

Nowoczesne języki programowania



UNIWERSYTET ŚLĄSKI
W KATOWICACH

Projekt zaliczeniowy
Igor Szałas
Grupa 3
Informatyka Stacjonarne I rok

Opis projektu

Projekt zaliczeniowy został zrealizowany z wykorzystaniem języka Kotlin oraz frameworka SpringBoot. Wykorzystana do jego realizacji została również baza danych MySQL której instancja została postawiona w formie obrazu Dockera. Aplikacja w założeniu ma być internetową platformą pozwalającą na znalezienie osoby chętnej pomóc z codziennymi czynnościami wymagających specjalnych narzędzi lub umiejętności. Aplikacja posiada część frontendową która zostanie stworzona u ukończona w późniejszym czasie lecz chcę by ten projekt był zrealizowany możliwie dobrze i strona ta została opublikowana. Planuję do części backendowej dodać również mechanizm JWT służący do logowania się do platformy i obsługi sesji. Część backendowa na razie jest we wczesnej fazie i posiada możliwość odczytania danych z kilku tabel. Obecnie baza danych zastosowana w tym projekcie posiada 3 tabele Opinions, Announcements, Users. Możliwe jest obecnie wykonywanie samych zapytań GET które wyświetlają przechowywane dane.

Prezentacja projektu

Projekt stosuje wzorzec Model-View-Controller i poniżej został przedstawiony przykładowy kontroler pozwalający na odczyt danych z tabeli Announcements. Możliwe jest wyszukanie ogłoszenie na podstawie tytułu oraz wyświetlenie wszystkich ogłoszeń. Ułatwiam sobie w tym przypadku pracę wykorzystując interfejsy które udostępniają mechanizmy wykorzystywania zasobów obsługiwanej bazy danych.

```
@RestController
@RequestMapping("/announcement")
class AnnouncementController(val announcementRepository:
AnnouncementRepository){
    @GetMapping("/")
    fun announcementTest(): String {
        return "Test announcement"
    }

    @GetMapping("/all")
    fun getAllAnnouncements(): List<Announcement> =
announcementRepository.findAll()

    @GetMapping("/title/{title}")
    fun getAnnouncementsByTitle(@PathVariable("title") title: String):
Optional<List<Announcement>> =
announcementRepository.findAnnouncementByTitle(title);
}
```

Poniżej przedstawiam jaki rezultat daje wysłanie zapytania na powyższe adresy.

```
localhost:8080/announcement/all
Zastosuj formatowanie stylistyczne ☒
[
  {
    "id": 1,
    "title": null,
    "content": "sss",
    "category": null,
    "authorId": null
  },
  {
    "id": 2,
    "title": "Test Title",
    "content": "test",
    "category": null,
    "authorId": 12
  }
]
```

```
localhost:8080/announcement/title/TestTitle
Zastosuj formatowanie stylistyczne ☒
[
  {
    "id": 2,
    "title": "TestTitle",
    "content": "test",
    "category": null,
    "authorId": 12
  }
]
```

Poniżej zamieszczam kod z kontrolera do obsługi bazy danych dla tabeli Users.

```
@RestController
@RequestMapping("/user")
class UserController(val userRepository: UserRepository){

    @GetMapping("/")
    fun test(): String {
        return "Test"
    }
}
```

```

    @GetMapping("/all")
    fun getAllUsers(): MutableIterable<User> = userRepository.findAll();

    @GetMapping("/id")
    fun getUserById(@RequestBody id: Int): User? =
userRepository.findById(id);

    @GetMapping("/name/{name}")
    fun getUserByName(@PathVariable(value = "name") name: String ):
List<User>? = userRepository.findUserByName(name)



    @GetMapping("/city/{city}")
    fun getUserByCity(@PathVariable(value = "city") city: String): List<User>?
= userRepository.findUsersByCity(city)

    @PostMapping("/add")
    fun addUser(@RequestBody user: User) : User = userRepository.save(user);

    @DeleteMapping("/remove/{id}")
    fun deleteUserById(@PathVariable(value = "id") id: Int) =
userRepository.deleteUserById(id);
}

```

Dodatkowo poniżej zamieszczam wynik dla działania przykładowego endpointu.

< > ↻   localhost:8080/user/all

Zastosuj formatowanie stylistyczne ☒

```
[
  {
    "id": 1,
    "name": "Igor",
    "surname": null,
    "address": "Test",
    "city": "Będzin",
    "description": null,
    "role": null
  },
  {
    "id": 3,
    "name": "Tomek",
    "surname": "Mak",
    "address": "",
    "city": "",
    "description": "",
    "role": "ADMIN"
  }
]
```