

Projekt

Reserve The Weather

Anna Jochymczyk

Monika Halek

Wiktoria Kowalska

Karol Kubek

Struktura plików back-endu (Java)

com.ioproject.reservetheweather/

├── api/

| ├── UserController → odpowiada za zarządzanie użytkownikami

| ├── EventController → zarządza zajęciami/wydarzeniami

| ├── WeatherController → zarządza danymi pogodowymi

├── config/

| ├── SecurityConfig → Konfiguruje zabezpieczenia aplikacji

| ├── WeatherApiConfig → Konfiguracja związana z OpenWeatherMap

API

├── model/

| ├── User → Reprezentuje użytkowników w systemie

| ├── AppUser → Zawiera dane służące do autentykacji użytkowników

| ├── Role → Reprezentuje role użytkowników w aplikacji (User, Admin)

| ├── Event → Reprezentuje zajęcia/wydarzenia

| ├── WeatherData → Reprezentuje dane pogodowe

├── repository/

| ├── EventRepository → Umożliwia dostęp do bazy zajęć

| ├── UserRepository → Umożliwia dostęp do bazy użytkowników

├── service/

| ├── EventService → Dostarcza metod do obsługi zajęć

| ├── UserService → Dostarcza metod związanych z

użytkownikami

| ├── WeatherService → Dostarcza metod pobierających dane pogodowe

| ├── AppUserService → Obsługuje dane związane z uwierzytelnianiem

ReservetheweatherApplication → Uruchamia aplikację SpringBoot

Metody

1. Widok

a) Użytkownik (UserService)

Metoda	Opis	Zwraca
getUsers()	zwraca wszystkich użytkowników z bazy	userRepository.findAll()
addNewUser(User user)	dodaje do bazy userRepository użytkownika user	true - jeśli udało się dodać false - jeśli użytkownik zawiera niepoprawne dane
deleteUser(Long userID)	usuwa użytkownika o podanym ID	true - jeśli usunięto rzuca wyjątek, jeśli nie znaleziono użytkownika w bazie
updateUserMail(Long userID, String email)	zmienia e-mail użytkownika o podanym ID	rzuca wyjątek, jeśli użytkownik o podanym ID nie istnieje lub podany e-mail jest zajęty
updateUserName(Long userID, String name)	zmienia nazwę użytkownika o podanym ID	rzuca wyjątek, jeśli użytkownik o podanym ID nie istnieje
showMyEvents(User user)	wyświetla zajęcia, na które jest zapisany dany użytkownik	listę zajęć
joinEvent(Long eventID, User user)	zapisuje użytkownika na zajęcia o podanym ID	true - jeśli udało się zapisać, false - w przeciwnym przypadku
resign(Long eventID, User user)	usuwa dane zajęcia z zajęć, na które jest zapisane użytkownik	true - jeśli udało się usunąć, false - w przeciwnym przypadku

b) Zajęcia (EventService)

Metoda	Opis	Zwraca
getEvents()	zwraca wszystkie dostępne zajęcia	eventRepository.findAll()
addEvent(Event event)	zapisuje zajęcia do bazy	true - jeśli udało się dodać, false - w przeciwnym przypadku
addPerson(Long eventID, User user)	dodaje użytkownika do listy użytkowników zapisanych na zajęcia	true - jeśli dodano, rzuca wyjątek jeśli nie znaleziono użytkownika w bazie
showDescription(Long eventID)	wyświetla szczegółowy opis danych zajęć	rzuca wyjątek, jeśli nie znaleziono podanych zajęć w bazie

checkWeather(Long eventID)	porównuje pogodę z pogodą dopuszczalną dla danych zajęć	funkcję przekazującą komunikat dla użytkownika
badWeatherInform(User user, Event event)	Informuje użytkownika o przewidywanej złej pogodzie na zajęcia i o możliwości rezygnacji, otrzymaniu zniżki lub o wzięciu udziału w zajęciach w innym terminie.	komunikat
resign(Long eventID, User user)	usuwa użytkownika z listy użytkowników zapisanych na dane zajęcia	true - jeśli usunięto rzuca wyjątek, jeśli użytkownik o podanym id nie jest zapisany na dane zajęcia
discount(Long eventID, User user)	jeśli użytkownik bierze udział w zajęciach pomimo złej pogody	true - jeśli przyznano zniżkę, false - w przeciwnym przypadku
reschedule(Long eventID, User user, String date1)	zapisuje użytkownika na zajęcia w innym terminie	true - jeśli zapisano, false - w przeciwnym przypadku

c) Dane pogodowe (WeatherService)

