



Employee.java

```
package com.test.domain;
```

```
public class Employee  
{
```

```
    private String empId;  
    private String empName;
```

```
    public String getEmpId() {  
        return empId;  
    }
```

```
    public void setEmpId(String empId) {  
        this.empId = empId;  
    }
```

```
    public String getEmpName() {
        return empName;
    }

    public void setEmpName(String empName) {
        this.empName = empName;
    }

    @Override
    public String toString() {
        return "Employee [empId=" + empId + ",
empName=" + empName + "]";
    }
}
```

CamelMain.java

```
package com.test.main;

import java.util.List;

import org.apache.camel.CamelContext;
import org.apache.camel.ProducerTemplate;
import
org.springframework.context.ApplicationContext;
import
org.springframework.context.support.ClassPathXmlAppli
cationContext;

import com.test.domain.Employee;

public class CamelMain {
```

```
public static void main(String[] args) {

    try {
        @SuppressWarnings("resource")
        ApplicationContext springCtx = new
ClassPathXmlApplicationContext("/application.xml");
        System.out.println("spring context loaded");
        CamelContext context =
springCtx.getBean("employeeContext",
CamelContext.class);

        context.start();

        ProducerTemplate producerTemplate =
context.createProducerTemplate();
        System.out.println("Context created
");

        // Insert Employee 1
        Employee emp1 = getEmployee1();
        String resp =
producerTemplate.requestBody("direct:insert", emp1,
String.class);
        System.out.println("Get Employee1
details ");

        // Insert Employee 2
        Employee emp2 = getEmployee2();
        resp =
producerTemplate.requestBody("direct:insert", emp2,
String.class);
        System.out.println("Get Employee2
details ");
```

```
        // Get Employee of inserted employees
        List<Employee> employees =
producerTemplate.requestBody("direct:select", null,
List.class);
        System.out.println("employees:" +
employees);

    }
    catch (Exception e) {

        e.printStackTrace();

    }
}

private static Employee getEmployee1() {

    Employee emp = new Employee();

    emp.setEmpId("empId1");
    emp.setEmpName("emp1");
    return emp;

}

private static Employee getEmployee2() {

    Employee emp = new Employee();

    emp.setEmpId("empId2");
    emp.setEmpName("emp2");
    return emp;

}
```

```
    }  
EmployeeRouter.java  
  
package com.test.router;  
  
import org.apache.camel.builder.RouteBuilder;  
  
public class EmployeeRouter extends RouteBuilder {  
  
    @SuppressWarnings("deprecation")  
    @Override  
    public void configure() throws Exception {  
  
        from("direct:insert").log("Inserted new  
Employee").beanRef("employeeMapper", "getMap")  
            .to("sqlComponent:{{sql.insertEmployee}}"  
);  
  
        from("direct:select").to("sqlComponent:{{sql.getAllEm  
ployees}}")  
            .beanRef("employeeMapper",  
"readEmployees").log("${body}");  
    }  
}
```

