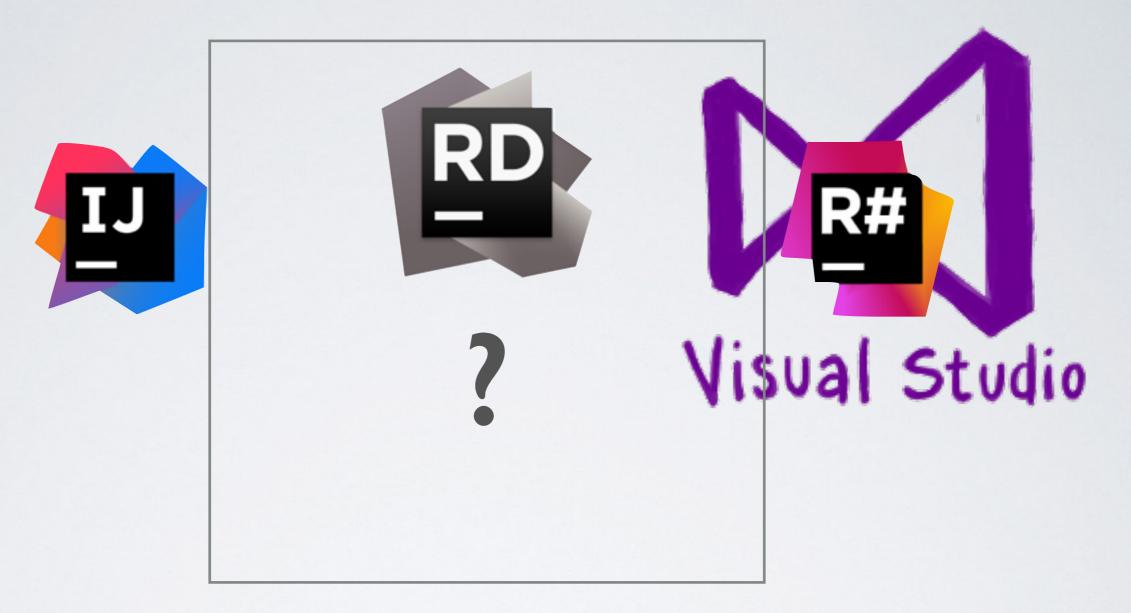
RIDER FRAMEWORK New languages: C++, JavaScript

Dmitry Ivanov, Rider Team, JetBrains



HISTORY of Project Rider

👫 Default 🔻 🗛 Any CPU 🖛 🏴 👫 🔍 🗈 Rider) 🔁 Platform) 🛅 Core) 🛅 Shell) 🛅 Util) 🛅 Src) 🛅 DataFlow) 🛅 Lifetime) 💭 Lifetime.cs) 😌 🐵 🕸 🕛 💭 odeBehindLanguageUtil.cs 🗶 💭 AspTypingAssistTest.cs 🗶 💭 ReactiveEx.cs 🗶 💭 TasksTest.cs 🗶 💭 DependencyFactory.cs 🗶 💭 Lifetime.cs 🗶 💭 IPsiFiles.cs 🗶 💭 PsiFiles.cs 🗶 💷 Compression namespace JetBrains.DataFlow Concurrency /// <summary> Console /// // contents the lifetime of the objects which have an explicit span of life and have actions to be executed upon termination. In that, repl
/// // Converters /// /// /// constructor and schedule termination actions on it. Instead of calling <see cref="IDisposable.Dispose"/>, 🐺 DataFlow /// <p Extension /// <para>To create a lifetime, use <see cref="Lifetimes.Define(JetBrains.DataFlow.Lifetime,string,System.Action[JetBrains.DataFlow.LifetimeDefiniti Infra /// <para>If you need a function-scoped lifetime, like the <c>using{}</c> construct for <see cref="IDisposable"/>, use the <see cref="Lifetimes.Usir /// </summary> Interop public sealed class Lifetime III Lifetime Disposable.cs #if JET MODE ANNOTATE CALL STACKS D EternalLifetime.cs internal Lifetime([NotNull] string id, LifetimeFlags flags, [NotNull] ILogger logger) #else D Lifetime.cs internal Lifetime(LifetimeFlags flags, [NotNull] ILogger logger) LifetimeDefinition.cs Wendif □ LifetimeEx.cs Lifetimes.cs #if JET_MODE_ANNOTATE_CALL_STACKS D ObsoleteLifetimeAndDef throw new ArgumentNullException("id"); D OuterLifetime.cs myId = id; □ StackTraceUtil.cs #endif Cosolete if(logger == null) throw new ArgumentNullException("logger"); StandardPreconditions if(flags != (flags & LifetimeFlags.IsEternal)) AddRemove.cs LifetimeFlags.IsEternal 1<<0, LifetimeFlags AddRemoveEventArgs!1.cs logger.Error(new allowed flag is IsEternal."), ExceptionOrigin.Assertion); 1<<1, LifetimeFlags D AddRemoveloderEventAr B ✓ 5 ReSharperHost.Generated (285 projects) (2 tests) Success AsksTest (1 test) [12.5 sec] ✓ Success V Test.Asp (1 test) Success JetBrains.Platform.Tests.Cases.Shell.Threading.TasksTest V J Test Shell (1 test) Success × 10 * J*c ୍ଦର ✓ ♥ TestDelayedExecution Success Þ ✓ [●] TestMainThreadPumping Success Ŭ. O TestPreprocessSingleThreadedAndParallelize Success 000 5 ✓ [©] TestRecurrentDelayed Success × ✓ [●] TestRecurrentFromNow Success Seal TestRunSynchronous Ignored: Test should be run explicitly 3 TestTaskBarrierWithWriteLock Success Ð TestTaskCancellation Success Ð ✓ [©] TestTasksBarrier Success TestTasksBarrierWithCancellation Success Ξ TestTasksBarrierWithFault Success UnitTesting 4 Build Terminal Q 3: Find 4: Run A 6: TODO Event Log > NuGet

