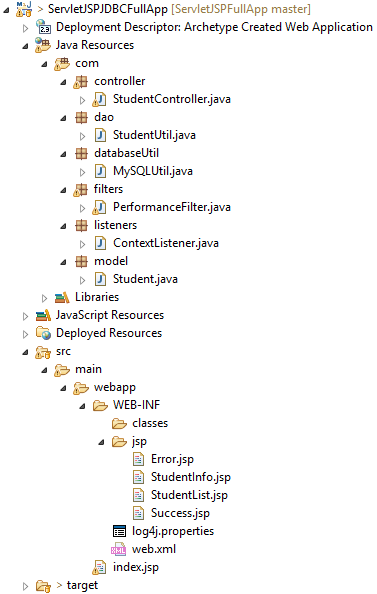
**What this project does??**

This project include almost all concepts in servlets and JSP technology viz. MVC design pattern, servlet as controller, filters, listeners, model, error handling using JSP, database handling using JDBC, logging mechanism etc.

\*\*Main difference between this project and 'ServletJSPHibernateXMLFullAp' project is, in this project instead of mapping various things in web.xml we have used annotations to map servlet, filters, listeners etc. Please go through respective files very carefully for the annotations. And also make a comparative analysis between these two projects. Note that we have used web.xml file only to declare context parameters here, as this is annotation based project rest of the things viz. servlet (in controller package here) , filters(in filter package here), listeners(in listeners package here) are mapped using annotations.

**(Please note that we have used web.xml file in this application only for initializing context parameters as that is not at all possible with annotations. Please refer link '**[**https://www.baeldung.com/context-servlet-initialization-param**](https://www.baeldung.com/context-servlet-initialization-param)**' for more details.**

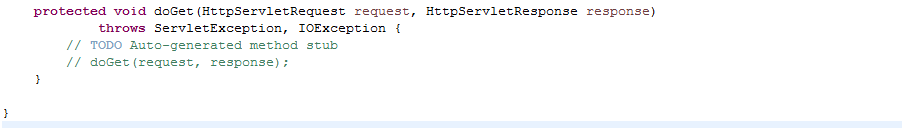
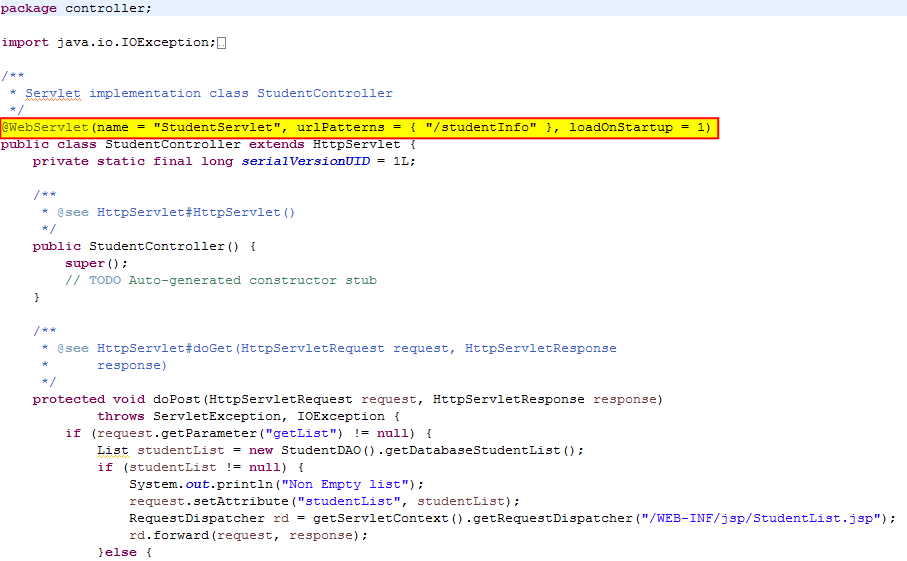
Structure of this project is as follows:



