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 ned 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)

MealREST

This controller takes care of managing meal objects, that represent meals available in a restaurant.

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

```
{
   "name":"pizza",
   "price":26,
   "mealIngredients":null,
   "orders":null
}
```

Example response to above query:

```
{
    "objectToReturn": {
        "id": 5,
        "name": "pizza",
        "price": 26.0,
        "availability": false,
        "mealIngredients": null
    },
    "responseCode": {
        "entity": null,
        "status": 202,
        "metadata": {},
        "annotations": null,
        "entityClass": null,
        "genericType": null,
        "cookies": {},
        "links": [],
        "entityTag": null,
        "allowedMethods": [],
        "mediaType": null,
        "stringHeaders": {},
        "statusInfo": "ACCEPTED",
        "length": -1,
        "location": null,
        "language": null,
        "date": null,
        "lastModified": null,
        "closed": false,
        "headers": {}
    }
}
```

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

```
[
   {
       "id": 5,
       "name": "pizza",
       "price": 60.0,
       "availability": false,
       "mealIngredients": []
   },
   {
       "id": 6,
       "name": "garlic bread",
       "price": 60.0,
       "availability": false,
       "mealIngredients": []
   }
]
```

Example result of GET query to `take/restaurant/meal/{id}` (id=5):

```
{
    "id": 5,
    "name": "pizza",
    "price": 26.0,
    "availability": false,
    "mealIngredients": []
}
```

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

```
{
    "id": 5,
    "name":"pizza",
    "price":60,
    "mealIngredients":null,
    "orders":null
}
```

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

(only 202 HTTP code returned)

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

(deleting meal with id = 6)

(only 204 HTTP code, meal deleted)

MealIngredientREST

This controller takes care of managing intermediate connection (as a additional table) between ingredient objects and meal objects. This way we can also store data about how much of given ingredient is needed for given meal.

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

```
{
    "quantity":3,
    "meal":{"id":5},
    "ingredient":{"id":3}
}
```

Example response to above query:

```
{
     "objectToReturn": {
         "id": 7,
         "quantity": 3,
         "ingredient": {
             "id": 3,
             "name": null,
             "quantity": 0.0
         }
```

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

After that, GET to meals endpoint returns:

```
[
    {
        "id": 5,
        "name": "pizza",
        "price": 60.0,
        "availability": false,
        "mealIngredients": [
            {
                "id": 7,
                "quantity": 3,
                "ingredient": {
                    "id": 3,
                    "name": "ser",
                    "quantity": 12.0
                }
            }
        ]
    }
]
```

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

```
[
   {
       "id": 7,
       "quantity": 3,
       "ingredient": {
           "id": 3,
           "name": "ser",
           "quantity": 12.0
       }
   }
]
```

Example result of GET query to `take/restaurant/mealIngredient/{id}` (id=7):

```
{
   "id": 7,
   "quantity": 3,
   "ingredient": {
       "id": 3,
       "name": "ser",
       "quantity": 12.0
   }
}
```

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

```
{
   "id": 7,
   "quantity":5,
   "meal":{"id":5},
   "ingredient":{"id":3}
}
```

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

(only 202 HTTP code returned)

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

(deleting mealIngredient connection with id = 8)

(only 204 HTTP code, connection removed from meal)

TemporaryOrderREST

This controller manages operation of ordering meals by a client. Name must be different from Order, because it is a keyword in both SQL and JAVA, which caused us many problems during development.

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

```
{
   "orderedAt":"2012-04-23T18:25:43.511Z",
   "totalPrice":"25",
   "meals":[{"id":5}],
   "client":{"id":1}
}
```

Example response to above query:

```
{
    "objectToReturn": {
        "id": 10,
        "orderedAt": 1657357714894,
        "totalPrice": 25.0,
        "meals": [
            {
                "id": 5,
                "name": null,
                "price": 0.0,
                "availability": false,
                "mealIngredients": null
            }
        ],
        "client": {
            "id": 1,
            "name": null,
            "surname": null,
            "address": null,
            "email": null,
            "clientOrders": null
        }
    },
    "responseCode": {
        "entity": null,
        "status": 202,
        "metadata": {},
        "annotations": null,
        "entityClass": null,
        "genericType": null,
        "cookies": {},
        "links": [],
        "entityTag": null,
        "allowedMethods": [],
        "mediaType": null,
        "stringHeaders": {},
        "statusInfo": "ACCEPTED",
        "length": -1,
        "location": null,
        "language": null,
        "date": null,
        "lastModified": null,
        "closed": false,
        "headers": {}
    }
}
```

