

Zaawansowane Technologie Baz Danych

System obsługi obiektu sportowego

- Dokumentacja

Maciej Drożdżal

Jarosław Górski

Łukasz Duda

Projekt realizowany jest jako RESTful API.

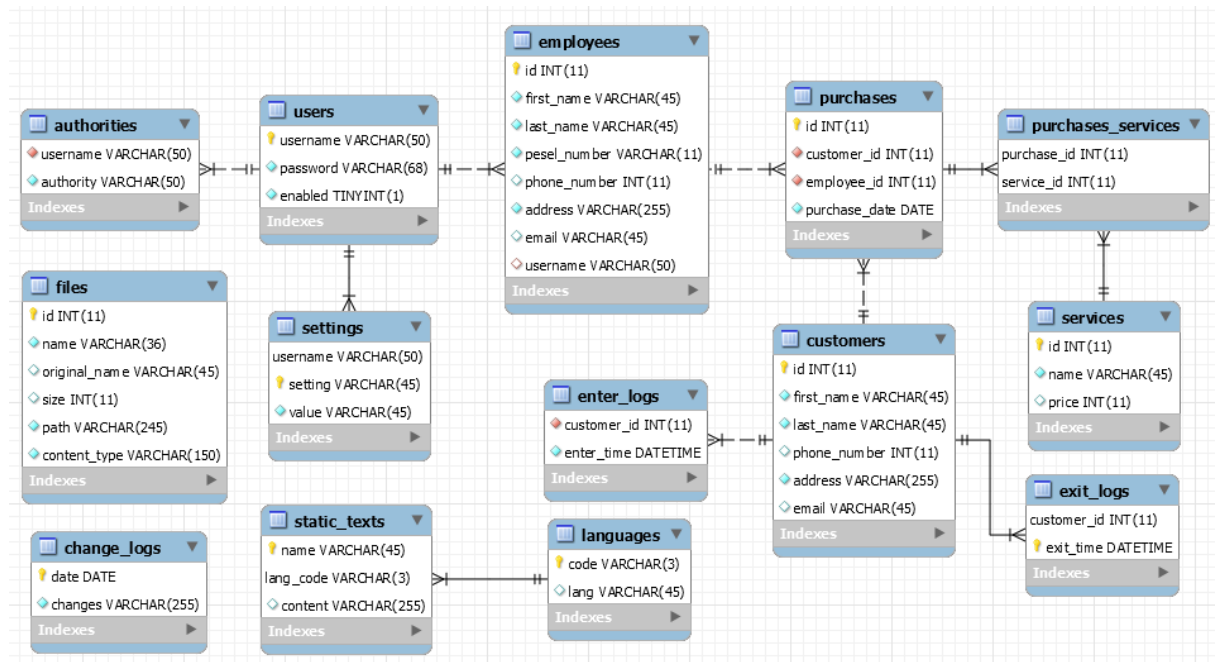
Wykonany został przy użyciu następujących technologii:

- Java wraz z użyciem kilku z bibliotek Spring'a,
- MySql 8.0.13.
- MongoDB.

Projekt zawiera następujące elementy:

- Rejestracja/edycja użytkowników,
- Ograniczenie przez role dostępu do odpowiednich endpoint'ów,
- Dodawanie/edycja klientów,
- Dodawanie/edycja pracowników,
- Dodawanie/edycja ofert obiektu sportowego,
- Przygotowywanie danych do sporządzenia prostej faktury (pracownik sprzedający ofertę, klient który ją kupuje, lista zakupionych ofert z ich cenami, data zakupu),
- Zapisywanie/edycja ustawień dla użytkowników (w formie "nazwa ustawienia" - "wartość"),
- Zarządzanie językami oraz listą statycznych tekstów w różnych językach (na podstawie kodów państw ISO 3166-1 alpha-3),
- Dodawanie/usuwanie plików oraz metadanych o tych plikach,
- Sporządzanie change logów mieszczących się w tablicy bazy danych,
- Rejestrowanie czasów wejść/wyjść klientów do/z obiektu sportowego.

Schemat:



REST API (logowanie, rejestracja, zarządzanie użytkownikami, role/dostępny)	4
Sposób przechowywania plików/słowników/StaticResources/ustawień, logger	16
Procedura, transakcja	17
Wyzwalacz	17
Tabela tymczasowa	18
Widok	18
Kopia zapasowa	18
MongoDB	19

REST API (logowanie, rejestracja, zarządzanie użytkownikami, role/dostępny)

Aplikacja posiada następujące *endpointy* wraz z rolami, które mają do nich dostęp – obsługiwane są tutaj wymagane do projektu podpunkty (logowanie, rejestracja, zarządzanie użytkownikami, role/dostępny):

1) Authorities (ROLE_ADMIN)

- `/api/authorities/{username}`

GET

`http://localhost:8080/api/authorities/jankow`

Odpowiedź:

```
[
  "ROLE_USER"
]
```

- `/api/authorities`

POST:

`http://localhost:8080/api/authorities`

```
{
  "primaryKey": {
    "user": {
      "username": "jankow"
    },
    "authority": "ROLE_ADMIN"
  }
}
```

Nowa *rola* dla użytkownika *jankow* została dodana do bazy danych:

username	authority
admin	ROLE_ADMIN
jankow	ROLE_ADMIN
jankow	ROLE_USER
marnow	ROLE_USER
NULL	NULL

- `/api/authorities/{username}/{authority}`

DELETE:

