File name, line numbers as in Github & Snapshots showing use in your code

#### AddMovieFeatureServlet.java

line 17-18:  


line 33:  


#### AddMovieServlet.java

line 18-19:  


line 34:  


#### AddStarServlet.java

line 21-22:  


line 39:  


#### CredentialServlet.java

line 19-20:  


line 37:   


#### GenresServlet.java

line 25-26:  


line 39:  


#### GetTransactionsSerlvet.java

line 26-27:  


line 41:  


#### LoginServlet.java

line 19-20:  


line 40:  


#### MetadataServlet.java

line 23-24:  


line 33:  


#### MoviesAlphabetServlet.java

line 22-23:  


line 43:  


#### MoviesFulltextSearchServlet.java

line 22-23:  


line 45:  


#### MoviesGenreServlet.java

line 22-23:  


line 42:  


#### MoviesSearchServlet.java

line 22-23:  


line 42:  


#### MoviesServlet.java

line 25-26:  


line 44:  


#### PayingServlet.java

line 26-27:  


line 44:  


#### SingleMovieServlet.java

line 25-26:  

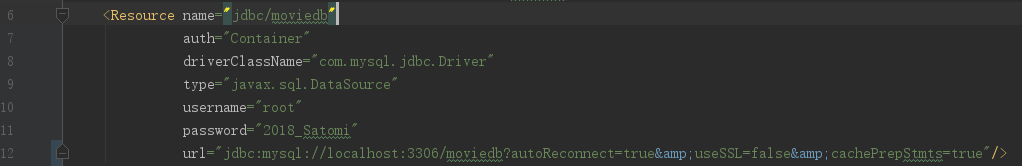

line 52:  


#### SingleStarServlet.java

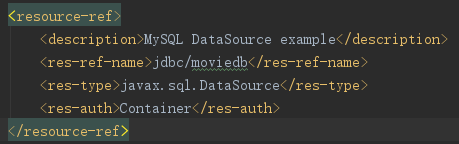
line 23-24:  


line 43:  

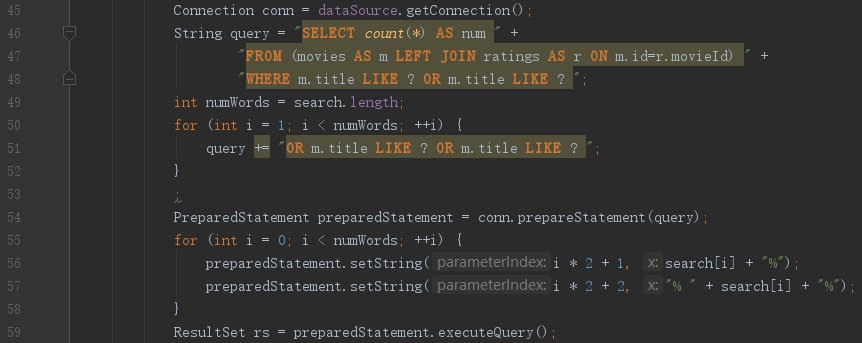

#### context.xml

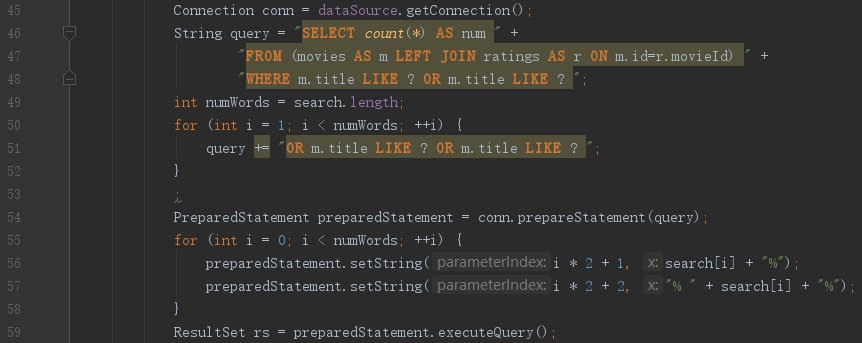
line 6-12:  