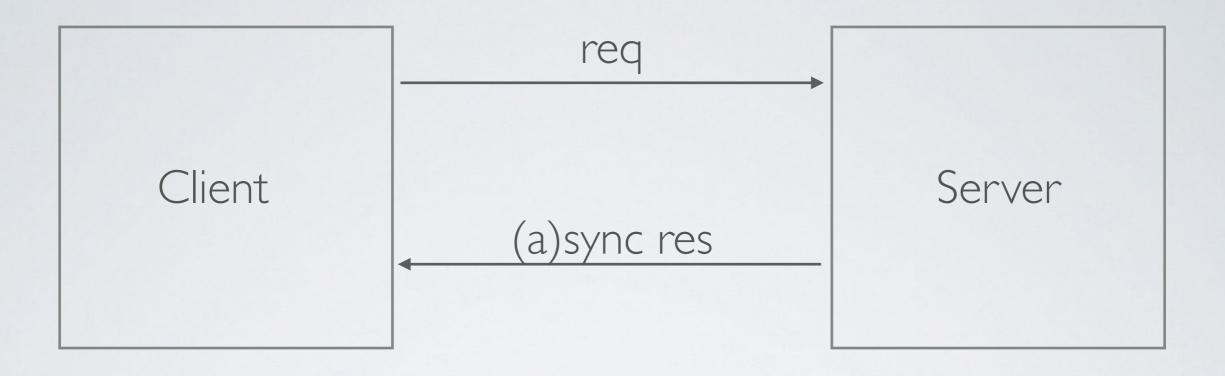
INCEPTION

Hierarchical nature of view model

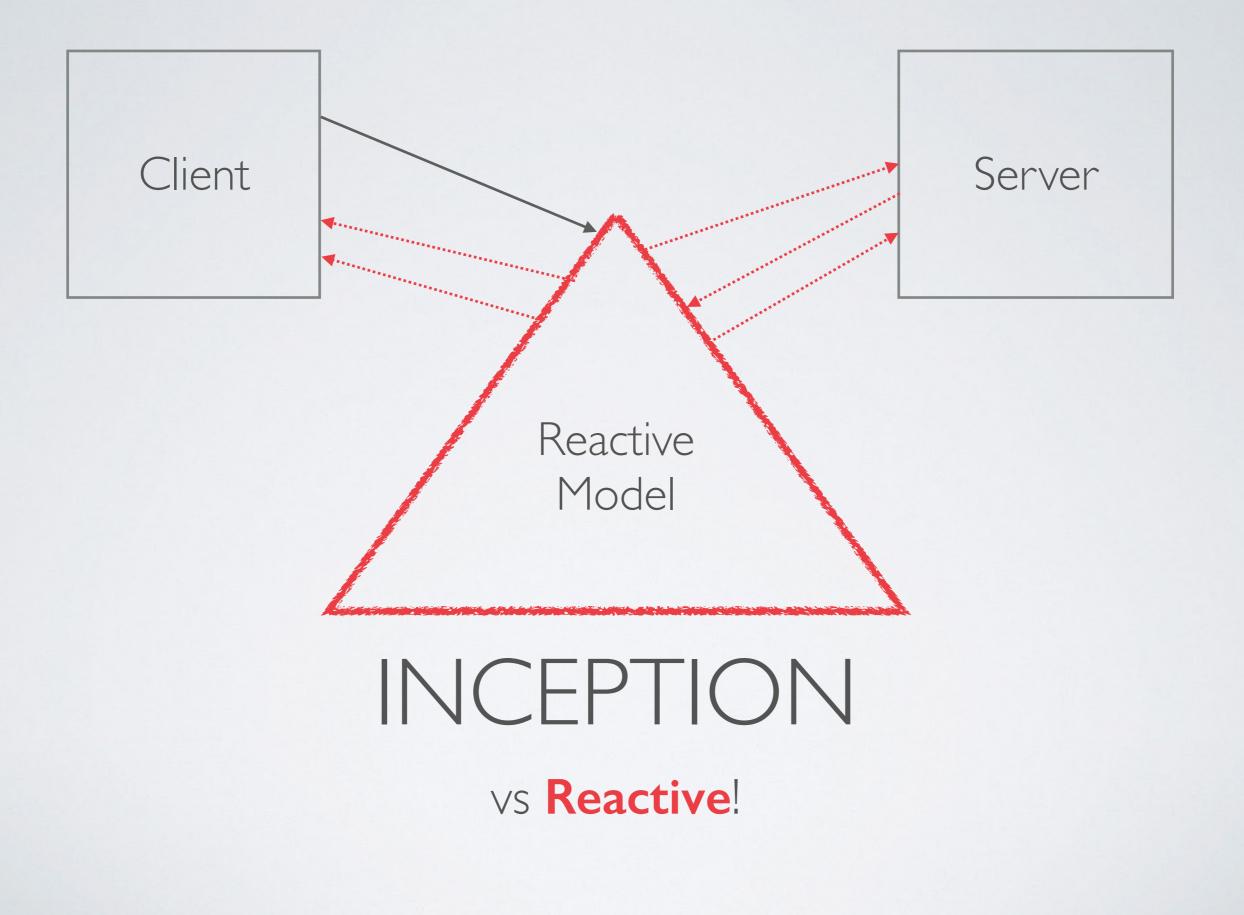
| DataFlow Default Any CPU Any | (C |
|---|----------|
| 🕼 CodeBehindLanguageUtil.cs × 🕼 AspTypingAssistTest.cs × 🥼 ReactiveEx.cs × 🥼 TasksTest.cs × 🕼 DependencyFactory.cs × 🕼 Lifetime.cs × 🕼 IPsiFiles.cs × 🕼 PsiFiles.cs × | ≣2 |
| 16 Jnamespace JetBrains.DataFlow | <u> </u> |
| 17 | =: |
| 18 /// <summary></summary> | |
| 19 /// <para>Represents the lifetime of the objects which have an explicit span of life and have actions to be executed upon termination. In that,</para> | |
| 20 /// <pre>// <pre< td=""><td></td></pre<></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre> | |
| /// <para> If a function gets lifetime in parameters, the lifetime defines the duration of the effect of the function. For instance, adding an i /// <para>To create a lifetime, use <see cref="Lifetimes.Define(JetBrains.DataFlow.Lifetime,string,System.Action{JetBrains.DataFlow.LifetimeDefi</p></td><td></td></tr><tr><td>22 /// <para>If you need a function-scoped lifetime, like the <c>using{}</c> construct for <see cref=" idisposable"=""></see>, use the <see cref="Lifetimes.