`http://localhost:8080/api/authorities/jankow/ROLE_ADMIN`

Rekord z *rolą* *ROLE_ADMIN* został usunięty dla użytkownika *jankow*:

username	authority
admin	ROLE_ADMIN
jankow	ROLE_USER
marnow	ROLE_USER
NULL	NULL

2) Users (ROLE_ADMIN)

- `/api/users`

GET:

`http://localhost:8080/api/users`

Odpowiedź:

```
[
  {
    "username": "admin",
    "password": "{bcrypt}$2a$10$XE474aijzJdmCA2PbKtS3unuskxgRK7KB7Qan3EGhSwuvMwL/XFGK",
    "enabled": true
  },
  {
    "username": "jankow",
    "password": "{bcrypt}$2a$10$QKS8t0PHPrf1.yGNrzJZW.yno/A1HaR3a6.RB5ZxCKcLO.MTfjuFG",
    "enabled": true
  },
  {
    "username": "marnow",
    "password": "{bcrypt}$2a$10$s5BaGQwIHtF7LTlr7mhFa01ubD3/0CtHE/Vhx.W5yui8X3cSnsFM.",
    "enabled": true
  }
]
```

- **/api/register** (dostęp bez autoryzacji)

POST:

http://localhost:8080/api/register

```
{
  "username": "kryjan",
  "password": "krysia1234",
  "enabled": 1
}
```

username	password	enabled
admin	{bcrypt}\$2a\$10\$XEA74ajzJdmCA2PbKtS3unus...	1
jankow	{bcrypt}\$2a\$10\$QKS8t0PHPrfl.yGNrzJZW.yno/...	1
kryjan	{bcrypt}\$2a\$10\$R/mJ31MlgS8c/EBOz7qusjeTj...	1
marnow	{bcrypt}\$2a\$10\$s5BaGQwIHtF7LTlr7mhFaO1u...	1
NULL	NULL	NULL

- **/api/users/{username}**

GET:

http://localhost:8080/api/users/jankow

Odpowiedź:

```
{
  "username": "jankow",
  "password": "{bcrypt}$2a$10$QKS8t0PHPrfl.yGNrzJZW.yno/AlHaR3a6.RB5ZxCKcLO.MTfjuFG",
  "enabled": true
}
```

DELETE:

http://localhost:8080/api/users/kryjan

username	password	enabled
admin	{bcrypt}\$2a\$10\$XEA74ajzJdmCA2PbKtS3unus...	1
jankow	{bcrypt}\$2a\$10\$QKS8t0PHPrfl.yGNrzJZW.yno/...	1
marnow	{bcrypt}\$2a\$10\$s5BaGQwIHtF7LTlr7mhFaO1u...	1
NULL	NULL	NULL

3) Customers (ROLE_USER, ROLE_ADMIN)

- **/api/customers**

GET:

http://localhost:8080/api/customers

Odpowiedź:

```
[
  {
    "id": 12,
    "firstName": "Witold",
    "lastName": "Mroczka",
    "phoneNumber": 222333444,
    "address": "Czerwona 2/43, Warszawa 23-321",
    "email": "witek.mroczka@gmail.com"
  }
]
```

POST:

http://localhost:8080/api/customers

```
{
  "firstName": "Krystyna",
  "lastName": "Janda",
  "phoneNumber": 555666777,
  "address": "Polna 45/13, Warszawa 32-329",
  "email": "krysia.janda@gmail.com"
}
```

id	first_name	last_name	phone_number	address	email
12	Witold	Mrocza	222333444	Czerwona 2/43, Warszawa 23-321	witek.mrocza@gmail.com
13	Krystyna	Janda	555666777	Polna 45/13, Warszawa 32-329	krystia.janda@gmail.com
NULL	NULL	NULL	NULL	NULL	NULL

- /api/customers/{customerId}

GET:

http://localhost:8080/api/customers/12

Odpowiedź:

```
{
  "id": 12,
  "firstName": "Witold",
  "lastName": "Mrocza",
  "phoneNumber": 222333444,
  "address": "Czerwona 2/43, Warszawa 23-321",
  "email": "witek.mrocza@gmail.com"
}
```

PUT:

http://localhost:8080/api/customers/12

```
{
  "email": "mrocza.witek@gmail.com"
}
```

id	first_name	last_name	phone_number	address	email
12	Witold	Mrocza	222333444	Czerwona 2/43, Warszawa 23-321	mrocza.witek@gmail.com
NULL	NULL	NULL	NULL	NULL	NULL

DELETE:

http://localhost:8080/api/customers/13

id	first_name	last_name	phone_number	address	email
12	Witold	Mrocza	222333444	Czerwona 2/43, Warszawa 23-321	mrocza.witek@gmail.com
NULL	NULL	NULL	NULL	NULL	NULL

4) Employees (ROLE_USER, ROLE_ADMIN)

- /api/employees

GET:

http://localhost:8080/api/employees

Odpowiedź:

```
[
  {
    "id": 5,
    "firstName": "Mariusz",
    "lastName": "Nowak",
    "peselNumber": "92031054451",
    "phoneNumber": 321321321,
    "address": "Cmentarna 12/21, Warszawa 43-323",
    "email": "mariusz.nowak@gmail.com",
    "user": {
      "username": "marnow",
      "password": "{bcrypt}$2a$10$58aGQwIHtF7LTlr7mhFa01ubD3/0CtHE/Vhx.W5yui8X3cSnsFM.",
      "enabled": true
    }
  },
  {
    "id": 6,
    "firstName": "Jan",
    "lastName": "Kowalski",
    "peselNumber": "89100264231",
    "phoneNumber": 123123123,
    "address": "Uroczna 2/13, Warszawa 21-123",
    "email": "janekowalski@gmail.com",
    "user": {
      "username": "jankow",
      "password": "{bcrypt}$2a$10$QKS8t0PHPrf1.yGNrzJJW.yno/AlHaR3a6.RB5ZxCKcLO.MTfjufG",
      "enabled": true
    }
  }
]
```

