Building End-to-End Multi-Client Service Oriented Applications

Module 06

Services & Engines – Part 1

pluralsight
hardcore developer training

Highlights

- **WCF Services**
  - Services contracts (business)
  - Custom data contracts
  - Dependency Injection
- **Business Engines**
  - Dependency Injection

Service Contracts

- **One for each pre-defined service**
  - Inventory, Rental, Account
- **Define operations needed, based on use cases.**
- **Operations return business entities**
  - Or custom data contract when needed
- **Will have two sets of contracts**
  - This module will deal with business side only

Services

- **One per contracts**
- **Common base class**
- **Use post-construction DI resolve**
- **Will have a reusable way to provide basic fault-handling in every operation**
- **Security**
  - WCF authorization
  - Custom user-based data authorization (**IAccountOwnedEntity**)

Business Engines

- **More fine-grained than services**
- **Use them for behavior reusability**
- **Most service operations [in this app] involve data access**
- **Business engines “injected” into services**
  - Each will have its interface
  - Same pattern as data repository factory

Summary

- **Service Contract & Service Separation**
- **Services use a base class for common functionality**
- **All dependencies are injected in by MEF**
  - Base constructor engages post-construction DI-resolve
- **Reusable behavior is contained in business engines**
  - Also benefit from having dependencies injected

End of module