

# Integration testing exercises

---

The `EmployeeProjects` database is being enhanced with timesheet tracking. A new DAO has been created to handle creating, reading, updating, and deleting records from the `timesheet` table. For this exercise, you'll be responsible for writing integration tests and using them to identify bugs in this new DAO. You'll then fix the bugs you've found.

## Learning objectives

After completing this exercise, you'll understand:

- How to write integration tests.
- How to use integration tests to find bugs in a DAO.

## Evaluation criteria and functional requirements

Your code will be evaluated based on the following criteria:

- The project must not have any build errors.
- You must fill out the provided `BugReport.txt` file for four bugs you found and fixed.
- The provided integration test methods must all be completed and passing.
- Code is clean, concise, and readable.

## Getting started

1. You'll use the same `EmployeeProjects` database you used for the DAO exercises.
2. [Import](#) the Integration Testing exercises project into IntelliJ.

## Step One: Explore starting code

Before you begin, review the provided classes in the `model` and `dao` packages.

You should also familiarize yourself with the provided test classes and the `test-data.sql` file.

## Step Two: Implement the `JdbcTimesheetDaoTests` methods

In the nine test methods, replace the `Assert.fail()` with the code necessary to implement the test described by the method name. You can refer to the comments in the `TimesheetDao` interface for descriptions of what each DAO method does.

Use today's lecture code and the integration tests from the DAO exercises as examples to reference while working. Static constant `Timesheets` have been provided that you can use in your tests.

When fully implemented, five of the tests pass, and four continue to fail due to bugs in `JdbcTimesheetDao`.

## Step Three: Complete bug reports and fix bugs

Fill out `BugReport.txt` with information about the four bugs you've identified in `JdbcTimesheetDao` using the integration tests.

---

## An example of reporting and fixing a bug

Consider the following `deleteTimesheet` method:

```
public void deleteTimesheet(Long timesheetId) {  
    String sql = "DELETE FROM timesheet WHERE timesheet_id = ?";  
    jdbcTemplate.update(sql, 1);  
}
```

If the method was written this way, it would contain a bug. It always deletes the record with a `timesheet_id` of 1 rather than using the value of `timesheetId`.

There are several ways this could cause the `deleted_timesheet_cant_be_retrieved` test to fail—for example, if the test called `deleteTimesheet(2)` and then found that `getTimesheet(2)` still retrieved the timesheet.

After that test fails, you'd fix the `deleteTimesheet` method like this:

```
public void deleteTimesheet(Long timesheetId) {  
    String sql = "DELETE FROM timesheet WHERE timesheet_id = ?";  
    jdbcTemplate.update(sql, timesheetId);  
}
```

Then you'd complete the bug report as follows:

```
Test that demonstrates problem:  
    deleted_timesheet_cant_be_retrieved  
Expected output:  
    getTimesheet(2) returns null after calling deleteTimesheet(2)  
Actual output:  
    getTimesheet(2) was still returning a Timesheet object  
How did you fix this bug?  
    Replaced hardcoded value of 1 in deleteTimesheet with timesheetId so it  
    doesn't always delete the same timesheet.
```

---

After you've found and fixed the four bugs, all nine of the tests in `JdbcTimesheetDaoTests` pass.