OrigoDB Workshop

Module 5 – Configuration

EngineConfiguration class – key properties

- AsyncJournaling
- EnsureSafeResults
- Kernel
- Location
- PacketOptions
- PersistenceMode
- SnapshotBehavior

Location

- File location for FileStorage
- Connection string when SqlStorage
- Defaults (when null)
  - Will look in app.config for connection string
  - Type name of model
  - Current working directory
  - App_Data in web context
- Magic
  - mode=remote;host=10.0.0.20;port=3001

Persistence modes

- Journaling (default)
- SnapshotPerTransaction
- ManualSnapshots

```
var config = EngineConfiguration.Create();
config.PersistenceMode = PersistenceMode.SnapshotPerTransaction;
var db = Engine.For<MyModel>(config);
```

Kernels

- OptimisticKernel (default)
- RoyalFoodTaster

```
var config = EngineConfiguration.Create();
config.Kernel = Kernels.RoyalFoodTaster;
var db = Db.For<MyModel>(config);
```

Logging

- Logs to console by default

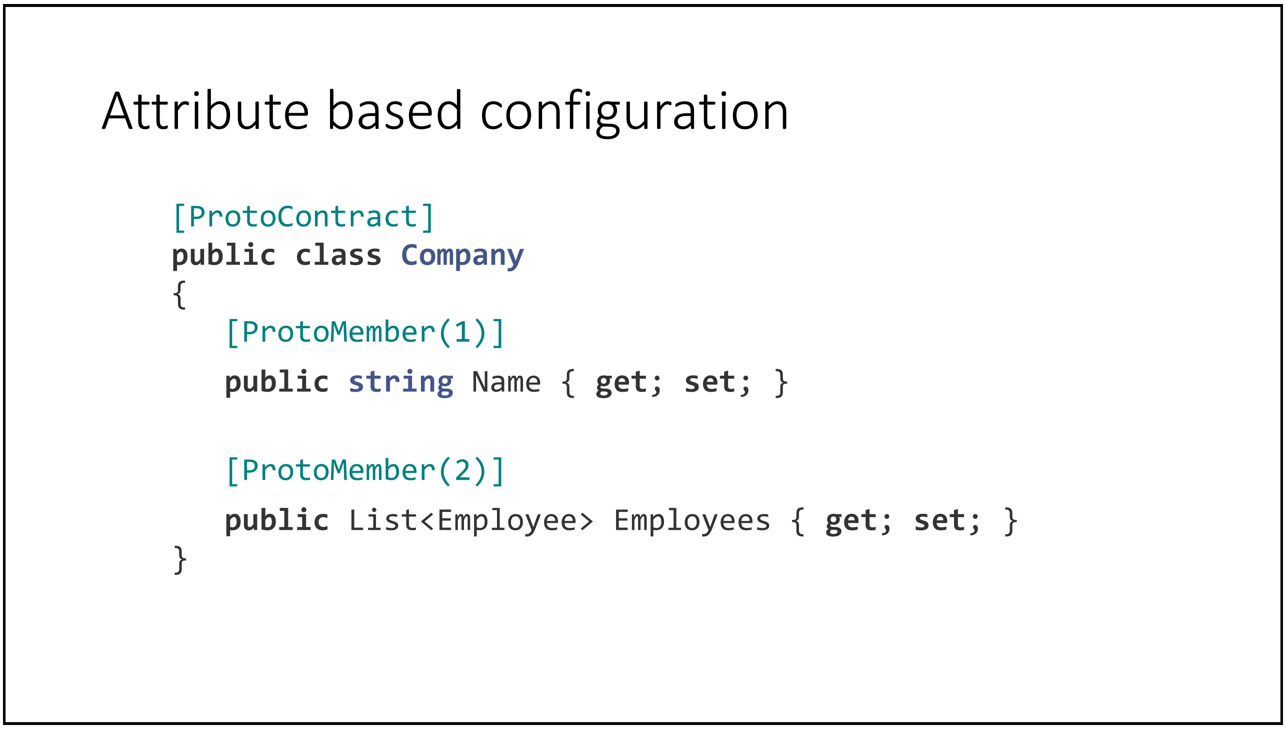
```
//disable logging
ConsoleLogger.MinimumLevel = LogLevel.Off;

//Plugin custom logger
LogProvider.SetFactory(new Log4NetLoggerFactory());
```

Extensibility

- EngineConfiguration methods
  - SetAuthorizerFactory()
  - SetSynchronizerFactory()
  - SetStoreFactory()
  - SetFormatterFactory()