</p></td><td></td></tr><tr><td>24 /// </summary></td><td></td></tr><tr><td>25 public sealed class Lifetime</td><td>at at</td></tr><tr><td>26 {</td><td></td></tr><tr><td>27 #if JET_MODE_ANNOTATE_CALL_STACKS</td><td></td></tr><tr><td>28 internal Lifetime([NotNull] string id, LifetimeFlags flags, [NotNull] ILogger logger)</td><td></td></tr><tr><td>29 #else</td><td>-</td></tr><tr><td>30 internal Lifetime(LifetimeFlags flags, [NotNull] ILogger logger)</td><td></td></tr><tr><td>31 🖯#endif</td><td></td></tr><tr><td>32 {</td><td>-</td></tr><tr><td><pre>33 #if JET_MODE_ANNOTATE_CALL_STACKS</pre></td><td></td></tr><tr><td>34 if(id.IsEmpty())</td><td></td></tr><tr><td><pre>35 throw new ArgumentNullException(" id");<="" pre=""></see></para></para> | |
| 36 myId = id; 37 #endif | |
| 38 if(logger == null) | |
| <pre>39 throw new ArgumentNullException("logger"); I</pre> | |
| 40 if(flags != (flag= 2 tifetimeflag=10tternal)) | |
| 41 { LifetimeFlags.IsEternal 1<<0, LifetimeFlags | |
| 42 logger.Error(now LifetimeFlags.IsTerminated 1<<1, LifetimeFlags allowed flag is IsEternal."), ExceptionOrigin.Assertion); | |
| 43 flags = (flags &) | |
| and LifetimeFlags π | 1 |

INCEPTION

Hierarchical nature of view model



INCEPTION RPC



- **Ready**: Framework for Kotlin(Java), C#
- **Todo:** Framework for C++, JavaScript
- Skills required:
 - Target language knowledge
 - Basic Java knowledge
 - Algorithms
 - Protocols, Networking (Optional)

WHAT IS DONE

and what we need to do

QUESTIONS AND ANSWERS

thanks for your attention

dmitry.ivanov@jetbrains.com