POST:

http://localhost:8080/api/employees

```
{
  "firstName": "Marcin",
  "lastName": "Stopyra",
  "peselNumber": "87120356251",
  "phoneNumber": 789789789,
  "address": "Zielona 32/1, Warszawa 41-301",
  "email": "marcin.stopyra@gmail.com"
}
```

Odpowiedź:

id	first_name	last_name	pesel_number	phone_number	address	email	username
5	Mariusz	Nowak	92031054451	321321321	Cmentarna 12/21, Warszawa 43-323	mariusz.nowak@gmail.com	marnow
6	Jan	Kowalski	89100264231	123123123	Uroczna 2/13, Warszawa 21-123	janekowalski@gmail.com	jankow
7	Marcin	Stopyra	87120356251	789789789	Zielona 32/1, Warszawa 41-301	marcin.stopyra@gmail.com	marsto
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

- /api/employees/{employeeId}

PUT:

http://localhost:8080/api/employees/7

```
{
  "user": {
    "username": "marsto",
    "password": "marcin123",
    "enabled": 1
  }
}
```

id	first_name	last_name	pesel_number	phone_number	address	email	username
5	Mariusz	Nowak	92031054451	321321321	Cmentarna 12/21, Warszawa 43-323	mariusz.nowak@gmail.com	marnow
6	Jan	Kowalski	89100264231	123123123	Uroczna 2/13, Warszawa 21-123	janekowalski@gmail.com	jankow
7	Marcin	Stopyra	87120356251	789789789	Zielona 32/1, Warszawa 41-301	marcin.stopyra@gmail.com	marsto
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

GET:

http://localhost:8080/api/employees/7

Odpowiedź:

```
{
  "id": 7,
  "firstName": "Marcin",
  "lastName": "Stopyra",
  "peselNumber": "87120356251",
  "phoneNumber": 789789789,
  "address": "Zielona 32/1, Warszawa 41-301",
  "email": "marcin.stopyra@gmail.com",
  "user": {
    "username": "marsto",
    "password": "{bcrypt}$2a$10$w68d1EeH1hdd62eHBCTwEueuSxjytSpzo1U9jySspGDWge02pFOri",
    "enabled": true
  }
}
```

DELETE:

http://localhost:8080/api/employees/7

id	first_name	last_name	pesel_number	phone_number	address	email	username
5	Mariusz	Nowak	92031054451	321321321	Cmentarna 12/21, Warszawa 43-323	mariusz.nowak@gmail.com	marnow
6	Jan	Kowalski	89100264231	123123123	Uroczna 2/13, Warszawa 21-123	janekowalski@gmail.com	jankow
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

5) Services (ROLE_USER, ROLE_ADMIN)

- /api/services

GET:

http://localhost:8080/api/services

Odpowiedź:

```
[
  {
    "id": 1,
    "name": "Gym",
    "price": 120
  },
  {
    "id": 2,
    "name": "Swimming pool",
    "price": 90
  },
  {
    "id": 3,
    "name": "Personal training",
    "price": 20
  }
]
```

POST:

http://localhost:8080/api/services

```
{
  "name": "Tennis",
  "price": 85
}
```

id	name	price
1	Gym	120
2	Swimming pool	90
3	Personal training	20
4	Tennis	85
NULL	NULL	NULL

- /api/services/{serviceId}

GET:

http://localhost:8080/api/services/4

Odpowiedź:

```
{
  "id": 4,
  "name": "Tennis",
  "price": 85
}
```

PUT:

http://localhost:8080/api/services/4

```
{
  "price": 70
}
```

id	name	price
1	Gym	120
2	Swimming pool	90
3	Personal training	20
4	Tennis	70
NULL	NULL	NULL

DELETE:

http://localhost:8080/api/services/4

id	name	price
1	Gym	120
2	Swimming pool	90
3	Personal training	20
NULL	NULL	NULL

6) Purchases (ROLE_USER, ROLE_ADMIN)

- **/api/purchases**

GET:

http://localhost:8080/api/purchases

Odpowiedź:

```
[
  {
    "id": 2,
    "purchaseDate": "2018-11-25",
    "services": [
      {
        "id": 1,
        "name": "Gym",
        "price": 120
      },
      {
        "id": 2,
        "name": "Swimming pool",
        "price": 90
      }
    ],
    "purchaseEmployee": "Nowak Mariusz",
    "purchaseCustomer": "Mroczka Witold"
  },
  {
    "id": 3,
    "purchaseDate": "2018-11-24",
    "services": [
      {
        "id": 3,
        "name": "Personal training",
        "price": 20
      }
    ],
    "purchaseEmployee": "Kowalski Jan",
    "purchaseCustomer": "Mroczka Witold"
  }
]
```

