

SOPKOM

System Obsługi Przewoźnika Komunikacji Miejskiej

Skład zespołu:

Artur Kuśka

Bartłomiej Kuziel

Julian Koterba

Michał Lach

Tomasz Goczoł

Tomasz Seroczyński

Spis Treści

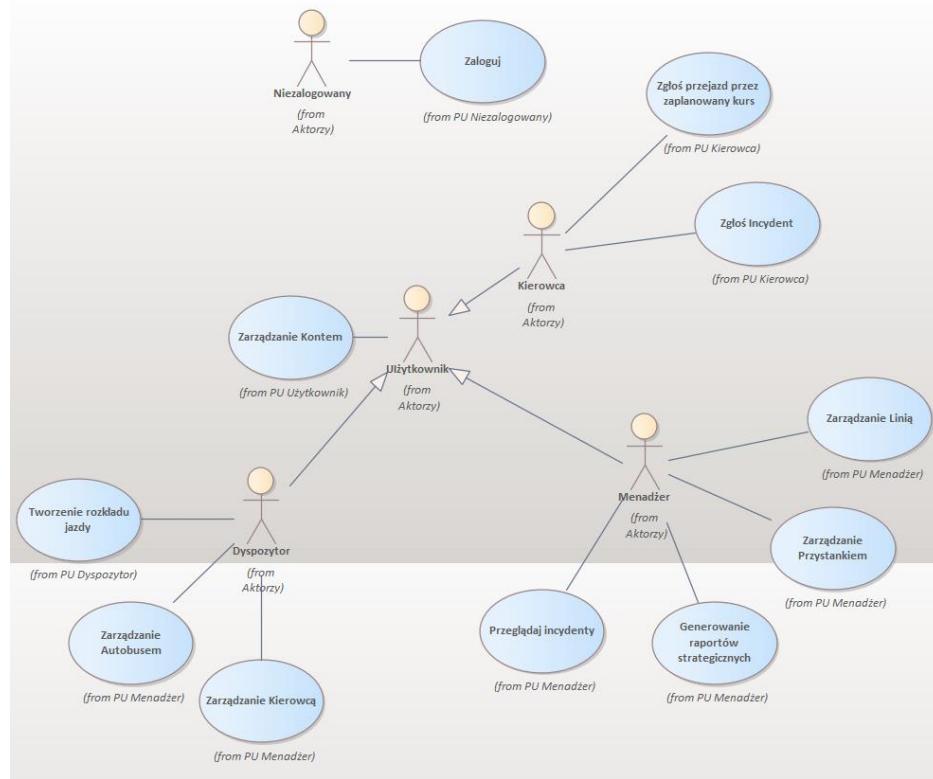
Spis Treści	2
Treść zadania	2
Analiza diagramów	3
Diagram przypadków użycia.....	3
Diagram klas.....	4
Opis architektury i użytego stosu technologicznego.....	7
Specyfikacja zewnętrzna	7
Specyfikacja wewnętrzna.....	42
Wnioski.....	8
Realizacja celów projektu.....	9
Napotkane problemy	9
Sugestie dotyczące dalszego rozwoju	9

Treść zadania

Projekt SOPKOM - System Obsługi Przewoźnika Komunikacji Miejskiej obejmuje zaprojektowanie i zrealizowanie systemu do zarządzania rozkładami jazdy, zarządzania flotą, naprawami/przeglądam, ewidencją kursów przypisanych do kierowców oraz generowaniem raportów operacyjnych i strategicznych dotyczących spalania, awaryjności, rentowności poszczególnych linii i spalania kierowców.

Analiza diagramów

Diagram przypadków użycia



Rys. 1. Diagram przypadków użycia

Diagram przypadków użycia przedstawia interakcje pomiędzy różnymi aktorami a systemem, ukazując funkcje dostępne dla poszczególnych ról użytkowników. W projekcie SOPKOM zidentyfikowano następujących aktorów i przypadki użycia:

- **Niezalogowany**
 - **Przypadki użycia:**
 - Zaloguj: Użytkownik niezalogowany może się zalogować do systemu, aby uzyskać dostęp do swoich funkcji.
- **Użytkownik**
 - **Przypadki użycia:**
 - Zarządzanie Kontem: Użytkownik zalogowany może zarządzać swoim kontem, co obejmuje aktualizację danych osobowych, zmianę hasła itp.
- **Kierowca**
 - **Przypadki użycia:**
 - Zgłoś przejazd przez zaplanowany kurs: Kierowca może zgłaszać realizację zaplanowanego kursu.
 - Zgłoś incident: Kierowca może zgłaszać incydenty, które miały miejsce podczas kursu.

- **Dyspozytor**

- **Przypadki użycia:**

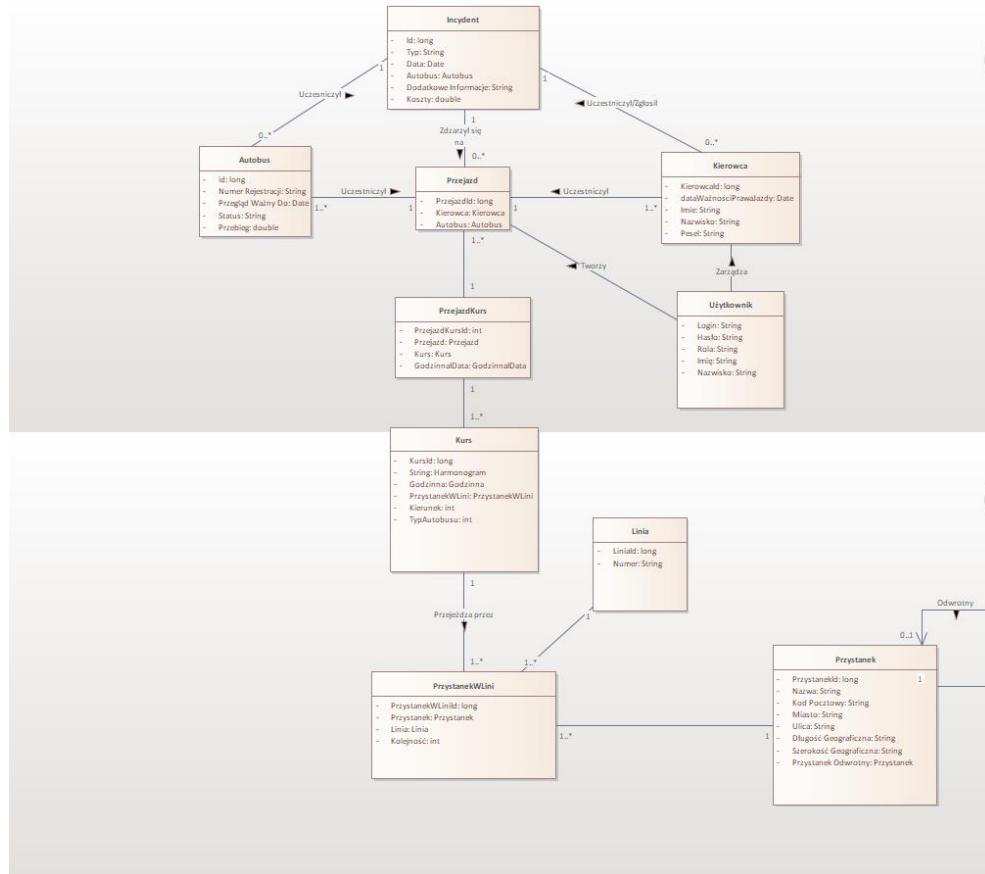
- Tworzenie rozkładu jazdy: Dyspozytor jest odpowiedzialny za tworzenie i aktualizowanie rozkładów jazdy.
 - Zarządzanie Autobusem: Dyspozytor zarządza flotą autobusów, w tym ich stanem technicznym i dostępnością.

- **Menadżer**

- **Przypadki użycia:**

- Zarządzanie linią: Menadżer może zarządzać trasami linii autobusowych.
 - Zarządzanie przystankiem: Menadżer ma możliwość zarządzania przystankami autobusowymi.
 - Zarządzanie kierowcą: Menadżer zarządza kierowcami, przypisując im kursy oraz monitorując ich pracę.
 - Przeglądaj incydenty: Menadżer może przeglądać zgłoszone incydenty.
 - Generowanie raportów strategicznych: Menadżer generuje raporty strategiczne, takie jak rentowność poszczególnych linii oraz analiza spalanja przez kierowców.

Diagram klas



Rys. 2. Diagram klas

Diagram klas jest kluczowym elementem modelowania systemu, który pokazuje strukturę danych i relacje między nimi. Poniżej znajdują się główne klasy oraz ich atrybuty i relacje:

- **Incydent**
 - **Atrybuty:**
 - Id: unikalny identyfikator incydentu (typ: long).
 - Typ: rodzaj incydentu (typ: String).
 - Data: data wystąpienia incydentu (typ: Date).
 - Autobus: odniesienie do obiektu klasy Autobus, który brał udział w incydencie.
 - Dodatkowe Informacje: dodatkowe szczegóły dotyczące incydentu (typ: String).
 - Koszty: koszty związane z incydentem (typ: double).
 - **Relacje:**
 - Związany z jednym obiektem klasy Autobus, który uczestniczył w incydencie.
 - Jeden incydent może być powiązany z wieloma obiektami klasy Przejazd.
- **Autobus**
 - **Atrybuty:**
 - Id: unikalny identyfikator autobusu (typ: long).
 - Numer Rejestracji: numer rejestracyjny autobusu (typ: String).
 - Przegląd Ważny Do: data ważności przeglądu technicznego (typ: Date).
 - Status: status autobusu (typ: String).
 - Przebieg: całkowity przebieg autobusu (typ: double).
 - **Relacje:**
 - Jeden autobus może uczestniczyć w wielu incydentach.
 - Jeden autobus może być przypisany do wielu przejazdów.
- **Przejazd**
 - **Atrybuty:**
 - PrzejazdId: unikalny identyfikator przejazdu (typ: long).
 - Kierowca: odniesienie do obiektu klasy Kierowca, który realizuje przejazd.
 - Autobus: odniesienie do obiektu klasy Autobus, który jest używany podczas przejazdu.
 - **Relacje:**
 - Każdy przejazd jest realizowany przez jednego kierowcę.
 - Każdy przejazd jest realizowany jednym autobusem.
- **Kierowca**
 - **Atrybuty:**
 - KierowcaId: unikalny identyfikator kierowcy (typ: long).
 - DataWaznosciPrawaJazdy: data ważności prawa jazdy (typ: Date).
 - Imię: imię kierowcy (typ: String).
 - Nazwisko: nazwisko kierowcy (typ: String).
 - Pesel: numer PESEL kierowcy (typ: String).
 - **Relacje:**
 - Kierowca może realizować wiele przejazdów.
- **Przystanek**
 - **Atrybuty:**

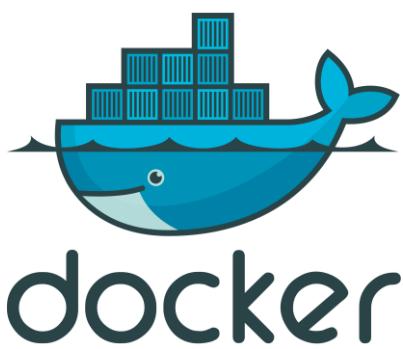
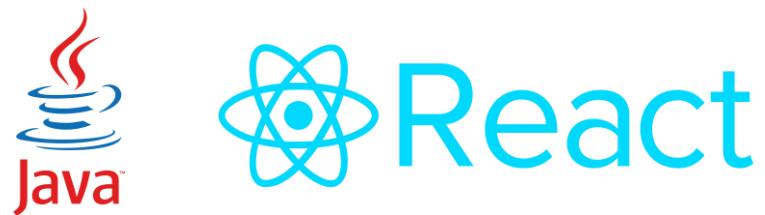
- PrzystanekId: unikalny identyfikator przystanku (typ: long).
 - Nazwa: nazwa przystanku (typ: String).
 - Kod Pocztowy: kod pocztowy przystanku (typ: String).
 - Miasto: miasto, w którym znajduje się przystanek (typ: String).
 - Ulica: ulica, na której znajduje się przystanek (typ: String).
 - Długość Geograficzna: długość geograficzna przystanku (typ: String).
 - Szerokość Geograficzna: szerokość geograficzna przystanku (typ: String).
 - PrzystanekOdwrotny: odniesienie do przystanku odwrotnego (typ: Przystanek).
- **Relacje:**
 - Każdy przystanek może być powiązany z wieloma przystankami w linii.
- **Linia**
 - **Atrybuty:**
 - Liniald: unikalny identyfikator linii (typ: long).
 - Numer: numer linii (typ: String).
 - **Relacje:**
 - Każda linia składa się z wielu przystanków w linii.
- **PrzystanekWLini**
 - **Atrybuty:**
 - PrzystanekWLiniId: unikalny identyfikator przystanku w linii (typ: long).
 - Przystanek: odniesienie do obiektu klasy Przystanek.
 - Linia: odniesienie do obiektu klasy Linia.
 - Kolejność: kolejność przystanku w linii (typ: int).
 - **Relacje:**
 - Każda linia składa się z wielu przystanków w linii, które są uporządkowane według kolejności.
- **PrzejazdKurs**
 - **Atrybuty:**
 - PrzejazdKursId: unikalny identyfikator przejazdu kursu (typ: int).
 - Przejazd: odniesienie do obiektu klasy Przejazd.
 - Kurs: odniesienie do obiektu klasy Kurs.
 - GodzinaData: godzina i data przejazdu (typ: GodzinaData).
 - **Relacje:**
 - Każdy przejazd kursu jest powiązany z jednym przejazdem i jednym kursem.
- **Kurs**
 - **Atrybuty:**
 - KursId: unikalny identyfikator kursu (typ: long).
 - Harmogram: harmonogram kursu (typ: String).
 - Godzina: godzina kursu (typ: Godzina).
 - PrzystanekWLIn: odniesienie do przystanku w linii (typ: PrzystanekWLIn).
 - **Relacje:**
 - Każdy kurs obejmuje wiele przejazdów kursu.

Opis architektury i użytego stosu technologicznego

Stos technologiczny

Projekt SOPKOM został zrealizowany z użyciem następujących technologii:

- **Spring**: Framework używany do tworzenia aplikacji w języku Java.
- **React**: Biblioteka JavaScript do tworzenia interfejsów użytkownika.
- **Docker**: Narzędzie do konteneryzacji aplikacji, pozwalające na uruchamianie aplikacji w izolowanych środowiskach.
- **Java**: Użyty język programowania.
- **Microsoft SQL Server**: System zarządzania relacyjnymi bazami danych, używany do przechowywania danych aplikacji.
- **JPA (Java Persistence API)**: Specyfikacja Java umożliwiająca zarządzanie trwałością danych w aplikacjach Java.



Architektura

System SOPKOM został zaprojektowany jako aplikacja wielowarstwowa, składająca się z następujących warstw:

Warstwa prezentacji (front-end):

React: Biblioteka JavaScript używana do tworzenia dynamicznych i responsywnych interfejsów użytkownika. React umożliwia tworzenie komponentów, które mogą być wielokrotnie wykorzystywane w różnych częściach aplikacji, co przyspiesza rozwój i utrzymanie kodu.

HTML/CSS: Podstawowe technologie webowe używane do strukturyzacji i stylizacji stron internetowych.

Warstwa logiki biznesowej (back-end):

Spring Framework: Framework dla języka Java, który ułatwia tworzenie aplikacji, zapewniając wsparcie dla zarządzania zależnościami, bezpieczeństwa, transakcji, oraz integracji z różnymi bazami danych. W projekcie SOPKOM użyto kilku modułów Spring:

Spring Boot: Narzędzie do szybkiego tworzenia aplikacji opartych na Spring, które minimalizuje konfigurację i upraszcza proces tworzenia aplikacji.

Spring Security: Moduł zapewniający mechanizmy uwierzytelniania i autoryzacji użytkowników.

Spring Data JPA: Abstrakcja nad JPA (Java Persistence API), która upraszcza dostęp do bazy danych, umożliwiając łatwe operacje CRUD (Create, Read, Update, Delete).

Warstwa bazy danych:

Microsoft SQL Server: Relacyjny system zarządzania bazą danych, który przechowuje wszystkie dane aplikacji, w tym informacje o użytkownikach, autobusach, przystankach, kursach, incydentach i raportach. Microsoft SQL Server zapewnia wydajność, skalowalność i bezpieczeństwo danych.

Konteneryzacja i zarządzanie środowiskiem:

Docker: Narzędzie do konteneryzacji, które umożliwia uruchamianie aplikacji w izolowanych środowiskach. Docker pozwala na łatwe wdrażanie i skalowanie aplikacji, minimalizując problemy związane z różnicami w konfiguracji środowisk developerskich i produkcyjnych.

Wnioski

Projekt SOPKOM - System Obsługi Przewoźnika Komunikacji Miejskiej - miał na celu stworzenie kompleksowego systemu zarządzania przewoźnikiem miejskim, obejmującego zarządzanie rozkładami jazdy, flotą pojazdów, naprawami i przeglądami, a także ewidencję kursów i generowanie raportów operacyjnych i strategicznych.

Realizacja celów projektu

Projekt został zrealizowany z sukcesem, spełniając wszystkie główne założenia:

- **Zarządzanie rozkładami jazdy:** Stworzono funkcjonalności pozwalające na tworzenie, edytowanie i przeglądanie rozkładów jazdy.
- **Zarządzanie flotą:** System umożliwia zarządzanie stanem technicznym autobusów, ich przeglądami oraz bieżącymi naprawami.
- **Ewidencja przejazdów:** Zaimplementowano funkcjonalności przypisywania kursów do kierowców oraz śledzenia realizacji przejazdu.
- **Generowanie raportów:** System generuje raporty operacyjne (np. spalanie, awaryjność).

Napotkane problemy

Podczas realizacji projektu zidentyfikowano kilka wyzwań:

- **Integracja różnych technologii:** Połączenie różnych technologii, takich jak Spring, React, Docker, i Microsoft SQL, wymagało dodatkowego nakładu pracy na zapewnienie płynnej integracji i komunikacji między komponentami.
- **Złożoność systemu:** Skala projektu i liczba funkcjonalności wymagały starannego planowania i podziału zadań, aby zapewnić terminowe ukończenie prac.
- **Zarządzanie danymi:** Implementacja mechanizmów zarządzania danymi, w tym zapewnienie spójności i integralności danych, była kluczowym elementem projektu.

Sugestie dotyczące dalszego rozwoju

W oparciu o doświadczenia z realizacji projektu, można rozważyć następujące sugestie dotyczące jego dalszego rozwoju:

- **Rozbudowa funkcjonalności raportowania:** Dodanie bardziej zaawansowanych opcji raportowania, takich jak interaktywne dashboardy, które umożliwią użytkownikom dynamiczne filtrowanie i analizowanie danych.
- **Optymalizacja wydajności:** Przeprowadzenie dalszych analiz wydajnościowych i optymalizacja kluczowych komponentów systemu, aby zapewnić jego skalowalność i szybkie działanie nawet przy dużej liczbie użytkowników.
- **Udoskonalenie interfejsu użytkownika:** Zwiększenie intuicyjności i ergonomii interfejsu użytkownika poprzez zastosowanie najlepszych praktyk UX/UI.
- **Automatyzacja procesów:** Wprowadzenie automatyzacji niektórych procesów zarządzania flotą, takich jak automatyczne przypomnienia o przeglądarkach technicznych czy naprawach.

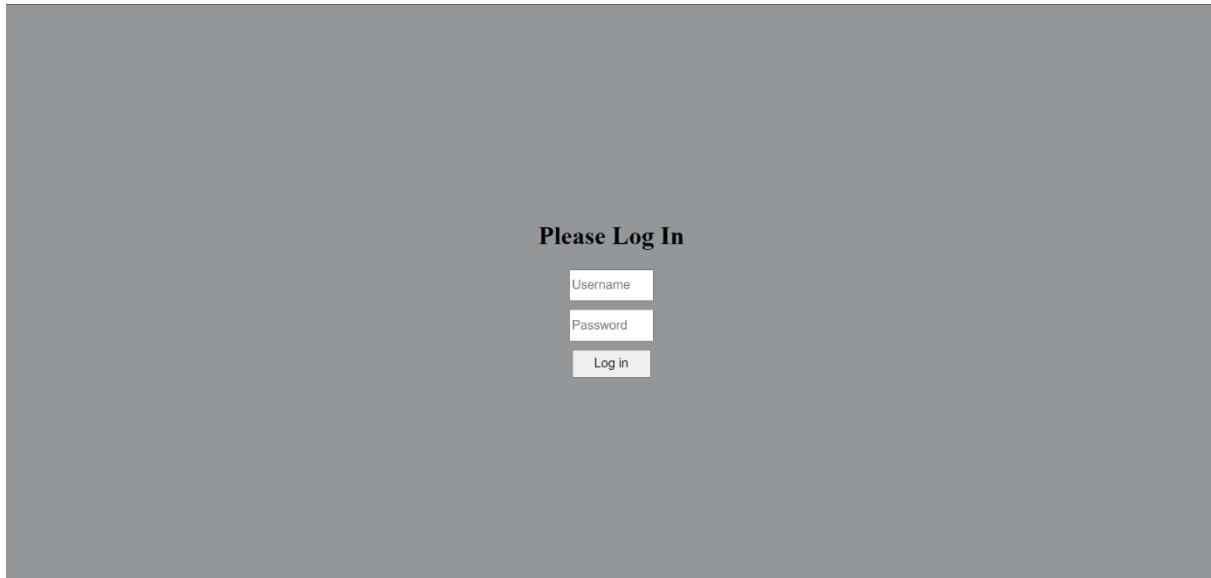
Specyfikacja zewnętrzna

Instrukcja obsługi

SOPKoM

System Obsługi Przewoźnika Komunikacji Miejskiej

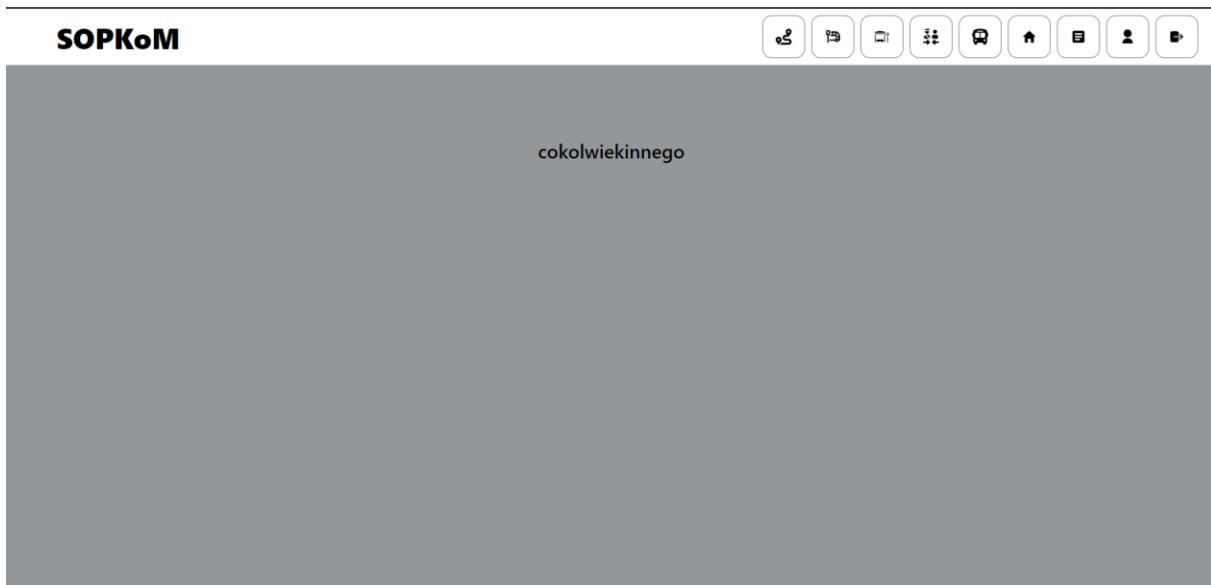
1. Logowanie



Ekran logowania

Po włączeniu aplikacji, użytkownik proszony jest o podanie nazwy użytkownika (Username) oraz hasła (Password). Po naciśnięciu przycisku **Log in** następuje zalogowanie.

2. Zarządzanie przystankami



Ekran strony domowej

Aby przejść do ekranu zarządzania przystankami należy kliknąć w odpowiednią ikonę znajdująca się w menu, na górze strony.



Ikona przystanku

The screenshot shows the SOPKoM application interface. At the top, there is a navigation bar with several icons. Below it, the title "SOPKoM" is displayed. On the right side of the header, there is a "Dodaj przystanek" button. The main area is titled "Lista przystanków:" and contains a table header with columns labeled "ID", "Nazwa", and "Adres".

Ekran zarządzania przystankami

Naciskając przycisk **Dodaj przystanek** użytkownik zostaje przekierowany do strony dodawania nowego przystanku.



Nazwa:

Miasto:

Ulica:

Kod pocztowy:

Długość geograficzna:

Szerokość geograficzna:

Dodaj przystanek

Powrót

Ekran dodawania nowego przystanku

W celu dodania nowego przystanku, należy wypełnić formularz wpisując informacje o przystanku, odpowiednio: nazwę, miasto, ulicę, kod pocztowy oraz długość i szerokość geograficzną.

Nazwa:	Gliwice Dworzec
Miasto:	Gliwice
Ulica:	Bohaterow Getta Wars
Kod pocztowy:	44-100
Długość geograficzna:	50.29
Szerokość geograficzna:	18.67

Dodaj przystanek

Powrót

Przykładowo wypełniony formularz

Zatwierdzenie następuje po naciśnięciu przycisku **Dodaj przystanek**. Użytkownik zostaje przekierowany do poprzedniego ekranu



Lista przystanków:

[Dodaj przystanek](#)

ID	Nazwa	Adres
1	Gliwice Dworzec	Gliwice - Bohaterow Getta Warszawskiego 12

Lista przystanków z nowo dodanym przystankiem

Po kliknięciu w nazwę danego przystanku mamy możliwość jego edytowania oraz usunięcia poprzez zmianę danych i kliknięcie przycisku **Zapisz zmiany** lub kliknięciu przycisku **Usuń**.



ID	Nazwa	Miasto	Ulica	Kod pocztowy	Długość geograficzna	Szerokość geograficzna	Przystanek odwrotny	
1	Gliwice Dworzec	Gliwice	Bohaterow Getta	44-100	50.29	18.67	Brak	Usuń

[Zapisz zmiany](#)[Pobierz Rozkład](#) [Powrót do przystanków](#) [Powrót do linii](#)

Ekran edycji przystanku

3. Zarządzanie liniami

Aby przejść do ekranu zarządzania liniami należy kliknąć w odpowiednią ikonę znajdująca się w menu, na górze strony.



Ikona linii

A screenshot of the SOPKoM application interface. At the top, there is a header with the logo "SOPKoM" and a navigation bar with several icons. Below the header, the main area is titled "Lista Lini Autobusowych:" (List of Bus Routes). A button labeled "Dodaj linię" (Add route) is visible. A table header is shown with columns "ID" and "Numer linii" (Route number).

Ekran zarządzania liniami

Naciskając przycisk **Dodaj linię** użytkownik zostaje przekierowany do strony dodawania nowej linii.



Numer Linii:

Kolejność jazdy Przystanek

Przystanek 1: Wybierz przystanek

Przystanek 2: Wybierz przystanek

Ekran dodawania nowej linii

W celu dodania nowej linii należy wpisać odpowiedni numer linii, a następnie z listy rozwijanej wybrać przystanki w odpowiedniej kolejności, aby dodać kolejny przystanek należy nacisnąć przycisk **Dodaj kolejny przystanek**.

Numer Linii: A4

Kolejność jazdy Przystanek

Przystanek 1: Gliwice Dworzec, Gliwice, Bohaterow Getta

Przystanek 2: Gliwice Jagielonska, Gliwice, Jagielonska 10

Przystanek 3: Wybierz przystanek

Przykład dodania kolejnego przystanku

Numer Linii: **A4**

Kolejność jazdy	Przystanek	
Przystanek 1:	Gliwice Dworzec, Gliwice, Bohaterow Getta	Usuń
Przystanek 2:	Gliwice Jagielonska, Gliwice, Jagielonska 10	Usuń
Przystanek 3:	Gliwice Zabrska, Gliwice, Zabrska	Usuń
Przystanek 4:	Gliwice Dabrowskiego, Gliwice , Dąbrowskie	Usuń
Przystanek 5:	Gliwice Zajezdnia, Gliwice, Chorzowska 12	Usuń

Dodaj kolejny przystanek

Dodaj linię

Powrót

Przykładowo wypełniony formularz

Po wypełnieniu formularza naciskając przycisk **Dodaj linię**, zostaje utworzona nowa linia oraz użytkownik zostaje przekierowany do ekranu zarządzania liniami.

SOPKoM

Lista Lini Autobusowych:

Dodaj linię

ID	Numer linii
1	A4

Lista linii autobusowych z nowo dodaną linią

Po kliknięciu w **Numer linii** mamy możliwość jej podglądu, a po naciśnięciu przycisku **Edytuj linię** mamy możliwość edytowania oraz usunięcia poprzez zmianę danych i kliknięcie przycisku **Aktualizuj linię** lub kliknięciu przycisku **Usuń linię**.

SOPKoM

Lista przystanków linii "A4":
Odwracalna

Kolejność	ID	Przystanek	Adres
1	1	Gliwice Dworzec	Gliwice, Bohaterow Getta Warszawskiego 12
2	2	Gliwice Jagielonska	Gliwice, Jagielonska 10
3	3	Gliwice Zabrska	Gliwice, Zabrska
4	4	Gliwice Dabrowskiego	Gliwice , Dąbrowskiego 5
5	5	Gliwice Zajezdnia	Gliwice, Chorzowska 12

Powrót **Edytuj linię**

Ekran podglądu przebiegu linii

SOPKoM

Numer Linii: **A4**

Kolejność jazdy	Przystanek
Przystanek 1:	Gliwice Dworzec, Gliwice, Bohaterow Getta
Przystanek 2:	Gliwice Jagielonska, Gliwice, Jagielonska 10
Przystanek 3:	Gliwice Zabrska, Gliwice, Zabrska
Przystanek 4:	Gliwice Dabrowskiego, Gliwice , Dąbrowskie
Przystanek 5:	Gliwice Zajezdnia, Gliwice, Chorzowska 12

Dodaj kolejny przystanek

Aktualizuj linię

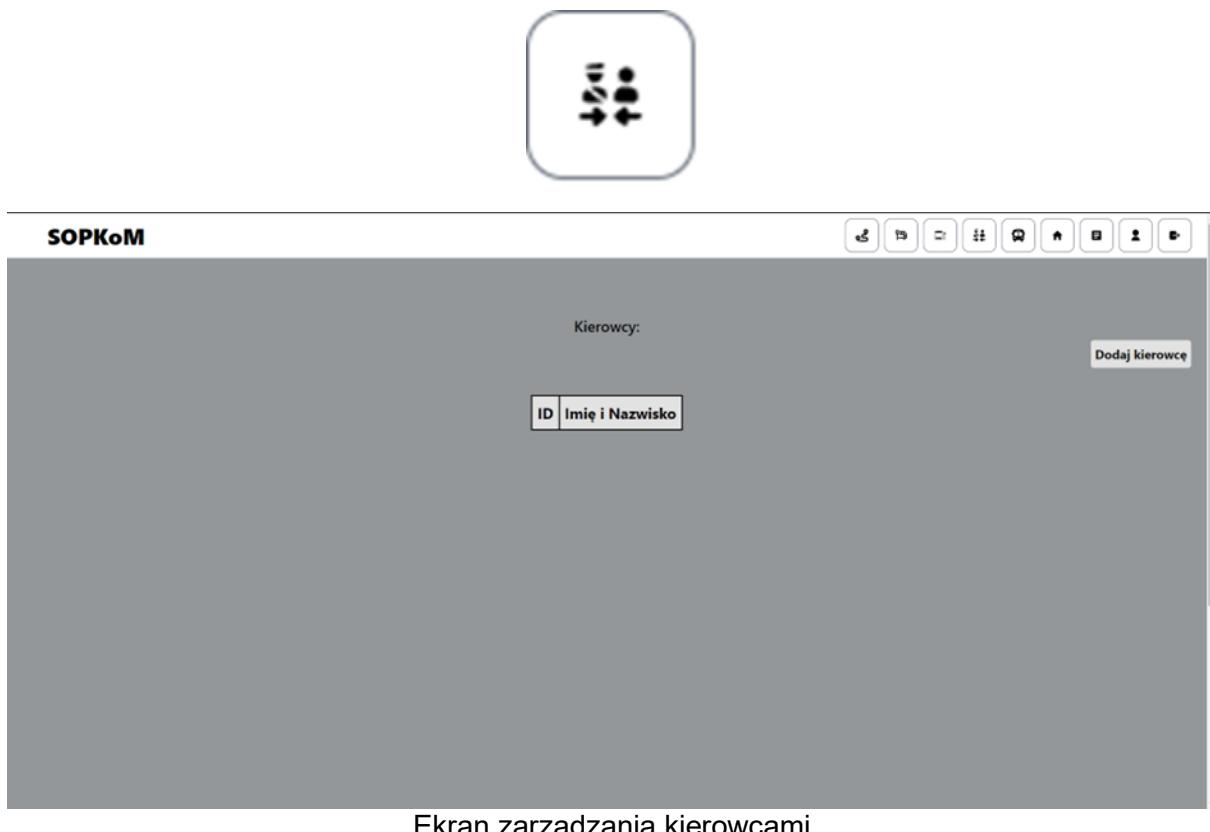
Usuń linię

Powrót

Ekran edycji linii

4. Zarządzanie kierowcami

Aby przejść do ekranu zarządzania kierowcami należy kliknąć w odpowiednią ikonę znajdująca się w menu, na górze strony.



