

# TAKE - restaurant backend documentation

Jakub Łyczko, Rafał Jurczyk, Michał Foks, Wojciech Janota

27.06.2022

## Overview of all controllers

### ClientREST

This controller handles all CRUD operations on client objects that represent a single client of the restaurant (endpoint: `take/restaurant/client`):

- adding a new client using POST query
- getting list of clients using GET query
- getting info about single client using GET query and appending `/id` to the end of URL
- updating info about client using PUT query

Example input for POST query to `take/restaurant/client`:

```
{
  "name": "Adam",
  "surname": "Nowak",
  "address": "ul. Wesoła 1, Warszawa",
  "email": "adam.nowak@example.com",
  "clientOrders": null
}
```

Example response to above query:

```
{
  "objectToReturn": {
    "id": 1,
    "name": "Adam",
    "surname": "Nowak",
    "address": "ul. Wesoła 1, Warszawa",
    "email": "adam.nowak@example.com",
    "clientOrders": null
  },
  "responseCode": {
    "entity": null,
    "status": 202,
    "metadata": {},
    "annotations": null,
    "entityClass": null,
    "genericType": null,
    "cookies": {},
    "date": null,
    "lastModified": null,
    "closed": false,
    "statusInfo": "ACCEPTED",
    "mediaType": null,
    "allowedMethods": [],
    "entityTag": null,
    "links": [],
    "stringHeaders": {},
    "length": -1,
  }
}
```

```

        "location": null,
        "language": null,
        "headers": {}
    }
}

```

Example result of GET query to `take/restaurant/client`:

```

[
  {
    "id": 1,
    "name": "Adam",
    "surname": "Nowak",
    "address": "ul. Wesoła 1, Warszawa",
    "email": "adam.nowak@example.com",
    "clientOrders": []
  }
]

```

Example result of GET query to `take/restaurant/client/{id}`:

```

{
  "id": 1,
  "name": "Adam",
  "surname": "Nowak",
  "address": "ul. Wesoła 1, Warszawa",
  "email": "adam.nowak@example.com",
  "clientOrders": []
}

```

Example input for PUT query to `take/restaurant/client`:

```

{
  "id":1,
  "name":"Adam",
  "surname":"Kowalski",
  "address":"ul. Wesoła 1, Warszawa",
  "email":"adam.nowak@example.com",
  "clientOrders":null
}

```

Example result of PUT query to `take/restaurant/client`:

(only 202 HTTP code returned)

Example result of DELETE query to `take/restaurant/client/{id}`:

(deleting client with id=2)

```

{
  "id": 2,
  "name": "Adam",
  "surname": "Słodowy",
  "address": "ul. Wesoła 1, Warszawa",
  "email": "adam.nowak@example.com",
  "clientOrders": []
}

```

## IngredientREST

This REST service handles managing ingredients to the application to be later used in meals.

Example input for POST query to `take/restaurant/ingredient`:

```

{
  "name":"pomidory",
  "quantity":12,

```

```

    "mealIngredients": null
}

```

Example response to above query:

```

{
  "objectToReturn": {
    "id": 3,
    "name": "pomidory",
    "quantity": 12.0
  },
  "responseCode": {
    "entity": null,
    "status": 202,
    "metadata": {},
    "annotations": null,
    "entityClass": null,
    "genericType": null,
    "cookies": {},
    "date": null,
    "lastModified": null,
    "closed": false,
    "statusInfo": "ACCEPTED",
    "mediaType": null,
    "allowedMethods": [],
    "entityTag": null,
    "links": [],
    "stringHeaders": {},
    "length": -1,
    "location": null,
    "language": null,
    "headers": {}
  }
}

```

Example result of GET query to `take/restaurant/ingredient`:

```

[
  {
    "id": 3,
    "name": "pomidory",
    "quantity": 12.0
  },
  {
    "id": 4,
    "name": "mozarella",
    "quantity": 12.0
  }
]

```

Example result of GET query to `take/restaurant/ingredient/{id}` (id=4):

```

{
  "id": 4,
  "name": "mozarella",
  "quantity": 12.0
}

```

Example input for PUT query to `take/restaurant/ingredient`:

```

{
  "id": 4,
  "name": "mozarella",

```

```
    "quantity": 50.0  
  }
```

Example result of PUT query to `take/restaurant/ingredient`:

(only 202 HTTP code returned)

Example result of DELETE query to `take/restaurant/ingredient/{id}`:

(deleting ingredient with `id = 4`)

(only 204 HTTP code)