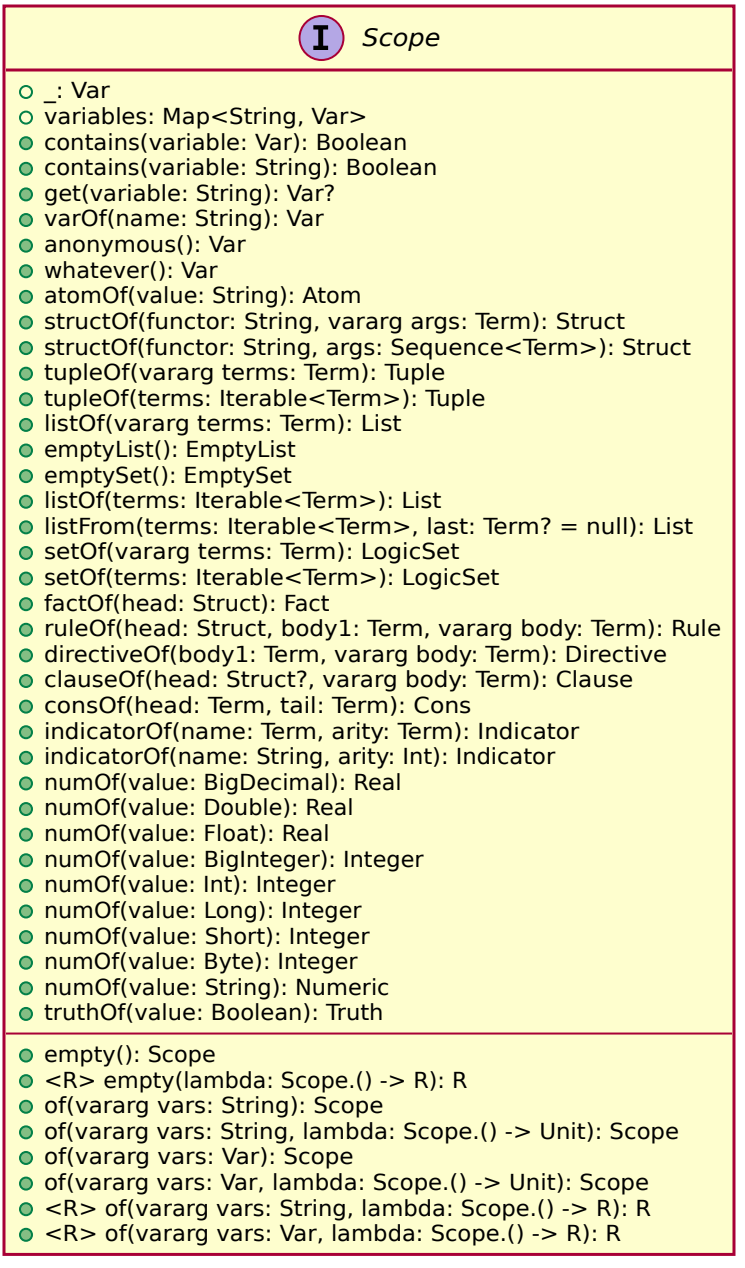
Scope
_: Var
variables: Map<String, Var>
contains(variable: Var): Boolean
contains(variable: String): Boolean
get(variable: String): Var?
varOf(name: String): Var
anonymous(): Var
whatever(): Var
atomOf(value: String): Atom
structOf(functor: String, vararg args: Term): Struct
structOf(functor: String, args: Sequence<Term>): Struct
tupleOf(vararg terms: Term): Tuple
tupleOf(terms: Iterable<Term>): Tuple
listOf(vararg terms: Term): List
emptyList(): EmptyList
emptySet(): EmptySet
listOf(terms: Iterable<Term>): List
listFrom(terms: Iterable<Term>, last: Term? = null): List
setOf(vararg terms: Term): LogicSet
setOf(terms: Iterable<Term>): LogicSet
factOf(head: Struct): Fact
ruleOf(head: Struct, body1: Term, vararg body: Term): Rule
directiveOf(body1: Term, vararg body: Term): Directive
clauseOf(head: Struct?, vararg body: Term): Clause
consOf(head: Term, tail: Term): Cons
indicatorOf(name: Term, arity: Term): Indicator
indicatorOf(name: String, arity: Int): Indicator
numOf(value: BigDecimal): Real
numOf(value: Double): Real
numOf(value: Float): Real
numOf(value: BigInteger): Integer
numOf(value: Int): Integer
numOf(value: Long): Integer
numOf(value: Short): Integer
numOf(value: Byte): Integer
numOf(value: String): Numeric
truthOf(value: Boolean): Truth
empty(): Scope
<R> empty(lambda: Scope.() -> R): R
of(vararg vars: String): Scope
of(vararg vars: String, lambda: Scope.() -> Unit): Scope
of(vararg vars: Var): Scope
of(vararg vars: Var, lambda: Scope.() -> Unit): Scope
<R> of(vararg vars: String, lambda: Scope.() -> R): R
<R> of(vararg vars: Var, lambda: Scope.() -> R): R