

I *ExecutionContext*

- procedure: Struct?
- substitution: Substitution.Unifier
- prologStackTrace: List<Struct>
- customData: CustomDataStore
- createSolver(...): Solver
- createMutableSolver(...): MutableSolver
- update(...): ExecutionContext
- apply(sideEffect: SideEffect): ExecutionContext



I *ExecutionContextAware*

- libraries: Libraries
- flags: FlagStore
- staticKb: Theory
- dynamicKb: Theory
- operators: OperatorSet
- inputChannels: InputStore
- outputChannels: OutputStore
- standardInput: InputChannel<String>
- standardOutput: OutputChannel<String>
- standardError: OutputChannel<String>
- warnings: OutputChannel<PrologWarning>

I *Solver*

- solve(goal: Struct, options: SolveOptions): Sequence<Solution>
- solveList(goal: Struct, options: SolveOptions): List<Solution>
- solveOnce(goal: Struct, options: SolveOptions): Solution
- clone(): Solver