POST:

http://localhost:8080/api/purchases

```
{
  "purchaseDate": "2018-11-27",
  "services": [
    {
      "id": 1
    }
  ],
  "employee": {
    "id": 5
  },
  "customer": {
    "id": 12
  }
}
```

id	customer_id	employee_id	purchase_date
2	12	5	2018-11-25
3	12	6	2018-11-24
4	12	5	2018-11-27
NULL	NULL	NULL	NULL

purchase_id	service_id
2	1
4	1
2	2
3	3
NULL	NULL

- **/api/purchases/employee/{employeeId}**

GET:

http://localhost:8080/api/purchases/employee/5

Odpowiedź:

```
[
  {
    "id": 2,
    "purchaseDate": "2018-11-25",
    "services": [
      {
        "id": 1,
        "name": "Gym",
        "price": 120
      },
      {
        "id": 2,
        "name": "Swimming pool",
        "price": 90
      }
    ],
    "purchaseEmployee": "Nowak Mariusz",
    "purchaseCustomer": "Mroczka Witold"
  },
  {
    "id": 4,
    "purchaseDate": "2018-11-27",
    "services": [
      {
        "id": 1,
        "name": "Gym",
        "price": 120
      }
    ],
    "purchaseEmployee": "Nowak Mariusz",
    "purchaseCustomer": "Mroczka Witold"
  }
]
```

- /api/purchases/customer/{customerId}

GET:

http://localhost:8080/api/purchases/customer/12

Odpowiedź:

```
[
  {
    "id": 2,
    "purchaseDate": "2018-11-25",
    "services": [
      {
        "id": 1,
        "name": "Gym",
        "price": 120
      },
      {
        "id": 2,
        "name": "Swimming pool",
        "price": 90
      }
    ],
    "purchaseEmployee": "Nowak Mariusz",
    "purchaseCustomer": "Mroczka Witold"
  },
  {
    "id": 4,
    "purchaseDate": "2018-11-27",
    "services": [
      {
        "id": 1,
        "name": "Gym",
        "price": 120
      }
    ],
    "purchaseEmployee": "Nowak Mariusz",
    "purchaseCustomer": "Mroczka Witold"
  },
  {
    "id": 3,
    "purchaseDate": "2018-11-24",
    "services": [
      {
        "id": 3,
        "name": "Personal training",
        "price": 20
      }
    ],
    "purchaseEmployee": "Kowalski Jan",
    "purchaseCustomer": "Mroczka Witold"
  }
]
```

- **/api/purchases/{purchaseId}**
GET:
http://localhost:8080/api/purchases/4

Odpowiedź:

```
{
  "id": 4,
  "purchaseDate": "2018-11-27",
  "services": [
    {
      "id": 1,
      "name": "Gym",
      "price": 120
    }
  ],
  "purchaseEmployee": "Nowak Mariusz",
  "purchaseCustomer": "Mroczka Witold"
}
```

7) Settings (ROLE_USER, ROLE_ADMIN)

- **/api/settings**
POST:
http://localhost:8080/api/settings

```
{
  "primaryKey": {
    "user": {
      "username": "jankow"
    },
    "setting": "font"
  },
  "value": "ARIAL"
}
```

username	setting	value
admin	language	POL
admin	Text size	12
admin	theme	RED
jankow	font	ARIAL
jankow	language	POL
jankow	Text size	8
jankow	Theme	BLUE
marnow	language	POL
marnow	Text size	10
marnow	theme	DARK BLUE
NULL	NULL	NULL

- **/api/settings/{username}**
GET:
http://localhost:8080/api/settings/jankow

Odpowiedź:

```
{
  "Text size": "8",
  "Theme": "BLUE",
  "language": "POL",
  "font": "ARIAL"
}
```

- **/api/settings/{username}/{setting}**
GET:
http://localhost:8080/api/settings/jankow/font

Odpowiedź:

```
ARIAL
```

- **/api/settings/{username}/{setting}**
DELETE:
http://localhost:8080/api/settings/jankow/font

username	setting	value
admin	language	POL
admin	Text size	12
admin	theme	RED
jankow	language	POL
jankow	Text size	8
jankow	Theme	BLUE
marnow	language	POL
marnow	Text size	10
marnow	theme	DARK BLUE
NULL	NULL	NULL

8) Languages (ROLE_ADMIN)

- **/api/languages**

GET:

<http://localhost:8080/api/languages>

Odpowiedź:

```
[
  {
    "code": "GBR",
    "lang": "english"
  },
  {
    "code": "POL",
    "lang": "polish"
  },
  {
    "code": "USA",
    "lang": "english"
  }
]
```

POST:

<http://localhost:8080/api/languages>

```
{
  "code": "FRA",
  "lang": "french"
}
```

code	lang
FRA	french
GBR	english
POL	polish
USA	english
NULL	NULL

- **/api/languages/{langCode}**

GET:

<http://localhost:8080/api/languages/FRA>

Odpowiedź:

```
{
  "code": "FRA",
  "lang": "french"
}
```

DELETE:

<http://localhost:8080/api/languages/FRA>

code	lang
GBR	english
POL	polish
USA	english
NULL	NULL

9) StaticTexts (ROLE_ADMIN)