Look for the package structure

1. There is controller package containing servlets acts as controller of MVC

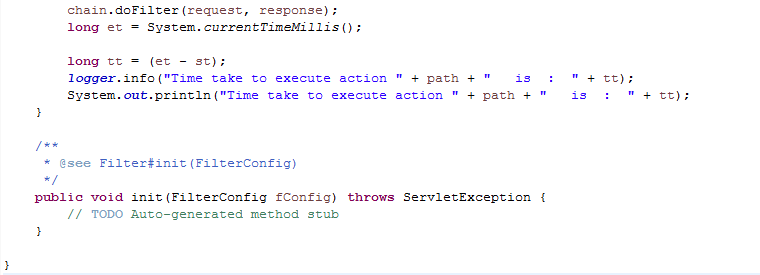
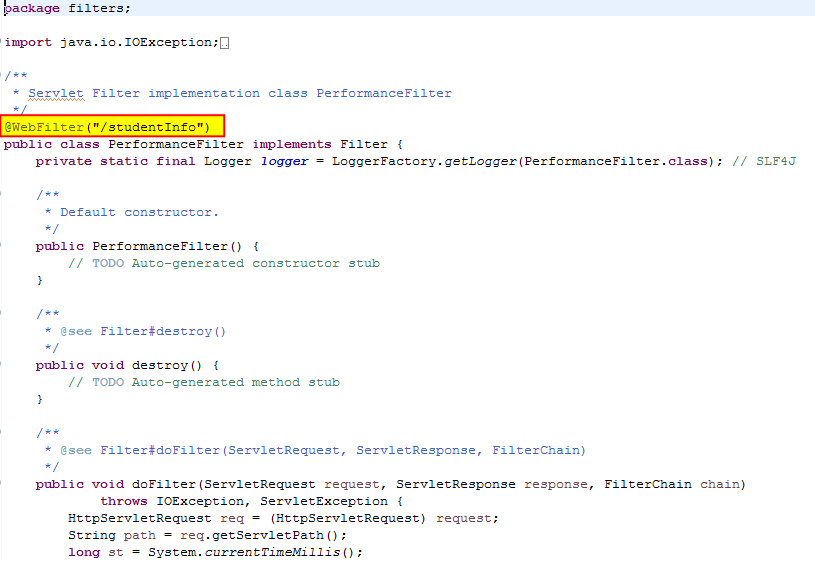
Controller act as a mediator between view and model. it is responsible to control the data transmission between the model and the view. It maps the user action into model updates.The controller layer is helpful to select the most appropriate view and delivers it to the user.



2. **dao** package containing Data Access Object class responsible for CRUD operations on underlying models classes(We have not given the screenshot of the file here as it is quite leanly.)

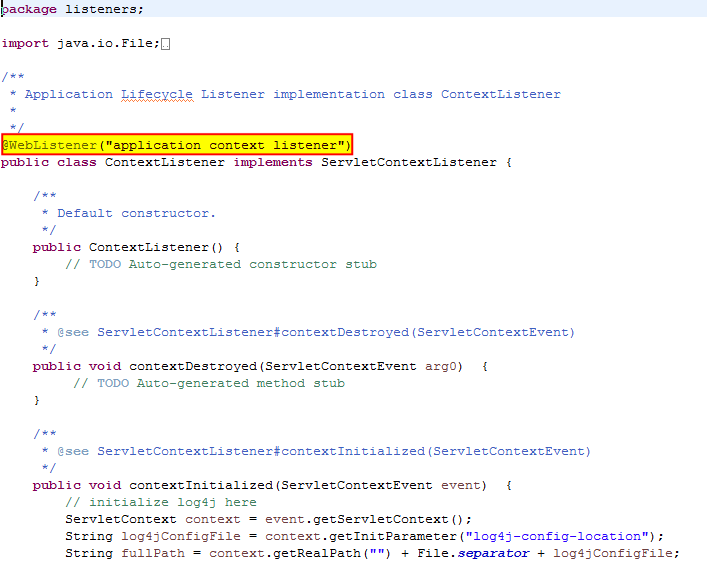
3. **databaseUtil** package for database configuration and utility class for database connection.

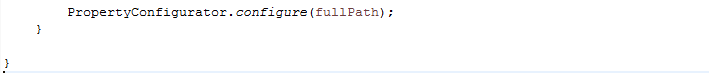
4. **filters** package responsible for containing PerformanceFilter.java file. This filter as the name suggest, responsible for calculating time between request and respective response. This class also contains logging operations.



\*\*Note the highlighted part in above code snippet. This is the annotation way to declare servlet filter.

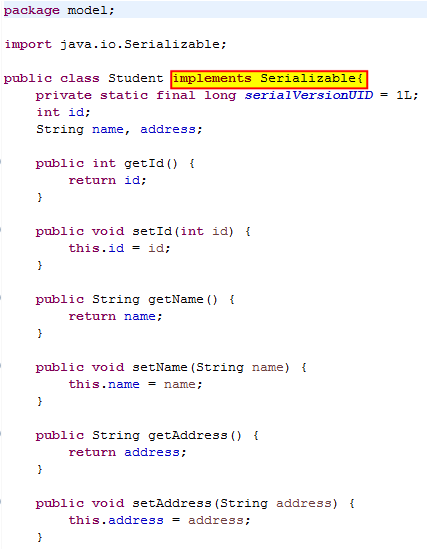
5. **listener** package containing class implementing Context Listener interface.

