hello_world / 2017-11-22 18:49:00 / 浏览数 2323 技术文章 技术文章 顶(1) 踩(0)

SessionFactory configuration.buildSessionFactory()

■■SessionFactory■■

Session sessionFactory.openSession()

```
这篇文章仅用于补充:JAVA代码审计的一些Tips(附脚本)一文中关于SQLi中不足部分
JDBC常用API
DriverManager:用于管理JDBC驱动的服务类。主要功能是获取Connection对象
public static Connection getConnection(String url, String user, String password) throws SQLException
  //=====url=====
Connection: 代表数据库连接对象。每一个Connection代表一个物理连接会话
Statement createStatement() throws SOLException;
     //
  PreparedStatement prepareStatement(String sgl)throws SOLException;
     //
  CallableStatement prepareCall(String sql) throws SQLException;
     // CallableStatement CallableStatement
  Savepoint setSavepoint() throws SQLException
  Savepoint setSavepoint(String name) throws SQLException;
     //
  void setTransactionIsolation(int level) throws SOLException;
     //
  void rollback() throws SOLException;
  void rollback(Savepoint savepoint) throws SQLException;
     //
  void setAutoCommit(boolean autoCommit) throws SOLException;
  void commit() throws SQLException;
Statement :用于执行SQL语句的工具接口。该对象既可以执行DDL,DCL语句,也可以执行DML语句 ,还可以用于执行SQL查询
ResultSet executeQuery(String sql) throws SQLException;
     //
  int executeUpdate(String sql) throws SQLException;
     //BBBBBDMLBBBBBBBBBBBBBBBBBBDDLBBBBDDLBBBBDDLBBBBDDL
  boolean execute(String sql) throws SOLException;
     PreparedStatement
: 预编译的Statement对象,它允许数据库预编译sql语句,以后每次只改变sql命令的参数,避免数据库每次都需要编译sql语句,无需再传入sql语句,
它比Statement多了以下方法
void setXxx(int parameterIndex, Xxx value):
     //===============================sql=======
Hibernate框架常用API
Configuration : 负责Hibernate的配置信息。包括运行的底层信息:数据库的URL、用户名、密码、JDBC驱动类,数据库Dialect,数据库连接池等
和持久化类与数据表的映射关系(*.hbm.xml文件)
//■■■■hibernate.properties■:
  Configuration cfg = new Configuration();
  //Xml■■■hibernate.cfg.xml■
  Configuration cfg = new Configuration().configure();
SessionFactory
:Configuration对象根据当前的配置信息生成SessionFactory对象,SessionFactory对象中保存了当前数据库的配置信息和所有映射关系以及预定义的SQL语句,同时还负
```