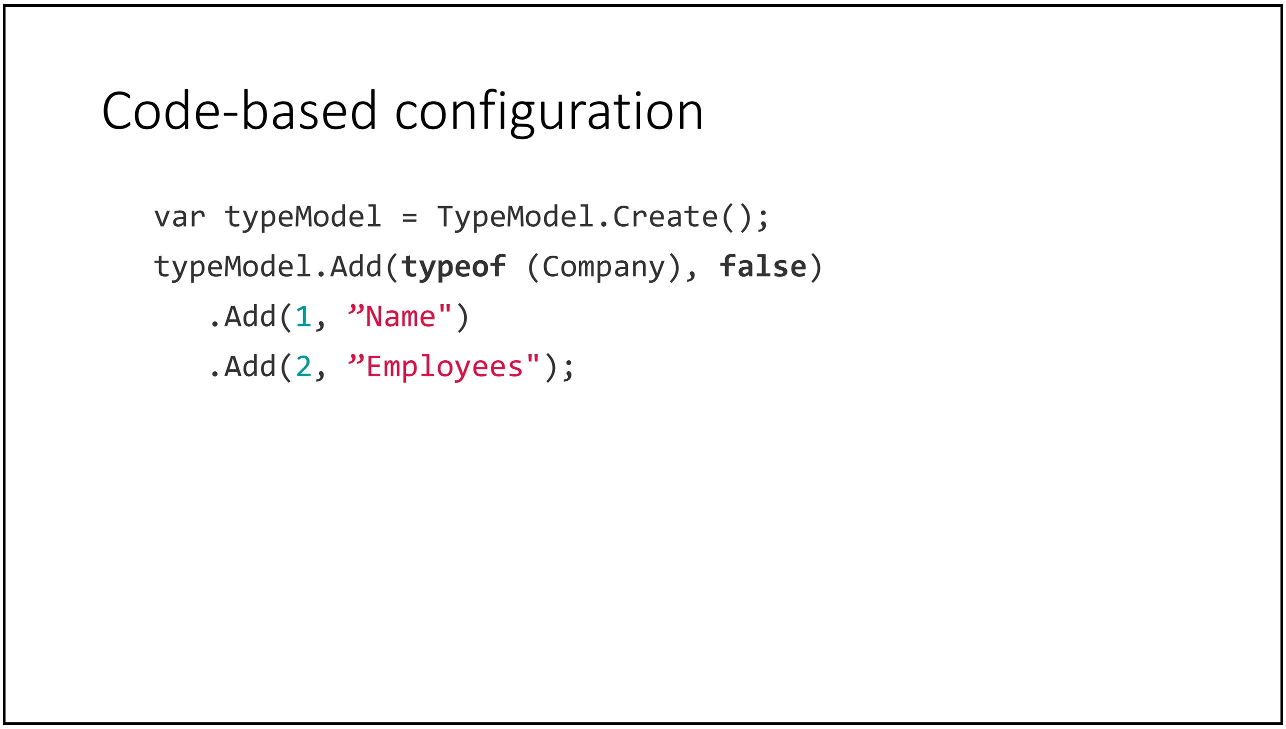
ProtobufFormatter

- Protocol Buffers by Google
- IFormatter wrapper around protobuf-net by @marcgravell
- Contract based as opposed to embedded metadata
- Compact, fast, loose coupling, cross platform
- Configure with attributes or code
- Use it!



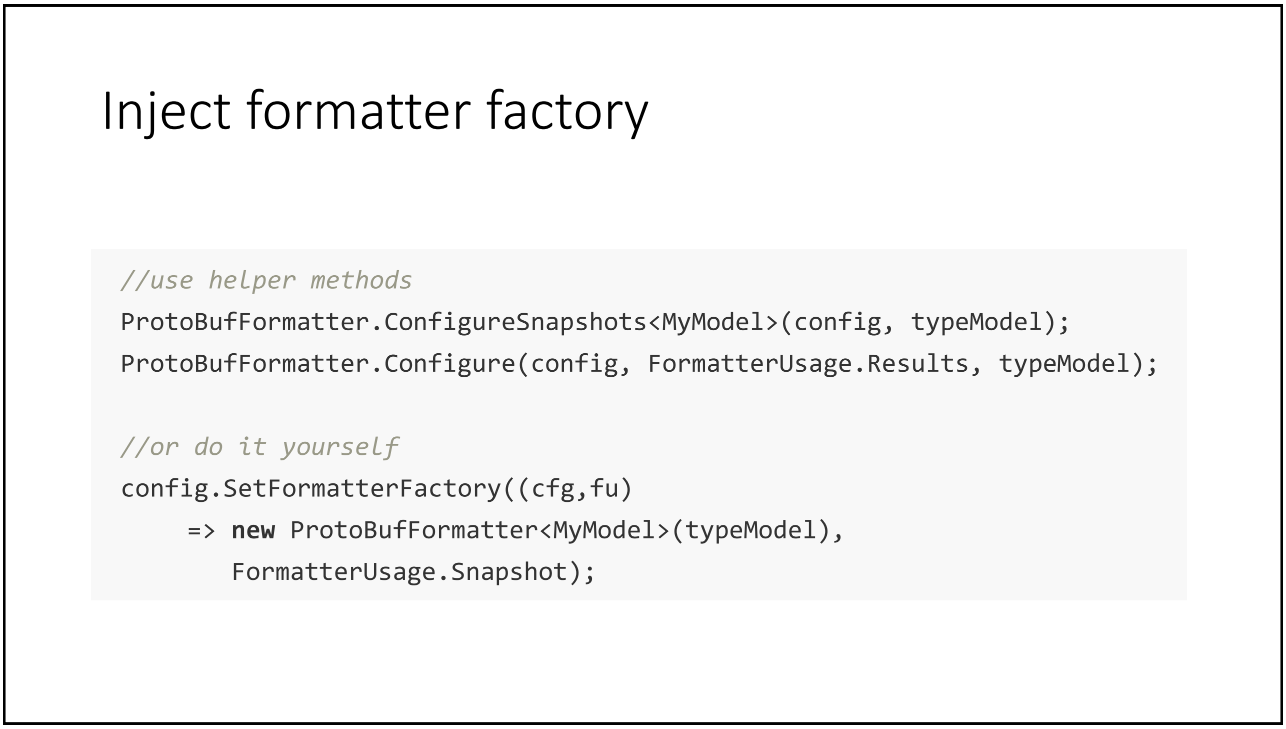
Attribute based configuration

```
[ProtoContract]
public class Company
{
   [ProtoMember(1)]
   public string Name { get; set; }

   [ProtoMember(2)]
   public List<Employee> Employees { get; set; }
}
```



