WISEJ.NET 2.0 CONTROLS AND A NORTHWIND APPLICATION · MODULE 6 · BIND A DATAGRIDVIEW AND VALIDATE SHARED FILTERS

AI Coding Exercise

Use the course-grounded prompt, review the generated patch and record verification evidence.

LearnWisej

AI Coding Exercise

Implement module 6 as a bounded patch, then verify the result against the lab acceptance criteria.

Before prompting

Provide your current practice page and the included example files named below. Keep a copy of the prompt, response, reviewed diff and verification evidence. This exercise works with a coding assistant of your choice.

Ready-to-use prompt

```
Help me learn Wisej.NET 2.0 controls in C#. Explain the properties and event path before
implementing a change.
Module 6: Bind a DataGridView and Validate Shared Filters
Use the original files included in this course: code/NorthwindCourse/OrderData.cs,
code/NorthwindCourse/NorthwindPage.cs. Ask for their contents if unavailable.
Compatibility: .NET Framework 4.8, Wisej-2 2.0.54 and Wisej-2-NavigationBar 2.0.54. The web
application host comes from the installed Wisej 2 template. CollapsibleSideButton is
supplied course code; TagTextBox is the tagged editor; Switch is a CheckBox appearance.
Task: Bind the supplied orders to explicit DataGridView columns and prove that one filter
method supplies consistent rows counts totals and selected-record details.
Acceptance criteria:
1. Inspect the six original OrderRow records and calculate their baseline count and total.
2. Create explicit read-only grid columns with correct DataPropertyName values.
3. Connect tags switch and date inputs to OrderData.Filter through one refresh method.
4. Read the selected OrderRow from DataBoundItem with null handling.
5. Test combined filters inclusive boundaries an invalid date range and an empty result.
Initialize controls once, keep mutable data per Page, use decimal amounts and apply the
documented inclusive date rules. Explain the exact files to change. Give one small patch and
independent tests with expected results. Do not claim a command or browser check ran unless
it did.
Evidence: A model-to-column mapping table; A grid bound to the six-row baseline; Expected
and observed results for three combined-filter cases; Selected-record details identified by
OrderId; Data-test output and browser evidence for invalid and empty cases.
Review questions: Why is DataPropertyName different from HeaderText? Why should selection
use the bound record instead of a row position? Which checks require a browser as well as a
pure data test?
```

Review and verify

- ✓ Inspect the six original OrderRow records and calculate their baseline count and total.
- ✓ Create explicit read-only grid columns with correct DataPropertyName values.
- ✓ Connect tags switch and date inputs to OrderData.Filter through one refresh method.
- ✓ Read the selected OrderRow from DataBoundItem with null handling.
- ✓ Test combined filters inclusive boundaries an invalid date range and an empty result.

Challenge the generated result

- ✓ Why is DataPropertyName different from HeaderText?
- ✓ Why should selection use the bound record instead of a row position?
- ✓ Which checks require a browser as well as a pure data test?

Record your decision

For every proposed file, record accept, revise or reject with a reason. Record exact build output and observed browser results. A plausible answer is not evidence that the application ran.

Included course examples

`code/NorthwindCourse/OrderData.cs` · `code/NorthwindCourse/NorthwindPage.cs`

These original files are included in the code download. Build the exercise before comparing with the completed example.