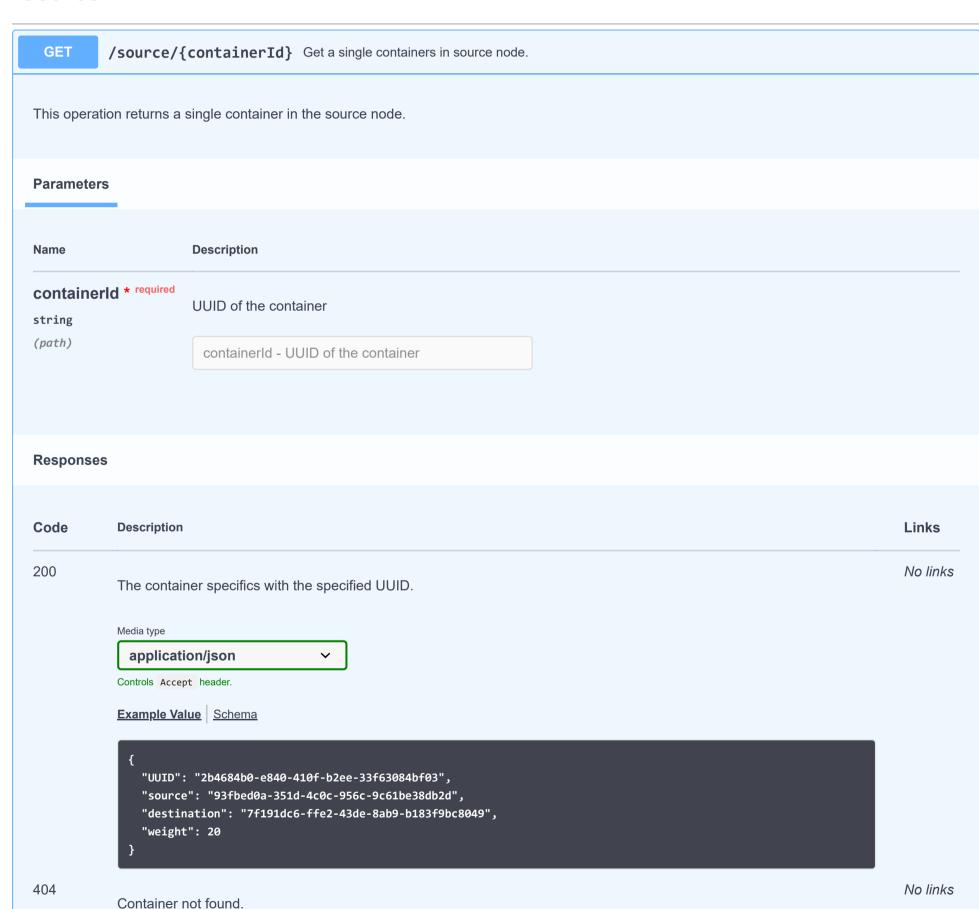📄 **SwaggerStation.yaml** 9.02 KB
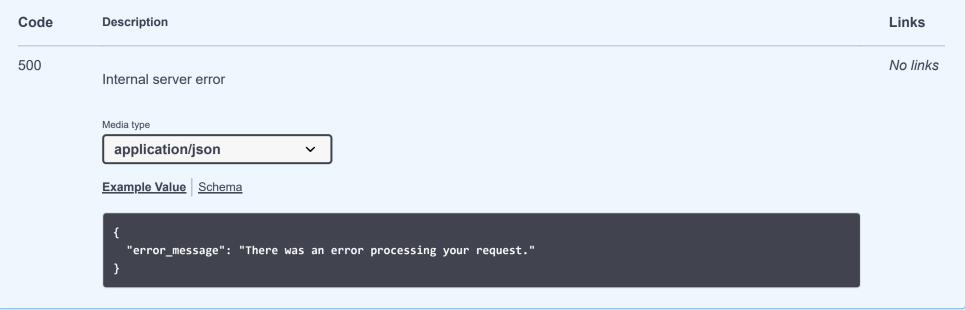
Source/Destination `1.0.0` `OAS3`
