Experimental / Images

API Version: 1.0.0

Organize and search the images of the construction site.

INDEX

API

1. IMAGES

1.1 GET /images/within_bounding_box

Get Images Within Bounding Box

Retrieve images captured within the given bounding box.

REQUEST

REQUEST BODY - application/json

{

Represent a query to the image database based on a bounding box.

```
interval_start*  integer    Time stamp when the first image was captured (inclusive, seconds since epoch in UTC)
interval_end*    integer    Time stamp after the last image was captured (exclusive, seconds since epoch in UTC)
source_urls*     [string]
bounding_box {
```

Represent a bounding box of an image query.

```
  xmin*  number  inclusive
  xmax*  number  exclusive
  ymin*  number  inclusive
  ymax*  number  exclusive
  zmin*  number  inclusive
  zmax*  number  exclusive
}
```

}

RESPONSE

STATUS CODE - 200: Successful Response

RESPONSE MODEL - application/json

[{

Array of object: Represent a retrievable image from our system.

```
timestamp*  integer  Timestamp when the image was taken (seconds since epoch in UTC)
location* {
```

Represent space coordinates where the image was taken.

```
  x*  number
  y*  number
  z*  number
}
angle_of_view* {
```

Specify the diagonal angle of view of an image.

TODO (mristin, 2021-04-16): We need yet to specify this.

```
}
direction* {
```

Represent the compass direction of the camera when the image was taken.

TODO (mristin, 2021-04-16): We need yet to specify this.

```
}
url*        string
```

}]

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
    Array of object:
      loc*  [string]
      msg*  string
      type* string
  }]
}
```

1.2 GET /images/of_elements

Get Images Of Elements

Retrieve images capturing the elements in their field-of-view.

REQUEST

REQUEST BODY - application/json

```
{
Represent a query to the image database based on building elements.
  interval_start* integer   Time stamp when the first image was captured (inclusive, seconds since epoch in UTC)
  interval_end*   integer   Time stamp after the last image was captured (exclusive, seconds since epoch in UTC)
  source_urls*    [string]
  elements* [{
    Array of object:  Specify the relevant building element for a query.
    guid*         string  PATTERN:^[a-zA-Z0-9_.:-]+$
                          Generally unique identifier of the element
    revision_url* string  URL to the revision of the building plan (*e.g.*, in BIM Sync)
    url*          string  URL to the description of the element (*e.g.*, in BIM Sync)
  }]
}
```

RESPONSE

STATUS CODE - 200: Successful Response

RESPONSE MODEL - application/json

```
[{
Array of object:  Represent a retrievable image from our system.
  timestamp* integer  Timestamp when the image was taken (seconds since epoch in UTC)
  location* {
  Represent space coordinates where the image was taken.
    x* number
    y* number
    z* number
  }
  angle_of_view* {
  Specify the diagonal angle of view of an image.

  TODO (mristin, 2021-04-16): We need yet to specify this.

  }
  direction* {
  Represent the compass direction of the camera when the image was taken.

  TODO (mristin, 2021-04-16): We need yet to specify this.

  }
  url*       string
}]
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
  Array of object:
    loc*  [string]
    msg*  string
    type* string
  }]
}
```

1.3 POST /images

Post Image

Upload an image together with its meta-data.

Return URL of the image.

REQUEST

FORM DATA PARAMETERS

NAME	TYPE	DESCRIPTION
file	string(binary)	
image	string	JSON-encoded meta data in form of Image structure.

RESPONSE

STATUS CODE - 200: Successful Response

RESPONSE MODEL - application/json

```
undefined
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
  Array of object:
    loc*  [string]
    msg*  string
    type* string
  }]
}
```

1.4 GET /images/{image_path}

Get Image

Check if the image exists.

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*image_path	string	

RESPONSE

STATUS CODE - 200: Successful Response

RESPONSE MODEL - application/json

undefined

RESPONSE MODEL - image/png

STATUS CODE - 404: Image not available

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
  Array of object:
    loc*  [string]
    msg*  string
    type* string
  }]
}
```

1.5 HEAD /images/{image_path}

Image Exists

Check if the image exists.

REQUEST

QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
*path	string	

RESPONSE

STATUS CODE - 200: Image available

RESPONSE MODEL - application/json

undefined

STATUS CODE - 404: Image not available

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
  detail [{
  Array of object:
    loc*  [string]
    msg*  string
    type* string
```

```
    }]
}
```

2. SOURCES

2.1 GET /sources

Get Sources

List all sources of all the images available in the database.

REQUEST

No request parameters

RESPONSE

STATUS CODE - 200: Successful Response

RESPONSE MODEL - application/json

```
[string]
```
