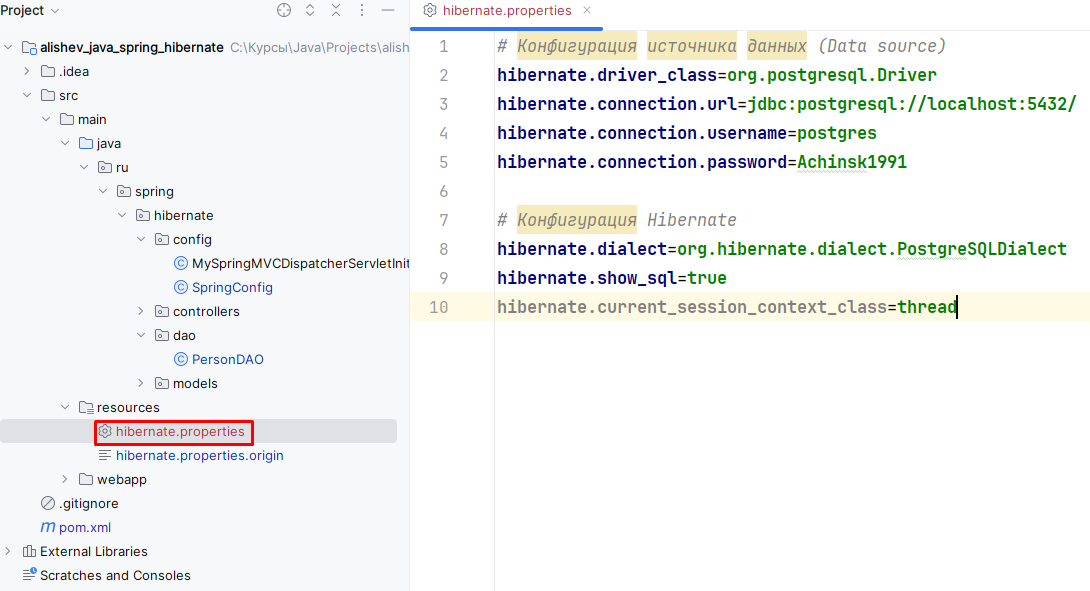
**Spring приложение с Hibernate**

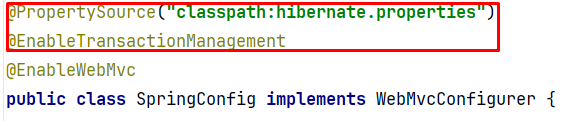
Чтобы в **Spring** приложение добавить поддержку **Hibernate** нужно:

**1.** Добавляем зависимости в pom.xml.

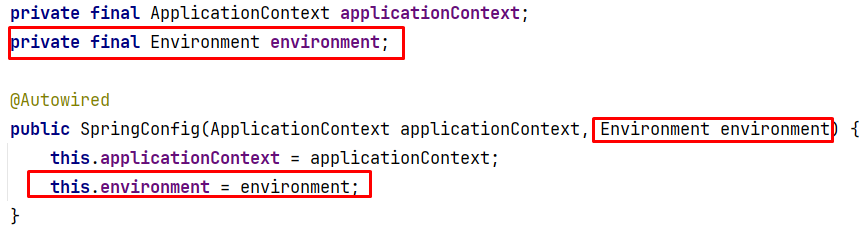
  
  
  
  
  
  
  
  
  
  
  
  
  
  
**2.** Создадим папку **resources** в которой будет лежать наш файл с конфигурацией **Hibernate**.

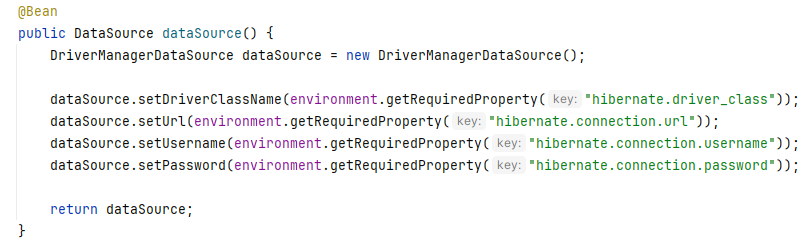
**3.** В **SpringConfig** прописать следующие аннотации:  
**@PropertySource** – здесь указываем путь до файла с конфигруацией **Hibernate** для того, чтобы использовать этот файл в этом классе.

