Contracts and Services

Miguel A. Castro
@miguelcastro67

pluralsight
hardcore developer training

The Story Line

- **Geographic Service**
- **Offers Zip Code related data and functionality to consumers**
  - Easy to talk about
  - Makes for a good example of something available across an enterprise
  - Also externally
- **Will build on this throughout course to maintain storyline**

Contracts Are NOT Evil

- **Contracts are part of WCF's explicitness**
- **Data Contracts define the shape of the data**
  - Request data goes TO the service
  - Response data returns FROM the service
- **Service Contracts define the API**
  - List of operations that a client can call
- **Only piece that both client and server share**
- **Can share assembly or through "equivalency"** ***(next module)***

Data Contracts

- **A service call needs a bunch of data going to it and a bunch of data coming out**
- **The shape of this data makes up the data contracts**
- **Incoming data (request) can be in a data contract or simple arguments**
- **Return data (response) must be a data contract**
  - A single simple type need not be wrapped in a data contract
- **Data contracts are simple classes with just properties**
  - Auto-implemented properties are just fine
- **Use DataContractSerializer instead of XmlSerializer**
  - Opt-in serializer
- **Attributes used:**
  - DataContract
  - DataMember

Demo Time

Service Contracts

- **Client needs to know what operations are available to call**
  - A well as what data to send and receive
- **Service contracts define the operations of a service**
  - “Define” means “interface” in C# terms
  - Service contract is simply an interface with method members only
- **Attributes used:**
  - ServiceContract
  - OperationContract

Demo Time

Services

- **Services are the implementation of the Service Contract for the service side of the wire**
  - Don't worry, the client comes later
- **A service is the first line of contact and manages all down-level calls**
- **Just classes that implement an interface**
  - For starters
- **Services characteristics will be changed [later] using attributes on this class**

Demo Time

Unit Testing

- **Need to make service operations "testable"**
- **Golden rule of testability:**
  - Do not instantiate dependencies
- **Data repositories need to be injected in**
- **Production code would use DI**
  - We're going to inject, but not with a container (that will come at the end)
- **Need to add additional constructors**
- **Operations will use class-wide repository variables**
  - Instantiate if null
- **Unit test will use constructor overloads and send in mocks**

Demo Time