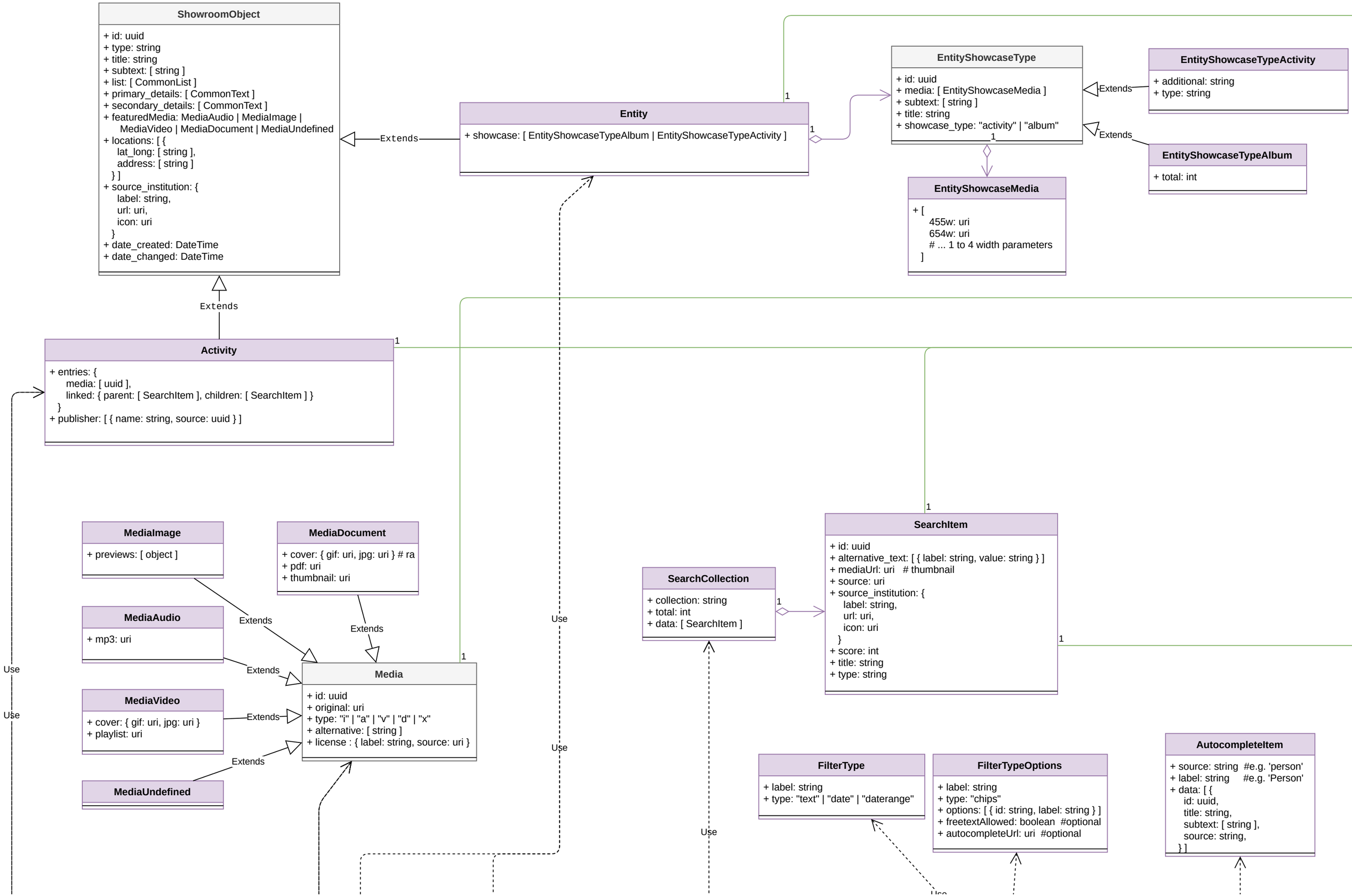
The Elaborated Data Quadrant



The Data Model Quadrant

SourceRepository

+ id: int
+ label_institution: string
+ label_repository: string
+ url_institution: uri
+ url_repository: uri
+ icon: uri
+ api_key: string

Entity

+ type: "person" | "institution" | "department"
+ expertise: [string]
+ showcase: [(uuid, type)] # referencing Activity and Album
+ photo: uri

parent, 0..1

Media

+ id: uuid
+ file: uri
+ type: "i" | "a" | "v" | "d" | "x"
+ activity_id: uuid
+ mime_type: string
+ exif: JSONField
+ license: JSONField
+ created: DateTime
+ modified: DateTime

AbstractBaseModel

+ date_created: DateTime
+ date_changed: DateTime

Extends

AbstractShowroomObject

+ id: uuid
+ title: string
+ list: [uri]
+ primary_details: [CommonText]
+ secondary_details: [CommonText]
+ locations: [{
    coordinates: [string],
    data: [string]
  }]
+ dates: [DateTime]
+ source_repo: int # foreign key
+ source_repo_entry_id: string # uuid for portfolio entries

belongs_to

Extends

Activity

+ type: JSONField # skosmos defined type
+ featured_media: MediaAudio | MediaImage |
  MediaVideo | MediaDocument | MediaUndefined

