Adding Features in an ASP.NET Core App

Thomas Claudius Huber
SOFTWARE DEVELOPER

@thomasclaudiush www.thomasclaudiushuber.com

Module Outline

Test driven development of user interfaces

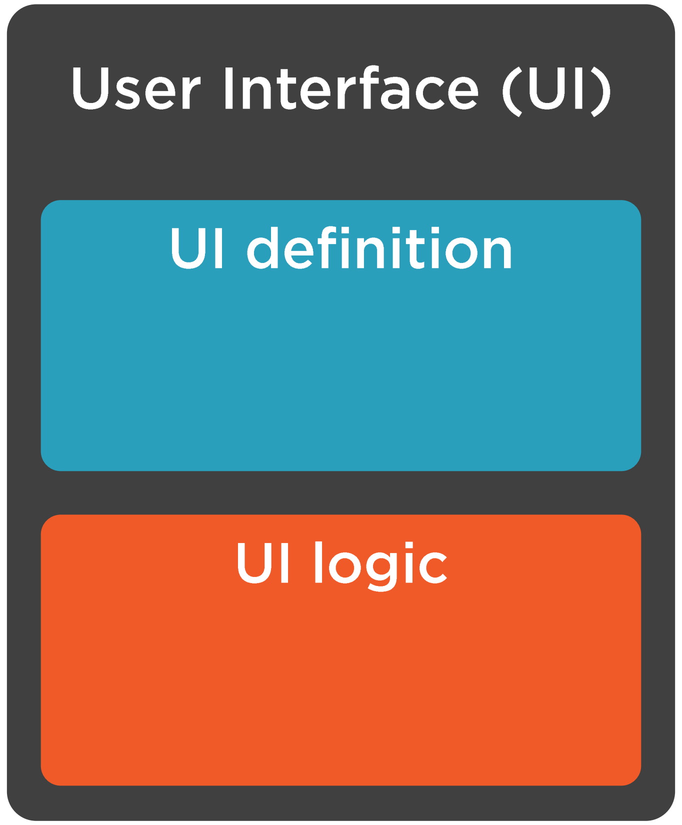
The DeskBooker solution

Understand the requirements

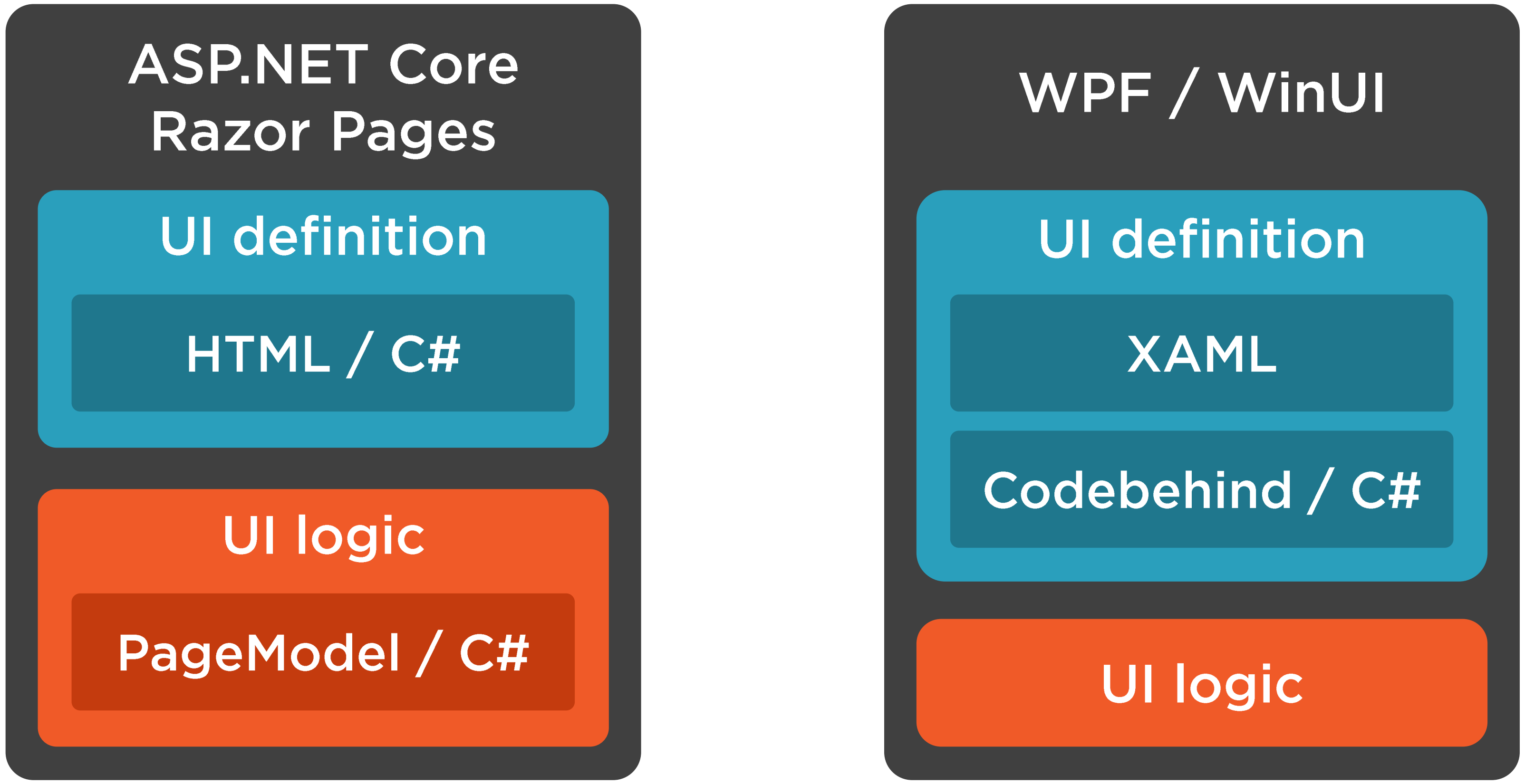
- Write the logic for the book desk page

