

Глоссарий

Объектная модель Continuity (Object model) представляет собой иерархически организованную структуру с гибкой настройкой типов и вложенности объектов.

Объект (Object) - сущность системы Continuity, имеющая уникальный идентификатор набор определенных свойств. Объект может принимать тип проекта, задачи или артефакта. Объект является единицей иерархии, то есть может содержать в себе вложенные объекты либо быть дочерним по отношению к вышестоящей сущности.

Тип объекта (Type) - это определение объекта, регламентирующее его вид и правила взаимодействия с ним. В системе есть 3 типа объекта: Проект, Задача, Артефакт.

Граф Continuity (Graph) - это механизм для автоматизации бизнес-процессов. У него есть ребра, которые определяют направление движения workflow, и ноды, являющиеся контейнерами для объектов Continuity.

Подробнее о графе: [Граф Continuity](#)

Нода (Node) - это элемент графа и контейнер для объекта Continuity. Необходим для корректной работы автоматизации процессов.

Ребро графа (Edge) обеспечивает связь между узлами. По ребрам графа задается направление workflow.

Шаблон объекта (Template) - это описание объекта, его свойств, внутренней иерархии. Шаблон позволяет переиспользовать объекты более одного раза. Пользователю доступно настройка шаблона через Конструктор.

Подробнее о шаблоне: [Шаблон объекта](#)

Контекст объекта (Context) - свойство объекта/узла/ребра. С помощью контекстов можно настраивать тип объекта, его отображение, условия перехода в рамках workflow.

Подробнее о контекстах: [Контекст объекта](#)

Свимлейн (Swimlane) - это инструмент Continuity для настройки доступов к объектам.

Подробнее о ролевой модели: [Ролевая модель](#)

Ключ (Key) - это бизнес-идентификатор (то есть текстовый ключ) объекта в системе.

Тэг (Tag) - это название версии объекта.

Бизнес тэг (Business tag) - это метка, которая может быть добавлена к любому объекту системы для использования при фильтрации.

Continuity SDK - python-библиотека предоставляющая полный функционал для удобной работы с API в интерфейсе Jupyter.

Сервисная задача (Service task) - объект типа task, который позволяет настраивать автоматическое исполнение задачи на базе python-скрипта.

Подробнее о настройке сервисных задач: [Сервисные задачи](#)