



MongoDB Physical Model

Schema for:

Model name: FightPandemics
Author: Nicolas de Araujo
Version: 3
File name: data_model.json
File path: U:\home\naraujo\Dev\FightPandemics\DataModel\v3\data_model.json
Printed On: Mon Sep 07 2020 22:13:33 GMT-0300 (Horário Padrão de Brasília)
Created with: [Hackolade](#) - Visual data modeling for NoSQL and multimodel databases

1. Model

2. Databases

2.1 applicationDB

2.1.2. Collections

- 2.1.2.1 comments
- 2.1.2.2 feedbacks
- 2.1.2.3 messages
- 2.1.2.4 posts
- 2.1.2.5 thread
- 2.1.2.6 users

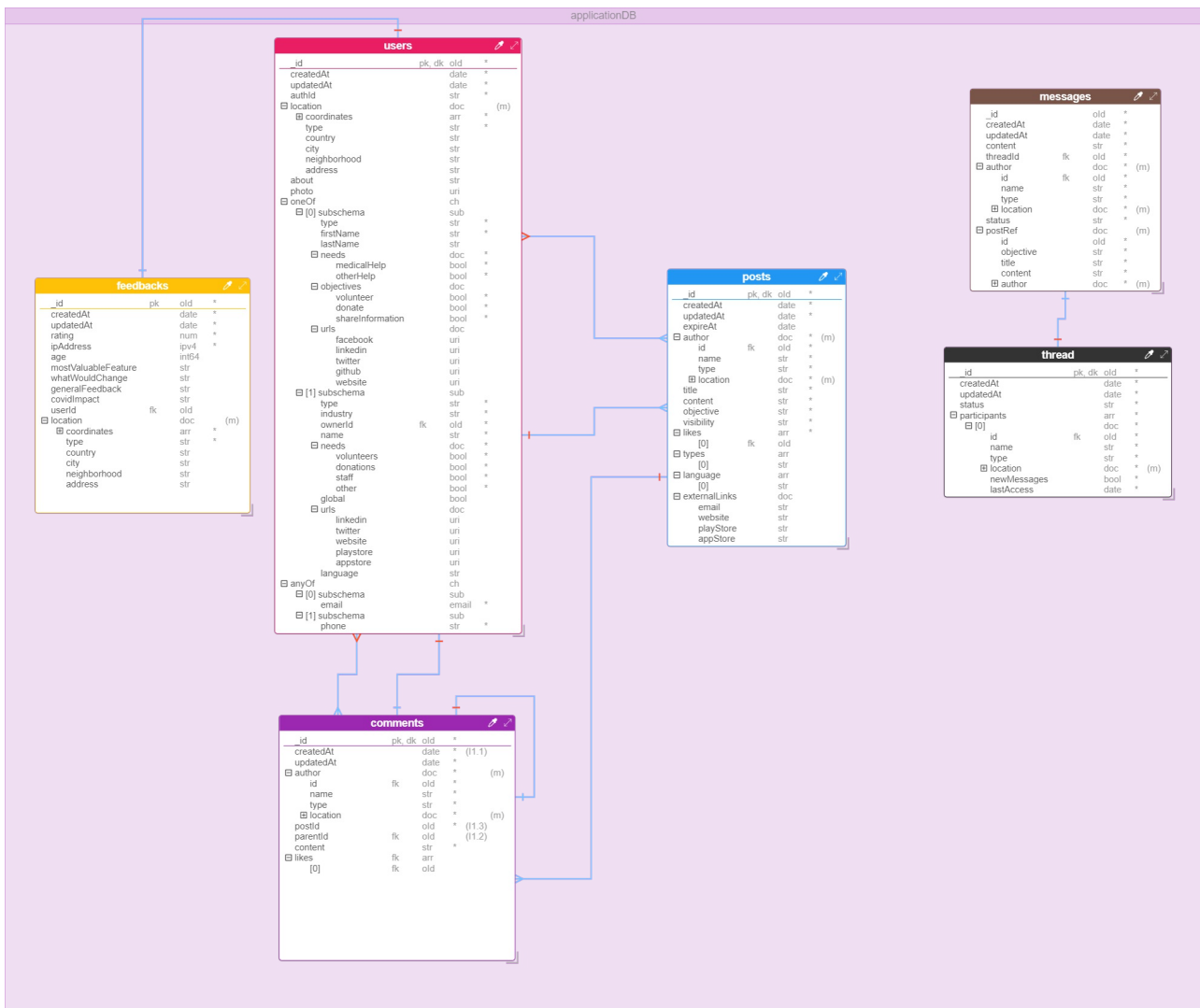
3. Relationships

- 3.1 CommentReferencesPost
- 3.2 PostHasComments
- 3.3 ThreadHasMessages
- 3.4 UserAuthorsComment
- 3.5 UserAuthorsPost
- 3.6 UserGivesFeedback
- 3.7 UserLikesComment
- 3.8 UserLikesPost
- 3.9 UserOwnsOrganization

1. MODEL

1.1 Model FightPandemics

1.1.1 FightPandemics Entity Relationship Diagram



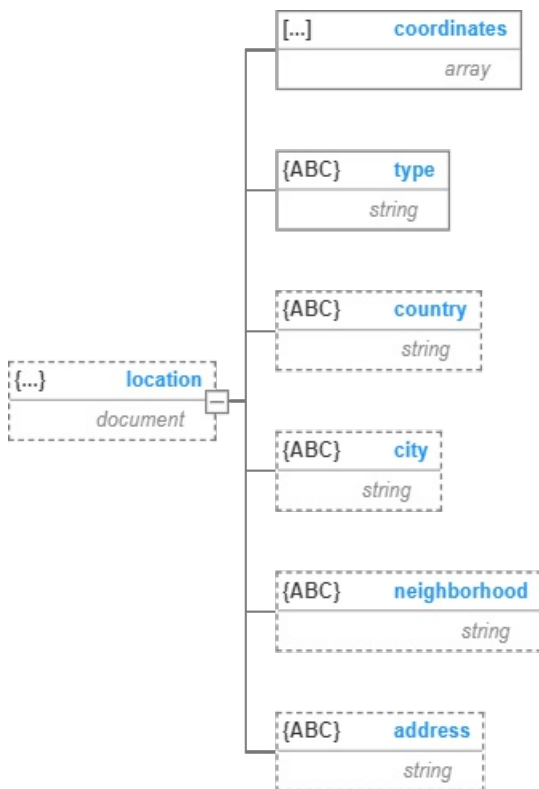
1.1.2 FightPandemics Properties

PROPERTY	VALUE
Model name	FightPandemics
Technical name	FightPandemics
Description	
Author	Nicolas de Araujo
Version	3
DB vendor	MongoDB
DB version	v4.2
Comments	

1.1.3 FightPandemics DB Definitions

1.1.3.1 Field location

1.1.3.1.1 location Tree Diagram



1.1.3.1.2 location Hierarchy

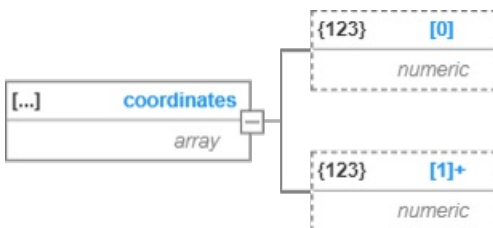
Parent field: **Definitions**

Child field(s):

FIELD	TYPE	REQ	KEY	DESCRIPTION	COMMENTS
coordinates	array	true		Array with the coordinates, specifying longitude first and latitude second , as default in the GeoJSON definition.	
type	string	true		Type of the GeoJSON Object. Please refer to the official documentation for more information. In our case, the value will always be of the type "Point".	
country	string	false		String holding the country's name.	
city	string	false		String holding the city's name.	
neighborhood	string	false		String holding the neighborhood's name.	
address	string	false		String holding the address (street, number and other details).	

1.1.3.1.3 location properties

PROPERTY	VALUE
Name	location
Technical name	
Id	
Type	document
Description	Location document according to the norms of a GeoJSON Object . The fields address, neighborhood, city, state and country are added to store the result of the geoprocessing.
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
DBRef	
Min Properties	
Max Properties	
Additional properties	false
Comments	

1.1.3.2 Field coordinates**1.1.3.2.1 coordinates Tree Diagram****1.1.3.2.2 coordinates Hierarchy**Parent field: **location**

Child field(s):

FIELD	TYPE	REQ	KEY	DESCRIPTION	COMMENTS
[0]	numeric	false		Location longitude	
[1]	numeric	false		Location latitude	

1.1.3.2.3 coordinates properties

PROPERTY	VALUE
Name	coordinates
Technical name	
Id	
Type	array
Description	Array with the coordinates, specifying longitude first and latitude second , as default in the GeoJSON definition.
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Min items	2
Max items	2
Unique items	false
Additional items	true
Comments	

1.1.3.3 Field [0]**1.1.3.3.1 [0] Tree Diagram****1.1.3.3.2 [0] properties**

PROPERTY	VALUE
Display name	
Id	lng
Type	numeric
Subtype	
Description	Location longitude
Dependencies	
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	0
Unit	degrees
Min value	-180
Excl min	false
Max value	180
Excl max	false
Multiple of	
Divisible by	
Enum	
Sample	35.26
Comments	

1.1.3.4 Field [1]

1.1.3.4.1 [1] Tree Diagram



1.1.3.4.2 [1] properties

PROPERTY	VALUE
Display name	
Id	lat
Type	numeric
Subtype	
Description	Location latitude
Dependencies	
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	0
Unit	degrees
Min value	-90
Excl min	false
Max value	90
Excl max	false
Multiple of	
Divisible by	
Enum	
Sample	-32.83
Comments	

1.1.3.5 Field type

1.1.3.5.1 type Tree Diagram



1.1.3.5.2 type properties

PROPERTY	VALUE
Name	type
Technical name	
Id	
Type	string
Description	Type of the GeoJSON Object. Please refer to the official documentation for more information. In our case, the value will always be of the type "Point".
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	Point
Sample	
Comments	

1.1.3.6 Field country

1.1.3.6.1 country Tree Diagram



1.1.3.6.2 country properties

PROPERTY	VALUE
Name	country
Technical name	
Id	
Type	string
Description	String holding the country's name.
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	
Sample	
Comments	

1.1.3.7 Field city

1.1.3.7.1 city Tree Diagram



1.1.3.7.2 city properties

PROPERTY	VALUE
Name	city
Technical name	
Id	
Type	string
Description	String holding the city's name.
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	
Sample	
Comments	

1.1.3.8 Field neighborhood

1.1.3.8.1 neighborhood Tree Diagram



1.1.3.8.2 neighborhood properties

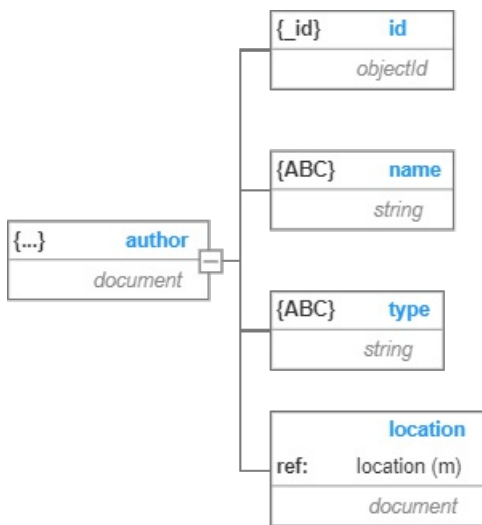
PROPERTY	VALUE
Name	neighborhood
Technical name	
Id	
Type	string
Description	String holding the neighborhood's name.
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	
Sample	
Comments	

1.1.3.9 Field address**1.1.3.9.1 address Tree Diagram****1.1.3.9.2 address properties**

PROPERTY	VALUE
Name	address
Technical name	
Id	
Type	string
Description	String holding the address (street, number and other details).
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	
Sample	
Comments	

1.1.3.10 Field author

1.1.3.10.1 author Tree Diagram



1.1.3.10.2 author Hierarchy

Parent field: **Definitions**

Child field(s):

FIELD	TYPE	REQ	KEY	DESCRIPTION	COMMENTS
id	objectId	true	fk	Foreign key to the user who created the post.	
name	string	true			
type	string	true			
location	reference	true			

1.1.3.10.3 author properties

PROPERTY	VALUE
Name	author
Technical name	
Id	
Type	document
Description	
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
DBRef	
Min Properties	
Max Properties	
Additional properties	false
Comments	

1.1.3.11 Field id

1.1.3.11.1 id Tree Diagram



1.1.3.11.2 id properties

PROPERTY	VALUE
Name	id
Technical name	
Id	
Type	objectId
Description	Foreign key to the user who created the post.
Dependencies	
Required	true
Primary key	false
Foreign collection	users
Foreign field	_id
Relationship type	Foreign Key
Default	
Sample	
Comments	

1.1.3.12 Field name

1.1.3.12.1 name Tree Diagram

{ABC}	name
	string

1.1.3.12.2 name properties

PROPERTY	VALUE
Name	name
Technical name	
Id	
Type	string
Description	
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	
Sample	
Comments	

1.1.3.13 Field type

1.1.3.13.1 type Tree Diagram

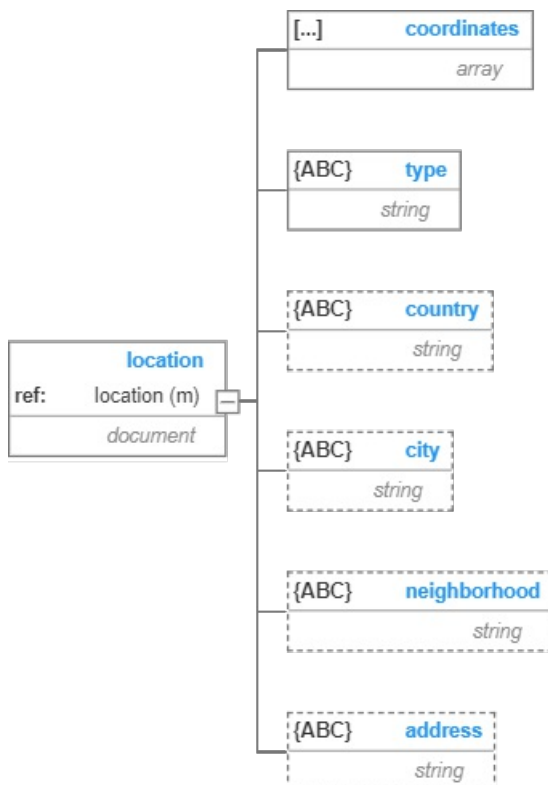
{ABC}	type
	string

1.1.3.13.2 type properties

PROPERTY	VALUE
Name	type
Technical name	
Id	
Type	string
Description	
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	
Sample	
Comments	

1.1.3.14 Field location

1.1.3.14.1 location Tree Diagram

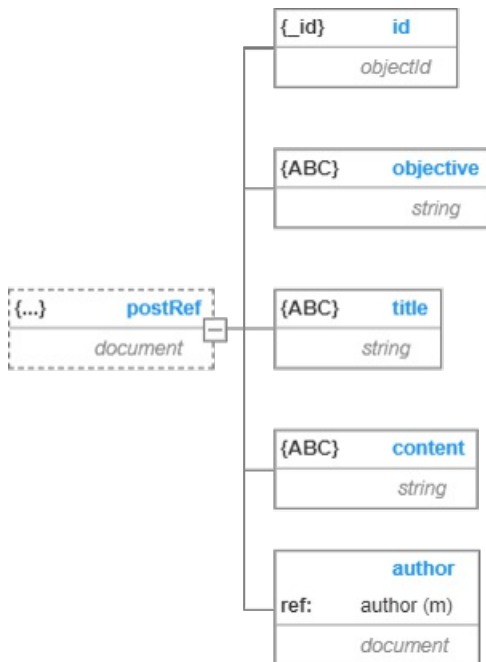


1.1.3.14.2 location properties

PROPERTY	VALUE
Name	location
Technical name	
\$ref	#model/definitions/location
Reference type	model
Reference description	

1.1.3.15 Field postRef

1.1.3.15.1 postRef Tree Diagram



1.1.3.15.2 postRef Hierarchy

Parent field: **Definitions**

Child field(s):

FIELD	TYPE	REQ	KEY	DESCRIPTION	COMMENTS
id	objectId	true			
objective	string	true		Field to indicate the post's objective. Currently only two values allowed: "offering help" and "looking for help".	
title	string	true		The post title.	
content	string	true		String for the post content.	
author	reference	true			

1.1.3.15.3 postRef properties

PROPERTY	VALUE
Name	postRef
Technical name	
Id	
Type	document
Description	Optional post reference, allowing a post to be referenced in a message thread.
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
DBRef	
Min Properties	
Max Properties	
Additional properties	false
Comments	

1.1.3.16 Field id**1.1.3.16.1 id Tree Diagram****1.1.3.16.2 id properties**

PROPERTY	VALUE
Name	id
Technical name	
Id	
Type	objectId
Description	
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Sample	
Comments	

1.1.3.17 Field objective**1.1.3.17.1 objective Tree Diagram**

1.1.3.17.2 objective properties

PROPERTY	VALUE
Name	objective
Technical name	
Id	
Type	string
Description	Field to indicate the post's objective. Currently only two values allowed: "offering help" and "looking for help".
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	request,offer
Sample	offer
Comments	

1.1.3.18 Field title**1.1.3.18.1 title Tree Diagram**

{ABC}	title
	string

1.1.3.18.2 title properties

PROPERTY	VALUE
Name	title
Technical name	
Id	
Type	string
Description	The post title.
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	
Sample	
Comments	

1.1.3.19 Field content

1.1.3.19.1 content Tree Diagram

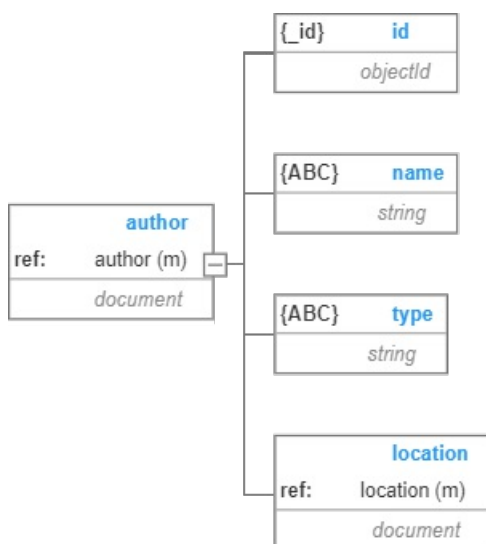


1.1.3.19.2 content properties

PROPERTY	VALUE
Name	content
Technical name	
Id	
Type	string
Description	String for the post content.
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	
Sample	
Comments	

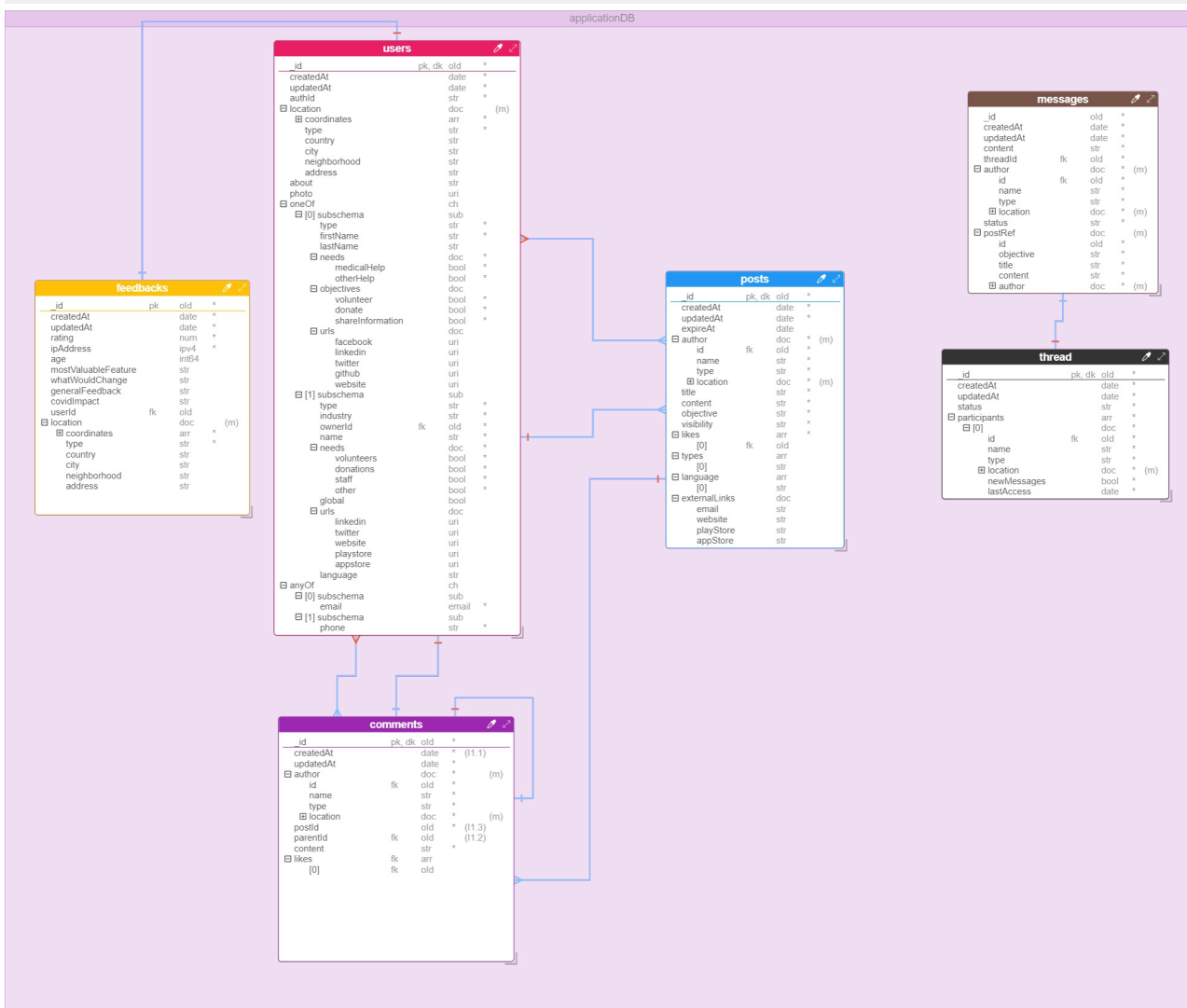
1.1.3.20 Field author

1.1.3.20.1 author Tree Diagram



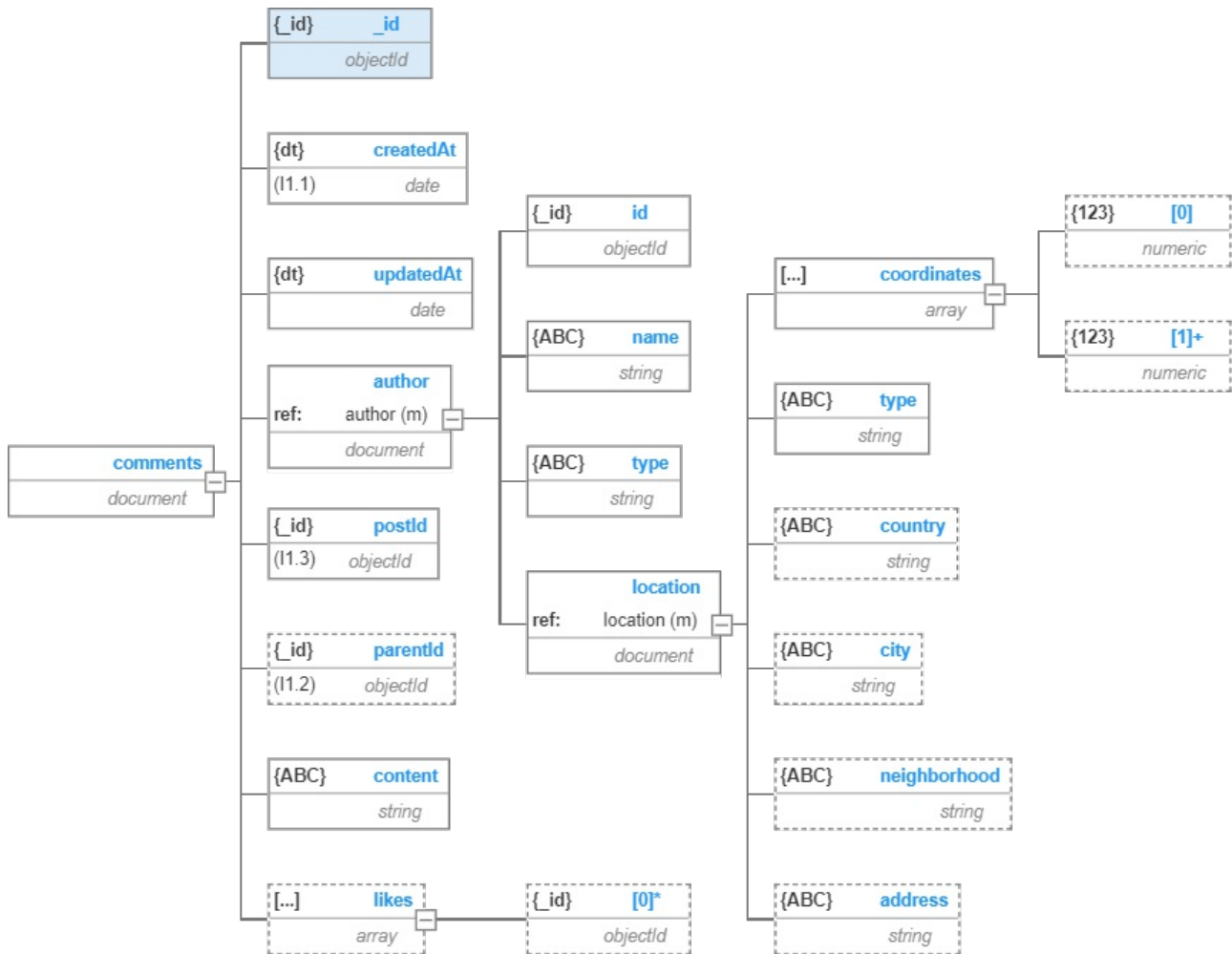
PROPERTY	VALUE
Name	author
Technical name	
\$ref	#model/definitions/author
Reference type	model
Reference description	