Use TDD to implement the requirements

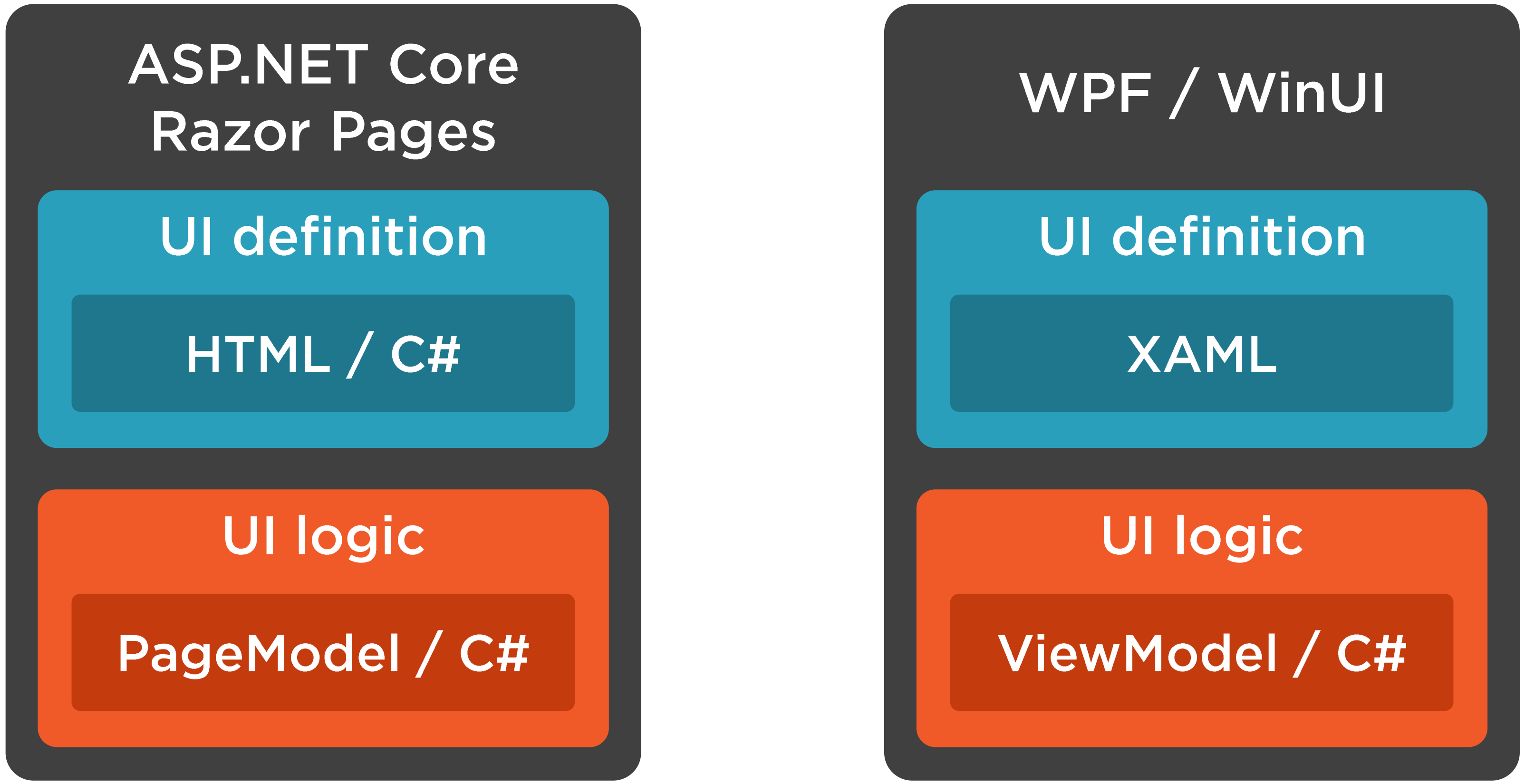
Test Driven Development of User Interfaces



Test Driven Development of User Interfaces



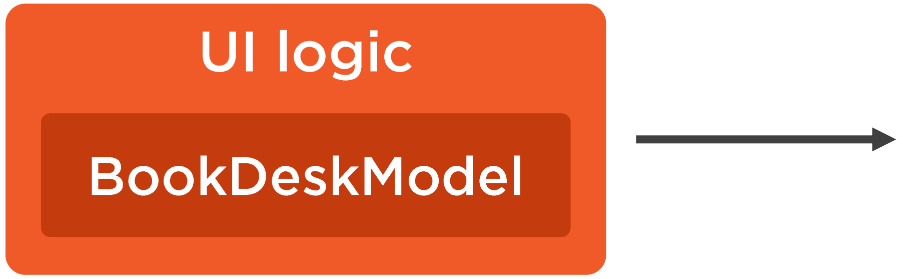
Test Driven Development of User Interfaces