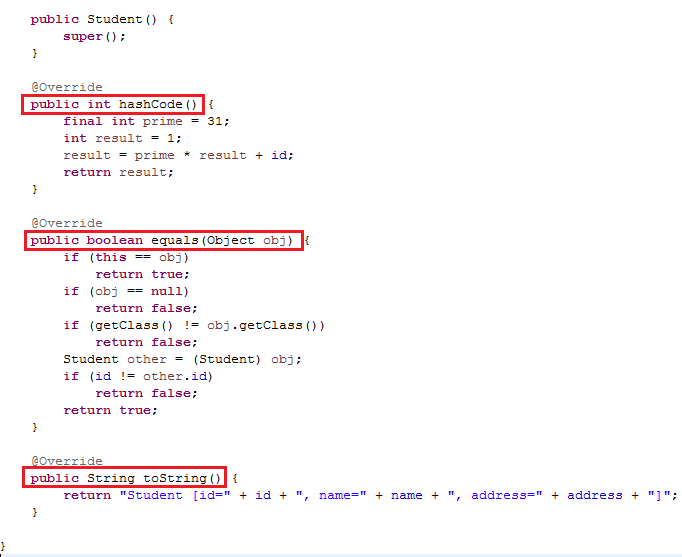




As this is annotation based project, please note the highlighted section of above code which is responsible for declaring the file as a filter in annotation way. This listener reads the logging file on application get initialized and set it to the application accordingly.

6. **model** package contains model classes.





**\*\*\*\*VVIMP: Before moving further, please go through package structure of this project along with use of each package very carefully. Just mug up it very well.**

**We are going to use almost same package structure in all above frameworks project viz. Struts2 and Spring.**

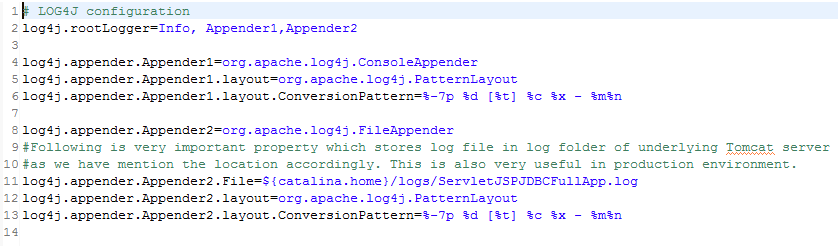
**How to create this project??**

a) This is maven web project.

(Please refer 'CreateMavenWebProjectWithEclipse.docx' file to get idea about creating such project.)

b) Add require maven dependencies

c) Create and place packages and files in the respective locations and shown. Pay special attention to log4j.xml and jsp folder in src/main/webapp folder.



**Files require more attention**

1. How we form package structure

2. Location and details in log4.xml file

**3. various files in jsp folder**

4. Files in controller package to study annotations we used instead of xml mapping in web.xml and to study the handling of the request.

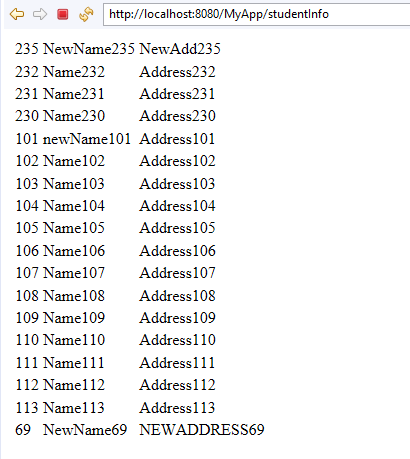
**To run the application:**

Right click on the project and Run As-> Run on server

When the application runs it will look as below:



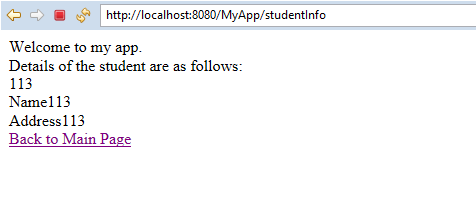
Click 'Get List" button, output is as below



Click back arrow at top left.



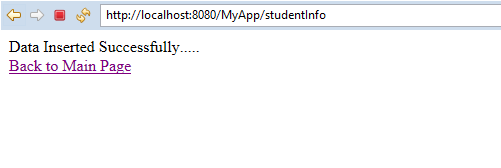
Now enter the given value and click 'Get Info' button. It will shows following details.



