***BEANS***

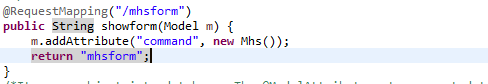
Beans di dalam aplikasi ini saya fungsikan sebagai wadah untuk menampung class controller,Dao(data access object) dan model yang natinya digunakan di spring-servlet yang berperan sebagai frontController/DispacherServlet.

Ini adalah DispacherServlet saya yang di berinama spring-servlet.



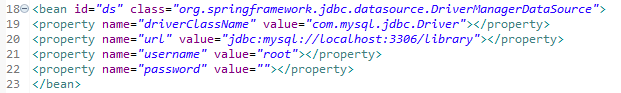


Bean pertama digunakan untuk InternalResourceViewResolver.  
berfungsi sebagai alamat akses directory project ketika akan dideploy ke web browser atau tomcat.  
jadi ketika akan membuat annotation request mapping developer tidak pelu mendekalrasikan fully qualified name dari file Jspnya.

Pengaplikasiannya seperti ini.  


@RequestMapping (“/mhsform”) akan mengakses file mhsform.jsp dari folder project dikarenakan di dalam frontController saya menggunakan InternalResourceViewResolver yang yang perlu menyebutkan “/mhsform” tanpa perlu menyebutkan fully qualified namenya yaitu “WEB-INF/jsp/mhsform.jsp” karena sudah disebutkan di value prefix dan suffinx diatas , jadi WEB-INF/jsp dan .jsp tidak perlu disebutkan lagi annotasi @RequestMapping.

Deklarasi Beans kedua dengan class org.springframework.jdbc.datasource.DriverManagerDataSource.  
digunakan untuk mendefinisikan koneksi database dari Jdbc ke Spring.  
driverClassName yang memiliki value library jdbc driver.  
url memiliki value lokasi database di mysql.  
kemudian username dan password dari mysql.



Beans ke tiga digunakan untuk menampung JdbcTemplate



Di dalam project ini saya menggunakan class bernama JdbcTemplate and dan dideklarasikan sebagai objek “template”, berfungsi untuk menghindari error yang biasanya terjadi kelita akan execute query sql atau melakukan update database, dilakukan melalui metode Resultset yang akan melakukan trows SqlException.

*“In simple way, JdbcTemplate is a bridge for spring to interact with database with given sql sintaks using ResultSet.”*



Bean ke empat digunakan untuk menampung class MhsDao, sperti screen capture di atas , class ini digunakan untuk menampung sintaks SQL yang akan digunak untuk JdbcTemplate agar bisa berinteraksi dengan mysql.



Kesimpulan dari saya setelah praktek menggunakan beans, beans berfungsi untuk enkapsulasi berbagi macam class yang akan digunakan di frontController atau DispacherServlet, dalam kasus ini saya gunakan spring-servlet.xml sebagai DispacherServlet/forntController project spring web mvc saya.

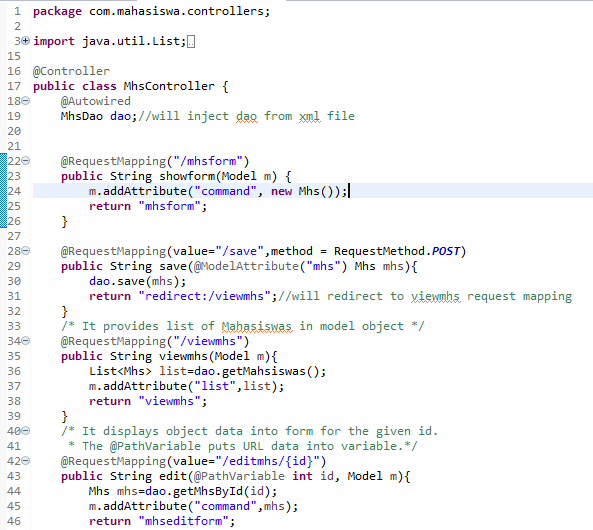
**Dependency Injection**

Dependency Injection (DI) adalah sebuah design pattern yang memungkinkan kita untuk menulis kode yang lebih baik, dimana DI menitikberatkan pada penulisan kode yang loosely coupled, highly modular dimana ketergantungan antar class tidaklah tinggi, sehingga kita dapat mengubah sebuah class tanpa diliputi rasa khawatir class tersebut membuat break class-class lain yang bergantung pada class yang kita rubah tadi.

Dari pemahaman saya Dependency Injection adalah proses injeksi object dari class satu ke class lainnya.  
Tapa perlu takut class yang diInject atau melakukan Injection akan rusak.  
dalam kasus ini saying membuat class bernama MhsDao, yang mana object dari class Mhs Dao akan di inject ke class MhsController.

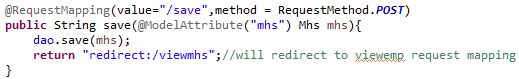


Di class MhsDao terdapat object save, update, delete, getMhsByIdm, getMahasiswas.  
masing-masing dari object tersebut memiliki sintak SQL yang nantinya akan diInject ke MhsController.



Di dalam MhsController diatas class MhsDao di Inject ke MhsController dengan diberikan value “dao”  
“dao” ini digunakan sesaui dengan tiap-tiap object yang ada di MhsDao.

Sebagai contoh pada object save dibawah.



Dideklarasikan “dao.save(mhs)” yang mana berfungsi untuk menyimpan data dari web ke database.  
kenapa bisa begitu, karena dari object “dao” sudah memberikan ijeksi ke object save yang ada di MhsContoller dari MhsDao. Injeksi tersebut berisi objek.  


Yang mana ijeksi tersebut mempunyai sintaks sql insert yang berfungsi intuk menyisipkan data ke database.  
  
Kesimpulan saya Dependency Injection adalah proses injectsi dari sebuah objek dari suatu class ke class lainnya, memberikan class yang sudah diinjectsi mempunyai kemampuan yang sama dari class penginjectnya.  
dan membuat class yang sudah diiject bergantung kepada class penginjectnya, seusai dengan namanya Dependency Injection.

<https://github.com/nhixs/Nusacamp/tree/master/training_code_id/Week%205>