EmployeeMapper.java

```
package com.test.util;  
  
import java.util.ArrayList;  
import java.util.HashMap;  
import java.util.List;
```

```
import java.util.Map;

import com.test.domain.Employee;

public class EmployeeMapper {

    public Map<String, Object> getMap(Employee
employee) {
        Map<String, Object> answer = new
HashMap<String, Object>();
        answer.put("EmpId", employee.getEmpId());
        answer.put("EmpName", employee.getEmpName());
        return answer;
    }

    public List<Employee>
readEmployees(List<Map<String, String>> dataList) {

        System.out.println("data:" + dataList);

        List<Employee> employees = new
ArrayList<Employee>();

        for (Map<String, String> data : dataList) {

            Employee employee = new Employee();

            employee.setEmpId(data.get("EmpId"));
            employee.setEmpName(data.get("EmpName"));

            employees.add(employee);
        }

        return employees;
    }
}
```

```
    }  
}  
Application.xml  
<?xml version="1.0" encoding="UTF-8"?>  
<beans  
  xmlns="http://www.springframework.org/schema/beans"  
    xmlns:jdbc="http://www.springframework.org/schema/  
jdbc"  
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-  
instance"  
    xmlns:context="http://www.springframework.org/sch  
ema/context"  
    xmlns:camel="http://camel.apache.org/schema/sprin  
g"  
    xsi:schemaLocation="  
      http://www.springframework.org/schema/beans  
  
http://www.springframework.org/schema/beans/spring-  
beans.xsd  
      http://www.springframework.org/schema/jdbc  
  
http://www.springframework.org/schema/jdbc/spring-  
jdbc-3.0.xsd  
      http://camel.apache.org/schema/spring  
      http://camel.apache.org/schema/spring/camel-  
spring.xsd  
      http://www.springframework.org/schema/context  
  
http://www.springframework.org/schema/context/spring-  
context.xsd ">  
  
  <bean id="dataSource"  
    class="org.apache.commons.dbcp.BasicDataSource"  
      destroy-method="close">
```

```
        <property name="driverClassName"
value="com.mysql.jdbc.Driver" />
        <property name="url"
value="jdbc:mysql://localhost:3306/test" />
        <property name="username" value="root"/>
        <property name="password" value="root"/>
        <property name="initialSize" value="2" />
        <property name="maxActive" value="5" />
    </bean>

    <jdbc:initialize-database data-
source="dataSource"
        enabled="true">
        <jdbc:script location="classpath:db-
schema.sql" />
    </jdbc:initialize-database>

    <bean id="sqlComponent"
class="org.apache.camel.component.sql.SqlComponent">
        <property name="dataSource" ref="dataSource"
/>
    </bean>

    <bean id="employeeMapper"
class="com.test.util.EmployeeMapper" />
    <bean id="employeeRouter"
class="com.test.router.EmployeeRouter" />

    <camel:camelContext id="employeeContext"
xmlns="http://camel.apache.org/schema/spring">

        <propertyPlaceholder id="placeholder"
location="classpath:sql.properties" />
        <routeBuilder ref="employeeRouter" />
    </camel:camelContext>
```

```
</camel:camelContext>
```

```
</beans>
```

```
Db-schema.sql
```

```
CREATE TABLE employs3 (  
    empId VARCHAR(10) NOT NULL,  
    empName VARCHAR(100) NOT NULL  
);
```

```
Sql.properties
```

```
sql.insertEmployee=INSERT INTO employs3(EmpId,  
EmpName) VALUES (:#EmpId, :#EmpName)
```

```
sql.getAllEmployees=select * from employs3
```

```
pom.xml
```

```
<project xmlns="http://maven.apache.org/POM/4.0.0"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
    xsi:schemaLocation="http://maven.apache.org/POM/4  
.0.0
```

```
https://maven.apache.org/xsd/maven-4.0.0.xsd">
```

```
    <modelVersion>4.0.0</modelVersion>  
    <groupId>com.test.database</groupId>  
    <artifactId>camel-sqltest</artifactId>  
    <version>0.0.1-SNAPSHOT</version>  
    <properties>
```

```
        <spring.version>4.1.6.RELEASE</spring.version>
```

```
<camelspring.version>2.16.0</camelspring.version>  
</properties>
```

```
<dependencies>  
  <dependency>  
    <groupId>org.apache.camel</groupId>  
    <artifactId>camel-core</artifactId>  
  
    <version>${camelspring.version}</version>  
  </dependency>  
  <dependency>  
    <groupId>org.apache.camel</groupId>  
    <artifactId>camel-cxf</artifactId>  
  
    <version>${camelspring.version}</version>  
  </dependency>  
  <dependency>  
    <groupId>org.apache.camel</groupId>  
    <artifactId>camel-spring</artifactId>  
  
    <version>${camelspring.version}</version>  
  </dependency>  
  <dependency>  
    <groupId>org.apache.camel</groupId>  
    <artifactId>camel-sql</artifactId>  
    <version>2.17.1</version>  
  </dependency>  
  
  <dependency>  
    <groupId>commons-dbcp</groupId>  
    <artifactId>commons-dbcp</artifactId>  
    <version>1.2.2</version>  
  </dependency>
```

```
<dependency>
  <groupId>commons-pool</groupId>
  <artifactId>commons-pool</artifactId>
  <version>1.6</version>
</dependency>

<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-core</artifactId>
  <version>${spring.version}</version>
</dependency>
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-
java</artifactId>
  <version>5.1.21</version>
</dependency>

</dependencies>
</project>
```