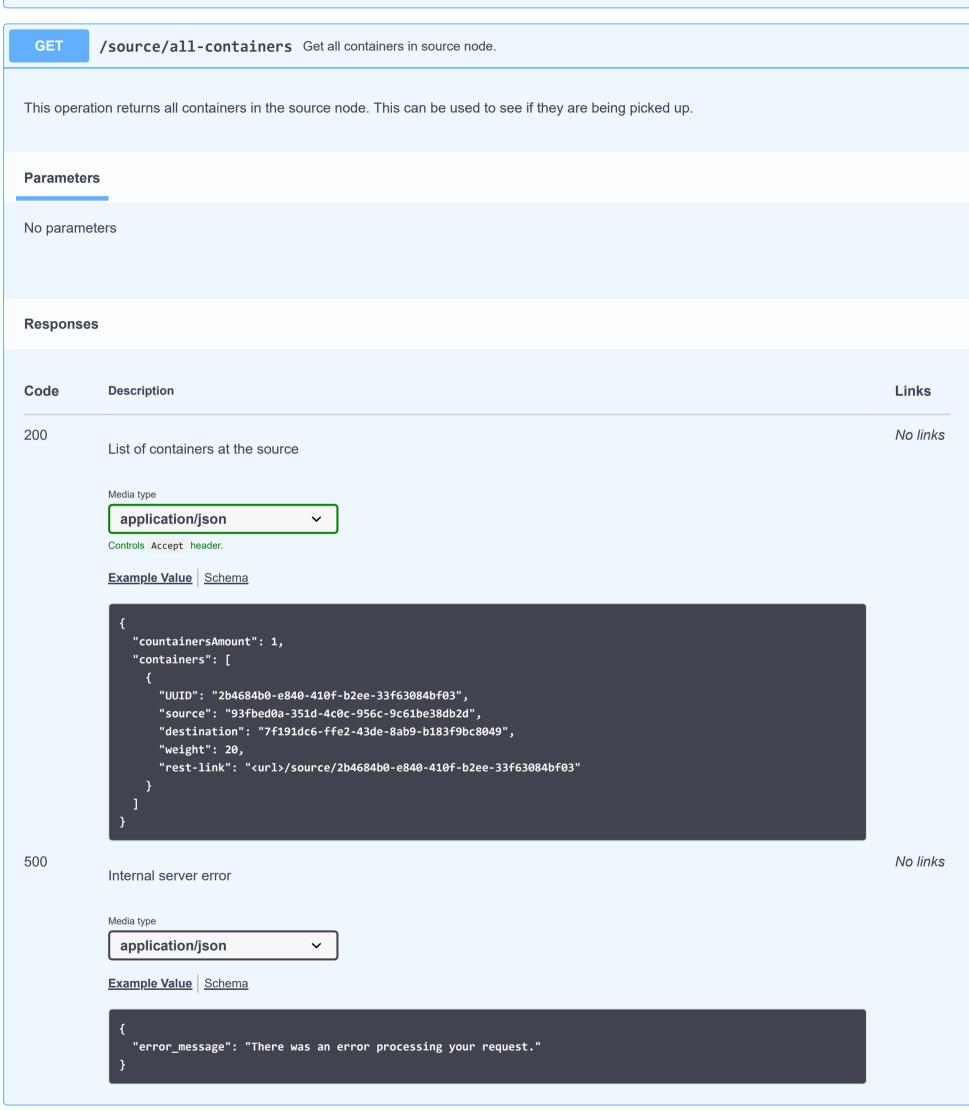
/vu-distributed-systems-engineering/students/2020w/DSE_Team_104/-/raw/master/OpenAPI%20Specification/SwaggerStation.yaml

