### Annotation @BLOCK and @DATABASE

Used on classes extending Form and Block.

Definition: BLOCK(alias: string, component: Block, usage: databaseUsage)

Definition: DATABASE({query: boolean, insert: boolean, update: boolean, delete: boolean})

Used in blocks:

@BLOCK("emp") // set default block alias

@DATABASE({query: true, insert: true, update: true}) // set default block database usage

export class TestBlock extends Block

Used in forms:

@BLOCK("emp1", TestBlock) // define block “emp1” with default database usage

@BLOCK("emp2", TestBlock,{query: true, insert: false}) // define block “emp1” overrides default database usage on query and insert

@DATABASE({query: true, insert: true, update: true, delete: true) // set form database usage

export class Test extends Form

{

@BLOCK() // define block, alias is taken from the block default. Type is defined by the property (TestBlock)

@DATABASE({delete: true}) // override database usage on delete

private manager1: TestBlock = new TestBlock(); // the block property, to be used somewhere

@BLOCK("manager2") // define block “manager2”. Type is defined by the property (TestBlock)

@DATABASE({update: true}) // override database usage on delete

private manager2: TestBlock = new TestBlock (); // the block property, to be used somewhere

@BLOCK("manager3", TestBlock) // define block “manager3”. Type is TestBlock

@DATABASE({query:false, insert: false, delete: true}) // override database usage

private manager3: TestBlock; // property manager3 will be created on the fly