

Entity

What is it for?

Entities are used to define the structure of your data.

What is it?

An entity is defined by a set of properties ([Attributes](#)) and can either be of a physical or conceptual nature. Examples of possible entities: Customer, Employee, Policy, House, Car, Driver, Call, Incident, Product etc.



Tip: The name of an entity should be singular.

Instances

In the runtime, entities take shape in the form of an [Instance](#).

Take for example an application for managing a queue. The entity Person has several [Attributes](#), such as Name and BirthDate. Multiple persons can be entered into the application, which means that there can also be multiple instances of the entity Person.

The application only handles one queue at a time, so there is also an entity Queue, of which there is only one instance. This is called a "singleton".

Please see the documentation on [Instance](#) for further information.

Abstract & base entities

Entities can be defined in terms of an existing entity by starting with the base entity's definition and adding to it. This principle is called inheritance. By selecting a base-entity while creating a new entity, the latter will inherit all attributes and relations of the base-entity. These inherited attribute definitions cannot be changed. Additional attributes can be added to the entity's definition.

An entity can be made abstract, in which case no instances can be created of this entity. Other entities can inherit attributes from this abstract entity, and these entities can be instantiated (if they are not abstract).

If the abstract class is also singleton, then it can only be used as base entity for singleton entities. Otherwise, it can be used by both singleton and multiton entities.

An attribute within an abstract entity cannot be used on a page.



Newly created entities are Singleton by default. This setting can be changed in the properties panel, where it is also possible to create an abstract entity, or extend an abstract entity.

How to - use the entity editor as domain editor

In the entity editor in Blueriq Encore, multiple elements that together form part of the domain model comes together.

In this editor, it is possible to:

- Edit the selected entity
- Get an overview of and navigate to attributes, relations, and contextual entities
- Create new attributes within a given entity
- Create a new relation from the selected entity to an entity of your choice
- Create a new derived entity

Edit the selected entity

Editing the selected entity can be done in the properties panel at the right. Here, it is possible to change the base entity, choose whether or not the entity is a singleton, and/or an abstract entity.

Get an overview of and navigate to attributes, relations, and contextual entities

In the center of the canvas, the selected entity and its attributes are shown. If this entity is an extension of another entity, the base entity is shown above.

Left and right from the selected entity, the contextual entities and its relations to this entity are shown. Only the relation that points toward this other entity is shown. So, left from the selected entity, only reverse relations are shown and right from the selected entity, only forward relations are shown.

All entities, relations, and attributes shown in this editor:

- Can be navigated to
- Can be deleted using the menu (by clicking the three dots or right clicking the item)
- Can be duplicated using the menu (by clicking the three dots or right clicking the item)

Create new attributes within a given entity

New attributes can be created within the selected entity by clicking "+ New attribute", providing a name and click the checkmark or by hitting the enter button. By default, new attributes are of data type string. In the particular input field in this editor however, it is possible to provide the data type of the attribute as well. The syntax is: [name of the attribute],[data type]. Since the input is intelligent, there is no need to fully type the data type.

For example:

Input	Result
FirstName	An attribute named "FirstName" will be created within the selected entity. Since no data type is provided, it will be of data type string.
Age, integer	An attribute named "Age" will be created within the selected entity. The data type of the attribute is integer.
Age,i	An attribute named "Age" will be created within the selected entity. Blueriq Encore recognizes "i" as data type integer since it is distinctive enough. The data type of the attribute will therefore be integer.

Create a new relation from the selected entity to an entity of your choice

By clicking the "+ New relation"-button, a new relation is created between the previously selected entity and an entity that is to be selected in the dropdown. After selecting the entity in the dropdown, Blueriq Encore will automatically create a 1:1 relation. Blueriq Encore will automatically give a name to the forward and reverse relation that is identical to the entity the relation is pointing to. If desired, it is possible to manually change the name and properties of the relation.

Create a new derived entity

By clicking the "+ New derived entity"-button, providing a name, and clicking the checkmark or hitting the enter button, Blueriq Encore will create a new singleton entity that extends the previously selected entity (or in other words, it selects the previous selected entity as base entity).