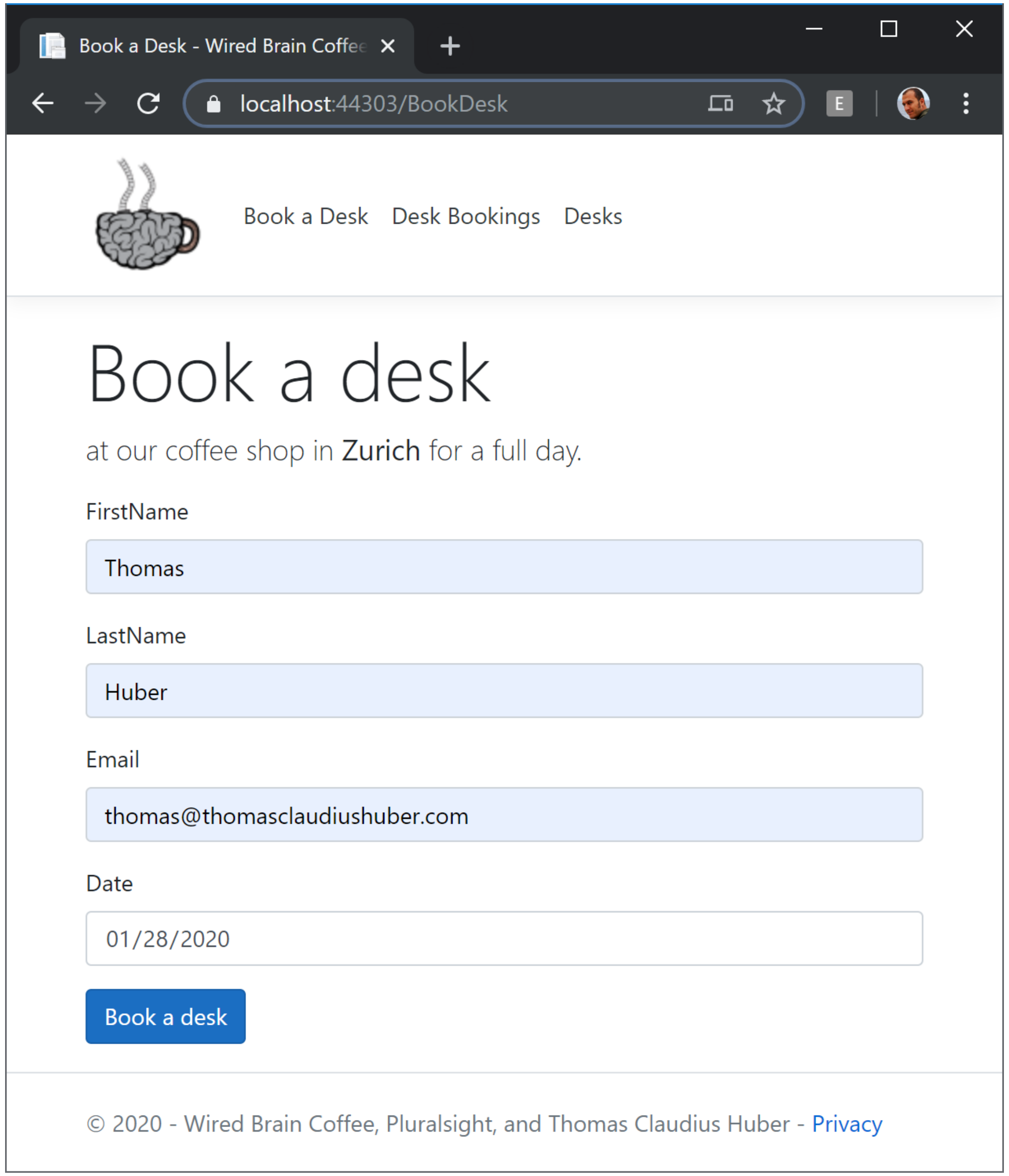


Model View ViewModel (MVVM)

Course:

WPF and MVVM:
Test Driven Development of
ViewModels





Book a desk

at our coffee shop in **Zurich** for a full day.