2.1 Database applicationDB



PROPERTY	VALUE
Database name	applicationDB
Technical name	
Enable sharding	
Description	
Comments	

2.1.2.1.1 comments Tree Diagram



2.1.2.1.2 comments Properties

PROPERTY	VALUE
Collection name	comments
Technical name	
Id	
Description	
Database	applicationDB
Capped	false
Size	
Max	
Storage engine	WiredTiger
Config String	
Validation level	Off
Validation action	Warn
Additional properties	false
Comments	

2.1.2.1.3 comments Fields

FIELD	TYPE	REQ	KEY	DESCRIPTION	COMMENTS
_id	objectId	true		Automatically generated object_id as primary key.	
createdAt	date	true		Timestamp with the comment's creation date	
updatedAt	date	true		Timestamp with the comment's last update date	
author	document	true			
id	objectId	true		Foreign key to the user who created the post.	
name	string	true			
type	string	true			
location	document	true		Location document according to the norms of a GeoJSON Object . The fields address, neighborhood, city, state and country are added to store the result of the geoprocessing.	
coordinates	array	true		Array with the coordinates, specifying longitude first and latitude second , as default in the GeoJSON definition.	
[0]	numeric	false		Location longitude	
[1]	numeric	false		Location latitude	
type	string	true		Type of the GeoJSON Object. Please refer to the official documentation for more information. In our case, the value will always be of the type "Point".	
country	string	false		String holding the country's name.	
city	string	false		String holding the city's name.	
neighborhood	string	false		String holding the neighborhood's name.	
address	string	false		String holding the address (street, number and other details).	
postId	objectId	true			
parentId	objectId	false		If the comment is nested (in another comment), this field will hold the _id of the parent comment. Optional field, where being blank means it is a root comment.	
content	string	true		String holding the comment's content.	
likes	array	false		Array holding the user_id of all the users who liked the comment.	
[0]	objectId	false		_id of the user who liked the comment	

2.1.2.1.3.1 Field _id

2.1.2.1.3.1.1 _id Tree Diagram

{_id}	_id
	<i>objectId</i>

2.1.2.1.3.1.2 _id properties

PROPERTY	VALUE
Name	_id
Technical name	
Id	
Type	objectId
Description	Automatically generated object_id as primary key.
Dependencies	
Required	true
Primary key	true
Foreign collection	
Foreign field	
Relationship type	
Default	
Sample	
Comments	

2.1.2.1.3.2 Field createdAt

2.1.2.1.3.2.1 createdAt Tree Diagram

{dt}	createdAt
(11.1)	<i>date</i>

2.1.2.1.3.2.2 createdAt properties

PROPERTY	VALUE
Name	createdAt
Technical name	
Id	
Type	date
Description	Timestamp with the comment's creation date
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Sample	
Comments	

2.1.2.1.3.3 Field updatedAt

2.1.2.1.3.3.1 updatedAt Tree Diagram

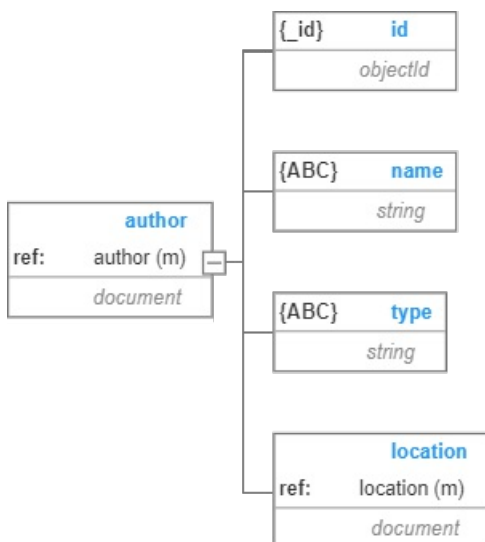
{dt}	updatedAt
	<i>date</i>

2.1.2.1.3.3.2 updatedAt properties

PROPERTY	VALUE
Name	updatedAt
Technical name	
Id	
Type	date
Description	Timestamp with the comment's last update date
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Sample	
Comments	

2.1.2.1.3.4 Field author

2.1.2.1.3.4.1 author Tree Diagram



2.1.2.1.3.4.2 author properties

PROPERTY	VALUE
Name	author
Technical name	
\$ref	#model/definitions/author
Reference type	model
Reference description	

2.1.2.1.3.5 Field postId

2.1.2.1.3.5.1 postId Tree Diagram



2.1.2.1.3.5.2 postId properties

PROPERTY	VALUE
Name	postId
Technical name	
Id	
Type	objectId
Description	
Dependencies	
Required	true
Primary key	false
Foreign collection	posts
Foreign field	
Relationship type	
Default	
Sample	
Comments	

2.1.2.1.3.6 Field parentId

2.1.2.1.3.6.1 parentId Tree Diagram



2.1.2.1.3.6.2 parentId properties

PROPERTY	VALUE
Name	parentId
Technical name	
Id	
Type	objectId
Description	If the comment is nested (in another comment), this field will hold the _id of the parent comment. Optional field, where being blank means it is a root comment.
Dependencies	
Required	false
Primary key	false
Foreign collection	comments
Foreign field	_id
Relationship type	Foreign Key
Default	
Sample	
Comments	

2.1.2.1.3.7 Field content

2.1.2.1.3.7.1 content Tree Diagram



2.1.2.1.3.7.2 content properties

PROPERTY	VALUE
Name	content
Technical name	
Id	
Type	string
Description	String holding the comment's content.
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	
Sample	
Comments	

2.1.2.1.3.8 Field likes

2.1.2.1.3.8.1 likes Tree Diagram



2.1.2.1.3.8.2 likes Hierarchy

Parent field: **comments**

Child field(s):

FIELD	TYPE	REQ	KEY	DESCRIPTION	COMMENTS
[0]	objectId	false	fk	_id of the user who liked the comment	

2.1.2.1.3.8.3 likes properties

PROPERTY	VALUE
Name	likes
Technical name	
Id	
Type	array
Description	Array holding the user_id of all the users who liked the comment.
Dependencies	
Required	false
Primary key	false
Foreign collection	users
Foreign field	_id
Relationship type	Foreign Key
Min items	
Max items	
Unique items	true
Additional items	true
Comments	

2.1.2.1.3.9 Field [0]**2.1.2.1.3.9.1 [0] Tree Diagram****2.1.2.1.3.9.2 [0] properties**

PROPERTY	VALUE
Display name	
Id	
Type	objectId
Description	_id of the user who liked the comment
Dependencies	
Primary key	false
Foreign collection	
Foreign field	
Relationship type	Foreign Key
Default	
Sample	
Comments	

2.1.2.1.4 comments Indexes

PROPERTY	NEW INDEX
Name	New Index
Key	createdAt('ascending'), parentId('ascending'), postId('ascending')
Hashed	
Unique	
Drop duplicates	
Sparse	
Background indexing	
Partial filter exp	
Expire after (seconds)	
Storage engine	WiredTiger
Comments	

2.1.2.1.5 comments JSON Schema

```

{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "comments",
  "additionalProperties": false,
  "properties": {
    "_id": {
      "type": "string",
      "description": "Automatically generated object_id as primary key",
      "pattern": "^[a-fA-F0-9]{24}$"
    },
    "createdAt": {
      "type": "string",
      "description": "Timestamp with the comment's creation date",
      "format": "date-time"
    },
    "updatedAt": {
      "type": "string",
      "description": "Timestamp with the comment's last update date",
      "format": "date-time"
    },
    "author": {
      "$ref": "#model/definitions/author"
    },
    "postId": {
      "type": "string",
      "pattern": "^[a-fA-F0-9]{24}$"
    },
    "parentId": {
      "type": "string",
      "description": "If the comment is nested (in another comment), this field will hold the _id of the parent comment. Optional field, where being blank means it is a root comment.",
      "pattern": "^[a-fA-F0-9]{24}$"
    },
    "content": {
      "type": "string",
      "description": "String holding the comment's content."
    },
    "likes": {
      "type": "array",
      "description": "Array holding the user_id of all the users who liked the comment.",
      "additionalItems": true,
      "uniqueItems": true,
      "items": {
        "type": "string",
        "description": "_id of the user who liked the comment",
        "pattern": "^[a-fA-F0-9]{24}$"
      }
    }
  },
  "required": [
    "_id",
    "createdAt",
    "updatedAt",
    "author",
    "postId",
    "content"
  ]
}

```

2.1.2.1.6 comments JSON data

```
{
  "_id": ObjectId("507f1f77bcf86cd799439011"),
  "createdAt": ISODate("2016-04-08T15:06:21.595Z"),
  "updatedAt": ISODate("2016-04-08T15:06:21.595Z"),
  "author": {
    "id": ObjectId("507f1f77bcf86cd799439011"),
    "name": "Lorem",
    "type": "Lorem",
    "location": {
      "coordinates": [
        35.26,
        -32.83
      ],
      "type": "Point",
      "country": "Lorem",
      "city": "Lorem",
      "neighborhood": "Lorem",
      "address": "Lorem"
    }
  },
  "postId": ObjectId("507f1f77bcf86cd799439011"),
  "parentId": ObjectId("507f1f77bcf86cd799439011"),
  "content": "Lorem",
  "likes": [
    ObjectId("507f1f77bcf86cd799439011")
  ]
}
```

2.1.2.1.7 comments Target Script

```
use applicationDB;

db.createCollection( "comments",{
  "storageEngine": {
    "wiredTiger": {}
  },
  "capped": false,
  "validator": {
    "$jsonSchema": {
      "bsonType": "object",
      "title": "comments",
      "additionalProperties": false,
      "properties": {
        "_id": {
          "bsonType": "objectId",
          "description": "Automatically generated object_id as primary key."
        },
        "createdAt": {
          "bsonType": "date",
          "description": "Timestamp with the comment's creation date"
        },
        "updatedAt": {
          "bsonType": "date",
          "description": "Timestamp with the comment's last update date"
        },
        "author": {
          "bsonType": "object",
          "properties": {
            "id": {
              "bsonType": "objectId",
              "description": "Foreign key to the user who created the post."
            },
            "name": {
              "bsonType": "string"
            }
          }
        }
      }
    }
  }
})
```

```

    },
    "type": {
      "bsonType": "string"
    },
    "location": {
      "bsonType": "object",
      "description": "Location document according to the norms of a [GeoJSON Object]
      (https://docs.mongodb.com/manual/reference/geojson/ ).\n\nThe fields address, neighborhood, city, state and country are added to store the
      result of the geoprocessing.",
      "properties": {
        "coordinates": {
          "bsonType": "array",
          "description": "Array with the coordinates, specifying **longitude first** and **latitude second**, as default in the [GeoJSON]
          (https://docs.mongodb.com/manual/reference/geojson/) definition.",
          "additionalItems": true,
          "minItems": 2,
          "maxItems": 2,
          "uniqueItems": false,
          "items": [
            {
              "bsonType": "number",
              "description": "Location longitude\n",
              "minimum": -180,
              "maximum": 180
            },
            {
              "bsonType": "number",
              "description": "Location latitude\n",
              "minimum": -90,
              "maximum": 90
            }
          ]
        },
        "type": {
          "bsonType": "string",
          "description": "Type of the GeoJSON Object.\n\nPlease refer to the [official documentation]
          (https://docs.mongodb.com/manual/reference/geojson/ ) for more information.\n\n**In our case, the value will always be of the type 'Point'.**",
          "enum": [
            "Point"
          ]
        },
        "country": {
          "bsonType": "string",
          "description": "String holding the country's name.\n"
        },
        "city": {
          "bsonType": "string",
          "description": "String holding the city's name."
        },
        "neighborhood": {
          "bsonType": "string",
          "description": "String holding the neighborhood's name."
        },
        "address": {
          "bsonType": "string",
          "description": "String holding the address (street, number and other details)."
        }
      },
      "additionalProperties": false,
      "required": [
        "coordinates",
        "type"
      ]
    },
    "id",
    "name",
    "type"
  ]
}

```

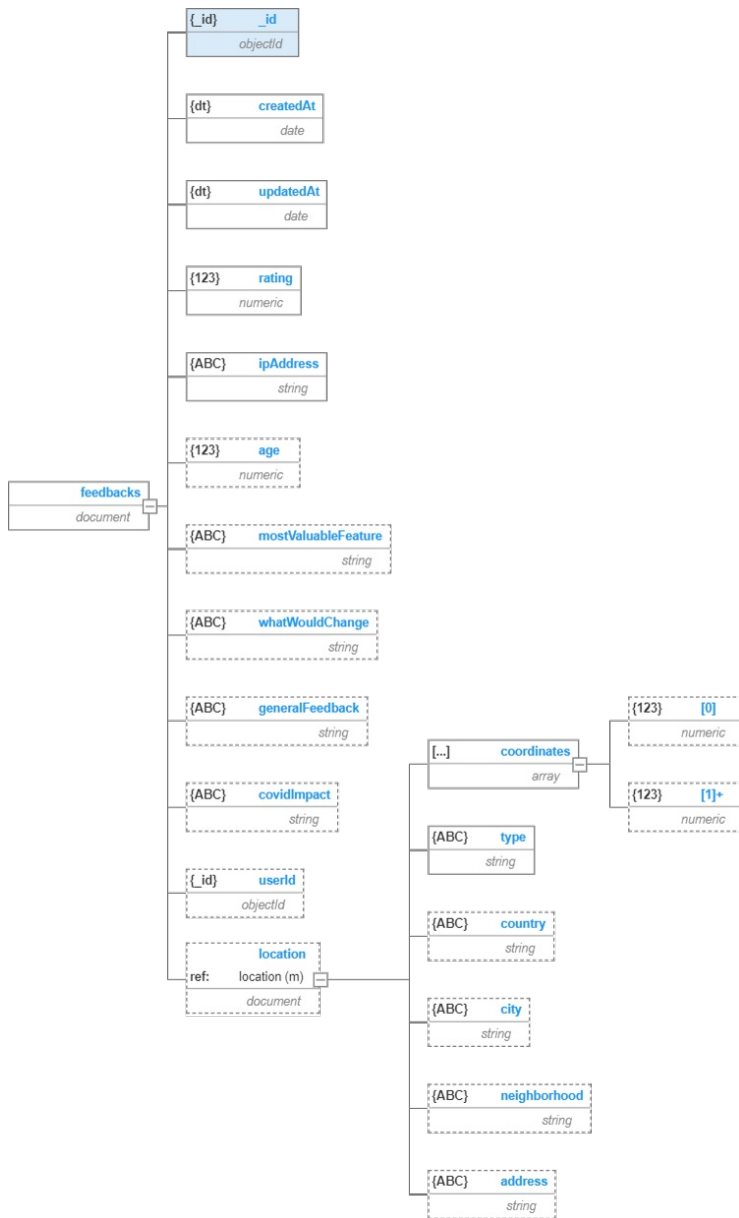
```

    ]
  },
  "postId": {
    "bsonType": "objectId"
  },
  "parentid": {
    "bsonType": "objectId",
    "description": "If the comment is nested (in another comment), this field will hold the _id of the parent comment. Optional field, where being blank means it is a root comment."
  },
  "content": {
    "bsonType": "string",
    "description": "String holding the comment's content."
  },
  "likes": {
    "bsonType": "array",
    "description": "Array holding the user_id of all the users who liked the comment.",
    "additionalItems": true,
    "uniqueItems": true,
    "items": {
      "bsonType": "objectId",
      "description": "_id of the user who liked the comment"
    }
  }
},
"required": [
  "_id",
  "createdAt",
  "updatedAt",
  "author",
  "postId",
  "content"
]
}
},
"validationLevel": "off",
"validationAction": "warn"
});
db.comments.createIndex(
{
  "createdAt": 1,
  "parentid": 1,
  "postId": 1
},
{
  "name": "New Index"
}
);

```

2.1.2.2 Collection feedbacks

2.1.2.2.1 feedbacks Tree Diagram



2.1.2.2.2 feedbacks Properties

PROPERTY	VALUE
Collection name	feedbacks
Technical name	
Id	
Description	Collection to hold feedback on the application. This may be done with the user logged or not and, depending on that, we may or may not have a reference to the user in question at this model.
Database	applicationDB
Capped	false
Size	
Max	
Storage engine	WiredTiger
Config String	
Validation level	Off
Validation action	Warn
Additional properties	false
Comments	

2.1.2.2.3 feedbacks Fields

FIELD	TYPE	REQ	KEY	DESCRIPTION	COMMENTS
_id	objectId	true		Automatically generated object_id as primary key.	
createdAt	date	true		Timestamp with the feedback's creation date	
updatedAt	date	true		Timestamp with the feedback's last update date	
rating	numeric	true		A rating, in a scale of 1 to 5, of the application. Currently present as: "How well does FightPandemics meet your needs?". Present in the first part of the feedback and, therefore, is required.	
ipAddress	ipv4	true		Captured IP address in order to get user's location even if they are not logged. Captured at the first part of the feedback and, therefore, is required.	
age	integer64	false		User's age. Currently present as: "What is your age?". Present in the third part of the feedback and, therefore, not required.	
mostValuableFeature	string	false		User's feedback into what is the features most valuable for him/her. Currently present as: "Which features are the most valuable for you?". Present in the second part of the feedback and, therefore, not	

				required.	
whatWouldChange	string	false		<p>User's feedback into what he/she would change about the application.</p> <p>Currently present as: "If you could change one thing about FightPandemics, what would it be?".</p> <p>Present in the second part of the feedback and, therefore, not required.</p>	
generalFeedback	string	false		<p>User's general feedback section.</p> <p>Currently present as: "Any other feedback for us?".</p> <p>Present in the second part of the feedback and, therefore, not required.</p>	
covidImpact	string	false		<p>User's feedback into how the COVID impacted him.</p> <p>Currently present as a multiple choice question.</p> <p>Question: "How has COVID-19 impacted you?"</p> <p>Options:</p> <ol style="list-style-type: none"> 1. I go to work/school normally 2. I am healthy but in a stay-at-home quarantine 3. I have mild symptoms but haven't been tested 4. I am diagnosed with COVID-19 <p>Present in the second part of the feedback and, therefore, not required.</p>	
userId	objectId	false		<p>Reference to the user's <code>_id</code>.</p> <p>This will be present only if the user was logged in when completing the feedback.</p>	
location	document	false		<p>Location document according to the norms of a GeoJSON Object. The fields address, neighborhood, city, state and country are added to store the result of the geoprocessing.</p>	
coordinates	array	true		<p>Array with the coordinates, specifying longitude first and latitude second, as default in the GeoJSON definition.</p>	
[0]	numeric	false		Location longitude	
[1]	numeric	false		Location latitude	
type	string	true		<p>Type of the GeoJSON Object. Please refer to the official documentation for more information.</p> <p>In our case, the value will always be of the type "Point".</p>	
country	string	false		String holding the country's name.	
city	string	false		String holding the city's name.	
neighborhood	string	false		String holding the neighborhood's name.	

address	string	false		String holding the address (street, number and other details).	
---------	--------	-------	--	--	--

2.1.2.2.3.1 Field createdAt

2.1.2.2.3.1.1 createdAt Tree Diagram

{dt}	createdAt
	date

2.1.2.2.3.1.2 createdAt properties

PROPERTY	VALUE
Name	createdAt
Technical name	
Id	
Type	date
Description	Timestamp with the feedback's creation date
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Sample	
Comments	

2.1.2.2.3.2 Field _id

2.1.2.2.3.2.1 _id Tree Diagram

{_id}	_id
	objectId

2.1.2.2.3.2.2 _id properties

PROPERTY	VALUE
Name	_id
Technical name	
Id	
Type	objectId
Description	Automatically generated object_id as primary key.
Dependencies	
Required	true
Primary key	true
Foreign collection	
Foreign field	
Relationship type	
Default	
Sample	
Comments	

2.1.2.2.3.3 Field updatedAt

2.1.2.2.3.3.1 updatedAt Tree Diagram

{dt}	updatedAt
	date

2.1.2.2.3.3.2 updatedAt properties

PROPERTY	VALUE
Name	updatedAt
Technical name	
Id	
Type	date
Description	Timestamp with the feedback's last update date
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Sample	
Comments	

2.1.2.2.3.4 Field rating**2.1.2.2.3.4.1 rating Tree Diagram**

{123}	rating
	numeric

2.1.2.2.3.4.2 rating properties

PROPERTY	VALUE
Name	rating
Technical name	
Id	
Type	numeric
Subtype	
Description	<p>A rating, in a scale of 1 to 5, of the application.</p> <p>Currently present as: "How well does FightPandemics meet your needs?".</p> <p>Present in the first part of the feedback and, therefore, is required.</p>
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Unit	
Min value	1
Excl min	false
Max value	5
Excl max	false
Multiple of	
Divisible by	
Enum	
Sample	
Comments	

2.1.2.2.3.5 Field ipAddress**2.1.2.2.3.5.1 ipAddress Tree Diagram**

{ABC}	ipAddress
	string

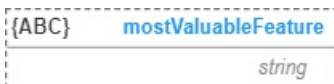
2.1.2.2.3.5.2 ipAddress properties

PROPERTY	VALUE
Name	ipAddress
Technical name	
Id	
Type	string
Description	Caputred IP address in order to get user's location even if they are not logged. Captured at the first part of the feedback and, therefore, is required.
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	ipv4
Enum	
Sample	
Comments	