Metoda	Opis	Zwraca
checkWeather(String date, String city)	sprawdza prognozę pogody w danym miejscu	weatherData

d) Uwierzytelnianie (AppUserService implements
org.springframework.security.core.userdetails.UserDetailsService)

Metoda	Opis	Zwraca
loadUserByUsername(String username)	szuka w bazie danego użytkownika	danego użytkownika, rzuca wyjątek jeśli nie znaleziono go w bazie

2. Kontroler

a) Użytkownik (UserController)

Metoda	Wywołanie metody Service / opis	Ścieżka
saveUser(User user)	userService.addNewUser(user)	POST: "/api/register"
login()	logowanie użytkownika: sprawdzenie, czy użytkownik o podanych danych znajduje się w bazie	POST: "/login"
getAllUsers()	userService getUsers()	GET: "/api/users/all" wymagana autoryzacja: Admin
getMyDetails()	przekierowanie do strony głównej danego użytkownika	GET: "/api/users/mypage" wymagana autoryzacja: Admin lub User
showMyEvents()	userService.showMyEvents(Long userID)	GET: "/api/events/myevents" wymagana autoryzacja: Admin lub User
kontakt()	przekierowanie do strony z danymi, pod którymi można się kontaktować z obsługą w przypadku błędów aplikacji/problemów	GET: "/api/kontakt"

b) Zajęcia (EventController)

Metoda	Wywołanie metody Service	Ścieżka
getAllEvents()	eventService.getEvents()	"/api/events/all"
showEventDescription(Long eventID)	eventService.showDescriptio n(Long eventID)	"/api/events/description"
signUpForEvent(Long eventID)	eventService.addPerson(eve ntID, user) userService.joinEvent(eventI D, user)	"/api/events/signup" wymagana autoryzacja: User lub Admin
resignEvent(Long eventID)	eventService.resign(eventID, user) userService.resign(eventID,	"/api/user/myevents/cancel" wymagana autoryzacja: User lub Admin

	user)	
discountEvent(Long eventId)	eventService.discount(eventId, user)	"/api/user/myevents/discount" wymagana autoryzacja: User lub Admin
rescheduleEvent(String date1)	eventService.reschedule(date1)	"/api/user/myevents/reschedule" wymagana autoryzacja: User lub Admin

c) Dane pogodowe (WeatherController)

Metoda	Wywołanie metody Service	Ścieżka
checkWeather()	weatherService.checkWeather(String date, String city)	"/api/checkweather"

3. Model

a) Użytkownik (User)

Atrybut	Typ
id	Long
name	String
mail	String
phoneNumber	long
password	String
roles	String
myEvents	List<Event>

b) Użytkownik aplikacji (AppUser implements org.springframework.security.core.userdetails.UserDetails)

Atrybut	Typ
email	String
password	String
roles	List<GrantedAuthority>

accountExpired	boolean
accountLocked	boolean
credentialsExpired	boolean
accountDisabled	boolean

c) Rola użytkownika (Role implements
org.springframework.security.core.GrantedAuthority)

Atrybut	Typ
id	Long
authority	String

d) Zajęcia (Event)

Atrybut	Typ
id	Long
name	String
place	String
price	int
time	LocalDateTime
duration	int
maxUsers	int
signedUsers	int
description	String
users	List<User>
minTemperature	int
maxTemperature	int
badWeather	boolean

e) Dane pogodowe (WeatherData)

Atrybut	Typ
coord	Coordinates [klasa: atrybuty: lon (double), lat (double)]
main	MainWeatherData [klasa: atrybuty: temp (double), feels_like (double), temp_min (double), temp_max (double), pressure (int), humidity (int)]
weather	WeatherDescription [klasa: atrybuty: id (int), main (String), description (String)]
visibility	int
wind	Wind [klasa: atrybuty: speed (double), deg (int), gust(double)]
sys	Sys [klasa: atrybuty: type (int), id (int), country (String), sunrise (long), sunset (long)]
timezone	int
id	int
name	String

Struktura Danych

Diagram ER