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

```
[
   {
       "id": 10,
       "orderedAt": 1657357714894,
       "totalPrice": 25.0,
       "meals": [
           {
               "id": 5,
               "name": "pizza",
               "price": 60.0,
               "availability": false,
               "mealIngredients": [
                   {
                       "id": 7,
                       "quantity": 5,
                       "ingredient": {
                           "id": 3,
                           "name": "ser",
                           "quantity": 12.0
                       }
                   }
               ]
           }
       ],
       "client": {
           "id": 1,
           "name": "Adam",
           "surname": "Nowak",
           "address": "ul. Wesoła 1, Warszawa",
           "email": "adam.nowak@example.com",
           "clientOrders": [
               {
                   "id": 10,
                   "orderedAt": 1657357714894,
                   "totalPrice": 25.0,
                   "meals": [
                       {
                           "id": 5,
                           "name": "pizza",
                           "price": 60.0,
                           "availability": false,
                           "mealIngredients": [
                               7
                           ]
                       }
                   ],
                   "client": 1
               }
           ]
       }
   }
]
```

Example result of GET query to `take/restaurant/order/{id}` (id=10):

```
{
    "id": 10,
    "orderedAt": 1657357714894,
    "totalPrice": 25.0,
    "meals": [
        {
            "id": 5,
            "name": "pizza",
            "price": 60.0,
            "availability": false,
            "mealIngredients": [
                {
                    "id": 7,
                    "quantity": 5,
                    "ingredient": {
                        "id": 3,
                        "name": "ser",
                        "quantity": 12.0
                    }
                }
            ]
        }
    ],
    "client": {
        "id": 1,
        "name": "Adam",
        "surname": "Nowak",
        "address": "ul. Wesoła 1, Warszawa",
        "email": "adam.nowak@example.com",
        "clientOrders": null
    }
}
```

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

```
{
    "id": 10,
    "orderedAt":"2012-04-23T18:25:43.511Z",
    "totalPrice":"65",
    "meals":[{"id":5}],
   "client":{"id":1}
}
```

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

(only 202 HTTP code returned, price changed to 65):

After check with GET query:

```
{
   "id": 10,
   "orderedAt": 1335205543511,
   "totalPrice": 65.0,
   "meals": [
       {
           "id": 5,
           "name": "pizza",
           "price": 60.0,
           "availability": false,
           "mealIngredients": [
               {
                   "id": 7,
                   "quantity": 5,
                   "ingredient": {
                       "id": 3,
                       "name": "ser",
                       "quantity": 12.0
                   }
               }
           ]
       }
   ],
   "client": {
       "id": 1,
       "name": "Adam",
       "surname": "Nowak",
       "address": "ul. Wesoła 1, Warszawa",
       "email": "adam.nowak@example.com",
       "clientOrders": null
   }
}
```

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

(deleting order with id = 11)

(only 204 HTTP code, order deleted)

Meal promotion endpoint

This is custom endpoint that applies a discount to a specific meal, eg. when there is much left at the end of a day.

This is a PUT request to `take/restaurant/meal/promotions/{id}/{discount}` endpoint, it takes id of the meal and discount percentage to be applied in form of path parameters. Example below:

GET on meal endpoint before promotions applied:

```
[
   {
       "id": 5,
       "name": "pizza",
       "price": 60.0,
       "availability": false,
       "mealIngredients": [
           {
               "id": 7,
               "quantity": 5,
               "ingredient": {
                   "id": 3,
                   "name": "ser",
                   "quantity": 12.0
               }
           }
       ]
   }
]
```

Executed query to that endpoint, it returnet empty response with HTTP code 204.

GET on meal endpoint after promotion applied:

```
[
    {
        "id": 5,
        "name": "pizza",
        "price": 54.0,
        "availability": false,
        "mealIngredients": [
            {
                "id": 7,
                "quantity": 5,
                "ingredient": {
                    "id": 3,
                    "name": "ser",
                    "quantity": 12.0
                }
            }
        ]
    }
]
```

So, 10% promotion has been applied successfully.

ERD Diagram

Below you can find ERD diagram for our final solution: