

Annotations (org.apache.isis.applib.annotation)			... annotations continued	
@DomainService <i>type</i>	Indicates a class is a domain service singleton that contributes <i>Actions</i> to the application's menu and/or to the REST backend.		@Domain.Include <i>field, method</i>	Indicates that a field or method must contribute to the metamodel.
@DomainServiceLayout <i>type</i>	Domain service layout customization. When present, overridden by menubars.layout.xml		@Domain.Exclude @Programmatic <i>field, method, type</i>	Indicates that a field, method or type must not contribute to the metamodel.
@HomePage <i>type</i>	Indicates which view-model should be used to render as homepage.		Services (most common)	
@DomainObject <i>type</i>	Indicates a class that has an <i>identity</i> and is <i>bookmarkable</i> , it is either a view-model or an entity . View-models may alternatively implement interface ViewModel and optionally java.io.Serializable .		RepositoryService	access to persistence layer
@DomainObjectLayout <i>type</i>	Domain object layout customization. When present, overridden by Xxx.layout[layoutName].xml > Xxx.layout.xml > Xxx.layout.fallback.xml .		MessageService	UI notifications
@Action <i>method, mixin</i>	Indicates that a method contributes an <i>Action</i> .	action-support methods @Action .. placeOrder(X x, Y y, Z z) boolean hide PlaceOrder() String ^[1] disable PlaceOrder() boolean hide {0 1 2}PlaceOrder(X Y Z); String ^[1] disable {0 1 2}PlaceOrder(X Y Z) String ^[1] validate {0 1 2}PlaceOrder(X Y Z) String ^[1] validate PlaceOrder(X x, Y y, Z z) Coll.<X Y Z> autoComplete {0 1 2}PlaceOrder(X Y Z, @MinLength(3) String search) 2 args Collection<X Y Z> choices {0 1 2}PlaceOrder(X Y Z) X Y Z default {0 1 2}PlaceOrder() String ^[1] named PlaceOrder() String ^[1] described PlaceOrder()	FactoryService	object construction
@ActionLayout <i>field, getter, mixin</i>	<i>Action</i> layout customization. When present, overridden by Xxx.layout[layoutName].xml > Xxx.layout.xml > Xxx.layout.fallback.xml .		ClockService	provides virtual clock
@Title <i>field, method</i>	Indicates which property or properties make up the object title.		WrapperFactory	enforce domain rules on domain objects
@Property <i>field, getter, mixin</i>	Indicates that a field or method contributes a <i>Property</i> .	property-support methods @Property .. email, getEmail (), setEmail (); boolean hide Email(); String ^[1] disable Email(); String ^[1] validate Email(X); Collection<..> autoComplete Email(), Collection<..> choices Email(); String ^[1] named Email() String ^[1] described Email()	EventBusService	emit custom events
Typically used with @lombok.Getter @lombok.Setter			BookmarkService	bookmark = object identity
@PropertyLayout <i>field, getter, mixin</i>	<i>Property</i> layout customization. When present, overridden by Xxx.layout[layoutName].xml > Xxx.layout.xml > Xxx.layout.fallback.xml .		InteractionService	<i>Action</i> execution and <i>Property</i> modification
@Collection <i>field, getter, mixin</i>	Indicates that a field or method contributes a <i>Collection</i> , meaning a non-scalar property.	collection-support methods @Collection .. orders, List<Order> getOrders () boolean hide Orders() String ^[1] disable Orders() String ^[1] named Orders() String ^[1] described Orders()	TransactionService	request transactions
Typically used with @lombok.Getter @lombok.Setter			MetaModelService	export domain model
@CollectionLayout <i>field, getter, mixin</i>	<i>Collection</i> layout customization. Overridden by Xxx.layout[layoutName].xml > Xxx.layout.xml > Xxx.layout.fallback.xml .		Object Methods	
@Parameter @MinLength <i>parameter</i>	Action parameter constraint and behavior customization.		boolean hidden()	hide all members
@ParameterLayout <i>parameter</i>	Action parameter layout customization.		String ^[1] disabled()	disable all members
@MemberSupport <i>method</i>	Indicates that a method supports either an <i>Action</i> or a <i>Property</i> or a <i>Collection</i> .		Layout Support	
			String ^[1] title()	imperative title literal
			String iconName()	icon file suffix (UI)
			String cssClass()	additional css class (UI)
			String layout()	layout file suffix (grid layout)
			Lifecycle Callback	
			void created()	emitted by FactoryService
			void loaded() void saved() void saving() void persisted() void persisting() void deleting() void removing() void updated() void updating()	emitted by persistence layer integration (JDO/JPA)
			Translation	
			[1] All member-support methods (and some object methods) that return String can optionally return TranslatableString .	