Ekran zarządzania kierowcami

Naciskając przycisk **Dodaj kierowcę** użytkownik zostaje przekierowany do strony dodawania nowego kierowcy

The screenshot shows a software window titled 'SOPKoM'. At the top right are several small icons. The main area contains fields for entering driver information: 'Imię:' with a placeholder 'Jan', 'Nazwisko:' with a placeholder 'Kowalski', and 'PESEL:' with a placeholder '94827462531'. Below these is a date input field 'Prawo jazdy ważne do:' with the value '10.08.2024 00:00'. At the bottom are two buttons: 'Dodaj kierowcę' (Add driver) and 'Powrót' (Return).

Ekran dodawania nowego kierowcy.

W celu dodania nowego kierowcy należy wpisać odpowiednie dane takie jak imię, nazwisko, numer pesel, datę ważności prawa jazdy oraz kliknąć przycisk **Dodaj kierowcę**.

The screenshot shows the same software window as the previous one, but with sample data entered into the fields. The 'Imię:' field contains 'Jan', 'Nazwisko:' contains 'Kowalski', and 'PESEL:' contains '94827462531'. The date field shows '10.08.2024 00:00'. The 'Dodaj kierowcę' button is visible at the bottom.

Przykładowo wypełniony formularz.

Po dodaniu nowego kierowcy, klikając ponownie ikonę zarządzania kierowcami możemy zauważyć, iż został on prawidłowo dodany.

The screenshot shows a list of drivers under the heading "Kierowcy:". A button "Dodaj kierowcę" is visible. The table contains one row:

ID	Imię i Nazwisko
1	Jan Kowalski

Lista kierowców z nowo dodanym kierowcą.

Po kliknięciu w **Imię i Nazwisko** danego kierowcy mamy możliwość jego edytowania oraz usunięcia poprzez zmianę danych i kliknięcie przycisku **Zapisz zmiany** lub kliknięciu przycisku **Usuń**.

The screenshot shows the edit screen for a driver. The table fields are:

ID	Imię	Nazwisko	Wygaśnięcie prawa jazdy	Pesel	
1	Jan	Kowalski	2025-08-11	94827462531	Usuń

Buttons at the bottom include "Powrot" and "Zapisz zmiany".

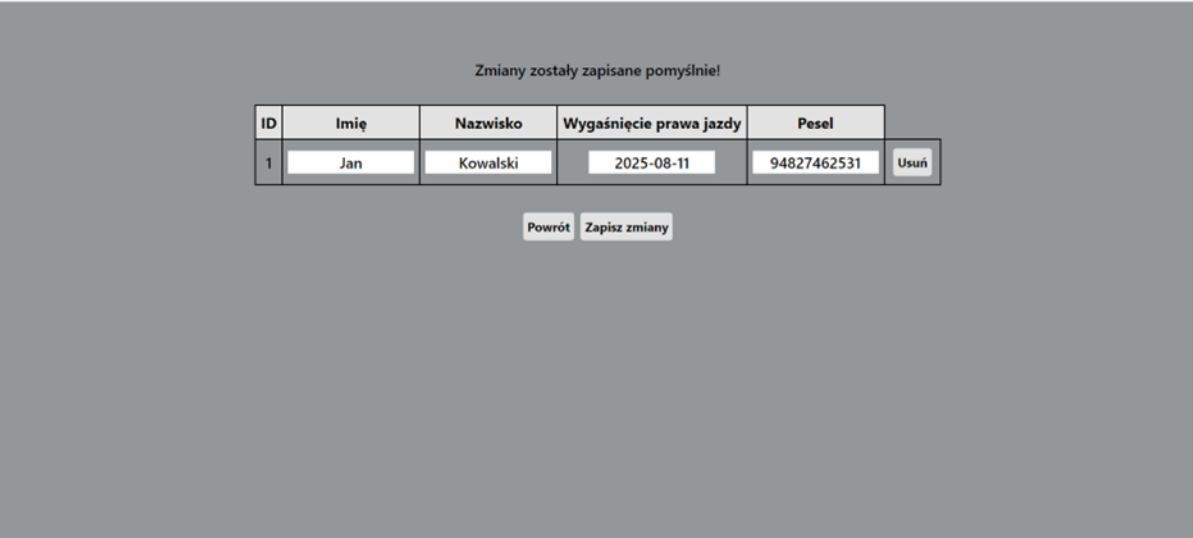
Ekran edycji kierowcy.

SOPKoM

Zmiany zostały zapisane pomyślnie!

ID	Imię	Nazwisko	Wygaśnięcie prawa jazdy	Pesel	Usuń
1	Jan	Kowalski	2025-08-11	94827462531	Usuń

[Powrót](#) [Zapisz zmiany](#)



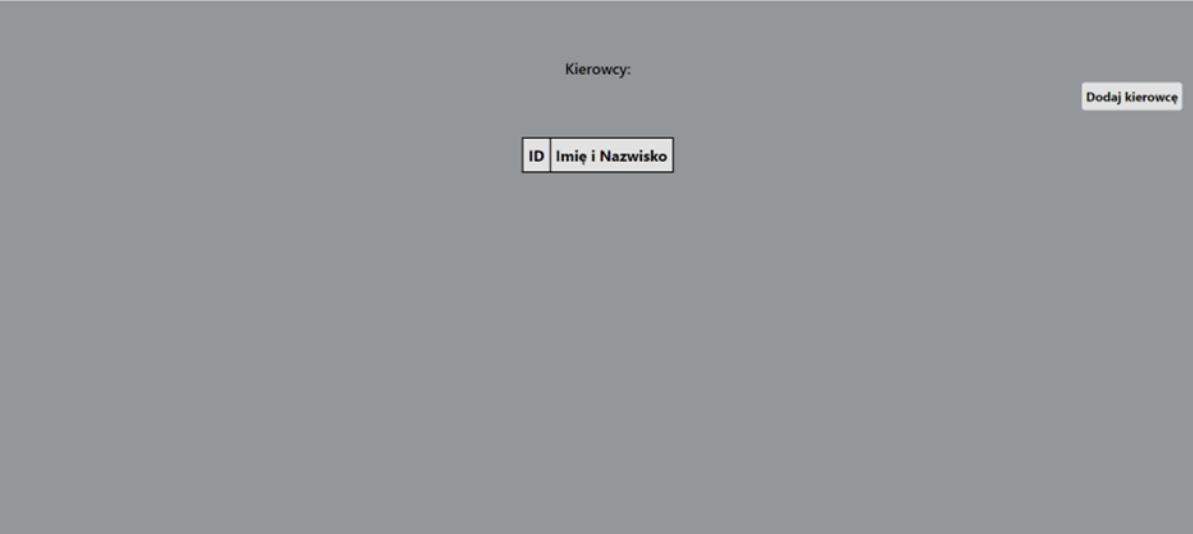
Zapisanie zmian danych kierowcy.

SOPKoM

Kierowcy:

[Dodaj kierowcę](#)

ID	Imię i Nazwisko
----	-----------------



Ekran po kliknięciu przycisku **Usuń**.

5. Zarządzanie incydentami

Aby przejść do ekranu zarządzania incydentami należy kliknąć w odpowiednią ikonę znajdująca się w menu, na górze strony.



SOPKoM

Incydenty:
Dodaj incydent

ID	Krótki opis	Koszt	Typ
----	-------------	-------	-----

Ekran zarządzania incydentami.

Aby dodać nowy incydent należy kliknąć w przycisk **Dodaj incydent**.

SOPKoM

Typ incydentu:	
Krótki opis:	
Data incydentu:	dd.mm.rrrr <input type="button" value="Wybierz"/>
Koszt:	
Kierowca:	Brak kierowcy
Autobus:	Brak autobusu
Przejazd:	Brak przejazdu
Dodatkowe informacje:	

Powrót **Dodaj incydent**

Ekran dodawania nowego incydentu.

W celu dodania nowego incydentu należy uzupełnić dane takie jak typ incydentu, krótki opis, data incydentu, koszt, kierowca, autobus, przejazd oraz dodatkowe informacje.

The screenshot shows a web-based application interface for managing incidents. At the top left is the logo 'SOPKoM'. A horizontal bar of icons is located at the top right. Below the header is a form for entering incident details:

Typ incydentu:	Kolizja
Krótki opis:	Kolizja autobusu
Data incydentu:	10.06.2024
Koszt:	1000
Kierowca:	2 - Jan Kowalski
Autobus:	1 - SGL27GT
Przejazd:	1 - M17 - 10.06.2024
Dodatkowe informacje:	Kolizja pomiędzy autobusem a samochodem osobowym.

At the bottom of the form are two buttons: 'Powrót' (Return) and 'Dodaj incydent' (Add incident).

Przykładowo wypełniony formularz.

Po kliknięciu przycisku **Dodaj incydent**, możemy go zobaczyć w bazie incydentów.

The screenshot shows a list of incidents in the database. At the top left is the 'SOPKoM' logo. A horizontal bar of icons is at the top right. The main area displays a table of incidents:

ID	Krótki opis	Koszt	Typ
1	Kolizja autobusu	1000	Kolizja

Widok bazy incydentów.

Po kliknięciu w dany incydent użytkownik ma możliwość edytowania incydentu.

SOPKoM

Zmiany zostały zapisane pomyślnie!

Typ incydentu:	Stłuczka
Krótki opis:	Stłuczka autobusu
Data incydentu:	09.06.2024
Koszt:	1000
Kierowca:	2 - Jan Kowalski
Autobus:	1 - SGL27GT
Przejazd:	1 - M17
Dodatkowe informacje:	Stłuczka pomiędzy autobusem a samochodem osobowym.

[Powrót](#) [Zapisz zmiany](#)

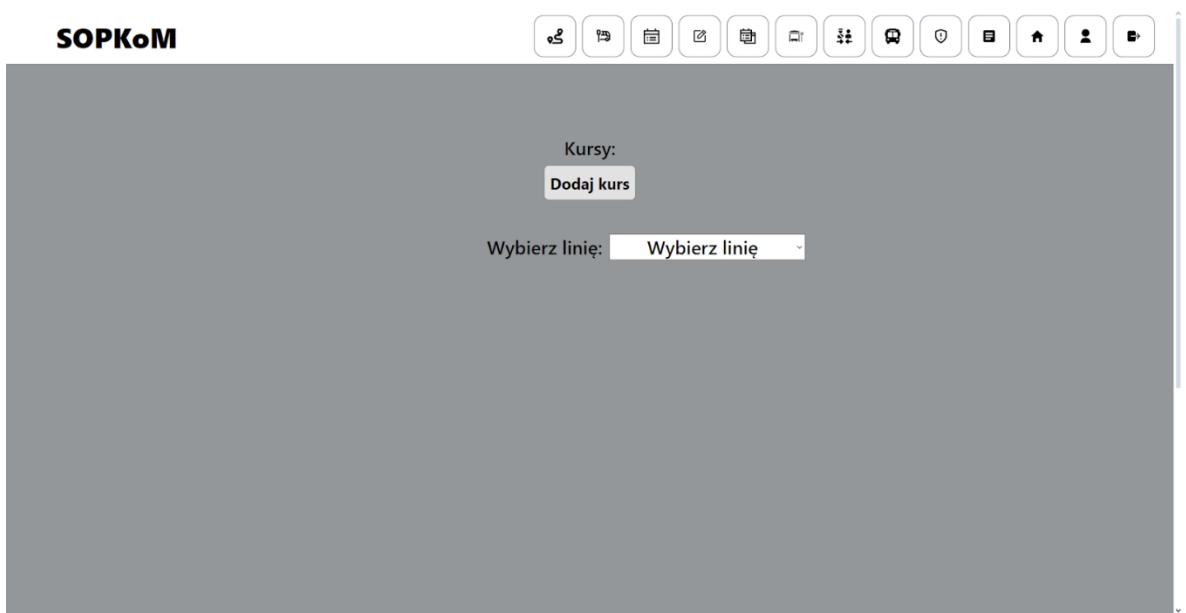
Ekran dokonania edycji incydentu.

6. Dodawanie nowego kursu

Aby przejść do ekranu zarządzania kursami należy kliknąć w odpowiednią ikonę znajdująca się w menu, na górze strony.



Ikona kursów



Ekran zarządzania kursami

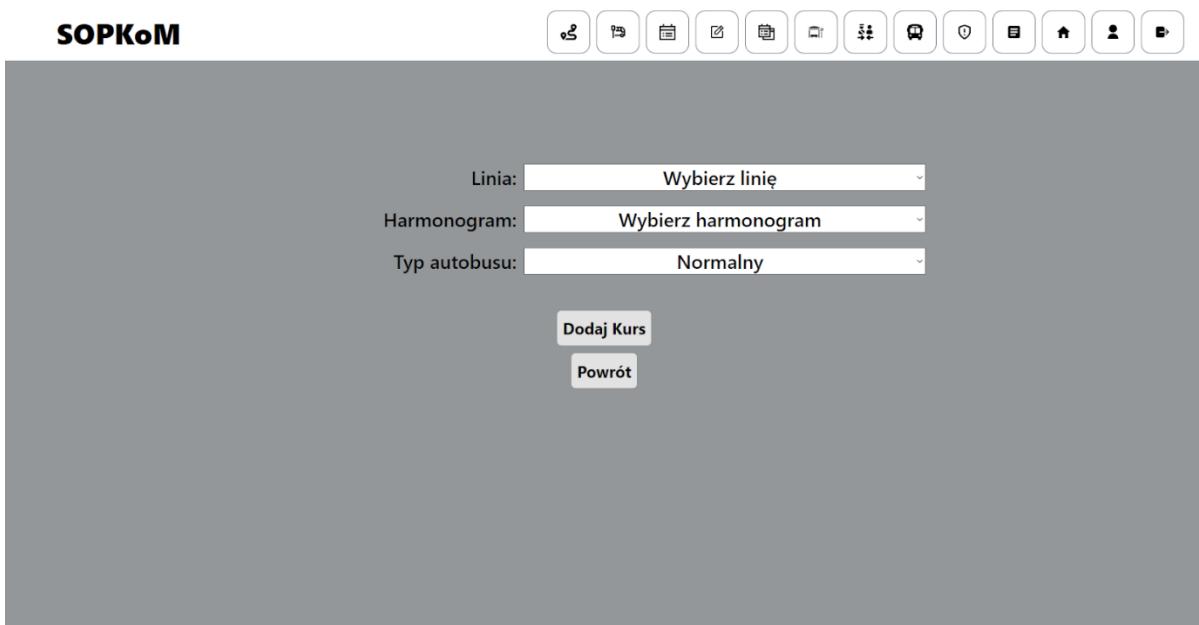
Do wyświetlenia kursów należy wybrać linię, a następnie wyświetla się informacje odnośnie kierunku, harmonogramu oraz tabela z godzinami kursu. Przy wielu kursach dla jednej linii są one wyświetlane niżej.

This screenshot shows the same software interface as the previous one, but with a dropdown menu 'Wybierz linię:' set to '378'. Below this, the text '6 Kierunek - Normalny' and 'Dni robocze' is displayed. A table titled 'Przystanek' and 'Godzina' lists three stops with their respective times: Centralny (18:03), Szkoła (18:23), and Poboczny ODW (18:40). At the bottom is a button labeled 'Edytuj kurs'.

Przystanek	Godzina
Centralny	18:03
Szkoła	18:23
Poboczny ODW	18:40

Ekran zarządzania kursami po wyborze linii

Naciskając przycisk **Dodaj kurs** użytkownik zostaje przekierowany do strony dodawania nowego kursu.



Ekran dodawania nowego kursu

W celu dodania nowego kursu należy wybrać z rozwijanej listy odpowiednią linię dla której chcemy utworzyć nowy kurs. Po wyborze linii ukazuje się lista przystanków z miejscem do wpisania godziny, a jeżeli jest to możliwe - pojawi się do wyboru kierunek (jest to możliwe tylko dla linii odwrotnych), odwrócenie kierunku odwraca kolejność tabeli z listą przystanków. Dodatkowo należy wybrać odpowiedni harmonogram oraz typ autobusu.

SOPKoM

Nazwa przystanku	Godzina
Szkoła	--:-- <input type="button" value="edit"/>
Poboczny	--:-- <input type="button" value="edit"/>

Ekran po wyborze linii z możliwością jej odwrócenia

SOPKoM

Nazwa przystanku	Godzina
Poboczny	19:43 <input type="button" value="edit"/>
Szkoła	20:09 <input type="button" value="edit"/>

Przykładowo wypełniony formularz

Po wypełnieniu formularza naciskając przycisk **Dodaj Kurs**, zostaje utworzony nowy kurs oraz użytkownik zostaje przekierowany do ekranu z kursami z wybraną linią dla której utworzono kurs.

SOPKoM

The screenshot shows a user interface for managing bus routes. At the top, there is a horizontal menu bar with various icons. Below it, the title "SOPKoM" is displayed. In the center, there is a section titled "Kursy:" (Routes) with a button labeled "Dodaj kurs" (Add route). A dropdown menu labeled "Wybierz linię:" (Select route) is open, showing the option "L 452". Below this, there is a table with two rows. The first row contains the header "Przystanek" (Stop) and "Godzina" (Time). The second row shows two stops: "Poboczny" at 19:43 and "Szkoła" at 20:09. At the bottom of this section is a button labeled "Edytuj kurs" (Edit route). The background of the main area is grey.

Lista kursów z nowym rekordem

7. Rozkład jazdy dla przystanku

Aby przejść do ekranu z rozkładem jazdy dla konkretnego przystanka należy kliknąć w odpowiednią ikonę znajdująca się w menu, na górze strony.



Ikona rozkładu jazdy



Rozkład jazdy dla przystanku:

Wybierz przystanek

09.06.2024



Ekran rozkładu jazdy

Do wyświetlenia rozkładu jazdy należy wybrać interesujący nas przystanek i dzień. Następnie wyświetli się tabelka z liniami przejeżdżającymi przez przystanek oraz godzinami planowanego przejazdu. Przy wielu kursach dla jednej linii są one wyświetlane obok siebie, a kursy niskopodłogowe są zaznaczone kolorem czerwonym.



Rozkład jazdy dla przystanku:

Wybierz przystanek

09.06.2024

Linia	Rozkład
L 452	09:10:00, 20:09:00
R 54	17:53:00



Ekran rozkładu po wyborze daty i przystanku



Rozkład jazdy dla przystanku:

Poboczny, Kurów, Jana Pawła II / 2

21.06.2024

Linia	Rozkład
236	18:53:00
L 452	09:23:00, 19:43:00

Ekran dla innego przystanku i innego dnia

8. Lista przejazdów (do zaplanowania)

Aby przejść do ekranu z listą przejazdów do zaplanowania należy kliknąć w odpowiednią ikonę znajdująca się w menu, na górze strony.



Ikona przejazdów do zaplanowania

The screenshot shows the SOPKoM application interface. At the top left is the logo 'SOPKoM'. To its right is a horizontal bar with various icons: a person, a map, a search bar, a gear, a bus, a car, a shield, a house, a person, a bus, and a car. Below this is a dark grey header bar with the text 'Lista Przejazdów:' and a dropdown menu labeled 'Filtrów Wybierz linię'. The main area is a table titled 'Lista Przejazdów:' with the following data:

Lp.	Kurs	Data	Status
1	236	18:39 10.06.2024	Do zaplanowania
2	R 54	17:23 10.06.2024	Do zaplanowania
3	G 17	20:26 10.06.2024	Do zaplanowania
4	L 452	19:39 10.06.2024	Do zaplanowania
5	378	17:33 10.06.2024	Do zaplanowania
6	236	18:39 11.06.2024	Do zaplanowania

Ekran listy przejazdów

Po kliknięciu, wyświetla się lista wszystkich niezaplanowanych przejazdów. Jeżeli niezaplanowany przejazd ma się odbyć w przeciągu 24h jest on pokolorowany na czerwono, jeżeli ma się odbyć do 5 dni na żółto. Można filtrować tabelę na podstawie linii.



Lista Przejazdów:

Filtry G 17

Lp.	Kurs	Data	Status
1	G 17	20:26 10.06.2024	Do zaplanowania
2	G 17	20:26 11.06.2024	Do zaplanowania
3	G 17	20:26 12.06.2024	Do zaplanowania
4	G 17	20:26 13.06.2024	Do zaplanowania
5	G 17	20:26 14.06.2024	Do zaplanowania
6	G 17	20:26 17.06.2024	Do zaplanowania
7	G 17	20:26 18.06.2024	Do zaplanowania

Ekran listy przejazdów do zaplanowania dla linii G17

Po kliknięciu w numer **Kursu** zostajemy przekierowani do ekranu z formularzem. Należy wybrać kierowcę oraz autobus, aby zaplanować wybrany przejazd. Po zaplanowaniu zostajemy przekierowani na zakładkę z listą zaplanowanych przejazdów



Zaplanuj Przejazd

Kurs: 236 19:09:00 ODW

Data przejazdu: 11.06.2024

Kierowca: Wybierz kierowcę

Autobus: Wybierz autobus

Zaplanuj**Powrót**

Ekran dodawania z wybranym przejazdem



Zaplanuj Przejazd

Kurs: 236 19:09:00 ODW

Data przejazdu: 11.06.2024

Kierowca: 2 - Błażej Pszczelewski

Autobus: 1 - SGL 023411

Zaplanuj

Powrót

Przykładowo wypełniony formularz

9. Dodawanie i lista przejazdów (zaplanowanych)

Aby przejść do ekranu z listą zaplanowanych przejazdów należy kliknąć w odpowiednią ikonę znajdująca się w menu, na górze strony.



Ikona przejazdów do zaplanowania

The screenshot shows the 'List of Trips' screen. At the top, there is a header with the 'SOPKoM' logo and a toolbar with several icons. Below the toolbar, the text 'Lista Przejazdów:' is displayed. On the right side of the screen, there is a button labeled 'Zaplanuj przejazd'. At the bottom, there are three dropdown menus: 'Filtry', 'Wybierz linię', and 'Wybierz status'. The main area contains a table with the following data:

Id.	Kurs	Data	Kierowca	Autobus	Status
1	236	18:39 3.06.2024	Marek Zaorski	SG 22341	Nie odbył się
2	236	18:39 4.06.2024	Błażej Pszczelewski	SGL 023411	Nie odbył się
3	R 54	17:23 2.06.2024	Marek Zaorski	SGL 023411	Nie odbył się
4	236	20:24 2.06.2024	Błażej Pszczelewski	SGL 023411	Nie odbył się
5	236	20:24 8.06.2024	Marek Zaorski	SGL 023411	Nie odbył się

Ekran zaplanowanych przejazdów

Po kliknięciu, wyświetla się lista wszystkich zaplanowanych przejazdów. Tabelę można filtrować na podstawie linii oraz statusu:

- Zakończony - przejazd się odbył i posiada dane dotyczące przejazdu
- Zaplanowany – przejazd dopiero się odbędzie
- Nie odbył się - przejazd się nie odbył i nie posiada danych przejazdu

Id.	Kurs	Data	Kierowca	Autobus	Status
10011	R 54	17:23 10.06.2024	Błażej Pszczelewski	SGL 023411	Zaplanowany
10013	R 54	17:23 10.06.2024	Marek Zaorski	SG 22341	Zaplanowany

Ekran z przefiltrowaną tabelą

Po kliknięciu w **Zaplanuj przejazd** zostajemy przekierowani do ekranu z formularzem. Na początku musimy wybrać sam kurs (są również ukazane kursy odwrotne z dopiskiem "ODW"). Następnie pojawia się lista dni w których może się odbyć dany przejazd. Po wyborze daty należy wybrać jeszcze kierowcę oraz autobus.

Zaplanuj Przejazd

Kurs:

Ekran dodawania przejazdu

SOPKoM

The screenshot shows a software interface with a dark grey header bar containing the logo 'SOPKoM' and several small, light-colored icons representing different functions or modules. Below the header is a large, semi-transparent light grey overlay covering most of the screen. At the top of this overlay, the text 'Zaplanuj Przejazd' is centered. Below it, there are four input fields with dropdown menus: 'Kurs:' set to '236 19:09:00 ODW', 'Data przejazdu:' set to '11.06.2024', 'Kierowca:' set to '1 - Marek Zaorski', and 'Autobus:' set to '2 - SG 22341'. Underneath these fields are two buttons: 'Dodaj przejazd' (Add trip) and 'Powrót' (Return). The rest of the screen below the overlay is completely blank and white.

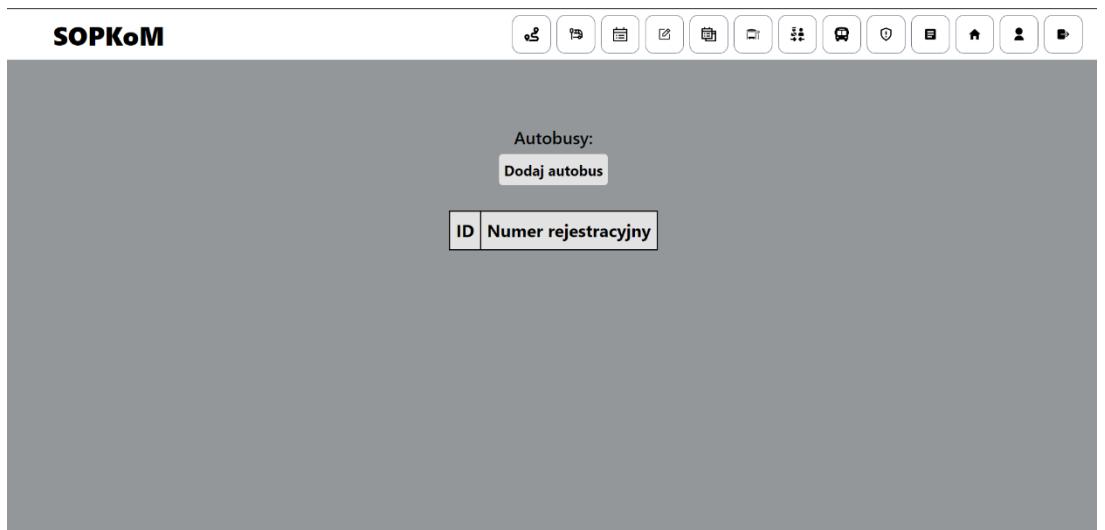
Przykładowo wypełniony formularz

10. Zarządzanie Autobusami

Aby przejść do ekranu zarządzania autobusami należy kliknąć w odpowiednią ikonę znajdująca się w menu, na górze strony.



Ikona autobusu



Ekran zarządzania autobusami

Naciskając przycisk **Dodaj autobus** użytkownik zostaje przekierowany do strony dodawania nowego autobusu.

The screenshot shows a dark grey header bar with the text "SOPKoM" on the left and a series of small, light-grey circular icons on the right. Below the header is a large, solid dark grey rectangular area containing a form for adding a new bus. The form includes fields for "Numer rejestracyjny" (with a text input field), "Przegląd ważny do" (with a date input field showing "dd.mm.rrrr --::--"), "Czy jest sprawny" (with a checkbox), "Przebieg" (with a text input field), and two buttons at the bottom: "Dodaj autobus" and "Powrót".

Ekran dodawania nowego autobusu

W celu dodania nowego autobusu, należy wypełnić formularz wpisując jego numer rejestracyjny, datę ważności przeglądu, przebieg oraz zaznaczyć (lub nie) pole czy autobus jest sprawny.

SOPKoM

Numer rejestracyjny: SG 12345
Przegląd ważny do: 10.03.2025 00:00
Czy jest sprawny:
Przebieg: 54321
Dodaj autobus
Powrót

Przykładowo wypełniony formularz

Zatwierdzenie następuje po naciśnięciu przycisku **Dodaj autobus**. Użytkownik zostaje przekierowany do poprzedniego ekranu

ID	Numer rejestracyjny
2	SG 12345

Lista autobusów z nowo dodanym autobusem

Po kliknięciu w **Numer rejestracyjny** danego autobusu mamy możliwość jego edytowania oraz usunięcia poprzez zmianę danych i kliknięcie przycisku **Zapisz zmiany** lub kliknięciu przycisku **Usuń**.