Now click 'Back on Main Page' link

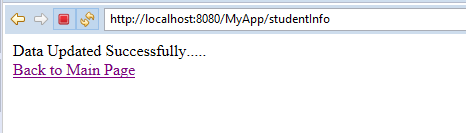
Now enter the values as below to add new entry and click 'Insert' button.





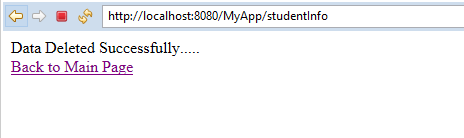
Now click 'Back to Main Page' link.

Now make change in the values of recently created record and click 'Update' button.



Click 'Back to Main Page' link.

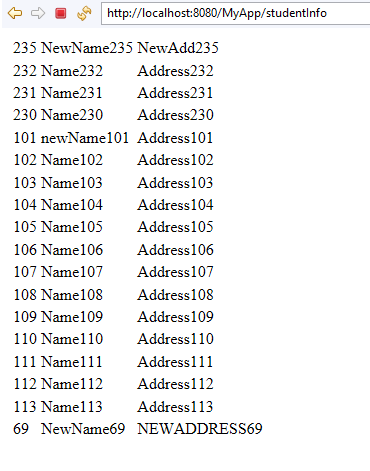
Enter primary key value and click 'Delete' button. It shows following screens:



Click 'Back to Main Page' link

Cross verify whether the record is deleted by clicking 'Get List' button:





Look here is no record with id as '114'.

Also go through console output as follows:

**Connected to the database**

**Creating statement...**

**Non Empty list**

**INFO 2020-09-10 16:04:44,295 [http-nio-8080-exec-5] filters.PerformanceFilter - Time take to execute action /studentInfo is : 1821**

**Time take to execute action /studentInfo is : 1821**

**Connected to the database**

**Creating statement...**

**Student [id=113, name=Name113, address=Address113]**

**INFO 2020-09-10 16:05:33,668 [http-nio-8080-exec-8] filters.PerformanceFilter - Time take to execute action /studentInfo is : 479**

**Time take to execute action /studentInfo is : 479**

**Insert button pressed.....**

**Connected to the database**

**Creating statement...**

**Data inserted successfully...........**

**INFO 2020-09-10 16:08:04,779 [http-nio-8080-exec-10] filters.PerformanceFilter - Time take to execute action /studentInfo is : 2396**

**Time take to execute action /studentInfo is : 2396**

**Update button pressed.....**

**Connected to the database**

**Creating statement...**

**Data updated successfully...........**

**INFO 2020-09-10 16:09:05,861 [http-nio-8080-exec-3] filters.PerformanceFilter - Time take to execute action /studentInfo is : 172**

**Time take to execute action /studentInfo is : 172**

**Delete button pressed.....**

**Connected to the database**

**Creating statement...**

**Data deleted successfully...........**

**INFO 2020-09-10 16:09:28,337 [http-nio-8080-exec-5] filters.PerformanceFilter - Time take to execute action /studentInfo is : 81**

**Time take to execute action /studentInfo is : 81**

**Connected to the database**

**Creating statement...**

**Non Empty list**

**INFO 2020-09-10 16:10:02,033 [http-nio-8080-exec-7] filters.PerformanceFilter - Time take to execute action /studentInfo is : 79**

**Time take to execute action /studentInfo is : 79**

**Connected to the database**

**Creating statement...**

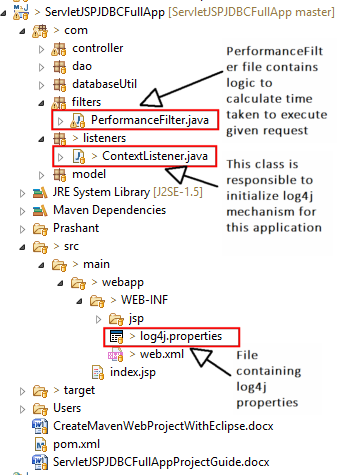
**Student [id=0, name=null, address=null]**

**INFO 2020-09-10 16:11:27,563 [http-nio-8080-exec-10] filters.PerformanceFilter - Time take to execute action /studentInfo is : 71**

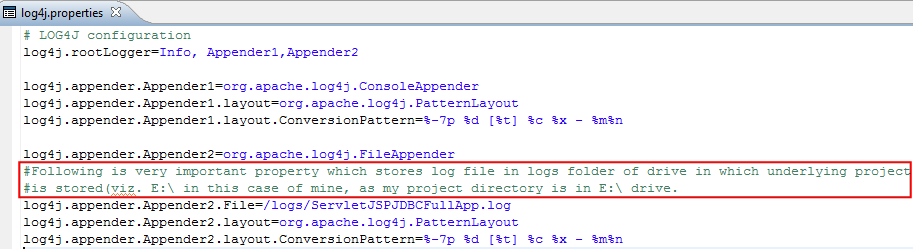
**Time take to execute action /studentInfo is : 71**

