My Open Source Projects : How to use

This page last changed on gen 17, 2008 by admin.

```
CREATE TABLE CUSTOMER (
        FIRST_NAME VARCHAR(30) NOT NULL,
        LAST_NAME VARCHAR(30) NOT NULL,
        ID INTEGER NOT NULL CONSTRAINT EMP_NO_PK PRIMARY KEY
        )
```

```
public class Customer {

        private String firstName;
        private String lastName;
        private int id;

        public final String getFirstName() {
                return firstName;
        }
        public final void setFirstName(String firstName) {
                this.firstName = firstName;
        }
        public final String getLastName() {
                return lastName;
        }
        public final void setLastName(String lastName) {
                this.lastName = lastName;
        }
        public final int getId() {
                return id;
        }
        public final void setId(int id) {
                this.id = id;
        }

}
```

```
public class CustomerBeanInfo extends AbstractManagedBeanInfo<Customer> {


        public CustomerBeanInfo() {
                setBeanClass(Customer.class);
        }

        public BeanDescriptor getBeanDescriptor() {
                return new BeanDescriptorEntity(getBeanClass())
                        .setEntityName("Customer");
        }

        public PropertyDescriptor[] getPropertyDescriptors() {
                try {
                        return new PropertyDescriptor[] {

                                        new PropertyDescriptorPK("id", getBeanClass(), "getId",
"setId" ),
                                        new PropertyDescriptorField("firstName",
getBeanClass(), "getFirstName", "setFirstName" )
                                                .setFieldName("FIRST_NAME"),
                                        new PropertyDescriptorField("lastName", getBeanClass(),
"getLastName", "setLastName" )
                                                .setFieldName("LAST_NAME"),
                        };
                } catch (IntrospectionException e) {
                        e.printStackTrace();

                }

                return EMPTY_PROPERTYDESCRIPTOR;
        }
```

```
}
```

```
public class TestCustomerManager {
        private Connection conn = null;
        private BeanManager<Customer> manager = null;

        @BeforeClass
        public static void loadDriver() throws Exception {

            String driver = "org.apache.derby.jdbc.EmbeddedDriver";
            Class.forName(driver);

            LogManager.getLogManager().readConfiguration(
TestBeanManager.class.getResourceAsStream("/logging.properties"));

        }

        @Before
        @SuppressWarnings("unchecked")
        public void connect() throws SQLException {

            String connectionURL = "jdbc:derby:./target/DerbyDB/testDB";

            Properties p = new Properties();
            p.setProperty("create", "true");

            conn = DriverManager.getConnection(connectionURL, p);

            manager = (BeanManager<Customer>)
BeanManagerFactory.getFactory().createBeanManager(Customer.class);

        }

        @After
        public void disconnect() throws SQLException {
                manager = null;

                if( null!=conn ) {
                        conn.close();
                }
        }

        @Test
        public void createBean() throws SQLException {

                Customer bean = new Customer();

                bean.setId(1);
                bean.setFirstName("name1");
                bean.setLastName("sname1");

                manager.create(conn, bean);


        }


        @Test
        public void findBeanById() throws SQLException {

                Customer bean = manager.findById(conn, 1);

                Assert.assertNotNull("Customer retreived is null", bean );
                Assert.assertEquals( "Customer.id doesn't match", bean.getId(), 1 );


        }

        @Test
        public void updateBeanInclude() throws SQLException {

                Customer bean = new Customer();

                bean.setId(1);
                bean.setFirstName("bartolomeo");
```

```
                manager.store(conn, bean, true, "firstName");

                bean = manager.findById(conn, 1);

                Assert.assertNotNull("Customer retreived is null", bean );
                Assert.assertEquals( "Customer.id doesn't match", bean.getId(), 1 );
                Assert.assertEquals( "Customer.firstName doesn't match", bean.getFirstName(),
"bartolomeo" );


        }

        @Test
        public void updateBeanExclude() throws SQLException {

                Customer bean = new Customer();

                bean.setId(1);
                bean.setLastName("sorrentino");

                manager.store(conn, bean, false, "firstName");

                bean = manager.findById(conn, 1);

                Assert.assertNotNull("Customer retreived is null", bean );
                Assert.assertEquals( "Customer.id doesn't match", bean.getId(), 1 );
                Assert.assertEquals( "Customer.lastName doesn't match", bean.getLastName(),
"sorrentino" );


        }

        @Test
        public void updateBean() throws SQLException {

                Customer bean = manager.findById(conn, 1);

                Assert.assertNotNull("Customer retreived is null", bean );
                Assert.assertEquals( "Customer.id doesn't match", bean.getId(), 1 );

                bean.setFirstName("bartolo");

                manager.store(conn, bean);

                bean = manager.findById(conn, 1);

                Assert.assertNotNull("Customer retreived is null", bean );
                Assert.assertEquals( "Customer.id doesn't match", bean.getId(), 1 );
                Assert.assertEquals( "Customer.firstName doesn't match", bean.getFirstName(),
"bartolo" );


        }

        @Test
        public void findBeans() throws SQLException {
                List<Customer> customers = new ArrayList<Customer>();

                manager.find(conn, customers, "${firstName} LIKE ? OR ${lastName}=? ORDER BY
#{lastName}", "b%", "sorrentino");

                for( Customer c: customers ) {

                        System.out.printf( "Customer name=%s, surname=%s, id=%d\n",
c.getFirstName(), c.getLastName(), c.getId());
                }

        }

        @Test
        public void removeBean() throws SQLException {

                manager.removeById(conn, 1);


        }
```

```
}
```