// SOL

```
Session
:是应用程序与数据库之间交互操作的单线程对象。session对象有一个一级缓存,显式执行flush之前,所有的持久层操作的数据都缓存在session对象处。相当于JDBC的C
//■■■■■■■■■get()■load()
  public Object get(Class clazz, Serializable id);
  public Object get(Class clazz, Serializable id, LockOptions lockOptions);
     //
  public Object get(String entityName, Serializable id);
     //
  public Object get(String entityName, Serializable id, LockOptions lockOptions);
     // -----
     //load() Execution
  public void load(Object object, Serializable id);
     //BBBBBBBBBBgetBBBBnullBloadBBBBBBrg.hibernate.ObjectNotFoundException
  Serializable save(Object object)
     //
  void update(Object object)
    void delete(Object object)
    void saveOrUpdate(Object object)
    //WIID save update id update, id save
  Query createQuery(String hgl)
    //MHHOL
  SQLQuery createSQLQuery(String sql)
    //EESOL
  Transaction beginTransaction()
    //
Transaction: 具有数据库事务的概念,所有持久层都应该在事务管理下进行,即使是只读操作
void commit()
    //
  void rollback()
    //
  boolean wasCommitted()
     //
用于从数据存储源查询对象及控制执行查询的过程,Query对象包装了一个HQL查询语句。Query对象在session对象关闭之前有效,否则会抛出Sessionexception异常
Query setxxx()
    //
  List list()
    //
  Obect uniqueResult()
    int executeUpdate()
     //
Criteria:
SpringJdbc常用API
JdbcTemplate: Spring对JDBC最低级别的封装,其他的工作模式事实上在底层使用了JdbcTemplate作为其底层的实现基础
void execute(String sql)
    //
  int update(String sql)
    //
  int[] batchUpdate(String sql)
  int[] batchUpdate(String sql,BatchPreparedStatementSetter pss)
```

```
queryForxxxxx(String sal)
      //
  Map<String,Object> call(CallableStatementCreator csc, List<SqlParameter> declaredParameters) throws DataAccessException
      //
NamedParameterJdbcTemplate : 对JdbcTemplate做了封装,提供了更加便捷的基于命名参数的使用方式
String sql = "INSERT INTO student(id,student_name,email,Dept_No) VALUES(:id,:name,:email,:deptid)";
      Map<String , Object> paramMap = new HashMap<>();
      paramMap.put("id",9);
      paramMap.put("name","Limbo");
      paramMap.put("email","1610770854@qq.com");
      paramMap.put("deptid",1);
      namedParameterJdbcTemplate.update(sgl,paramMap);
Mybatis常用API
SqlSessionFactory : 是单个数据库映射关系经过编译后的内存镜像,其主要作用是创建SqlSession对象
InputStream inputStream = Resources.getResourceAsStream("mybatis-config.xml");
  SqlSessionFactory sqlSessionFactory = new SqlSessionFactoryBuilder().build(inputStream);
  SqlSession openSession()
  SqlSession openSession(Connection connection)
      ■■SqlSessioon■■
SqlSession : 应用程序与持久层之间执行交互操作的一个单线程对象,其主要作用是执行持久化操作
void commit()
      //
  int delete(String statement)
  int delete(String statement, Object parameter)
      //
  int insert(String statement)
  int insert(String statement, Object parameter)
      //
  <T> T selectOne(String statement)
  <T> T selectOne(String statement, Object parameter)
      //
  <E> List<E> selectList(String statement)
  <E> List<E> selectList(String statement, Object parameter)
      //
  <T> T getMapper(Class<T> type)
      //IIIIIMapper
Spring data JPA
Spring Data JPA 简化持久层开发大致需要如下三个步骤
public interface UserRepository extends Repository<User, Long> { }
在持久层的接口中声明需要的业务方法,Spring Data JPA将会根据指定的策略为该方法生成实现代码。用户不需要实现该接口
List<User> findByLastname(String lastname);
在Spring的配置文件中添加配置,为声明的接口设定代理对象
<jpa:repositories base-package="com.zhutougg.jpa" entity-manager-factory-ref="entityManagerFactory" transaction-manager-ref="t</pre>
获得并使用repository的实例
//■Spring Container■■■
public class TestJPA {
  @Autowired
  private UserRepository repository;
  public void doSomething() {
      User u = new User();
      User user = repository. save (u);
```

}

//■Spring Container■■■

RepositoryFactorySupport factory = ... // Instantiate factory here
UserRepository repository = factory.getRepository(UserRepository.class)

Respository : 是SpringData的核心接口 , 并不提供任何方法 , 用户需要自己定义需要的方法

- 1. public interface UserDao extends Repository<User, Long> { }
- 2. @RepositoryDefinition(domainClass = User.class, idClass = Long.class)
 public interface UserDao { }

点击收藏 | 0 关注 | 1

上一篇: JAVA代码审计的一些Tips(附脚本) 下一篇: LCTF 2017 三道Web题的...

1. 3条回复



hello_world 2017-11-22 18:50:28

Heibernate框架的Criteria类没用过,所以暂时留空

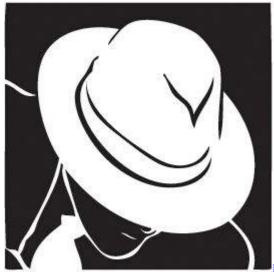
0 回复Ta



cryin 2017-11-23 11:08:59

赞

0 回复Ta



藏青 2017-11-24 12:33:16

期待大佬下次更新

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS <u>关于社区</u> 友情链接 社区小黑板