This is a service to perform various actions on a single Source/Destination node.

source Operations on containers inserted into the Source (to be sent away) ⌄

GET	**/source/{containerId}** Get a single containers in source node.

This operation returns a single container in the source node.

Parameters

Name	Description
containerId * required **string** *(path)*	UUID of the container `containerId - UUID of the container`

Responses

Code	Description	Links
200	The container specifics with the specified UUID. Media type `application/json` ⌄ Controls `Accept` header. **Example Value** | Schema	*No links*

```
{
    "UUID": "2b4684b0-e840-410f-b2ee-33f63084bf03",
    "source": "93fbed0a-351d-4c0c-956c-9c61be38db2d",
    "destination": "7f191dc6-ffe2-43de-8ab9-b183f9bc8049",
    "weight": 20
}
```

| 404 | Container not found. | *No links* |

Code	Description	Links
500	Internal server error	*No links*

Media type

application/json ▼

Example Value | Schema

```
{
    "error_message": "There was an error processing your request."
}
```

GET `/source/all-containers` Get all containers in source node.

This operation returns all containers in the source node. This can be used to see if they are being picked up.

Parameters

No parameters

Responses

Code	Description	Links
200	List of containers at the source	*No links*

Media type

application/json ▼

Controls `Accept` header.

Example Value | Schema

```
{
    "countainersAmount": 1,
    "containers": [
      {
        "UUID": "2b4684b0-e840-410f-b2ee-33f63084bf03",
        "source": "93fbed0a-351d-4c0c-956c-9c61be38db2d",
        "destination": "7f191dc6-ffe2-43de-8ab9-b183f9bc8049",
        "weight": 20,
        "rest-link": "<url>/source/2b4684b0-e840-410f-b2ee-33f63084bf03"
      }
    ]
}
```

Code	Description	Links
500	Internal server error	*No links*

Media type

application/json ▼

Example Value | Schema

```
{
    "error_message": "There was an error processing your request."
}
```
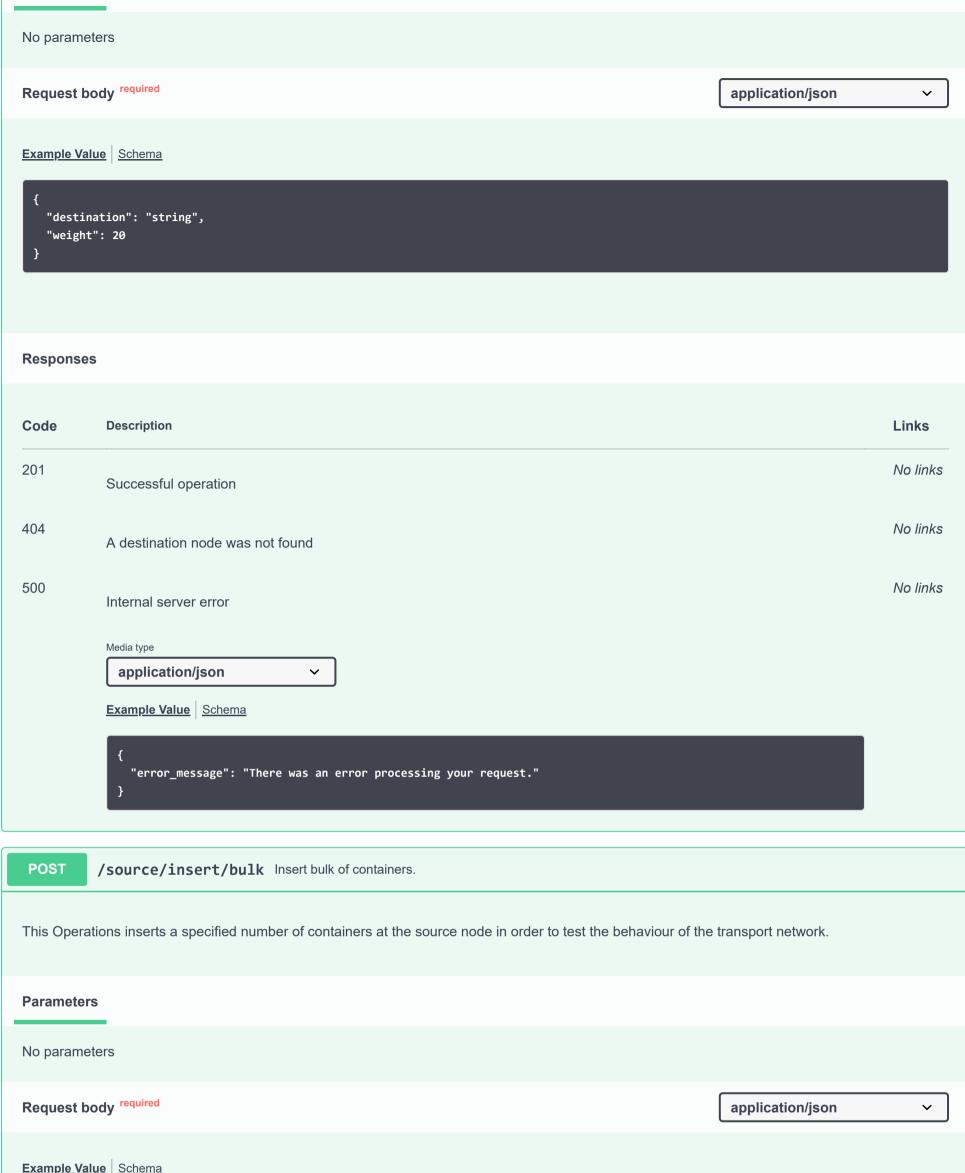
POST `/source/insert` Insert single Container into the Source.