**Note the logging mechanism in this application:**

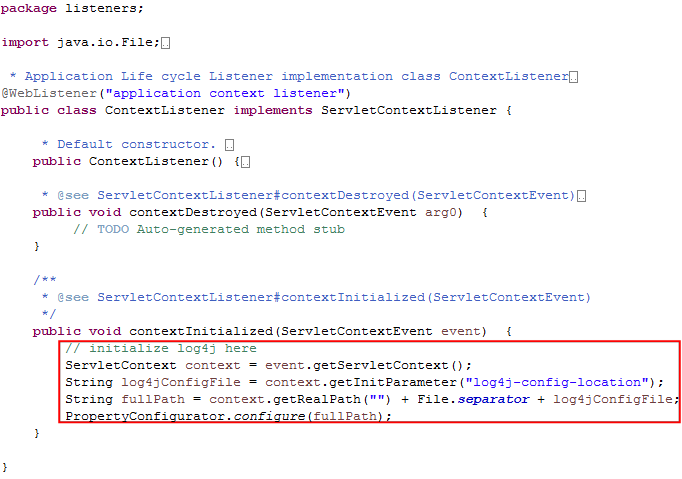
Various files involved in the logging mechanism are as follows:



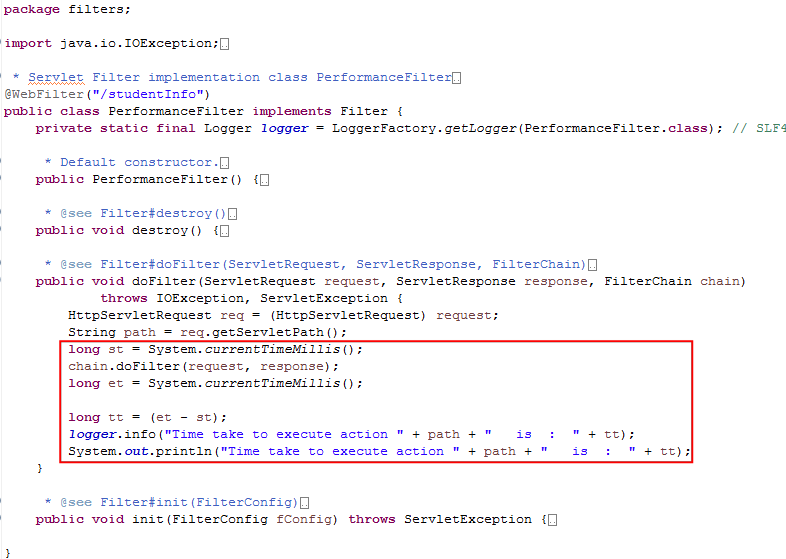
1. Log4J.properties file



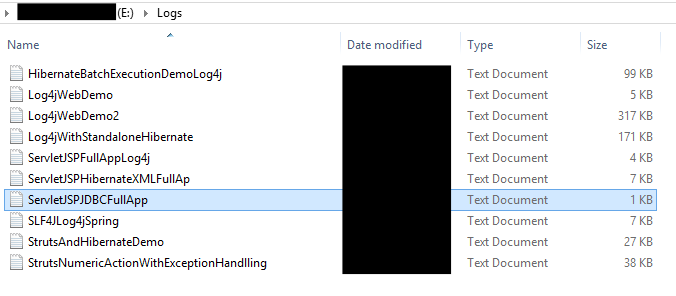
2. listeners.ContextListener class containing logic to configure log4j mechanism for this application:



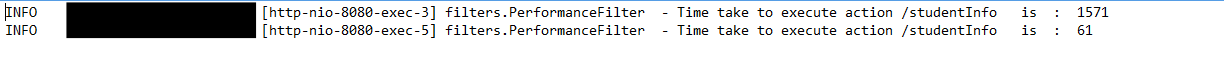
3. filters.PerformanceFilter class which contains logic to calculate time taken to execute each request:



4. Location of created log file in the system is as follows:



5. Contents of the above file are as below:



**\*\*\*Please note all above logging mechanism very carefully to understand role of logging in live project.**

Thus in this application we have covered almost all concepts in servlets and JSP technology viz. MVC design pattern, servlet as controller, filters, listeners, model, error handling using JSP, database handling using JDBC, logging mechanism etc.