FirstName: Thomas

LastName: Huber

Email: thomas@thomasclaudiushuber.com

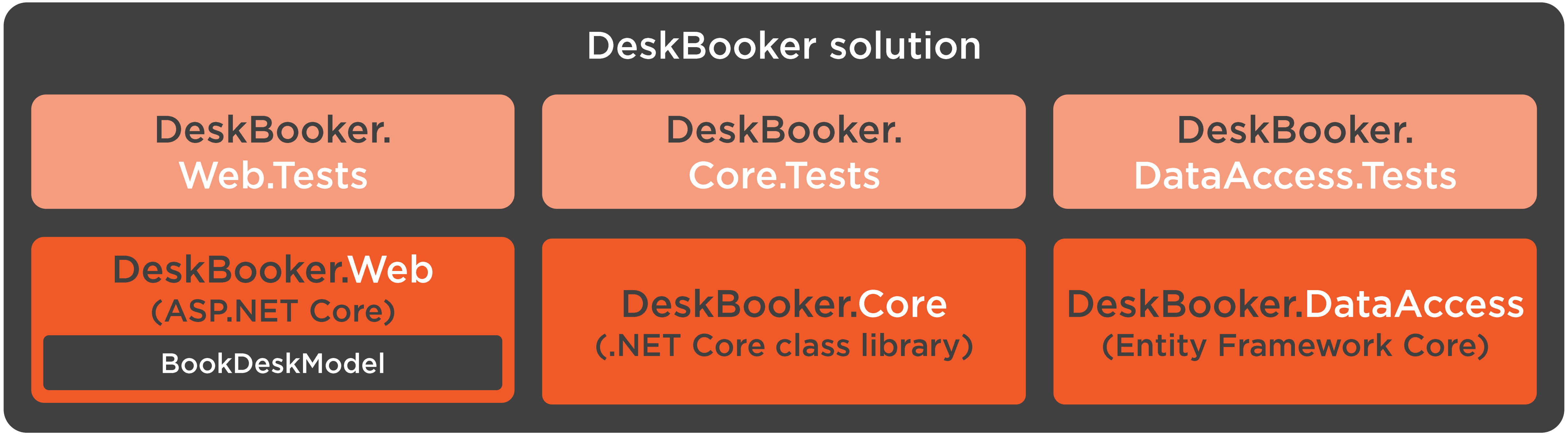
Date: 01/28/2020

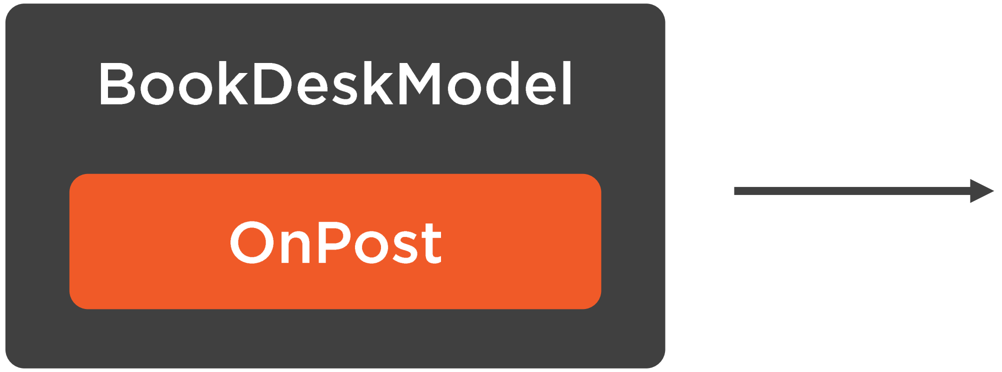
© 2020 - Wired Brain Coffee, Pluralsight, and Thomas Claudius Huber - Privacy

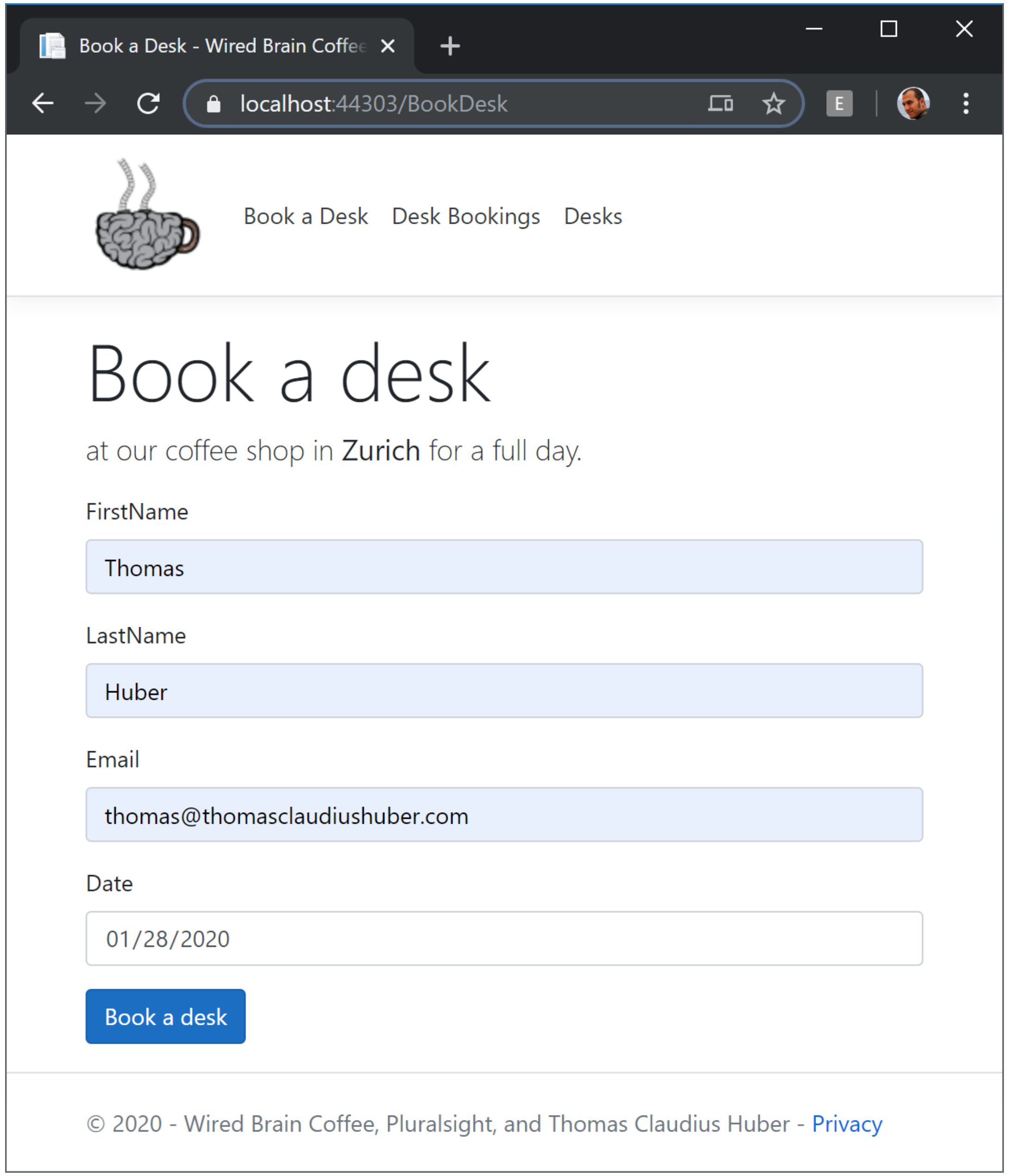
Demo

Explore the DeskBooker solution

Understand the Requirements


DeskBooker solution
DeskBooker.Web.Tests
DeskBooker.Core.Tests
DeskBooker.DataAccess.Tests
DeskBooker.Web (ASP.NET Core)
BookDeskModel
DeskBooker.Core (.NET Core class library)
DeskBooker.DataAccess (Entity Framework Core)




Book a Desk - Wired Brain Coffee

localhost:44303/BookDesk

Book a Desk Desk Bookings Desks

Book a desk

at our coffee shop in **Zurich** for a full day.