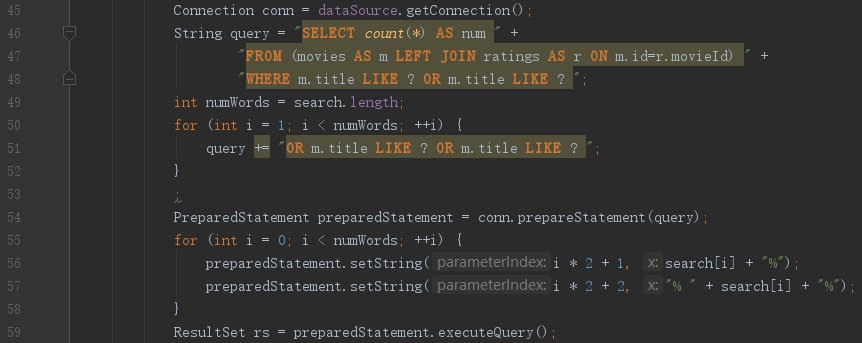
#### web.xml

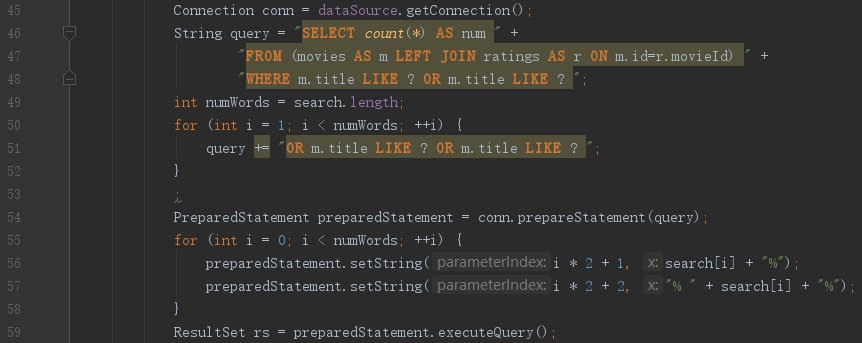
line 11-16:  


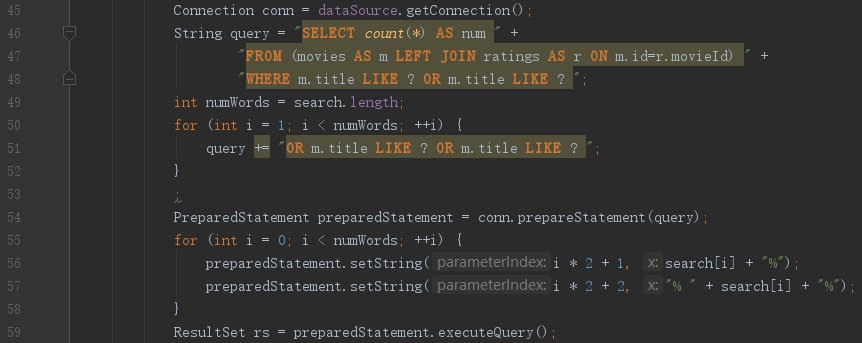
## How did you use Prepared Statements?

First set the context.xml and get Connections in servlets in the same way above:  


Set a String with “?”’s:  


Get PreparedStatement:  


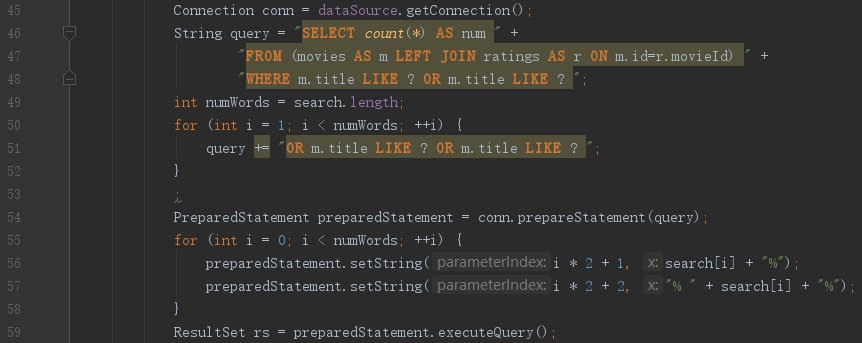
Set the values of “?”’s:  


