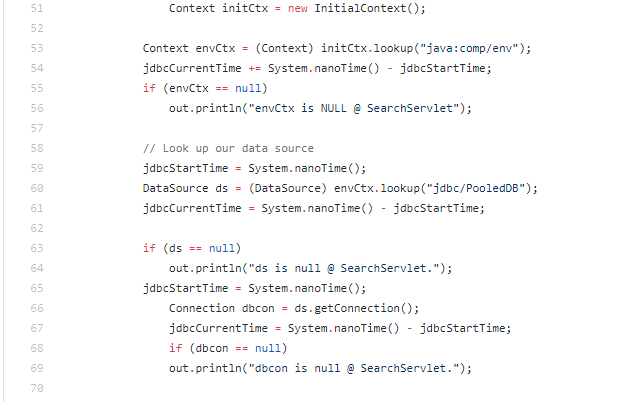
**Task 1:**

The File Path: [cs122b-winter19-team-29](https://github.com/UCI-Chenli-teaching/cs122b-winter19-team-29)/[Fabflix\_Web\_Project](https://github.com/UCI-Chenli-teaching/cs122b-winter19-team-29/tree/master/Fabflix_Web_Project)/[src](https://github.com/UCI-Chenli-teaching/cs122b-winter19-team-29/tree/master/Fabflix_Web_Project/src)/SearchServletAPI.java

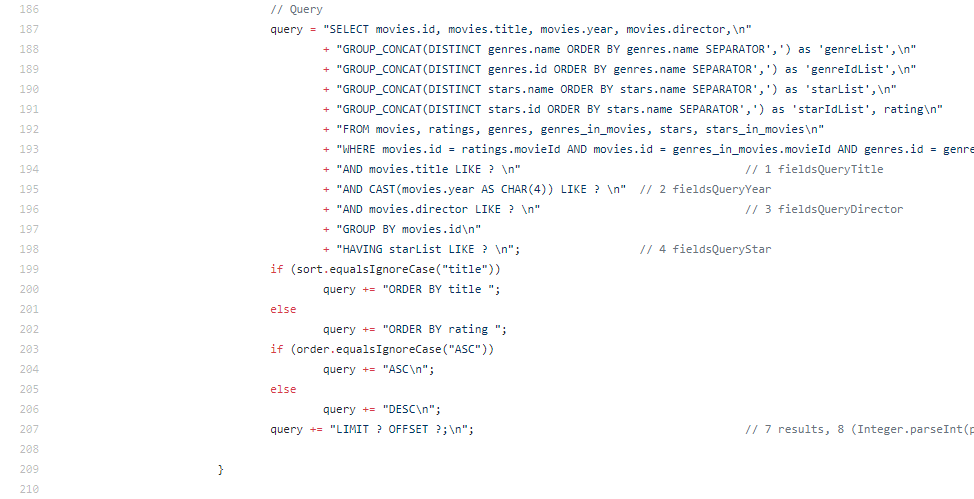
The Line numbers for pooled connection: 51 - 69

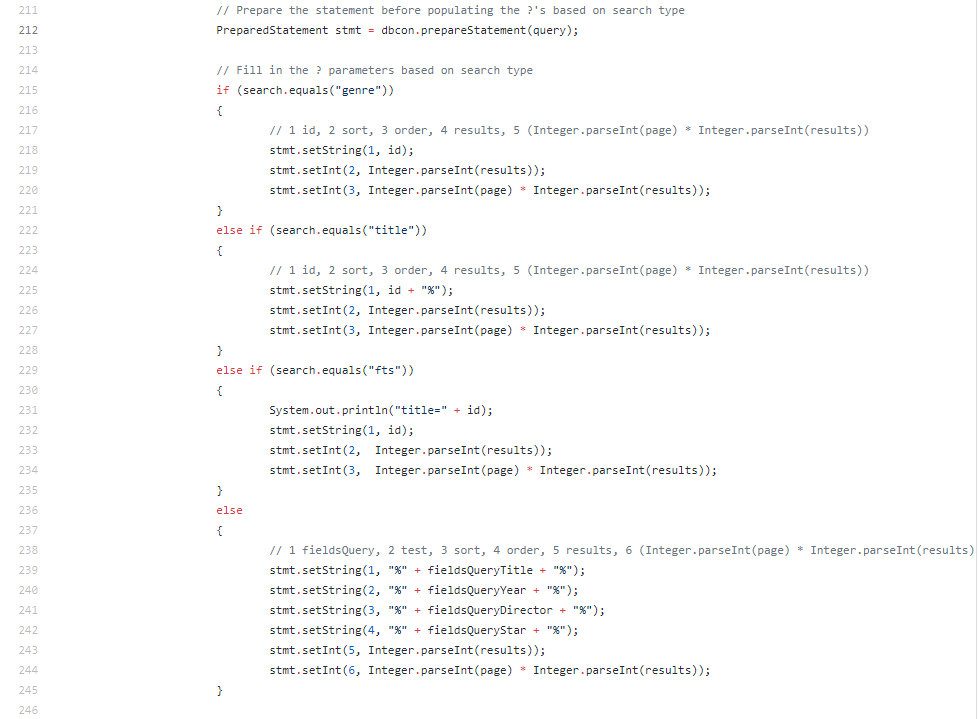
The How:

* First, we added the database connections into the context.xml and web.xml files in both master and slave AWS instances
* Next, we start with creating the pooled connection by creating a new Context object and binding it to the JDBC driver.
* Finally, we establish the connection to the database ‘moviedb’ by looking it up in the context of the pooled connection under PooledDB, and assuming everything is fine and not null up to this point, we have connected this instance of the pooled connection context to work with PreparedStatements to query the ‘moviedb’ database.