2.1.2.2.3.6 Field age**2.1.2.2.3.6.1 age Tree Diagram**

2.1.2.2.3.6.2 age properties

PROPERTY	VALUE
Name	age
Technical name	
Id	
Type	numeric
Subtype	integer64
Description	User's age. Currently present as: "What is your age?". Present in the third part of the feedback and, therefore, not required.
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Unit	
Min value	18
Excl min	false
Max value	
Excl max	false
Multiple of	
Divisible by	
Enum	
Sample	
Comments	

2.1.2.2.3.7 Field mostValuableFeature**2.1.2.2.3.7.1 mostValuableFeature Tree Diagram**

2.1.2.2.3.7.2 mostValuableFeature properties

PROPERTY	VALUE
Name	mostValuableFeature
Technical name	
Id	
Type	string
Description	User's feedback into what is the features most valuable for him/her. Currently present as: "Which features are the most valuable for you?". Present in the second part of the feedback and, therefore, not required.
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	
Sample	
Comments	

2.1.2.2.3.8 Field whatWouldChange**2.1.2.2.3.8.1 whatWouldChange Tree Diagram**

2.1.2.2.3.8.2 whatWouldChange properties

PROPERTY	VALUE
Name	whatWouldChange
Technical name	
Id	
Type	string
Description	User's feedback into what he/she would change about the application. Currently present as: "If you could change one thing about FightPandemics, what would it be?". Present in the second part of the feedback and, therefore, not required.
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	
Sample	
Comments	

2.1.2.2.3.9 Field generalFeedback**2.1.2.2.3.9.1 generalFeedback Tree Diagram**

2.1.2.2.3.9.2 generalFeedback properties

PROPERTY	VALUE
Name	generalFeedback
Technical name	
Id	
Type	string
Description	User's general feedback section. Currently present as: "Any other feedback for us?". Present in the second part of the feedback and, therefore, not required.
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	
Sample	
Comments	

2.1.2.2.3.10 Field covidImpact**2.1.2.2.3.10.1 covidImpact Tree Diagram**

2.1.2.2.3.10.2 covidImpact properties

PROPERTY	VALUE
Name	covidImpact
Technical name	
Id	
Type	string
Description	<p>User's feedback into how the COVID impacted him.</p> <p>Currently present as a multiple choice question.</p> <p>Question: "How has COVID-19 impacted you?"</p> <p>Options:</p> <ol style="list-style-type: none"> 1. I go to work/school normally 2. I am healthy but in a stay-at-home quarantine 3. I have mild symptoms but haven't been tested 4. I am diagnosed with COVID-19 <p>Present in the second part of the feedback and, therefore, not required.</p>
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	
Sample	
Comments	

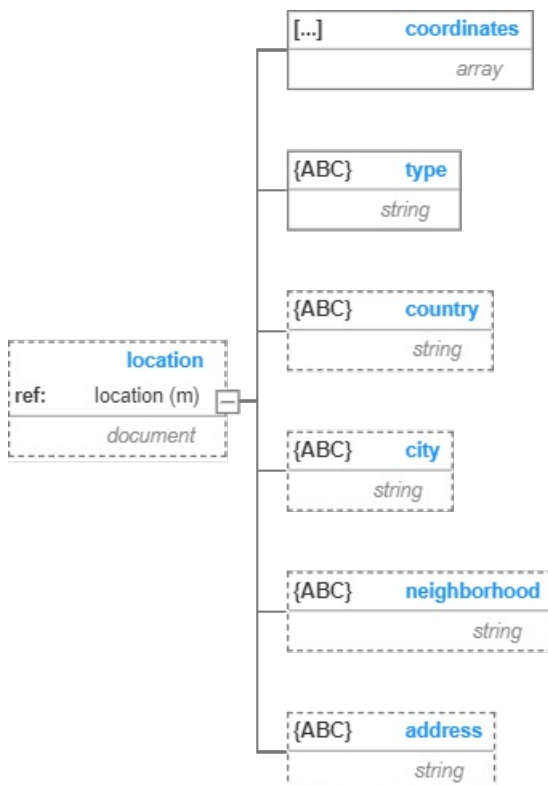
2.1.2.2.3.11 Field userId**2.1.2.2.3.11.1 userId Tree Diagram**

2.1.2.2.3.11.2 userId properties

PROPERTY	VALUE
Name	userId
Technical name	
Id	
Type	objectId
Description	Reference to the user's _id. This will be present only if the user was logged in when completing the feedback.
Dependencies	
Required	false
Primary key	false
Foreign collection	users
Foreign field	_id
Relationship type	Foreign Key
Default	
Sample	
Comments	

2.1.2.2.3.12 Field location

2.1.2.2.3.12.1 location Tree Diagram



2.1.2.2.3.12.2 location properties

PROPERTY	VALUE
Name	location
Technical name	
\$ref	#model/definitions/location
Reference type	model
Reference description	

2.1.2.2.4 feedbacks JSON Schema

```

{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "feedbacks",
  "description": "Collection to hold feedback on the application.\n\nThis may be done with the user logged or not and, depending on that, we may or may not have a reference to the user in question at this model.",
  "additionalProperties": false,
  "properties": {
    "createdAt": {
      "type": "string",
      "description": "Timestamp with the feedback's creation date",
      "format": "date-time"
    },
    "_id": {
      "type": "string",
      "description": "Automatically generated object_id as primary key",
      "pattern": "^a-fA-F0-9]{24}$"
    },
    "updatedAt": {
      "type": "string",
      "description": "Timestamp with the feedback's last update date",
      "format": "date-time"
    },
    "rating": {
      "type": "number",
      "description": "A rating, in a scale of 1 to 5, of the application.\n\nCurrently present as: \"How well does FightPandemics meet your needs?\".\n\nPresent in the first part of the feedback and, therefore, is required.",
      "minimum": 1,
      "maximum": 5
    },
    "ipAddress": {
      "type": "string",
      "description": "Captured IP address in order to get user's location even if they are not logged.\n\nCaptured at the first part of the feedback and, therefore, is required.",
      "format": "ipv4"
    },
    "age": {
      "type": "integer",
      "description": "User's age.\n\nCurrently present as: \"What is your age?\".\n\nPresent in the third part of the feedback and, therefore, not required.",
      "minimum": 18
    },
    "mostValuableFeature": {
      "type": "string",
      "description": "User's feedback into what is the features most valuable for him/her.\n\nCurrently present as: \"Which features are the most valuable for you?\".\n\nPresent in the second part of the feedback and, therefore, not required."
    },
    "whatWouldChange": {
      "type": "string",
      "description": "User's feedback into what he/she would change about the application.\n\nCurrently present as: \"If you could change one thing about FightPandemics, what would it be?\".\n\nPresent in the second part of the feedback and, therefore, not required."
    },
    "generalFeedback": {
      "type": "string",
      "description": "User's general feedback section.\n\nCurrently present as: \"Any other feedback for us?\".\n\nPresent in the second part of the feedback and, therefore, not required."
    },
    "covidImpact": {
      "type": "string",
      "description": "User's feedback into how the COVID impacted him.\n\nCurrently present as a multiple choice question.\n\n**Question:** \"How has COVID-19 impacted you?\"\n\n**Options:** \n1. I go to work/school normally\n2. I am healthy but in a stay-at-home quarantine\n3. I have mild symptoms but haven't been tested\n4. I am diagnosed with COVID-19\n\nPresent in the second part of the feedback and, therefore, not required."
    },
    "userId": {
      "type": "string",
      "description": "Reference to the user's _id.\n\nThis will be present only if the user was logged in when completing the feedback.",
      "pattern": "^a-fA-F0-9]{24}$"
    }
  }
}

```

```

    "location": {
      "$ref": "#model/definitions/location"
    }
  },
  "required": [
    "createdAt",
    "_id",
    "updatedAt",
    "rating",
    "ipAddress"
  ]
}

```

2.1.2.2.5 feedbacks JSON data

```

{
  "createdAt": ISODate("2016-04-08T15:06:21.595Z"),
  "_id": ObjectId("507f1f77bcf86cd799439011"),
  "updatedAt": ISODate("2016-04-08T15:06:21.595Z"),
  "rating": 1,
  "ipAddress": "192.168.0.1",
  "age": 8528994625872795000,
  "mostValuableFeature": "Lorem",
  "whatWouldChange": "Lorem",
  "generalFeedback": "Lorem",
  "covidImpact": "Lorem",
  "userId": ObjectId("507f1f77bcf86cd799439011"),
  "location": {
    "coordinates": [
      35.26,
      -32.83
    ],
    "type": "Point",
    "country": "Lorem",
    "city": "Lorem",
    "neighborhood": "Lorem",
    "address": "Lorem"
  }
}

```

2.1.2.2.6 feedbacks Target Script

```

use applicationDB;

db.createCollection( "feedbacks",{
  "storageEngine": {
    "wiredTiger": {}
  },
  "capped": false,
  "validator": {
    "$jsonSchema": {
      "bsonType": "object",
      "title": "feedbacks",
      "description": "Collection to hold feedback on the application.\n\nThis may be done with the user logged or not and, depending on that, we may or may not have a reference to the user in question at this model.",
      "additionalProperties": false,
      "properties": {
        "_id": {
          "bsonType": "objectId",
          "description": "Automatically generated object_id as primary key."
        },

```

```

    "createdAt": {
      "bsonType": "date",
      "description": "Timestamp with the feedback's creation date"
    },
    "updatedAt": {
      "bsonType": "date",
      "description": "Timestamp with the feedback's last update date"
    },
    "rating": {
      "bsonType": "number",
      "description": "A rating, in a scale of 1 to 5, of the application.\n\nCurrently present as: \"How well does FightPandemics meet your needs?\".\n\nPresent in the first part of the feedback and, therefore, is required.",
      "minimum": 1,
      "maximum": 5
    },
    "ipAddress": {
      "bsonType": "string",
      "description": "Captured IP address in order to get user's location even if they are not logged.\n\nCaptured at the first part of the feedback and, therefore, is required."
    },
    "age": {
      "bsonType": "long",
      "description": "User's age.\n\nCurrently present as: \"What is your age?\".\n\nPresent in the third part of the feedback and, therefore, not required.",
      "minimum": 18
    },
    "mostValuableFeature": {
      "bsonType": "string",
      "description": "User's feedback into what is the features most valuable for him/her.\n\nCurrently present as: \"Which features are the most valuable for you?\".\n\nPresent in the second part of the feedback and, therefore, not required."
    },
    "whatWouldChange": {
      "bsonType": "string",
      "description": "User's feedback into what he/she would change about the application.\n\nCurrently present as: \"If you could change one thing about FightPandemics, what would it be?\".\n\nPresent in the second part of the feedback and, therefore, not required."
    },
    "generalFeedback": {
      "bsonType": "string",
      "description": "User's general feedback section.\n\nCurrently present as: \"Any other feedback for us?\".\n\nPresent in the second part of the feedback and, therefore, not required."
    },
    "covidImpact": {
      "bsonType": "string",
      "description": "User's feedback into how the COVID impacted him.\n\nCurrently present as a multiple choice question.\n\n**Question:** \"How has COVID-19 impacted you?\"\n\n**Options:** \n1. I go to work/school normally\n2. I am healthy but in a stay-at-home quarantine\n3. I have mild symptoms but haven't been tested\n4. I am diagnosed with COVID-19\n\nPresent in the second part of the feedback and, therefore, not required."
    },
    "userId": {
      "bsonType": "objectId",
      "description": "Reference to the user's _id.\n\nThis will be present only if the user was logged in when completing the feedback."
    },
    "location": {
      "bsonType": "object",
      "description": "Location document according to the norms of a [GeoJSON Object] (https://docs.mongodb.com/manual/reference/geojson/).\n\nThe fields address, neighborhood, city, state and country are added to store the result of the geoprocessing.",
      "properties": {
        "coordinates": {
          "bsonType": "array",
          "description": "Array with the coordinates, specifying **longitude first** and **latitude second**, as default in the [GeoJSON] (https://docs.mongodb.com/manual/reference/geojson/) definition.",
          "additionalItems": true,
          "minItems": 2,
          "maxItems": 2,
          "uniqueItems": false,
          "items": [
            {
              "bsonType": "number",
              "description": "Location longitude\n",

```

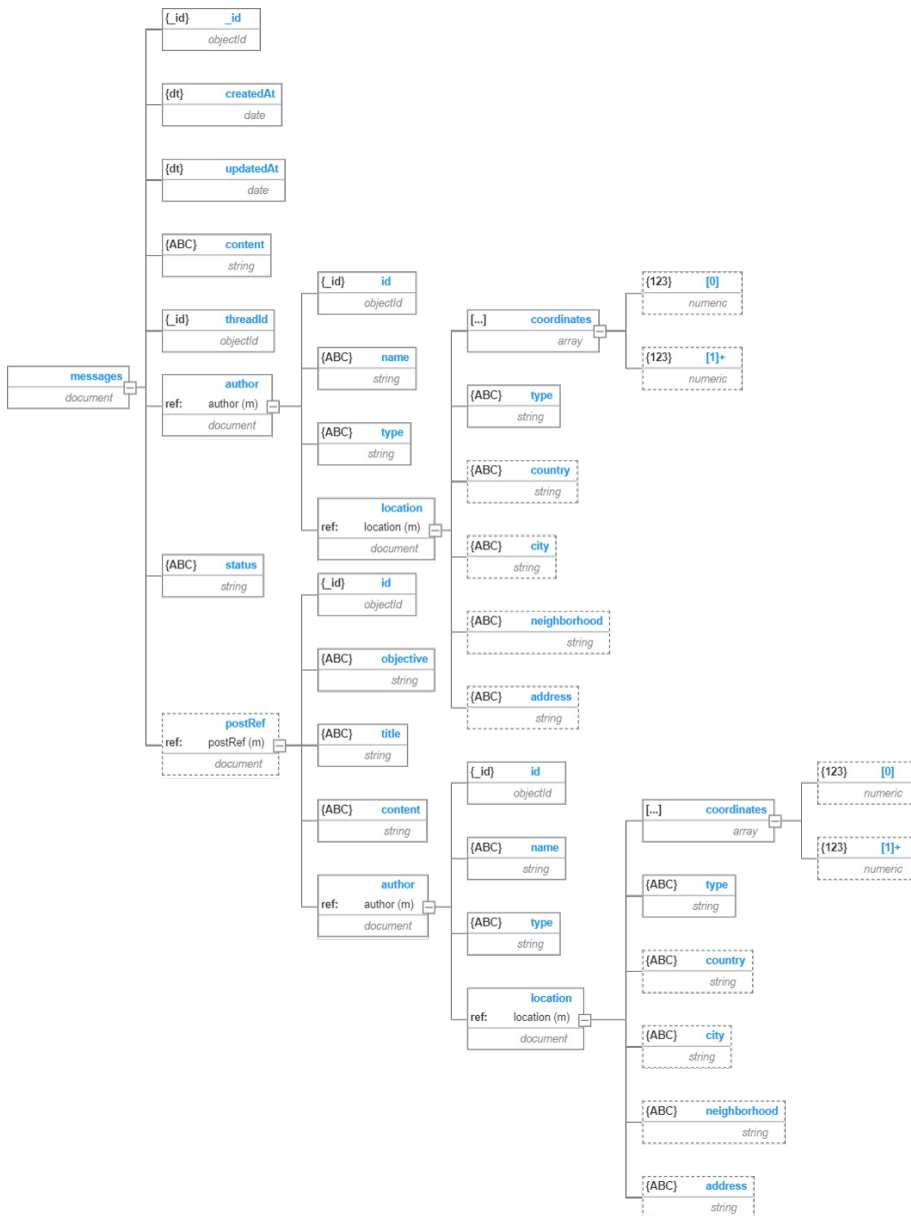
```

        "minimum": -180,
        "maximum": 180
    },
    {
        "bsonType": "number",
        "description": "Location latitude\n",
        "minimum": -90,
        "maximum": 90
    }
]
},
"type": {
    "bsonType": "string",
    "description": "Type of the GeoJSON Object.\nPlease refer to the [official documentation]
(https://docs.mongodb.com/manual/reference/geojson/ ) for more information.\n\n**In our case, the value will always be of the type 'Point'.**",
    "enum": [
        "Point"
    ]
},
"country": {
    "bsonType": "string",
    "description": "String holding the country's name.\n"
},
"city": {
    "bsonType": "string",
    "description": "String holding the city's name."
},
"neighborhood": {
    "bsonType": "string",
    "description": "String holding the neighborhood's name."
},
"address": {
    "bsonType": "string",
    "description": "String holding the address (street, number and other details)."
}
},
"additionalProperties": false,
"required": [
    "coordinates",
    "type"
]
}
},
"required": [
    "createdAt",
    "_id",
    "updatedAt",
    "rating",
    "ipAddress"
]
}
},
"validationLevel": "off",
"validationAction": "warn"
});

```

2.1.2.3 Collection messages

2.1.2.3.1 messages Tree Diagram



2.1.2.3.2 messages Properties

PROPERTY	VALUE
Collection name	messages
Technical name	
Id	
Description	Message sent in a private conversation
Database	applicationDB
Capped	false
Size	
Max	
Storage engine	WiredTiger
Config String	
Validation level	Off
Validation action	Warn
Additional properties	false
Comments	

2.1.2.3.3 messages Fields

FIELD	TYPE	REQ	KEY	DESCRIPTION	COMMENTS
_id	objectId	true		ObjectId used as primary key for a message	
createdAt	date	true		Timestamp with message's creation date	
updatedAt	date	true		Timestamp with message's update date	
content	string	true		Messages content, containing html links and emojis	
threadId	objectId	true		Reference to the thread where the message was sent	
author	document	true			
id	objectId	true		Foreign key to the user who created the post.	
name	string	true			
type	string	true			
location	document	true		Location document according to the norms of a GeoJSON Object . The fields address, neighborhood, city, state and country are added to store the result of the geoprocessing.	
coordinates	array	true		Array with the coordinates, specifying longitude first and latitude second , as default in the GeoJSON definition.	
[0]	numeric	false		Location longitude	
[1]	numeric	false		Location latitude	
type	string	true		Type of the GeoJSON Object. Please refer to the official documentation for more information. In our case, the value will always be of the type "Point".	
country	string	false		String holding the country's name.	
city	string	false		String holding the city's name.	
neighborhood	string	false		String holding the neighborhood's name.	
address	string	false		String holding the address (street, number and other details).	
status	string	true		Current message status, used to define if it was edited or deleted: <ul style="list-style-type: none"> sent: message was sent and not edited or deleted. edited: message was edited by the author deleted: message was deleted by the author 	
postRef	document	false		Optional post reference, allowing a post to be referenced in a message thread.	
id	objectId	true			
objective	string	true		Field to indicate the post's objective. Currently only two values allowed: "offering help" and "looking for help".	

title	string	true		The post title.	
content	string	true		String for the post content.	
author	document	true			
id	objectId	true		Foreign key to the user who created the post.	
name	string	true			
type	string	true			
location	document	true		Location document according to the norms of a GeoJSON Object . The fields address, neighborhood, city, state and country are added to store the result of the geoprocessing.	
coordinates	array	true		Array with the coordinates, specifying longitude first and latitude second , as default in the GeoJSON definition.	
[0]	numeric	false		Location longitude	
[1]	numeric	false		Location latitude	
type	string	true		Type of the GeoJSON Object. Please refer to the official documentation for more information. In our case, the value will always be of the type "Point".	
country	string	false		String holding the country's name.	
city	string	false		String holding the city's name.	
neighborhood	string	false		String holding the neighborhood's name.	
address	string	false		String holding the address (street, number and other details).	

2.1.2.3.3.1 Field _id

2.1.2.3.3.1.1 _id Tree Diagram



2.1.2.3.3.1.2 _id properties

PROPERTY	VALUE
Name	_id
Technical name	
Id	
Type	objectId
Description	ObjectId used as primary key for a message
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Sample	
Comments	

2.1.2.3.3.2 Field createdAt**2.1.2.3.3.2.1 createdAt Tree Diagram**

{dt}	createdAt
	date

2.1.2.3.3.2.2 createdAt properties

PROPERTY	VALUE
Name	createdAt
Technical name	
Id	
Type	date
Description	Timestamp with message's creation date
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Sample	
Comments	

2.1.2.3.3.3 Field updatedAt**2.1.2.3.3.3.1 updatedAt Tree Diagram**

{dt}	updatedAt
	date

2.1.2.3.3.3.2 updatedAt properties

PROPERTY	VALUE
Name	updatedAt
Technical name	
Id	
Type	date
Description	Timestamp with message's update date
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Sample	
Comments	

2.1.2.3.3.4 Field content**2.1.2.3.3.4.1 content Tree Diagram**

{ABC}	content
	<i>string</i>

2.1.2.3.3.4.2 content properties

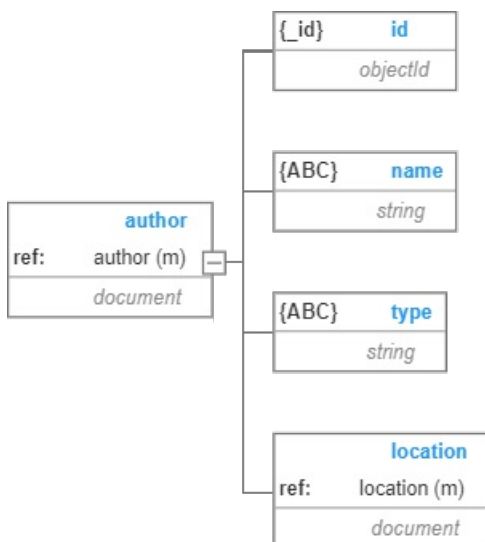
PROPERTY	VALUE
Name	content
Technical name	
Id	
Type	string
Description	Messages content, containing html links and emojis
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	
Sample	
Comments	

2.1.2.3.3.5 Field threadId**2.1.2.3.3.5.1 threadId Tree Diagram**

{_id}	threadId
	<i>objectId</i>

2.1.2.3.3.5.2 threadId properties

PROPERTY	VALUE
Name	threadId
Technical name	
Id	
Type	objectId
Description	Reference to the thread were the message was sent
Dependencies	
Required	true
Primary key	false
Foreign collection	thread
Foreign field	_id
Relationship type	Foreign Key
Default	
Sample	
Comments	

2.1.2.3.3.6 Field author**2.1.2.3.3.6.1 author Tree Diagram****2.1.2.3.3.6.2 author properties**

PROPERTY	VALUE
Name	author
Technical name	
\$ref	#model/definitions/author
Reference type	model
Reference description	

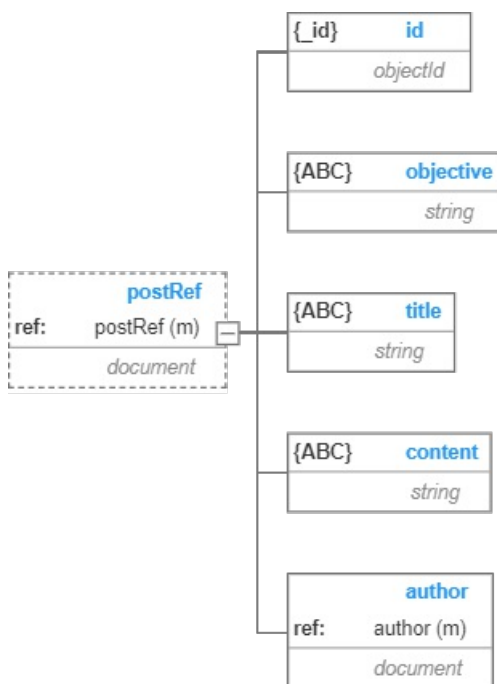
2.1.2.3.3.7 Field status**2.1.2.3.3.7.1 status Tree Diagram**

