Experimental / Localization

API Version: 1.0.0

Provide spatio-temporal localization of the instances.

INDEX

API

1. COHORTS

1.1 GET /cohorts

List Cohorts

List the GUIDs of the available cohorts.

REQUEST

No request parameters

RESPONSE

STATUS CODE - 200: Successful Response

RESPONSE MODEL - application/json

```
[string]
```

1.2 PUT /cohorts

Put Cohort

Put a cohort based on its GUID.

REQUEST

REQUEST BODY - application/json
```
{
```
Represent a cohort of BIM instances.

url*	string	URL to the cohort data (e.g., in BIM Sync or BIM Extended)
guid*	string	**PATTERN:**`^[a-zA-Z0-9_.:-]+$`
		generally unique identifier of the cohort
instances*	[string]	

```
}
```

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
   }]
}
```

1.3 GET /cohorts/{guid}

Get Cohort
Retrieve the cohort by the guid.

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*guid	string PATTERN: ^[a-zA-Z0-9_.:-]+$	GUID of the cohort

RESPONSE

STATUS CODE - 200: Successful Response

RESPONSE MODEL - application/json

```
{
Represent a cohort of BIM instances.
   url*        string      URL to the cohort data (e.g., in BIM Sync or BIM Extended)
   guid*       string      PATTERN:^[a-zA-Z0-9_.:-]+$
                           generally unique identifier of the cohort
   instances*  [string]
}
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

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

1.4 DELETE /cohorts/{guid}

Delete Cohort
Delete the indicated cohort, or do nothing if it does not exist.

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*guid	string PATTERN: ^[a-zA-Z0-9_.:-]+$	GUID of the cohort

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
  }]
}
```

1.5 GET /localization/concrete/cohort/{guid}

Concrete Observations Of Cohort

Get all the location observations recorded for the cohort.

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*guid	string PATTERN: ^[a-zA-Z0-9_.:-]+$	GUID of the cohort

RESPONSE

STATUS CODE - 200: Successful Response

RESPONSE MODEL - application/json

```
[{
Array of object:  Observe something relative to the reference point of the site.
  coordinates* {
  Concrete coordinates
    x*  number
    y*  number
    z*  number
  }
  timestamp*  integer   Time stamp when the location has been recorded (milliseconds since epoch in UTC)
}]
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

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

2. INSTANCES

2.1 GET /instances

List Instances

List the identifiers of the available instances.

REQUEST

No request parameters

RESPONSE

STATUS CODE - 200: Successful Response

RESPONSE MODEL - application/json

```
[string]
PATTERN:^[a-zA-Z0-9_.:-]+$
```

2.2 PUT /instances

Put Instance

Put an instance based on its GUID.

REQUEST

REQUEST BODY - application/json

```
{
Represent a track of a BIM instance.
  guid*       string      PATTERN:^[a-zA-Z0-9_.:-]+$
                          Generally unique identifier of the instance
  cohorts*   [string]
  url*        string      URL to the instance data (e.g., in BIM Sync or BIM Extended)
}
```

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
  }]
}
```

2.3 GET /instances/{guid}

Get Instance

Retrieve the instance by its GUID.

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*guid	string PATTERN: ^[a-zA-Z0-9_.:-]+$	GUID of the instance

RESPONSE

STATUS CODE - 200: Successful Response

RESPONSE MODEL - application/json

```
{
```
Represent a track of a BIM instance.
```
    guid*      string      PATTERN:^[a-zA-Z0-9_.:-]+$
```
Generally unique identifier of the instance
```
    cohorts*  [string]
    url*       string      URL to the instance data (e.g., in BIM Sync or BIM Extended)
}
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

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

2.4 DELETE /instances/{guid}

Delete Instance

Delete the indicated instance, or do nothing if it does not exist.

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*guid	string PATTERN: ^[a-zA-Z0-9_.:-]+$	GUID of the instance

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
  }]
}
```

2.5 GET /localization/concrete/instance/{guid}

Concrete Observations Of Instance

Get all the location observations recorded for the instance.

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*guid	string PATTERN: ^[a-zA-Z0-9_.:-]+$	GUID of the instance

RESPONSE

STATUS CODE - 200: Successful Response

RESPONSE MODEL - application/json

```
[{
Array of object:  Observe something relative to the reference point of the site.
  coordinates* {
  Concrete coordinates
    x*  number
    y*  number
    z*  number
  }
  timestamp*  integer   Time stamp when the location has been recorded (milliseconds since epoch in UTC)
}]
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

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

2.6 POST /localization/concrete/instance/{guid}

Post Concrete Observation Of Instance

Post a recorded location of an instance.