* Below are screenshots of PreparedStatements being used for searching in Fabflix:



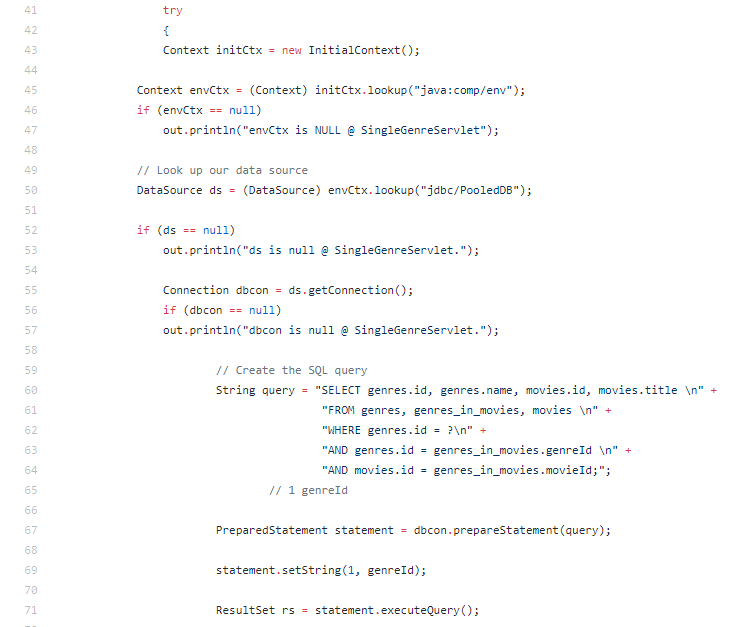


**Note:** The remainder of the task 1 document will show a few more file paths for other servlets utilizing the pooled connection code and PreparedStatements.

The File Path: [cs122b-winter19-team-29](https://github.com/UCI-Chenli-teaching/cs122b-winter19-team-29)/[Fabflix\_Web\_Project](https://github.com/UCI-Chenli-teaching/cs122b-winter19-team-29/tree/master/Fabflix_Web_Project)/[src](https://github.com/UCI-Chenli-teaching/cs122b-winter19-team-29/tree/master/Fabflix_Web_Project/src)/SingleGenreServletAPI.java

The Line numbers for pooled connection: 41 - 57

More screenshots:



The File Path: [cs122b-winter19-team-29](https://github.com/UCI-Chenli-teaching/cs122b-winter19-team-29)/[Fabflix\_Web\_Project](https://github.com/UCI-Chenli-teaching/cs122b-winter19-team-29/tree/master/Fabflix_Web_Project)/[src](https://github.com/UCI-Chenli-teaching/cs122b-winter19-team-29/tree/master/Fabflix_Web_Project/src)/SingleMovieServletAPI.java

The Line numbers for pooled connection: 44 - 58

More screenshots:

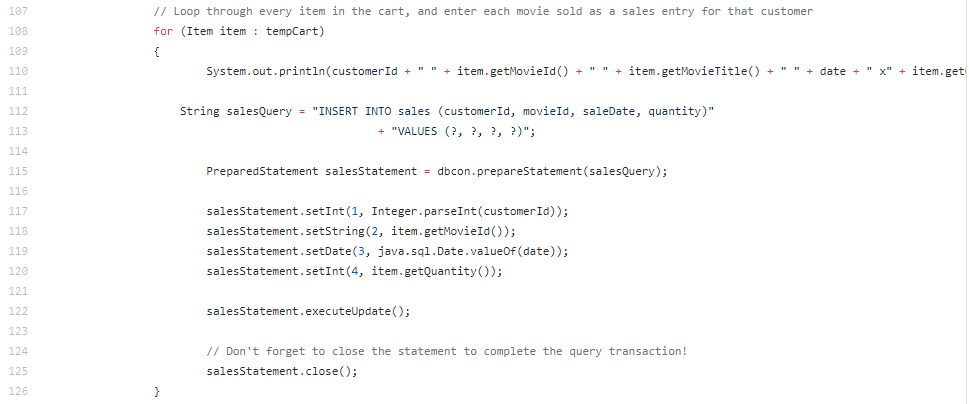


The File Path: [cs122b-winter19-team-29](https://github.com/UCI-Chenli-teaching/cs122b-winter19-team-29)/[Fabflix\_Web\_Project](https://github.com/UCI-Chenli-teaching/cs122b-winter19-team-29/tree/master/Fabflix_Web_Project)/[src](https://github.com/UCI-Chenli-teaching/cs122b-winter19-team-29/tree/master/Fabflix_Web_Project/src)/CheckoutServletAPI.java