FirstName: Thomas

LastName: Huber

Email: thomas@thomasclaudiushuber.com

Date: 01/28/2020

Book a desk

© 2020 - Wired Brain Coffee, Pluralsight, and Thomas Claudius Huber - Privacy

Understand the Requirements

BookDeskModel

OnPost

Requirements

- Call the BookDesk method of the processor
- Check if the model is valid
- Add a model error if no desk is available
- Return the expected IActionResult
- Redirect to the BookDeskConfirmation page

Demo

Call the BookDesk method of the DeskBookingRequestProcessor

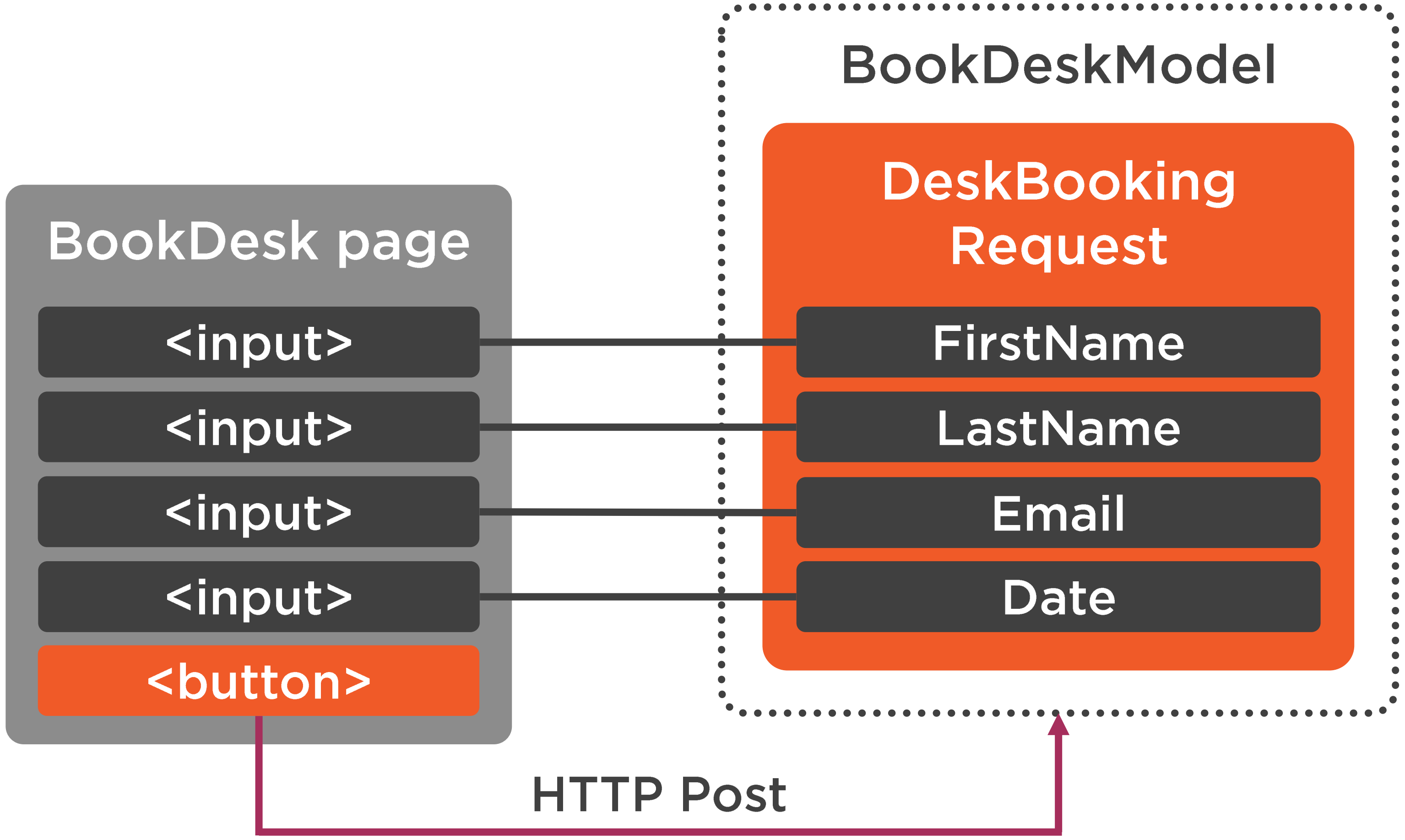
- Write a test
- Implement the logic

Demo

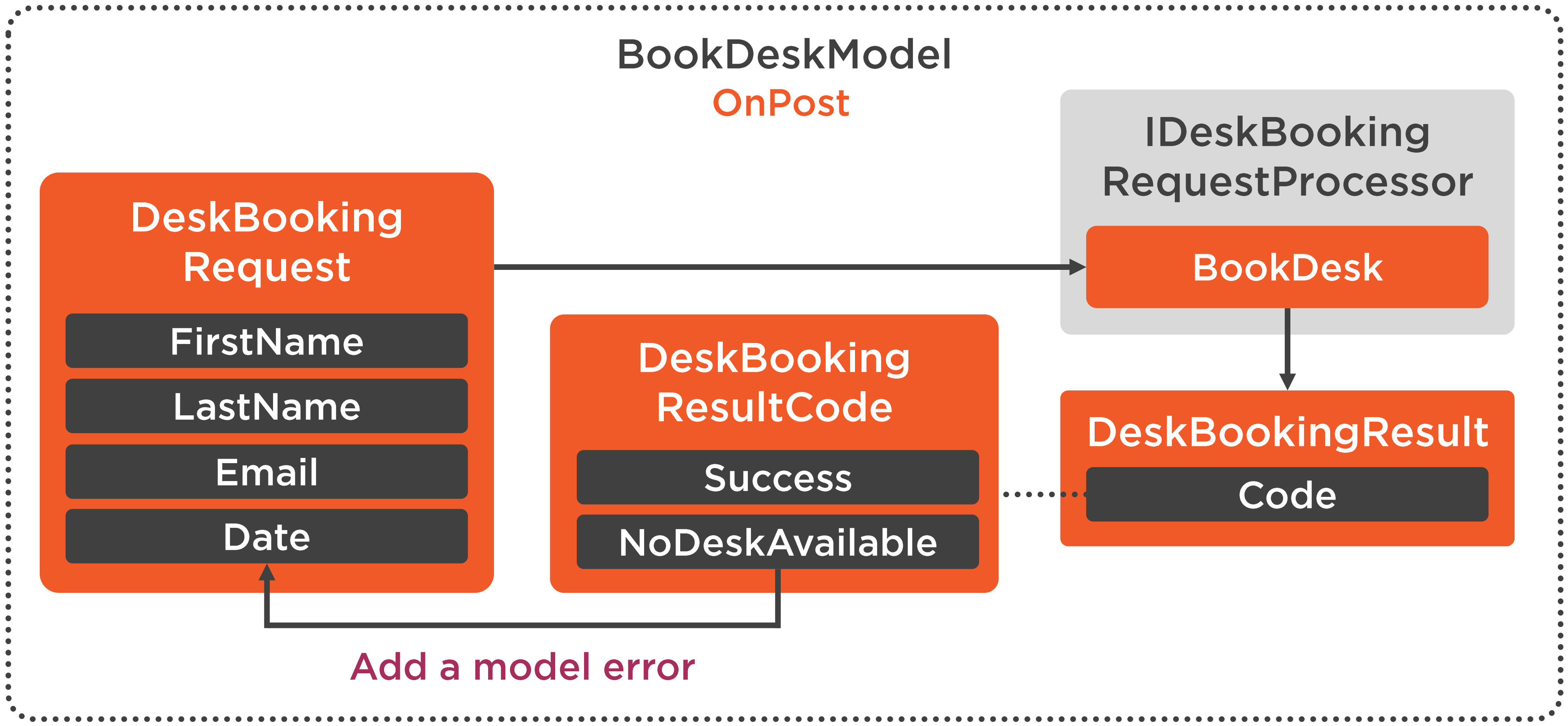
Check if the model is valid

- Change the existing test to a data-driven test
- Implement the logic

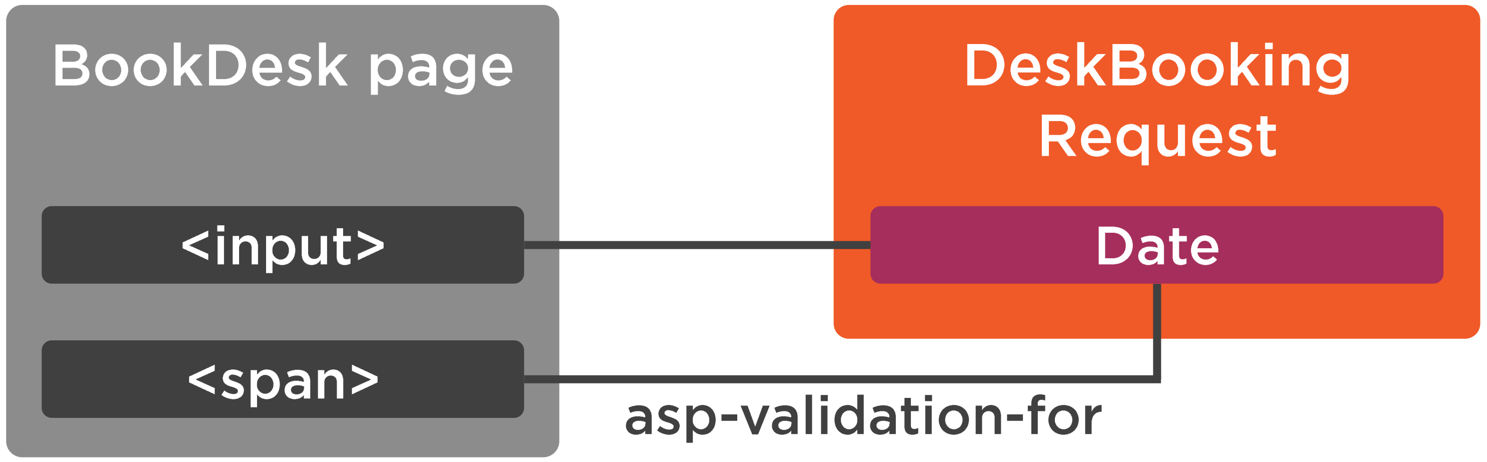
Add a Model Error if No Desk Is Available