2.1.2.3.3.7.2 status properties

PROPERTY	VALUE
Name	status
Technical name	
Id	
Type	string
Description	<p>Current message status, used to define if it was edited or deleted:</p> <ul style="list-style-type: none"> • sent: message was sent and not edited or deleted. • edited: message was edited by the author • deleted: message was deleted by the author
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	sent
Min length	
Max length	
Pattern	
Format	
Enum	edited,deleted,sent
Sample	
Comments	

2.1.2.3.3.8 Field postRef

2.1.2.3.3.8.1 postRef Tree Diagram



2.1.2.3.3.8.2 postRef properties

PROPERTY	VALUE
Name	postRef
Technical name	
\$ref	#model/definitions/postRef
Reference type	model
Reference description	

2.1.2.3.4 messages JSON Schema

```

{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "messages",
  "description": "Message sent in a private conversation\n",
  "additionalProperties": false,
  "properties": {
    "_id": {
      "type": "string",
      "description": "ObjectId used as primary key for a message",
      "pattern": "^[a-fA-F0-9]{24}$"
    },
    "createdAt": {
      "type": "string",
      "description": "Timestamp with message's creation date",
      "format": "date-time"
    },
    "updatedAt": {
      "type": "string",
      "description": "Timestamp with message's update date",
      "format": "date-time"
    },
    "content": {
      "type": "string",
      "description": "Messages content, containing html links and emojis "
    },
    "threadId": {
      "type": "string",
      "description": "Reference to the thread were the message was sent",
      "pattern": "^[a-fA-F0-9]{24}$"
    },
    "author": {
      "$ref": "#model/definitions/author"
    },
    "status": {
      "type": "string",
      "description": "Curerent message status, used to define if it was edited or deleted.\n\n* **sent***: message was sent and not edited or deleted.\n* **edited***: message was edited by the author\n* **deleted***: message was deleted by the author",
      "default": "sent",
      "enum": [
        "edited",
        "deleted",
        "sent"
      ]
    },
    "postRef": {
      "$ref": "#model/definitions/postRef"
    }
  },
  "required": [
    "_id",
    "createdAt",
    "updatedAt",
    "content",
    "threadId",
    "author",
    "status"
  ]
}

```

2.1.2.3.5 messages JSON data

```
{
  "_id": ObjectId("507f1f77bcf86cd799439011"),
  "createdAt": ISODate("2016-04-08T15:06:21.595Z"),
  "updatedAt": ISODate("2016-04-08T15:06:21.595Z"),
  "content": "Lorem",
  "threadId": ObjectId("507f1f77bcf86cd799439011"),
  "author": {
    "id": ObjectId("507f1f77bcf86cd799439011"),
    "name": "Lorem",
    "type": "Lorem",
    "location": {
      "coordinates": [
        35.26,
        -32.83
      ],
      "type": "Point",
      "country": "Lorem",
      "city": "Lorem",
      "neighborhood": "Lorem",
      "address": "Lorem"
    }
  },
  "status": "sent",
  "postRef": {
    "id": ObjectId("507f1f77bcf86cd799439011"),
    "objective": "offer",
    "title": "Lorem",
    "content": "Lorem",
    "author": {
      "id": ObjectId("507f1f77bcf86cd799439011"),
      "name": "Lorem",
      "type": "Lorem",
      "location": {
        "coordinates": [
          35.26,
          -32.83
        ],
        "type": "Point",
        "country": "Lorem",
        "city": "Lorem",
        "neighborhood": "Lorem",
        "address": "Lorem"
      }
    }
  }
}
```

2.1.2.3.6 messages Target Script

```
use applicationDB;

db.createCollection( "messages",{
  "storageEngine": {
    "wiredTiger": {}
  },
  "capped": false,
  "validator": {
    "$jsonSchema": {
      "bsonType": "object",
      "title": "messages",
      "description": "Message sent in a private conversation\n",
      "additionalProperties": false,
      "properties": {
        "_id": {
```



```

    "bsonType": "objectId",
    "description": "ObjectId used as primary key for a message"
  },
  "createdAt": {
    "bsonType": "date",
    "description": "Timestamp with message's creation date"
  },
  "updatedAt": {
    "bsonType": "date",
    "description": "Timestamp with message's update date"
  },
  "content": {
    "bsonType": "string",
    "description": "Messages content, containing html links and emojis "
  },
  "threadId": {
    "bsonType": "objectId",
    "description": "Reference to the thread where the message was sent"
  },
  "author": {
    "bsonType": "object",
    "properties": {
      "id": {
        "bsonType": "objectId",
        "description": "Foreign key to the user who created the post."
      },
      "name": {
        "bsonType": "string"
      },
      "type": {
        "bsonType": "string"
      },
      "location": {
        "bsonType": "object",
        "description": "Location document according to the norms of a [GeoJSON Object]
        (https://docs.mongodb.com/manual/reference/geojson/).\n
        The fields address, neighborhood, city, state and country are added to store the
        result of the geoprocessing.",
        "properties": {
          "coordinates": {
            "bsonType": "array",
            "description": "Array with the coordinates, specifying **longitude first** and **latitude second**, as default in the [GeoJSON]
            (https://docs.mongodb.com/manual/reference/geojson/) definition.",
            "additionalItems": true,
            "minItems": 2,
            "maxItems": 2,
            "uniqueItems": false,
            "items": [
              {
                "bsonType": "number",
                "description": "Location longitude\n",
                "minimum": -180,
                "maximum": 180
              },
              {
                "bsonType": "number",
                "description": "Location latitude\n",
                "minimum": -90,
                "maximum": 90
              }
            ]
          },
          "type": {
            "bsonType": "string",
            "description": "Type of the GeoJSON Object.\n
            Please refer to the [official documentation]
            (https://docs.mongodb.com/manual/reference/geojson/) for more information.\n
            \n\n
            In our case, the value will always be of the type \"Point\".***",
            "enum": [
              "Point"
            ]
          },
          "country": {

```

```

        "bsonType": "string",
        "description": "String holding the country's name.\n"
    },
    "city": {
        "bsonType": "string",
        "description": "String holding the city's name."
    },
    "neighborhood": {
        "bsonType": "string",
        "description": "String holding the neighborhood's name."
    },
    "address": {
        "bsonType": "string",
        "description": "String holding the address (street, number and other details)."
    }
},
"additionalProperties": false,
"required": [
    "coordinates",
    "type"
]
}
},
"additionalProperties": false,
"required": [
    "id",
    "name",
    "type"
]
},
"status": {
    "bsonType": "string",
    "description": "Current message status, used to define if it was edited or deleted:\n\n* **sent***: message was sent and not edited or deleted.\n* **edited***: message was edited by the author\n* **deleted***: message was deleted by the author",
    "enum": [
        "edited",
        "deleted",
        "sent"
    ]
},
"postRef": {
    "bsonType": "object",
    "description": "Optional post reference, allowing a post to be referenced in a message thread.",
    "properties": {
        "id": {
            "bsonType": "objectId"
        },
        "objective": {
            "bsonType": "string",
            "description": "Field to indicate the post's objective.\nCurrently only two values allowed: \"offering help\" and \"looking for help\".",
            "enum": [
                "request",
                "offer"
            ]
        },
        "title": {
            "bsonType": "string",
            "description": "The post title."
        },
        "content": {
            "bsonType": "string",
            "description": "String for the post content."
        },
        "author": {
            "bsonType": "object",
            "properties": {
                "id": {
                    "bsonType": "objectId",
                    "description": "Foreign key to the user who created the post."
                },

```

```

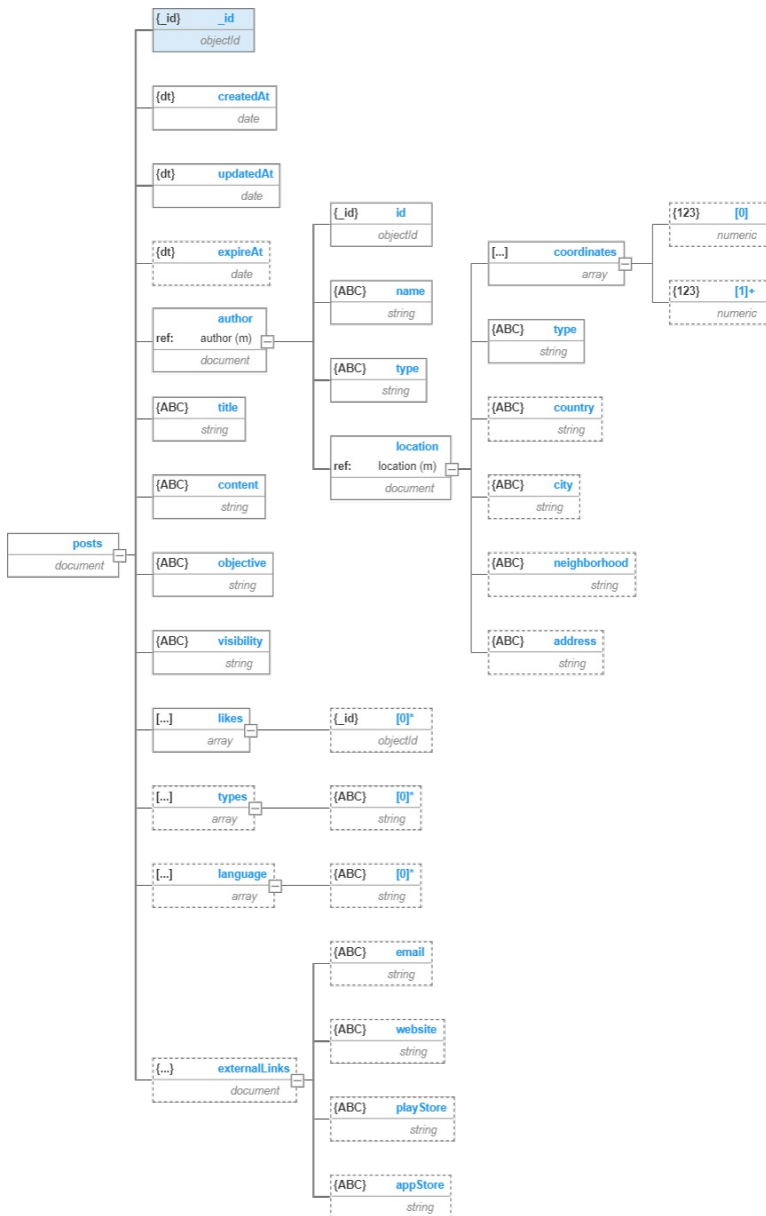
    "name": {
      "bsonType": "string"
    },
    "type": {
      "bsonType": "string"
    },
    "location": {
      "bsonType": "object",
      "description": "Location document according to the norms of a [GeoJSON Object]
      (https://docs.mongodb.com/manual/reference/geojson/ ).\n\nThe fields address, neighborhood, city, state and country are added to store the
      result of the geoprocessing.",
      "properties": {
        "coordinates": {
          "bsonType": "array",
          "description": "Array with the coordinates, specifying **longitude first** and **latitude second**, as default in the
          [GeoJSON](https://docs.mongodb.com/manual/reference/geojson/) definition.",
          "additionalItems": true,
          "minItems": 2,
          "maxItems": 2,
          "uniqueItems": false,
          "items": [
            {
              "bsonType": "number",
              "description": "Location longitude\n",
              "minimum": -180,
              "maximum": 180
            },
            {
              "bsonType": "number",
              "description": "Location latitude\n",
              "minimum": -90,
              "maximum": 90
            }
          ]
        },
        "type": {
          "bsonType": "string",
          "description": "Type of the GeoJSON Object.\nPlease refer to the [official documentation]
          (https://docs.mongodb.com/manual/reference/geojson/ ) for more information.\n\n**In our case, the value will always be of the type 'Point'.**",
          "enum": [
            "Point"
          ]
        },
        "country": {
          "bsonType": "string",
          "description": "String holding the country's name.\n"
        },
        "city": {
          "bsonType": "string",
          "description": "String holding the city's name."
        },
        "neighborhood": {
          "bsonType": "string",
          "description": "String holding the neighborhood's name."
        },
        "address": {
          "bsonType": "string",
          "description": "String holding the address (street, number and other details)."
        }
      },
      "additionalProperties": false,
      "required": [
        "coordinates",
        "type"
      ]
    }
  },
  "additionalProperties": false,
  "required": [
    "id",

```

```
        "name",
        "type"
      ]
    }
  },
  "additionalProperties": false,
  "required": [
    "id",
    "objective",
    "title",
    "content",
    "author"
  ]
}
},
"required": [
  "_id",
  "createdAt",
  "updatedAt",
  "content",
  "threadId",
  "author",
  "status"
]
}
},
"validationLevel": "off",
"validationAction": "warn"
});
```

2.1.2.4 Collection posts

2.1.2.4.1 posts Tree Diagram



2.1.2.4.2 posts Properties

PROPERTY	VALUE
Collection name	posts
Technical name	
Id	
Description	Document to represent a post, asking or offering help to be displayed on the feed
Database	applicationDB
Capped	false
Size	
Max	
Storage engine	WiredTiger
Config String	
Validation level	Off
Validation action	Warn
Additional properties	false
Comments	

2.1.2.4.3 posts Fields

FIELD	TYPE	REQ	KEY	DESCRIPTION	COMMENTS
_id	objectId	true			
createdAt	date	true		Timestamp with the post's creation date	
updatedAt	date	true		Timestamp with the post's last update date	
expireAt	date	false		Timestamp with the date the post will expire. If no expiration date is present, the post does not expire.	
author	document	true			
id	objectId	true		Foreign key to the user who created the post.	
name	string	true			
type	string	true			
location	document	true		Location document according to the norms of a GeoJSON Object . The fields address, neighborhood, city, state and country are added to store the result of the geoprocessing.	
coordinates	array	true		Array with the coordinates, specifying longitude first and latitude second , as default in the GeoJSON definition.	
[0]	numeric	false		Location longitude	
[1]	numeric	false		Location latitude	
type	string	true		Type of the GeoJSON Object. Please refer to the official documentation for more information. In our case, the value will always be of the type "Point".	
country	string	false		String holding the country's name.	
city	string	false		String holding the city's name.	
neighborhood	string	false		String holding the neighborhood's name.	
address	string	false		String holding the address (street, number and other details).	
title	string	true		The post title.	
content	string	true		String for the post content.	
objective	string	true		Field to indicate the post's objective. Currently only two values allowed: "offering help" and "looking for help".	
visibility	string	true		Field to specify the post's visibility. It can be: worldwide, country, state and zipcode.	
likes	array	true		Array holding the user_id of all the users who liked the post.	
[0]	objectId	false		_id of the user who liked the post	

types	array	false		Array with the post's tags. Can be: Medical Supplies, Groceries/Food, Business, Education, Legal, Wellbeing/Mental, Entertainment, Information, Funding, R&D and Others	
[0]	string	false			
language	array	false			
[0]	string	false			
externalLinks	document	false			
email	string	false		An email related to the post's content.	
website	string	false		Link for a website related to the post's content.	
playStore	string	false		Link for a Play Store application related to the post's content.	
appStore	string	false		Link for a AppStore application related to the post's content.	

2.1.2.4.3.1 Field _id

2.1.2.4.3.1.1 _id Tree Diagram

{_id}	_id
	<i>objectId</i>

2.1.2.4.3.1.2 _id properties

PROPERTY	VALUE
Name	_id
Technical name	
Id	
Type	objectId
Description	
Dependencies	
Required	true
Primary key	true
Foreign collection	
Foreign field	
Relationship type	
Default	
Sample	
Comments	

2.1.2.4.3.2 Field createdAt

2.1.2.4.3.2.1 createdAt Tree Diagram

{dt}	createdAt
	<i>date</i>

2.1.2.4.3.2.2 createdAt properties

PROPERTY	VALUE
Name	createdAt
Technical name	
Id	
Type	date
Description	Timestamp with the post's creation date
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Sample	
Comments	

2.1.2.4.3.3 Field updatedAt**2.1.2.4.3.3.1 updatedAt Tree Diagram**

{dt}	updatedAt
	date

2.1.2.4.3.3.2 updatedAt properties

PROPERTY	VALUE
Name	updatedAt
Technical name	
Id	
Type	date
Description	Timestamp with the post's last update date
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Sample	
Comments	

2.1.2.4.3.4 Field expireAt**2.1.2.4.3.4.1 expireAt Tree Diagram**

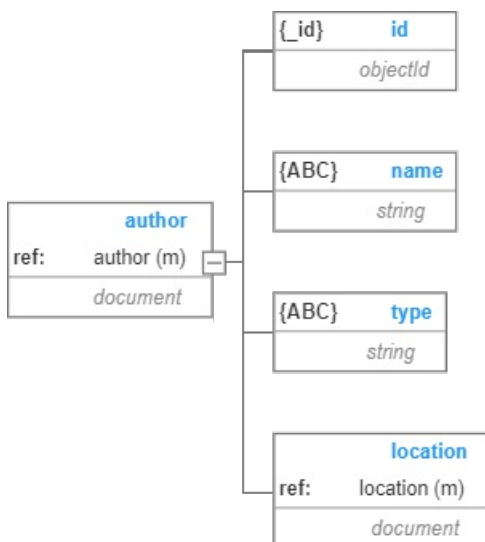
{dt}	expireAt
	date

2.1.2.4.3.4.2 expireAt properties

PROPERTY	VALUE
Name	expireAt
Technical name	
Id	
Type	date
Description	Timestamp with the date the post will expire. If no expiration date is present, the post does not expire.
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Sample	
Comments	

2.1.2.4.3.5 Field author

2.1.2.4.3.5.1 author Tree Diagram



2.1.2.4.3.5.2 author properties

PROPERTY	VALUE
Name	author
Technical name	
\$ref	#model/definitions/author
Reference type	model
Reference description	

2.1.2.4.3.6 Field title

2.1.2.4.3.6.1 title Tree Diagram



2.1.2.4.3.6.2 title properties

PROPERTY	VALUE
Name	title
Technical name	
Id	
Type	string
Description	The post title.
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	
Sample	
Comments	

2.1.2.4.3.7 Field content**2.1.2.4.3.7.1 content Tree Diagram**

{ABC}	content
	<i>string</i>

2.1.2.4.3.7.2 content properties

PROPERTY	VALUE
Name	content
Technical name	
Id	
Type	string
Description	String for the post content.
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	
Sample	
Comments	

2.1.2.4.3.8 Field objective

2.1.2.4.3.8.1 objective Tree Diagram

{ABC}	objective
	<i>string</i>

2.1.2.4.3.8.2 objective properties

PROPERTY	VALUE
Name	objective
Technical name	
Id	
Type	string
Description	Field to indicate the post's objective. Currently only two values allowed: "offering help" and "looking for help".
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	request,offer
Sample	offer
Comments	

2.1.2.4.3.9 Field visibility

2.1.2.4.3.9.1 visibility Tree Diagram

{ABC}	visibility
	<i>string</i>

2.1.2.4.3.9.2 visibility properties

PROPERTY	VALUE
Name	visibility
Technical name	
Id	
Type	string
Description	Field to specify the post's visibility. It can be: worldwide, country, state and zipcode.
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	zipcode,state,country,worldwide
Sample	
Comments	

2.1.2.4.3.10 Field likes**2.1.2.4.3.10.1 likes Tree Diagram****2.1.2.4.3.10.2 likes Hierarchy**Parent field: **posts**

Child field(s):

FIELD	TYPE	REQ	KEY	DESCRIPTION	COMMENTS
[0]	objectId	false	fk	_id of the user who liked the post	

2.1.2.4.3.10.3 likes properties

PROPERTY	VALUE
Name	likes
Technical name	
Id	
Type	array
Description	Array holding the user_id of all the users who liked the post.
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Min items	
Max items	
Unique items	true
Additional items	true
Comments	

2.1.2.4.3.11 Field [0]**2.1.2.4.3.11.1 [0] Tree Diagram****2.1.2.4.3.11.2 [0] properties**

PROPERTY	VALUE
Display name	
Id	
Type	objectId
Description	_id of the user who liked the post
Dependencies	
Primary key	false
Foreign collection	users
Foreign field	_id
Relationship type	Foreign Key
Default	
Sample	
Comments	

2.1.2.4.3.12 Field types**2.1.2.4.3.12.1 types Tree Diagram****2.1.2.4.3.12.2 types Hierarchy**Parent field: **posts**

Child field(s):

FIELD	TYPE	REQ	KEY	DESCRIPTION	COMMENTS
[0]	string	false			

2.1.2.4.3.12.3 types properties

PROPERTY	VALUE
Name	types
Technical name	
Id	
Type	array
Description	Array with the post's tags. Can be: Medical Supplies, Groceries/Food, Business, Education, Legal, Wellbeing/Mental, Entertainment, Information, Funding, R&D and Others
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Min items	
Max items	
Unique items	false
Additional items	true
Comments	

2.1.2.4.3.13 Field [0]**2.1.2.4.3.13.1 [0] Tree Diagram****2.1.2.4.3.13.2 [0] properties**

PROPERTY	VALUE
Display name	
Id	
Type	string
Description	
Dependencies	
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	Medical Supplies,Groceries/Food,Business,Education,Legal,Wellbeing/Mental,Entertainment,Information,Funding,R&D,Others
Sample	Groceries/Food
Comments	

2.1.2.4.3.14 Field language

2.1.2.4.3.14.1 language Tree Diagram



2.1.2.4.3.14.2 language Hierarchy

Parent field: **posts**

Child field(s):

FIELD	TYPE	REQ	KEY	DESCRIPTION	COMMENTS
[0]	string	false			

2.1.2.4.3.14.3 language properties

PROPERTY	VALUE
Name	language
Technical name	
Id	
Type	array
Description	
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Min items	
Max items	
Unique items	false
Additional items	true
Comments	

2.1.2.4.3.15 Field [0]

2.1.2.4.3.15.1 [0] Tree Diagram

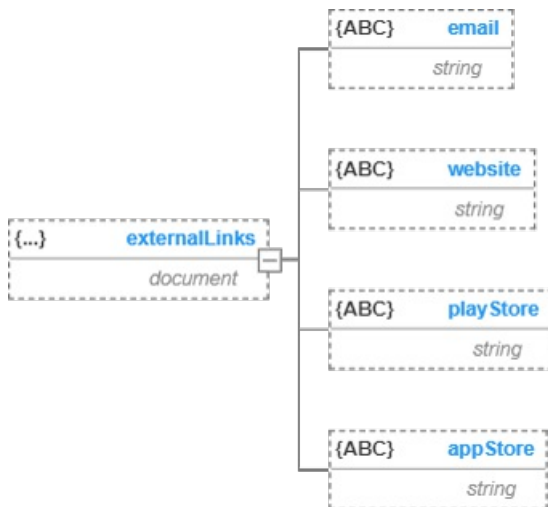


2.1.2.4.3.15.2 [0] properties

PROPERTY	VALUE
Display name	
Id	
Type	string
Description	
Dependencies	
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	
Sample	
Comments	

2.1.2.4.3.16 Field externalLinks

2.1.2.4.3.16.1 externalLinks Tree Diagram



2.1.2.4.3.16.2 externalLinks Hierarchy

Parent field: **posts**

Child field(s):

FIELD	TYPE	REQ	KEY	DESCRIPTION	COMMENTS
email	string	false		An email related to the post's content.	
website	string	false		Link for a website related to the post's content.	
playStore	string	false		Link for a Play Store application related to the post's content.	
appStore	string	false		Link for a AppStore application related to the post's content.	