**@EnableTransactionManagement –** эта аннотация нужно, чтобы транзакции управлялись Спрингом. Теперь не нужно будет самостоятельно начинать и заканчивать транзакции, теперь Спринг это будет делать за нас.



**4.** В классе **SpringConfig** нужно добавить еще одно поле **Envirment** и внедрить его с помощью **@Autowired.** С помощью этого мы сможем использовать конфигурацию **hibernate.properties** в нашем классе **SpringConfig**.



**5.** В методе **d*ataSource()*** используем наш **Envirment**, чтобы получить настройки конфигурации из файла **hibernate.properties**.  
  
  
  
  
  
  
  
  
  
  
  
**6.** В **SpringConfig** добавляем еще три метода:

***hibernateProperties()*** - нужен для того, чтобы описать дополнительные настройки конфигурации **hibernate**.

***sessionFactory()*** - здесь **Spring** создает **sessionFactory**. В методе ***setPackagesToScan()*** указываем пакет в котором будут хранится наши сущности.

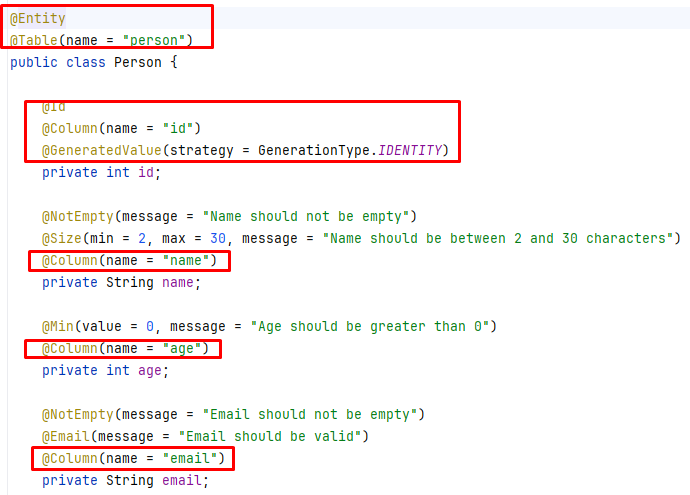
***hibernateTransactionManager()*** - этот бин будет говорить **спрингу** как работать с транзакциями, теперь нам не нужно бдует самостоятельно открывать и закрывать транзакции.

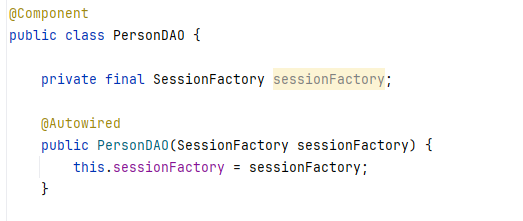


На этом настройка **Hibernate** в **Spring** приложении завершена. Можно приступать к работе.

**Пример использования Hibernate в Spring приложении**

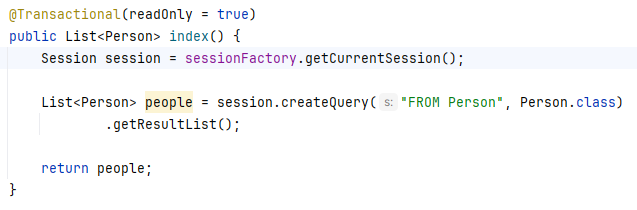
**1.** Сделаем из обычных классов сущности, чтобы связать её с базой данных.



**2.** Внедрим в **DAO** классы **sessionFactory**.  


**3.** В **Spring** приложении транзакции мы не открываем самостоятельно, тоесть не нужно вызывать метод ***beginTransaction()*** как это делается в чистом **hibernate**.

Чтобы начать транзакцию внутри какого-то метода нужно использовать специальную аннотацию которая называется **@Transactional**. Тоесть всё что будет происходит в этом методе будет присходить внутри транзакции и как только в этом методе все строчки будут пройдены транзакция автоматически будет закомченна(на сессии будет вызван метод **commit()**).

У аннотации **@Transactional** есть аргумент **readOnly.** Если в методе мы только читаем данные из базы данных, то этот аругмент следует поставить в положение **true**. Это может ускорить работы с базой данных.