Value Types

Number Types	byte, Byte, short, Short, int, Integer, long, Long, float, Float, double, Double, BigInteger, BigDecimal
Boolean Types	boolean, Boolean
Text Types	char, Character, String, Password
Markup Types	Markup, AsciiDoc, Markdown
Temporal Types	java.util.Date, java.sql.{Date Timestamp} java.time.{LocalDate LocalDateTime} Joda Time (deprecated) time-of-day types yet not supported by UI, hence not listed
Enum Types	any
Collection Types	Collection<T>, List<T>, Set<T>, Can<T>
Additional Types	BufferedImage, Blob, Clob, UUID, Url, LocalResourcePath, SSE (ServerSentEvents)

Error & Exception Handling

RecoverableException	anticipated error that results in an UI notification popup
NonRecoverableException	error that results in an error page showing the stacktrace

Spring Annotations and Standard API

@SpringBootApplication @SpringBootTest @Configuration @Import @ComponentScan @EntityScan	Bootstrapping
@Component @Service @Repository	Indicates that a class is a "component". Such classes are considered as candidates for auto-detection when using annotation-based configuration and classpath scanning.
@EventListener	Marks a method as a listener for application events. Domain Events .. subscribe using Spring's @EventListener @DomainObject(xxxLifecycleEvent=....) @DomainObjectLayout(xxxUiEvent=....) @Action(domainEvent=....) @Property(domainEvent=...) @Collection(domainEvent=...)
@Nullable (org.springframework.lang)	declares that annotated elements can be null
@javax.inject.Inject @Autowired	Marks a constructor, field, setter method, or config method as to be autowired by Spring's dependency injection facilities.
@PostConstruct	Marks a method to be executed after dependency injection is done to perform any initialization.
@PreDestroy	Marks a method as a listener for notification that the instance is in the process of being removed by the container .

New Programming Style: Parameters as Typed Tuple [2.0.0-M6]

Action parameters can now be collected into an immutable value type say **Parameters**, a typed tuple. The name is arbitrary. Future releases might support Java records. Instances of the **Parameters** type are passed to the various action-support methods, which need to be single-arg, except for **autoComplete**, which is required to be bi-arg.

For regular objects, action-support methods must reference parameters by index (0, 1, 2, ...). However, with *Mixins* it is allowed to reference parameters by name.

Example: *Action Mixin*, with nested class **Parameters**, using parameter references by name.

```
@Action
@RequiredArgsConstructor
public class Customer_placeOrder {

    private final Customer target;

    // typed tuple made of all the action parameters
    @Value @Accessors(fluent = true)
    public static class Parameters {
        Product product;
        int quantity;
    }

    public Customer act(
        @Parameter Product product,
        @Parameter int quantity) {
        // ...
        return target;
    }

    // supporting methods (no action name reference required)
    public boolean hide() { ... }
    public String disable() { ... }
    public String validate(Parameters params) { ... }

    // parameter supporting methods (exemplified on first parameter)
    public boolean hideProduct(Parameters params) { ... }
    public String disableProduct(Parameters params) { ... }
    public String validateProduct(Parameters params) { ... }
    public Collection<Product> choicesProduct(Parameters params){ ... }
    // note: additional search parameter required
    public Collection<Product> autoCompleteProduct(
        Parameters params, String search) { ... }
    public Product defaultProduct(Parameters params) { ... }

    // parameter supporting methods (exemplified on second parameter)
    public boolean hideQuantity(Parameters params) { ... }
    // ...
}
```

Entity and Viewmodel Annotations

JPA Entities (javax.persistence)
@Entity @Table @NamedQueries @Id @Version @Column @Transient
JDO Entities (javax.jdo.annotation)
@PersistenceCapable @DatastoreIdentity @Inheritance @Discriminator @Version @Queries @Uniques @Indices @NotPersistent @Column @PrimaryKey @Join @Element
JAXB View Models (javax.xml.bind.annotation)
@XmlRootElement; @XmlType @XmlAccessorType; @XmlTransient referenced entities: @XmlJavaTypeAdapter(PersistentEntityAdapter.class)