2.1.2.4.3.16.3 externalLinks properties

PROPERTY	VALUE
Name	externalLinks
Technical name	
Id	
Type	document
Description	
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
DBRef	
Min Properties	
Max Properties	
Additional properties	false
Comments	

2.1.2.4.3.17 Field email

2.1.2.4.3.17.1 email Tree Diagram



2.1.2.4.3.17.2 email properties

PROPERTY	VALUE
Name	email
Technical name	
Id	
Type	string
Description	An email related to the post's content.
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	
Sample	
Comments	

2.1.2.4.3.18 Field website

2.1.2.4.3.18.1 website Tree Diagram



2.1.2.4.3.18.2 website properties

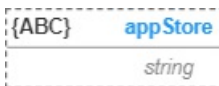
PROPERTY	VALUE
Name	website
Technical name	
Id	
Type	string
Description	Link for a website related to the post's content.
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	
Sample	
Comments	

2.1.2.4.3.19 Field playStore**2.1.2.4.3.19.1 playStore Tree Diagram****2.1.2.4.3.19.2 playStore properties**

PROPERTY	VALUE
Name	playStore
Technical name	
Id	
Type	string
Description	Link for a Play Store application related to the post's content.
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	
Sample	
Comments	

2.1.2.4.3.20 Field appStore

2.1.2.4.3.20.1 appStore Tree Diagram



2.1.2.4.3.20.2 appStore properties

PROPERTY	VALUE
Name	appStore
Technical name	
Id	
Type	string
Description	Link for a AppStore application related to the post's content.
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	
Sample	
Comments	

2.1.2.4.4 posts JSON Schema

```

{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "posts",
  "description": "Document to represent a post, asking or offering help to be displayed on the feed",
  "additionalProperties": false,
  "properties": {
    "_id": {
      "type": "string",
      "pattern": "^[a-fA-F0-9]{24}$"
    },
    "createdAt": {
      "type": "string",
      "description": "Timestamp with the post's creation date",
      "format": "date-time"
    },
    "updatedAt": {
      "type": "string",
      "description": "Timestamp with the post's last update date",
      "format": "date-time"
    },
    "expireAt": {
      "type": "string",
      "description": "Timestamp with the date the post will expire.\\nIf no expiration date is present, the post does not expire.",
      "format": "date-time"
    },
    "author": {
      "$ref": "#model/definitions/author"
    }
  }
}

```

```

"title": {
  "type": "string",
  "description": "The post title."
},
"content": {
  "type": "string",
  "description": "String for the post content."
},
"objective": {
  "type": "string",
  "description": "Field to indicate the post's objective.\nCurrently only two values allowed: \"offering help\" and \"looking for help\".",
  "enum": [
    "request",
    "offer"
  ]
},
"visibility": {
  "type": "string",
  "description": "Field to specify the post's visibility. It can be: worldwide, country, state and zipcode.",
  "enum": [
    "zipcode",
    "state",
    "country",
    "worldwide"
  ]
},
"likes": {
  "type": "array",
  "description": "Array holding the user_id of all the users who liked the post.",
  "additionalItems": true,
  "uniqueItems": true,
  "items": {
    "type": "string",
    "description": " _id of the user who liked the post",
    "pattern": "^[a-zA-F0-9]{24}$"
  }
},
"types": {
  "type": "array",
  "description": "Array with the post's tags.\nCan be: Medical Supplies, Groceries/Food, Business, Education, Legal, Wellbeing/Mental, Entertainment, Information, Funding, R&D and Others",
  "additionalItems": true,
  "uniqueItems": false,
  "items": {
    "type": "string",
    "enum": [
      "Medical Supplies",
      "Groceries/Food",
      "Business",
      "Education",
      "Legal",
      "Wellbeing/Mental",
      "Entertainment",
      "Information",
      "Funding",
      "R&D",
      "Others"
    ]
  }
},
"language": {
  "type": "array",
  "additionalItems": true,
  "uniqueItems": false,
  "items": {
    "type": "string"
  }
},
"externalLinks": {
  "type": "object",

```

```
"properties": {
  "email": {
    "type": "string",
    "description": "An email related to the post's content."
  },
  "website": {
    "type": "string",
    "description": "Link for a website related to the post's content."
  },
  "playStore": {
    "type": "string",
    "description": "Link for a Play Store application related to the post's content."
  },
  "appStore": {
    "type": "string",
    "description": "Link for a AppStore application related to the post's content."
  }
},
"additionalProperties": false
},
"required": [
  "_id",
  "createdAt",
  "updatedAt",
  "author",
  "title",
  "content",
  "objective",
  "visibility",
  "likes"
]
}
```

2.1.2.4.5 posts JSON data

```
{
  "_id": ObjectId("507f1f77bcf86cd799439011"),
  "createdAt": ISODate("2016-04-08T15:06:21.595Z"),
  "updatedAt": ISODate("2016-04-08T15:06:21.595Z"),
  "expireAt": ISODate("2016-04-08T15:06:21.595Z"),
  "author": {
    "id": ObjectId("507f1f77bcf86cd799439011"),
    "name": "Lorem",
    "type": "Lorem",
    "location": {
      "coordinates": [
        35.26,
        -32.83
      ],
      "type": "Point",
      "country": "Lorem",
      "city": "Lorem",
      "neighborhood": "Lorem",
      "address": "Lorem"
    }
  },
  "title": "Lorem",
  "content": "Lorem",
  "objective": "offer",
  "visibility": "worldwide",
  "likes": [
    ObjectId("507f1f77bcf86cd799439011")
  ],
  "types": [
    "Groceries/Food"
  ],
  "language": [
    "Lorem"
  ],
  "externalLinks": {
    "email": "Lorem",
    "website": "Lorem",
    "playStore": "Lorem",
    "appStore": "Lorem"
  }
}
```

2.1.2.4.6 posts Target Script

```
use applicationDB;

db.createCollection( "posts",{
  "storageEngine": {
    "wiredTiger": {}
  },
  "capped": false,
  "validator": {
    "$jsonSchema": {
      "bsonType": "object",
      "title": "posts",
      "description": "Document to represent a post, asking or offering help to be displayed on the feed",
      "additionalProperties": false,
      "properties": {
        "_id": {
          "bsonType": "objectId"
        },
        "createdAt": {
          "bsonType": "date",
          "description": "Timestamp with the post's creation date"
        }
      }
    }
  }
})
```

```

    },
    "updatedAt": {
      "bsonType": "date",
      "description": "Timestamp with the post's last update date"
    },
    "expireAt": {
      "bsonType": "date",
      "description": "Timestamp with the date the post will expire.\nIf no expiration date is present, the post does not expire."
    },
    "author": {
      "bsonType": "object",
      "properties": {
        "id": {
          "bsonType": "objectId",
          "description": "Foreign key to the user who created the post."
        },
        "name": {
          "bsonType": "string"
        },
        "type": {
          "bsonType": "string"
        },
        "location": {
          "bsonType": "object",
          "description": "Location document according to the norms of a [GeoJSON Object]
          (https://docs.mongodb.com/manual/reference/geojson/ ).\nThe fields address, neighborhood, city, state and country are added to store the
          result of the geoprocessing.",
          "properties": {
            "coordinates": {
              "bsonType": "array",
              "description": "Array with the coordinates, specifying **longitude first** and **latitude second**, as default in the [GeoJSON]
              (https://docs.mongodb.com/manual/reference/geojson/) definition.",
              "additionalItems": true,
              "minItems": 2,
              "maxItems": 2,
              "uniqueItems": false,
              "items": [
                {
                  "bsonType": "number",
                  "description": "Location longitude\n",
                  "minimum": -180,
                  "maximum": 180
                },
                {
                  "bsonType": "number",
                  "description": "Location latitude\n",
                  "minimum": -90,
                  "maximum": 90
                }
              ]
            },
            "type": {
              "bsonType": "string",
              "description": "Type of the GeoJSON Object.\nPlease refer to the [official documentation]
              (https://docs.mongodb.com/manual/reference/geojson/ ) for more information.\n\n**In our case, the value will always be of the type 'Point'.**",
              "enum": [
                "Point"
              ]
            },
            "country": {
              "bsonType": "string",
              "description": "String holding the country's name.\n"
            },
            "city": {
              "bsonType": "string",
              "description": "String holding the city's name."
            },
            "neighborhood": {
              "bsonType": "string",
              "description": "String holding the neighborhood's name."
            }
          }
        }
      }
    }
  }

```

```

        },
        "address": {
          "bsonType": "string",
          "description": "String holding the address (street, number and other details)."
        },
      },
      "additionalProperties": false,
      "required": [
        "coordinates",
        "type"
      ]
    },
  },
  "additionalProperties": false,
  "required": [
    "id",
    "name",
    "type"
  ]
},
"title": {
  "bsonType": "string",
  "description": "The post title."
},
"content": {
  "bsonType": "string",
  "description": "String for the post content."
},
"objective": {
  "bsonType": "string",
  "description": "Field to indicate the post's objective.\nCurrently only two values allowed: \"offering help\" and \"looking for help\".",
  "enum": [
    "request",
    "offer"
  ]
},
"visibility": {
  "bsonType": "string",
  "description": "Field to specify the post's visibility. It can be: worldwide, country, state and zipcode.",
  "enum": [
    "zipcode",
    "state",
    "country",
    "worldwide"
  ]
},
"likes": {
  "bsonType": "array",
  "description": "Array holding the user_id of all the users who liked the post.",
  "additionalItems": true,
  "uniqueItems": true,
  "items": {
    "bsonType": "objectId",
    "description": " _id of the user who liked the post"
  }
},
"types": {
  "bsonType": "array",
  "description": "Array with the post's tags.\nCan be: Medical Supplies, Groceries/Food, Business, Education, Legal, Wellbeing/Mental, Entertainment, Information, Funding, R&D and Others",
  "additionalItems": true,
  "uniqueItems": false,
  "items": {
    "bsonType": "string",
    "enum": [
      "Medical Supplies",
      "Groceries/Food",
      "Business",
      "Education",
      "Legal",

```



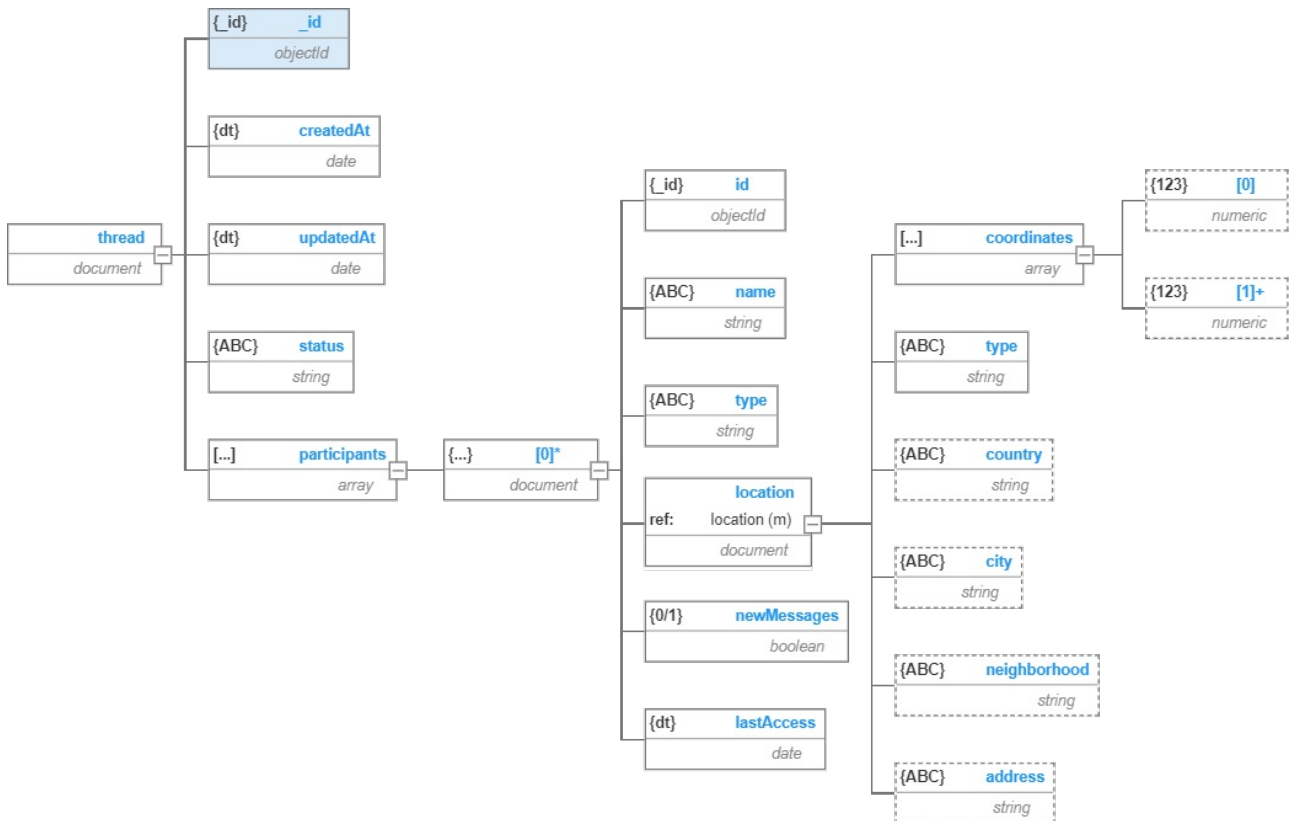
```

        "Wellbeing/Mental",
        "Entertainment",
        "Information",
        "Funding",
        "R&D",
        "Others"
    ]
  },
  "language": {
    "bsonType": "array",
    "additionalItems": true,
    "uniqueItems": false,
    "items": {
      "bsonType": "string"
    }
  },
  "externalLinks": {
    "bsonType": "object",
    "properties": {
      "email": {
        "bsonType": "string",
        "description": "An email related to the post's content."
      },
      "website": {
        "bsonType": "string",
        "description": "Link for a website related to the post's content."
      },
      "playStore": {
        "bsonType": "string",
        "description": "Link for a Play Store application related to the post's content."
      },
      "appStore": {
        "bsonType": "string",
        "description": "Link for a AppStore application related to the post's content."
      }
    }
  },
  "additionalProperties": false
}
},
"required": [
  "_id",
  "createdAt",
  "updatedAt",
  "author",
  "title",
  "content",
  "objective",
  "visibility",
  "likes"
]
}
},
"validationLevel": "off",
"validationAction": "warn"
});

```

2.1.2.5 Collection thread

2.1.2.5.1 thread Tree Diagram



2.1.2.5.2 thread Properties

PROPERTY	VALUE
Collection name	thread
Technical name	
Id	
Description	ObjectId that serves as primary key for the message thread.
Database	applicationDB
Capped	false
Size	
Max	
Storage engine	WiredTiger
Config String	
Validation level	Off
Validation action	Warn
Additional properties	false
Comments	

2.1.2.5.3 thread Fields

FIELD	TYPE	REQ	KEY	DESCRIPTION	COMMENTS
_id	objectId	true			
createdAt	date	true		Timestamp with the threads's creation date	
updatedAt	date	true		Timestamp with the thread's update date	
status	string	true		<p>Current thread status, used to define archives, soft deletes and other conversation status. The statuses are:</p> <ul style="list-style-type: none"> pending: if the user didn't accept the message request yet blocked: if the one of the user's blocked the other 	

				<ul style="list-style-type: none"> • accepted: message request was accepted and users exchange messages • archived: user archived the message thread • deleted: user deleted the conversation 	
participants	array	true		<p>Array of user reference's, containing the participants of the message thread.</p> <p>It is modelled as an array to allow expansion to group chats in the future.</p>	
[0]	document	true			
id	objectId	true		Foreign key to the user who created the post.	
name	string	true		User's name	
type	string	true		User's type	
location	document	true		Location document according to the norms of a GeoJSON Object . The fields address, neighborhood, city, state and country are added to store the result of the geoprocessing.	
coordinates	array	true		Array with the coordinates, specifying longitude first and latitude second , as default in the GeoJSON definition.	
[0]	numeric	false		Location longitude	
[1]	numeric	false		Location latitude	
type	string	true		<p>Type of the GeoJSON Object. Please refer to the official documentation for more information.</p> <p>In our case, the value will always be of the type "Point".</p>	
country	string	false		String holding the country's name.	
city	string	false		String holding the city's name.	
neighborhood	string	false		String holding the neighborhood's name.	
address	string	false		String holding the address (street, number and other details).	
newMessages	boolean	true		Flag to indicate whether there are new messages since last time the user accessed the thread.	
lastAccess	date	true		Timestamp with the last time the user accessed the message thread.	

2.1.2.5.3.1 Field _id

2.1.2.5.3.1.1 _id Tree Diagram



2.1.2.5.3.1.2 _id properties

PROPERTY	VALUE
Name	_id
Technical name	
Id	
Type	objectId
Description	
Dependencies	
Required	true
Primary key	true
Foreign collection	
Foreign field	
Relationship type	
Default	
Sample	
Comments	

2.1.2.5.3.2 Field createdAt**2.1.2.5.3.2.1 createdAt Tree Diagram**

{dt}	createdAt
	date

2.1.2.5.3.2.2 createdAt properties

PROPERTY	VALUE
Name	createdAt
Technical name	
Id	
Type	date
Description	Timestamp with the threads's creation date
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Sample	
Comments	

2.1.2.5.3.3 Field updatedAt**2.1.2.5.3.3.1 updatedAt Tree Diagram**

{dt}	updatedAt
	date

2.1.2.5.3.3.2 updatedAt properties

PROPERTY	VALUE
Name	updatedAt
Technical name	
Id	
Type	date
Description	Timestamp with the thread's update date
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Sample	
Comments	

2.1.2.5.3.4 Field status

2.1.2.5.3.4.1 status Tree Diagram

{ABC}	status
	string

2.1.2.5.3.4.2 status properties

PROPERTY	VALUE
Name	status
Technical name	
Id	
Type	string
Description	<p>Current thread status, used to define archives, soft deletes and other conversation status. The statuses are:</p> <ul style="list-style-type: none"> • pending: if the user didn't accept the message request yet • blocked: if the one of the user's blocked the other • accepted: message request was accepted and users exchange messages • archived: user archived the message thread • deleted: user deleted the conversation
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	pending
Min length	
Max length	
Pattern	
Format	
Enum	deleted,blocked,accepted,pending,archived
Sample	
Comments	

2.1.2.5.3.5 Field participants

2.1.2.5.3.5.1 participants Tree Diagram



2.1.2.5.3.5.2 participants Hierarchy

Parent field: **thread**

Child field(s):

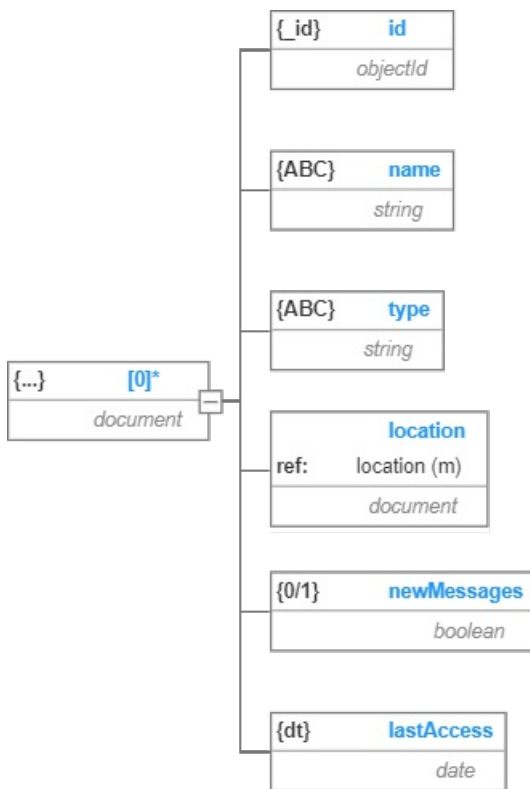
FIELD	TYPE	REQ	KEY	DESCRIPTION	COMMENTS
[0]	document	true			

2.1.2.5.3.5.3 participants properties

PROPERTY	VALUE
Name	participants
Technical name	
Id	
Type	array
Description	Array of user reference's, containing the participants of the message thread. It is modelled as an array to allow expansion to group chats in the future.
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Min items	2
Max items	
Unique items	false
Additional items	true
Comments	

2.1.2.5.3.6 Field [0]

2.1.2.5.3.6.1 [0] Tree Diagram



2.1.2.5.3.6.2 [0] Hierarchy

Parent field: **participants**

Child field(s):

FIELD	TYPE	REQ	KEY	DESCRIPTION	COMMENTS
id	objectId	true	fk	Foreign key to the user who created the post.	
name	string	true		User's name	
type	string	true		User's type	
location	reference	true			
newMessages	boolean	true		Flag to indicate whether there are new messages since last time the user accessed the thread.	
lastAccess	date	true		Timestamp with the last time the user accessed the message thread.	