Add a Model Error if No Desk Is Available



Add a Model Error if No Desk Is Available

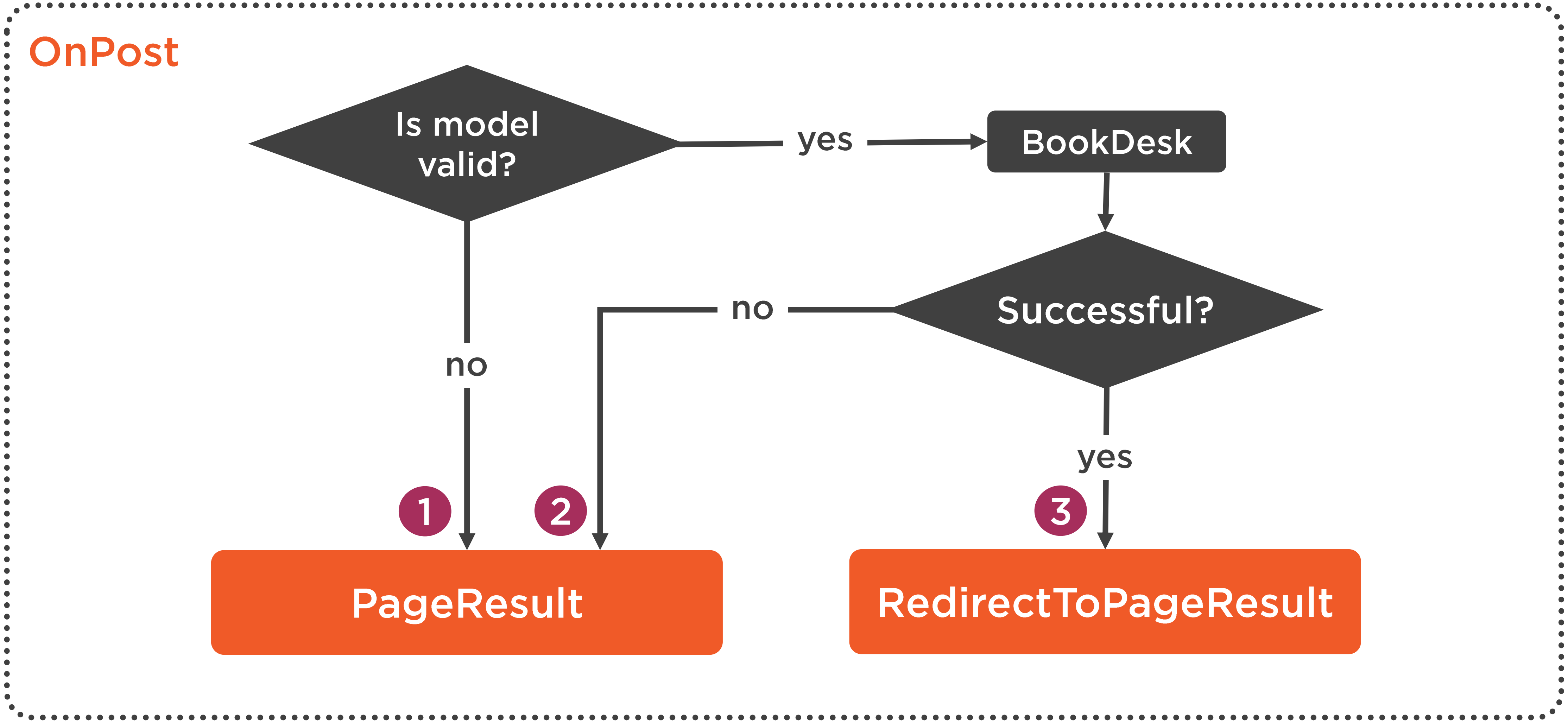


Demo

Add a model error if no desk is available

- Write a test
- Implement the logic
- Refactor the code

Return the Expected IActionResult

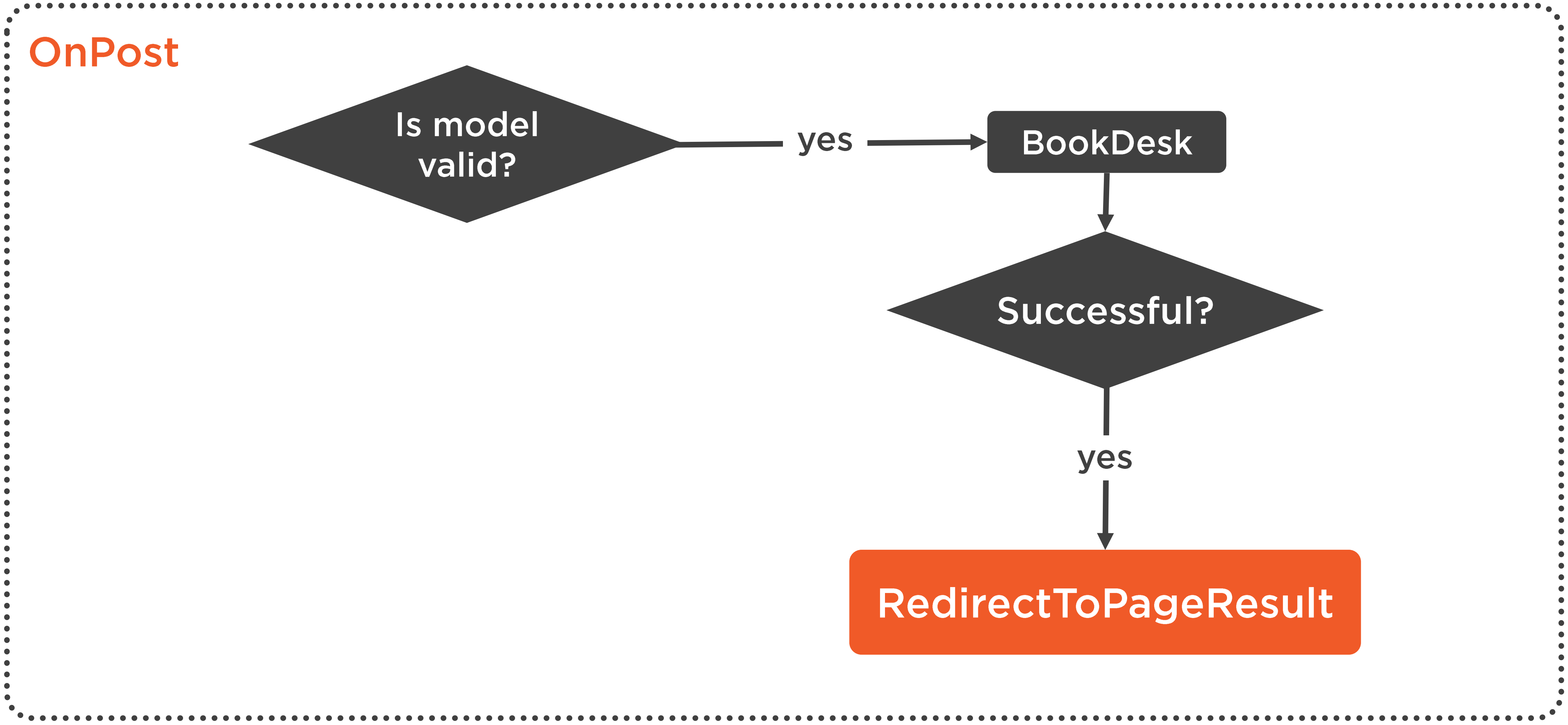


Demo

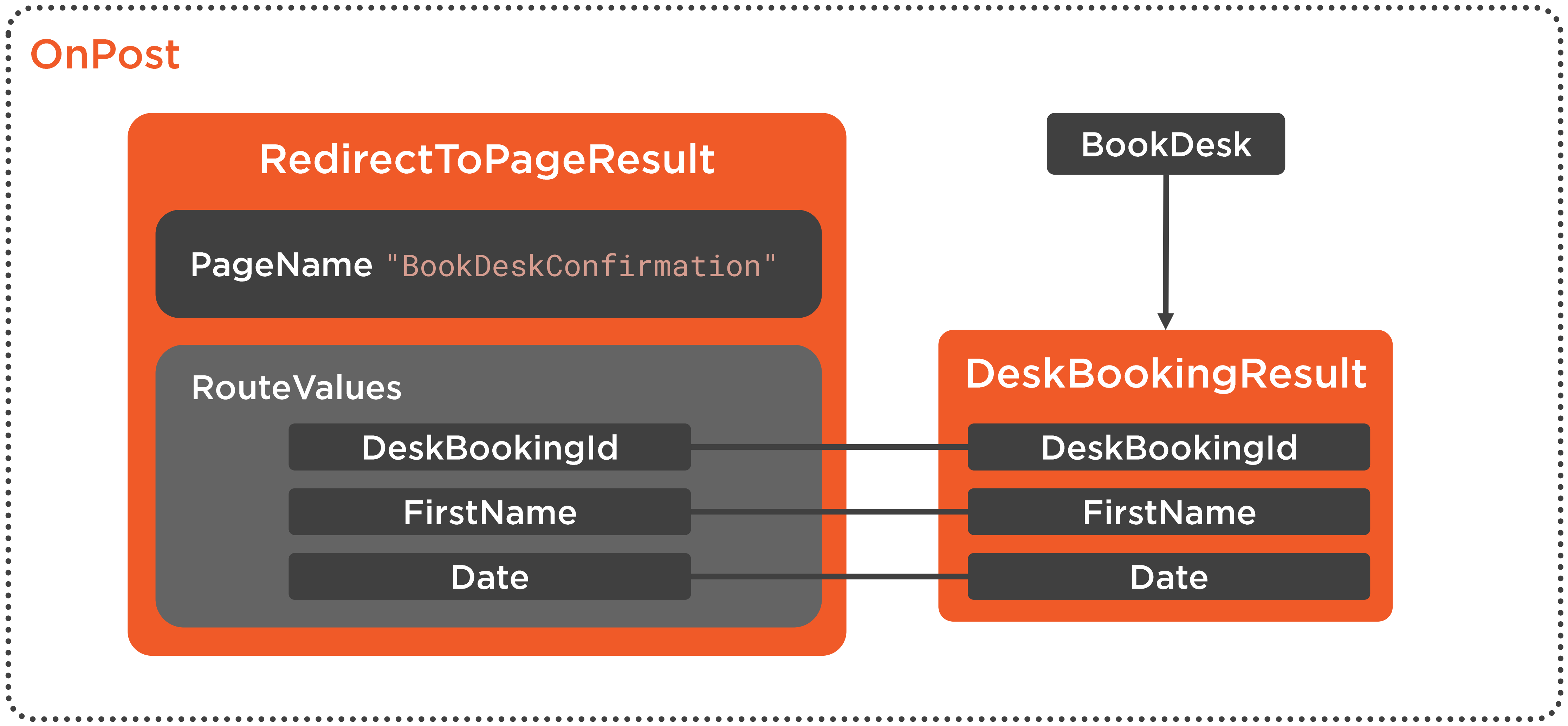
Return the expected IActionResult

- Write a data-driven test
- Implement the logic

Redirect to the BookDeskConfirmation Page



Redirect to the BookDeskConfirmation Page



Demo

Write a test

Implement the requirement

Run the ASP.NET Core Application

BookDeskModel

OnPost

Requirements

- Call the BookDesk method of the processor
- Check if the model is valid
- Add a model error if no desk is available
- Return the expected IActionResult
- Redirect to the BookDeskConfirmation page

Demo

Run the ASP.NET Core application

Summary

Test driven development of user interfaces

- Separate UI logic from UI definition
- ASP.NET Core Razor Pages have UI logic in PageModel

Use TDD to implement requirements in the BookDeskModel

Congratulations!

Course Summary

Test Driven Development

- Red
- Green
- Refactor

Advantages of TDD

- Think about what the code should do
- Get fast feedback
- Write modular and maintainable code

TDD is not just about tests

- It is a powerful approach to build robust software

Test Driven Development in C#

Thomas Claudius Huber

@thomasclaudiush
www.thomasclaudiushuber.com