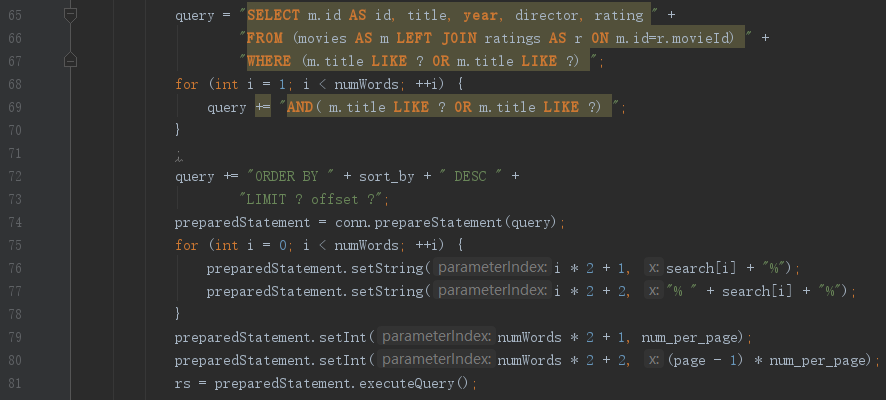
Execute:  


### File name, line numbers as in Github & Snapshots showing use in your code

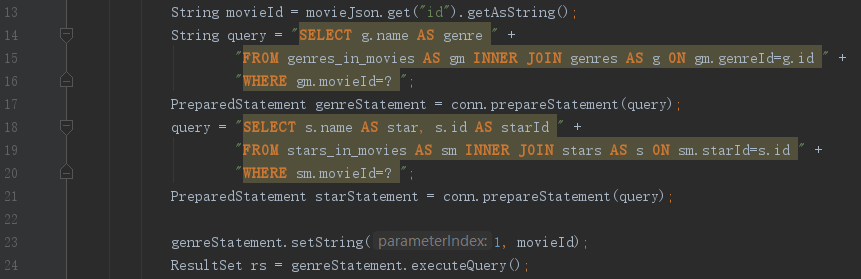
#### MoviesFulltextSearchServlet.java

line 22-23:  


line 45-59:  


line 65-81:  


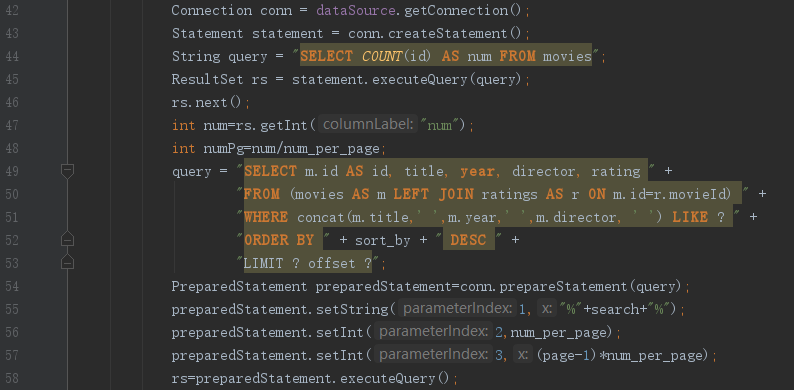
#### GetMovieGenreStar.java

line 13-24:  


line 30-31:  


#### MoviesSearchServlet.java

line 22-23:  


line 42-58:  