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*guid	string PATTERN: ^[a-zA-Z0-9_.:-]+$	GUID of the instance

REQUEST BODY - application/json
```
{
```
Observe something relative to the reference point of the site.
```
  coordinates* {
```
Concrete coordinates
```
    x*  number
    y*  number
    z*  number
  }
  timestamp*  integer   Time stamp when the location has been recorded (milliseconds since epoch in UTC)
}
```

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
  }]
}
```

2.7 POST /localization/logical/instance/{guid}

Post Logical Observation Of Instance

Post a recorded logical location of an instance.

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*guid	string PATTERN: ^[a-zA-Z0-9_.:-]+$	GUID of the instance

REQUEST BODY - application/json

{

Represent an observation of a logical location for an instance.

    guid*   string   **PATTERN**:^[a-zA-Z0-9_.:-]+$

                       Globally unique identifier of the location

    timestamp* integer   Time stamp when the location has been recorded (milliseconds since epoch in UTC)

}

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

  }]

}

3. LOGICAL

3.1 GET /logical_locations/{guid}

Get Logical Location

Retrieve the logical location by its GUID.

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*guid	string PATTERN: ^[a-zA-Z0-9_.:-]+$	GUID of the logical location

RESPONSE

STATUS CODE - 200: Successful Response

RESPONSE MODEL - application/json

```
{
```
Represent a track of a BIM instance.
```
  guid*      string    PATTERN:^[a-zA-Z0-9_.:-]+$
```
Generally unique identifier of the instance
```
  cohorts*   [string]
  url*       string    URL to the instance data (e.g., in BIM Sync or BIM Extended)
}
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

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

3.2 GET /logical_locations

List Logical Locations

List the available logical locations.

REQUEST

No request parameters

RESPONSE

STATUS CODE - 200: Successful Response

RESPONSE MODEL - application/json

```
[{
```

Array of object: Represent a logical location.

This can be arbitrarily abstract and the location may not even exist
in the physical world with a concrete spatial zone.

```
   guid*  string  PATTERN:^[a-zA-Z0-9_.:-]+$
                          Generally unique identifier of the location
   url*   string  URL to the location, e.g., in BIM Sync or BIM Extended
}]
```

3.3 PUT /logical_locations

Put Logical Location

Put a logical location based on its GUID.

REQUEST

REQUEST BODY - application/json
```
{
```
Represent a logical location.

This can be arbitrarily abstract and the location may not even exist
in the physical world with a concrete spatial zone.

```
   guid*  string  PATTERN:^[a-zA-Z0-9_.:-]+$
                          Generally unique identifier of the location
   url*   string  URL to the location, e.g., in BIM Sync or BIM Extended
}
```

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
   }]
}
```

3.4 GET /localization/logical/instance/{guid}

Logical Observations Of Instance

Get all the logical locations recorded for the instance.

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*guid	string PATTERN: ^[a-zA-Z0-9_.:-]+$	GUID of the instance

RESPONSE

STATUS CODE - 200: Successful Response

RESPONSE MODEL - application/json

```
[{
```
Array of object: Represent an observation of a logical location for an instance.

```
   guid*         string    PATTERN:^[a-zA-Z0-9_.:-]+$
```
Globally unique identifier of the location

```
   timestamp*  integer   Time stamp when the location has been recorded (milliseconds since epoch in UTC)
}]
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

```
{
   detail [{
```
Array of object:

```
      loc*   [string]
      msg*   string
      type*  string
   }]
}
```

3.5 POST /localization/logical/instance/{guid}

Post Logical Observation Of Instance

Post a recorded logical location of an instance.

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*guid	string PATTERN: ^[a-zA-Z0-9_.:-]+$	GUID of the instance

REQUEST BODY - application/json

```
{
```
Represent an observation of a logical location for an instance.

```
   guid*         string    PATTERN:^[a-zA-Z0-9_.:-]+$
```
Globally unique identifier of the location

```
   timestamp*  integer   Time stamp when the location has been recorded (milliseconds since epoch in UTC)
}
```

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
  }]
}
```

3.6 GET /localization/logical/cohort/{guid}

Logical Observations Of Cohort

Get all the logical locations recorded for the cohort.

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*guid	string PATTERN: ^[a-zA-Z0-9_.:-]+$	GUID of the cohort

RESPONSE

STATUS CODE - 200: Successful Response

RESPONSE MODEL - application/json

```
[{
Array of object:  Represent an observation of a logical location for an instance.

  guid*        string    PATTERN:^[a-zA-Z0-9_.:-]+$
                         Globally unique identifier of the location
  timestamp*   integer   Time stamp when the location has been recorded (milliseconds since epoch in UTC)
}]
```

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

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