WISEJ.NET 2.0 CONTROLS AND A NORTHWIND APPLICATION · MODULE 4 · TAGTEXTBOX FOR MULTIPLE FILTER VALUES

AI Coding Exercise

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

LearnWisej

AI Coding Exercise

Implement module 4 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 4: TagTextBox for Multiple Filter Values
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: Build a TagTextBox practice page that prints normalized country values and filters the
supplied six-order dataset predictably.
Acceptance criteria:
1. Add a TagTextBox with comma separators and a three-tag limit.
2. Log TextChanged and tag lifecycle events while adding and removing values.
3. Use OrderData.ParseCountries for trimming duplicate removal and case policy.
4. Apply the documented empty-filter and country-OR rules to the small dataset.
5. Verify one country multiple countries unknown input and clearing the last tag.
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 configured TagTextBox and explained event path; An event log showing addition
removal and rejection; Normalization examples with expected outputs; Filter counts for
Germany Germany plus UK and no tags; Clear empty-result feedback and a recovery action.
Review questions: How does a visual tag become a filter value? Which event owns the refresh?
What should clearing the final tag do?
```

Review and verify

- ✓ Add a TagTextBox with comma separators and a three-tag limit.

- ✓ Log TextChanged and tag lifecycle events while adding and removing values.
- ✓ Use OrderData.ParseCountries for trimming duplicate removal and case policy.
- ✓ Apply the documented empty-filter and country-OR rules to the small dataset.
- ✓ Verify one country multiple countries unknown input and clearing the last tag.

Challenge the generated result

- ✓ How does a visual tag become a filter value?
- ✓ Which event owns the refresh?
- ✓ What should clearing the final tag do?

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.