jdbcTemplate 教学

## 1、spring 配置文件生成jdbcTemplate

|  |
| --- |
| <bean id=*"jdbcTemplate"* class=*"org.springframework.jdbc.core.JdbcTemplate"*>  <property name=*"dataSource"* ref=*"dataSource"*/>  </bean> |

## 2、编辑接口类

|  |
| --- |
| **public** **interface** ISequenceDao {  /\*\*  \* 获取批次号  \*/  **public** **abstract** String getBatchNo();  /\*\*  \* 获取交易序列号  \*/  **public** **abstract** String getAppseriono();    /\*\*  \* 获取系统公共流水号  \*/  **public** **abstract** String getSerialNo();  **public** **abstract** <T> T get(String sequence, Class<T> clazz);    /\*\*  \* 获取缴费申请编号  \*/  **public** **abstract** String getContriApplyNo();  /\*\*  \* 获取缴费申请编号  \*/  **public** **abstract** String getInsnId();    /\*\*  \* 获取交易流水号  \*/  **public** **abstract** String getRefNo();    /\*\*  \* 获取文件从处理流水号  \*/  **public** **abstract** String getFlowNo();  **public** **abstract** String getappno();      **public** **abstract** List<PayReceiptDetialEO> getBySql();  **public** **abstract** PayReceiptDetialEO getByObjectSql();    } |

## 3、实现类

|  |
| --- |
| **public** **class** SequenceDaoImp **implements** ISequenceDao {    @Autowired  **private** JdbcTemplate jdbcTemplate;    @Override  **public** String getBatchNo() {  **return** get("batchno\_sequence", String.**class**);  }    @Override  **public** String getAppseriono() {  **return** jdbcTemplate.queryForObject  ("select lpad(appseriono\_sequence.nextval, 20, '0') from dual", String.**class**);  }    @Override  **public** String getSerialNo() {  **return** get("serial\_no\_sequence", String.**class**);  }    @Override  **public** <T> T get(String sequence, Class<T> clazz) {  String sql = "select " + sequence + ".nextval from dual";  **return** jdbcTemplate.queryForObject(sql, clazz);  }  @Override  **public** String getContriApplyNo() {  **return** jdbcTemplate.queryForObject  ("select 'JF'||lpad(contriapplyno\_sequence.nextval, 6, '0') from dual", String.**class**);  }  @Override  **public** String getInsnId() {  **return** jdbcTemplate.queryForObject  ("select lpad(APPSERIONO\_CCB\_SEQUENCE.nextval, 6, '0') from dual", String.**class**);  }  @Override  **public** String getRefNo() {    **return** jdbcTemplate.queryForObject  (" select to\_char(sysdate,'yyyymmdd')||lpad(Ref\_NO\_SEQUENCE.Nextval,6,'0') from dual", String.**class**);  }  @Override  **public** String getFlowNo() {  **return** jdbcTemplate.queryForObject(" select 'w'||to\_char(sysdate,'yyyymmdd')||lpad(Flow\_NO\_SEQUENCE.Nextval,7,'0') from dual", String.**class**);  }  @Override  **public** String getappno() {    **return** jdbcTemplate.queryForObject(" SELECT lpad(APP\_NO\_SEQUENCE.Nextval,9,'0') FROM dual", String.**class**);  }  @Override  **public** List<PayReceiptDetialEO> getBySql() {  String sql =  "SELECT cp.ssid AS ssid," +  " cp.planid AS planid," +  " cp.num AS NUM," +  " cp.personid AS personid," +  " cp.ssnum AS ssnum," +  " cp.idtype AS idtype," +  " cp.idno AS idno," +  " cp.staffname AS staffname," +  " cp.benebankid AS bankType," +  " cp.beneBankBranchId AS bankId," +  " cp.accno AS accno," +  " cp.accname AS accname," +  " cp.PAYSUMUNTAX AS PAYSUMUNTAX," +  " cp.PAYTAX AS PAYTAX," +  " cp.payaftersum AS payAmt," +  " cp.RETRYFLAG AS RETRYFLAG,\n" + //失败重发标识  " cp.benetype AS benetype," +  " cp.memo AS memo," +  " case cp.ssid when '4124768' then '0' else'1' end as flag, "+  " case cp.ssid when '4124768' then '3001' else '0000' end as retCode,"+  " to\_char(sysdate,'yyyyMMdd') as drawDate,"+  " to\_char(sysdate,'HH24mmss') as drawTime"+  " FROM CSIP\_OA\_PAYLISTINFO CPI" +  " JOIN CSIP\_OA\_TASKFILE CT" +  " ON CPI.PK\_PAYLISTINFO = CT.PK\_PAYLISTINFO" +  " JOIN CSIP\_OA\_PAYLIST CP" +  " ON CP.PK\_TASKFILE = CT.PK\_TASKFILE" +  " WHERE CPI.INNER\_LISTNO = 'w201704210000704'" ;    List<PayReceiptDetialEO> rows = jdbcTemplate.query(sql,**new** BeanPropertyRowMapper<PayReceiptDetialEO>(PayReceiptDetialEO.**class**));    **return** rows;  }  @Override  **public** PayReceiptDetialEO getByObjectSql() {  String sql =  "SELECT \* from CSIP\_OA\_PAYLISTINFO cop where cop.INNER\_LISTNO = 'w201704210000704'" ;    PayReceiptDetialEO payReceiptDetialEO = jdbcTemplate.queryForObject(sql, **new** BeanPropertyRowMapper<PayReceiptDetialEO>(PayReceiptDetialEO.**class**));      **return** payReceiptDetialEO;  }  } |

## 4、获取对象

|  |
| --- |
| @Override  **public** PayReceiptDetialEO getByObjectSql() {  String sql = "SELECT \* from CSIP\_OA\_PAYLISTINFO cop where cop.INNER\_LISTNO = 'w201704210000704'" ;    PayReceiptDetialEO payReceiptDetialEO = jdbcTemplate.queryForObject(sql, **new** BeanPropertyRowMapper<PayReceiptDetialEO>(PayReceiptDetialEO.**class**));      **return** payReceiptDetialEO;  } |