ID	Numer rejestracyjny	Czy jest sprawny?	Ważność przeglądu	Przebieg	
2	SG 12345	<input checked="" type="checkbox"/>	2025-03-10	54321	Usuń

Powrót **Zapisz zmiany**

Ekran edycji autobusu

Specyfikacja wewnętrzna

Poniżej

SOPKoM

Wygenerowano przez Doxygen 1.9.3

1 Indeks przestrzeni nazw	1
1.1 Pakiety	1
2 Indeks hierarchiczny	3
2.1 Hierarchia klas	3
3 Indeks klas	7
3.1 Lista klas	7
4 Indeks plików	11
4.1 Lista plików	11
5 Dokumentacja przestrzeni nazw	17
5.1 Pakiet com.mklachl.sopkom	17
5.2 Pakiet com.mklachl.sopkom.config	17
5.3 Pakiet com.mklachl.sopkom.controller	18
5.4 Pakiet com.mklachl.sopkom.exceptions	18
5.5 Pakiet com.mklachl.sopkom.filter	18
5.6 Pakiet com.mklachl.sopkom.helpers	18
5.7 Pakiet com.mklachl.sopkom.model.dto	18
5.8 Pakiet com.mklachl.sopkom.model.dto.harmonogram	19
5.9 Pakiet com.mklachl.sopkom.model.dto.kurs	19
5.10 Pakiet com.mklachl.sopkom.model.dto.kurs.OUTDATED	19
5.11 Pakiet com.mklachl.sopkom.model.dto.linia	20
5.12 Pakiet com.mklachl.sopkom.model.dto.przejazd	20
5.13 Pakiet com.mklachl.sopkom.model.dto.rozklad	20
5.14 Pakiet com.mklachl.sopkom.model.entity	20
5.15 Pakiet com.mklachl.sopkom.repository	21
5.16 Pakiet com.mklachl.sopkom.security	21
5.17 Pakiet com.mklachl.sopkom.services	21
5.18 Pakiet com.mklachl.sopkom.services.impl	22
5.19 Pakiet com.mklachl.sopkom.topdf	22
6 Dokumentacja klas	23
6.1 Dokumentacja klasy com.mklachl.sopkom.model.dto.kurs.AbstractKursDto	23
6.1.1 Opis szczegółowy	24
6.1.2 Dokumentacja konstruktora i destruktora	24
6.1.2.1 AbstractKursDto() [1/2]	24
6.1.2.2 AbstractKursDto() [2/2]	24
6.1.3 Dokumentacja funkcji składowych	24
6.1.3.1 getHarmonogramId()	24
6.1.3.2 getKierunek()	25
6.1.3.3 getKursId()	25
6.1.3.4 getLinialId()	25

6.1.3.5 getTyp_autobusu()	25
6.1.3.6 setHarmonogramId()	25
6.1.3.7 setKierunek()	26
6.1.3.8 setKursId()	26
6.1.3.9 setLiniald()	26
6.1.3.10 setTyp_autobusu()	27
6.1.4 Dokumentacja atrybutów składowych	27
6.1.4.1 harmonogramId	27
6.1.4.2 kierunek	27
6.1.4.3 kursId	28
6.1.4.4 liniald	28
6.1.4.5 typ_autobusu	28
6.2 Dokumentacja klasy com.mklachl.sopkom.model.dto.kurs.AbstractPrzystanekWKursieDto	28
6.2.1 Opis szczegółowy	29
6.2.2 Dokumentacja konstruktora i destruktora	29
6.2.2.1 AbstractPrzystanekWKursieDto() [1/2]	29
6.2.2.2 AbstractPrzystanekWKursieDto() [2/2]	29
6.2.3 Dokumentacja funkcji składowych	29
6.2.3.1 getGodzinna()	30
6.2.3.2 getPrzystanekwKursield()	30
6.2.3.3 setGodzinna()	30
6.2.3.4 setPrzystanekwKursield()	30
6.2.4 Dokumentacja atrybutów składowych	31
6.2.4.1 godzinna	31
6.2.4.2 przystanekwKursield	31
6.3 Dokumentacja klasy com.mklachl.sopkom.model.entity.Autobus	31
6.3.1 Opis szczegółowy	32
6.3.2 Dokumentacja funkcji składowych	32
6.3.2.1 getAutbousId()	32
6.3.2.2 getNumerRejestracyjny()	32
6.3.2.3 getPrzebieg()	32
6.3.2.4 getPrzegladWaznyDo()	33
6.3.2.5 getStatus()	33
6.3.2.6 setAutbousId()	33
6.3.2.7 setNumerRejestracyjny()	33
6.3.2.8 setPrzebieg()	34
6.3.2.9 setPrzegladWaznyDo()	34
6.3.2.10 setStatus()	34
6.4 Dokumentacja klasy com.mklachl.sopkom.controller.AutobusController	35
6.4.1 Opis szczegółowy	35
6.4.2 Dokumentacja funkcji składowych	35
6.4.2.1 addAutobus()	35

6.4.2.2 deleteAutobus()	36
6.4.2.3 getAllBuses()	36
6.4.2.4 getAllBusesFilteredStatus()	36
6.4.2.5 getAutobus()	37
6.4.2.6 template()	37
6.4.2.7 test()	37
6.4.2.8 updateAutobus()	37
6.5 Dokumentacja klasy com.mklachl.sopkom.model.dto.AutobusDto	38
6.5.1 Opis szczegółowy	38
6.5.2 Dokumentacja konstruktora i destruktora	38
6.5.2.1 AutobusDto() [1/2]	39
6.5.2.2 AutobusDto() [2/2]	39
6.5.3 Dokumentacja funkcji składowych	40
6.5.3.1 getAutbousId()	40
6.5.3.2 getNumerRejestracyjny()	40
6.5.3.3 getPrzebieg()	40
6.5.3.4 getPrzegladWaznyDo()	41
6.5.3.5 getStatus()	41
6.5.3.6 setAutbousId()	41
6.5.3.7 setNumerRejestracyjny()	41
6.5.3.8 setPrzebieg()	42
6.5.3.9 setPrzegladWaznyDo()	42
6.5.3.10 setStatus()	42
6.5.3.11 toString()	43
6.6 Dokumentacja interfejsu com.mklachl.sopkom.repository.AutobusRepository	43
6.6.1 Opis szczegółowy	44
6.6.2 Dokumentacja funkcji składowych	44
6.6.2.1 findAll()	44
6.6.2.2 findAllByAvailable()	44
6.6.2.3 findAllByNumerRejestracyjnyLike()	44
6.6.2.4 findAllByPrzebieg()	45
6.6.2.5 findAllByPrzebiegGreaterThanOrEqual()	45
6.6.2.6 findAllByPrzebiegLessThan()	45
6.6.2.7 findAllByStatus()	47
6.6.2.8 findByNumerRejestracyjny()	47
6.6.2.9 findByPrzegladWaznyDoAfter()	47
6.6.2.10 findByPrzegladWaznyDoBefore()	49
6.6.2.11 findByStatusLike()	49
6.7 Dokumentacja interfejsu com.mklachl.sopkom.services.AutobusService	50
6.7.1 Opis szczegółowy	50
6.7.2 Dokumentacja funkcji składowych	50
6.7.2.1 findAutobusByAvailable()	50

6.7.2.2 findAutobusById()	51
6.7.2.3 findAutobusByNumerRejestracyjny()	51
6.7.2.4 findAutobusByPrzebieg()	51
6.7.2.5 findAutobusByPrzebiegGreaterThanOrEqual()	52
6.7.2.6 findAutobusByPrzebiegLessThanOrEqual()	52
6.7.2.7 findAutobusByPrzegladWaznyDoAfter()	53
6.7.2.8 findAutobusByPrzegladWaznyDoBefore()	53
6.7.2.9 findAutobusByStatus()	53
6.7.2.10 saveAutobus() [1/2]	54
6.7.2.11 saveAutobus() [2/2]	54
6.8 Dokumentacja klasy com.mklachl.sopkom.services.impl.AutobusServiceImpl	55
6.8.1 Opis szczegółowy	55
6.8.2 Dokumentacja konstruktora i destruktora	55
6.8.2.1 AutobusServiceImpl()	55
6.8.3 Dokumentacja funkcji składowych	55
6.8.3.1 findAutobusByAvailable()	55
6.8.3.2 findAutobusById()	56
6.8.3.3 findAutobusByNumerRejestracyjny()	56
6.8.3.4 findAutobusByPrzebieg()	57
6.8.3.5 findAutobusByPrzebiegGreaterThanOrEqual()	57
6.8.3.6 findAutobusByPrzebiegLessThanOrEqual()	58
6.8.3.7 findAutobusByPrzegladWaznyDoAfter()	58
6.8.3.8 findAutobusByPrzegladWaznyDoBefore()	58
6.8.3.9 findAutobusByStatus()	59
6.8.3.10 saveAutobus() [1/2]	59
6.8.3.11 saveAutobus() [2/2]	60
6.9 Dokumentacja klasy com.mklachl.sopkom.model.entity.Bilet	60
6.9.1 Opis szczegółowy	60
6.9.2 Dokumentacja funkcji składowych	61
6.9.2.1 getId()	61
6.9.2.2 getCena()	61
6.9.2.3 getNazwa()	61
6.9.2.4 getPrzejazdBilet()	62
6.9.2.5 setId()	62
6.9.2.6 setCena()	63
6.9.2.7 setNazwa()	63
6.9.2.8 setPrzejazdBilet()	63
6.10 Dokumentacja klasy com.mklachl.sopkom.model.dto.BiletDto	64
6.10.1 Opis szczegółowy	64
6.10.2 Dokumentacja konstruktora i destruktora	64
6.10.2.1 BiletDto() [1/2]	64
6.10.2.2 BiletDto() [2/2]	64

6.10.3 Dokumentacja funkcji składowych	65
6.10.3.1 getBiletId()	65
6.10.3.2 getCena()	65
6.10.3.3 getNazwa()	65
6.10.3.4 setBiletId()	65
6.10.3.5 setCena()	66
6.10.3.6 setNazwa()	66
6.11 Dokumentacja interfejsu com.mklachl.sopkom.repository.BiletRepository	66
6.11.1 Opis szczegółowy	67
6.11.2 Dokumentacja funkcji składowych	67
6.11.2.1 findAll()	67
6.11.2.2 findAllByCena()	67
6.11.2.3 findAllByNazwa()	68
6.12 Dokumentacja klasy com.mklachl.sopkom.security.CommandLineRunnerTest	68
6.12.1 Opis szczegółowy	69
6.12.2 Dokumentacja funkcji składowych	69
6.12.2.1 run()	69
6.12.3 Dokumentacja atrybutów składowych	69
6.12.3.1 ADMIN_LOGIN	69
6.12.3.2 ADMIN_NAME	69
6.12.3.3 ADMIN_PASS	70
6.12.3.4 MANAGER_LOGIN	70
6.12.3.5 MANAGER_NAME	70
6.12.3.6 MANAGER_PASS	70
6.12.3.7 userRepository	70
6.13 Dokumentacja klasy com.mklachl.sopkom.controller.Controller	70
6.13.1 Opis szczegółowy	71
6.13.2 Dokumentacja funkcji składowych	71
6.13.2.1 getString()	71
6.13.2.2 getString2()	71
6.14 Dokumentacja klasy com.mklachl.sopkom.security.CustomUserDetailsService	71
6.14.1 Opis szczegółowy	72
6.14.2 Dokumentacja konstruktora i destruktora	72
6.14.2.1 CustomUserDetailsService()	72
6.14.3 Dokumentacja funkcji składowych	72
6.14.3.1 loadUserByUsername()	72
6.15 Dokumentacja klasy com.mklachl.sopkom.model.dto.DateWrapperDto	72
6.15.1 Opis szczegółowy	73
6.15.2 Dokumentacja konstruktora i destruktora	73
6.15.2.1 DateWrapperDto() [1/2]	73
6.15.2.2 DateWrapperDto() [2/2]	73
6.15.3 Dokumentacja funkcji składowych	73

6.15.3.1 getDate()	73
6.15.3.2 setDate()	74
6.16 Dokumentacja klasy com.mklachl.sopkom.model.entity.Harmonogram	74
6.16.1 Opis szczegółowy	74
6.16.2 Dokumentacja funkcji składowych	75
6.16.2.1 getDodatkoweInf()	75
6.16.2.2 getHamonogramId()	75
6.16.2.3 getNazwa()	75
6.16.2.4 isCzw()	76
6.16.2.5 isNie()	76
6.16.2.6 isPia()	76
6.16.2.7 isPon()	76
6.16.2.8 isSob()	77
6.16.2.9 isSro()	77
6.16.2.10 isWto()	77
6.16.2.11 setCzw()	77
6.16.2.12 setDodatkoweInf()	78
6.16.2.13 setHamonogramId()	78
6.16.2.14 setNazwa()	78
6.16.2.15 setNie()	79
6.16.2.16 setPia()	79
6.16.2.17 setPon()	79
6.16.2.18 setSob()	80
6.16.2.19 setSro()	80
6.16.2.20 setWto()	80
6.17 Dokumentacja klasy com.mklachl.sopkom.controller.HarmonogramController	81
6.17.1 Opis szczegółowy	81
6.17.2 Dokumentacja funkcji składowych	81
6.17.2.1 createHamonogram()	81
6.17.2.2 deleteHamonogram()	82
6.17.2.3 getAllHamonograms()	83
6.17.2.4 getHamonogramById()	83
6.18 Dokumentacja klasy com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto	84
6.18.1 Opis szczegółowy	84
6.18.2 Dokumentacja konstruktora i destruktora	84
6.18.2.1 HarmonogramDto() [1/2]	84
6.18.2.2 HarmonogramDto() [2/2]	84
6.18.3 Dokumentacja funkcji składowych	85
6.18.3.1 getDodatkoweInf()	85
6.18.3.2 getHamonogramId()	85
6.18.3.3 getNazwa()	85
6.18.3.4 isCzw()	86

6.18.3.5 isNie()	86
6.18.3.6 isPia()	86
6.18.3.7 isPon()	86
6.18.3.8 isSob()	87
6.18.3.9 isSro()	87
6.18.3.10 isWto()	87
6.18.3.11 setCzw()	87
6.18.3.12 setDodatkowelnf()	88
6.18.3.13 setHarmonogramId()	88
6.18.3.14 setNazwa()	88
6.18.3.15 setNie()	89
6.18.3.16 setPia()	89
6.18.3.17 setPon()	89
6.18.3.18 setSob()	90
6.18.3.19 setSro()	90
6.18.3.20 setWto()	90
6.19 Dokumentacja interfejsu com.mklachl.sopkom.repository.HarmonogramRepository	91
6.19.1 Opis szczegółowy	91
6.19.2 Dokumentacja funkcji składowych	91
6.19.2.1 findAll()	91
6.19.2.2 findAllByCzw()	91
6.19.2.3 findAllByDodatkowelnfLike()	92
6.19.2.4 findAllByNazwa()	92
6.19.2.5 findAllByNie()	93
6.19.2.6 findAllByPia()	93
6.19.2.7 findAllByPon()	93
6.19.2.8 findAllBySob()	94
6.19.2.9 findAllBySro()	94
6.19.2.10 findAllByWto()	94
6.20 Dokumentacja interfejsu com.mklachl.sopkom.services.HarmonogramService	95
6.20.1 Opis szczegółowy	95
6.20.2 Dokumentacja funkcji składowych	95
6.20.2.1 findHarmonogramByCzw()	95
6.20.2.2 findHarmonogramByDodatkowelnfLike()	96
6.20.2.3 findHarmonogramById()	96
6.20.2.4 findHarmonogramByNazwa()	97
6.20.2.5 findHarmonogramByNie()	97
6.20.2.6 findHarmonogramByPia()	97
6.20.2.7 findHarmonogramByPon()	98
6.20.2.8 findHarmonogramBySob()	98
6.20.2.9 findHarmonogramBySro()	98
6.20.2.10 findHarmonogramByWto()	99

6.20.2.11 saveHarmonogram() [1/2]	99
6.20.2.12 saveHarmonogram() [2/2]	100
6.21 Dokumentacja klasy com.mklachl.sopkom.services.impl.HarmonogramServiceImpl	100
6.21.1 Opis szczegółowy	101
6.21.2 Dokumentacja konstruktora i destruktora	101
6.21.2.1 HarmonogramServiceImpl()	101
6.21.3 Dokumentacja funkcji składowych	101
6.21.3.1 findHarmonogramByCzw()	101
6.21.3.2 findHarmonogramByDodatkowelnfLike()	101
6.21.3.3 findHarmonogramById()	102
6.21.3.4 findHarmonogramByNazwa()	102
6.21.3.5 findHarmonogramByNie()	103
6.21.3.6 findHarmonogramByPia()	103
6.21.3.7 findHarmonogramByPon()	104
6.21.3.8 findHarmonogramBySob()	104
6.21.3.9 findHarmonogramBySro()	104
6.21.3.10 findHarmonogramByWto()	105
6.21.3.11 saveHarmonogram() [1/2]	105
6.21.3.12 saveHarmonogram() [2/2]	106
6.22 Dokumentacja klasy com.mklachl.sopkom.helpers.Helpers	106
6.22.1 Opis szczegółowy	106
6.22.2 Dokumentacja funkcji składowych	106
6.22.2.1 getUserFromContext()	107
6.22.2.2 inject()	107
6.23 Dokumentacja klasy com.mklachl.sopkom.model.entity.Incydent	107
6.23.1 Opis szczegółowy	107
6.23.2 Dokumentacja funkcji składowych	108
6.23.2.1 getAutobus()	108
6.23.2.2 getDate()	108
6.23.2.3 getDodatkowelnformacje()	108
6.23.2.4 getIncydentId()	109
6.23.2.5 getKierowca()	109
6.23.2.6 getKoszty()	109
6.23.2.7 getKrotko()	109
6.23.2.8 getPrzejazd()	110
6.23.2.9 getTyp()	110
6.23.2.10 setAutobus()	110
6.23.2.11 setDate()	110
6.23.2.12 setDodatkowelnformacje()	111
6.23.2.13 setIncydentId()	111
6.23.2.14 setKierowca()	111
6.23.2.15 setKoszty()	112

6.23.2.16 setKrotko()	112
6.23.2.17 setPrzejazd()	112
6.23.2.18 setTyp()	113
6.24 Dokumentacja klasy com.mklachl.sopkom.controller.IncydentController	113
6.24.1 Opis szczegółowy	114
6.24.2 Dokumentacja funkcji składowych	114
6.24.2.1 deleteIncydent()	114
6.24.2.2 getIncydent()	114
6.24.2.3 getIncydents()	115
6.24.2.4 getIncydentsAfterDate()	115
6.24.2.5 getIncydentsBeforeDate()	115
6.24.2.6 getIncydentsBetweenDates()	116
6.24.2.7 getIncydentsByAutobus()	116
6.24.2.8 getIncydentsByKierowca()	117
6.24.2.9 getIncydentsByPrzejazd()	117
6.24.2.10 saveIncydent()	117
6.24.2.11 updateIncydent()	118
6.24.3 Dokumentacja atrybutów składowych	118
6.24.3.1 incydentRepository	118
6.24.3.2 incydentService	118
6.25 Dokumentacja klasy com.mklachl.sopkom.model.dto.IncydentDto	119
6.25.1 Opis szczegółowy	119
6.25.2 Dokumentacja konstruktora i destruktora	119
6.25.2.1 IncydentDto() [1/2]	119
6.25.2.2 IncydentDto() [2/2]	120
6.25.3 Dokumentacja funkcji składowych	120
6.25.3.1 getAutobusId()	120
6.25.3.2 getDate()	120
6.25.3.3 getDodatkoweInformacje()	121
6.25.3.4 getIncydentId()	121
6.25.3.5 getKierowcaId()	121
6.25.3.6 getKoszty()	121
6.25.3.7 getKrotko()	122
6.25.3.8 getPrzejazdId()	122
6.25.3.9 getTyp()	122
6.25.3.10 setAutobusId()	122
6.25.3.11 setDate()	123
6.25.3.12 setDodatkoweInformacje()	123
6.25.3.13 setIncydentId()	123
6.25.3.14 setKierowcaId()	124
6.25.3.15 setKoszty()	124
6.25.3.16 setKrotko()	124

6.25.3.17 setPrzejazdId()	125
6.25.3.18 setTyp()	125
6.26 Dokumentacja klasy com.mklachl.sopkom.model.dto.IncydentDtoOutput	125
6.26.1 Opis szczegółowy	126
6.26.2 Dokumentacja konstruktora i destruktora	126
6.26.2.1 IncydentDtoOutput()	126
6.26.3 Dokumentacja funkcji składowych	126
6.26.3.1 getAutobus()	127
6.26.3.2 getKierowca()	127
6.26.3.3 getKurs()	127
6.26.3.4 setAutobus()	127
6.26.3.5 setKierowca()	128
6.26.3.6 setKurs()	128
6.27 Dokumentacja klasy com.mklachl.sopkom.exceptions.IncydentException	128
6.27.1 Opis szczegółowy	129
6.27.2 Dokumentacja konstruktora i destruktora	129
6.27.2.1 IncydentException()	129
6.28 Dokumentacja interfejsu com.mklachl.sopkom.repository.IncydentRepository	129
6.28.1 Opis szczegółowy	130
6.28.2 Dokumentacja funkcji składowych	130
6.28.2.1 findAll()	130
6.28.2.2 findAllByAutobus()	130
6.28.2.3 findAllByDate()	131
6.28.2.4 findAllByDateAfter()	131
6.28.2.5 findAllByDateBefore()	131
6.28.2.6 findAllByDateBetween()	132
6.28.2.7 findAllByDodatkoweInformacje()	132
6.28.2.8 findAllByKierowca()	133
6.28.2.9 findAllByKoszty()	133
6.28.2.10 findAllByKosztyGreaterThan()	133
6.28.2.11 findAllByKosztyLessThan()	134
6.28.2.12 findAllByPrzejazd()	134
6.28.2.13 findAllByTyp()	134
6.29 Dokumentacja interfejsu com.mklachl.sopkom.services.IncydentService	135
6.29.1 Opis szczegółowy	135
6.29.2 Dokumentacja funkcji składowych	135
6.29.2.1 findIncydentByAutobus()	135
6.29.2.2 findIncydentByDate()	136
6.29.2.3 findIncydentByDateAfter()	136
6.29.2.4 findIncydentByDateBefore()	137
6.29.2.5 findIncydentByDateBetween()	137
6.29.2.6 findIncydentByDodatkoweInformacje()	137

6.29.2.7 findIncydentById()	138
6.29.2.8 findIncydentByKierowca()	138
6.29.2.9 findIncydentByKoszty()	139
6.29.2.10 findIncydentByKosztyGreaterThan()	139
6.29.2.11 findIncydentByKosztyLessThan()	139
6.29.2.12 findIncydentByPrzejazd()	140
6.29.2.13 findIncydentByTyp()	140
6.29.2.14 saveIncydent() [1/2]	140
6.29.2.15 saveIncydent() [2/2]	141
6.29.2.16 updateIncydent()	141
6.30 Dokumentacja klasy com.mklachl.sopkom.services.impl.IncydentServiceImpl	142
6.30.1 Opis szczegółowy	142
6.30.2 Dokumentacja konstruktora i destruktora	143
6.30.2.1 IncydentServiceImpl()	143
6.30.3 Dokumentacja funkcji składowych	143
6.30.3.1 findIncydentByAutobus()	143
6.30.3.2 findIncydentByDate()	143
6.30.3.3 findIncydentByDateAfter()	144
6.30.3.4 findIncydentByDateBefore()	144
6.30.3.5 findIncydentByDateBetween()	145
6.30.3.6 findIncydentByDodatkoweInformacje()	145
6.30.3.7 findIncydentById()	146
6.30.3.8 findIncydentByKierowca()	146
6.30.3.9 findIncydentByKoszty()	147
6.30.3.10 findIncydentByKosztyGreaterThan()	147
6.30.3.11 findIncydentByKosztyLessThan()	147
6.30.3.12 findIncydentByPrzejazd()	148
6.30.3.13 findIncydentByTyp()	148
6.30.3.14 saveIncydent() [1/2]	149
6.30.3.15 saveIncydent() [2/2]	149
6.30.3.16 updateIncydent()	150
6.31 Dokumentacja klasy com.mklachl.sopkom.model.dto.kurs.InputKursDto	150
6.31.1 Opis szczegółowy	151
6.31.2 Dokumentacja konstruktora i destruktora	151
6.31.2.1 InputKursDto()	151
6.31.3 Dokumentacja funkcji składowych	151
6.31.3.1 getPrzystanki()	151
6.31.3.2 setPrzystanki()	151
6.31.4 Dokumentacja atrybutów składowych	152
6.31.4.1 przystanki	152
6.32 Dokumentacja klasy com.mklachl.sopkom.model.dto.kurs.InputPrzystanekWKursieDto	152
6.32.1 Opis szczegółowy	152

6.32.2 Dokumentacja konstruktora i destruktora	153
6.32.2.1 InputPrzystanekWKursieDto() [1/2]	153
6.32.2.2 InputPrzystanekWKursieDto() [2/2]	153
6.32.3 Dokumentacja funkcji składowych	153
6.32.3.1 getPrzystanekWLini()	153
6.32.3.2 setPrzystanekWLini()	153
6.33 Dokumentacja klasy com.mklachl.sopkom.model.entity.Kierowca	154
6.33.1 Opis szczegółowy	154
6.33.2 Dokumentacja funkcji składowych	154
6.33.2.1 equals()	154
6.33.2.2 getImie()	155
6.33.2.3 getKierowcald()	155
6.33.2.4 getNazwisko()	155
6.33.2.5 getPesel()	156
6.33.2.6 getPrawoJazdyWazneDo()	156
6.33.2.7 getUser()	156
6.33.2.8 hashCode()	156
6.33.2.9 setImie()	156
6.33.2.10 setKierowcald()	157
6.33.2.11 setNazwisko()	157
6.33.2.12 setPesel()	157
6.33.2.13 setPrawoJazdyWazneDo()	158
6.33.2.14 setUser()	158
6.34 Dokumentacja klasy com.mklachl.sopkom.controller.KierowcaController	158
6.34.1 Opis szczegółowy	159
6.34.2 Dokumentacja konstruktora i destruktora	159
6.34.2.1 KierowcaController()	159
6.34.3 Dokumentacja funkcji składowych	159
6.34.3.1 deleteKierowca()	160
6.34.3.2 findAll()	160
6.34.3.3 findKierowcaById()	160
6.34.3.4 findKierowcaByImieAndNazwisko()	161
6.34.3.5 findKierowcaByNazwisko()	161
6.34.3.6 findKierowcaByPesel()	162
6.34.3.7 findKierowcaByPrawoJazdyWazneDoAfter()	162
6.34.3.8 findKierowcaByPrawoJazdyWazneDoBefore()	162
6.34.3.9 findKierowcaByUser()	163
6.34.3.10 saveKierowca()	163
6.34.3.11 template()	164
6.34.3.12 updateKierowca()	164
6.35 Dokumentacja klasy com.mklachl.sopkom.model.dto.KierowcaDto	164
6.35.1 Opis szczegółowy	165

6.35.2 Dokumentacja konstruktora i destruktora	165
6.35.2.1 KierowcaDto() [1/2]	165
6.35.2.2 KierowcaDto() [2/2]	165
6.35.3 Dokumentacja funkcji składowych	165
6.35.3.1 getImie()	166
6.35.3.2 getKierowcaId()	166
6.35.3.3 getNazwisko()	166
6.35.3.4 getPesel()	166
6.35.3.5 getPrawoJazdyWazneDo()	167
6.35.3.6 getUser()	167
6.35.3.7 setImie()	167
6.35.3.8 setKierowcaId()	167
6.35.3.9 setNazwisko()	168
6.35.3.10 setPesel()	168
6.35.3.11 setPrawoJazdyWazneDo()	168
6.35.3.12 setUserDto()	169
6.35.3.13 toString()	169
6.36 Dokumentacja interfejsu com.mklachl.sopkom.repository.KierowcaRepository	169
6.36.1 Opis szczegółowy	170
6.36.2 Dokumentacja funkcji składowych	170
6.36.2.1 findAll()	170
6.36.2.2 findAllByAvailability()	170
6.36.2.3 findAllByImieAndNazwisko()	171
6.36.2.4 findAllByImieAndNazwiskoLike()	171
6.36.2.5 findAllByPeselLike()	171
6.36.2.6 findAllByPrawoJazdyWazneDoAfter()	172
6.36.2.7 findAllByPrawoJazdyWazneDoBefore()	172
6.36.2.8 findByNazwiskoLike()	173
6.36.2.9 findByPesel()	173
6.36.2.10 findByUser()	173
6.37 Dokumentacja interfejsu com.mklachl.sopkom.services.KierowcaService	174
6.37.1 Opis szczegółowy	174
6.37.2 Dokumentacja funkcji składowych	174
6.37.2.1 findKierowcaByAvailability()	174
6.37.2.2 findKierowcaById()	175
6.37.2.3 findKierowcaByImieAndNazwisko()	175
6.37.2.4 findKierowcaByPesel()	176
6.37.2.5 findKierowcaByPrawoJazdyWazneDoAfter()	176
6.37.2.6 findKierowcaByPrawoJazdyWazneDoBefore()	176
6.37.2.7 findKierowcaByUser()	177
6.37.2.8 saveKierowca() [1/2]	177
6.37.2.9 saveKierowca() [2/2]	178

6.38 Dokumentacja klasy com.mklachl.sopkom.services.impl.KierowcaServiceImpl	178
6.38.1 Opis szczegółowy	178
6.38.2 Dokumentacja konstruktora i destruktora	179
6.38.2.1 KierowcaServiceImpl()	179
6.38.3 Dokumentacja funkcji składowych	179
6.38.3.1 findKierowcaByAvailability()	179
6.38.3.2 findKierowcaById()	179
6.38.3.3 findKierowcaByImieAndNazwisko()	180
6.38.3.4 findKierowcaByPesel()	180
6.38.3.5 findKierowcaByPrawoJazdyWazneDoAfter()	181
6.38.3.6 findKierowcaByPrawoJazdyWazneDoBefore()	181
6.38.3.7 findKierowcaByUser()	182
6.38.3.8 saveKierowca() [1/2]	182
6.38.3.9 saveKierowca() [2/2]	182
6.39 Dokumentacja klasy com.mklachl.sopkom.model.entity.Kurs	183
6.39.1 Opis szczegółowy	183
6.39.2 Dokumentacja funkcji składowych	183
6.39.2.1 getHarmonogram()	184
6.39.2.2 getKierunek()	184
6.39.2.3 getKursId()	184
6.39.2.4 getKursPrzystanekWlinii()	184
6.39.2.5 getLinia()	185
6.39.2.6 getTypAutobusu()	185
6.39.2.7 setHarmonogram()	185
6.39.2.8 setKierunek()	185
6.39.2.9 setKursId()	186
6.39.2.10 setKursPrzystanekWlinii()	186
6.39.2.11 setLinia()	186
6.39.2.12 setTypAutobusu()	187
6.40 Dokumentacja klasy com.mklachl.sopkom.controller.KursController	187
6.40.1 Opis szczegółowy	188
6.40.2 Dokumentacja funkcji składowych	188
6.40.2.1 addKurs()	188
6.40.2.2 deleteKurs()	188
6.40.2.3 getAllKursy()	189
6.40.2.4 getKurs()	189
6.40.2.5 getTemplateKurs()	190
6.40.2.6 getTemplateKursInput()	190
6.40.2.7 updateKurs()	190
6.40.3 Dokumentacja atrybutów składowych	191
6.40.3.1 kursRepo	191
6.40.3.2 kursService	191

6.40.3.3 kursy	191
6.40.3.4 liniaRepository	191
6.40.3.5 liniaService	192
6.40.3.6 przystnekWliniRepository	192
6.41 Dokumentacja klasy com.mklachl.sopkom.model.dto.kurs.KursDto	192
6.41.1 Opis szczegółowy	192
6.41.2 Dokumentacja konstruktora i destruktora	193
6.41.2.1 KursDto() [1/2]	193
6.41.2.2 KursDto() [2/2]	193
6.41.3 Dokumentacja funkcji składowych	193
6.41.3.1 getHarmonogram()	193
6.41.3.2 getLinia()	194
6.41.3.3 getOdwracalny()	194
6.41.3.4 getPrzystanki()	194
6.41.3.5 setHarmonogram()	194
6.41.3.6 setLinia()	195
6.41.3.7 setOdwracalny()	195
6.41.3.8 setPrzystanki()	195
6.42 Dokumentacja klasy com.mklachl.sopkom.model.dto.kurs.UPDATED.KursDtoUPDATED	196
6.42.1 Opis szczegółowy	196
6.42.2 Dokumentacja konstruktora i destruktora	196
6.42.2.1 KursDtoUPDATED() [1/2]	197
6.42.2.2 KursDtoUPDATED() [2/2]	197
6.42.3 Dokumentacja funkcji składowych	197
6.42.3.1 getHarmonogram()	197
6.42.3.2 getKieruenk()	197
6.42.3.3 getKursId()	197
6.42.3.4 getLinia()	197
6.42.3.5 getPrzystanki()	198
6.42.3.6 isOdwracalny()	198
6.42.3.7 setHarmonogram()	198
6.42.3.8 setKieruenk()	198
6.42.3.9 setKursId()	198
6.42.3.10 setLinia()	198
6.42.3.11 setOdwracalny()	199
6.42.3.12 setPrzystanki()	199
6.42.4 Dokumentacja atrybutów składowych	199
6.42.4.1 harmonogram	199
6.42.4.2 kieruenk	199
6.42.4.3 kursId	199
6.42.4.4 linia	200
6.42.4.5 odwracalny	200