- **/api/staticTexts**

GET:

http://localhost:8080/api/staticTexts

Odpowiedź:

```
[
  {
    "content": "Add new customer.",
    "name": "addCustomer",
    "language": {
      "code": "GBR",
      "lang": "english"
    }
  },
  {
    "content": "Dodaj klienta.",
    "name": "addCustomer",
    "language": {
      "code": "POL",
      "lang": "polish"
    }
  },
  {
    "content": "Add new customer.",
    "name": "addCustomer",
    "language": {
      "code": "USA",
      "lang": "english"
    }
  },
  {
    "content": "Prepare data for invoice.",
    "name": "prepInvoiceData",
    "language": {
      "code": "GBR",
      "lang": "english"
    }
  },
  {
    "content": "Przygotuj dane do faktury.",
    "name": "prepInvoiceData",
    "language": {
      "code": "POL",
      "lang": "polish"
    }
  },
  {
    "content": "Prepare data for invoice.",
    "name": "prepInvoiceData",
    "language": {
      "code": "USA",
      "lang": "english"
    }
  }
]
```

POST:

http://localhost:8080/api/staticTexts

```
{
  "primaryKey": {
    "language": {
      "code": "POL"
    },
    "name": "addService"
  },
  "content": "Dodaj nowa oferte"
}
```

name	lang_code	content
addCustomer	GBR	Add new customer.
addCustomer	POL	Dodaj klienta.
addCustomer	USA	Add new customer.
addService	POL	Dodaj nowa oferte
prepInvoiceData	GBR	Prepare data for invoice.
prepInvoiceData	POL	Przygotuj dane do faktury.
prepInvoiceData	USA	Prepare data for invoice.
NULL	NULL	NULL

- /api/staticTexts/{name}/{langCode}

GET:

http://localhost:8080/api/staticTexts/addService/POL

Odpowiedź:

```
{
  "content": "Dodaj nowa oferte",
  "name": "addService",
  "language": {
    "code": "POL",
    "lang": "polish"
  }
}
```

DELETE:

http://localhost:8080/api/staticTexts/addService/POL

name	lang_code	content
addCustomer	GBR	Add new customer.
addCustomer	POL	Dodaj klienta.
addCustomer	USA	Add new customer.
prepInvoiceData	GBR	Prepare data for invoice.
prepInvoiceData	POL	Przygotuj dane do faktury.
prepInvoiceData	USA	Prepare data for invoice.
NULL	NULL	NULL

10) Files (ROLE_ADMIN)

- /api/files

GET:

http://localhost:8080/api/files

Odpowiedź:

```
{
  "id": 2,
  "name": "99ee6271-3a39-487d-a536-e66d6e5ee8b1",
  "originalName": "plan.xls",
  "size": 128512,
  "path": "C:\\Users\\M\\Desktop\\Bazy Danych\\ObiektSportowy\\files\\99ee6271-3a39-487d-a536-e66d6e5ee8b1",
  "contentType": "application/vnd.ms-excel"
}
```

POST:

POST http://localhost:8080/api/files

Params Authorization Headers (2)

none form-data x-www-form-urlencoded

KEY	VALUE
<input checked="" type="checkbox"/> multipartFile	Wybierz pliki eski.txt

id	name	original_name	size	path	content_type
2	99ee6271-3a39-487d-a536-e66d6e5ee8b1	plan.xls	128512	C:\Users\M\Desktop\Bazy Danych\ObiektSportowy\files\99ee6271-3a39-487d-a536-e66d6e5ee8b1	application/vnd.ms-excel
3	9ad716ca-5ab1-4cb5-8dc5-7670ba77f8ee	eski.txt	328	C:\Users\M\Desktop\Bazy Danych\ObiektSportowy\files\9ad716ca-5ab1-4cb5-8dc5-7670ba77f8ee	text/plain
NULL	NULL	NULL	NULL	NULL	NULL

Bazy Danych > ObiektSportowy > files

Umieść w bibliotece Udostępnij Nowy folder

	Nazwa	Data modyfikacji	Typ	Rozmiar
one				
atnie miejsca	9ad716ca-5ab1-4cb5-8dc5-7670ba77f8ee	2018-11-27 17:48	Plik	1 KB
rane	99ee6271-3a39-487d-a536-e66d6e5ee8b1	2018-11-25 16:37	Plik	126 KB

- /api/files/{fileId}

GET:

http://localhost:8080/api/files/3

Odpowiedź:

```
{
  "id": 3,
  "name": "9ad716ca-5ab1-4cb5-8dc5-7670ba77f8ee",
  "originalName": "eski.txt",
  "size": 328,
  "path": "C:\\Users\\M\\Desktop\\Bazy Danych\\ObiektSportowy\\files\\9ad716ca-5ab1-4cb5-8dc5-7670ba77f8ee",
  "contentType": "text/plain"
}
```

DELETE:

http://localhost:8080/api/files/3

id	name	original_name	size	path	content_type
2	99ee6271-3a39-487d-a536-e66d6e5ee8b1	plan.xls	128512	C:\\Users\\M\\Desktop\\Bazy Danych\\ObiektSporto...	application/vnd.ms-excel
NULL	NULL	NULL	NULL	NULL	NULL

one	Nazwa	Data modyfikacji	Typ	Rozmiar
atnie miejsca	99ee6271-3a39-487d-a536-e66d6e5ee8b1	2018-11-25 16:37	Plik	126 KB
irane				

11) ChangeLogs (ROLE_ADMIN)

- /api/changeLogs

POST:

http://localhost:8080/api/changeLogs

```
{
  "date": "2018-11-27",
  "changes": "Dodanie możliwości rejestracji użytkowników oraz przypisania im roli"
}
```

date	changes
2018-11-27	Dodanie możliwości rejestracji użytkowników or...
NULL	NULL

