

Simplifying Code with Dynamic C#



Jason Roberts

.NET Developer

@robertsjason

dontcodetired.com



Overview



Simplifying reflection code

COM interop without interop assemblies

Less plumbing code with dynamic JSON

Dynamic SQL query results with Dapper

Dynamic code considerations



Dynamic Code Considerations

Testing considerations

- Sacrificing some static compile-time checks
- Gaining convenience/benefits of dynamic C#
- Unit/integration tests
- Automated UI tests

Performance

- Performance overhead of dynamic C#
- Dynamic call “lookup” at runtime
- Call site caching behind the scenes
- Performance testing for performant-critical applications



Summary



Simplifying reflection code

COM interop without interop assemblies

Less plumbing code with dynamic JSON

Dynamic SQL query results with Dapper

Dynamic code considerations



Up Next:
Creating Custom Dynamic Classes