6.42.4.6 przystanki	200
6.43 Dokumentacja klasy com.mklachl.sopkom.model.entity.KursPrzystanekWlini	200
6.43.1 Opis szczegółowy	200
6.43.2 Dokumentacja funkcji składowych	201
6.43.2.1 getGodzinna()	201
6.43.2.2 getKurs()	201
6.43.2.3 getKursPrzystanekWlinild()	201
6.43.2.4 getPrzystanekWlini()	202
6.43.2.5 setGodzinna()	202
6.43.2.6 setKurs()	203
6.43.2.7 setKursPrzystanekWlinild()	203
6.43.2.8 setPrzystanekWlini()	203
6.44 Dokumentacja interfejsu com.mklachl.sopkom.repository.KursPrzystanekWliniRepository	204
6.44.1 Opis szczegółowy	204
6.44.2 Dokumentacja funkcji składowych	204
6.44.2.1 findAll()	205
6.44.2.2 findAllByGodzinna()	205
6.44.2.3 findAllByGodzinnaAfter()	205
6.44.2.4 findAllByGodzinnaBefore()	206
6.44.2.5 findAllByKurs()	206
6.44.2.6 findAllByPrzystanekWlini()	206
6.45 Dokumentacja interfejsu com.mklachl.sopkom.repository.KursRepository	207
6.45.1 Opis szczegółowy	207
6.45.2 Dokumentacja funkcji składowych	207
6.45.2.1 findAll()	207
6.45.2.2 findAllByHarmonogram()	207
6.45.2.3 findAllByKierunek()	208
6.45.2.4 findAllByLinia()	208
6.45.2.5 findAllByTypAutobusu()	208
6.46 Dokumentacja interfejsu com.mklachl.sopkom.services.KursService	210
6.46.1 Opis szczegółowy	210
6.46.2 Dokumentacja funkcji składowych	210
6.46.2.1 findKursByHarmonogram()	210
6.46.2.2 findKursByKierunek()	211
6.46.2.3 findKursByLinia()	211
6.46.2.4 findKursByTypAutobusu()	212
6.46.2.5 saveKurs() [1/2]	212
6.46.2.6 saveKurs() [2/2]	212
6.46.2.7 updateKurs()	213
6.47 Dokumentacja klasy com.mklachl.sopkom.services.impl.KursServicelmpl	213
6.47.1 Opis szczegółowy	214
6.47.2 Dokumentacja konstruktora i destruktora	214

6.47.2.1 KursServiceImpl()	214
6.47.3 Dokumentacja funkcji składowych	214
6.47.3.1 findKursByHarmonogram()	214
6.47.3.2 findKursByKierunek()	215
6.47.3.3 findKursByLinia()	215
6.47.3.4 findKursByTypAutobusu()	216
6.47.3.5 saveKurs() [1/2]	216
6.47.3.6 saveKurs() [2/2]	216
6.47.3.7 updateKurs()	218
6.48 Dokumentacja klasy com.mklachl.sopkom.model.entity.Linia	218
6.48.1 Opis szczegółowy	219
6.48.2 Dokumentacja funkcji składowych	219
6.48.2.1 getLiniald()	219
6.48.2.2 getNumer()	219
6.48.2.3 getPrzystankiWlini()	220
6.48.2.4 odwracalna()	220
6.48.2.5 setLiniald()	220
6.48.2.6 setNumer()	220
6.48.2.7 setPrzystankiWlini()	221
6.49 Dokumentacja klasy com.mklachl.sopkom.controller.LiniaController	221
6.49.1 Opis szczegółowy	221
6.49.2 Dokumentacja funkcji składowych	221
6.49.2.1 addLinia()	222
6.49.2.2 deleteLinia()	223
6.49.2.3 getAllPrzystanki()	223
6.49.2.4 getAllPrzystankiFull()	224
6.49.2.5 getLinia()	224
6.49.2.6 getPrzystanekFull()	224
6.49.2.7 template()	225
6.49.2.8 templateInput()	225
6.49.2.9 updateLinia()	225
6.50 Dokumentacja klasy com.mklachl.sopkom.model.dto.linia.LiniaDto	226
6.50.1 Opis szczegółowy	226
6.50.2 Dokumentacja konstruktora i destruktora	226
6.50.2.1 LiniaDto() [1/2]	226
6.50.2.2 LiniaDto() [2/2]	226
6.50.3 Dokumentacja funkcji składowych	227
6.50.3.1 getId()	227
6.50.3.2 getNumer()	227
6.50.3.3 setId()	227
6.50.3.4 setNumer()	228
6.50.4 Dokumentacja atrybutów składowych	228

6.50.4.1 id	228
6.50.4.2 numer	228
6.51 Dokumentacja klasy com.mklachl.sopkom.model.dto.linia.LiniaDtoInput	229
6.51.1 Opis szczegółowy	229
6.51.2 Dokumentacja konstruktora i destruktora	229
6.51.2.1 LiniaDtoInput() [1/2]	229
6.51.2.2 LiniaDtoInput() [2/2]	229
6.51.3 Dokumentacja funkcji składowych	230
6.51.3.1 getPrzystanki()	230
6.51.3.2 setPrzystanki()	230
6.52 Dokumentacja klasy com.mklachl.sopkom.model.dto.linia.LiniaDtoOutput	230
6.52.1 Opis szczegółowy	231
6.52.2 Dokumentacja konstruktora i destruktora	231
6.52.2.1 LiniaDtoOutput() [1/2]	231
6.52.2.2 LiniaDtoOutput() [2/2]	231
6.52.3 Dokumentacja funkcji składowych	232
6.52.3.1 getPrzystanki()	232
6.52.3.2 isOdwracalna()	232
6.52.3.3 setOdwracalna()	232
6.52.3.4 setPrzystanki()	233
6.52.4 Dokumentacja atrybutów składowych	233
6.52.4.1 odwracalna	233
6.53 Dokumentacja klasy com.mklachl.sopkom.exceptions.LiniaNotFoundException	233
6.53.1 Opis szczegółowy	234
6.53.2 Dokumentacja konstruktora i destruktora	234
6.53.2.1 LiniaNotFoundException()	234
6.54 Dokumentacja interfejsu com.mklachl.sopkom.repository.LiniaRepository	234
6.54.1 Opis szczegółowy	234
6.54.2 Dokumentacja funkcji składowych	235
6.54.2.1 findAll()	235
6.54.2.2 findById()	235
6.54.2.3 findByNumer()	235
6.55 Dokumentacja interfejsu com.mklachl.sopkom.services.LiniaService	236
6.55.1 Opis szczegółowy	236
6.55.2 Dokumentacja funkcji składowych	236
6.55.2.1 deleteLinia()	236
6.55.2.2 findLiniaById()	237
6.55.2.3 findLiniaByNumer()	237
6.55.2.4 patchLinia()	237
6.55.2.5 saveLinia()	238
6.56 Dokumentacja klasy com.mklachl.sopkom.services.impl.LiniaServiceImpl	238
6.56.1 Opis szczegółowy	239

6.56.2 Dokumentacja funkcji składowych	239
6.56.2.1 deleteLinia()	239
6.56.2.2 findLiniaById()	239
6.56.2.3 findLiniaByNumer()	240
6.56.2.4 patchLinia()	240
6.56.2.5 saveLinia()	241
6.57 Dokumentacja klasy com.mklachl.sopkom.filter.LoggingFilter	241
6.57.1 Opis szczegółowy	242
6.57.2 Dokumentacja funkcji składowych	242
6.57.2.1 doFilterInternal()	242
6.58 Dokumentacja klasy com.mklachl.sopkom.controller.LoginController	242
6.58.1 Opis szczegółowy	242
6.58.2 Dokumentacja funkcji składowych	242
6.58.2.1 getLoginPage()	243
6.58.2.2 getLogoutPage()	243
6.59 Dokumentacja klasy com.mklachl.sopkom.model.dto.Message	243
6.59.1 Opis szczegółowy	244
6.59.2 Dokumentacja konstruktora i destruktora	244
6.59.2.1 Message() [1/2]	244
6.59.2.2 Message() [2/2]	244
6.59.3 Dokumentacja funkcji składowych	244
6.59.3.1 SetMessage()	244
6.59.3.2 SetMessageInt()	245
6.59.4 Dokumentacja atrybutów składowych	245
6.59.4.1 message	245
6.60 Dokumentacja klasy com.mklachl.sopkom.topdf.Pdf	246
6.60.1 Opis szczegółowy	246
6.60.2 Dokumentacja funkcji składowych	246
6.60.2.1 genRozklad()	246
6.60.2.2 genruj()	246
6.60.2.3 getAllByPrzystanekId()	247
6.60.2.4 getLiczbaLiniiNaPrzystanku()	247
6.60.3 Dokumentacja atrybutów składowych	247
6.60.3.1 kursRepository	247
6.60.3.2 liniaRepos	247
6.60.3.3 przystanekRepos	247
6.60.3.4 przystanekWliniRepos	248
6.61 Dokumentacja klasy com.mklachl.sopkom.model.dto.przejazd.PKPWLDtoUpdate	248
6.61.1 Opis szczegółowy	248
6.61.2 Dokumentacja konstruktora i destruktora	248
6.61.2.1 PKPWLDtoUpdate()	248
6.61.3 Dokumentacja funkcji składowych	248

6.61.3.1 getDate()	249
6.61.3.2 getId()	249
6.61.3.3 setDate()	249
6.61.3.4 setId()	249
6.62 Dokumentacja klasy com.mklachl.sopkom.model.entity.Przejazd	250
6.62.1 Opis szczegółowy	250
6.62.2 Dokumentacja funkcji składowych	250
6.62.2.1 getAutobus()	251
6.62.2.2 getCenaZaLitr()	251
6.62.2.3 getData()	251
6.62.2.4 getDataKonca()	251
6.62.2.5 getDataStartu()	252
6.62.2.6 getDlugoscTrasy()	252
6.62.2.7 getKierowca()	252
6.62.2.8 getKurs()	252
6.62.2.9 getPrzejazdBilet()	253
6.62.2.10 getPrzejazdId()	253
6.62.2.11 getPrzejazdKursPrzystanekWlini()	253
6.62.2.12 getSpalanie()	253
6.62.2.13 setAutobus()	253
6.62.2.14 setCenaZaLitr()	254
6.62.2.15 setData()	254
6.62.2.16 setDataKonca()	254
6.62.2.17 setDataStartu()	255
6.62.2.18 setDlugoscTrasy()	255
6.62.2.19 setKierowca()	255
6.62.2.20 setKurs()	256
6.62.2.21 setPrzejazdBilet()	256
6.62.2.22 setPrzejazdId()	256
6.62.2.23 setPrzejazdKursPrzystanekWlini()	256
6.62.2.24 setSpalanie()	257
6.63 Dokumentacja klasy com.mklachl.sopkom.model.entity.PrzejazdBilet	257
6.63.1 Opis szczegółowy	257
6.63.2 Dokumentacja funkcji składowych	258
6.63.2.1 getBilet()	258
6.63.2.2 getCenaBiletu()	258
6.63.2.3 getPrzejazd()	258
6.63.2.4 getPrzejazdBiletId()	259
6.63.2.5 setBilet()	259
6.63.2.6 setCenaBiletu()	260
6.63.2.7 setPrzejazd()	260
6.63.2.8 setPrzejazdBiletId()	260

6.64 Dokumentacja klasy com.mklachl.sopkom.model.dto.PrzejazdBiletDto	261
6.64.1 Opis szczegółowy	261
6.64.2 Dokumentacja konstruktora i destruktora	261
6.64.2.1 PrzejazdBiletDto() [1/2]	261
6.64.2.2 PrzejazdBiletDto() [2/2]	261
6.64.3 Dokumentacja funkcji składowych	262
6.64.3.1 getBiletId()	262
6.64.3.2 getCenaBiletu()	262
6.64.3.3 getPrzejazdBiletId()	262
6.64.3.4 getPrzejazdId()	263
6.64.3.5 setBiletId()	263
6.64.3.6 setCenaBiletu()	263
6.64.3.7 setPrzejazdBiletId()	263
6.64.3.8 setPrzejazdId()	265
6.65 Dokumentacja interfejsu com.mklachl.sopkom.repository.PrzejazdBiletRepository	265
6.65.1 Opis szczegółowy	266
6.65.2 Dokumentacja funkcji składowych	266
6.65.2.1 findAll()	266
6.65.2.2 findAllByBilet()	266
6.65.2.3 findAllByCenaBiletu()	266
6.65.2.4 findAllByCenaBiletuGreaterThanOrEqual()	267
6.65.2.5 findAllByCenaBiletuLessThan()	267
6.65.2.6 findAllByPrzejazd()	268
6.66 Dokumentacja klasy com.mklachl.sopkom.controller.PrzejazdController	268
6.66.1 Opis szczegółowy	268
6.66.2 Dokumentacja funkcji składowych	269
6.66.2.1 deletePrzejazd()	269
6.66.2.2 getAllAvailableBusesForKurs()	269
6.66.2.3 getAllAvailableDatesForKurs()	270
6.66.2.4 getAllAvailableDriversForKurs()	270
6.66.2.5 getPrzejazd()	270
6.66.2.6 getPrzejazdy()	271
6.66.2.7 getPrzyszłePrzejazdy()	271
6.66.2.8 savePrzejazd()	271
6.66.3 Dokumentacja atrybutów składowych	272
6.66.3.1 autobusRepository	272
6.66.3.2 harmonogramRepository	272
6.66.3.3 kierowcaRepository	272
6.66.3.4 kursRepository	272
6.66.3.5 kursService	273
6.66.3.6 przejazdRepository	273
6.66.3.7 przejazdService	273

6.67 Dokumentacja klasy com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoInput	273
6.67.1 Opis szczegółowy	273
6.67.2 Dokumentacja konstruktora i destruktora	274
6.67.2.1 PrzejazdDtoInput()	274
6.67.3 Dokumentacja funkcji składowych	274
6.67.3.1 getAutobusId()	274
6.67.3.2 getDataPrzejazdu()	274
6.67.3.3 getKierowcaId()	275
6.67.3.4 getKursId()	275
6.67.3.5 setAutobusId()	275
6.67.3.6 setDataPrzejazdu()	275
6.67.3.7 setKierowcaId()	276
6.67.3.8 setKursId()	276
6.67.3.9 toString()	276
6.68 Dokumentacja klasy com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput	277
6.68.1 Opis szczegółowy	277
6.68.2 Dokumentacja konstruktora i destruktora	277
6.68.2.1 PrzejazdDtoOutput() [1/3]	278
6.68.2.2 PrzejazdDtoOutput() [2/3]	278
6.68.2.3 PrzejazdDtoOutput() [3/3]	278
6.68.3 Dokumentacja funkcji składowych	278
6.68.3.1 getAutobusNazwa()	278
6.68.3.2 getAutobusNumerRejestracyjny()	279
6.68.3.3 getCenaZaLitr()	279
6.68.3.4 getData()	279
6.68.3.5 getDataKonca()	280
6.68.3.6 getDataStartu()	280
6.68.3.7 getDlugoscTrasy()	280
6.68.3.8 getKierowcaImieNazwisko()	280
6.68.3.9 getKierunek()	281
6.68.3.10 getKursId()	281
6.68.3.11 getLiczbaBiletow()	281
6.68.3.12 getLiniaNumer()	281
6.68.3.13 getPrzejazdId()	282
6.68.3.14 getPrzystanki()	282
6.68.3.15 getSpalanie()	282
6.68.3.16 getStatus()	282
6.68.3.17 setAutobusNazwa()	282
6.68.3.18 setAutobusNumerRejestracyjny()	283
6.68.3.19 setCenaZaLitr()	283
6.68.3.20 setData()	283
6.68.3.21 setDataKonca()	284

6.68.3.22 setDataStartu()	284
6.68.3.23 setDlugoscTrasy()	284
6.68.3.24 setKierowcamiNazwisko()	285
6.68.3.25 setKierunek()	285
6.68.3.26 setKursId()	285
6.68.3.27 setLiczbaBiletow()	285
6.68.3.28 setLiniaNumer()	286
6.68.3.29 setPrzejazdId()	286
6.68.3.30 setPrzystanki()	286
6.68.3.31 setSpalanie()	287
6.68.3.32 setStatus()	287
6.69 Dokumentacja klasy com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate	287
6.69.1 Opis szczegółowy	288
6.69.2 Dokumentacja konstruktora i destruktora	288
6.69.2.1 PrzejazdDtoUpdate()	288
6.69.3 Dokumentacja funkcji składowych	288
6.69.3.1 getCenaZaLitr()	288
6.69.3.2 getDlugoszTrasy()	289
6.69.3.3 getLiczbaBiletowNormalnych()	289
6.69.3.4 getLiczbaBiletowUlgowych()	289
6.69.3.5 getPrzejazdId()	289
6.69.3.6 getPrzystanki()	290
6.69.3.7 getSpalanie()	290
6.69.3.8 setCenaZaLitr()	290
6.69.3.9 setDlugoszTrasy()	290
6.69.3.10 setLiczbaBiletowNormalnych()	291
6.69.3.11 setLiczbaBiletowUlgowych()	291
6.69.3.12 setPrzejazdId()	291
6.69.3.13 setPrzystanki()	292
6.69.3.14 setSpalanie()	292
6.69.3.15 toString()	292
6.70 Dokumentacja klasy com.mklachl.sopkom.model.entity.PrzejazdKursPrzystanekWlini	293
6.70.1 Opis szczegółowy	293
6.70.2 Dokumentacja funkcji składowych	293
6.70.2.1 getKursPrzystanekWlini()	293
6.70.2.2 getPrzejazd()	293
6.70.2.3 getPrzejazdKursPrzystanekWliniId()	294
6.70.2.4 getRealnaGodzinna()	294
6.70.2.5 setKursPrzystanekWlini()	294
6.70.2.6 setPrzejazd()	294
6.70.2.7 setPrzejazdKursPrzystanekWliniId()	295
6.70.2.8 setRealnaGodzinna()	295

6.71 Dokumentacja klasy com.mklachl.sopkom.model.dto.PrzejazdKursPrzystanekWliniDto	295
6.71.1 Opis szczegółowy	296
6.71.2 Dokumentacja konstruktora i destruktora	296
6.71.2.1 PrzejazdKursPrzystanekWliniDto() [1/2]	296
6.71.2.2 PrzejazdKursPrzystanekWliniDto() [2/2]	296
6.71.3 Dokumentacja funkcji składowych	297
6.71.3.1 getKolejnosc()	297
6.71.3.2 getNazwa()	297
6.71.3.3 getPrzejazdKursPrzystanekWliniId()	297
6.71.3.4 getPrzewidywanaGodzinna()	298
6.71.3.5 getPrzystanekId()	298
6.71.3.6 getRealnaGodzinna()	298
6.71.3.7 setKolejnosc()	298
6.71.3.8 setNazwa()	299
6.71.3.9 setPrzejazdKursPrzystanekWliniId()	299
6.71.3.10 setPrzewidywanaGodzinna()	299
6.71.3.11 setPrzystanekId()	300
6.71.3.12 setRealnaGodzinna()	300
6.72 Dokumentacja interfejsu com.mklachl.sopkom.repository.PrzejazdKursPrzystanekWliniRepository	300
6.72.1 Opis szczegółowy	301
6.72.2 Dokumentacja funkcji składowych	301
6.72.2.1 findAll()	301
6.72.2.2 findAllByKursPrzystanekWlini()	301
6.72.2.3 findAllByPrzejazd()	302
6.72.2.4 findAllByRealnaGodzinna()	302
6.72.2.5 findAllByRealnaGodzinnaAfter()	303
6.72.2.6 findAllByRealnaGodzinnaBefore()	303
6.73 Dokumentacja interfejsu com.mklachl.sopkom.repository.PrzejazdRepository	303
6.73.1 Opis szczegółowy	304
6.73.2 Dokumentacja funkcji składowych	304
6.73.2.1 findAll()	304
6.73.2.2 findAllByAutobus()	304
6.73.2.3 findAllByCenaZaLitru()	305
6.73.2.4 findAllByCenaZaLitruGreaterThanOrEqual()	305
6.73.2.5 findAllByCenaZaLitruLessThan()	306
6.73.2.6 findAllByData() [1/2]	306
6.73.2.7 findAllByData() [2/2]	306
6.73.2.8 findAllByDataKonca()	307
6.73.2.9 findAllByDataKoncaAfter()	307
6.73.2.10 findAllByDataKoncaBefore()	307
6.73.2.11 findAllByDataStartu()	308
6.73.2.12 findAllByDataStartuAfter()	308

6.73.2.13 findAllByDataStartuBefore()	308
6.73.2.14 findAllByDlugoscTrasy()	310
6.73.2.15 findAllByDlugoscTrasyGreaterThan()	310
6.73.2.16 findAllByDlugoscTrasyLessThan()	311
6.73.2.17 findAllByKierowca()	311
6.73.2.18 findAllByKurs()	311
6.73.2.19 findAllByKursAndData()	312
6.73.2.20 findAllBySpalanie()	312
6.73.2.21 findAllBySpalanieGreaterThan()	312
6.73.2.22 findAllBySpalanieLessThan()	313
6.74 Dokumentacja interfejsu com.mklachl.sopkom.services.PrzejazdService	313
6.74.1 Opis szczegółowy	314
6.74.2 Dokumentacja funkcji składowych	314
6.74.2.1 createDetachedPrzejazd()	314
6.74.2.2 findAllDatesForKurs()	315
6.74.2.3 findPrzejazdByAutobus()	315
6.74.2.4 findPrzejazdByCenaZaLitr()	315
6.74.2.5 findPrzejazdByCenaZaLitrGreaterThan()	316
6.74.2.6 findPrzejazdByCenaZaLitrLessThan()	316
6.74.2.7 findPrzejazdByDataKonca()	317
6.74.2.8 findPrzejazdByDataKoncaAfter()	317
6.74.2.9 findPrzejazdByDataKoncaBefore()	317
6.74.2.10 findPrzejazdByDataStartu()	318
6.74.2.11 findPrzejazdByDataStartuAfter()	318
6.74.2.12 findPrzejazdByDataStartuBefore()	319
6.74.2.13 findPrzejazdByDlugoscTrasy()	319
6.74.2.14 findPrzejazdByDlugoscTrasyGreaterThan()	319
6.74.2.15 findPrzejazdByDlugoscTrasyLessThan()	320
6.74.2.16 findPrzejazdById()	320
6.74.2.17 findPrzejazdByKierowca()	321
6.74.2.18 findPrzejazdByKurs()	321
6.74.2.19 findPrzejazdBySpalanie()	321
6.74.2.20 findPrzejazdBySpalanieGreaterThan()	322
6.74.2.21 findPrzejazdBySpalanieLessThan()	322
6.74.2.22 findPrzejazdyByKursAndData()	323
6.74.2.23 savePrzejazd() [1/2]	323
6.74.2.24 savePrzejazd() [2/2]	323
6.74.2.25 updatePrzejazd()	324
6.75 Dokumentacja klasy com.mklachl.sopkom.services.impl.PrzejazdServiceImpl	324
6.75.1 Opis szczegółowy	325
6.75.2 Dokumentacja konstruktora i destruktora	325
6.75.2.1 PrzejazdServiceImpl()	325

6.75.3 Dokumentacja funkcji składowych	326
6.75.3.1 createDetachedPrzejazd()	326
6.75.3.2 findAllDatesForKurs()	326
6.75.3.3 findPrzejazdByAutobus()	327
6.75.3.4 findPrzejazdByCenaZaLitr()	327
6.75.3.5 findPrzejazdByCenaZaLitrGreaterThan()	327
6.75.3.6 findPrzejazdByCenaZaLitrLessThan()	328
6.75.3.7 findPrzejazdByDataKonca()	328
6.75.3.8 findPrzejazdByDataKoncaAfter()	329
6.75.3.9 findPrzejazdByDataKoncaBefore()	329
6.75.3.10 findPrzejazdByDataStartu()	330
6.75.3.11 findPrzejazdByDataStartuAfter()	330
6.75.3.12 findPrzejazdByDataStartuBefore()	330
6.75.3.13 findPrzejazdByDlugoscTrasy()	331
6.75.3.14 findPrzejazdByDlugoscTrasyGreaterThan()	331
6.75.3.15 findPrzejazdByDlugoscTrasyLessThan()	332
6.75.3.16 findPrzejazdById()	332
6.75.3.17 findPrzejazdByKierowca()	333
6.75.3.18 findPrzejazdByKurs()	333
6.75.3.19 findPrzejazdBySpalanie()	333
6.75.3.20 findPrzejazdBySpalanieGreaterThan()	334
6.75.3.21 findPrzejazdBySpalanieLessThan()	334
6.75.3.22 findPrzejazdyByKursAndData()	335
6.75.3.23 savePrzejazd() [1/2]	335
6.75.3.24 savePrzejazd() [2/2]	336
6.75.3.25 updatePrzejazd()	336
6.76 Dokumentacja klasy com.mklachl.sopkom.model.entity.Przystanek	337
6.76.1 Opis szczegółowy	337
6.76.2 Dokumentacja funkcji składowych	337
6.76.2.1 getDlugoscGeograficzna()	337
6.76.2.2 getKodPocztowy()	338
6.76.2.3 getMiasto()	338
6.76.2.4 getNazwa()	338
6.76.2.5 getPrzystanekId()	338
6.76.2.6 getPrzystanekOdwrotny()	339
6.76.2.7 getPrzystankiWlini()	339
6.76.2.8 getSzerokoscGeograficzna()	339
6.76.2.9 getUlica()	339
6.76.2.10 setDlugoscGeograficzna()	339
6.76.2.11 setKodPocztowy()	340
6.76.2.12 setMiasto()	340
6.76.2.13 setNazwa()	340

6.76.2.14 setPrzystanekId()	341
6.76.2.15 setPrzystanekOdwrorny()	341
6.76.2.16 setPrzystankiWlini()	341
6.76.2.17 setSzerokoscGeograficzna()	342
6.76.2.18 setUlica()	342
6.77 Dokumentacja klasy com.mklachl.sopkom.controller.PrzystanekController	342
6.77.1 Opis szczegółowy	343
6.77.2 Dokumentacja funkcji składowych	343
6.77.2.1 addPrzystanek()	343
6.77.2.2 deletePrzystanek()	343
6.77.2.3 getAllPrzystanki()	344
6.77.2.4 getAllPrzystankiFull()	344
6.77.2.5 getPrzystanek()	344
6.77.2.6 getPrzystanekFull()	345
6.77.2.7 template()	345
6.77.2.8 updatePrzystanek()	345
6.78 Dokumentacja klasy com.mklachl.sopkom.model.dto.PrzystanekDto	346
6.78.1 Opis szczegółowy	346
6.78.2 Dokumentacja konstruktora i destruktora	346
6.78.2.1 PrzystanekDto() [1/2]	347
6.78.2.2 PrzystanekDto() [2/2]	347
6.78.3 Dokumentacja funkcji składowych	347
6.78.3.1 getDlugoscGeograficzna()	347
6.78.3.2 getKodPocztowy()	347
6.78.3.3 getMiasto()	348
6.78.3.4 getNazwa()	348
6.78.3.5 getPrzystanekId()	348
6.78.3.6 getPrzystanekOdwrorny()	348
6.78.3.7 getPrzystanekOdwrornyNazwa()	349
6.78.3.8 getSzerokoscGeograficzna()	349
6.78.3.9 getUlica()	349
6.78.3.10 setDlugoscGeograficzna()	349
6.78.3.11 setKodPocztowy()	350
6.78.3.12 setMiasto()	350
6.78.3.13 setNazwa()	350
6.78.3.14 setPrzystanekId()	351
6.78.3.15 setPrzystanekOdwrorny()	351
6.78.3.16 setPrzystanekOdwrornyNazwa()	351
6.78.3.17 setSzerokoscGeograficzna()	352
6.78.3.18 setUlica()	352
6.79 Dokumentacja klasy com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLinia	352
6.79.1 Opis szczegółowy	353

6.79.2 Dokumentacja konstruktora i destruktora	353
6.79.2.1 PrzystanekDtoDlaLinia() [1/2]	353
6.79.2.2 PrzystanekDtoDlaLinia() [2/2]	353
6.79.3 Dokumentacja funkcji składowych	354
6.79.3.1 getAdres()	354
6.79.3.2 getKolejnosc()	354
6.79.3.3 getPrzystanekwlinid()	354
6.79.3.4 setAdres()	354
6.79.3.5 setKolejnosc()	355
6.79.3.6 setPrzystanekwlinid()	355
6.80 Dokumentacja klasy com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLinialInput	355
6.80.1 Opis szczegółowy	356
6.80.2 Dokumentacja konstruktora i destruktora	356
6.80.2.1 PrzystanekDtoDlaLinialInput() [1/2]	356
6.80.2.2 PrzystanekDtoDlaLinialInput() [2/2]	356
6.80.3 Dokumentacja funkcji składowych	356
6.80.3.1 getKolejnosc()	357
6.80.3.2 getPrzystanekId()	357
6.80.3.3 setKolejnosc()	357
6.80.3.4 setPrzystanekId()	357
6.81 Dokumentacja klasy com.mklachl.sopkom.model.dto.PrzystanekDtoFull	358
6.81.1 Opis szczegółowy	359
6.81.2 Dokumentacja konstruktora i destruktora	359
6.81.2.1 PrzystanekDtoFull() [1/2]	359
6.81.2.2 PrzystanekDtoFull() [2/2]	359
6.81.3 Dokumentacja funkcji składowych	359
6.81.3.1 getDlugoscGeograficzna()	359
6.81.3.2 getKodPocztowy()	360
6.81.3.3 getMiasto()	360
6.81.3.4 getNazwa()	360
6.81.3.5 getPrzystanekId()	360
6.81.3.6 getPrzystanekOdwrotny()	361
6.81.3.7 getSzerokoscGeograficzna()	361
6.81.3.8 getUlica()	361
6.81.3.9 setDlugoscGeograficzna()	361
6.81.3.10 setKodPocztowy()	362
6.81.3.11 setMiasto()	362
6.81.3.12 setNazwa()	362
6.81.3.13 setPrzystanekId()	363
6.81.3.14 setPrzystanekOdwrotny()	363
6.81.3.15 setSzerokoscGeograficzna()	363
6.81.3.16 setUlica()	364

6.81.4 Dokumentacja atrybutów składowych	364
6.81.4.1 dlugoscGeograficzna	364
6.81.4.2 kodPocztowy	364
6.81.4.3 miasto	364
6.81.4.4 nazwa	365
6.81.4.5 przystanekId	365
6.81.4.6 przystanekOdwrotny	365
6.81.4.7 szerokoscGeograficzna	365
6.81.4.8 ulica	365
6.82 Dokumentacja klasy com.mklachl.sopkom.exceptions.PrzystanekNotFoundException	366
6.82.1 Opis szczegółowy	366
6.82.2 Dokumentacja konstruktora i destruktora	366
6.82.2.1 PrzystanekNotFoundException()	366
6.83 Dokumentacja interfejsu com.mklachl.sopkom.repository.PrzystanekRepository	366
6.83.1 Opis szczegółowy	367
6.83.2 Dokumentacja funkcji składowych	367
6.83.2.1 findAll()	367
6.83.2.2 findAllByDlugoscGeograficzna()	367
6.83.2.3 findAllByKodPocztowy()	368
6.83.2.4 findAllByMiasto()	368
6.83.2.5 findAllByNazwaLike()	368
6.83.2.6 findAllByPrzystanekOdwrotny()	369
6.83.2.7 findAllBySzerokoscGeograficzna()	369
6.83.2.8 findAllByUlica()	370
6.83.2.9 findAllByUlicaLike()	370
6.83.2.10 findByNazwa()	370
6.84 Dokumentacja interfejsu com.mklachl.sopkom.services.PrzystanekService	371
6.84.1 Opis szczegółowy	371
6.84.2 Dokumentacja funkcji składowych	371
6.84.2.1 findPrzystanekByDlugoscGeograficzna()	371
6.84.2.2 findPrzystanekById()	372
6.84.2.3 findPrzystanekByKodPocztowy()	372
6.84.2.4 findPrzystanekByMiasto()	373
6.84.2.5 findPrzystanekByNazwa()	373
6.84.2.6 findPrzystanekByPrzystanekOdwrotny()	373
6.84.2.7 findPrzystanekBySzerokoscGeograficzna()	374
6.84.2.8 findPrzystanekByUlica()	374
6.84.2.9 savePrzystanek() [1/2]	375
6.84.2.10 savePrzystanek() [2/2]	375
6.85 Dokumentacja klasy com.mklachl.sopkom.services.impl.PrzystanekServiceImpl	375
6.85.1 Opis szczegółowy	376
6.85.2 Dokumentacja konstruktora i destruktora	376