GET:

http://localhost:8080/api/changeLogs

Odpowiedź:

```
[
  {
    "date": "2018-11-27",
    "changes": "Dodanie możliwości rejestracji użytkowników oraz przypisania im roli"
  }
]
```

- /api/changeLogs/{year}/{month}/{day}

GET:

http://localhost:8080/api/changeLogs/2018/11/27

Odpowiedź:

```
{
  "date": "2018-11-27",
  "changes": "Dodanie możliwości rejestracji użytkowników oraz przypisania im roli"
}
```

DELETE:

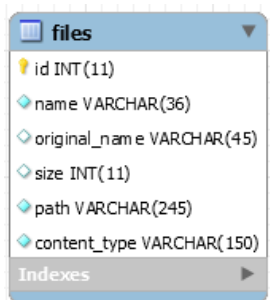
http://localhost:8080/api/changeLogs/2018/11/27

date	changes
NULL	NULL

Sposób przechowywania plików/słowników/StaticResources/ustawień, logger

Zgodnie z wymaganiami projektu propozycja przechowywania wyżej wymienionych danych to dodatkowe tabele w bazie danych:

Pliki:



Name – wygenerowana automatycznie unikalna nazwa pliku.

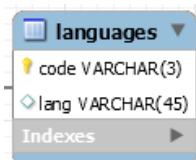
Original_name – oryginalna nazwa pliku sprzed wygenerowania unikalnej.

Size – rozmiar pliku.

Path – ścieżka dostępu do pliku na dysku.

Content_type – typ przechowywanego pliku.

Słowniki:

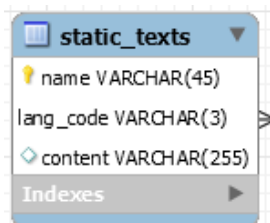


Przykładem przechowywania słowników jest np. język zapisywany jako kod państwa.

Code – kod państwa wg ISO 3166-1 alpha-3.

Lang – pełna nazwa używanego języka.

StaticResources:



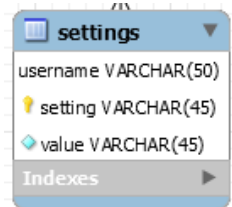
Przykładem takiego zasobu są np. statyczne dane tekstowe wyświetlane przez aplikację. Gdyby jakiś tekst przekraczał ilość znaków 255, wtedy jest przechowywany jako plik tekstowy w tabeli z plikami.

Name – unikalna nazwa statycznego tekstu.

Lang_code – klucz obcy do tabeli z językami (język tekstu do wyświetlenia).

Content – statyczny tekst do wyświetlenia.

Ustawienia:



Ustawienia przechowywane na zasadzie klucz-wartość (np. font-size -> 12).

Setting – kod opcji do ustawienia.

Value – wartość opcji/ustawienia.

Username – klucz obcy zawierający informację dla jakiego użytkownika przechowywane są dane ustawienia.

Logger:

change_logs	enter_logs
date DATE	customer_id INT(11)
changes VARCHAR(255)	enter_time DATETIME
Indexes	Indexes

Przykładowe tabele do logów. Pierwsza zawiera typowy changelog kluczem głównym jest tutaj data, a pole tekstowe przechowuje opis zmian. Druga zawiera zapis wejść klienta na obiekt, unikalnym kluczem są obie wartości czyli id klienta oraz czas wejścia do obiektu.

Procedura, transakcja

Przykład procedury i transakcji zaprezentowano na podstawie dodawania usługi do bazy. W przypadku niepowodzenia transakcji zdefiniowano handler odpowiadający za ROLLBACK.

```
DELIMITER $$
CREATE DEFINER=`mdrozdz`@`localhost` PROCEDURE `add_service`(IN id INT(11),
    IN service_name VARCHAR(45),
    IN price INT(11))
BEGIN
    DECLARE exit handler for sqlexception
    BEGIN
        ROLLBACK;
    END;

    START TRANSACTION;

    INSERT INTO `services` (`id`, `name`, `price`)
    VALUES (id, service_name, price);
    COMMIT;
END$$
DELIMITER ;
```

Przykład wywołania procedury:

```
call sports_facility.add_service(3, 'Jogging', 120);
```

Wyzwalacz

Przykładem wyzwalacza (triggera) jest sprawdzanie przed dodaniem usługi do bazy czy podany id już nie istnieje. Jeśli istnieje, id rekordu który ma zostać dodany jest zwiększany o 1 odnośnie największej obecnie wartości id.

```
delimiter $$
create trigger sports_facility.before_service_insert
before insert on sports_facility.services
for each row
begin
    if exists (select 1 from sports_facility.services
        where sports_facility.services.id = new.id)
    then
        set new.id = (select max(id) + 1 from sports_facility.services);
    end if;
end$$
delimiter ;
```

Tabela tymczasowa

Przykładem wykorzystania tymczasowej tabeli jest stworzenie tabeli zawierającej 3 najczęściej kupowane usługi. Tabela jest tworzona na podstawie wyciągniętych odpowiednich danych z istniejących wcześniej tabel. Następnie realizowany jest SELECT wyświetlający dane tabeli tymczasowej po czym tabela jest usuwana.

```
create temporary table `sports_facility`.`top3_services`
select `sports_facility`.`services`.`id`,
`sports_facility`.`services`.`name`,
`sports_facility`.`services`.`price`,
count(*) as `count`
from `sports_facility`.`services` join `sports_facility`.`purchases_services`
on `sports_facility`.`services`.`id` = `sports_facility`.`purchases_services`.`service_id`
group by `sports_facility`.`purchases_services`.`service_id`
order by `count` desc limit 3;

select * from `sports_facility`.`top3_services`;

drop temporary table `sports_facility`.`top3_services`;
```

