Reservation Portal
Interface description

Contents

Service description

This document describes the web service interface of the Reservation Portal (Reservation Service), which is responsible for the administrative data management of resource-reservations.

General business terminology and rules:

- A "**resource**" is a general thing (for example: hotel room, equipment, a person, etc.)
- A resource is identified by a number (**resource-identifier**).
- A resource can be reserved for a **period of days** (e.g. from 2013-10-13 to 2013-11-05)
- The entity requesting the reservation becomes the **owner** of the reservation.
- The users of the service are able to place reservation for a resource.
- Each confirmed reservation is identified by a number (**reservation-identifier**)
- When a resource is reserved by someone, no conflicting reservations can be made for it within an overlapping time period. Such requests are rejected by the service.
- Reservations may be deleted by its reservation-id
- A reservation's details can be fetched by its reservation-id
- The service can tell, for a time period which reservations are present in the system.

The API (Application Program Interface) of the service

The users of the service (for example external clients, or Javascript client) communicate with the service via HTTP messages.
The following sections describe the format of the requests and responses.

List all reservations

Request	GET http://localhost/api/reservations/
	Optional query Parameters:
	• page: the index of the requested page. The first page is addressed with 0.
	• size: the number of records per page
	Example:
	http://localhost/api/reservations/?page=0&size=10
Response	
Status	HTTP-200 success
Body	Example body content:

```
{
    "content": [
        {
            "id": 1,
            "resourceId": 1,
            "startDate": "2014-11-01",
            "endDate": "2014-11-01",
            "owner": "Joe"
        },
        {
            "id": 2,
            "resourceId": 2,
            "startDate": "2014-11-01",
            "endDate": "2014-11-01",
            "owner": "Mary"
        }
    ],
    "last": true,
    "totalElements": 2,
    "totalPages": 1,
    "size": 10,
    "number": 0,
    "sort": null,
    "numberOfElements": 2,
    "first": true
}
```

List all reservations for a period

Request	GET http://localhost/api/reservations/from-{startDate}/to-{endDate}/
	Path variables: • startDate: the start date of the queried period. Format: YYYY-MM-DD • endDate: the end date of the queried period. Format: YYYY-MM-DD Optional query parameters: • page: the index of the requested page. The first page is noted with 0. • size: the number of records per page Example: http://localhost/api/reservations/from-2015-01-21/to-2015-12-10/
Response	
Status	HTTP-200 success HTTP-400 format of the input parameters are not valid
| Body | Example body content:<br><br>```json
{
    "content": [
        {
            "id": 2,
            "resourceId": 2,
            "startDate": "2014-11-01",
            "endDate": "2014-11-01",
            "owner": "Mary"
        }
    ],
    "totalPages": 1,
    "totalElements": 1,
    "last": true,
    "size": 20,
    "number": 0,
    "sort": null,
    "first": true,
    "numberOfElements": 1
}
``` |

Place reservation

Request	POST http://localhost:8080/reservation-service/api/reservation/{resourceId}/from-{startDate}/to-{endDate}/?owner={reservation-owner} Path variables: • resourceId: id of the resource (long) • startDate: the start date of the reservation. Format: YYYY-MM-DD • endDate: the end date of the reservation. Format: YYYY-MM-DD Query parameters: • owner: specifies the owning entity of the reservation (for example the name of the customer)
Response	
Status	HTTP-201 success (reservation confirmed) HTTP-400 format of the input parameters are not valid HTTP-409 resource is already reserved for the specified period
Header	If the reservation was successful, the following header field is present in the response: • reservation-id: <the created reservation's id>

Body	Empty – for successful reservation
	
	Error reason for failed reservation, example:
	
	{
	   "timestamp": 1497969074616,
	   "status": 409,
	   "error": "Conflict",
	   "exception": "org.training.reservations.service.ReservationException",
	   "message": "Reservation failed: resource already reserved for the specified period",
	   "path": "/api/reservations/2/from-2015-01-01/to-2015-01-01/"
	}

Get reservation details

Request	GET http://localhost/api/reservations/{reservationId}/ Path variables: • reservationId: id of the reservation (long)
Response	
Status	HTTP-200 success HTTP-404 reservation does not exist
Header	n.a.

Body	Example body content:
	{
	   "id": 2,
	   "resourceId": 2,
	   "startDate": "2014-11-01",
	   "endDate": "2014-11-01",
	   "owner": "Mary"
	}

Delete a reservation

Request	DELETE http://localhost/api/reservations/{reservationId}/ Path variables: • reservationId: id of the reservation (long)	
Response		
Status	HTTP-202 success HTTP-404 reservation does not exist	

Delete all reservations (for testing purposes only)

Request	DELETE http://localhost/api/reservations/
Response	
Status	HTTP-204 success