# Task 2

## Address of AWS and Google instances

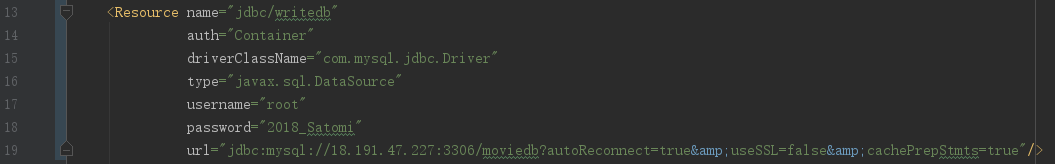
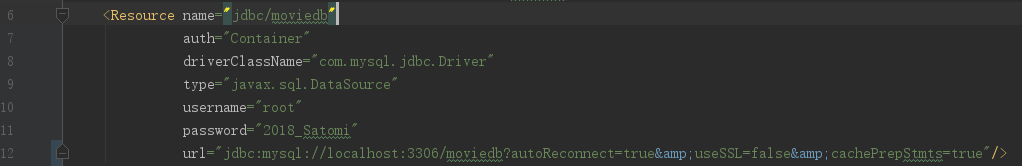
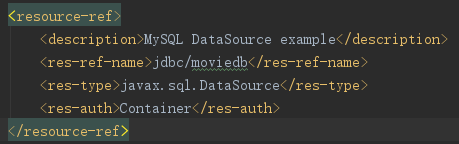
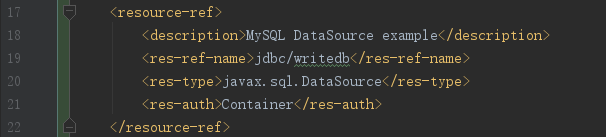
|  |  |  |  |
| --- | --- | --- | --- |
| Google IP | AWS1 IP | AWS2 IP(Master) | AWS2 IP(Slave) |
| 35.237.243.43 | 13.59.21.230 | 18.191.47.227 | 13.59.251.169 |

## Have you verified that they are accessible? Does Fablix site get opened both on Google’s 80 port and AWS’ 8080 port?

Yes.

## Explain how connection pooling works with two backend SQL (in your code)?

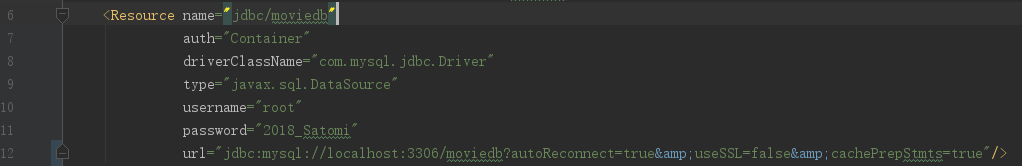
In terms of reading, either of the backends chooses their own local database as their database source. As for writing, they all refer to the master’s database.

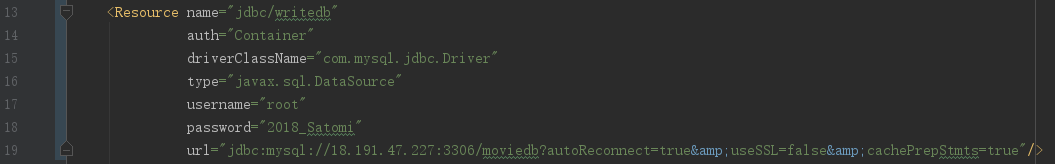
We can separate those two request in the context.xml:  
  
and web.xml:  
  


In other words, servlets only reading databases will connect to jdbc/moviedb, and those writing databases will only connect to jdbc/writedb.

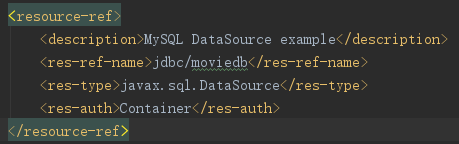
### File name, line numbers as in Github & snapshots

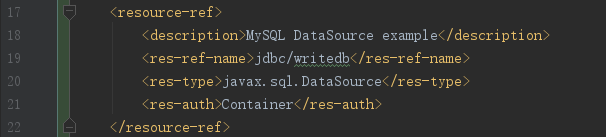
#### context.xml

line 6-12:  


line 13-19:  


#### web.xml

line 11-16:  


line 17-22:  