Widok

Wykorzystanie widoku w prezentowanej bazie danych służy do bardziej czytelnej reprezentacji podstawowych danych do faktury.

```
CREATE ALGORITHM=UNDEFINED DEFINER=`mdrozd`@`localhost` SQL SECURITY DEFINER VIEW `sports_facility`.`invoice_data` AS
select `sports_facility`.`customers`.`first_name` AS `C First Name`,`sports_facility`.`customers`.`last_name` AS
`C Last Name`,
`sports_facility`.`customers`.`address` AS `C Address`,`sports_facility`.`employees`.`first_name` AS `first_name`,
`sports_facility`.`employees`.`last_name` AS `last_name`,`sports_facility`.`employees`.`address` AS `address`,
`sports_facility`.`services`.`name` AS `name`,`sports_facility`.`services`.`price` AS `price`,
`sports_facility`.`purchases`.`purchase_date` AS `purchase_date`
from ((((`sports_facility`.`purchases` join `sports_facility`.`employees`
on((`sports_facility`.`employees`.`id` = `sports_facility`.`purchases`.`employee_id`)))
join `sports_facility`.`customers`
on((`sports_facility`.`customers`.`id` = `sports_facility`.`purchases`.`customer_id`)))
left join `sports_facility`.`purchases_services`
on((`sports_facility`.`purchases`.`id` = `sports_facility`.`purchases_services`.`purchase_id`)))
left join `sports_facility`.`services`
on((`sports_facility`.`purchases_services`.`service_id` = `sports_facility`.`services`.`id`)));
```

Przykład użycia:

```
1 • select * from sports_facility.invoice_data;
```

C First Name	C Last Name	C Address	first_name	last_name	address	name	price	purchase_date
Krystyna	Janda	Polna 45/13, Warszawa 32-329	Marcin	Stopyra	Zielona 32/1, Warszawa 41-301	Tennis	85	2018-11-27

Kopia zapasowa

Kopia zapasowa struktury bazy wraz z jej danymi realizowana jest za pomocą programu *mysqldump*. W konsoli należy przejść do folderu zainstalowanego serwera MySQL i użyć odpowiedniej komendy. Podajemy login, hasło, nazwę bazy oraz lokalizację do której ma zostać zapisany plik z kopią zapasową.

```
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysqldump -u mdrozd -pmdrozd sport
s_facility > "C:\Users\M\Desktop\Bazy Danych\backup\backup.sql"
mysqldump: [Warning] Using a password on the command line interface can be insecure.
```

MongoDB

Po zmianie bazy z MySQL na MongoDB endpointy w API pozostały bez zmian, zmienił się tylko sposób przechowywania danych. Stworzenie repozytoriów na podstawie interfejsu udostępnionego przez Spring Framework pozwoliło dostać się do metod pozwalających na zapis, odczyt, modyfikację i usunięcie danych z dokumentów MongoDB.

```
package com.ztbd.sportsfacility.dao;

import com.ztbd.sportsfacility.model.Employee;
import org.springframework.data.mongodb.repository.MongoRepository;

public interface EmployeeRepository extends MongoRepository<Employee, String> {}
```

Dane przechowywane są w dokumentach w formacie podobnym do JSON. Strukturę zatem można uprościć, nie ma potrzeby tworzenia osobno dokumentów dla odpowiednika tabeli Authority, za to role dostępu można umieścić bezpośrednio w dokumentach przeznaczonych do przechowywania użytkowników. W razie potrzeby utworzenia relacji pomiędzy dwoma kolekcjami w dokumentach umieszczane są Id odpowiednich elementów np. dla dokumentu przechowującego pracownika istnieje pole przechowujące Id odpowiadające jego użytkownikowi. Spring Framework pozwala na automatyczne podstawienie całego dokumentu w miejscu gdzie powinno być jego Id, dlatego w polu użytkownika nie zobaczylibyśmy informacji o wartości Id, a wyświetlone by zostały dane użytkownika (username itd.).

```
@Document(collection = "employee")
public class Employee {

    @Id
    private String id;

    private String firstName;

    private String lastName;

    private String peselNumber;

    private int phoneNumber;

    private String address;

    private String email;

    @DBRef
    private User user;

    @DBRef
    private List<Purchase> purchases;
}
```

Dane przekształcone dla przeniesienia bazy z MySQL do MongoDB posiadają następującą strukturę.

Nazwa bazy:

```
> show dbs
admin          0.0000GB
config         0.0000GB
local          0.0000GB
sportsFacility 0.0000GB
>
```