2.1.2.5.3.6.3 [0] properties

PROPERTY	VALUE
Id	
Type	document
Description	
Dependencies	
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
DBRef	
Min Properties	
Max Properties	
Additional properties	false
Comments	

2.1.2.5.3.7 Field id**2.1.2.5.3.7.1 id Tree Diagram**

{_id}	id
	<i>objectId</i>

2.1.2.5.3.7.2 id properties

PROPERTY	VALUE
Name	id
Technical name	
Id	
Type	objectId
Description	Foreign key to the user who created the post.
Dependencies	
Required	true
Primary key	false
Foreign collection	users
Foreign field	_id
Relationship type	Foreign Key
Default	
Sample	
Comments	

2.1.2.5.3.8 Field name**2.1.2.5.3.8.1 name Tree Diagram**

{ABC}	name
	<i>string</i>

2.1.2.5.3.8.2 name properties

PROPERTY	VALUE
Name	name
Technical name	
Id	
Type	string
Description	User's name
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	
Sample	
Comments	

2.1.2.5.3.9 Field type**2.1.2.5.3.9.1 type Tree Diagram**

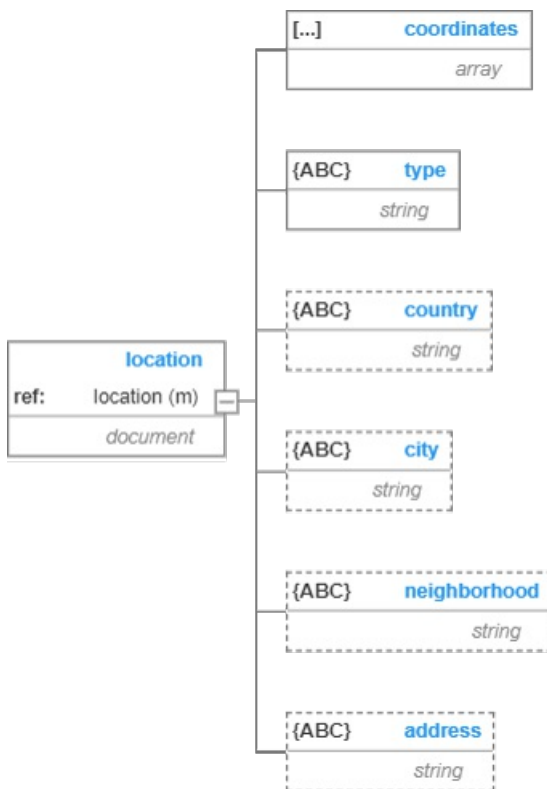
{ABC}	type
	string

2.1.2.5.3.9.2 type properties

PROPERTY	VALUE
Name	type
Technical name	
Id	
Type	string
Description	User's type
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	
Sample	
Comments	

2.1.2.5.3.10 Field location

2.1.2.5.3.10.1 location Tree Diagram



2.1.2.5.3.10.2 location properties

PROPERTY	VALUE
Name	location
Technical name	
\$ref	#model/definitions/location
Reference type	model
Reference description	User's location

2.1.2.5.3.11 Field newMessages

2.1.2.5.3.11.1 newMessages Tree Diagram



2.1.2.5.3.11.2 newMessages properties

PROPERTY	VALUE
Name	newMessages
Technical name	
Id	
Type	boolean
Description	Flag to indicate whether there are new messages since last time the user accessed the thread.
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	false
Sample	
Comments	

2.1.2.5.3.12 Field lastAccess**2.1.2.5.3.12.1 lastAccess Tree Diagram**

{dt}	lastAccess
	date

2.1.2.5.3.12.2 lastAccess properties

PROPERTY	VALUE
Name	lastAccess
Technical name	
Id	
Type	date
Description	Timestamp with the last time the user accessed the message thread.
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Sample	
Comments	

2.1.2.5.4 thread JSON Schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "thread",
  "description": "ObjectId that serves as primary key for the message thread.",
  "additionalProperties": false,
  "properties": {
    "_id": {
      "type": "string",
      "pattern": "^a-fA-F0-9]{24}$"
    }
  },
}
```

```

"createdAt": {
  "type": "string",
  "description": "Timestamp with the threads's creation date",
  "format": "date-time"
},
"updatedAt": {
  "type": "string",
  "description": "Timestamp with the thread's update date",
  "format": "date-time"
},
"status": {
  "type": "string",
  "description": "Current thread status, used to define archives, soft deletes and other conversation status. The statuses are:\n\n*
**pending**: if the user didn't accept the message request yet\n* **blocked**: if the one of the user's blocked the other\n* **accepted**:
message request was accepted and users exchange messages\n* **archived**: user archived the message thread\n* **deleted**: user
deleted the conversation",
  "default": "pending",
  "enum": [
    "deleted",
    "blocked",
    "accepted",
    "pending",
    "archived"
  ]
},
"participants": {
  "type": "array",
  "description": "Array of user reference's, containing the participants of the message thread.\n\nIt is modelled as an array to allow
expansion to group chats in the future.",
  "additionalItems": true,
  "minItems": 2,
  "uniqueItems": false,
  "items": {
    "type": "object",
    "properties": {
      "id": {
        "type": "string",
        "description": "Foreign key to the user who created the post.",
        "pattern": "^a-fA-F0-9{24}$"
      },
      "name": {
        "type": "string",
        "description": "User's name"
      },
      "type": {
        "type": "string",
        "description": "User's type"
      },
      "location": {
        "$ref": "#model/definitions/location"
      },
      "newMessages": {
        "type": "boolean",
        "description": "Flag to indicate whether there are new messages since last time the user accessed the thread.",
        "default": false
      },
      "lastAccess": {
        "type": "string",
        "description": "Timestamp with the last time the user accessed the message thread.",
        "format": "date-time"
      }
    }
  },
  "additionalProperties": false,
  "required": [
    "id",
    "name",
    "type",
    "location",
    "newMessages",
    "lastAccess"
  ]
}

```

```

    ]
  }
}
},
"required": [
  "_id",
  "createdAt",
  "updatedAt",
  "status",
  "participants"
]
}
}

```

2.1.2.5.5 thread JSON data

```

{
  "_id": ObjectId("507f1f77bcf86cd799439011"),
  "createdAt": ISODate("2016-04-08T15:06:21.595Z"),
  "updatedAt": ISODate("2016-04-08T15:06:21.595Z"),
  "status": "pending",
  "participants": [
    {
      "id": ObjectId("507f1f77bcf86cd799439011"),
      "name": "Lorem",
      "type": "Lorem",
      "location": {
        "coordinates": [
          35.26,
          -32.83
        ],
        "type": "Point",
        "country": "Lorem",
        "city": "Lorem",
        "neighborhood": "Lorem",
        "address": "Lorem"
      },
      "newMessages": false,
      "lastAccess": ISODate("2016-04-08T15:06:21.595Z")
    }
  ]
}

```

2.1.2.5.6 thread Target Script

```

use applicationDB;

db.createCollection( "thread",{
  "storageEngine": {
    "wiredTiger": {}
  },
  "capped": false,
  "validator": {
    "$jsonSchema": {
      "bsonType": "object",
      "title": "thread",
      "description": "ObjectId that serves as primary key for the message thread.",
      "additionalProperties": false,
      "properties": {
        "_id": {
          "bsonType": "objectId"
        },

```

```

    "createdAt": {
      "bsonType": "date",
      "description": "Timestamp with the threads's creation date"
    },
    "updatedAt": {
      "bsonType": "date",
      "description": "Timestamp with the thread's update date"
    },
    "status": {
      "bsonType": "string",
      "description": "Current thread status, used to define archives, soft deletes and other conversation status. The statuses are:\n\n*
**pending**: if the user didn't accept the message request yet\n* **blocked**: if the one of the user's blocked the other\n* **accepted**:
message request was accepted and users exchange messages\n* **archived**: user archived the message thread\n* **deleted**: user
deleted the conversation",
      "enum": [
        "deleted",
        "blocked",
        "accepted",
        "pending",
        "archived"
      ]
    },
    "participants": {
      "bsonType": "array",
      "description": "Array of user reference's, containing the participants of the message thread.\n\nIt is modelled as an array to allow
expansion to group chats in the future.",
      "additionalItems": true,
      "minItems": 2,
      "uniqueItems": false,
      "items": {
        "bsonType": "object",
        "properties": {
          "id": {
            "bsonType": "objectId",
            "description": "Foreign key to the user who created the post."
          },
          "name": {
            "bsonType": "string",
            "description": "User's name"
          },
          "type": {
            "bsonType": "string",
            "description": "User's type"
          },
          "location": {
            "bsonType": "object",
            "description": "Location document according to the norms of a [GeoJSON Object]
(https://docs.mongodb.com/manual/reference/geojson/).\n\nThe fields address, neighborhood, city, state and country are added to store the
result of the geoprocessing.",
            "properties": {
              "coordinates": {
                "bsonType": "array",
                "description": "Array with the coordinates, specifying **longitude first** and **latitude second**, as default in the
[GeoJSON](https://docs.mongodb.com/manual/reference/geojson/) definition.",
                "additionalItems": true,
                "minItems": 2,
                "maxItems": 2,
                "uniqueItems": false,
                "items": [
                  {
                    "bsonType": "number",
                    "description": "Location longitude\n",
                    "minimum": -180,
                    "maximum": 180
                  },
                  {
                    "bsonType": "number",
                    "description": "Location latitude\n",
                    "minimum": -90,
                    "maximum": 90
                  }
                ]
              }
            }
          }
        }
      }
    }
  }

```

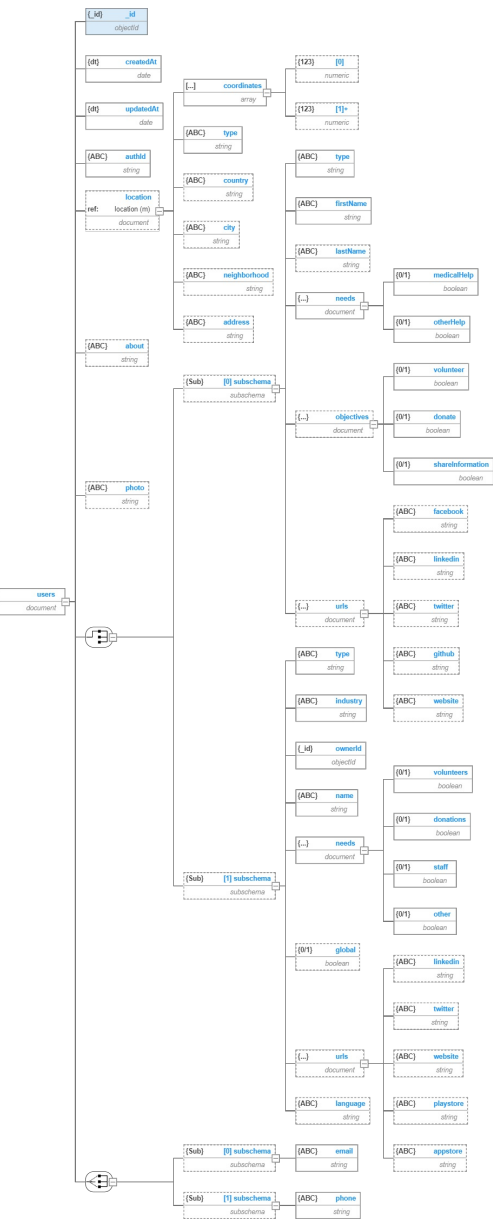
```

