Annotations (org.apache	e.isis.applib.annotations)		
@DomainService type	Indicates a class is a domain the application's menu and/	-	
@DomainServiceLayout <i>type</i>	Domain service layout custor menubars.layout.xml	mization. When pre	sent, overridden by
@HomePage type	Indicates which view- model should be used to render as homepage.	@Title field, method	Indicates which <i>Property</i> or <i>Properties</i> make up the object title.
@DomainObject type	Indicates a class that has an view-model or an entity. View-models may alternative optionally java.io.Serializable	ely implement interf	
@DomainObjectLayout type	Domain object layout customization. When present, overriden by Xxx.layout[.layoutName].xml > Xxx.layout.xml > Xxx.layout.fallback.xml.		
@Action method, mixin	Indicates that a method contributes an Action. Example: @Action placeOrder(X x, Y y, Z z)		
Action support methods	String ^[1] namedPlaceOrder() boolean hidePlaceOrder() bring ^[1] disablePlaceOrder() String ^[1] disablePlaceOrder() String ^[1] disablePlaceOrder(X Y Z) String ^[1] disable[0 1 2]PlaceOrder(X Y Z) X Y Z default{0 1 2}PlaceOrder() Collection X Y Z > choices{0 1 2}PlaceOrder(X Y Z) Collection X Y Z > autoComplete{0 1 2}PlaceOrder(X Y Z,		
@ActionLayout method, mixin	Action layout customization. Xxx.layout[.layoutName].xml		
@Property field, getter, mixin	Indicates that a field or meth @Title, then used for (part of with @lombok.Getter @lombol Example: @Property Email	f) the title of the obj k.Setter	ect. Typically also used
Property support methods	String ^[1] namedEmail() boolean hideEmail() Email defaultEmail() Collection <email> autoCompl String^[1] validateEmail(Email)</email>	String ^[1] disa ollection <email> choi eteEmail(@MinLeng</email>	icesEmail()
@PropertyLayout field, getter, mixin	Property layout customizatio Xxx.layout[.layoutName].xml		-
@Collection field, getter, mixin	Indicates that a field or meth scalar property. Typically use Example: @Collection List<	ed with @lombok.Get	ter @lombok.Setter
Collection support methods	String ⁽¹⁾ namedOrders() boolean hideOrders()	String ^[1] described String ^[1] disableOrd	
@CollectionLayout field, getter, mixin	Collection layout customizati Xxx.layout[.layoutName].xml		The state of the s
@Parameter parameter	Action parameter constraint and behavior customization.	@MinLength parameter	Search parameter's minimum required character count.
@ParameterLayout parameter	Action parameter layout cust	tomization.	
@ObjectSupport method	Indicates that a method supports its <i>Object</i> . Not allowed on <i>Mixins</i> .	@ObjectLifecycle method	Object lifecycle callback method. (no-arg, void)
@MemberSupport <i>method</i>	Indicates that a method supposite with Objects and Mixins. Also		The state of the s

	annotations continued	
	@Domain.Include field, method	Indicates that a field or method must contribute to the metamodel.
	@Domain.Exclude @Programmatic field, method, type	Indicates that a field, method or type must not contribute to the metamodel.

Services (most common)

RepositoryService	access to persistence layer
MessageService	UI notifications
FactoryService	object construction
ClockService	provides virtual clock
WrapperFactory	enforce domain rules on domain objects
EventBusService	emit custom events
BookmarkService	bookmark = object identity
InteractionService	Action execution and Property modification
TransactionService	request transactions
MetaModelService	export domain model

Object Methods

Object Support used with @ObjectSupport	
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)
boolean hidden()	hide all members
String ^[1] disabled()	disable all members
Lifecycle Callback used with @ObjectLifecycle	
void created()	emitted by FactoryService
void loaded() void persisting()	emitted by persistence layer integration (JDO/JPA)
void persisted() void updating()	supported synonyms: saving() = persisting()
void updated()	saved() = persisted()

Translation

void removing()

[1] All member-support methods (and some object methods) that return String can optionally return TranslatableString.

deleting() = removing()

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	
Temportal 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>, T[]</t></t></t></t>	
Additional Types	BufferedImage, Blob, Clob, UUID, Url, LocalResourcePath, SSE (ServerSentEvents)	

Error & Exception Handling

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

Spring Annotations and Standard API

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.

```
@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:
  // support methods (no action name reference required)
  public boolean hide() { ... }
  public String disable() { ... }
  public String validate(Parameters params) { ... }
  // parameter support 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: search
  public Collection<Product> autoCompleteProduct(
      Parameters params, @MinLength(3) 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)

```
@XmRootElement; @XmlType
@XmlAccessorType; @XmlTransient
referenced entities: @XmlJavaTypeAdapter(PersistentEntityAdapter.class)
```