**类描述**

单元测试类 indexControllerTest 使用 @RunWith(MockitoJUnitRunner.class) 注解，表示它使用 Mockito 框架进行单元测试。它包含对 indexController 的测试，主要测试了 get，add 和 delete 方法。通过模拟 VOService 和 IMapper，我们可以隔离测试对象的依赖，确保测试的独立性和可靠性。

**测试方法**

**1. testGetCZGLs**

**目的**

测试 indexController.getCZGLs 方法在获取列表成功时的行为。

**步骤**

1. **Arrange**：设置共享测试数据 mockCZGLList 和 mockCZGL。
2. **Mock**：
   * 设置 mapper.getCZGL() 方法返回 mockCZGLList。
   * 设置 mapper.getCZByCD(anyString()) 方法返回 "Mock Station Name"。
3. **Act**：调用 controller.getCZGLs() 方法。
4. **Assert**：
   * 断言返回的列表大小为 1。
   * 断言返回列表中第一个元素的 STNM 属性为 "Mock Station Name"。
   * 验证 mapper.getCZGL() 方法被调用一次。
   * 验证 mapper.getCZByCD(anyString()) 方法被调用一次。

**2. testGetCZGLEmpty**

**目的**

测试 indexController.getCZGLs 方法在获取空列表时的行为。

**步骤**

1. **Arrange**：无特殊数据设置。
2. **Mock**：
   * 设置 mapper.getCZGL() 方法返回空列表。
3. **Act**：调用 controller.getCZGLs() 方法。
4. **Assert**：
   * 断言返回的列表大小为 0。
   * 验证 mapper.getCZGL() 方法被调用一次。
   * 验证 mapper.getCZByCD(anyString()) 方法从未被调用。

**3. testAddSuccess**

**目的**

测试 indexController.add 方法在成功添加时的行为。

**步骤**

1. **Arrange**：创建一个 CZ 对象，并设置其属性。
2. **Mock**：设置 mapper 对象的 addCZ1, addCZ2, addCZ3, addCZ4 方法的返回值为 true。
3. **Act**：调用 controller.add(cz) 方法。
4. **Assert**：
   * 断言返回结果的代码为 "200"。
   * 验证 mapper 的 addCZ1, addCZ2, addCZ3, addCZ4 方法均被调用一次。

**4. testAddFailure**

**目的**

测试 indexController.add 方法在某次数据库操作失败时的行为。

**步骤**

1. **Arrange**：创建一个 CZ 对象，并设置其属性。
2. **Mock**：设置 mapper 对象的 addCZ2 方法的返回值为 false，其他方法返回 true。
3. **Act**：调用 controller.add(cz) 方法。
4. **Assert**：
   * 断言返回结果的代码为 "200"。
   * 验证 mapper 的 addCZ1, addCZ2, addCZ3, addCZ4 方法均被调用一次，即使其中一个方法返回 false。

**5. testDeleteSuccess**

**目的**

测试 indexController.delete 方法在成功删除时的行为。

**步骤**

1. **Arrange**：创建一个 testCode 字符串。
2. **Mock**：设置 mapper 对象的 delete1, delete2, delete3, delete4 方法的返回值为 1。
3. **Act**：调用 controller.delete(testCode) 方法。
4. **Assert**：
   * 验证 mapper 的 delete1, delete2, delete3, delete4 方法均被调用一次。
   * 断言返回结果的代码为 "200"。

**6. testDeleteFailure**

**目的**

测试 indexController.delete 方法在删除操作失败时的行为。

**步骤**

1. **Arrange**：创建一个 testCodeFailure 字符串。
2. **Mock**：设置 mapper 对象的 delete1, delete2, delete3, delete4 方法的返回值为 0。
3. **Act**：调用 controller.delete(testCodeFailure) 方法。
4. **Assert**：
   * 验证 mapper 的 delete1, delete2, delete3, delete4 方法均被调用一次。
   * 断言返回结果的代码为 "200"。