        }
    ]
},
"type": {
    "bsonType": "string",
    "description": "Type of the GeoJSON Object.\nPlease refer to the [official documentation]
(https://docs.mongodb.com/manual/reference/geojson/ ) for more information.\n\n**In our case, the value will always be of the type 'Point'.**",
    "enum": [
        "Point"
    ]
},
"country": {
    "bsonType": "string",
    "description": "String holding the country's name.\n"
},
"city": {
    "bsonType": "string",
    "description": "String holding the city's name."
},
"neighborhood": {
    "bsonType": "string",
    "description": "String holding the neighborhood's name."
},
"address": {
    "bsonType": "string",
    "description": "String holding the address (street, number and other details)."
}
},
"additionalProperties": false,
"required": [
    "coordinates",
    "type"
]
},
"newMessages": {
    "bsonType": "bool",
    "description": "Flag to indicate whether there are new messages since last time the user accessed the thread."
},
"lastAccess": {
    "bsonType": "date",
    "description": "Timestamp with the last time the user accessed the message thread."
}
},
"additionalProperties": false,
"required": [
    "id",
    "name",
    "type",
    "newMessages",
    "lastAccess"
]
}
},
},
"required": [
    "_id",
    "createdAt",
    "updatedAt",
    "status",
    "participants"
]
}
},
"validationLevel": "off",
"validationAction": "warn"
});

```

2.1.2.6 Collection users

2.1.2.6.1 users Tree Diagram



2.1.2.6.2 users Properties

PROPERTY	VALUE
Collection name	users
Technical name	
Id	
Description	<p>Collection that holds all the data on system's users.</p> <p>The users can be of two types: (1) Individual, representing a private user account; and (2) Organization, representing an organization of various types.</p> <p>The field 'type' is used to differentiate between the two subschemas, with 'individual' representing private user accounts and "Community", "Government", "Health care provider", "Non-profit", "Other", "R&D", "Startup", "Traditional Company" and "University".</p>
Database	applicationDB
Capped	false
Size	
Max	
Storage engine	WiredTiger
Config String	
Validation level	Off
Validation action	Warn
Additional properties	true
Comments	

2.1.2.6.3 users Fields

FIELD	TYPE	REQ	KEY	DESCRIPTION	COMMENTS
_id	objectId	true		Automatically generated _id as primary key.	
createdAt	date	true		Timestamp with the user's creation date	Thu Apr 16 2020 04:59:24 GMT-0300 (Horário Padrão de Brasília): changed the name to registeredAt
updatedAt	date	true		Timestamp indicating last update on the database.	
authId	string	true		String holding the authentication service id (Auth0). This field is a foreign key to the user in the Auth0 database , connecting through the user_id field.	
location	document	false		Location document according to the norms of a GeoJSON Object . The fields address, neighborhood, city, state and country are added to store the result of the geoprocessing.	
coordinates	array	true		Array with the coordinates, specifying longitude first and latitude second , as default in the GeoJSON definition.	

[0]	numeric	false		Location longitude	
[1]	numeric	false		Location latitude	
type	string	true		Type of the GeoJSON Object. Please refer to the official documentation for more information. In our case, the value will always be of the type "Point".	
country	string	false		String holding the country's name.	
city	string	false		String holding the city's name.	
neighborhood	string	false		String holding the neighborhood's name.	
address	string	false		String holding the address (street, number and other details).	
about	string	false		Small self-introduction, limited to 100 char	
photo	uri	false		URI with the location of the profile's photo.	
oneOf	choice	false			
[0] subschema	subschema	false		Subschema representing a individual account.	
type	string	true			
firstName	string	true		The first name of the user	
lastName	string	false		The last name of the user. Optional field.	
needs	document	true		User's need in the app, representing what kind of help he/she needs	
medicalHelp	boolean	true		Flag to indicate wheter the user needs medical help. Marking this indicates he shows symptoms of COVID-19.	
otherHelp	boolean	true		Flag to indicate wheter the user needs non-medical help, such as getting groceries, medice and etc..	
objectives	document	false		Document with what are the objectives of the user through the application, such as donating, volunteering or sharing information.	
volunteer	boolean	true		Flag to indicate wheter the user wants to volunteer for iniciatives through the app.	
donate	boolean	true		Flag to indicate wheter the user wants to donate for iniciatives through the app.	
shareInformation	boolean	true		Flag to indicate wheter the user wants to share information through the app.	

urls	document	false		Document holding the user urls to external websites and social networks	
facebook	uri	false		URL to user's Facebook page.	
linkedin	uri	false		URL to user's Linkedin page.	
twitter	uri	false		URL to user's Twitter page.	
github	uri	false		URL to user's Github page.	
website	uri	false		URL to user's personal website.	
[1] subschema	subschema	false		Subschema representing an organization (collective) account.	
type	string	true		The type of the organization. Can be: Community, Government, Health care provider, Non-profit, Other, R&D, Startup, Traditional Company and University. All names are store in lowercase only.	
industry	string	true		Industry from which the organization participates. Plan is for this values to be scraped from linkedin industries.	
ownerId	objectId	true		_id of the user who owns the organization.	
name	string	true		Organization's name.	
needs	document	true		Document with what are the objectives of the organization through the application, such as finding volunteers, staff, investors and donors.	
volunteers	boolean	true		Flag to indicate wheter the organization is seeking volunteers through the app.	
donations	boolean	true		Flag to indicate wheter the organization is seeking donations through the app.	
staff	boolean	true		Flag to indicate wheter the organization is seeking staff through the app.	
other	boolean	true		Flag to indicate wheter the organization is seeking other needs through the app.	
global	boolean	false		Flag to indicate wheter this organization is global.	
urls	document	false		Document holding external urls and refs to the organization such as social media profiles.	
linkedin	uri	false		URL to organization's Linkedin page.	

twitter	uri	false		URL to organization's Twitter page.	
website	uri	false		URL to organization's website.	
playstore	uri	false		URL to organization's page at Google Play Store.	
appstore	uri	false		URL to organization's page at Apple App Store	
language	string	false		Field to indicate the language used by the organization.	
anyOf	choice	false			
[0] subschema	subschema	false		Case where the user comes from social networks with a registration that uses email. In this case, the email field is required. This case covers the great majority of currently supported social network authentications.	
email	email	true		User's email validated by a regex pattern.	
[1] subschema	subschema	false		Case where the user comes from social networks with a registration that allows the use of only the phone. In this case, the phone field is required, and the email isn't. An example of social network that falls into this case is Twitter.	
phone	string	true			

2.1.2.6.3.1 Field createdAt

2.1.2.6.3.1.1 createdAt Tree Diagram

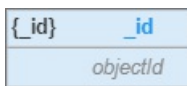
{dt}	createdAt
	date

2.1.2.6.3.1.2 createdAt properties

PROPERTY	VALUE
Name	createdAt
Technical name	
Id	
Type	date
Description	Timestamp with the user's creation date
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Sample	
Comments	Thu Apr 16 2020 04:59:24 GMT-0300 (Horário Padrão de Brasília): changed the name to registeredAt

2.1.2.6.3.2 Field _id

2.1.2.6.3.2.1 _id Tree Diagram

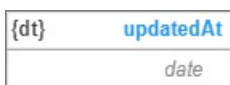


2.1.2.6.3.2.2 _id properties

PROPERTY	VALUE
Name	_id
Technical name	
Id	
Type	objectId
Description	Automatically generated _id as primary key.
Dependencies	
Required	true
Primary key	true
Foreign collection	
Foreign field	
Relationship type	
Default	
Sample	
Comments	

2.1.2.6.3.3 Field updatedAt

2.1.2.6.3.3.1 updatedAt Tree Diagram



2.1.2.6.3.3.2 updatedAt properties

PROPERTY	VALUE
Name	updatedAt
Technical name	
Id	
Type	date
Description	Timestamp indicating last update on the database.
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Sample	
Comments	

2.1.2.6.3.4 Field authId

2.1.2.6.3.4.1 authId Tree Diagram

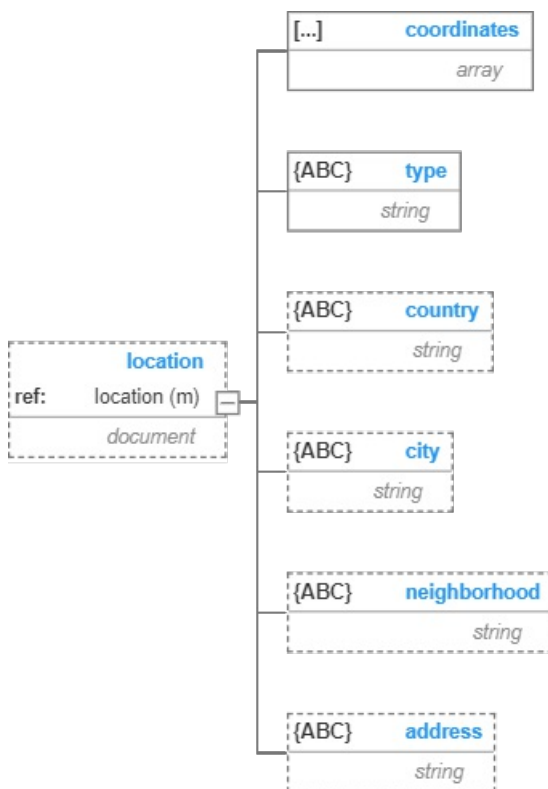


2.1.2.6.3.4.2 authId properties

PROPERTY	VALUE
Name	authId
Technical name	
Id	
Type	string
Description	String holding the authentication service id (Auth0). This field is a foreign key to the user in the Auth0 database , connecting through the user_id field.
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	
Sample	
Comments	

2.1.2.6.3.5 Field location

2.1.2.6.3.5.1 location Tree Diagram



2.1.2.6.3.5.2 location properties

PROPERTY	VALUE
Name	location
Technical name	
\$ref	#model/definitions/location
Reference type	model
Reference description	

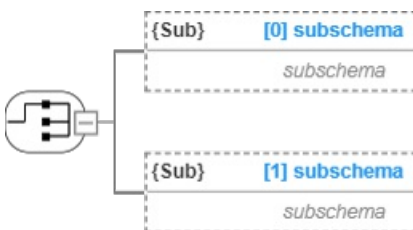
2.1.2.6.3.6 Field about**2.1.2.6.3.6.1 about Tree Diagram****2.1.2.6.3.6.2 about properties**

PROPERTY	VALUE
Name	about
Technical name	
Id	
Type	string
Description	Small self-introduction, limited to 100 char
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	
Sample	
Comments	

2.1.2.6.3.7 Field photo**2.1.2.6.3.7.1 photo Tree Diagram**

2.1.2.6.3.7.2 photo properties

PROPERTY	VALUE
Name	photo
Technical name	
Id	
Type	string
Description	URI with the location of the profile's photo.
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	uri
Enum	
Sample	
Comments	

2.1.2.6.3.8 Field oneOf**2.1.2.6.3.8.1 oneOf Tree Diagram****2.1.2.6.3.8.2 oneOf Hierarchy**Parent field: **users**

Child field(s):

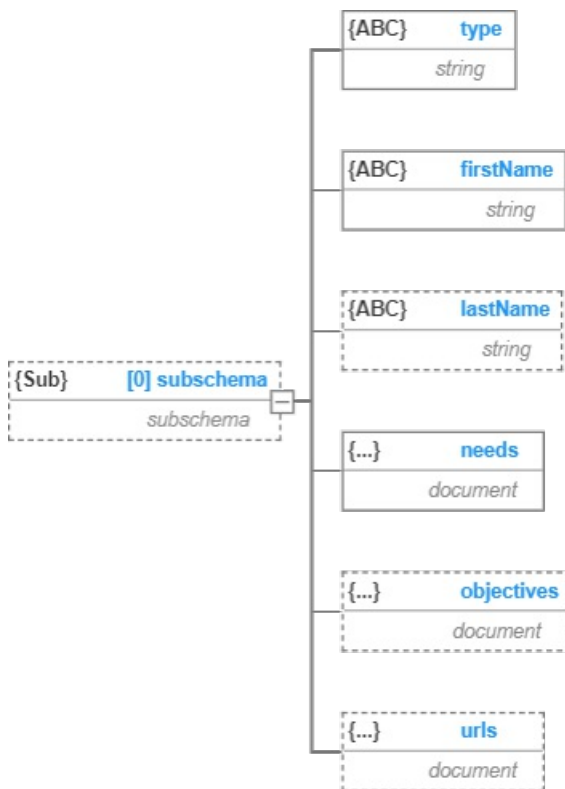
FIELD	TYPE	REQ	KEY	DESCRIPTION	COMMENTS
subschema	subschema	false		Subschema representing a individual account.	
subschema	subschema	false		Subschema representing an organization (colective) account.	

2.1.2.6.3.8.3 oneOf properties

PROPERTY	VALUE
Choice	oneOf
Description	
Comments	

2.1.2.6.3.9 Field [0] subschema

2.1.2.6.3.9.1 [0] subschema Tree Diagram



2.1.2.6.3.9.2 [0] subschema Hierarchy

Parent field: **oneOf**

Child field(s):

FIELD	TYPE	REQ	KEY	DESCRIPTION	COMMENTS
type	string	true			
firstName	string	true		The first name of the user	
lastName	string	false		The last name of the user. Optional field.	
needs	document	true		User's need in the app, representing what kind of help he/she needs	
objectives	document	false		Document with what are the objectives of the user through the application, such as donating, volunteering or sharing information.	
urls	document	false		Document holding the user urls to external websites and social networks	

2.1.2.6.3.9.3 [0] subschema properties

PROPERTY	VALUE
Display name	subschema
Id	individual
Description	Subschema representing a individual account.
Required	
Min Properties	
Max Properties	
Additional properties	true
Comments	

2.1.2.6.3.10 Field type**2.1.2.6.3.10.1 type Tree Diagram**

{ABC}	type
	string

2.1.2.6.3.10.2 type properties

PROPERTY	VALUE
Name	type
Technical name	
Id	
Type	string
Description	
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	individual
Sample	
Comments	

2.1.2.6.3.11 Field firstName**2.1.2.6.3.11.1 firstName Tree Diagram**

{ABC}	firstName
	string

2.1.2.6.3.11.2 firstName properties

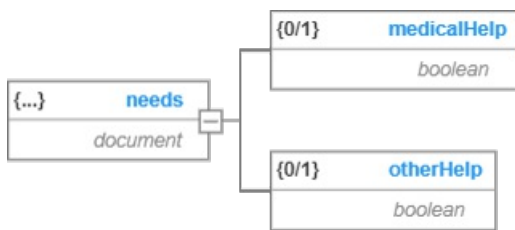
PROPERTY	VALUE
Name	firstName
Technical name	
Id	
Type	string
Description	The first name of the user
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	
Sample	
Comments	

2.1.2.6.3.12 Field lastName**2.1.2.6.3.12.1 lastName Tree Diagram****2.1.2.6.3.12.2 lastName properties**

PROPERTY	VALUE
Name	lastName
Technical name	
Id	
Type	string
Description	The last name of the user. Optional field.
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	
Sample	
Comments	

2.1.2.6.3.13 Field needs

2.1.2.6.3.13.1 needs Tree Diagram



2.1.2.6.3.13.2 needs Hierarchy

Parent field: **[0] subschema**

Child field(s):

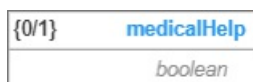
FIELD	TYPE	REQ	KEY	DESCRIPTION	COMMENTS
medicalHelp	boolean	true		Flag to indicate wheter the user needs medical help. Marking this indicates he shows symptoms of COVID-19.	
otherHelp	boolean	true		Flag to indicate wheter the user needs non-medical help, such as getting groceries, medice and etc..	

2.1.2.6.3.13.3 needs properties

PROPERTY	VALUE
Name	needs
Technical name	
Id	
Type	document
Description	User's need in the app, representing what kind of help he/she needs
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
DBRef	
Min Properties	
Max Properties	
Additional properties	false
Comments	

2.1.2.6.3.14 Field medicalHelp

2.1.2.6.3.14.1 medicalHelp Tree Diagram



2.1.2.6.3.14.2 medicalHelp properties

PROPERTY	VALUE
Name	medicalHelp
Technical name	
Id	
Type	boolean
Description	Flag to indicate wheter the user needs medical help. Marking this indicates he shows symptoms of COVID-19.
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	false
Sample	false
Comments	

2.1.2.6.3.15 Field otherHelp**2.1.2.6.3.15.1 otherHelp Tree Diagram**

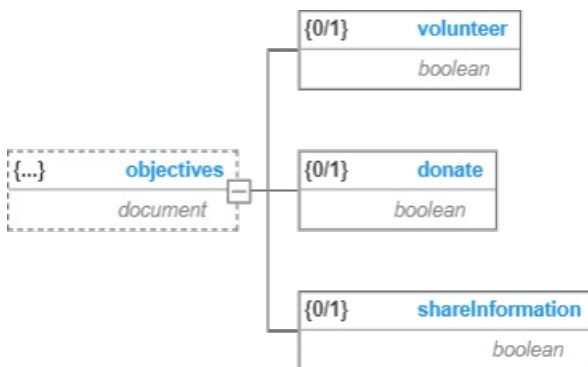
{0/1}	otherHelp
	<i>boolean</i>

2.1.2.6.3.15.2 otherHelp properties

PROPERTY	VALUE
Name	otherHelp
Technical name	
Id	
Type	boolean
Description	Flag to indicate wheter the user needs non-medical help, such as getting groceries, medice and etc..
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	false
Sample	false
Comments	

2.1.2.6.3.16 Field objectives

2.1.2.6.3.16.1 objectives Tree Diagram



2.1.2.6.3.16.2 objectives Hierarchy

Parent field: **[0] subschema**

Child field(s):

FIELD	TYPE	REQ	KEY	DESCRIPTION	COMMENTS
volunteer	boolean	true		Flag to indicate wheter the user wants to volunteer for iniciatives through the app.	
donate	boolean	true		Flag to indicate wheter the user wants to donate for iniciatives through the app.	
shareInformation	boolean	true		Flag to indicate wheter the user wants to share information through the app.	

2.1.2.6.3.16.3 objectives properties

PROPERTY	VALUE
Name	objectives
Technical name	
Id	
Type	document
Description	Document with what are the objectives of the user through the application, such as donating, volunteering or sharing information.
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
DBRef	
Min Properties	
Max Properties	
Additional properties	false
Comments	

2.1.2.6.3.17 Field volunteer

2.1.2.6.3.17.1 volunteer Tree Diagram

{0/1}	volunteer
	<i>boolean</i>

2.1.2.6.3.17.2 volunteer properties

PROPERTY	VALUE
Name	volunteer
Technical name	
Id	
Type	boolean
Description	Flag to indicate wheter the user wants to volunteer for iniciatives through the app.
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	false
Sample	false
Comments	

2.1.2.6.3.18 Field donate

2.1.2.6.3.18.1 donate Tree Diagram

{0/1}	donate
	<i>boolean</i>

2.1.2.6.3.18.2 donate properties

PROPERTY	VALUE
Name	donate
Technical name	
Id	
Type	boolean
Description	Flag to indicate wheter the user wants to donate for iniciatives through the app.
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	false
Sample	false
Comments	

2.1.2.6.3.19 Field shareInformation

2.1.2.6.3.19.1 shareInformation Tree Diagram

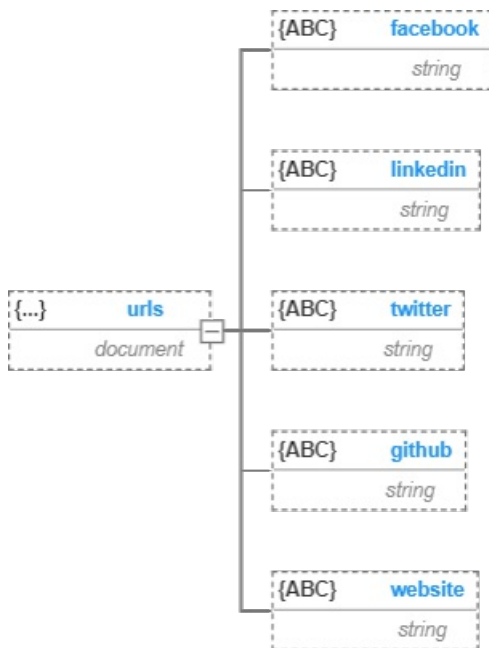
{0/1}	shareInformation
	<i>boolean</i>

2.1.2.6.3.19.2 shareInformation properties

PROPERTY	VALUE
Name	shareInformation
Technical name	
Id	
Type	boolean
Description	Flag to indicate wheter the user wants to share information through the app.
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	true
Sample	true
Comments	

2.1.2.6.3.20 Field urls

2.1.2.6.3.20.1 urls Tree Diagram



2.1.2.6.3.20.2 urls Hierarchy

Parent field: **[0] subschema**

Child field(s):

FIELD	TYPE	REQ	KEY	DESCRIPTION	COMMENTS
facebook	string	false		URL to user's Facebook page.	
linkedin	string	false		URL to user's Linkedin page.	
twitter	string	false		URL to user's Twitter page.	
github	string	false		URL to user's Github page.	
website	string	false		URL to user's personal website.	

2.1.2.6.3.20.3 urls properties

PROPERTY	VALUE
Name	urls
Technical name	
Id	
Type	document
Description	Document holding the user urls to external websites and social networks
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
DBRef	
Min Properties	
Max Properties	
Additional properties	false
Comments	

2.1.2.6.3.21 Field facebook**2.1.2.6.3.21.1 facebook Tree Diagram****2.1.2.6.3.21.2 facebook properties**

PROPERTY	VALUE
Name	facebook
Technical name	
Id	
Type	string
Description	URL to user's Facebook page.
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	uri
Enum	
Sample	http://facebook.com/myProfile
Comments	

2.1.2.6.3.22 Field linkedin**2.1.2.6.3.22.1 linkedin Tree Diagram**

2.1.2.6.3.22.2 linkedin properties

PROPERTY	VALUE
Name	linkedin
Technical name	
Id	
Type	string
Description	URL to user's Linkedin page.
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	uri
Enum	
Sample	http://linkedin.com/in/myProfile
Comments	

2.1.2.6.3.23 Field twitter**2.1.2.6.3.23.1 twitter Tree Diagram****2.1.2.6.3.23.2 twitter properties**

PROPERTY	VALUE
Name	twitter
Technical name	
Id	
Type	string
Description	URL to user's Twitter page.
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	uri
Enum	
Sample	http://twitter.com/myProfile
Comments	

2.1.2.6.3.24 Field github

2.1.2.6.3.24.1 github Tree Diagram



2.1.2.6.3.24.2 github properties

PROPERTY	VALUE
Name	github
Technical name	
Id	
Type	string
Description	URL to user's Github page.
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	uri
Enum	
Sample	http://github.com/myProfile
Comments	

2.1.2.6.3.25 Field website

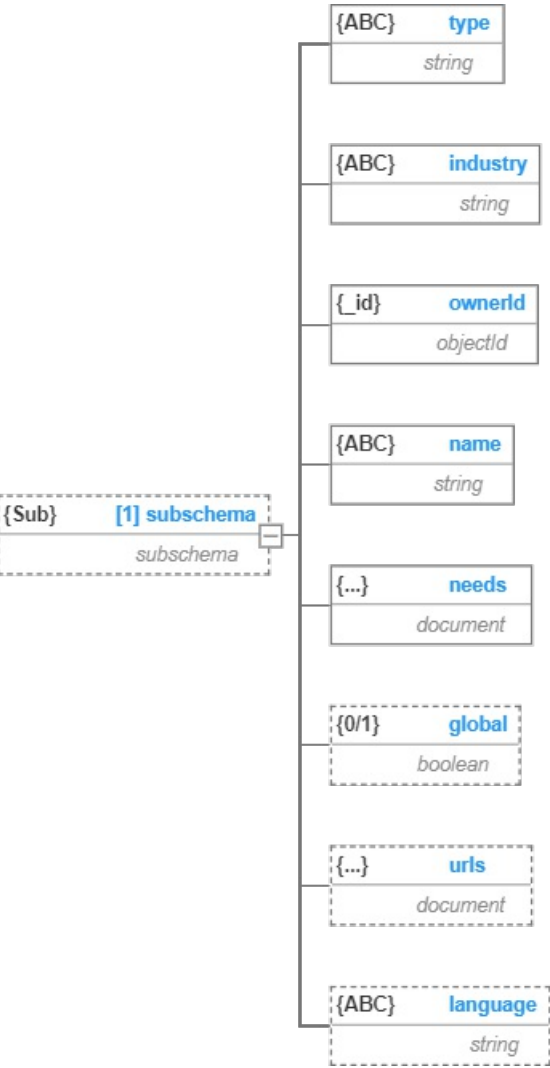
2.1.2.6.3.25.1 website Tree Diagram



2.1.2.6.3.25.2 website properties

PROPERTY	VALUE
Name	website
Technical name	
Id	
Type	string
Description	URL to user's personal website.
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	uri
Enum	
Sample	http://mypersonalwebsite.com
Comments	

2.1.2.6.3.26 Field [1] subschema**2.1.2.6.3.26.1 [1] subschema Tree Diagram**



2.1.2.6.3.26.2 [1] subschema Hierarchy

Parent field: **oneOf**
Child field(s):

FIELD	TYPE	REQ	KEY	DESCRIPTION	COMMENTS
type	string	true		The type of the organization. Can be: Community, Government, Health care provider, Non-profit, Other, R&D, Startup, Traditional Company and University. All names are store in lowercase only.	
industry	string	true		Industry from which the organization participates. Plan is for this values to be scraped from linkedin industries.	
ownerId	objectId	true	fk	_id of the user who owns the organization.	
name	string	true		Organization's name.	
needs	document	true		Document with what are the objectives of the organization through the application, such as finding volunteers, staff, investors and donors.	
global	boolean	false		Flag to indicate wheter this organization is global.	
urls	document	false		Document holding external urls and refs to the organization such as social media profiles.	
language	string	false		Field to inidicate the language used by the organization.	

2.1.2.6.3.26.3 [1] subschema properties

PROPERTY	VALUE
Display name	subschema
Id	organization
Description	Subschema representing an organization (colective) account.
Required	
Min Properties	
Max Properties	
Additional properties	false
Comments	

2.1.2.6.3.27 Field type

2.1.2.6.3.27.1 type Tree Diagram

{ABC}	type
	string

2.1.2.6.3.27.2 type properties

PROPERTY	VALUE
Name	type
Technical name	
Id	
Type	string
Description	The type of the organization. Can be: Community, Government, Health care provider, Non-profit, Other, R&D, Startup, Traditional Company and University. All names are store in lowercase only.
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	startup,traditional company,community,government,r&d,non-profit,university,health care provider,other
Sample	
Comments	

2.1.2.6.3.28 Field industry**2.1.2.6.3.28.1 industry Tree Diagram**

{ABC}	industry
	string

2.1.2.6.3.28.2 industry properties

PROPERTY	VALUE
Name	industry
Technical name	
Id	
Type	string
Description	Industry from which the organization participates. Plan is for this values to be scraped from linkedin industries.
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	
Sample	
Comments	

2.1.2.6.3.29 Field ownerId**2.1.2.6.3.29.1 ownerId Tree Diagram****2.1.2.6.3.29.2 ownerId properties**

PROPERTY	VALUE
Name	ownerId
Technical name	
Id	
Type	objectId
Description	_id of the user who owns the organization.
Dependencies	
Required	true
Primary key	false
Foreign collection	users
Foreign field	_id
Relationship type	Foreign Key
Default	
Sample	
Comments	

2.1.2.6.3.30 Field name

2.1.2.6.3.30.1 name Tree Diagram

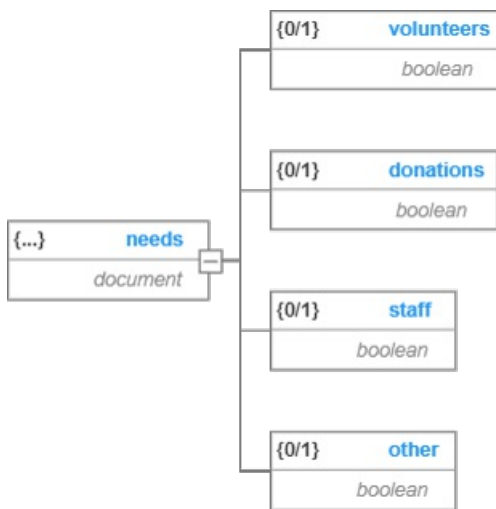
{ABC}	name
	string

2.1.2.6.3.30.2 name properties

PROPERTY	VALUE
Name	name
Technical name	
Id	
Type	string
Description	Organization's name.
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	
Sample	
Comments	

2.1.2.6.3.31 Field needs

2.1.2.6.3.31.1 needs Tree Diagram



2.1.2.6.3.31.2 needs Hierarchy

Parent field: **[1] subschema**

Child field(s):

FIELD	TYPE	REQ	KEY	DESCRIPTION	COMMENTS
volunteers	boolean	true		Flag to indicate wheter the organization is seeking volunteers through the app.	
donations	boolean	true		Flag to indicate wheter the organization is seeking donations through the app.	
staff	boolean	true		Flag to indicate wheter the organization is seeking staff through the app.	
other	boolean	true		Flag to indicate wheter the organization is seeking other needs through the app.	

2.1.2.6.3.31.3 needs properties

PROPERTY	VALUE
Name	needs
Technical name	
Id	
Type	document
Description	Document with what are the objectives of the organization through the application, such as finding volunteers, staff, investors and donors.
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
DBRef	
Min Properties	
Max Properties	
Additional properties	false
Comments	

2.1.2.6.3.32 Field volunteers**2.1.2.6.3.32.1 volunteers Tree Diagram**

{0/1}	volunteers
	<i>boolean</i>

2.1.2.6.3.32.2 volunteers properties

PROPERTY	VALUE
Name	volunteers
Technical name	
Id	
Type	boolean
Description	Flag to indicate wheter the organization is seeking volunteers through the app.
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	false
Sample	true
Comments	

2.1.2.6.3.33 Field donations**2.1.2.6.3.33.1 donations Tree Diagram**

{0/1}	donations
	<i>boolean</i>

2.1.2.6.3.33.2 donations properties

PROPERTY	VALUE
Name	donations
Technical name	
Id	
Type	boolean
Description	Flag to indicate wheter the organization is seeking donations through the app.
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	false
Sample	false
Comments	

2.1.2.6.3.34 Field staff**2.1.2.6.3.34.1 staff Tree Diagram**

{0/1}	staff
	boolean

2.1.2.6.3.34.2 staff properties

PROPERTY	VALUE
Name	staff
Technical name	
Id	
Type	boolean
Description	Flag to indicate wheter the organization is seeking staff through the app.
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	false
Sample	false
Comments	

2.1.2.6.3.35 Field other**2.1.2.6.3.35.1 other Tree Diagram**

{0/1}	other
	boolean

2.1.2.6.3.35.2 other properties

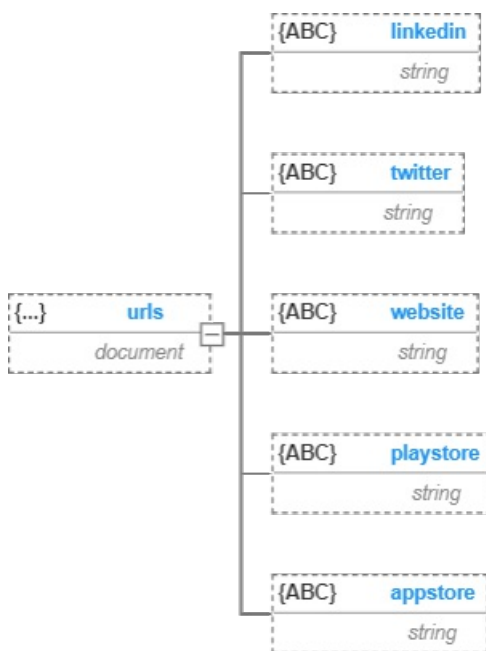
PROPERTY	VALUE
Name	other
Technical name	
Id	
Type	boolean
Description	Flag to indicate wheter the organization is seeking other needs through the app.
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	false
Sample	false
Comments	

2.1.2.6.3.36 Field global**2.1.2.6.3.36.1 global Tree Diagram****2.1.2.6.3.36.2 global properties**

PROPERTY	VALUE
Name	global
Technical name	
Id	
Type	boolean
Description	Flag to indicate wheter this organization is global.
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	false
Sample	
Comments	

2.1.2.6.3.37 Field urls

2.1.2.6.3.37.1 urls Tree Diagram



2.1.2.6.3.37.2 urls Hierarchy

Parent field: **[1] subschema**

Child field(s):

FIELD	TYPE	REQ	KEY	DESCRIPTION	COMMENTS
linkedin	string	false		URL to organization's Linkedin page.	
twitter	string	false		URL to organization's Twitter page.	
website	string	false		URL to organization's website.	
playstore	string	false		URL to organization's page at Google Play Store.	
appstore	string	false		URL to organization's page at Apple App Store	

2.1.2.6.3.37.3 urls properties

PROPERTY	VALUE
Name	urls
Technical name	
Id	
Type	document
Description	Document holding external urls and refs to the organization such as social media profiles.
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
DBRef	
Min Properties	
Max Properties	
Additional properties	false
Comments	

2.1.2.6.3.38 Field linkedin**2.1.2.6.3.38.1 linkedin Tree Diagram****2.1.2.6.3.38.2 linkedin properties**

PROPERTY	VALUE
Name	linkedin
Technical name	
Id	
Type	string
Description	URL to organization's LinkedIn page.
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	uri
Enum	
Sample	http://linkedin.com/in/myProfile
Comments	

2.1.2.6.3.39 Field twitter**2.1.2.6.3.39.1 twitter Tree Diagram**

2.1.2.6.3.39.2 twitter properties

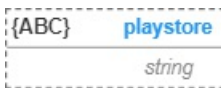
PROPERTY	VALUE
Name	twitter
Technical name	
Id	
Type	string
Description	URL to organization's Twitter page.
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	uri
Enum	
Sample	http://twitter.com/myProfile
Comments	

2.1.2.6.3.40 Field website**2.1.2.6.3.40.1 website Tree Diagram****2.1.2.6.3.40.2 website properties**

PROPERTY	VALUE
Name	website
Technical name	
Id	
Type	string
Description	URL to organization's website.
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	uri
Enum	
Sample	http://mypersonalwebsite.com
Comments	

2.1.2.6.3.41 Field playstore

2.1.2.6.3.41.1 playstore Tree Diagram

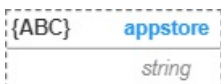


2.1.2.6.3.41.2 playstore properties

PROPERTY	VALUE
Name	playstore
Technical name	
Id	
Type	string
Description	URL to organization's page at Google Play Store.
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	uri
Enum	
Sample	http://playstore.com/myOrg
Comments	

2.1.2.6.3.42 Field appstore

2.1.2.6.3.42.1 appstore Tree Diagram



2.1.2.6.3.42.2 appstore properties

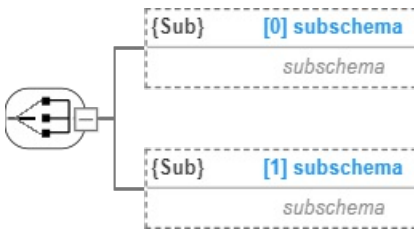
PROPERTY	VALUE
Name	appstore
Technical name	
Id	
Type	string
Description	URL to organization's page at Apple App Store
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	uri
Enum	
Sample	http://appstore.com/myOrg
Comments	

2.1.2.6.3.43 Field language**2.1.2.6.3.43.1 language Tree Diagram****2.1.2.6.3.43.2 language properties**

PROPERTY	VALUE
Name	language
Technical name	
Id	
Type	string
Description	Field to indicate the language used by the organization.
Dependencies	
Required	false
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	
Sample	
Comments	

2.1.2.6.3.44 Field anyOf

2.1.2.6.3.44.1 anyOf Tree Diagram



2.1.2.6.3.44.2 anyOf Hierarchy

Parent field: **users**

Child field(s):

FIELD	TYPE	REQ	KEY	DESCRIPTION	COMMENTS
subschema	subschema	false		Case where the user comes from social networks with a registration that uses email. In this case, the email field is required. This case covers the great majority of currently supported social network authentications.	
subschema	subschema	false		Case where the user comes from social networks with a registration that allows the use of only the phone. In this case, the phone field is required, and the email isn't. An example of ocial network that falls into this case is Twitter.	

2.1.2.6.3.44.3 anyOf properties

PROPERTY	VALUE
Choice	anyOf
Description	
Comments	

2.1.2.6.3.45 Field [0] subschema

2.1.2.6.3.45.1 [0] subschema Tree Diagram



2.1.2.6.3.45.2 [0] subschema Hierarchy

Parent field: **anyOf**

Child field(s):

FIELD	TYPE	REQ	KEY	DESCRIPTION	COMMENTS
email	string	true		User's email validated by a regex pattern.	

2.1.2.6.3.45.3 [0] subschema properties

PROPERTY	VALUE
Display name	subschema
Id	emailPK
Description	Case where the user comes from social networks with a registration that uses email. In this case, the email field is required. This case covers the great majority of currently supported social network authentications.
Required	
Min Properties	
Max Properties	
Additional properties	true
Comments	

2.1.2.6.3.46 Field email**2.1.2.6.3.46.1 email Tree Diagram**

{ABC}	email
	string

2.1.2.6.3.46.2 email properties

PROPERTY	VALUE
Name	email
Technical name	
Id	
Type	string
Description	User's email validated by a regex pattern.
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	email
Enum	
Sample	test_user@fightpandemics.com
Comments	

2.1.2.6.3.47 Field [1] subschema

2.1.2.6.3.47.1 [1] subschema Tree Diagram**2.1.2.6.3.47.2 [1] subschema Hierarchy**Parent field: **anyOf**

Child field(s):

FIELD	TYPE	REQ	KEY	DESCRIPTION	COMMENTS
phone	string	true			

2.1.2.6.3.47.3 [1] subschema properties

PROPERTY	VALUE
Display name	subschema
Id	phonePK
Description	Case where the user comes from social networks with a registration that allows the use of only the phone. In this case, the phone field is required, and the email isn't. An example of ocial network that falls into this case is Twitter.
Required	
Min Properties	
Max Properties	
Additional properties	false
Comments	

2.1.2.6.3.48 Field phone**2.1.2.6.3.48.1 phone Tree Diagram**

2.1.2.6.3.48.2 phone properties

PROPERTY	VALUE
Name	phone
Technical name	
Id	
Type	string
Description	
Dependencies	
Required	true
Primary key	false
Foreign collection	
Foreign field	
Relationship type	
Default	
Min length	
Max length	
Pattern	
Format	
Enum	
Sample	
Comments	

2.1.2.6.4 users JSON Schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "users",
  "description": "Collection that holds all the data on system's users. \n\nThe users can be of **two types**: *(1) Individual**, representing a private user account; and *(2) Organization**, representing an organization of various types.\n\nThe field 'type' is used to differentiate between the two subschemas, with 'individual' representing private user accounts and 'Community', 'Government', 'Health care provider', 'Non-profit', 'Other', 'R&D', 'Startup', 'Traditional Company' and 'University'.",
  "additionalProperties": true,
  "properties": {
    "createdAt": {
      "type": "string",
      "description": "Timestamp with the user's creation date",
      "format": "date-time"
    },
    "_id": {
      "type": "string",
      "description": "Automatically generated _id as primary key.",
      "pattern": "^a-fA-F0-9]{24}$"
    },
    "updatedAt": {
      "type": "string",
      "description": "Timestamp indicating last update on the database.",
      "format": "date-time"
    },
    "authId": {
      "type": "string",
      "description": "String holding the authentication service id (Auth0).\n\nThis field is a **foreign key to the user in the Auth0 database**, connecting through the user_id field."
    },
    "location": {
      "$ref": "#model/definitions/location"
    },
    "about": {
      "type": "string",
      "description": "Small self-introduction, limited to 100 char"
    },
    "photo": {
```

```

    "type": "string",
    "description": "URI with the location of the profile's photo.",
    "format": "uri"
  }
},
"oneOf": [
  {
    "id": "individual",
    "type": "object",
    "description": "Subschema representing a individual account.",
    "properties": {
      "type": {
        "type": "string",
        "enum": [
          "individual"
        ]
      },
      "firstName": {
        "type": "string",
        "description": "The first name of the user"
      },
      "lastName": {
        "type": "string",
        "description": "The last name of the user. Optional field."
      },
      "needs": {
        "type": "object",
        "description": "User's need in the app, representing what kind of help he/she needs",
        "properties": {
          "medicalHelp": {
            "type": "boolean",
            "description": "Flag to indicate wheter the user needs medical help. \nMarking this indicates he shows symptoms of COVID-19.",
            "default": false
          },
          "otherHelp": {
            "type": "boolean",
            "description": "Flag to indicate wheter the user needs non-medical help, such as getting groceries, medice and etc..",
            "default": false
          }
        }
      },
      "additionalProperties": false,
      "required": [
        "medicalHelp",
        "otherHelp"
      ]
    }
  },
  {
    "id": "objectives",
    "type": "object",
    "description": "Document with what are the objectives of the user through the application, such as donating, volunteering or sharing information.",
    "properties": {
      "volunteer": {
        "type": "boolean",
        "description": "Flag to indicate wheter the user wants to volunteer for iniciatives through the app.",
        "default": false
      },
      "donate": {
        "type": "boolean",
        "description": "Flag to indicate wheter the user wants to donate for iniciatives through the app.",
        "default": false
      },
      "shareInformation": {
        "type": "boolean",
        "description": "Flag to indicate wheter the user wants to share information through the app.",
        "default": true
      }
    },
    "additionalProperties": false,
    "required": [
      "volunteer",