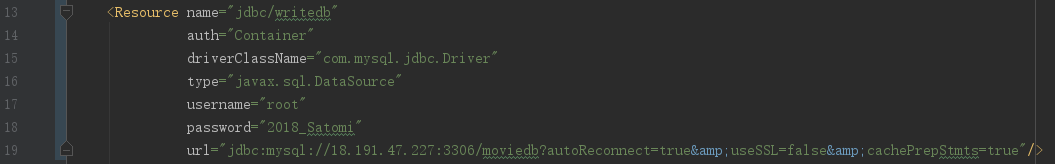
#### And all other files Task one, with same lines and snapshots.

## How read/write requests were routed?

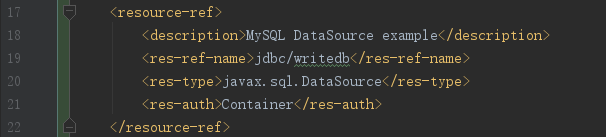
Add a new resource (jdbc/writedb) in context which uses the master mysql (mysql://18.191.47.227:3306/moviedb) instead of the local one. Set any servlet having writing action to use jdbc/writedb instead of jdbc/moviedb, and then it’s read/write separated.

### File name, line numbers as in Github & Snapshots

#### context.xml

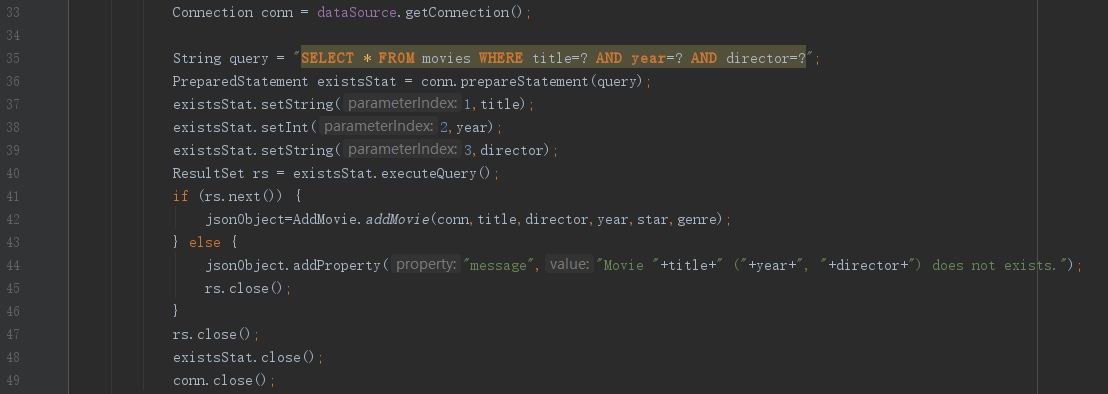
line 13-19:  


#### web.xml

line 17-22:  


#### AddMovieFeatureServlet.java

line 17-18:  


line 33-49:  


#### AddMovieServlet.java

line 18-19:  


line 34:  


#### AddStarServlet.java

line 21-22:  


line 39:  


#### PayingServlet.java

line 26-27:  


line 44:  


#### And all other files in Task 1, with “Resource(name=”jdbc/moviedb”)”

# Task 3

## Have you uploaded the log files to Github? Where is it located?

Yes.

/WebContent/single-instance-http-10-full/

/WebContent/single-instance-http-10-full/

/WebContent/single-instance-http-10-full/

/WebContent/single-instance-http-10-full/

/WebContent/single-instance-http-10-full/

/WebContent/single-instance-http-10-full/

/WebContent/single-instance-http-10-full/

/WebContent/single-instance-http-10-full/

/WebContent/single-instance-http-10-full/

## Have you uploaded the HTML file (with all sections including analysis, written up) to Github? Where is it located?

Yes. /WebContent/jmeterResult.html

## Have you uploaded the script  to Github? Where is it located?

Yes. /script.py

## Have you uploaded the WAR file and README  to Github? Where is it located?

Yes. /project2.war