6.85.2.1 PrzystanekServiceImpl()	376
6.85.3 Dokumentacja funkcji składowych	376
6.85.3.1 findPrzystanekByDlugoscGeograficzna()	376
6.85.3.2 findPrzystanekById()	377
6.85.3.3 findPrzystanekByKodPocztowy()	377
6.85.3.4 findPrzystanekByMiasto()	378
6.85.3.5 findPrzystanekByNazwa()	378
6.85.3.6 findPrzystanekByPrzystanekOdwrotny()	378
6.85.3.7 findPrzystanekBySzerokoscGeograficzna()	379
6.85.3.8 findPrzystanekByUlica()	379
6.85.3.9 savePrzystanek() [1/2]	380
6.85.3.10 savePrzystanek() [2/2]	380
6.86 Dokumentacja klasy com.mklachl.sopkom.model.dto.kurs.PrzystanekWKursieDto	381
6.86.1 Opis szczegółowy	381
6.86.2 Dokumentacja konstruktora i destruktora	381
6.86.2.1 PrzystanekWKursieDto() [1/2]	381
6.86.2.2 PrzystanekWKursieDto() [2/2]	381
6.86.3 Dokumentacja funkcji składowych	382
6.86.3.1 getPrzystanekWLini()	382
6.86.3.2 setPrzystanekWLini()	382
6.87 Dokumentacja klasy com.mklachl.sopkom.model.entity.PrzystanekWlini	382
6.87.1 Opis szczegółowy	383
6.87.2 Dokumentacja funkcji składowych	383
6.87.2.1 getKolejnosc()	383
6.87.2.2 getKursPrzystanekWlinii()	383
6.87.2.3 getLinia()	384
6.87.2.4 getPrzystanek()	384
6.87.2.5 getPrzystanekWlinild()	384
6.87.2.6 setKolejnosc()	384
6.87.2.7 setKursPrzystanekWlinii()	385
6.87.2.8 setLinia()	385
6.87.2.9 setPrzystanek()	385
6.87.2.10 setPrzystanekWlinild()	386
6.88 Dokumentacja interfejsu com.mklachl.sopkom.repository.PrzystanekWliniRepository	386
6.88.1 Opis szczegółowy	386
6.88.2 Dokumentacja funkcji składowych	387
6.88.2.1 findAll()	387
6.88.2.2 findAllByLinia()	387
6.88.2.3 findAllByPrzystanek()	387
6.89 Dokumentacja klasy com.mklachl.sopkom.model.dto.kurs.OLDATED.PrzystanekWKursieDto ← OUTDATED	388
6.89.1 Opis szczegółowy	388

6.89.2 Dokumentacja konstruktora i destruktora	388
6.89.2.1 PrzystenekWKursieDtoOUTDATED() [1/2]	388
6.89.2.2 PrzystenekWKursieDtoOUTDATED() [2/2]	389
6.89.3 Dokumentacja funkcji składowych	389
6.89.3.1 getGodzinna()	389
6.89.3.2 getPrzystanek()	389
6.89.3.3 getPrzystanekkurs_id()	389
6.89.3.4 setGodzinna()	389
6.89.3.5 setPrzystanek()	390
6.89.3.6 setPrzystanekkurs_id()	390
6.89.4 Dokumentacja atrybutów składowych	390
6.89.4.1 godzinna	390
6.89.4.2 kolejnosc	390
6.89.4.3 przystanek	390
6.89.4.4 przystanekkurs_id	391
6.90 Dokumentacja klasy com.mklachl.sopkom.controller.RaportyController	391
6.90.1 Opis szczegółowy	391
6.90.2 Dokumentacja funkcji składowych	391
6.90.2.1 generateRozklad()	391
6.91 Dokumentacja wyliczenia com.mklachl.sopkom.model.entity.Role	392
6.91.1 Opis szczegółowy	392
6.91.2 Dokumentacja atrybutów składowych	392
6.91.2.1 ADMIN	392
6.91.2.2 DISPATCHER	392
6.91.2.3 DRIVER	393
6.91.2.4 MANAGER	393
6.92 Dokumentacja klasy com.mklachl.sopkom.controller.RozkladController	393
6.92.1 Opis szczegółowy	393
6.92.2 Dokumentacja funkcji składowych	393
6.92.2.1 test()	393
6.92.3 Dokumentacja atrybutów składowych	394
6.92.3.1 repo	394
6.92.3.2 rozkladHelper	394
6.93 Dokumentacja klasy com.mklachl.sopkom.model.dto.rozklad.RozkladDto	394
6.93.1 Opis szczegółowy	395
6.93.2 Dokumentacja konstruktora i destruktora	395
6.93.2.1 RozkladDto() [1/2]	395
6.93.2.2 RozkladDto() [2/2]	395
6.93.3 Dokumentacja funkcji składowych	396
6.93.3.1 addKurs()	396
6.93.3.2 getDate()	396
6.93.3.3 getNormalny()	396

6.93.3.4 getRozklad()	397
6.93.3.5 setDate()	397
6.93.3.6 setNormalny()	397
6.93.3.7 setRozklad()	397
6.94 Dokumentacja klasy com.mklachl.sopkom.model.dto.rozklad.RozkladKursDto	399
6.94.1 Opis szczegółowy	399
6.95 Dokumentacja klasy com.mklachl.sopkom.model.dto.rozklad.RozkladLiniaDto	399
6.95.1 Opis szczegółowy	399
6.96 Dokumentacja klasy com.mklachl.sopkom.config.SecurityConfig	399
6.96.1 Opis szczegółowy	400
6.96.2 Dokumentacja funkcji składowych	400
6.96.2.1 authenticationManager()	400
6.96.2.2 passwordEncoder()	400
6.96.2.3 securityFilterChain()	400
6.97 Dokumentacja klasy com.mklachl.sopkom.controller.SimpleMessage	400
6.97.1 Opis szczegółowy	401
6.97.2 Dokumentacja konstruktora i destruktora	401
6.97.2.1 SimpleMessage() [1/2]	401
6.97.2.2 SimpleMessage() [2/2]	401
6.97.3 Dokumentacja funkcji składowych	401
6.97.3.1 getMessage()	401
6.97.3.2 setMessage()	402
6.97.4 Dokumentacja atrybutów składowych	402
6.97.4.1 message	402
6.98 Dokumentacja klasy com.mklachl.sopkom.SOPKoMApplication	402
6.98.1 Opis szczegółowy	402
6.98.2 Dokumentacja funkcji składowych	403
6.98.2.1 main()	403
6.99 Dokumentacja klasy com.mklachl.sopkom.model.entity.User	403
6.99.1 Opis szczegółowy	403
6.99.2 Dokumentacja konstruktora i destruktora	403
6.99.2.1 User() [1/3]	404
6.99.2.2 User() [2/3]	404
6.99.2.3 User() [3/3]	404
6.99.3 Dokumentacja funkcji składowych	405
6.99.3.1 getId()	405
6.99.3.2 getLogin()	405
6.99.3.3 getName()	405
6.99.3.4 getPassword()	406
6.99.3.5 getRoles()	406
6.99.3.6 setId()	406
6.99.3.7 setLogin()	406

6.99.3.8 setName()	407
6.99.3.9 setPassword()	407
6.99.3.10 setRoles()	407
6.100 Dokumentacja klasy com.mklachl.sopkom.controller.UserController	408
6.100.1 Opis szczegółowy	408
6.100.2 Dokumentacja funkcji składowych	408
6.100.2.1 getLoggedInUser()	408
6.101 Dokumentacja klasy com.mklachl.sopkom.model.dto.UserDto	409
6.101.1 Opis szczegółowy	409
6.101.2 Dokumentacja konstruktora i destruktora	409
6.101.2.1 UserDto() [1/2]	409
6.101.2.2 UserDto() [2/2]	409
6.101.3 Dokumentacja funkcji składowych	410
6.101.3.1 getEmail()	410
6.101.3.2 getId()	410
6.101.3.3 getName()	410
6.101.3.4 getPassword()	411
6.101.3.5 getRole()	411
6.101.3.6 getRoles()	411
6.101.3.7 setEmail()	411
6.101.3.8 setId()	412
6.101.3.9 setName()	412
6.101.3.10 setPassword()	412
6.101.3.11 setRole()	413
6.101.3.12 setRoles()	413
6.102 Dokumentacja interfejsu com.mklachl.sopkom.repository.UserRepository	413
6.102.1 Opis szczegółowy	414
6.102.2 Dokumentacja funkcji składowych	414
6.102.2.1 findByLogin()	414
6.103 Dokumentacja interfejsu com.mklachl.sopkom.services.UserService	414
6.103.1 Opis szczegółowy	415
6.103.2 Dokumentacja funkcji składowych	415
6.103.2.1 findUserById()	415
6.103.2.2 findUserByLogin()	415
6.103.2.3 saveUser() [1/2]	416
6.103.2.4 saveUser() [2/2]	416
6.104 Dokumentacja klasy com.mklachl.sopkom.services.impl.UserServiceImpl	416
6.104.1 Opis szczegółowy	417
6.104.2 Dokumentacja konstruktora i destruktora	417
6.104.2.1 UserServiceImpl()	417
6.104.3 Dokumentacja funkcji składowych	417
6.104.3.1 findUserById()	417

6.104.3.2 findUserByLogin()	418
6.104.3.3 saveUser() [1/2]	418
6.104.3.4 saveUser() [2/2]	418
7 Dokumentacja plików	421
7.1 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/config/SecurityConfig.java	421
7.2 SecurityConfig.java	421
7.3 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/controller/AutobusController.java	423
7.4 AutobusController.java	423
7.5 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/controller/Controller.java	424
7.6 Controller.java	425
7.7 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/controller/HarmonogramController.java	425
7.8 HarmonogramController.java	425
7.9 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/controller/IcydentController.java	426
7.10 IcydentController.java	426
7.11 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_← _Backend/src/main/java/com/mklachl/sopkom/controller/KierowcaController.java	428
7.12 KierowcaController.java	429
7.13 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_← _Backend/src/main/java/com/mklachl/sopkom/controller/KursController.java	431
7.14 KursController.java	431
7.15 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_← _Backend/src/main/java/com/mklachl/sopkom/controller/LiniaController.java	435
7.16 LiniaController.java	435
7.17 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_← _Backend/src/main/java/com/mklachl/sopkom/controller/LoginController.java	438
7.18 LoginController.java	438
7.19 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_← _Backend/src/main/java/com/mklachl/sopkom/controller/PrzejazdController.java	438
7.20 PrzejazdController.java	439
7.21 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_← _Backend/src/main/java/com/mklachl/sopkom/controller/PrzystanekController.java	442
7.22 PrzystanekController.java	442
7.23 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_← _Backend/src/main/java/com/mklachl/sopkom/controller/RaportyController.java	444
7.24 RaportyController.java	444
7.25 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_← _Backend/src/main/java/com/mklachl/sopkom/controller/RozkladController.java	445
7.26 RozkladController.java	445
7.27 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_← _Backend/src/main/java/com/mklachl/sopkom/controller/SimpleMessage.java	446

7.28 SimpleMessage.java	446
7.29 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/controller/UserController.java	447
7.30 UserController.java	447
7.31 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/exceptions/IncydentException.java	448
7.32 IncydentException.java	448
7.33 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/exceptions/LiniaNotFoundException.java	448
7.34 LiniaNotFoundException.java	448
7.35 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/exceptions/PrzystanekNotFoundException.java	449
7.36 PrzystanekNotFoundException.java	449
7.37 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/filter/LoggingFilter.java	449
7.38 LoggingFilter.java	450
7.39 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/helpers/Helpers.java	450
7.40 Helpers.java	450
7.41 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/model/dto/AutobusDto.java	451
7.42 AutobusDto.java	451
7.43 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/model/dto/BiletDto.java	452
7.44 BiletDto.java	452
7.45 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/model/dto/DateWrapperDto.java	453
7.46 DateWrapperDto.java	453
7.47 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/model/dto/harmonogram/HarmonogramDto.java	454
7.48 HarmonogramDto.java	454
7.49 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/model/dto/IncydentDto.java	456
7.50 IncydentDto.java	456
7.51 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/model/dto/IncydentDtoOutput.java	458
7.52 IncydentDtoOutput.java	458
7.53 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/model/dto/KierowcaDto.java	459
7.54 KierowcaDto.java	459
7.55 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/model/dto/kurs/AbstractKursDto.java	460
7.56 AbstractKursDto.java	461
7.57 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/model/dto/kurs/AbstractPrzystanekWKursieDto.java	462
7.58 AbstractPrzystanekWKursieDto.java	462

7.59 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/model/dto/kurs/InputKursDto.java	462
7.60 InputKursDto.java	463
7.61 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/model/dto/kurs/InputPrzystanekWKursieDto.java	463
7.62 InputPrzystanekWKursieDto.java	463
7.63 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/model/dto/kurs/KursDto.java	464
7.64 KursDto.java	464
7.65 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/model/dto/kurs/OUTDATED/KursDtoOUTDATED.java	465
7.66 KursDtoOUTDATED.java	465
7.67 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/model/dto/kurs/OUTDATED/PrzystenekWKursie←DtoOUTDATED.java	466
7.68 PrzystenekWKursieDtoOUTDATED.java	467
7.69 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/model/dto/kurs/PrzystanekWKursieDto.java	467
7.70 PrzystanekWKursieDto.java	468
7.71 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/model/dto/linia/LiniaDto.java	468
7.72 LiniaDto.java	468
7.73 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/model/dto/linia/LiniaDtoInput.java	469
7.74 LiniaDtoInput.java	469
7.75 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/model/dto/linia/LiniaDtoOutput.java	470
7.76 LiniaDtoOutput.java	470
7.77 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/model/dto/linia/PrzystanekDtoDlaLinia.java	471
7.78 PrzystanekDtoDlaLinia.java	471
7.79 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/model/dto/linia/PrzystanekDtoDlaLiniaInput.java	472
7.80 PrzystanekDtoDlaLiniaInput.java	472
7.81 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/model/dto/przejazd/PKPWLDtoUpdate.java	473
7.82 Message.java	473
7.83 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/model/dto/przejazd/PKPWLDtoUpdate.java	473
7.84 PKPWLDtoUpdate.java	474
7.85 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/model/dto/przejazd/PrzejazdDtoInput.java	474
7.86 PrzejazdDtoInput.java	474
7.87 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/model/dto/przejazd/PrzejazdDtoOutput.java	475
7.88 PrzejazdDtoOutput.java	475

7.89 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/model/dto/przejazd/PrzejazdDtoUpdate.java	479
7.90 PrzejazdDtoUpdate.java	479
7.91 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/model/dto/PrzejazdBiletDto.java	480
7.92 PrzejazdBiletDto.java	480
7.93 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/model/dto/PrzejazdKursPrzystanekWliniDto.java	481
7.94 PrzejazdKursPrzystanekWliniDto.java	481
7.95 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/model/dto/PrzystanekDto.java	483
7.96 PrzystanekDto.java	483
7.97 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/model/dto/PrzystanekDtoFull.java	484
7.98 PrzystanekDtoFull.java	485
7.99 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM←_Backend/src/main/java/com/mklachl/sopkom/model/dto/rozklad/RozkladDto.java	486
7.100 RozkladDto.java	486
7.101 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/model/dto/rozklad/RozkladKursDto.java	488
7.102 RozkladKursDto.java	488
7.103 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/model/dto/rozklad/RozkladLiniaDto.java	488
7.104 RozkladLiniaDto.java	488
7.105 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/model/dto/UserDto.java	489
7.106 UserDto.java	489
7.107 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/model/entity/Autobus.java	490
7.108 Autobus.java	490
7.109 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/model/entity/Bilet.java	491
7.110 Bilet.java	492
7.111 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/model/entity/Harmonogram.java	492
7.112 Harmonogram.java	493
7.113 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/model/entity/Incydent.java	494
7.114 Incydent.java	495
7.115 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/model/entity/Kierowca.java	496
7.116 Kierowca.java	496
7.117 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/model/entity/Kurs.java	498
7.118 Kurs.java	498

7.119 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/model/entity/KursPrzystanekWlini.java	499
7.120 KursPrzystanekWlini.java	499
7.121 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/model/entity/Linia.java	500
7.122 Linia.java	500
7.123 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/model/entity/Przejazd.java	501
7.124 Przejazd.java	501
7.125 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/model/entity/PrzejazdBilet.java	503
7.126 PrzejazdBilet.java	504
7.127 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/model/entity/PrzejazdKursPrzystanekWlini.java . .	504
7.128 PrzejazdKursPrzystanekWlini.java	505
7.129 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/model/entity/Przystanek.java	505
7.130 Przystanek.java	506
7.131 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/model/entity/PrzystanekWlini.java	507
7.132 PrzystanekWlini.java	507
7.133 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/model/entity/Role.java	508
7.134 Role.java	509
7.135 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/model/entity/User.java	509
7.136 User.java	509
7.137 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/raporty/rozkład/DateHelper.java	510
7.138 DateHelper.java	510
7.139 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/repository/AutobusRepository.java	510
7.140 AutobusRepository.java	511
7.141 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/repository/BiletRepository.java	511
7.142 BiletRepository.java	512
7.143 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/repository/HarmonogramRepository.java	512
7.144 HarmonogramRepository.java	512
7.145 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/repository/IcydentRepository.java	513
7.146 IcydentRepository.java	513
7.147 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/repository/KierowcaRepository.java	513
7.148 KierowcaRepository.java	514

7.149 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/repository/KursPrzystanekWliniRepository.java . . .	514
7.150 KursPrzystanekWliniRepository.java	515
7.151 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/repository/KursRepository.java	515
7.152 KursRepository.java	515
7.153 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/repository/LiniaRepository.java	516
7.154 LiniaRepository.java	516
7.155 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/repository/PrzejazdBiletRepository.java	516
7.156 PrzejazdBiletRepository.java	517
7.157 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/← SOPKoM_Backend/src/main/java/com/mklachl/sopkom/repository/PrzejazdKursPrzystanekWlini← Repository.java	517
7.158 PrzejazdKursPrzystanekWliniRepository.java	517
7.159 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/repository/PrzejazdRepository.java	518
7.160 PrzejazdRepository.java	518
7.161 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/repository/PrzystanekRepository.java	519
7.162 PrzystanekRepository.java	519
7.163 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/repository/PrzystanekWliniRepository.java	520
7.164 PrzystanekWliniRepository.java	520
7.165 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/repository/UserRepository.java	520
7.166 UserRepository.java	521
7.167 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/security/CommandLineRunnerTest.java	521
7.168 CommandLineRunnerTest.java	521
7.169 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/security/CustomUserDetailsService.java	523
7.170 CustomUserDetailsService.java	524
7.171 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/services/AutobusService.java	524
7.172 AutobusService.java	525
7.173 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/services/HarmonogramService.java	525
7.174 HarmonogramService.java	525
7.175 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/services/impl/AutobusServiceImpl.java	526
7.176 AutobusServiceImpl.java	526
7.177 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/services/impl/HarmonogramServiceImpl.java	527
7.178 HarmonogramServiceImpl.java	528

7.179 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/services/impl/IcydentServiceImpl.java	529
7.180 IncydentServiceImpl.java	529
7.181 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/services/impl/KierowcaServiceImpl.java	532
7.182 KierowcaServiceImpl.java	532
7.183 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/services/impl/KursServiceImpl.java	533
7.184 KursServiceImpl.java	533
7.185 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/services/impl/LiniaServiceImpl.java	536
7.186 LiniaServiceImpl.java	536
7.187 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/services/impl/PrzejazdServiceImpl.java	538
7.188 PrzejazdServiceImpl.java	538
7.189 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/services/impl/PrzystanekServiceImpl.java	544
7.190 PrzystanekServiceImpl.java	544
7.191 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/services/impl/UserServiceImpl.java	545
7.192 UserServiceImpl.java	546
7.193 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/services/IcydentService.java	546
7.194 IcydentService.java	547
7.195 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/services/KierowcaService.java	547
7.196 KierowcaService.java	548
7.197 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/services/KursService.java	548
7.198 KursService.java	548
7.199 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/services/LiniaService.java	549
7.200 LiniaService.java	549
7.201 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/services/PrzejazdService.java	550
7.202 PrzejazdService.java	550
7.203 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/services/PrzystanekService.java	551
7.204 PrzystanekService.java	551
7.205 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/services/UserService.java	552
7.206 UserService.java	552
7.207 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/SOPKoMApplication.java	552
7.208 SOPKoMApplication.java	553

7.209 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/main/java/com/mklachl/sopkom/topdf/Pdf.java	553
7.210 Pdf.java	553
7.211 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKo← M_Backend/src/test/java/com/mklachl/sopkom/SOPKoMApplicationTests.java	558
7.212 SOPKoMApplicationTests.java	558
Skorowidz	561

Rozdział 1

Indeks przestrzeni nazw

1.1 Pakiety

Oto lista pakietów wraz z krótkim opisem (o ile jest dostępny):

com.mklachl.sopkom	17
com.mklachl.sopkom.config	17
com.mklachl.sopkom.controller	18
com.mklachl.sopkom.exceptions	18
com.mklachl.sopkom.filter	18
com.mklachl.sopkom.helpers	18
com.mklachl.sopkom.model.dto	18
com.mklachl.sopkom.model.dto.harmonogram	19
com.mklachl.sopkom.model.dto.kurs	19
com.mklachl.sopkom.model.dto.kurs.Obsolete	19
com.mklachl.sopkom.model.dto.linia	20
com.mklachl.sopkom.model.dto.przejazd	20
com.mklachl.sopkom.model.dto.rozklad	20
com.mklachl.sopkom.model.entity	20
com.mklachl.sopkom.repository	21
com.mklachl.sopkom.security	21
com.mklachl.sopkom.services	21
com.mklachl.sopkom.services.impl	22
com.mklachl.sopkom.topdf	22

Rozdział 2

Indeks hierarchiczny

2.1 Hierarchia klas

Ta lista dziedziczenia posortowana jest z grubsza, choć nie całkowicie, alfabetycznie:

com.mklachl.sopkom.model.dto.kurs.AbstractKursDto	23
com.mklachl.sopkom.model.dto.kurs.InputKursDto	150
com.mklachl.sopkom.model.dto.kurs.KursDto	192
com.mklachl.sopkom.model.dto.kurs.AbstractPrzystanekWKursieDto	28
com.mklachl.sopkom.model.dto.kurs.InputPrzystanekWKursieDto	152
com.mklachl.sopkom.model.dto.kurs.PrzystanekWKursieDto	381
com.mklachl.sopkom.model.entity.Autobus	31
com.mklachl.sopkom.controller.AutobusController	35
com.mklachl.sopkom.model.dto.AutobusDto	38
com.mklachl.sopkom.services.AutobusService	50
com.mklachl.sopkom.services.impl.AutobusServiceImpl	55
com.mklachl.sopkom.model.entity.Bilet	60
com.mklachl.sopkom.model.dto.BiletDto	64
com.mklachl.sopkom.controller.Controller	70
com.mklachl.sopkom.model.dto.DateWrapperDto	72
Exception	
com.mklachl.sopkom.exceptions.IncydentException	128
com.mklachl.sopkom.exceptions.LiniaNotFoundException	233
com.mklachl.sopkom.exceptions.PrzystanekNotFoundException	366
com.mklachl.sopkom.model.entity.Harmonogram	74
com.mklachl.sopkom.controller.HarmonogramController	81
com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto	84
com.mklachl.sopkom.services.HarmonogramService	95
com.mklachl.sopkom.services.impl.HarmonogramServiceImpl	100
com.mklachl.sopkom.helpers.Helpers	106
com.mklachl.sopkom.model.entity.Incydent	107
com.mklachl.sopkom.controller.IncydentController	113
com.mklachl.sopkom.model.dto.IncydentDto	119
com.mklachl.sopkom.model.dto.IncydentDtoOutput	125
com.mklachl.sopkom.services.IncydentService	135
com.mklachl.sopkom.services.impl.IncydentServiceImpl	142
com.mklachl.sopkom.model.entity.Kierowca	154
com.mklachl.sopkom.controller.KierowcaController	158

com.mklachl.sopkom.model.dto.KierowcaDto	164
com.mklachl.sopkom.services.KierowcaService	174
com.mklachl.sopkom.services.impl.KierowcaServiceImpl	178
com.mklachl.sopkom.model.entity.Kurs	183
com.mklachl.sopkom.controller.KursController	187
com.mklachl.sopkom.model.dto.kurs.OLDATED.KursDtoOLDATED	196
com.mklachl.sopkom.model.entity.KursPrzystanekWlini	200
com.mklachl.sopkom.services.KursService	210
com.mklachl.sopkom.services.impl.KursServiceImpl	213
com.mklachl.sopkom.model.entity.Linia	218
com.mklachl.sopkom.controller.LiniaController	221
com.mklachl.sopkom.model.dto.linia.LiniaDto	226
com.mklachl.sopkom.model.dto.linia.LiniaDtoInput	229
com.mklachl.sopkom.model.dto.linia.LiniaDtoOutput	230
com.mklachl.sopkom.services.LiniaService	236
com.mklachl.sopkom.services.impl.LiniaServiceImpl	238
com.mklachl.sopkom.controller.LoginController	242
com.mklachl.sopkom.model.dto.Message	243
com.mklachl.sopkom.topdf.Pdf	246
com.mklachl.sopkom.model.dto.przejazd.PKPWLDDtoUpdate	248
com.mklachl.sopkom.model.entity.Przejazd	250
com.mklachl.sopkom.model.entity.PrzejazdBilet	257
com.mklachl.sopkom.model.dto.PrzejazdBiletDto	261
com.mklachl.sopkom.controller.PrzejazdController	268
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoInput	273
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput	277
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate	287
com.mklachl.sopkom.model.entity.PrzejazdKursPrzystanekWlini	293
com.mklachl.sopkom.model.dto.PrzejazdKursPrzystanekWliniDto	295
com.mklachl.sopkom.services.PrzejazdService	313
com.mklachl.sopkom.services.impl.PrzejazdServiceImpl	324
com.mklachl.sopkom.model.entity.Przystanek	337
com.mklachl.sopkom.controller.PrzystanekController	342
com.mklachl.sopkom.model.dto.PrzystanekDto	346
com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLiniaInput	355
com.mklachl.sopkom.model.dto.PrzystanekDtoFull	358
com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLinia	352
com.mklachl.sopkom.services.PrzystanekService	371
com.mklachl.sopkom.services.impl.PrzystanekServiceImpl	375
com.mklachl.sopkom.model.entity.PrzystanekWlini	382
com.mklachl.sopkom.model.dto.kurs.OLDATED.PrzystenekWKursieDtoOLDATED	388
com.mklachl.sopkom.controller.RaportyController	391
com.mklachl.sopkom.model.entity.Role	392
com.mklachl.sopkom.controller.RozkladController	393
com.mklachl.sopkom.model.dto.rozklad.RozkladDto	394
com.mklachl.sopkom.model.dto.rozklad.RozkladKursDto	399
com.mklachl.sopkom.model.dto.rozklad.RozkladLiniaDto	399
com.mklachl.sopkom.config.SecurityConfig	399
com.mklachl.sopkom.controller.SimpleMessage	400
com.mklachl.sopkom.SOPKoMApplication	402
com.mklachl.sopkom.model.entity.User	403
com.mklachl.sopkom.controller.UserController	408
com.mklachl.sopkom.model.dto.UserDto	409
com.mklachl.sopkom.services.UserService	414
com.mklachl.sopkom.services.impl.UserServiceImpl	416
CommandLineRunner	

com.mklachl.sopkom.security.CommandLineRunnerTest	68
CrudRepository	
com.mklachl.sopkom.repository.AutobusRepository	43
com.mklachl.sopkom.repository.BiletRepository	66
com.mklachl.sopkom.repository.HarmonogramRepository	91
com.mklachl.sopkom.repository.IncydentRepository	129
com.mklachl.sopkom.repository.KierowcaRepository	169
com.mklachl.sopkom.repository.KursPrzystanekWliniRepository	204
com.mklachl.sopkom.repository.KursRepository	207
com.mklachl.sopkom.repository.LiniaRepository	234
com.mklachl.sopkom.repository.PrzejazdBiletRepository	265
com.mklachl.sopkom.repository.PrzejazdKursPrzystanekWliniRepository	300
com.mklachl.sopkom.repository.PrzejazdRepository	303
com.mklachl.sopkom.repository.PrzystanekRepository	366
com.mklachl.sopkom.repository.PrzystanekWliniRepository	386
com.mklachl.sopkom.repository.UserRepository	413
OncePerRequestFilter	
com.mklachl.sopkom.filter.LoggingFilter	241
UserDetailsService	
com.mklachl.sopkom.security.CustomUserDetailsService	71

Rozdział 3

Indeks klas

3.1 Lista klas

Tutaj znajdują się klasy, struktury, unie i interfejsy wraz z ich krótkimi opisami:

com.mklachl.sopkom.model.dto.kurs.AbstractKursDto	23
com.mklachl.sopkom.model.dto.kurs.AbstractPrzystanekWKursieDto	28
com.mklachl.sopkom.model.entity.Autobus	31
com.mklachl.sopkom.controller.AutobusController	35
com.mklachl.sopkom.model.dto.AutobusDto	38
com.mklachl.sopkom.repository.AutobusRepository	43
com.mklachl.sopkom.services.AutobusService	50
com.mklachl.sopkom.services.impl.AutobusServiceImpl	55
com.mklachl.sopkom.model.entity.Bilet	60
com.mklachl.sopkom.model.dto.BiletDto	64
com.mklachl.sopkom.repository.BiletRepository	66
com.mklachl.sopkom.security.CommandLineRunnerTest	68
com.mklachl.sopkom.controller.Controller	70
com.mklachl.sopkom.security.CustomUserDetailsService	71
com.mklachl.sopkom.model.dto.DateWrapperDto	72
com.mklachl.sopkom.model.entity.Harmonogram	74
com.mklachl.sopkom.controller.HarmonogramController	81
com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto	84
com.mklachl.sopkom.repository.HarmonogramRepository	91
com.mklachl.sopkom.services.HarmonogramService	95
com.mklachl.sopkom.services.impl.HarmonogramServiceImpl	100
com.mklachl.sopkom.helpers.Helpers	106
com.mklachl.sopkom.model.entity.Incydent	107
com.mklachl.sopkom.controller.IncydentController	113
com.mklachl.sopkom.model.dto.IncydentDto	119
com.mklachl.sopkom.model.dto.IncydentDtoOutput	125
com.mklachl.sopkom.exceptions.IncydentException	128
com.mklachl.sopkom.repository.IncydentRepository	129
com.mklachl.sopkom.services.IncydentService	135
com.mklachl.sopkom.services.impl.IncydentServiceImpl	142
com.mklachl.sopkom.model.dto.kurs.InputKursDto	150
com.mklachl.sopkom.model.dto.kurs.InputPrzystanekWKursieDto	152
com.mklachl.sopkom.model.entity.Kierowca	154
com.mklachl.sopkom.controller.KierowcaController	158
com.mklachl.sopkom.model.dto.KierowcaDto	164

com.mklachl.sopkom.repository.KierowcaRepository	169
com.mklachl.sopkom.services.KierowcaService	174
com.mklachl.sopkom.services.impl.KierowcaServiceImpl	178
com.mklachl.sopkom.model.entity.Kurs	183
com.mklachl.sopkom.controller.KursController	187
com.mklachl.sopkom.model.dto.kurs.KursDto	192
com.mklachl.sopkom.model.dto.kurs.UPDATED.KursDtoUPDATED	196
com.mklachl.sopkom.model.entity.KursPrzystanekWlini	200
com.mklachl.sopkom.repository.KursPrzystanekWliniRepository	204
com.mklachl.sopkom.repository.KursRepository	207
com.mklachl.sopkom.services.KursService	210
com.mklachl.sopkom.services.impl.KursServiceImpl	213
com.mklachl.sopkom.model.entity.Linia	218
com.mklachl.sopkom.controller.LiniaController	221
com.mklachl.sopkom.model.dto.linia.LiniaDto	226
com.mklachl.sopkom.model.dto.linia.LiniaDtoInput	229
com.mklachl.sopkom.model.dto.linia.LiniaDtoOutput	230
com.mklachl.sopkom.exceptions.LiniaNotFoundException	233
com.mklachl.sopkom.repository.LiniaRepository	234
com.mklachl.sopkom.services.LiniaService	236
com.mklachl.sopkom.services.impl.LiniaServiceImpl	238
com.mklachl.sopkom.filter.LoggingFilter	241
com.mklachl.sopkom.controller.LoginController	242
com.mklachl.sopkom.model.dto.Message	243
com.mklachl.sopkom.topdf.Pdf	246
com.mklachl.sopkom.model.dto.przejazd.PKPWLDtoUpdate	248
com.mklachl.sopkom.model.entity.Przejazd	250
com.mklachl.sopkom.model.entity.PrzejazdBilet	257
com.mklachl.sopkom.model.dto.PrzejazdBiletDto	261
com.mklachl.sopkom.repository.PrzejazdBiletRepository	265
com.mklachl.sopkom.controller.PrzejazdController	268
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoInput	273
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput	277
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate	287
com.mklachl.sopkom.model.entity.PrzejazdKursPrzystanekWlini	293
com.mklachl.sopkom.model.dto.PrzejazdKursPrzystanekWliniDto	295
com.mklachl.sopkom.repository.PrzejazdKursPrzystanekWliniRepository	300
com.mklachl.sopkom.repository.PrzejazdRepository	303
com.mklachl.sopkom.services.PrzejazdService	313
com.mklachl.sopkom.services.impl.PrzejazdServiceImpl	324
com.mklachl.sopkom.model.entity.Przystanek	337
com.mklachl.sopkom.controller.PrzystanekController	342
com.mklachl.sopkom.model.dto.PrzystanekDto	346
com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLinia	352
com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLinialInput	355
com.mklachl.sopkom.model.dto.PrzystanekDtoFull	358
com.mklachl.sopkom.exceptions.PrzystanekNotFoundException	366
com.mklachl.sopkom.repository.PrzystanekRepository	366
com.mklachl.sopkom.services.PrzystanekService	371
com.mklachl.sopkom.services.impl.PrzystanekServiceImpl	375
com.mklachl.sopkom.model.dto.kurs.PrzystanekWKursieDto	381
com.mklachl.sopkom.model.entity.PrzystanekWlini	382
com.mklachl.sopkom.repository.PrzystanekWliniRepository	386
com.mklachl.sopkom.model.dto.kurs.UPDATED.PrzystanekWKursieDtoUPDATED	388
com.mklachl.sopkom.controller.RaportyController	391
com.mklachl.sopkom.model.entity.Role	392
com.mklachl.sopkom.controller.RozkladController	393
com.mklachl.sopkom.model.dto.rozklad.RozkladDto	394

com.mklachl.sopkom.model.dto.rozklad.RozkladKursDto	399
com.mklachl.sopkom.model.dto.rozklad.RozkladLiniaDto	399
com.mklachl.sopkom.config.SecurityConfig	399
com.mklachl.sopkom.controller.SimpleMessage	400
com.mklachl.sopkom.SOPKoMApplication	402
com.mklachl.sopkom.model.entity.User	403
com.mklachl.sopkom.controller.UserController	408
com.mklachl.sopkom.model.dto.UserDto	409
com.mklachl.sopkom.repository.UserRepository	413
com.mklachl.sopkom.services.UserService	414
com.mklachl.sopkom.services.impl.UserServiceImpl	416