This operation inserts a single container into the Source. It will then be transported through the transport network to the specified destination.
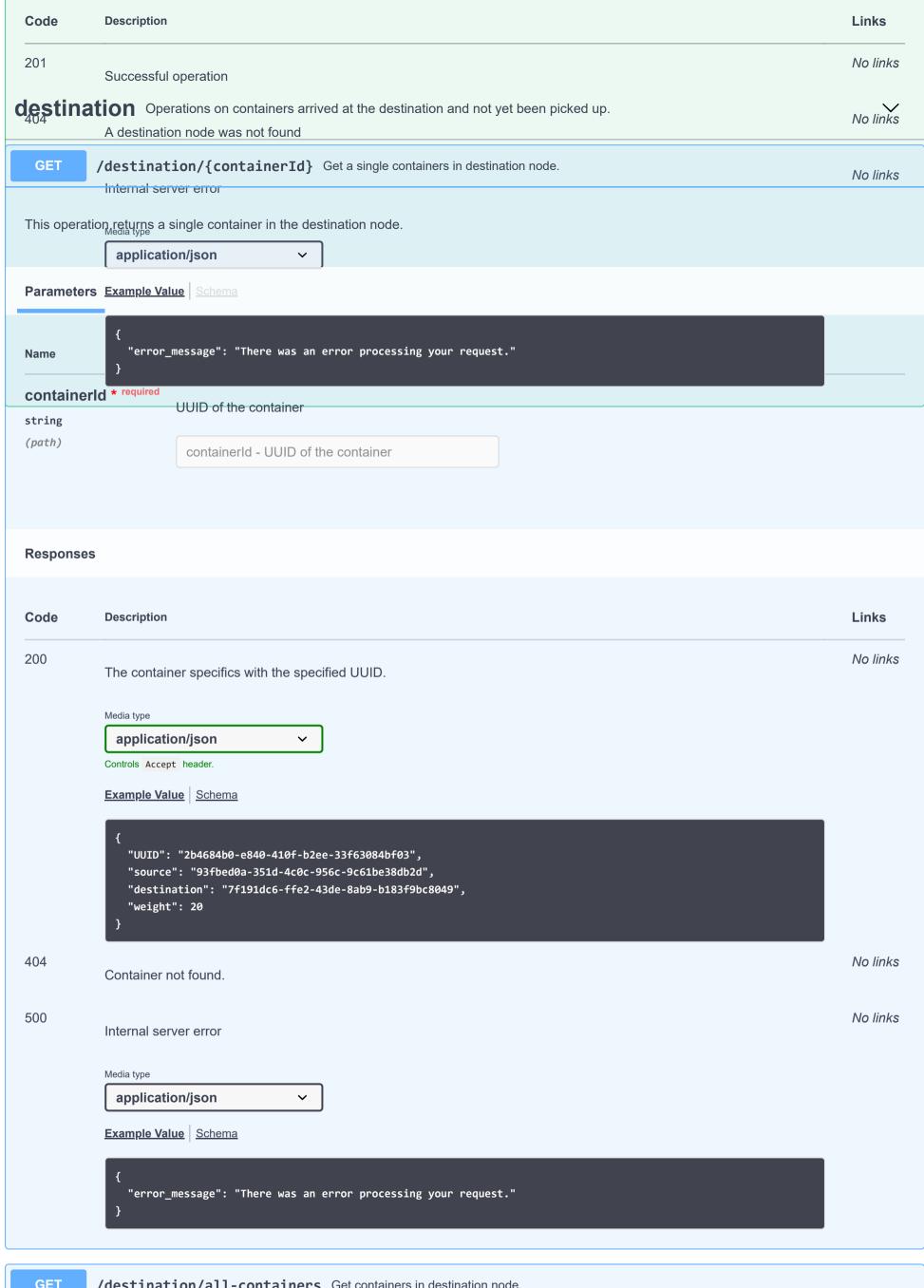
Parameters