```



```

        "donate",
        "shareInformation"
    ]
},
"urls": {
    "type": "object",
    "description": "Document holding the user urls to external websites and social networks",
    "properties": {
        "facebook": {
            "type": "string",
            "description": "URL to user's Facebook page.",
            "format": "uri"
        },
        "linkedin": {
            "type": "string",
            "description": "URL to user's Linkedin page.",
            "format": "uri"
        },
        "twitter": {
            "type": "string",
            "description": "URL to user's Twitter page.",
            "format": "uri"
        },
        "github": {
            "type": "string",
            "description": "URL to user's Github page.",
            "format": "uri"
        },
        "website": {
            "type": "string",
            "description": "URL to user's personal website.",
            "format": "uri"
        }
    }
},
"additionalProperties": false
}
},
"additionalProperties": true,
"required": [
    "type",
    "firstName",
    "needs"
]
},
{
    "id": "organization",
    "type": "object",
    "description": "Subschema representing an organization (colective) account.",
    "properties": {
        "type": {
            "type": "string",
            "description": "The type of the organization. Can be: Community, Government, Health care provider, Non-profit, Other, R&D, Startup, Traditional Company and University. All names are store in lowercase only.",
            "enum": [
                "startup",
                "traditional company",
                "community",
                "government",
                "r&d",
                "non-profit",
                "university",
                "health care provider",
                "other"
            ]
        },
        "industry": {
            "type": "string",
            "description": "Industry from which the organization participates. Plan is for this values to be scraped from linkedin industries."
        },
        "ownerId": {

```

```

    "type": "string",
    "description": "_id of the user who owns the organization.",
    "pattern": "^a-fA-F0-9{24}$"
  },
  "name": {
    "type": "string",
    "description": "Organization's name.\n"
  },
  "needs": {
    "type": "object",
    "description": "Document with what are the objectives of the organization through the application, such as finding volunteers, staff, investors and donors.",
    "properties": {
      "volunteers": {
        "type": "boolean",
        "description": "Flag to indicate wheter the organization is seeking volunteers through the app.",
        "default": false
      },
      "donations": {
        "type": "boolean",
        "description": "Flag to indicate wheter the organization is seeking donations through the app.",
        "default": false
      },
      "staff": {
        "type": "boolean",
        "description": "Flag to indicate wheter the organization is seeking staff through the app.",
        "default": false
      },
      "other": {
        "type": "boolean",
        "description": "Flag to indicate wheter the organization is seeking other needs through the app.",
        "default": false
      }
    }
  },
  "additionalProperties": false,
  "required": [
    "volunteers",
    "donations",
    "staff",
    "other"
  ]
},
"global": {
  "type": "boolean",
  "description": "Flag to indicate wheter this organization is global.",
  "default": false
},
"urls": {
  "type": "object",
  "description": "Document holding external urls and refs to the organization such as social media profiles.",
  "properties": {
    "linkedin": {
      "type": "string",
      "description": "URL to organization's Linkedin page.",
      "format": "uri"
    },
    "twitter": {
      "type": "string",
      "description": "URL to organization's Twitter page.",
      "format": "uri"
    },
    "website": {
      "type": "string",
      "description": "URL to organization's website.",
      "format": "uri"
    },
    "playstore": {
      "type": "string",
      "description": "URL to organization's page at Google Play Store.",
      "format": "uri"
    }
  }
}

```

```

    },
    "appstore": {
      "type": "string",
      "description": "URL to organization's page at Apple App Store",
      "format": "uri"
    },
    },
    "additionalProperties": false
  },
  "language": {
    "type": "string",
    "description": "Field to indicate the language used by the organization."
  },
  },
  "additionalProperties": false,
  "required": [
    "type",
    "industry",
    "ownerId",
    "name",
    "needs"
  ]
}
],
"anyOf": [
  {
    "id": "emailPK",
    "type": "object",
    "description": "Case where the user comes from social networks with a registration that uses email. In this case, the email field is required. This case covers the great majority of currently supported social network authentications.",
    "properties": {
      "email": {
        "type": "string",
        "description": "User's email validated by a regex pattern.",
        "format": "email"
      }
    },
    },
    "additionalProperties": true,
    "required": [
      "email"
    ]
  },
  {
    "id": "phonePK",
    "type": "object",
    "description": "Case where the user comes from social networks with a registration that allows the use of only the phone. In this case, the phone field is required, and the email isn't. \n\nAn example of social network that falls into this case is Twitter.",
    "properties": {
      "phone": {
        "type": "string"
      }
    },
    },
    "additionalProperties": false,
    "required": [
      "phone"
    ]
  }
],
"required": [
  "createdAt",
  "_id",
  "updatedAt",
  "authId"
]
}

```

2.1.2.6.5 users JSON data

```
{
  "createdAt": ISODate("2016-04-08T15:06:21.595Z"),
  "_id": ObjectId("507f1f77bcf86cd799439011"),
  "updatedAt": ISODate("2016-04-08T15:06:21.595Z"),
  "authId": "Lorem",
  "location": {
    "coordinates": [
      35.26,
      -32.83
    ],
    "type": "Point",
    "country": "Lorem",
    "city": "Lorem",
    "neighborhood": "Lorem",
    "address": "Lorem"
  },
  "about": "Lorem",
  "photo": "/resource.txt#frag01",
  "type": "other",
  "industry": "Lorem",
  "ownerId": ObjectId("507f1f77bcf86cd799439011"),
  "name": "Lorem",
  "needs": {
    "volunteers": true,
    "donations": false,
    "staff": false,
    "other": false
  },
  "global": false,
  "urls": {
    "linkedin": "http://linkedin.com/in/myProfile",
    "twitter": "http://twitter.com/myProfile",
    "website": "http://mypersonalwebsite.com",
    "playstore": "http://playstore.com/myOrg",
    "appstore": "http://appstore.com/myOrg"
  },
  "language": "Lorem",
  "email": "test_user@fightpandemics.com"
}
```

2.1.2.6.6 users Target Script

```
use applicationDB;

db.createCollection( "users",{
  "storageEngine": {
    "wiredTiger": {}
  },
  "capped": false,
  "validator": {
    "$jsonSchema": {
      "bsonType": "object",
      "title": "users",
      "description": "Collection that holds all the data on system's users. \n\nThe users can be of **two types**: *(1) Individual**, representing a private user account; and *(2) Organization**, representing an organization of various types.\n\nThe field 'type' is used to differentiate between the two subschemas, with 'individual' representing private user accounts and 'Community', 'Government', 'Health care provider', 'Non-profit', 'Other', 'R&D', 'Startup', 'Traditional Company' and 'University'.",
      "additionalProperties": true,
      "properties": {
        "_id": {
          "bsonType": "objectId",
          "description": "Automatically generated _id as primary key."
        },
        "createdAt": {
```

```

    "bsonType": "date",
    "description": "Timestamp with the user's creation date"
  },
  "updatedAt": {
    "bsonType": "date",
    "description": "Timestamp indicating last update on the database."
  },
  "authId": {
    "bsonType": "string",
    "description": "String holding the authentication service id (Auth0).\n\nThis field is a **foreign key to the user in the Auth0 database**, connecting through the user_id field."
  },
  "location": {
    "bsonType": "object",
    "description": "Location document according to the norms of a [GeoJSON Object]
    (https://docs.mongodb.com/manual/reference/geojson/ ).\n\nThe fields address, neighborhood, city, state and country are added to store the result of the geoprocessing.",
    "properties": {
      "coordinates": {
        "bsonType": "array",
        "description": "Array with the coordinates, specifying **longitude first** and **latitude second**, as default in the [GeoJSON]
        (https://docs.mongodb.com/manual/reference/geojson/) definition.",
        "additionalItems": true,
        "minItems": 2,
        "maxItems": 2,
        "uniqueItems": false,
        "items": [
          {
            "bsonType": "number",
            "description": "Location longitude\n",
            "minimum": -180,
            "maximum": 180
          },
          {
            "bsonType": "number",
            "description": "Location latitude\n",
            "minimum": -90,
            "maximum": 90
          }
        ]
      },
      "type": {
        "bsonType": "string",
        "description": "Type of the GeoJSON Object.\nPlease refer to the [official documentation]
        (https://docs.mongodb.com/manual/reference/geojson/ ) for more information.\n\n**In our case, the value will always be of the type 'Point'.**",
        "enum": [
          "Point"
        ]
      },
      "country": {
        "bsonType": "string",
        "description": "String holding the country's name.\n"
      },
      "city": {
        "bsonType": "string",
        "description": "String holding the city's name."
      },
      "neighborhood": {
        "bsonType": "string",
        "description": "String holding the neighborhood's name."
      },
      "address": {
        "bsonType": "string",
        "description": "String holding the address (street, number and other details).\"
      }
    },
    "additionalProperties": false,
    "required": [
      "coordinates",
      "type"
    ]
  }
}

```

```

    ]
  },
  "about": {
    "bsonType": "string",
    "description": "Small self-introduction, limited to 100 char"
  },
  "photo": {
    "bsonType": "string",
    "description": "URI with the location of the profile's photo."
  }
},
"oneOf": [
  {
    "bsonType": "object",
    "description": "Subschema representing a individual account.",
    "properties": {
      "type": {
        "bsonType": "string",
        "enum": [
          "individual"
        ]
      },
      "firstName": {
        "bsonType": "string",
        "description": "The first name of the user"
      },
      "lastName": {
        "bsonType": "string",
        "description": "The last name of the user. Optional field."
      },
      "needs": {
        "bsonType": "object",
        "description": "User's need in the app, representing what kind of help he/she needs",
        "properties": {
          "medicalHelp": {
            "bsonType": "bool",
            "description": "Flag to indicate wheter the user needs medical help. \nMarking this indicates he shows symptoms of COVID-
19."
          },
          "otherHelp": {
            "bsonType": "bool",
            "description": "Flag to indicate wheter the user needs non-medical help, such as getting groceries, medice and etc.."
          }
        },
        "additionalProperties": false,
        "required": [
          "medicalHelp",
          "otherHelp"
        ]
      },
      "objectives": {
        "bsonType": "object",
        "description": "Document with what are the objectives of the user through the application, such as donating, volunteering or
sharing information.",
        "properties": {
          "volunteer": {
            "bsonType": "bool",
            "description": "Flag to indicate wheter the user wants to volunteer for iniciatives through the app."
          },
          "donate": {
            "bsonType": "bool",
            "description": "Flag to indicate wheter the user wants to donate for iniciatives through the app."
          },
          "shareInformation": {
            "bsonType": "bool",
            "description": "Flag to indicate wheter the user wants to share information through the app."
          }
        },
        "additionalProperties": false,
        "required": [

```

```

        "volunteer",
        "donate",
        "shareInformation"
    ]
},
"urls": {
    "bsonType": "object",
    "description": "Document holding the user urls to external websites and social networks",
    "properties": {
        "facebook": {
            "bsonType": "string",
            "description": "URL to user's Facebook page."
        },
        "linkedin": {
            "bsonType": "string",
            "description": "URL to user's Linkedin page."
        },
        "twitter": {
            "bsonType": "string",
            "description": "URL to user's Twitter page."
        },
        "github": {
            "bsonType": "string",
            "description": "URL to user's Github page."
        },
        "website": {
            "bsonType": "string",
            "description": "URL to user's personal website."
        }
    }
},
"additionalProperties": false
}
},
"additionalProperties": true,
"required": [
    "type",
    "firstName",
    "needs"
]
},
{
    "bsonType": "object",
    "description": "Subschema representing an organization (colective) account.",
    "properties": {
        "type": {
            "bsonType": "string",
            "description": "The type of the organization. Can be: Community, Government, Health care provider, Non-profit, Other, R&D, Startup, Traditional Company and University. All names are store in lowercase only.",
            "enum": [
                "startup",
                "traditional company",
                "community",
                "government",
                "r&d",
                "non-profit",
                "university",
                "health care provider",
                "other"
            ]
        },
        "industry": {
            "bsonType": "string",
            "description": "Industry from which the organization participates. Plan is for this values to be scraped from linkedin industries."
        },
        "ownerId": {
            "bsonType": "objectId",
            "description": "_id of the user who owns the organization."
        },
        "name": {
            "bsonType": "string",

```

```

    "description": "Organization's name.\n"
  },
  "needs": {
    "bsonType": "object",
    "description": "Document with what are the objectives of the organization through the application, such as finding volunteers,
staff, investors and donors.",
    "properties": {
      "volunteers": {
        "bsonType": "bool",
        "description": "Flag to indicate wheter the organization is seeking volunteers through the app."
      },
      "donations": {
        "bsonType": "bool",
        "description": "Flag to indicate wheter the organization is seeking donations through the app."
      },
      "staff": {
        "bsonType": "bool",
        "description": "Flag to indicate wheter the organization is seeking staff through the app."
      },
      "other": {
        "bsonType": "bool",
        "description": "Flag to indicate wheter the organization is seeking other needs through the app."
      }
    },
    "additionalProperties": false,
    "required": [
      "volunteers",
      "donations",
      "staff",
      "other"
    ]
  },
  "global": {
    "bsonType": "bool",
    "description": "Flag to indicate wheter this organization is global."
  },
  "urls": {
    "bsonType": "object",
    "description": "Document holding external urls and refs to the organization such as social media profiles.",
    "properties": {
      "linkedin": {
        "bsonType": "string",
        "description": "URL to organization's Linkedin page."
      },
      "twitter": {
        "bsonType": "string",
        "description": "URL to organization's Twitter page."
      },
      "website": {
        "bsonType": "string",
        "description": "URL to organization's website."
      },
      "playstore": {
        "bsonType": "string",
        "description": "URL to organization's page at Google Play Store."
      },
      "appstore": {
        "bsonType": "string",
        "description": "URL to organization's page at Apple App Store"
      }
    },
    "additionalProperties": false
  },
  "language": {
    "bsonType": "string",
    "description": "Field to indicate the language used by the organization."
  }
},
"additionalProperties": false,
"required": [

```



```

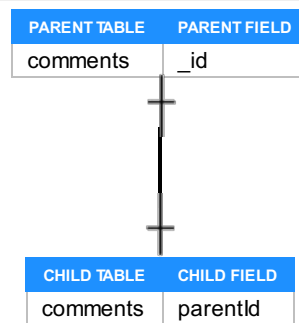
        "type",
        "industry",
        "ownerId",
        "name",
        "needs"
    ]
}
],
"anyOf": [
    {
        "bsonType": "object",
        "description": "Case where the user comes from social networks with a registration that uses email. In this case, the email field is required. This case covers the great majority of currently supported social network authentications.",
        "properties": {
            "email": {
                "bsonType": "string",
                "description": "User's email validated by a regex pattern."
            }
        },
        "additionalProperties": true,
        "required": [
            "email"
        ]
    },
    {
        "bsonType": "object",
        "description": "Case where the user comes from social networks with a registration that allows the use of only the phone. In this case, the phone field is required, and the email isn't. \n\nAn example of social network that falls into this case is Twitter.",
        "properties": {
            "phone": {
                "bsonType": "string"
            }
        },
        "additionalProperties": false,
        "required": [
            "phone"
        ]
    }
],
"required": [
    "createdAt",
    "_id",
    "updatedAt",
    "authId"
]
}
},
"validationLevel": "off",
"validationAction": "warn"
});

```

3. RELATIONSHIPS

3.1 Relationship CommentReferencesPost

3.1.1 CommentReferencesPost Diagram

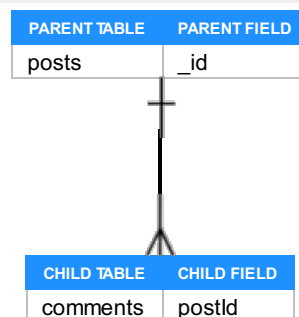


3.1.2 CommentReferencesPost Properties

PROPERTY	VALUE
Name	CommentReferencesPost
Description	
Parent Collection	comments
Parent field	_id
Parent Cardinality	1
Child Collection	comments
Child field	parentId
Child Cardinality	1
Comments	

3.2 Relationship PostHasComments

3.2.1 PostHasComments Diagram

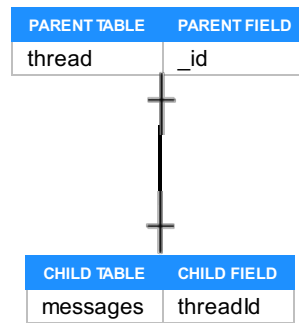


3.2.2 PostHasComments Properties

PROPERTY	VALUE
Name	PostHasComments
Description	Relationship between a Post and it's respective Comments made at it.
Parent Collection	posts
Parent field	_id
Parent Cardinality	1
Child Collection	comments
Child field	postId
Child Cardinality	n
Comments	

3.3 Relationship ThreadHasMessages

3.3.1 ThreadHasMessages Diagram

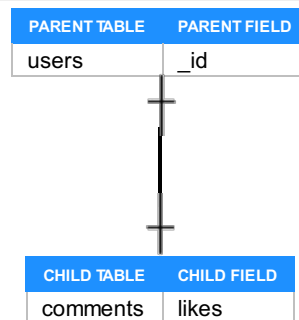


3.3.2 ThreadHasMessages Properties

PROPERTY	VALUE
Name	ThreadHasMessages
Description	
Parent Collection	thread
Parent field	<u>_id</u>
Parent Cardinality	1
Child Collection	messages
Child field	<u>threadId</u>
Child Cardinality	1
Comments	

3.4 Relationship UserAuthorsComment

3.4.1 UserAuthorsComment Diagram

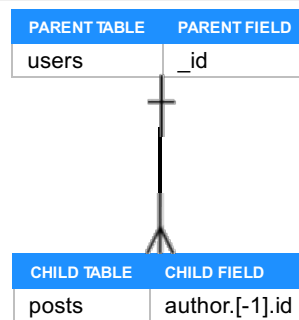


3.4.2 UserAuthorsComment Properties

PROPERTY	VALUE
Name	UserAuthorsComment
Description	
Parent Collection	users
Parent field	<u>_id</u>
Parent Cardinality	1
Child Collection	comments
Child field	<u>likes</u>
Child Cardinality	1
Comments	

3.5 Relationship UserAuthorsPost

3.5.1 UserAuthorsPost Diagram

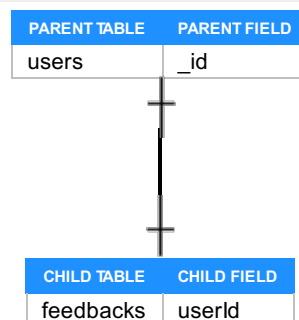


3.5.2 UserAuthorsPost Properties

PROPERTY	VALUE
Name	UserAuthorsPost
Description	Relation between a User and the Post created by him/her.
Parent Collection	users
Parent field	_id
Parent Cardinality	1
Child Collection	posts
Child field	id
Child Cardinality	n
Comments	

3.6 Relationship UserGivesFeedback

3.6.1 UserGivesFeedback Diagram

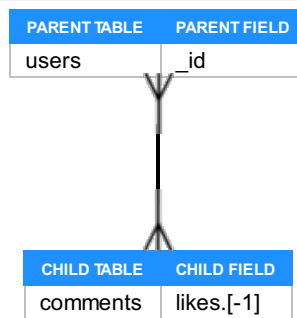


3.6.2 UserGivesFeedback Properties

PROPERTY	VALUE
Name	UserGivesFeedback
Description	
Parent Collection	users
Parent field	_id
Parent Cardinality	1
Child Collection	feedbacks
Child field	userId
Child Cardinality	1
Comments	

3.7 Relationship UserLikesComment

3.7.1 UserLikesComment Diagram

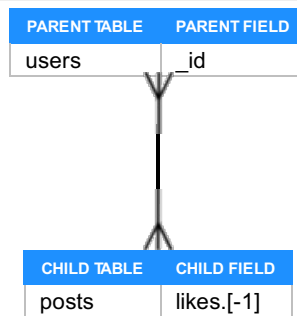


3.7.2 UserLikesComment Properties

PROPERTY	VALUE
Name	UserLikesComment
Description	Relation to represent the User's who liked a Comment. Even though this is a N-N relation, we opt to store the ids only at the Comment side as an array of _id (with the _id of the users who liked it).
Parent Collection	users
Parent field	_id
Parent Cardinality	n
Child Collection	comments
Child field	
Child Cardinality	n
Comments	

3.8 Relationship UserLikesPost

3.8.1 UserLikesPost Diagram

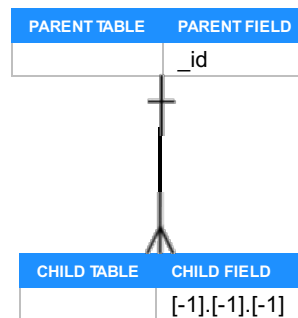


3.8.2 UserLikesPost Properties

PROPERTY	VALUE
Name	UserLikesPost
Description	Relation to represent the User's who liked a Post. Even though this is a N-N relation, we opt to store the ids only at the Post side as an array of _id (with the _id of the users who liked it).
Parent Collection	users
Parent field	_id
Parent Cardinality	n
Child Collection	posts
Child field	
Child Cardinality	n
Comments	

3.9 Relationship UserOwnsOrganization

3.9.1 UserOwnsOrganization Diagram



3.9.2 UserOwnsOrganization Properties

PROPERTY	VALUE
Name	UserOwnsOrganization
Description	Relation between an User and an Organization created by him/her. Initially projected as 1-N, with only one User owning the Organization, it is though to later on be evolved into N-N, with many Users being able to manage the Organization with different roles.
Parent Collection	
Parent field	_id
Parent Cardinality	1
Child Collection	
Child field	
Child Cardinality	n
Comments	