Rozdział 4

Indeks plików

4.1 Lista plików

Tutaj znajduje się lista wszystkich plików z ich krótkimi opisami:

X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/contr⁵⁵²
X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/config⁴²¹
X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/contr⁴²³
X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/contr⁴²⁴
X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/contr⁴²⁵
X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/contr⁴²⁶
X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/contr⁴²⁸
X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/contr⁴³¹
X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/contr⁴³⁵
X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/contr⁴³⁸
X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/contr⁴³⁸
X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/contr⁴⁴²
X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/contr⁴⁴⁴
X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/contr⁴⁴⁵
X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/contr⁴⁴⁶
X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/contr⁴⁴⁷
X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/excep⁴⁴⁸
X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/excep⁴⁴⁸

X:/java/TAB/Doxxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/test/java/com/mklachl/sopkom/SOPKoMBackendTest.java
558

Rozdział 5

Dokumentacja przestrzeni nazw

5.1 Pakiet com.mklachl.sopkom

Pakiety

- package config
- package controller
- package exceptions
- package filter
- package helpers
- package repository
- package security
- package services
- package topdf

Komponenty

- class SOPKoMApplication
- class SOPKoMApplicationTests

5.2 Pakiet com.mklachl.sopkom.config

Komponenty

- class SecurityConfig

5.3 Pakiet com.mklachl.sopkom.controller

Komponenty

- class AutobusController
- class Controller
- class HarmonogramController
- class IncydentController
- class KierowcaController
- class KursController
- class LiniaController
- class LoginController
- class PrzejazdController
- class PrzystanekController
- class RaportyController
- class RozkladController
- class SimpleMessage
- class UserController

5.4 Pakiet com.mklachl.sopkom.exceptions

Komponenty

- class IncydentException
- class LiniaNotFoundException
- class PrzystanekNotFoundException

5.5 Pakiet com.mklachl.sopkom.filter

Komponenty

- class LoggingFilter

5.6 Pakiet com.mklachl.sopkom.helpers

Komponenty

- class Helpers

5.7 Pakiet com.mklachl.sopkom.model.dto

Pakiety

- package harmonogram
- package kurs
- package linia
- package przejazd
- package rozklad

Komponenty

- class AutobusDto
- class BiletDto
- class DateWrapperDto
- class IncydentDto
- class IncydentDtoOutput
- class KierowcaDto
- class Message
- class PrzejazdBiletDto
- class PrzejazdKursPrzystanekWliniDto
- class PrzystanekDto
- class PrzystanekDtoFull
- class UserDto

5.8 Pakiet com.mklachl.sopkom.model.dto.harmonogram

Komponenty

- class HarmonogramDto

5.9 Pakiet com.mklachl.sopkom.model.dto.kurs

Pakiety

- package OUTDATED

Komponenty

- class AbstractKursDto
- class AbstractPrzystanekWKursieDto
- class InputKursDto
- class InputPrzystanekWKursieDto
- class KursDto
- class PrzystanekWKursieDto

5.10 Pakiet com.mklachl.sopkom.model.dto.kurs.OUTDATED

Komponenty

- class KursDtoOUTDATED
- class PrzystenekWKursieDtoOUTDATED

5.11 Pakiet com.mklachl.sopkom.model.dto.linia

Komponenty

- class [LiniaDto](#)
- class [LiniaDtoInput](#)
- class [LiniaDtoOutput](#)
- class [PrzystanekDtoDlaLinia](#)
- class [PrzystanekDtoDlaLinialInput](#)

5.12 Pakiet com.mklachl.sopkom.model.dto.przejazd

Komponenty

- class [PKPWLDtoUpdate](#)
- class [PrzejazdDtoInput](#)
- class [PrzejazdDtoOutput](#)
- class [PrzejazdDtoUpdate](#)

5.13 Pakiet com.mklachl.sopkom.model.dto.rozklad

Komponenty

- class [RozkladDto](#)
- class [RozkladKursDto](#)
- class [RozkladLiniaDto](#)

5.14 Pakiet com.mklachl.sopkom.model.entity

Komponenty

- class [Autobus](#)
- class [Bilet](#)
- class [Harmonogram](#)
- class [Incydent](#)
- class [Kierowca](#)
- class [Kurs](#)
- class [KursPrzystanekWlini](#)
- class [Linia](#)
- class [Przejazd](#)
- class [PrzejazdBilet](#)
- class [PrzejazdKursPrzystanekWlini](#)
- class [Przystanek](#)
- class [PrzystanekWlini](#)
- enum [Role](#)
- class [User](#)

5.15 Pakiet com.mklachl.sopkom.repository

Komponenty

- interface AutobusRepository
- interface BiletRepository
- interface HarmonogramRepository
- interface IncydentRepository
- interface KierowcaRepository
- interface KursPrzystanekWliniRepository
- interface KursRepository
- interface LiniaRepository
- interface PrzejazdBiletRepository
- interface PrzejazdKursPrzystanekWliniRepository
- interface PrzejazdRepository
- interface PrzystanekRepository
- interface PrzystanekWliniRepository
- interface UserRepository

5.16 Pakiet com.mklachl.sopkom.security

Komponenty

- class CommandLineRunnerTest
- class CustomUserDetailsService

5.17 Pakiet com.mklachl.sopkom.services

Pakiet

- package impl

Komponenty

- interface AutobusService
- interface HarmonogramService
- interface IncydentService
- interface KierowcaService
- interface KursService
- interface LiniaService
- interface PrzejazdService
- interface PrzystanekService
- interface UserService

5.18 Pakiet com.mklachl.sopkom.services.impl

Komponenty

- class [AutobusServiceImpl](#)
- class [HarmonogramServiceImpl](#)
- class [IncydentServiceImpl](#)
- class [KierowcaServiceImpl](#)
- class [KursServiceImpl](#)
- class [LiniaServiceImpl](#)
- class [PrzejazdServiceImpl](#)
- class [PrzystanekServiceImpl](#)
- class [UserServiceImpl](#)

5.19 Pakiet com.mklachl.sopkom.topdf

Komponenty

- class [Pdf](#)

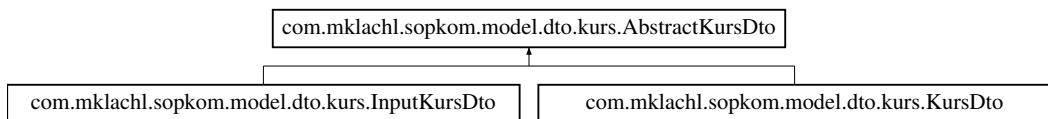
Rozdział 6

Dokumentacja klas

6.1 Dokumentacja klas

com.mklachl.sopkom.model.dto.kurs.AbstractKursDto

Diagram dziedziczenia dla com.mklachl.sopkom.model.dto.kurs.AbstractKursDto



Metody publiczne

- `AbstractKursDto (Kurs kurs)`
- `AbstractKursDto ()`
- `Long getKursId ()`
- `void setKursId (Long kursId)`
- `Long getLiniald ()`
- `void setLiniald (Long liniald)`
- `Short getHarmonogramId ()`
- `void setHarmonogramId (Short harmonogramId)`
- `Short getKierunek ()`
- `void setKierunek (Short kierunek)`
- `Short getTyp_autobusu ()`
- `void setTyp_autobusu (Short typ_autobusu)`

Atrybuty chronione

- `Long kursId`
- `Long liniald`
- `Short harmonogramId`
- `Short kierunek`
- `Short typ_autobusu`

6.1.1 Opis szczegółowy

Abstrakcyjna klasa Data Transfer Object (DTO) dla encji Kurs.

Definicja w linii 8 pliku [AbstractKursDto.java](#).

6.1.2 Dokumentacja konstruktora i destruktora

6.1.2.1 AbstractKursDto() [1/2]

```
com.mklachl.sopkom.model.dto.kurs.AbstractKursDto.AbstractKursDto (   
    Kurs kurs )
```

Konstruktor przyjmujący obiekt encji Kurs.

Parametry

<i>kurs</i>	Obiekt encji Kurs.
-------------	--------------------

Definicja w linii 39 pliku [AbstractKursDto.java](#).

6.1.2.2 AbstractKursDto() [2/2]

```
com.mklachl.sopkom.model.dto.kurs.AbstractKursDto.AbstractKursDto ( )
```

Konstruktor domyślny.

Definicja w linii 48 pliku [AbstractKursDto.java](#).

6.1.3 Dokumentacja funkcji składowych

6.1.3.1 getHarmonogramId()

```
Short com.mklachl.sopkom.model.dto.kurs.AbstractKursDto.getHarmonogramId ( )
```

Zwraca identyfikator harmonogramu związanego z kursem.

Zwrota

harmonogramId Identyfikator harmonogramu.

Definicja w linii 89 pliku [AbstractKursDto.java](#).

6.1.3.2 getKierunek()

Short com.mklachl.sopkom.model.dto.kurs.AbstractKursDto.getKierunek ()

Zwraca kierunek kursu.

Zwraca

kierunek Kierunek kursu.

Definicja w linii 105 pliku [AbstractKursDto.java](#).

6.1.3.3 getKursId()

Long com.mklachl.sopkom.model.dto.kurs.AbstractKursDto.getKursId ()

Zwraca identyfikator kursu.

Zwraca

kursId Identyfikator kursu.

Definicja w linii 57 pliku [AbstractKursDto.java](#).

6.1.3.4 getLiniaId()

Long com.mklachl.sopkom.model.dto.kurs.AbstractKursDto.getLiniaId ()

Zwraca identyfikator linii związanej z kursem.

Zwraca

liniaId Identyfikator linii.

Definicja w linii 73 pliku [AbstractKursDto.java](#).

6.1.3.5 getTyp_autobusu()

Short com.mklachl.sopkom.model.dto.kurs.AbstractKursDto.getTyp_autobusu ()

Zwraca typ autobusu używanego w kursie.

Zwraca

typ_autobusu Typ autobusu.

Definicja w linii 121 pliku [AbstractKursDto.java](#).

6.1.3.6 setHarmonogramId()

```
void com.mklachl.sopkom.model.dto.kurs.AbstractKursDto.setHarmonogramId (
    Short harmonogramId )
```

Ustawia identyfikator harmonogramu związanego z kursem.

Parametry

<i>harmonogram</i> ↪ <i>Id</i>	Identyfikator harmonogramu.
-----------------------------------	-----------------------------

Definicja w linii 97 pliku [AbstractKursDto.java](#).

6.1.3.7 setKierunek()

```
void com.mklachl.sopkom.model.dto.kurs.AbstractKursDto.setKierunek (  
    Short kierunek )
```

Ustawia kierunek kursu.

Parametry

<i>kierunek</i>	Kierunek kursu.
-----------------	-----------------

Definicja w linii 113 pliku [AbstractKursDto.java](#).

6.1.3.8 setKursId()

```
void com.mklachl.sopkom.model.dto.kurs.AbstractKursDto.setKursId (   
    Long kursId )
```

Ustawia identyfikator kursu.

Parametry

<i>kurs</i> ↪ <i>Id</i>	Identyfikator kursu.
----------------------------	----------------------

Definicja w linii 65 pliku [AbstractKursDto.java](#).

6.1.3.9 setLiniaId()

```
void com.mklachl.sopkom.model.dto.kurs.AbstractKursDto.setLiniaId (   
    Long liniaId )
```

Ustawia identyfikator linii związanej z kursem.

Parametry

<i>linia</i> ↵	Identyfikator linii.
<i>Id</i>	

Definicja w linii 81 pliku [AbstractKursDto.java](#).

6.1.3.10 setTyp_autobusu()

```
void com.mklachl.sopkom.model.dto.kurs.AbstractKursDto.setTyp_autobusu (
    Short typ_autobusu )
```

Ustawia typ autobusu używanego w kursie.

Parametry

<i>typ_autobusu</i>	Typ autobusu.
---------------------	---------------

Definicja w linii 129 pliku [AbstractKursDto.java](#).

6.1.4 Dokumentacja atrybutów składowych**6.1.4.1 harmonogramId**

```
Short com.mklachl.sopkom.model.dto.kurs.AbstractKursDto.harmonogramId [protected]
```

Identyfikator harmonogramu związanego z kursem.

Definicja w linii 23 pliku [AbstractKursDto.java](#).

6.1.4.2 kierunek

```
Short com.mklachl.sopkom.model.dto.kurs.AbstractKursDto.kierunek [protected]
```

Kierunek kursu.

Definicja w linii 28 pliku [AbstractKursDto.java](#).

6.1.4.3 kursId

Long com.mklachl.sopkom.model.dto.kurs.AbstractKursDto.kursId [protected]

Identyfikator kursu.

Definicja w linii 13 pliku [AbstractKursDto.java](#).

6.1.4.4 liniaId

Long com.mklachl.sopkom.model.dto.kurs.AbstractKursDto.liniaId [protected]

Identyfikator linii związanej z kursem.

Definicja w linii 18 pliku [AbstractKursDto.java](#).

6.1.4.5 typ_autobusu

Short com.mklachl.sopkom.model.dto.kurs.AbstractKursDto.typ_autobusu [protected]

Typ autobusu używanego w kursie.

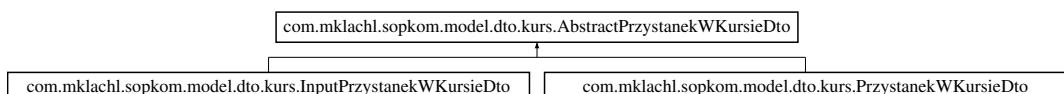
Definicja w linii 33 pliku [AbstractKursDto.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.2 Dokumentacja klasy com.mklachl.sopkom.model.dto.kurs.AbstractPrzystanekWKursieDto

Diagram dziedziczenia dla com.mklachl.sopkom.model.dto.kurs.AbstractPrzystanekWKursieDto



Metody publiczne

- [AbstractPrzystanekWKursieDto \(\)](#)
- [AbstractPrzystanekWKursieDto \(KursPrzystanekWlini kpw\)](#)
- Long [getPrzystanekwKursield \(\)](#)
- void [setPrzystanekwKursield \(Long przystanekwKursield\)](#)
- Time [getGodzinna \(\)](#)
- void [setGodzinna \(Time godzinna\)](#)

Atrybuty chronione

- Long przystanekwKursield
- Time godzinna

6.2.1 Opis szczegółowy

Abstrakcyjna klasa Data Transfer Object (DTO) dla encji KursPrzystanekWlini.

Definicja w linii 10 pliku [AbstractPrzystanekWKursieDto.java](#).

6.2.2 Dokumentacja konstruktora i destruktora

6.2.2.1 AbstractPrzystanekWKursieDto() [1/2]

```
com.mklachl.sopkom.model.dto.kurs.AbstractPrzystanekWKursieDto.AbstractPrzystanekWKursieDto ( )
```

Konstruktor domyślny.

Definicja w linii 26 pliku [AbstractPrzystanekWKursieDto.java](#).

6.2.2.2 AbstractPrzystanekWKursieDto() [2/2]

```
com.mklachl.sopkom.model.dto.kurs.AbstractPrzystanekWKursieDto.AbstractPrzystanekWKursieDto ( KursPrzystanekWlini kpw )
```

Konstruktor przyjmujący obiekt encji KursPrzystanekWlini.

Parametry

<i>kpw</i>	Obiekt encji KursPrzystanekWlini.
------------	-----------------------------------

Definicja w linii 33 pliku [AbstractPrzystanekWKursieDto.java](#).

6.2.3 Dokumentacja funkcji składowych

6.2.3.1 getGodzinna()

```
Time com.mklachl.sopkom.model.dto.kurs.AbstractPrzystanekWKursieDto.getGodzinna ( )
```

Zwraca godzinę przystanku.

Zwrota

godzinna Godzina przystanku.

Definicja w linii 60 pliku [AbstractPrzystanekWKursieDto.java](#).

6.2.3.2 getPrzystanekwKursield()

```
Long com.mklachl.sopkom.model.dto.kurs.AbstractPrzystanekWKursieDto.getPrzystanekwKursieId ( )
```

Zwraca identyfikator przystanku w kursie.

Zwrota

przystanekwKursield Identyfikator przystanku w kursie.

Definicja w linii 44 pliku [AbstractPrzystanekWKursieDto.java](#).

6.2.3.3 setGodzinna()

```
void com.mklachl.sopkom.model.dto.kurs.AbstractPrzystanekWKursieDto.setGodzinna ( Time godzinna )
```

Ustawia godzinę przystanku.

Parametry

godzinna	Godzina przystanku.
----------	---------------------

Definicja w linii 68 pliku [AbstractPrzystanekWKursieDto.java](#).

6.2.3.4 setPrzystanekwKursield()

```
void com.mklachl.sopkom.model.dto.kurs.AbstractPrzystanekWKursieDto.setPrzystanekwKursieId ( Long przystanekwKursieId )
```

Ustawia identyfikator przystanku w kursie.

Parametry

<i>przystanekwKursield</i>	Identyfikator przystanku w kursie.
----------------------------	------------------------------------

Definicja w linii 52 pliku [AbstractPrzystanekWKursieDto.java](#).

6.2.4 Dokumentacja atrybutów składowych

6.2.4.1 godzinna

Time com.mklachl.sopkom.model.dto.kurs.AbstractPrzystanekWKursieDto.godzinna [protected]

Godzina przystanku w formacie "hh:MM:ss".

Definicja w linii 21 pliku [AbstractPrzystanekWKursieDto.java](#).

6.2.4.2 przystanekwKursield

Long com.mklachl.sopkom.model.dto.kurs.AbstractPrzystanekWKursieDto.przystanekwKursieId [protected]

Identyfikator przystanku w kursie.

Definicja w linii 15 pliku [AbstractPrzystanekWKursieDto.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.3 Dokumentacja klasy com.mklachl.sopkom.model.entity.Autobus

Metody publiczne

- Long [getAutbousId](#) ()
- String [getNumerRejestracyjny](#) ()
- Date [getPrzegladWaznyDo](#) ()
- String [getStatus](#) ()
- float [getPrzebieg](#) ()
- void [setAutbousId](#) (Long autbousId)
- void [setNumerRejestracyjny](#) (String numerRejestracyjny)
- void [setPrzegladWaznyDo](#) (Date przegladWaznyDo)
- void [setStatus](#) (String status)
- void [setPrzebieg](#) (float przebieg)

6.3.1 Opis szczegółowy

Klasa reprezentująca encję [Autobus](#).

Definicja w linii [18](#) pliku [Autobus.java](#).

6.3.2 Dokumentacja funkcji składowych

6.3.2.1 getAutbousId()

```
Long com.mklachl.sopkom.model.entity.Autobus.getAutbousId ( )
```

Pobiera unikalny identyfikator autobusu.

Zwrota

Unikalny identyfikator autobusu.

Definicja w linii [59](#) pliku [Autobus.java](#).

6.3.2.2 getNumerRejestracyjny()

```
String com.mklachl.sopkom.model.entity.Autobus.getNumerRejestracyjny ( )
```

Pobiera numer rejestracyjny autobusu.

Zwrota

Numer rejestracyjny autobusu.

Definicja w linii [67](#) pliku [Autobus.java](#).

6.3.2.3 getPrzebieg()

```
float com.mklachl.sopkom.model.entity.Autobus.getPrzebieg ( )
```

Pobiera przebieg autobusu.

Zwrota

Przebieg autobusu.

Definicja w linii [91](#) pliku [Autobus.java](#).

6.3.2.4 getPrzegladWaznyDo()

```
Date com.mklachl.sopkom.model.entity.Autobus.getPrzegladWaznyDo ( )
```

Pobiera datę ważności przeglądu technicznego.

Zwroca

Data ważności przeglądu technicznego.

Definicja w linii 75 pliku [Autobus.java](#).

6.3.2.5 getStatus()

```
String com.mklachl.sopkom.model.entity.Autobus.getStatus ( )
```

Pobiera status autobusu.

Zwroca

Status autobusu.

Definicja w linii 83 pliku [Autobus.java](#).

6.3.2.6 setAutbousId()

```
void com.mklachl.sopkom.model.entity.Autobus.setAutbousId ( Long autbousId )
```

Ustawia unikalny identyfikator autobusu.

Parametry

<i>autbous</i> <i>Id</i>	Unikalny identyfikator autobusu.
-----------------------------	----------------------------------

Definicja w linii 101 pliku [Autobus.java](#).

6.3.2.7 setNumerRejestracyjny()

```
void com.mklachl.sopkom.model.entity.Autobus.setNumerRejestracyjny ( String numerRejestracyjny )
```

Ustawia numer rejestracyjny autobusu.

Parametry

<i>numerRejestracyjny</i>	Numer rejestracyjny autobusu.
---------------------------	-------------------------------

Definicja w linii 109 pliku [Autobus.java](#).

6.3.2.8 setPrzebieg()

```
void com.mklachl.sopkom.model.entity.Autobus.setPrzebieg (
    float przebieg )
```

Ustawia przebieg autobusu.

Parametry

<i>przebieg</i>	Przebieg autobusu.
-----------------	--------------------

Definicja w linii 133 pliku [Autobus.java](#).

6.3.2.9 setPrzegladWaznyDo()

```
void com.mklachl.sopkom.model.entity.Autobus.setPrzegladWaznyDo (
    Date przegladWaznyDo )
```

Ustawia datę ważności przeglądu technicznego.

Parametry

<i>przegladWaznyDo</i>	Data ważności przeglądu technicznego.
------------------------	---------------------------------------

Definicja w linii 117 pliku [Autobus.java](#).

6.3.2.10 setStatus()

```
void com.mklachl.sopkom.model.entity.Autobus.setStatus (
    String status )
```

Ustawia status autobusu.

Parametry

<i>status</i>	Status autobusu.
---------------	------------------

Definicja w linii 125 pliku Autobus.java.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.4 Dokumentacja klasy com.mklachl.sopkom.controller.AutobusController

Metody publiczne

- ResponseEntity< String > test ()
- ResponseEntity<?> template ()
- ResponseEntity<?> addAutobus (@RequestBody AutobusDto autobusDto)
- ResponseEntity<?> getAutobus (@PathVariable("id") Long id)
- ResponseEntity<?> deleteAutobus (@PathVariable("id") Long id)
- ResponseEntity<?> updateAutobus (@PathVariable("id") Long id, @RequestBody AutobusDto autobusDto)
- ResponseEntity<?> getAllBuses ()
- ResponseEntity<?> getAllBusesFilteredStatus (@RequestParam(name="status", required=false) String status)

6.4.1 Opis szczegółowy

Definicja w linii 26 pliku AutobusController.java.

6.4.2 Dokumentacja funkcji składowych

6.4.2.1 addAutobus()

```
ResponseEntity<?> com.mklachl.sopkom.controller.AutobusController.addAutobus (   
    @RequestBody AutobusDto autobusDto )
```

Tworzy nowy autobus.

Parametry

autobusDto	Obiekt DTO autobusu.
------------	----------------------

Zwraca

Utworzony obiekt DTO autobusu.

Definicja w linii 64 pliku AutobusController.java.

6.4.2.2 deleteAutobus()

```
ResponseEntity<?> com.mklachl.sopkom.controller.AutobusController.deleteAutobus ( @PathVariable("id") Long id )
```

Usuwa autobus po identyfikatorze.

Parametry

<i>id</i>	Identyfikator autobusu.
-----------	-------------------------

Zwroca

Wiadomość o statusie operacji.

Definicja w linii [89](#) pliku [AutobusController.java](#).

6.4.2.3 getAllBuses()

```
ResponseEntity<?> com.mklachl.sopkom.controller.AutobusController.getAllBuses ( )
```

Pobiera wszystkie autobusy.

Zwroca

Lista obiektów DTO autobusów.

Definicja w linii [123](#) pliku [AutobusController.java](#).

6.4.2.4 getAllBusesFilteredStatus()

```
ResponseEntity<?> com.mklachl.sopkom.controller.AutobusController.getAllBusesFilteredStatus ( @RequestParam(name="status", required=false) String status )
```

Pobiera autobusy filtrowane po statusie.

Parametry

<i>status</i>	Status autobusu.
---------------	------------------

Zwroca

Lista obiektów DTO autobusów.

Definicja w linii [135](#) pliku [AutobusController.java](#).

6.4.2.5 getAutobus()

```
ResponseEntity<?> com.mklachl.sopkom.controller.AutobusController.getAutobus ( @PathVariable("id") Long id )
```

Pobiera autobus po identyfikatorze.

Parametry

<i>id</i>	Identyfikator autobusu.
-----------	-------------------------

Zwroca

Obiekt DTO autobusu lub wiadomość o błędzie.

Definicja w linii [75](#) pliku [AutobusController.java](#).

6.4.2.6 template()

```
ResponseEntity<?> com.mklachl.sopkom.controller.AutobusController.template ( )
```

Endpoint zwracający przykładowy obiekt AutobusDto.

Zwroca

Przykładowy obiekt AutobusDto.

Definicja w linii [48](#) pliku [AutobusController.java](#).

6.4.2.7 test()

```
ResponseEntity< String > com.mklachl.sopkom.controller.AutobusController.test ( )
```

Endpoint testowy.

Zwroca

Wiadomość testowa.

Definicja w linii [39](#) pliku [AutobusController.java](#).

6.4.2.8 updateAutobus()

```
ResponseEntity<?> com.mklachl.sopkom.controller.AutobusController.updateAutobus ( @PathVariable("id") Long id, @RequestBody AutobusDto autobusDto )
```

Aktualizuje autobus po identyfikatorze.

Parametry

<i>id</i>	Identyfikator autobusu.
<i>autobusDto</i>	Obiekt DTO autobusu.

Zwraca

Wiadomość o statusie operacji.

Definicja w linii 104 pliku [AutobusController.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/com/mklachl/sopkom/controller/AutobusController.java

6.5 Dokumentacja klasy com.mklachl.sopkom.model.dto.AutobusDto

Metody publiczne

- `AutobusDto ()`
- `AutobusDto (Autobus autobus)`
- `Long getAutbousId ()`
- `String getNumerRejestracyjny ()`
- `Date getPrzegladWaznyDo ()`
- `String getStatus ()`
- `float getPrzebieg ()`
- `void setAutbousId (Long autbousId)`
- `void setNumerRejestracyjny (String numerRejestracyjny)`
- `void setPrzegladWaznyDo (Date przegladWaznyDo)`
- `void setStatus (String status)`
- `void setPrzebieg (float przebieg)`
- `String toString ()`

6.5.1 Opis szczegółowy

Data Transfer Object (DTO) dla encji Autobus.

Definicja w linii 9 pliku [AutobusDto.java](#).

6.5.2 Dokumentacja konstruktora i destruktora

6.5.2.1 AutobusDto() [1/2]

```
com.mklachl.sopkom.model.dto.AutobusDto.AutobusDto ( )
```

Konstruktor domyślny.

Definicja w linii 39 pliku [AutobusDto.java](#).

6.5.2.2 AutobusDto() [2/2]

```
com.mklachl.sopkom.model.dto.AutobusDto.AutobusDto (  
    Autobus autobus )
```

Konstruktor przyjmujący obiekt encji Autobus.

Parametry

<code>autobus</code>	Obiekt encji Autobus.
----------------------	-----------------------

Definicja w linii [46](#) pliku [AutobusDto.java](#).

6.5.3 Dokumentacja funkcji składowych

6.5.3.1 `getAutbousId()`

`Long com.mklachl.sopkom.model.dto.AutobusDto.getAutbousId ()`

Zwraca identyfikator autobusu.

Zwraca

`autbousId` Identyfikator autobusu.

Definicja w linii [60](#) pliku [AutobusDto.java](#).

6.5.3.2 `getNumerRejestracyjny()`

`String com.mklachl.sopkom.model.dto.AutobusDto.getNumerRejestracyjny ()`

Zwraca numer rejestracyjny autobusu.

Zwraca

`numerRejestracyjny` Numer rejestracyjny autobusu.

Definicja w linii [68](#) pliku [AutobusDto.java](#).

6.5.3.3 `getPrzebieg()`

`float com.mklachl.sopkom.model.dto.AutobusDto.getPrzebieg ()`

Zwraca przebieg autobusu.

Zwraca

`przebieg` Przebieg autobusu.

Definicja w linii [92](#) pliku [AutobusDto.java](#).

6.5.3.4 getPrzegladWaznyDo()

```
Date com.mklachl.sopkom.model.dto.AutobusDto.getPrzegladWaznyDo ( )
```

Zwraca datę ważności przeglądu technicznego autobusu.

Zwrota

przegladWaznyDo Data ważności przeglądu technicznego autobusu.

Definicja w linii 76 pliku [AutobusDto.java](#).

6.5.3.5 getStatus()

```
String com.mklachl.sopkom.model.dto.AutobusDto.getStatus ( )
```

Zwraca status autobusu.

Zwrota

status Status autobusu.

Definicja w linii 84 pliku [AutobusDto.java](#).

6.5.3.6 setAutbousId()

```
void com.mklachl.sopkom.model.dto.AutobusDto.setAutbousId ( Long autbousId )
```

Ustawia identyfikator autobusu.

Parametry

<i>autbous</i> <i>Id</i>	Identyfikator autobusu.
-----------------------------	-------------------------

Definicja w linii 102 pliku [AutobusDto.java](#).

6.5.3.7 setNumerRejestracyjny()

```
void com.mklachl.sopkom.model.dto.AutobusDto.setNumerRejestracyjny ( String numerRejestracyjny )
```

Ustawia numer rejestracyjny autobusu.