No parameters

Request body required

application/json ▼

Example Value | Schema

```
{
  "destination": "string",
  "weight": 20
}
```

Responses

Code	Description	Links
201	Successful operation	No links
404	A destination node was not found	No links
500	Internal server error	No links

Media type

application/json ▼

Example Value | Schema

```
{
  "error_message": "There was an error processing your request."
}
```

POST **/source/insert/bulk** Insert bulk of containers.

This Operations inserts a specified number of containers at the source node in order to test the behaviour of the transport network.

Parameters

No parameters

Request body required

application/json ▼

Example Value | Schema

```
{
  "containers": [
    {
      "destination": "string",
      "weight": 20
    }
  ]
}
```

Responses

Code	Description	Links

Code	Description	Links
201	Successful operation	No links

destination   Operations on containers arrived at the destination and not yet been picked up.   ⌄

| 404 | | No links |

A destination node was not found

GET `/destination/{containerId}`   Get a single containers in destination node.   No links

Internal server error

This operation returns a single container in the destination node.

Media type

application/json ⌄

Parameters   **Example Value** | Schema

Name	

```
{
   "error_message": "There was an error processing your request."
}
```

containerId * required
string
(path)

UUID of the container

containerId - UUID of the container

Responses

Code	Description	Links
200	The container specifics with the specified UUID.	No links

Media type

application/json ⌄

Controls `Accept` header.

Example Value | Schema

```
{
   "UUID": "2b4684b0-e840-410f-b2ee-33f63084bf03",
   "source": "93fbed0a-351d-4c0c-956c-9c61be38db2d",
   "destination": "7f191dc6-ffe2-43de-8ab9-b183f9bc8049",
   "weight": 20
}
```

| 404 | Container not found. | No links |
| 500 | Internal server error | No links |

Media type

application/json ⌄

Example Value | Schema

```
{
   "error_message": "There was an error processing your request."
}
```

GET `/destination/all-containers`   Get containers in destination node

Get all containers which have been received at the destination node and are ready to be picked up (deleted).

Parameters

No parameters

Responses

Code	Description	Links
200	All containers are returned	*No links*

Media type

<div>application/json ▼</div>

Controls `Accept` header.

Example Value | Schema

```
{
  "destination": "string",
  "containers": [
    {
      "UUID": "2b4684b0-e840-410f-b2ee-33f63084bf03",
      "source": "93fbed0a-351d-4c0c-956c-9c61be38db2d",
      "destination": "7f191dc6-ffe2-43de-8ab9-b183f9bc8049",
      "weight": 20,
      "rest-link": "<url>/destination/2b4684b0-e840-410f-b2ee-33f63084bf03"
    }
  ]
}
```

Code	Description	Links
500	Internal server error	*No links*

Media type

<div>application/json ▼</div>

Example Value | Schema

```
{
  "error_message": "There was an error processing your request."
}
```

DELETE `/destination/pickup/{containerId}` Delete a container

Pickes up (deletes) a container from the destination node and returns the container properties. Container must be in the destination. Note, source and destination are logically seperated in this regard.

Parameters

Name	Description
containerId * required **string** *(path)*	UUID of the container `containerId - UUID of the container`

Responses

Code	Description	Links
200		*No links*
404	Container could not be found	*No links*

Code	Description	Links
500	Internal Server Error	*No links*

Media type

application/json ⌄

Example Value | Schema

```json
{
  "error_message": "There was an error processing your request."
}
```

Schemas ⌄

Container ›

ContainerInsertion ›

ErrorMessage ›