Contents

[Purpose of this document 1](#_Toc119178690)

[REST resources (services) 1](#_Toc119178691)

[REST requests and responses 3](#_Toc119178692)

[1 Get all clothes 3](#_Toc119178693)

[2 Insert new clothing item 3](#_Toc119178694)

[3 Get single clothing item 5](#_Toc119178695)

[4 Edit clothing item 5](#_Toc119178696)

[5 Delete clothing item 5](#_Toc119178697)

[6 Get all producers 6](#_Toc119178698)

[7 Insert new producer 6](#_Toc119178699)

[8 Get one producer 7](#_Toc119178700)

[9 Edit producer 7](#_Toc119178701)

[10 Delete producer 7](#_Toc119178702)

[HTTP methods 8](#_Toc119178703)

# Purpose of this document

This document describes what REST services DogClothing-application provides. It contains all needed information so that client can be implemented.

# REST resources (services)

1. Get all clothes -> http://localhost:8080/api/clothes/ (use plurals)

2. Insert new clothing item-> http://localhost:8080/api/clothes/

3. Get one of the clothes -> http://localhost:8080/api/clothes/{id}

4. Edit clothing item -> http://localhost:8080/api/clothes/{id} (id parameter needed)

5. Delete clothing item -> http://localhost:8080/api/clothes/{id}

6. Get all producers -> http://localhost:8080/api/producers/ (use plurals)

7. Insert new producer -> http://localhost:8080/api/producers/

8. Get one producer -> http://localhost:8080/api/producers/{id}

9. Edit producer -> http://localhost:8080/api/producers/{id} (id parameter needed)

10. Delete producer -> http://localhost:8080/api/producers/{id}

# REST requests and responses

## 1 Get all clothes

HTTP method: GET

API call: <http://localhost:8080/api/clothes/>

Request body: -

Response example below:

[

    {

        "id": 3,

        "name": "JoustavaMeno",

        "type": "Haalari",

        "price": 59.0,

        "producer": {

            "producerid": 1,

            "name": "M&M"

        }

    },

    {

        "id": 4,

        "name": "70-luku",

        "type": "Haalari",

        "price": 32.0,

        "producer": {

            "producerid": 2,

            "name": "Leikki"

        }

    }

]

## 2 Insert new clothing item

HTTP method: POST

API call: <http://localhost:8080/api/clothes/>

Request body:

{

    "name": "Uusi Vaate",

    "type": "Sukat",

    "price": 19.0,

    "producer": {

        "producerid": "1",

        "name": "M&M"

    }

}

Response example below (Status 200 OK):

{

    "id": 9,

    "name": "Uusi Vaate",

    "type": "Sukat",

    "price": 19.0,

    "producer": {

        "producerid": 1,

        "name": "M&M"

    }

}

## 3 Get single clothing item

HTTP method: GET

API call: <http://localhost:8080/api/clothes/3>

Request body: -

Response example below (Status 200 OK):

{

    "id": 3,

    "name": "JoustavaMeno",

    "type": "Haalari",

    "price": 59.0,

    "producer": {

        "producerid": 1,

        "name": "M&M"

    }

}

## 4 Edit clothing item

HTTP method: PUT

API call: <http://localhost:8080/api/clothes/4>

Request body:

{

    "name": "70-luku edited",

    "type": "Haalari",

    "price": 32.0,

    "producer": {

        "producerid": 2,

        "name": "Leikki"

    }

}

Response example below (Status 200 OK):

{

    "id": 4,

    "name": "70-luku edited",

    "type": "Haalari",

    "price": 32.0,

    "producer": {

        "producerid": 2,

        "name": "Leikki"

    }

}

## 5 Delete clothing item

HTTP method: DELETE

API call: <http://localhost:8080/api/clothes/4>

Request body: -

Response: Status 200 OK

## 6 Get all producers

HTTP method: GET

API call: <http://localhost:8080/api/producers/>

Request body: -

Response example below:

[

    {

        "producerid": 1,

        "name": "M&M"

    },

    {

        "producerid": 2,

        "name": "Leikki"

    }

]

## 7 Insert new producer

HTTP method: POST

API call: <http://localhost:8080/api/producers/>

Request body:

{

    "name": "Valmistaja"

}

Response example below (Status 200 OK):

{

    "producerid": 5,

    "name": "Valmistaja"

}

## 8 Get one producer

HTTP method: GET

API call: <http://localhost:8080/api/producers/5>

Request body: -

Response example below (Status 200 OK):

{

    "producerid": 5,

    "name": "Valmistaja"

}

## 9 Edit producer

HTTP method: PUT

API call: <http://localhost:8080/api/producers/5>

Request body:

{

    "name": "ValmistajaEdited"

}

Response example below (Status 200 OK):

{

    "producerid": 5,

    "name": "ValmistajaEdited"

}

## 10 Delete producer

HTTP method: DELETE

API call: <http://localhost:8080/api/producers/5>

Request body: -

Response: Status 200 OK

# HTTP methods

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Http**  **Method** | **Resource EndPoint** | **Input** | **Success Response** | **Error**  **Response** | **Description** |
| GET | /api/clothes | Body: empty | Status: 200  List of clothes | Status 500 | Retrieves all clothes |
| POST | /api/clothes | Body: new clothing data | Status: 200  New clothing with id information | Status: 500 | Creates new clothing item |
| PUT | /api/clothes | - | - | Status: 405 | Method not allowed |
| DELETE | /api/clothes | - | - | Status: 405 | Method not allowed |
| GET | api/clothes/{id} | Body: empty | Status: 200  Information of one clothing item | Status 500 | Get information of one clothing item. |
| POST | api/clothes/{id} | - | - | Status: 405 | Method not allowed |
| PUT | api/clothes/{id} | Body: clothing data with updates | Status: 200  Body: clothing with edited information | Status:404 or 500 | Updates an existing clothing item |
| DELETE | api/clothes/{id} | Body: empty | Status: 200  Body: Empty | Status:404 or 500 | Deletes an existing clothing item |
| GET | /api/producers | Body: empty | Status: 200  List of producers | Status 500 | producers |
| POST | /api/producers | Body: new producer data | Status: 200  New producer with id information | Status: 500 | Creates new producer |
| PUT | /api/producers | - | - | Status: 405 | Method not allowed |
| DELETE | /api/producers | - | - | Status: 405 | Method not allowed |
| GET | api/producers/{id} | Body: empty | Status: 200  Information of one producer | Status 500 | Get information of one producer |
| POST | api/producers/{id} | - | - | Status: 405 | Method not allowed |
| PUT | api/producers/{id} | Body: producer data with updates | Status: 200  Body: producer with edited information | Status:404 or 500 | Updates an existing clothing item |
| DELETE | api/producers/{id} | Body: empty | Status: 200  Body: Empty | Status:404 or 500 | Deletes an existing producer |