Parametry

<i>numerRejestracyjny</i>	Numer rejestracyjny autobusu.
---------------------------	-------------------------------

Definicja w linii 110 pliku [AutobusDto.java](#).

6.5.3.8 setPrzebieg()

```
void com.mklachl.sopkom.model.dto.AutobusDto.setPrzebieg (
    float przebieg )
```

Ustawia przebieg autobusu.

Parametry

<i>przebieg</i>	Przebieg autobusu.
-----------------	--------------------

Definicja w linii 134 pliku [AutobusDto.java](#).

6.5.3.9 setPrzegladWaznyDo()

```
void com.mklachl.sopkom.model.dto.AutobusDto.setPrzegladWaznyDo (
    Date przegladWaznyDo )
```

Ustawia datę ważności przeglądu technicznego autobusu.

Parametry

<i>przegladWaznyDo</i>	Data ważności przeglądu technicznego autobusu.
------------------------	--

Definicja w linii 118 pliku [AutobusDto.java](#).

6.5.3.10 setStatus()

```
void com.mklachl.sopkom.model.dto.AutobusDto.setStatus (
    String status )
```

Ustawia status autobusu.

Parametry

<i>status</i>	Status autobusu.
---------------	------------------

Definicja w linii 126 pliku AutobusDto.java.

6.5.3.11 `toString()`

```
String com.mklachl.sopkom.model.dto.AutobusDto.toString ( )
```

Zwraca reprezentację obiektu w postaciłańcucha znaków.

Zwraca

Reprezentacja obiektu w postaciłańcucha znaków.

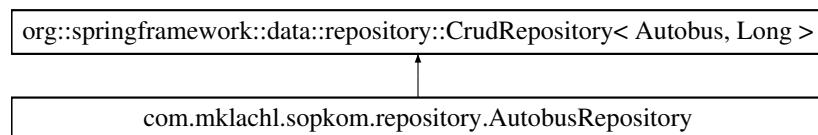
Definicja w linii 143 pliku AutobusDto.java.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.6 Dokumentacja interfejsu com.mklachl.sopkom.repository.AutobusRepository

Diagram dziedziczenia dla com.mklachl.sopkom.repository.AutobusRepository



Metody publiczne

- Autobus `findByNumerRejestracyjny` (String numerRejestracyjny)
- Autobus `findByPrzegladWaznyDoBefore` (Date date)
- Autobus `findByPrzegladWaznyDoAfter` (Date date)
- List< Autobus > `findAllByStatus` (String status)
- List< Autobus > `findAllByPrzebiegGreaterThan` (float przebieg)
- List< Autobus > `findAllByPrzebiegLessThan` (float przebieg)
- List< Autobus > `findAllByPrzebieg` (float przebieg)
- List< Autobus > `findByStatusLike` (String status)
- List< Autobus > `findAllByNumerRejestracyjnyLike` (String numerRejestracyjny)
- List< Autobus > `findAll` ()
- List< Autobus > `findAllByAvailable` (@Param("startTime") Date startTime, @Param("endTime") Date end←Time)

6.6.1 Opis szczegółowy

Interfejs repository dla encji Autobus

Definicja w linii 14 pliku [AutobusRepository.java](#).

6.6.2 Dokumentacja funkcji składowych

6.6.2.1 findAll()

```
List< Autobus > com.mklachl.sopkom.repository.AutobusRepository.findAll ( )
```

Znajduje wszystkie autobusy

Zwroca

Lista autobusów

6.6.2.2 findAllByAvailable()

```
List< Autobus > com.mklachl.sopkom.repository.AutobusRepository.findAllByAvailable (   
    @Param("startTime") Date startTime,  
    @Param("endTime") Date endTime )
```

Znajduje wszystkie dostępne autobusy w podanym przedziale czasowym

Parametry

<i>startTime</i>	czas rozpoczęcia
<i>endTime</i>	czas zakończenia

Zwroca

Lista autobusów

6.6.2.3 findAllByNumerRejestracyjnyLike()

```
List< Autobus > com.mklachl.sopkom.repository.AutobusRepository.findAllByNumerRejestracyjnyLike (   
    String numerRejestracyjny )
```

Znajduje wszystkie autobusy o numerze rejestracyjnym pasującym do wzorca

Parametry

<i>numerRejestracyjny</i>	numer rejestracyjny
---------------------------	---------------------

Zwrota

Lista autobusów

6.6.2.4 findAllByPrzebieg()

```
List< Autobus > com.mklachl.sopkom.repository.AutobusRepository.findAllByPrzebieg ( float przebieg )
```

Znajduje wszystkie autobusy z dokładnym przebiegiem

Parametry

<i>przebieg</i>	przebieg
-----------------	----------

Zwrota

Lista autobusów

6.6.2.5 findAllByPrzebiegGreaterThanOrEqual()

```
List< Autobus > com.mklachl.sopkom.repository.AutobusRepository.findAllByPrzebiegGreaterThanOrEqual ( float przebieg )
```

Znajduje wszystkie autobusy z przebiegiem większym niż podany

Parametry

<i>przebieg</i>	przebieg
-----------------	----------

Zwrota

Lista autobusów

6.6.2.6 findAllByPrzebiegLessThan()

```
List< Autobus > com.mklachl.sopkom.repository.AutobusRepository.findAllByPrzebiegLessThan ( float przebieg )
```

Znajduje wszystkie autobusy z przebiegiem mniejszym niż podany

Parametry

<i>przebieg</i>	<i>przebieg</i>
-----------------	-----------------

Zwroca

Lista autobusów

6.6.2.7 findAllByStatus()

```
List< Autobus > com.mklachl.sopkom.repository.AutobusRepository.findAllByStatus ( String status )
```

Znajduje wszystkie autobusy o określonym statusie

Parametry

<i>status</i>	<i>status</i>
---------------	---------------

Zwroca

Lista autobusów

6.6.2.8 findByNumerRejestracyjny()

```
Autobus com.mklachl.sopkom.repository.AutobusRepository.findByNumerRejestracyjny ( String numerRejestracyjny )
```

Znajduje autobus po numerze rejestracyjnym

Parametry

<i>numerRejestracyjny</i>	<i>numer rejestracyjny</i>
---------------------------	----------------------------

Zwroca

Autobus

6.6.2.9 findByPrzegladWaznyDoAfter()

```
Autobus com.mklachl.sopkom.repository.AutobusRepository.findByPrzegladWaznyDoAfter ( Date date )
```

Znajduje autobus z przeglądem ważnym po określonej dacie

Parametry

<i>date</i>	data
-------------	------

Zwroca

Autobus

6.6.2.10 findByPrzegladWaznyDoBefore()

```
Autobus com.mklachl.sopkom.repository.AutobusRepository.findByPrzegladWaznyDoBefore (
    Date date )
```

Znajduje autobus z przeglądem ważnym przed określoną datą

Parametry

<i>date</i>	data
-------------	------

Zwroca

Autobus

6.6.2.11 findByStatusLike()

```
List< Autobus > com.mklachl.sopkom.repository.AutobusRepository.findByStatusLike (
    String status )
```

Znajduje wszystkie autobusy o statusie pasującym do wzorca

Parametry

<i>status</i>	status
---------------	--------

Zwroca

Lista autobusów

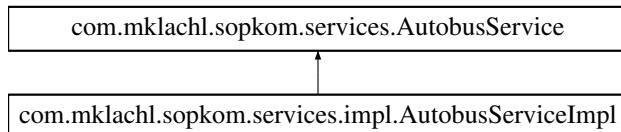
Dokumentacja dla tego interfejsu została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/rep

6.7 Dokumentacja interfejsu

com.mklachl.sopkom.services.AutobusService

Diagram dziedziczenia dla com.mklachl.sopkom.services.AutobusService



Metody publiczne

- Autobus saveAutobus (Autobus autobus)
- Autobus saveAutobus (AutobusDto autobusDto)
- Autobus findAutobusByNumerRejestracyjny (String numerRejestracyjny)
- Optional< Autobus > findAutobusById (long id)
- Autobus findAutobusByPrzegladWaznyDoBefore (Date date)
- Autobus findAutobusByPrzegladWaznyDoAfter (Date date)
- List< Autobus > findAutobusByStatus (String status)
- List< Autobus > findAutobusByPrzebiegGreaterThanOrEqual (float przebieg)
- List< Autobus > findAutobusByPrzebiegLessThan (float przebieg)
- List< Autobus > findAutobusByPrzebieg (float przebieg)
- List< Autobus > findAutobusByAvailable (Date startTime, Date endTime)

6.7.1 Opis szczegółowy

Interfejs serwisu dla obsługi autobusów

Definicja w linii 14 pliku [AutobusService.java](#).

6.7.2 Dokumentacja funkcji składowych

6.7.2.1 findAutobusByAvailable()

```
List< Autobus > com.mklachl.sopkom.services.AutobusService.findAutobusByAvailable (
    Date startTime,
    Date endTime )
```

Znajduje dostępne autobusy w podanym przedziale czasowym

Parametry

<code>startTime</code>	czas rozpoczęcia
<code>endTime</code>	czas zakończenia

Zwrota

lista dostępnych autobusów

Implementowany w [com.mklachl.sopkom.services.impl.AutobusServiceImpl](#).

6.7.2.2 findAutobusById()

```
Optional< Autobus > com.mklachl.sopkom.services.AutobusService.findAutobusById ( long id )
```

Znajduje autobus po ID

Parametry

<i>id</i>	ID autobusu
-----------	-------------

Zwrota

Optional z autobusem

Implementowany w [com.mklachl.sopkom.services.impl.AutobusServiceImpl](#).

6.7.2.3 findAutobusByNumerRejestacyjny()

```
Autobus com.mklachl.sopkom.services.AutobusService.findAutobusByNumerRejestacyjny ( String numerRejestacyjny )
```

Znajduje autobus po numerze rejestracyjnym

Parametry

<i>numerRejestacyjny</i>	numer rejestracyjny
--------------------------	---------------------

Zwrota

znaleziony autobus

Implementowany w [com.mklachl.sopkom.services.impl.AutobusServiceImpl](#).

6.7.2.4 findAutobusByPrzebieg()

```
List< Autobus > com.mklachl.sopkom.services.AutobusService.findAutobusByPrzebieg ( float przebieg )
```

Znajduje autobusy po dokładnym przebiegu

Parametry

<i>przebieg</i>	przebieg
-----------------	----------

Zwroca

lista autobusów

Implementowany w [com.mklachl.sopkom.services.impl.AutobusServiceImpl](#).

6.7.2.5 **findAutobusByPrzebiegGreaterThanOr()**

```
List< Autobus > com.mklachl.sopkom.services.AutobusService.findAutobusByPrzebiegGreaterThanOr ( float przebieg )
```

Znajduje autobusy, których przebieg jest większy niż podany

Parametry

<i>przebieg</i>	przebieg
-----------------	----------

Zwroca

lista autobusów

Implementowany w [com.mklachl.sopkom.services.impl.AutobusServiceImpl](#).

6.7.2.6 **findAutobusByPrzebiegLessThan()**

```
List< Autobus > com.mklachl.sopkom.services.AutobusService.findAutobusByPrzebiegLessThan ( float przebieg )
```

Znajduje autobusy, których przebieg jest mniejszy niż podany

Parametry

<i>przebieg</i>	przebieg
-----------------	----------

Zwroca

lista autobusów

Implementowany w [com.mklachl.sopkom.services.impl.AutobusServiceImpl](#).

6.7.2.7 findAutobusByPrzegladWaznyDoAfter()

```
Autobus com.mklachl.sopkom.services.AutobusService.findAutobusByPrzegladWaznyDoAfter (
    Date date )
```

Znajduje autobus, którego przegląd jest ważny po podanej dacie

Parametry

<i>date</i>	data
-------------	------

Zwrota

znaleziony autobus

Implementowany w [com.mklachl.sopkom.services.impl.AutobusServiceImpl](#).

6.7.2.8 findAutobusByPrzegladWaznyDoBefore()

```
Autobus com.mklachl.sopkom.services.AutobusService.findAutobusByPrzegladWaznyDoBefore (
    Date date )
```

Znajduje autobus, którego przegląd jest ważny do podanej daty

Parametry

<i>date</i>	data
-------------	------

Zwrota

znaleziony autobus

Implementowany w [com.mklachl.sopkom.services.impl.AutobusServiceImpl](#).

6.7.2.9 findAutobusByStatus()

```
List< Autobus > com.mklachl.sopkom.services.AutobusService.findAutobusByStatus (
    String status )
```

Znajduje autobusy po statusie

Parametry

<i>status</i>	status
---------------	--------

Zwroca

lista autobusów

Implementowany w [com.mklachl.sopkom.services.impl.AutobusServiceImpl](#).

6.7.2.10 saveAutobus() [1/2]

```
Autobus com.mklachl.sopkom.services.AutobusService.saveAutobus (
    Autobus autobus )
```

Zapisuje nowy autobus

Parametry

autobus	encja autobus
---------	---------------

Zwroca

zapisany autobus

Implementowany w [com.mklachl.sopkom.services.impl.AutobusServiceImpl](#).

6.7.2.11 saveAutobus() [2/2]

```
Autobus com.mklachl.sopkom.services.AutobusService.saveAutobus (
    AutobusDto autobusDto )
```

Zapisuje nowy autobus na podstawie DTO

Parametry

autobusDto	DTO autobusu
------------	--------------

Zwroca

zapisany autobus

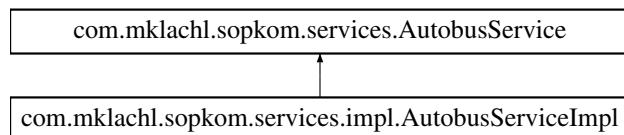
Implementowany w [com.mklachl.sopkom.services.impl.AutobusServiceImpl](#).

Dokumentacja dla tego interfejsu została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/ser...

6.8 Dokumentacja klasy com.mklachl.sopkom.services.impl.AutobusServiceImpl

Diagram dziedziczenia dla com.mklachl.sopkom.services.impl.AutobusServiceImpl



Metody publiczne

- AutobusServiceImpl ([AutobusRepository](#) autobusRepository)
- Autobus saveAutobus ([AutobusDto](#) autobusDto)
- Autobus saveAutobus ([Autobus](#) autobusDto)
- Autobus findAutobusByNumerRejestracyjny (String numerRejestracyjny)
- Optional<[Autobus](#)> findAutobusById (long id)
- Autobus findAutobusByPrzegladWaznyDoBefore (Date date)
- Autobus findAutobusByPrzegladWaznyDoAfter (Date date)
- List<[Autobus](#)> findAutobusByStatus (String status)
- List<[Autobus](#)> findAutobusByPrzebiegGreaterThanOrEqual (float przebieg)
- List<[Autobus](#)> findAutobusByPrzebiegLessThan (float przebieg)
- List<[Autobus](#)> findAutobusByPrzebieg (float przebieg)
- List<[Autobus](#)> findAutobusByAvailable (Date startTime, Date endTime)

6.8.1 Opis szczegółowy

Definicja w linii 14 pliku [AutobusServiceImpl.java](#).

6.8.2 Dokumentacja konstruktora i destruktora

6.8.2.1 AutobusServiceImpl()

```
com.mklachl.sopkom.services.impl.AutobusServiceImpl.AutobusServiceImpl (   
    AutobusRepository autobusRepository )
```

Definicja w linii 17 pliku [AutobusServiceImpl.java](#).

6.8.3 Dokumentacja funkcji składowych

6.8.3.1 findAutobusByAvailable()

```
List<Autobus> com.mklachl.sopkom.services.impl.AutobusServiceImpl.findAutobusByAvailable (   
    Date startTime,   
    Date endTime )
```

Znajduje dostępne autobusy w podanym przedziale czasowym

Parametry

<i>startTime</i>	czas rozpoczęcia
<i>endTime</i>	czas zakończenia

Zwroca

lista dostępnych autobusów

Implementuje [com.mklachl.sopkom.services.AutobusService](#).

Definicja w linii [85](#) pliku [AutobusServiceImpl.java](#).

6.8.3.2 findAutobusById()

```
Optional< Autobus > com.mklachl.sopkom.services.impl.AutobusServiceImpl.findAutobusById ( long id )
```

Znajduje autobus po ID

Parametry

<i>id</i>	ID autobusu
-----------	-------------

Zwroca

Optional z autobusem

Implementuje [com.mklachl.sopkom.services.AutobusService](#).

Definicja w linii [50](#) pliku [AutobusServiceImpl.java](#).

6.8.3.3 findAutobusByNumerRejestracyjny()

```
Autobus com.mklachl.sopkom.services.impl.AutobusServiceImpl.findAutobusByNumerRejestracyjny ( String numerRejestracyjny )
```

Znajduje autobus po numerze rejestracyjnym

Parametry

<i>numerRejestracyjny</i>	numer rejestracyjny
---------------------------	---------------------

Zwrota

znaleziony autobus

Implementuje [com.mklachl.sopkom.services.AutobusService](#).

Definicja w linii 45 pliku [AutobusServiceImpl.java](#).

6.8.3.4 findAutobusByPrzebieg()

```
List< Autobus > com.mklachl.sopkom.services.impl.AutobusServiceImpl.findAutobusByPrzebieg ( float przebieg )
```

Znajduje autobusy po dokładnym przebiegu

Parametry

<i>przebieg</i>	przebieg
-----------------	----------

Zwrota

lista autobusów

Implementuje [com.mklachl.sopkom.services.AutobusService](#).

Definicja w linii 80 pliku [AutobusServiceImpl.java](#).

6.8.3.5 findAutobusByPrzebiegGreaterThanOrEqual()

```
List< Autobus > com.mklachl.sopkom.services.impl.AutobusServiceImpl.findAutobusByPrzebieg ( GreaterThan ( float przebieg ) )
```

Znajduje autobusy, których przebieg jest większy niż podany

Parametry

<i>przebieg</i>	przebieg
-----------------	----------

Zwrota

lista autobusów

Implementuje [com.mklachl.sopkom.services.AutobusService](#).

Definicja w linii 70 pliku [AutobusServiceImpl.java](#).

6.8.3.6 findAutobusByPrzebiegLessThan()

```
List< Autobus > com.mklachl.sopkom.services.impl.AutobusServiceImpl.findAutobusByPrzebieg->
LessThan (
    float przebieg )
```

Znajduje autobusy, których przebieg jest mniejszy niż podany

Parametry

<code>przebieg</code>	<code>przebieg</code>
-----------------------	-----------------------

Zwroca

lista autobusów

Implementuje [com.mklachl.sopkom.services.AutobusService](#).

Definicja w linii [75](#) pliku [AutobusServiceImpl.java](#).

6.8.3.7 findAutobusByPrzegladWaznyDoAfter()

```
Autobus com.mklachl.sopkom.services.impl.AutobusServiceImpl.findAutobusByPrzegladWaznyDoAfter
(
    Date date )
```

Znajduje autobus, którego przegląd jest ważny po podanej dacie

Parametry

<code>date</code>	<code>data</code>
-------------------	-------------------

Zwroca

znaleziony autobus

Implementuje [com.mklachl.sopkom.services.AutobusService](#).

Definicja w linii [60](#) pliku [AutobusServiceImpl.java](#).

6.8.3.8 findAutobusByPrzegladWaznyDoBefore()

```
Autobus com.mklachl.sopkom.services.impl.AutobusServiceImpl.findAutobusByPrzegladWaznyDoBefore
(
    Date date )
```

Znajduje autobus, którego przegląd jest ważny do podanej daty

Parametry

<i>date</i>	data
-------------	------

Zwroca

znaleziony autobus

Implementuje [com.mklachl.sopkom.services.AutobusService](#).

Definicja w linii [55](#) pliku [AutobusServiceImpl.java](#).

6.8.3.9 findAutobusByStatus()

```
List< Autobus > com.mklachl.sopkom.services.impl.AutobusServiceImpl.findAutobusByStatus ( String status )
```

Znajduje autobusy po statusie

Parametry

<i>status</i>	status
---------------	--------

Zwroca

lista autobusów

Implementuje [com.mklachl.sopkom.services.AutobusService](#).

Definicja w linii [65](#) pliku [AutobusServiceImpl.java](#).

6.8.3.10 saveAutobus() [1/2]

```
Autobus com.mklachl.sopkom.services.impl.AutobusServiceImpl.saveAutobus ( Autobus autobus )
```

Zapisuje nowy autobus

Parametry

<i>autobus</i>	encja autobus
----------------	---------------

Zwrota

zapisany autobus

Implementuje [com.mklachl.sopkom.services.AutobusService](#).

Definicja w linii [33](#) pliku [AutobusServiceImpl.java](#).

6.8.3.11 saveAutobus() [2/2]

```
Autobus com.mklachl.sopkom.services.impl.AutobusServiceImpl.saveAutobus (
    AutobusDto autobusDto )
```

Zapisuje nowy autobus na podstawie DTO

Parametry

<code>autobusDto</code>	DTO autobusu
-------------------------	--------------

Zwrota

zapisany autobus

Implementuje [com.mklachl.sopkom.services.AutobusService](#).

Definicja w linii [21](#) pliku [AutobusServiceImpl.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/ser...

6.9 Dokumentacja klasy com.mklachl.sopkom.model.entity.Bilet

Metody publiczne

- Short `getBiletid` ()
- void `setBiletid` (Short biletId)
- float `getCena` ()
- void `setCena` (float cena)
- String `getNazwa` ()
- void `setNazwa` (String nazwa)
- List< [PrzejazdBilet](#) > `getPrzejazdBilet` ()
- void `setPrzejazdBilet` (List< [PrzejazdBilet](#) > PrzejazdBilet)

6.9.1 Opis szczegółowy

Klasa reprezentująca encję [Bilet](#).

Definicja w linii [12](#) pliku [Bilet.java](#).

6.9.2 Dokumentacja funkcji składowych

6.9.2.1 getBileId()

```
Short com.mklachl.sopkom.model.entity.Bilet.getBileId ( )
```

Zwraca identyfikator biletu.

Zwraca

bileId Identyfikator biletu.

Definicja w linii [46](#) pliku [Bilet.java](#).

6.9.2.2 getCena()

```
float com.mklachl.sopkom.model.entity.Bilet.getCena ( )
```

Zwraca cenę biletu.

Zwraca

cena Cena biletu.

Definicja w linii [62](#) pliku [Bilet.java](#).

6.9.2.3 getNazwa()

```
String com.mklachl.sopkom.model.entity.Bilet.getNazwa ( )
```

Zwraca nazwę biletu.

Zwraca

nazwa Nazwa biletu.

Definicja w linii [78](#) pliku [Bilet.java](#).

6.9.2.4 getPrzejazdBilet()

```
List< PrzejazdBilet > com.mklachl.sopkom.model.entity.Bilet.getPrzejazdBilet ( )
```

Zwraca listę przejazdów związanych z biletem.

Zwroca

[PrzejazdBilet](#) Lista przejazdów.

Definicja w linii [94](#) pliku [Bilet.java](#).

6.9.2.5 setBiletId()

```
void com.mklachl.sopkom.model.entity.Bilet.setBiletId (  
    Short biletId )
```

Ustawia identyfikator biletu.

Parametry

<i>bilet</i> ↪ <i>Id</i>	Identyfikator biletu.
-----------------------------	-----------------------

Definicja w linii [54](#) pliku [Bilet.java](#).

6.9.2.6 setCena()

```
void com.mklachl.sopkom.model.entity.Bilet.setCena ( float cena )
```

Ustawia cenę biletu.

Parametry

<i>cena</i>	Cena biletu.
-------------	--------------

Definicja w linii [70](#) pliku [Bilet.java](#).

6.9.2.7 setNazwa()

```
void com.mklachl.sopkom.model.entity.Bilet.setNazwa ( String nazwa )
```

Ustawia nazwę biletu.

Parametry

<i>nazwa</i>	Nazwa biletu.
--------------	---------------

Definicja w linii [86](#) pliku [Bilet.java](#).

6.9.2.8 setPrzejazdBilet()

```
void com.mklachl.sopkom.model.entity.Bilet.setPrzejazdBilet ( List< PrzejazdBilet > PrzejazdBilet )
```

Ustawia listę przejazdów związanych z biletem.

Parametry

PrzejazdBilet	Lista przejazdów.
-------------------------------	-------------------

Definicja w linii [102](#) pliku [Bilet.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.10 Dokumentacja klasy com.mklachl.sopkom.model.dto.BiletDto

Metody publiczne

- [BiletDto \(\)](#)
- [BiletDto \(Bilet bilet\)](#)
- Short [getBiletId \(\)](#)
- void [setBiletId \(Short biletId\)](#)
- float [getCena \(\)](#)
- void [setCena \(float cena\)](#)
- String [getNazwa \(\)](#)
- void [setNazwa \(String nazwa\)](#)

6.10.1 Opis szczegółowy

Data Transfer Object (DTO) dla encji Bilet.

Definicja w linii [8](#) pliku [BiletDto.java](#).

6.10.2 Dokumentacja konstruktora i destruktora

6.10.2.1 BiletDto() [1/2]

```
com.mklachl.sopkom.model.dto.BiletDto.BiletDto ( )
```

Konstruktor domyślny.

Definicja w linii [28](#) pliku [BiletDto.java](#).

6.10.2.2 BiletDto() [2/2]

```
com.mklachl.sopkom.model.dto.BiletDto.BiletDto (   
    Bilet bilet )
```

Konstruktor przyjmujący obiekt encji Bilet.

Parametry

<i>bilet</i>	Obiekt encji Bilet.
--------------	---------------------

Definicja w linii [35](#) pliku [BiletDto.java](#).

6.10.3 Dokumentacja funkcji składowych

6.10.3.1 getBileId()

```
Short com.mklachl.sopkom.model.dto.BiletDto.getBileId ( )
```

Zwraca identyfikator biletu.

Zwraca

bileId Identyfikator biletu.

Definicja w linii [47](#) pliku [BiletDto.java](#).

6.10.3.2 getCena()

```
float com.mklachl.sopkom.model.dto.BiletDto.getCena ( )
```

Zwraca cenę biletu.

Zwraca

cena Cena biletu.

Definicja w linii [63](#) pliku [BiletDto.java](#).

6.10.3.3 getNazwa()

```
String com.mklachl.sopkom.model.dto.BiletDto.getNazwa ( )
```

Zwraca nazwę biletu.

Zwraca

nazwa Nazwa biletu.

Definicja w linii [79](#) pliku [BiletDto.java](#).

6.10.3.4 setBileId()

```
void com.mklachl.sopkom.model.dto.BiletDto.setBileId ( Short biletId )
```

Ustawia identyfikator biletu.

Parametry

<i>bilet</i> ↪ <i>Id</i>	Identyfikator biletu.
-----------------------------	-----------------------

Definicja w linii [55](#) pliku [BiletDto.java](#).

6.10.3.5 setCena()

```
void com.mklachl.sopkom.model.dto.BiletDto.setCena (   
    float cena )
```

Ustawia cenę biletu.

Parametry

<i>cena</i>	Cena biletu.
-------------	--------------

Definicja w linii [71](#) pliku [BiletDto.java](#).

6.10.3.6 setNazwa()

```
void com.mklachl.sopkom.model.dto.BiletDto.setNazwa (   
    String nazwa )
```

Ustawia nazwę biletu.

Parametry

<i>nazwa</i>	Nazwa biletu.
--------------	---------------

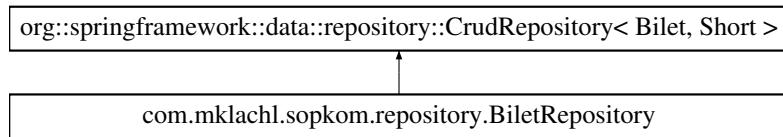
Definicja w linii [87](#) pliku [BiletDto.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.11 Dokumentacja interfejsu com.mklachl.sopkom.repository.BiletRepository

Diagram dziedziczenia dla com.mklachl.sopkom.repository.BiletRepository



Metody publiczne

- List< [Bilet](#) > [findAllByCena](#) (float cena)
- List< [Bilet](#) > [findAllByNazwa](#) (String nazwa)
- List< [Bilet](#) > [findAll](#) ()

6.11.1 Opis szczegółowy

Interfejs repository dla encji Bilet

Definicja w linii 11 pliku [BiletRepository.java](#).

6.11.2 Dokumentacja funkcji składowych

6.11.2.1 findAll()

```
List< Bilet > com.mklachl.sopkom.repository.BiletRepository.findAll ( )
```

Znajduje wszystkie bilety

Zwroca

Lista biletów

6.11.2.2 findAllByCena()

```
List< Bilet > com.mklachl.sopkom.repository.BiletRepository.findAllByCena ( float cena )
```

Znajduje wszystkie bilety o określonej cenie

Parametry

<i>cena</i>	cena
-------------	------

Zwroca

Lista biletów

6.11.2.3 findAllByNazwa()

```
List< Bilet > com.mklachl.sopkom.repository.BiletRepository.findAllByNazwa ( String nazwa )
```

Znajduje wszystkie bilety o określonej nazwie

Parametry

<i>nazwa</i>	<i>nazwa</i>
--------------	--------------

Zwroca

Lista biletów

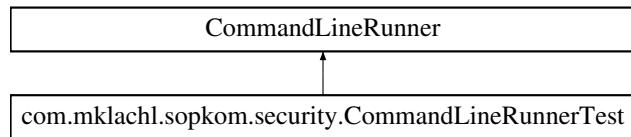
Dokumentacja dla tego interfejsu została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/rep

6.12 Dokumentacja klasy

com.mklachl.sopkom.security.CommandLineRunnerTest

Diagram dziedziczenia dla com.mklachl.sopkom.security.CommandLineRunnerTest



Metody publiczne

- void [run](#) (String... args) throws Exception

Atrybuty publiczne

- [UserRepository userRepository](#)

Statyczne atrybuty publiczne

- static final String ADMIN_LOGIN = "admin"
- static final String ADMIN_NAME = "admin"
- static final String ADMIN_PASS = "admin"
- static final String MANAGER_LOGIN = "manager"
- static final String MANAGER_NAME = "manager"
- static final String MANAGER_PASS = "manager"

6.12.1 Opis szczegółowy

Definicja w linii 22 pliku [CommandLineRunnerTest.java](#).

6.12.2 Dokumentacja funkcji składowych

6.12.2.1 run()

```
void com.mklachl.sopkom.security.CommandLineRunnerTest.run (
    String... args) throws Exception
```

Definicja w linii 49 pliku [CommandLineRunnerTest.java](#).

6.12.3 Dokumentacja atrybutów składowych

6.12.3.1 ADMIN_LOGIN

```
final String com.mklachl.sopkom.security.CommandLineRunnerTest.ADMIN_LOGIN = "admin" [static]
```

Definicja w linii 33 pliku [CommandLineRunnerTest.java](#).

6.12.3.2 ADMIN_NAME

```
final String com.mklachl.sopkom.security.CommandLineRunnerTest.ADMIN_NAME = "admin" [static]
```

Definicja w linii 34 pliku [CommandLineRunnerTest.java](#).

6.12.3.3 ADMIN_PASS

```
final String com.mklachl.sopkom.security.CommandLineRunnerTest.ADMIN_PASS = "admin" [static]
```

Definicja w linii 35 pliku [CommandLineRunnerTest.java](#).

6.12.3.4 MANAGER_LOGIN

```
final String com.mklachl.sopkom.security.CommandLineRunnerTest.MANAGER_LOGIN = "manager" [static]
```

Definicja w linii 37 pliku [CommandLineRunnerTest.java](#).

6.12.3.5 MANAGER_NAME

```
final String com.mklachl.sopkom.security.CommandLineRunnerTest.MANAGER_NAME = "manager" [static]
```

Definicja w linii 38 pliku [CommandLineRunnerTest.java](#).

6.12.3.6 MANAGER_PASS

```
final String com.mklachl.sopkom.security.CommandLineRunnerTest.MANAGER_PASS = "manager" [static]
```

Definicja w linii 39 pliku [CommandLineRunnerTest.java](#).

