实验十四　Servlet+Spring的应用

```
SpringBSDemo
      |
      |-------------tarena
                      |
                      |-------spring-config.xml
                      |
                      |-------DBBean
                      |
                      |-------JdbcTemplate
                      |
                      |-------DBServlet
```
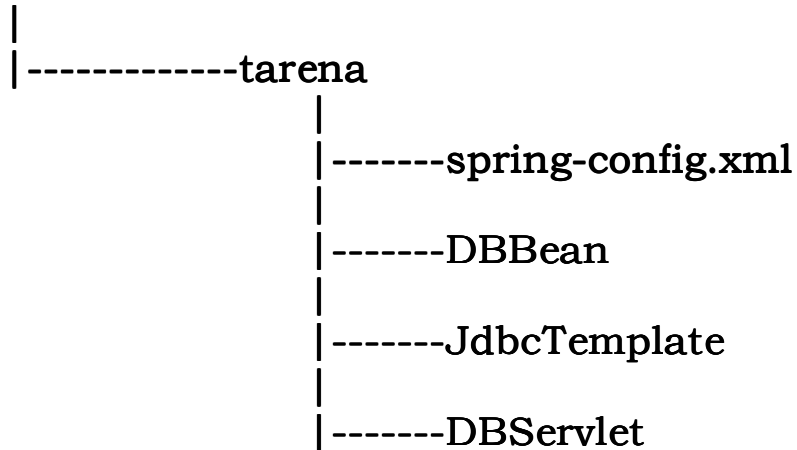
加入Spring2Core
加入Spring2Web
加入MySQL

步一，创建DBBean

```java
public class DBBean {

        private String driver;
        private String url;
        private String user;
        private String password;

        public DBBean() {
        }

        public String getDriver() {
                return driver;
        }

        public void setDriver(String driver) {
```

```java
                this.driver = driver;
        }

        public String getUrl() {
                return url;
        }

        public void setUrl(String url) {
                this.url = url;
        }

        public String getUser() {
                return user;
        }

        public void setUser(String user) {
                this.user = user;
        }

        public String getPassword() {
                return password;
        }

        public void setPassword(String password) {
                this.password = password;
        }

}
```

步二，创建JdbcTemplate
```java
public class JdbcTemplate {

        public static Connection getConnection(String driver, String url,
                                String user, String password) {
                Connection conn = null;
                try {
                        Class.forName(driver);
```

実験十四.txt

```
                conn = DriverManager.getConnection(url, user, password);
            } catch (Exception e) {
                e.printStackTrace();
            }
            return conn;
        }

        public static void close(Connection conn) {
            if (conn != null) {
                try {
                    conn.close();
                } catch (SQLException e) {
                    e.printStackTrace();
                }
            }
        }

        public static void close(PreparedStatement pstmt) {
            if (pstmt != null) {
                try {
                    pstmt.close();
                } catch (SQLException e) {
                    e.printStackTrace();
                }
            }
        }

        public static void close(ResultSet rs) {
            if (rs != null) {
                try {
                    rs.close();
                } catch (SQLException e) {
                    e.printStackTrace();
                }
            }
        }
```

実验十四.txt
}

步三，修改spring-config.xml文件
```xml
<?xml version="1.0" encoding="UTF-8"?>
<beans
xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-2.0.xsd">


        <!-- 注册DBBean -->
        <bean id="db" class="tarena.DBBean">
                <property name="driver">

<value>com.mysql.jdbc.Driver</value>
                </property>
                <property name="url">

<value>jdbc:mysql://localhost:3306/test</value>
                </property>
                <property name="user">
                        <value>root</value>
                </property>
                <property name="password">
                        <value>admin</value>
                </property>
        </bean>

</beans>
```

步四，修改web.xml文件(重点)
```xml
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5"
```

实验十四.txt

```xml
xmlns="http://java.sun.com/xml/ns/javaee"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://java.sun.com/xml/ns/javaee

http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">


        <!-- 加载spring-config.xml文件 -->
        <context-param>

<param-name>contextConfigLocation</param-name>
                <param-value>

/WEB-INF/classes/tarena/spring-config.xml
                </param-value>
        </context-param>



        <!-- spring提供的Servlet
                <servlet>
                <servlet-name>context</servlet-name>
                <servlet-class>

org.springframework.web.context.ContextLoaderServlet
                </servlet-class>
                <load-on-startup>1</load-on-startup>
                </servlet>
        -->


        <!-- spring提供的Listener -->
        <listener>
                <listener-class>

org.springframework.web.context.ContextLoaderListener
```

```
            </listener-class>
        </listener>



        <servlet>
            <servlet-name>DBServlet</servlet-name>
<servlet-class>tarena.DBServlet</servlet-class>
            <load-on-startup>1</load-on-startup>
        </servlet>

        <servlet-mapping>
            <servlet-name>DBServlet</servlet-name>
            <url-pattern>/db</url-pattern>
        </servlet-mapping>
</web-app>
```

步五，创建DBServlet

```
public class DBServlet extends HttpServlet {

        private Connection conn;

        public void destroy() {
                super.destroy();
                JdbcTemplate.close(conn);
        }

        public void doGet(HttpServletRequest request,
HttpServletResponse response)
                                throws ServletException,
IOException {

                System.out.println(conn!=null);

        }

        public void init(ServletConfig config) throws
```

实验十四.txt

ServletException {

```
//ApplicationContext context =
//           new ClassPathXmlApplicationContext(
//
"tarena/spring-config.xml");

        WebApplicationContext wac =

WebApplicationContextUtils.getWebApplicationContext(config.getServletContext());

        DBBean db = (DBBean)wac.getBean("db");

        String driver = db.getDriver();
        String url = db.getUrl();
        String user = db.getUser();
        String password = db.getPassword();

        conn =

JdbcTemplate.getConnection(driver, url, user, password);

        }

}
```

步六，结果
true（创建数据库连接）

实验十四.txt