Hibernate Configuration to work with H2 Database

1. Open Backend project and create a package inside com.health.HealthMedicineQuestBackEnd named “config”.
2. Create a class inside the config package named “HibernateConfig”.
3. Annotate the HibernateConfig class with @Configuration, @ConponentScan and @EnableTransactionManagement.
4. Create five static final variable to declare database related information.

//Database related information

**public** **static** **final** String ***DATABASE\_URL***="jdbc:h2:tcp://localhost/~/Health2";

**public** **static** **final** String ***DATABASE\_DRIVER***="org.h2.Driver";

**public** **static** **final** String ***DATABASE\_DIALECT***="org.hibernate.dialect.H2Dialect";

**public** **static** **final** String ***DATABASE\_USERNAME***="sa2";

**public** **static** **final** String ***DATABASE\_PASSWORD***="";

1. Create 3 Beans
   1. DataSource- providing the connection information of the database

@Bean

**private** DataSource getDataSource(){

BasicDataSource dataSource=**new** BasicDataSource();

dataSource.setDriverClassName(***DATABASE\_DRIVER***);

dataSource.setUrl(***DATABASE\_URL***);

dataSource.setUsername(***DATABASE\_USERNAME***);

dataSource.setPassword(***DATABASE\_PASSWORD***);

//datasource bean will be available

**return** dataSource;

}

* 1. SessionFactory – Configues application to use hibernate and create a session object

//sessionFactory will be available

@Bean

**public** SessionFactory getSessionFactory(DataSource dataSource){

LocalSessionFactoryBuilder builder= **new** LocalSessionFactoryBuilder(dataSource);

builder.addProperties(getHibernateProperties());

builder.scanPackages("com.health.HealthMedicineQuestBackEnd.model");

**return** builder.buildSessionFactory();

}

//All the hibernate properties will be returned by this method

**private** Properties getHibernateProperties(){

Properties prop=**new** Properties();

prop.put("hibernate.dialect", ***DATABASE\_DIALECT***);

prop.put("hibernate.show\_sql", “**true”**);

prop.put("hibernate.format\_sql", “**true”**);

**return** prop;

}

* 1. HibernateTransactionManager – to manage the hibernate transaction

//Transaction Management

@Bean

**public** HibernateTransactionManager getTransactionManager(SessionFactory sessionFactory){

HibernateTransactionManager transactionManager=**new** HibernateTransactionManager(sessionFactory);

**return** transactionManager;

}

NOTE: import all packages with respect to the dependency were selected.