0..n, child

parent, 0..n

CommonListItem

+ attributes: [string]
+ value: string
+ id: uuid

CommonText

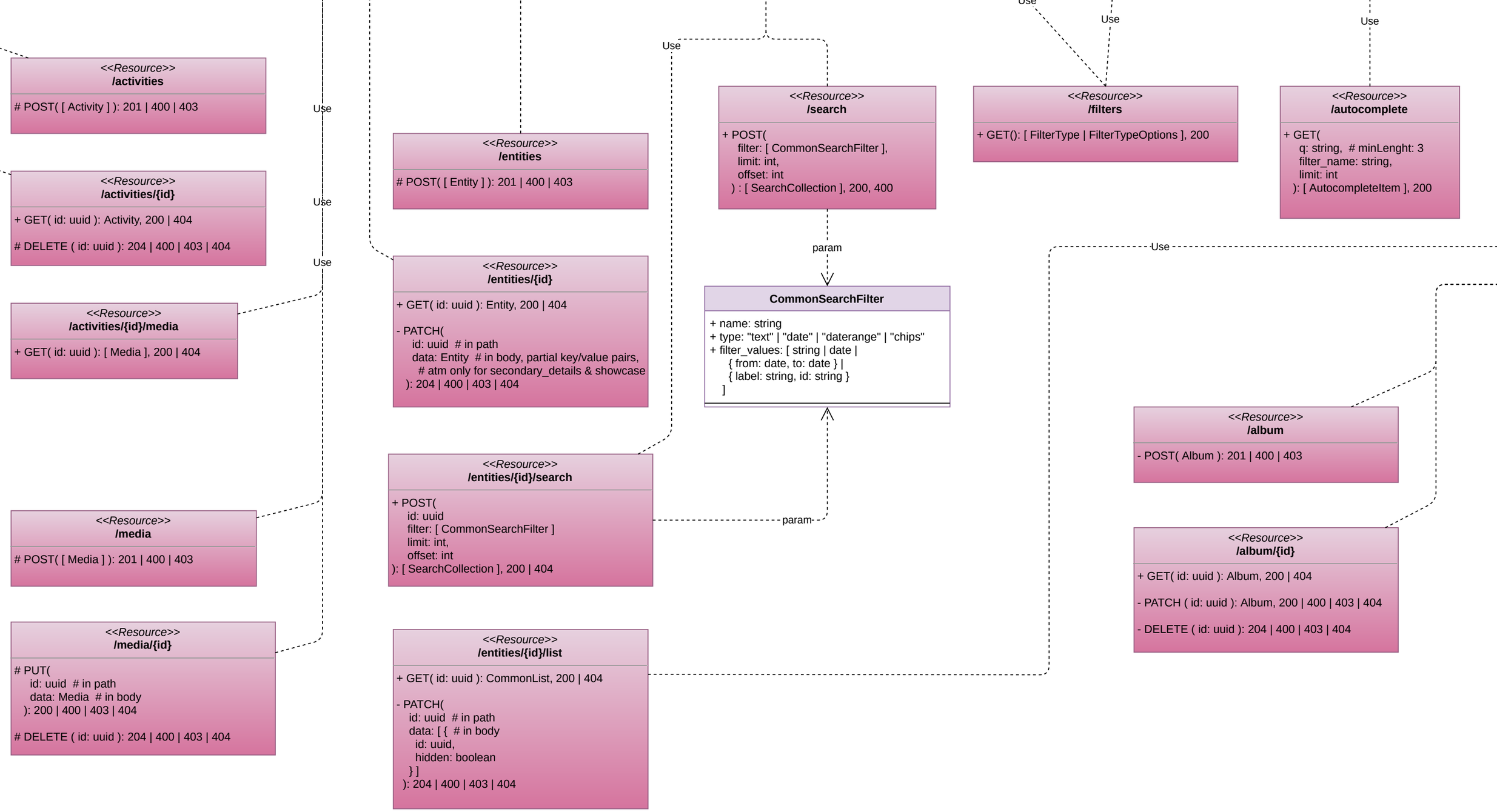
+ label: string
+ data: [string | [string] |
    {
      label: string,
      value: string,
      url: uri,
      source: string
    }
  ]

Album

+ id: uuid
+ title: string
+ subtitle: string
+ secondary_details: [CommonText]

CommonList

+ id: uuid
+ label: string
+ hidden: boolean
+ data: [CommonList | CommonListItem]



The API Quadrant

Use Use

Showroom Data Model

Version: 1.0.0 corresponds to API v1

Last updated: 2020-02-24

Authors: jackie <andrea.klaura@uni-aka.ac.at>

Legend

This is a derivation of a standard UML class diagramme to model Showroom's internal data model as well as the REST API that is served. REST resources are modeled as UML interfaces, the methods are either public (+), private (-) or protected (#). A private method needs to be authenticated with the base user auth. A protected method can be called by using an API key (e.g. for Portfolio to sync objects to showroom).

AbstractClassName
+ ...

An abstract class that is inherited by another abstract or a final model, as it is stored in the databse. If not prefixed with "Abstract" it can be used to generalize a schema type.

ModelName
+ ...

An actual model as it will be stored in the database and represented as Django model.

SchemaType
+ ...

A schema or template for data, that is either aggregated in acutal objects as a JSON string, generated dynamically for output in the API, or used in a request parameter

<<Resource>> **/endpoint**
+ GET(param: type): ResultType, StatusCodes - POST(param: type): ResultType, StatusCodes

A REST resource in form of an endpoint, with the available methods, denoting the parameters as well as the type of result and the status codes.

Extends

A generalization/implementation relationshop. A extends B means that A is a subclass of B.

Use

Points to the data model or schema that an API endpoint uses to provide its response.

1

name

Points to the data model that a schema uses to transform/elaborate the backend data for the API endpoints.

1

Aggregation relationship, where an object incorporates 0 to many other objects. Can be either modeled as a list (m:n) or as a foreign key (1:n).

General notes:
- All endpoints are relativ to the base path for version 1 (/api/v1).
- A uuid in showroom can be formatted as a ShortUUID (this is not fully specified yet)
- A generalized schema type used by an endpoint refers always to one of its specifications.
- If a parameter or contents of an array can consist of different types, these are separated by "|" (pipe)