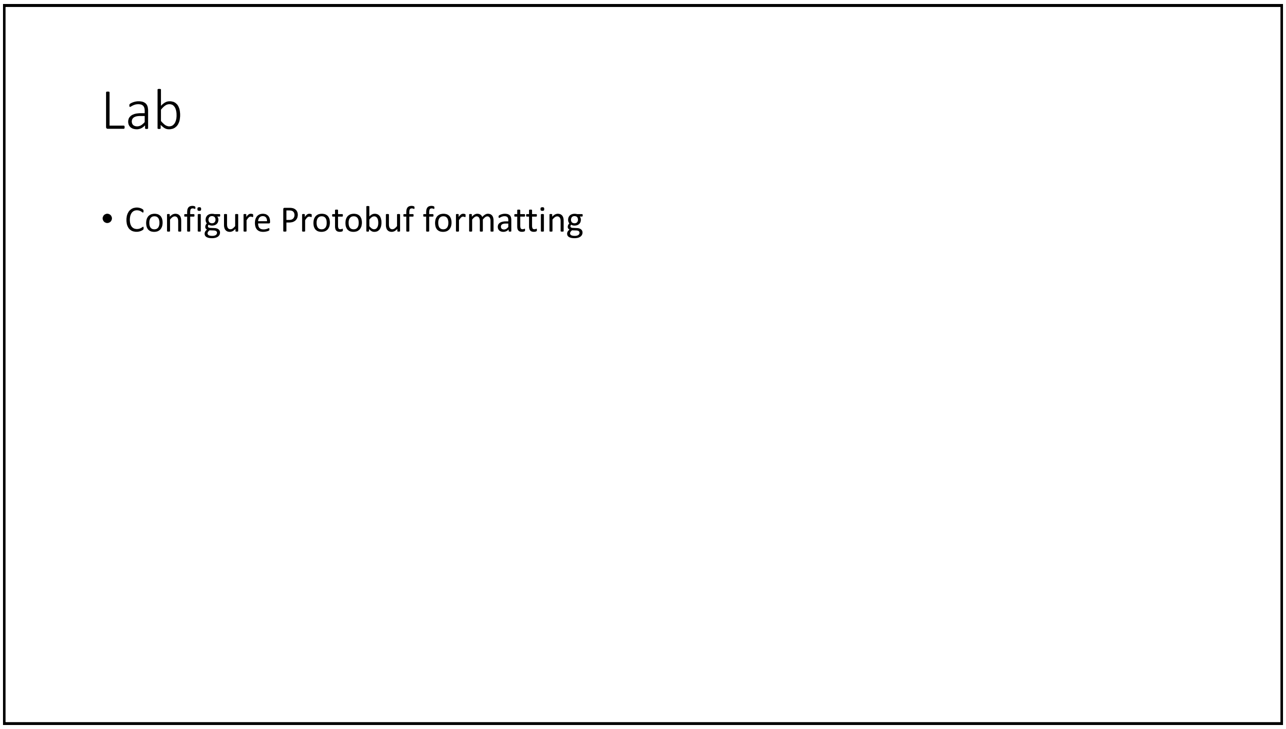
Code-based configuration

```
var typeModel = TypeModel.Create();
typeModel.Add(typeof (Company), false)
   .Add(1, ”Name")
   .Add(2, ”Employees");
```



Inject formatter factory

```
//use helper methods
ProtoBufFormatter.ConfigureSnapshots<MyModel>(config, typeModel);
ProtoBufFormatter.Configure(config, FormatterUsage.Results, typeModel);

//or do it yourself
config.SetFormatterFactory((cfg,fu)
    => new ProtoBufFormatter<MyModel>(typeModel),
       FormatterUsage.Snapshot);
```



Lab

- Configure Protobuf formatting