The Line numbers for pooled connection: 56 - 70

More screenshots:

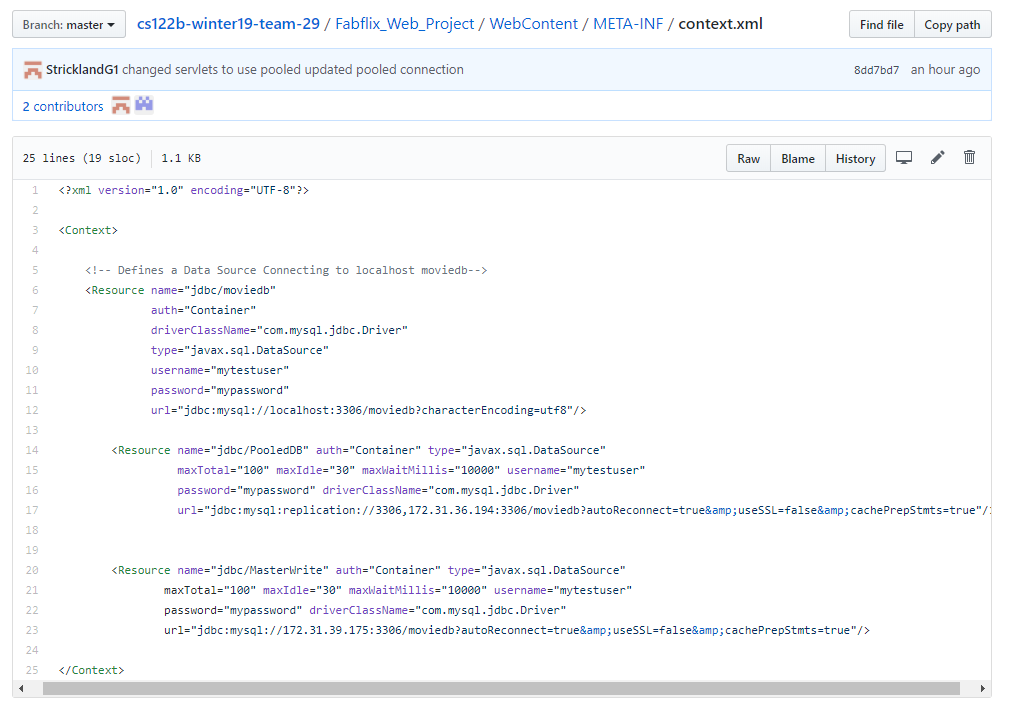




**Task 2:**

We updated the Fabflix project’s context.xml and web.xml to include the connection pooling resource for the local version, which is also applied to the master and slave versions when the project’s .war file is deployed to the respective AWS instances. We pooled the master and slave connections under one “PooledDB” resource, and another for just strictly writing to the master instance under “MasterWrite”:

[cs122b-winter19-team-29](https://github.com/UCI-Chenli-teaching/cs122b-winter19-team-29)/[Fabflix\_Web\_Project](https://github.com/UCI-Chenli-teaching/cs122b-winter19-team-29/tree/master/Fabflix_Web_Project)/[WebContent](https://github.com/UCI-Chenli-teaching/cs122b-winter19-team-29/tree/master/Fabflix_Web_Project/WebContent)/[META-INF](https://github.com/UCI-Chenli-teaching/cs122b-winter19-team-29/tree/master/Fabflix_Web_Project/WebContent/META-INF)/context.xml



[cs122b-winter19-team-29](https://github.com/UCI-Chenli-teaching/cs122b-winter19-team-29)/[Fabflix\_Web\_Project](https://github.com/UCI-Chenli-teaching/cs122b-winter19-team-29/tree/master/Fabflix_Web_Project)/[WebContent](https://github.com/UCI-Chenli-teaching/cs122b-winter19-team-29/tree/master/Fabflix_Web_Project/WebContent)/[WEB-INF](https://github.com/UCI-Chenli-teaching/cs122b-winter19-team-29/tree/master/Fabflix_Web_Project/WebContent/WEB-INF)/web.xml



**Note:** For task 2, we noticed that we can access the project from GCP’s public IP at port 80 using the public IP’s of the master and slave AWS instances in the load balancer, but we need to manually type in a HTML page in the URL (i.e. /index.html).

**Details regarding our AWS/GCP IP’s and instances:**

AWS Master instance:

* Public IP: 3.16.158.144
* Public DNS: ec2-3-16-158-144.us-east-2.compute.amazonaws.com

AWS Slave instance:

* Public IP: 13.59.184.185
* Public DNS: ec2-13-59-184-185.us-east-2.compute.amazonaws.com

GCP instance:

* Public IP: 35.236.87.69

Project URL:

* http:// (Pick any of the IP’s above) :80/CS\_122B\_Fablix\_Project\_API\_Version/index.html