Kolekcje:

```
> show collections
changeLog
customer
employee
file
language
purchase
service
user
>
```

changeLog:

```
> db.changeLog.find().pretty()
{
  "_id" : ObjectId("5c3f6a2f80c2e8177ce28bcf"),
  "date" : ISODate("2019-01-15T23:00:00Z"),
  "changes" : "blablabla bla bla bla",
  "_class" : "com.ztbd.sportsfacility.model.ChangeLog"
}
{
  "_id" : ObjectId("5c3f6a4e80c2e8177ce28bd0"),
  "date" : ISODate("2019-01-14T23:00:00Z"),
  "changes" : "blablabla bla bla bla",
  "_class" : "com.ztbd.sportsfacility.model.ChangeLog"
}
>
```

file:

```
> db.file.find().pretty()
{
  "_id" : ObjectId("5c46ec00f3356117b8200d62"),
  "name" : "d8d26af1-f386-4ccd-99f8-7510f58e93c5",
  "originalName" : "Lab7.xlsx",
  "size" : 47108,
  "path" : "C:\\Users\\M\\Desktop\\Bazy Danych\\sports-facility-mongo\\src\\main\\resources\\static\\files\\d8d26af1-f386-4ccd-99f8-7510f58e93c5",
  "contentType" : "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet",
  "_class" : "com.ztbd.sportsfacility.model.File"
}
>
```

service:

```
> db.service.find().pretty()
{
  "_id" : ObjectId("5c46ee12f3356117b8200d64"),
  "name" : "gym",
  "price" : 70,
  "_class" : "com.ztbd.sportsfacility.model.Service"
}
{
  "_id" : ObjectId("5c46ee21f3356117b8200d65"),
  "name" : "tennis",
  "price" : 85,
  "_class" : "com.ztbd.sportsfacility.model.Service"
}
>
```

customer:

```
> db.customer.find().pretty()
{
  "_id" : ObjectId("5c3f6cb780c2e8047044f820"),
  "firstName" : "Krystyna",
  "lastName" : "Janda",
  "phoneNumber" : 555666777,
  "address" : "Polna 45/13, Warszawa 32-329",
  "email" : "krysia.janda@gmail.com",
  "_class" : "com.ztbd.sportsfacility.model.Customer"
}
{
  "_id" : ObjectId("5c3f6d0480c2e8047044f822"),
  "firstName" : "Jan",
  "lastName" : "Kowalski",
  "phoneNumber" : 111222333,
  "address" : "Słoneczna 21/3, Kraków 10-200",
  "email" : "jan.kowalski@gmail.com",
  "_class" : "com.ztbd.sportsfacility.model.Customer"
}
>
```

user zawierający teraz ustawienia (settings) oraz role (authorities) bez tworzenia relacji z innymi kolekcjami:

```
> db.user.find().pretty()
{
  "_id" : ObjectId("5c3f670000c2e81650d1ab79"),
  "username" : "admin",
  "password" : "{bcrypt}$2a$10$A2U83yN7Eo2FyqFAGGqCU0LsgkaltKLMiGs1qhxyIDrAxhUzTsp6",
  "authorities" : [
    "ADMIN",
    "USER"
  ],
  "settings" : [
    {
      "setting" : "font-size",
      "value" : "12"
    },
    {
      "setting" : "font-color",
      "value" : "black"
    }
  ],
  "_class" : "com.ztbd.sportsfacility.model.User"
}
>
```

language zawiera informacje o statycznym tekście, w MySQL tabelę do tych danych były oddzielne:

```
> db.language.find().pretty()
{
  "_id" : ObjectId("5c46ed90f3356117b8200d63"),
  "code" : "POL",
  "lang" : "polski",
  "staticTexts" : {
    "login" : "Zaloguj",
    "addService" : "Dodaj nową usługę",
    "addCustomer" : "Dodaj nowego klienta"
  },
  "_class" : "com.ztbd.sportsfacility.model.Language"
}
```

employee odebrane przez aplikację/postmana zawiera uzupełnione już dane (user) zebrane z innych kolekcji/dokumentów, natomiast w bazie przechowywane są, jak już zostało wspomniane, jedynie Id:

```
{
  "id": "5c3f6e2180c2e8047044f823",
  "firstName": "Marcin",
  "lastName": "Stopyra",
  "peselNumber": "87120356251",
  "phoneNumber": "789789789",
  "address": "Zielona 32/1, Warszawa 41-301",
  "email": "marcin.stopyra@gmail.com",
  "user": {
    "id": "5c3f682480c2e8177ce28bce",
    "username": "user",
    "password": "{bcrypt}$2a$10$8GJMiascYqISlaIMx01vKuG3m6s3j45BjOLaD7Gq7Io82rAjz13e.",
    "authorities": [
      "USER"
    ],
    "settings": [
      {
        "setting": "font-size",
        "value": "10"
      },
      {
        "setting": "font-color",
        "value": "dark-blue"
      }
    ]
  },
  "purchases": null
}
```

```
> db.employee.find().pretty()
{
  "_id" : ObjectId("5c3f6e2180c2e8047044f823"),
  "firstName" : "Marcin",
  "lastName" : "Stopyra",
  "peselNumber" : "87120356251",
  "phoneNumber" : "789789789",
  "address" : "Zielona 32/1, Warszawa 41-301",
  "email" : "marcin.stopyra@gmail.com",
  "user" : DBRef("user", ObjectId("5c3f682480c2e8177ce28bce")),
  "_class" : "com.ztbd.sportsfacility.model.Employee"
}
```

purchase:

```
> db.purchase.find().pretty()
{
  "_id" : ObjectId("5c46efa2f3356117b8200d66"),
  "purchaseDate" : ISODate("2019-01-21T23:00:00Z"),
  "customer" : DBRef("customer", ObjectId("5c3f6cb780c2e8047044f820")),
  "employee" : DBRef("employee", ObjectId("5c3f6e2180c2e8047044f823")),
  "services" : [
    DBRef("service", ObjectId("5c46ee12f3356117b8200d64")),
    DBRef("service", ObjectId("5c46ee21f3356117b8200d65"))
  ],
  "_class" : "com.ztbd.sportsfacility.model.Purchase"
}
```