6.12.3.7 userRepository

```
UserRepository com.mklachl.sopkom.security.CommandLineRunnerTest.userRepository
```

Definicja w linii 25 pliku [CommandLineRunnerTest.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/se

6.13 Dokumentacja klasy com.mklachl.sopkom.controller.Controller

Metody publiczne

- ResponseEntity< String > [getString \(\)](#)
- ResponseEntity< String > [getString2 \(\)](#)

6.13.1 Opis szczegółowy

Kontroler ogólny obsługujący różne żądania GET.

Definicja w linii 12 pliku [Controller.java](#).

6.13.2 Dokumentacja funkcji składowych

6.13.2.1 getString()

```
ResponseEntity< String > com.mklachl.sopkom.controller.Controller.getString( )
```

Endpoint zwracający wiadomość dla nieautoryzowanych użytkowników.

Zwroca

Wiadomość tekstowa.

Definicja w linii 19 pliku [Controller.java](#).

6.13.2.2 getString2()

```
ResponseEntity< String > com.mklachl.sopkom.controller.Controller.getString2( )
```

Endpoint zwracający wiadomość tylko dla autoryzowanych użytkowników.

Zwroca

Wiadomość tekstowa.

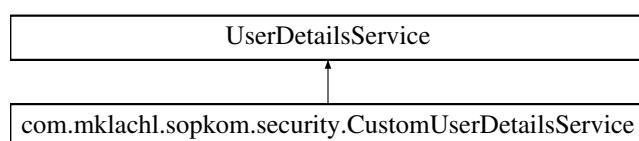
Definicja w linii 28 pliku [Controller.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/com

6.14 Dokumentacja klasy com.mklachl.sopkom.security.CustomUserDetailsService

Diagram dziedziczenia dla com.mklachl.sopkom.security.CustomUserDetailsService



Metody publiczne

- [CustomUserDetailsService](#) ([UserRepository](#) userRepository)
- [UserDetails](#) [loadUserByUsername](#) (String login) throws [UsernameNotFoundException](#)

6.14.1 Opis szczegółowy

Definicja w linii 13 pliku [CustomUserDetailsService.java](#).

6.14.2 Dokumentacja konstruktora i destruktora

6.14.2.1 CustomUserDetailsService()

```
com.mklachl.sopkom.security.CustomUserDetailsService.CustomUserDetailsService (
    UserRepository userRepository )
```

Definicja w linii 17 pliku [CustomUserDetailsService.java](#).

6.14.3 Dokumentacja funkcji składowych

6.14.3.1 loadUserByUsername()

```
UserDetails com.mklachl.sopkom.security.CustomUserDetailsService.loadUserByUsername (
    String login ) throws UsernameNotFoundException
```

Definicja w linii 22 pliku [CustomUserDetailsService.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/se

6.15 Dokumentacja klasy

com.mklachl.sopkom.model.dto.DateWrapperDto

Metody publiczne

- [DateWrapperDto](#) ([Date](#) date)
- [DateWrapperDto](#) ()
- [Date](#) [getDate](#) ()
- void [setDate](#) ([Date](#) date)

6.15.1 Opis szczegółowy

Data Transfer Object (DTO) dla obiektu opakowującego datę.

Definicja w linii 9 pliku [DateWrapperDto.java](#).

6.15.2 Dokumentacja konstruktora i destruktora

6.15.2.1 DateWrapperDto() [1/2]

```
com.mklachl.sopkom.model.dto.DateWrapperDto.DateWrapperDto (
    Date date )
```

Konstruktor przyjmujący datę.

Parametry

<i>date</i>	Data.
-------------	-------

Definicja w linii 21 pliku [DateWrapperDto.java](#).

6.15.2.2 DateWrapperDto() [2/2]

```
com.mklachl.sopkom.model.dto.DateWrapperDto.DateWrapperDto ( )
```

Konstruktor domyślny.

Definicja w linii 28 pliku [DateWrapperDto.java](#).

6.15.3 Dokumentacja funkcji składowych

6.15.3.1 getDate()

```
Date com.mklachl.sopkom.model.dto.DateWrapperDto.getDate ( )
```

Zwraca datę.

Zwraca

date Data.

Definicja w linii 35 pliku [DateWrapperDto.java](#).

6.15.3.2 setDate()

```
void com.mklachl.sopkom.model.dto.DateWrapperDto.setDate (
    Date date )
```

Ustawia datę.

Parametry

<i>date</i>	Data.
-------------	-------

Definicja w linii 43 pliku [DateWrapperDto.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.16 Dokumentacja klasy **com.mklachl.sopkom.model.entity.Harmonogram**

Metody publiczne

- Short [getHamonogramId](#) ()
- void [setHamonogramId](#) (Short hamonogramId)
- String [getNazwa](#) ()
- void [setNazwa](#) (String nazwa)
- boolean [isPon](#) ()
- void [setPon](#) (boolean pon)
- boolean [isWto](#) ()
- void [setWto](#) (boolean wto)
- boolean [isSro](#) ()
- void [setSro](#) (boolean sro)
- boolean [isCzw](#) ()
- void [setCzw](#) (boolean czw)
- boolean [isPia](#) ()
- void [setPia](#) (boolean pia)
- boolean [isSob](#) ()
- void [setSob](#) (boolean sob)
- boolean [isNie](#) ()
- void [setNie](#) (boolean nie)
- String [getDodatkowelInf](#) ()
- void [setDodatkowelInf](#) (String dodatkowelInf)

6.16.1 Opis szczegółowy

Klasa reprezentująca encję [Harmonogram](#).

Definicja w linii 15 pliku [Harmonogram.java](#).

6.16.2 Dokumentacja funkcji składowych

6.16.2.1 getDodatkoweInf()

```
String com.mklachl.sopkom.model.entity.Harmonogram.getDodatkoweInf ( )
```

Zwraca dodatkowe informacje dotyczące harmonogramu.

Zwrota

dodatkoweInf Dodatkowe informacje.

Definicja w linii [227](#) pliku [Harmonogram.java](#).

6.16.2.2 getHamonogramId()

```
Short com.mklachl.sopkom.model.entity.Harmonogram.getHamonogramId ( )
```

Zwraca identyfikator harmonogramu.

Zwrota

hamonogramId Identyfikator harmonogramu.

Definicja w linii [83](#) pliku [Harmonogram.java](#).

6.16.2.3 getNazwa()

```
String com.mklachl.sopkom.model.entity.Harmonogram.getNazwa ( )
```

Zwraca nazwę harmonogramu.

Zwrota

nazwa Nazwa harmonogramu.

Definicja w linii [99](#) pliku [Harmonogram.java](#).

6.16.2.4 isCzw()

```
boolean com.mklachl.sopkom.model.entity.Harmonogram.isCzw ( )
```

Zwraca flagę wskazującą, czy harmonogram obowiązuje w czwartek.

Zwraca

czw Flaga dla czwartku.

Definicja w linii [163](#) pliku [Harmonogram.java](#).

6.16.2.5 isNie()

```
boolean com.mklachl.sopkom.model.entity.Harmonogram.isNie ( )
```

Zwraca flagę wskazującą, czy harmonogram obowiązuje w niedzielę.

Zwraca

nie Flaga dla niedzieli.

Definicja w linii [211](#) pliku [Harmonogram.java](#).

6.16.2.6 isPia()

```
boolean com.mklachl.sopkom.model.entity.Harmonogram.isPia ( )
```

Zwraca flagę wskazującą, czy harmonogram obowiązuje w piątek.

Zwraca

pia Flaga dla piątku.

Definicja w linii [179](#) pliku [Harmonogram.java](#).

6.16.2.7 isPon()

```
boolean com.mklachl.sopkom.model.entity.Harmonogram.isPon ( )
```

Zwraca flagę wskazującą, czy harmonogram obowiązuje w poniedziałek.

Zwraca

pon Flaga dla poniedziałku.

Definicja w linii [115](#) pliku [Harmonogram.java](#).

6.16.2.8 isSob()

```
boolean com.mklachl.sopkom.model.entity.Harmonogram.isSob ( )
```

Zwraca flagę wskazującą, czy harmonogram obowiązuje w sobotę.

Zwrota

sob Flaga dla soboty.

Definicja w linii [195](#) pliku [Harmonogram.java](#).

6.16.2.9 isSro()

```
boolean com.mklachl.sopkom.model.entity.Harmonogram.isSro ( )
```

Zwraca flagę wskazującą, czy harmonogram obowiązuje w środę.

Zwrota

sro Flaga dla środy.

Definicja w linii [147](#) pliku [Harmonogram.java](#).

6.16.2.10 isWto()

```
boolean com.mklachl.sopkom.model.entity.Harmonogram.isWto ( )
```

Zwraca flagę wskazującą, czy harmonogram obowiązuje we wtorek.

Zwrota

wto Flaga dla wtorku.

Definicja w linii [131](#) pliku [Harmonogram.java](#).

6.16.2.11 setCzw()

```
void com.mklachl.sopkom.model.entity.Harmonogram.setCzw ( boolean czw )
```

Ustawia flagę wskazującą, czy harmonogram obowiązuje w czwartek.

Parametry

<i>czw</i>	Flaga dla czwartku.
------------	---------------------

Definicja w linii [171](#) pliku [Harmonogram.java](#).

6.16.2.12 setDodatkoweInf()

```
void com.mklachl.sopkom.model.entity.Harmonogram.setDodatkoweInf (
    String dodatkoweInf )
```

Ustawia dodatkowe informacje dotyczące harmonogramu.

Parametry

<i>dodatkoweInf</i>	Dodatkowe informacje.
---------------------	-----------------------

Definicja w linii [235](#) pliku [Harmonogram.java](#).

6.16.2.13 setHamonogramId()

```
void com.mklachl.sopkom.model.entity.Harmonogram.setHamonogramId (
    Short hamonogramId )
```

Ustawia identyfikator harmonogramu.

Parametry

<i>hamonogram</i> ↪ <i>Id</i>	Identyfikator harmonogramu.
----------------------------------	-----------------------------

Definicja w linii [91](#) pliku [Harmonogram.java](#).

6.16.2.14 setNazwa()

```
void com.mklachl.sopkom.model.entity.Harmonogram.setNazwa (
    String nazwa )
```

Ustawia nazwę harmonogramu.

Parametry

<i>nazwa</i>	Nazwa harmonogramu.
--------------	---------------------

Definicja w linii [107](#) pliku [Harmonogram.java](#).

6.16.2.15 setNie()

```
void com.mklachl.sopkom.model.entity.Harmonogram.setNie (
    boolean nie )
```

Ustawia flagę wskazującą, czy harmonogram obowiązuje w niedzielę.

Parametry

<i>nie</i>	Flaga dla niedzieli.
------------	----------------------

Definicja w linii [219](#) pliku [Harmonogram.java](#).

6.16.2.16 setPia()

```
void com.mklachl.sopkom.model.entity.Harmonogram.setPia (
    boolean pia )
```

Ustawia flagę wskazującą, czy harmonogram obowiązuje w piątek.

Parametry

<i>pia</i>	Flaga dla piątku.
------------	-------------------

Definicja w linii [187](#) pliku [Harmonogram.java](#).

6.16.2.17 setPon()

```
void com.mklachl.sopkom.model.entity.Harmonogram.setPon (
    boolean pon )
```

Ustawia flagę wskazującą, czy harmonogram obowiązuje w poniedziałek.

Parametry

<i>pon</i>	Flaga dla poniedziałku.
------------	-------------------------

Definicja w linii [123](#) pliku [Harmonogram.java](#).

6.16.2.18 setSob()

```
void com.mklachl.sopkom.model.entity.Harmonogram.setSob (  
    boolean sob )
```

Ustawia flagę wskazującą, czy harmonogram obowiązuje w sobotę.

Parametry

<i>sob</i>	Flaga dla soboty.
------------	-------------------

Definicja w linii [203](#) pliku [Harmonogram.java](#).

6.16.2.19 setSro()

```
void com.mklachl.sopkom.model.entity.Harmonogram.setSro (  
    boolean sro )
```

Ustawia flagę wskazującą, czy harmonogram obowiązuje w środę.

Parametry

<i>sro</i>	Flaga dla środy.
------------	------------------

Definicja w linii [155](#) pliku [Harmonogram.java](#).

6.16.2.20 setWto()

```
void com.mklachl.sopkom.model.entity.Harmonogram.setWto (   
    boolean wto )
```

Ustawia flagę wskazującą, czy harmonogram obowiązuje we wtorek.

Parametry

<i>wto</i>	Flaga dla wtorku.
------------	-------------------

Definicja w linii [139](#) pliku [Harmonogram.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.17 Dokumentacja klasy com.mklachl.sopkom.controller.HarmonogramController

Metody publiczne

- ResponseEntity< HarmonogramDto > **createHarmonogram** (@RequestBody HarmonogramDto harmonogram)
- ResponseEntity< List< HarmonogramDto > > **getAllHarmonograms** ()
- ResponseEntity< HarmonogramDto > **getHarmonogramById** (@PathVariable(name="id") Short id)
- ResponseEntity< Void > **deleteHarmonogram** (@PathVariable Short id)

6.17.1 Opis szczegółowy

Kontroler dla zasobu Harmonogram. Obsługuje operacje tworzenia, pobierania i usuwania harmonogramów.

Definicja w linii 21 pliku [HarmonogramController.java](#).

6.17.2 Dokumentacja funkcji składowych

6.17.2.1 createHarmonogram()

```
ResponseEntity< HarmonogramDto > com.mklachl.sopkom.controller.HarmonogramController.create←
Harmonogram (←
    @RequestBody HarmonogramDto harmonogram )
```

Tworzy nowy harmonogram.

Parametry

<i>harmonogram</i>	Obiekt DTO harmonogramu.
--------------------	--------------------------

Zwraca

Utworzony obiekt DTO harmonogramu.

Definicja w linii 35 pliku [HarmonogramController.java](#).

6.17.2.2 deleteHarmonogram()

```
ResponseEntity< Void > com.mklachl.sopkom.controller.HarmonogramController.deleteHarmonogram (   
    @PathVariable Short id )
```

Usuwa harmonogram po identyfikatorze.

Parametry

<i>id</i>	Identyfikator harmonogramu.
-----------	-----------------------------

Zwroca

Status operacji.

Definicja w linii 72 pliku [HarmonogramController.java](#).

6.17.2.3 getAllHarmonograms()

```
ResponseEntity< List< HarmonogramDto > > com.mklachl.sopkom.controller.HarmonogramController.getAllHarmonograms ( )
```

Pobiera wszystkie harmonogramy.

Zwroca

Lista obiektów DTO harmonogramów.

Definicja w linii 44 pliku [HarmonogramController.java](#).

6.17.2.4 getHarmonogramById()

```
ResponseEntity< HarmonogramDto > com.mklachl.sopkom.controller.HarmonogramController.getHarmonogramById ( @PathVariable(name="id") Short id )
```

Pobiera harmonogram po identyfikatorze.

Parametry

<i>id</i>	Identyfikator harmonogramu.
-----------	-----------------------------

Zwroca

Obiekt DTO harmonogramu lub wiadomość o błędzie.

Definicja w linii 57 pliku [HarmonogramController.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/com

6.18 Dokumentacja klasy

com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto

Metody publiczne

- [HarmonogramDto \(\)](#)
- [HarmonogramDto \(Harmonogram harmonogram\)](#)
- short [getHarmonogramId \(\)](#)
- void [setHarmonogramId \(short harmonogramId\)](#)
- String [getNazwa \(\)](#)
- void [setNazwa \(String nazwa\)](#)
- boolean [isPon \(\)](#)
- void [setPon \(boolean pon\)](#)
- boolean [isWto \(\)](#)
- void [setWto \(boolean wto\)](#)
- boolean [isSro \(\)](#)
- void [setSro \(boolean sro\)](#)
- boolean [isCzw \(\)](#)
- void [setCzw \(boolean czw\)](#)
- boolean [isPia \(\)](#)
- void [setPia \(boolean pia\)](#)
- boolean [isSob \(\)](#)
- void [setSob \(boolean sob\)](#)
- boolean [isNie \(\)](#)
- void [setNie \(boolean nie\)](#)
- String [getDodatkowelInf \(\)](#)
- void [setDodatkowelInf \(String dodatkowelInf\)](#)

6.18.1 Opis szczegółowy

Data Transfer Object (DTO) dla encji Harmonogram.

Definicja w linii 8 pliku [HarmonogramDto.java](#).

6.18.2 Dokumentacja konstruktora i destruktora

6.18.2.1 HarmonogramDto() [1/2]

```
com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto.HarmonogramDto ( )
```

Konstruktor domyślny.

Definicja w linii 63 pliku [HarmonogramDto.java](#).

6.18.2.2 HarmonogramDto() [2/2]

```
com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto.HarmonogramDto (   
    Harmonogram harmonogram )
```

Konstruktor przyjmujący obiekt encji Harmonogram.

Parametry

<i>harmonogram</i>	Obiekt encji Harmonogram.
--------------------	---------------------------

Definicja w linii [70](#) pliku [HarmonogramDto.java](#).

6.18.3 Dokumentacja funkcji składowych

6.18.3.1 getDodatkoweInf()

`String com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto.getDodatkoweInf ()`

Zwraca dodatkowe informacje dotyczące harmonogramu.

Zwrota

`dodatkoweInf` Dodatkowe informacje dotyczące harmonogramu.

Definicja w linii [233](#) pliku [HarmonogramDto.java](#).

6.18.3.2 getHarmonogramId()

`short com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto.getHarmonogramId ()`

Zwraca identyfikator harmonogramu.

Zwrota

`harmonogramId` Identyfikator harmonogramu.

Definicja w linii [89](#) pliku [HarmonogramDto.java](#).

6.18.3.3 getNazwa()

`String com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto.getNazwa ()`

Zwraca nazwę harmonogramu.

Zwrota

`nazwa` Nazwa harmonogramu.

Definicja w linii [105](#) pliku [HarmonogramDto.java](#).

6.18.3.4 isCzw()

```
boolean com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto.isCzw ( )
```

Zwraca wartość, czy harmonogram dotyczy czwartku.

Zwraca

czw Wartość czy harmonogram dotyczy czwartku.

Definicja w linii [169](#) pliku [HarmonogramDto.java](#).

6.18.3.5 isNie()

```
boolean com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto.isNie ( )
```

Zwraca wartość, czy harmonogram dotyczy niedzieli.

Zwraca

nie Wartość czy harmonogram dotyczy niedzieli.

Definicja w linii [217](#) pliku [HarmonogramDto.java](#).

6.18.3.6 isPia()

```
boolean com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto.isPia ( )
```

Zwraca wartość, czy harmonogram dotyczy piątku.

Zwraca

pia Wartość czy harmonogram dotyczy piątku.

Definicja w linii [185](#) pliku [HarmonogramDto.java](#).

6.18.3.7 isPon()

```
boolean com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto.isPon ( )
```

Zwraca wartość, czy harmonogram dotyczy poniedziałku.

Zwraca

pon Wartość czy harmonogram dotyczy poniedziałku.

Definicja w linii [121](#) pliku [HarmonogramDto.java](#).

6.18.3.8 isSob()

```
boolean com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto.isSob ( )
```

Zwraca wartość, czy harmonogram dotyczy soboty.

Zwrota

sob Wartość czy harmonogram dotyczy soboty.

Definicja w linii 201 pliku [HarmonogramDto.java](#).

6.18.3.9 isSro()

```
boolean com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto.isSro ( )
```

Zwraca wartość, czy harmonogram dotyczy środy.

Zwrota

sro Wartość czy harmonogram dotyczy środy.

Definicja w linii 153 pliku [HarmonogramDto.java](#).

6.18.3.10 isWto()

```
boolean com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto.isWto ( )
```

Zwraca wartość, czy harmonogram dotyczy wtorku.

Zwrota

wto Wartość czy harmonogram dotyczy wtorku.

Definicja w linii 137 pliku [HarmonogramDto.java](#).

6.18.3.11 setCzw()

```
void com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto.setCzw ( boolean czw )
```

Ustawia wartość, czy harmonogram dotyczy czwartku.

Parametry

<i>czw</i>	Wartość czy harmonogram dotyczy czwartku.
------------	---

Definicja w linii [177](#) pliku [HarmonogramDto.java](#).

6.18.3.12 setDodatkoweInf()

```
void com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto.setDodatkoweInf (
    String dodatkoweInf )
```

Ustawia dodatkowe informacje dotyczące harmonogramu.

Parametry

<i>dodatkoweInf</i>	Dodatkowe informacje dotyczące harmonogramu.
---------------------	--

Definicja w linii [241](#) pliku [HarmonogramDto.java](#).

6.18.3.13 setHarmonogramId()

```
void com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto.setHarmonogramId (
    short harmonogramId )
```

Ustawia identyfikator harmonogramu.

Parametry

<i>harmonogram</i> ↗ <i>Id</i>	Identyfikator harmonogramu.
-----------------------------------	-----------------------------

Definicja w linii [97](#) pliku [HarmonogramDto.java](#).

6.18.3.14 setNazwa()

```
void com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto.setNazwa (
    String nazwa )
```

Ustawia nazwę harmonogramu.

Parametry

<i>nazwa</i>	Nazwa harmonogramu.
--------------	---------------------

Definicja w linii [113](#) pliku [HarmonogramDto.java](#).

6.18.3.15 setNie()

```
void com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto.setNie (
    boolean nie )
```

Ustawia wartość, czy harmonogram dotyczy niedzieli.

Parametry

<i>nie</i>	Wartość czy harmonogram dotyczy niedzieli.
------------	--

Definicja w linii [225](#) pliku [HarmonogramDto.java](#).

6.18.3.16 setPia()

```
void com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto.setPia (
    boolean pia )
```

Ustawia wartość, czy harmonogram dotyczy piątku.

Parametry

<i>pia</i>	Wartość czy harmonogram dotyczy piątku.
------------	---

Definicja w linii [193](#) pliku [HarmonogramDto.java](#).

6.18.3.17 setPon()

```
void com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto.setPon (
    boolean pon )
```

Ustawia wartość, czy harmonogram dotyczy poniedziałku.

Parametry

<i>pon</i>	Wartość czy harmonogram dotyczy poniedziałku.
------------	---

Definicja w linii [129](#) pliku [HarmonogramDto.java](#).

6.18.3.18 setSob()

```
void com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto.setSob (  
    boolean sob )
```

Ustawia wartość, czy harmonogram dotyczy soboty.

Parametry

<i>sob</i>	Wartość czy harmonogram dotyczy soboty.
------------	---

Definicja w linii [209](#) pliku [HarmonogramDto.java](#).

6.18.3.19 setSro()

```
void com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto.setSro (  
    boolean sro )
```

Ustawia wartość, czy harmonogram dotyczy środy.

Parametry

<i>sro</i>	Wartość czy harmonogram dotyczy środy.
------------	--

Definicja w linii [161](#) pliku [HarmonogramDto.java](#).

6.18.3.20 setWto()

```
void com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto.setWto (   
    boolean wto )
```

Ustawia wartość, czy harmonogram dotyczy wtorku.

Parametry

<i>wto</i>	Wartość czy harmonogram dotyczy wtorku.
------------	---

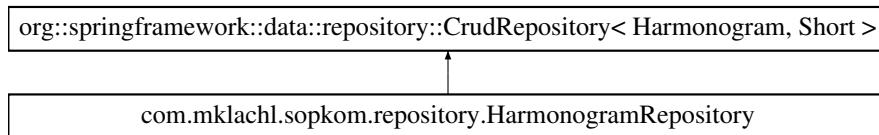
Definicja w linii [145](#) pliku [HarmonogramDto.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.19 Dokumentacja interfejsu com.mklachl.sopkom.repository.HarmonogramRepository

Diagram dziedziczenia dla com.mklachl.sopkom.repository.HarmonogramRepository



Metody publiczne

- List< [Harmonogram](#) > [findAllByNazwa](#) (String nazwa)
- List< [Harmonogram](#) > [findAllByPon](#) (boolean pon)
- List< [Harmonogram](#) > [findAllByWto](#) (boolean wto)
- List< [Harmonogram](#) > [findAllBySro](#) (boolean sro)
- List< [Harmonogram](#) > [findAllByCzw](#) (boolean czw)
- List< [Harmonogram](#) > [findAllByPia](#) (boolean pia)
- List< [Harmonogram](#) > [findAllBySob](#) (boolean sob)
- List< [Harmonogram](#) > [findAllByNie](#) (boolean nie)
- List< [Harmonogram](#) > [findAllByDodatkoweInflLike](#) (String dodatkoweInfl)
- List< [Harmonogram](#) > [findAll](#) ()

6.19.1 Opis szczegółowy

Interfejs repository dla encji Harmonogram

Definicja w linii 12 pliku [HarmonogramRepository.java](#).

6.19.2 Dokumentacja funkcji składowych

6.19.2.1 findAll()

```
List< Harmonogram > com.mklachl.sopkom.repository.HarmonogramRepository.findAll ( )
```

Znajduje wszystkie harmonogramy

Zwiera

Lista harmonogramów

6.19.2.2 findAllByCzw()

```
List< Harmonogram > com.mklachl.sopkom.repository.HarmonogramRepository.findAllByCzw ( boolean czw )
```

Znajduje wszystkie harmonogramy, które obowiązują w czwartki

Parametry

<i>czw</i>	true jeśli obowiązuje, false w przeciwnym razie
------------	---

Zwraca

Lista harmonogramów

6.19.2.3 findAllByDodatkoweInfLike()

```
List< Harmonogram > com.mklachl.sopkom.repository.HarmonogramRepository.findAllByDodatkowe←
InfLike (
    String dodatkoweInf )
```

Znajduje wszystkie harmonogramy o dodatkowych informacjach pasujących do wzorca

Parametry

<i>dodatkoweInf</i>	dodatkowe informacje
---------------------	----------------------

Zwraca

Lista harmonogramów

6.19.2.4 findAllByNazwa()

```
List< Harmonogram > com.mklachl.sopkom.repository.HarmonogramRepository.findAllByNazwa (
    String nazwa )
```

Znajduje wszystkie harmonogramy o określonej nazwie

Parametry

<i>nazwa</i>	nazwa
--------------	-------

Zwraca

Lista harmonogramów

6.19.2.5 findAllByNie()

```
List< Harmonogram > com.mklachl.sopkom.repository.HarmonogramRepository.findAllByNie (  
    boolean nie )
```

Znajduje wszystkie harmonogramy, które obowiązują w niedziele

Parametry

<i>nie</i>	true jeśli obowiązuje, false w przeciwnym razie
------------	---

Zwroca

Lista harmonogramów

6.19.2.6 findAllByPia()

```
List< Harmonogram > com.mklachl.sopkom.repository.HarmonogramRepository.findAllByPia (  
    boolean pia )
```

Znajduje wszystkie harmonogramy, które obowiązują w piątki

Parametry

<i>pia</i>	true jeśli obowiązuje, false w przeciwnym razie
------------	---

Zwroca

Lista harmonogramów

6.19.2.7 findAllByPon()

```
List< Harmonogram > com.mklachl.sopkom.repository.HarmonogramRepository.findAllByPon (   
    boolean pon )
```

Znajduje wszystkie harmonogramy, które obowiązują w poniedziałki

Parametry

<i>pon</i>	true jeśli obowiązuje, false w przeciwnym razie
------------	---

Zwrota

Lista harmonogramów

6.19.2.8 findAllBySob()

```
List< Harmonogram > com.mklachl.sopkom.repository.HarmonogramRepository.findAllBySob ( boolean sob )
```

Znajduje wszystkie harmonogramy, które obowiązują w soboty

Parametry

sob	true jeśli obowiązuje, false w przeciwnym razie
-----	---

Zwrota

Lista harmonogramów

6.19.2.9 findAllBySro()

```
List< Harmonogram > com.mklachl.sopkom.repository.HarmonogramRepository.findAllBySro ( boolean sro )
```

Znajduje wszystkie harmonogramy, które obowiązują w środy

Parametry

sro	true jeśli obowiązuje, false w przeciwnym razie
-----	---

Zwrota

Lista harmonogramów

6.19.2.10 findAllByWto()

```
List< Harmonogram > com.mklachl.sopkom.repository.HarmonogramRepository.findAllByWto ( boolean wto )
```

Znajduje wszystkie harmonogramy, które obowiązują we wtorki

Parametry

wto	true jeśli obowiązuje, false w przeciwnym razie
-----	---

Zwraca

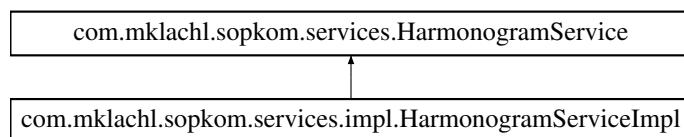
Lista harmonogramów

Dokumentacja dla tego interfejsu została wygenerowana z pliku:

- X:/java/TAB/Dxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/repo...

6.20 Dokumentacja interfejsu com.mklachl.sopkom.services.HarmonogramService

Diagram dziedziczenia dla com.mklachl.sopkom.services.HarmonogramService



Metody publiczne

- [Harmonogram saveHarmonogram \(Harmonogram harmonogram\)](#)
- [Harmonogram saveHarmonogram \(HarmonogramDto harmonogramDto\)](#)
- [Optional< Harmonogram > findHarmonogramById \(Short id\)](#)
- [List< Harmonogram > findHarmonogramByNazwa \(String nazwa\)](#)
- [List< Harmonogram > findHarmonogramByPon \(boolean pon\)](#)
- [List< Harmonogram > findHarmonogramByWto \(boolean wto\)](#)
- [List< Harmonogram > findHarmonogramBySro \(boolean sro\)](#)
- [List< Harmonogram > findHarmonogramByCzw \(boolean czw\)](#)
- [List< Harmonogram > findHarmonogramByPia \(boolean pia\)](#)
- [List< Harmonogram > findHarmonogramBySob \(boolean sob\)](#)
- [List< Harmonogram > findHarmonogramByNie \(boolean nie\)](#)
- [List< Harmonogram > findHarmonogramByDodatkoweInflLike \(String dodatkoweInfl\)](#)

6.20.1 Opis szczegółowy

Interfejs serwisu dla obsługi harmonogramów

Definicja w linii 12 pliku [HarmonogramService.java](#).

6.20.2 Dokumentacja funkcji składowych

6.20.2.1 findHarmonogramByCzw()

```
List< Harmonogram > com.mklachl.sopkom.services.HarmonogramService.findHarmonogramByCzw ( boolean czw )
```

Znajduje harmonogramy, które są aktywne w czwartki

Parametry

czw	czwartek
-----	----------

Zwraca

lista harmonogramów

Implementowany w [com.mklachl.sopkom.services.impl.HarmonogramServiceImpl](#).

6.20.2.2 findHarmonogramByDodatkoweInfLike()

```
List< Harmonogram > com.mklachl.sopkom.services.HarmonogramService.findHarmonogramByDodatkowe←  
InfLike (←  
    String dodatkoweInf )
```

Znajduje harmonogramy, które pasują do podanych dodatkowych informacji

Parametry

dodatkoweInf	dodatkowe informacje
--------------	----------------------

Zwraca

lista harmonogramów

Implementowany w [com.mklachl.sopkom.services.impl.HarmonogramServiceImpl](#).

6.20.2.3 findHarmonogramById()

```
Optional< Harmonogram > com.mklachl.sopkom.services.HarmonogramService.findHarmonogramById (←  
    Short id )
```

Znajduje harmonogram po ID

Parametry

id	ID harmonogramu
----	-----------------

Zwraca

Optional z harmonogramem

Implementowany w [com.mklachl.sopkom.services.impl.HarmonogramServiceImpl](#).

6.20.2.4 findHarmonogramByNazwa()

```
List< Harmonogram > com.mklachl.sopkom.services.HarmonogramService.findHarmonogramByNazwa (  
    String nazwa )
```

Znajduje harmonogramy po nazwie

Parametry

<i>nazwa</i>	nazwa
--------------	-------

Zwroca

lista harmonogramów

Implementowany w [com.mklachl.sopkom.services.impl.HarmonogramServiceImpl](#).

6.20.2.5 findHarmonogramByNie()

```
List< Harmonogram > com.mklachl.sopkom.services.HarmonogramService.findHarmonogramByNie (   
    boolean nie )
```

Znajduje harmonogramy, które są aktywne w niedziele

Parametry

<i>nie</i>	niedziela
------------	-----------

Zwroca

lista harmonogramów

Implementowany w [com.mklachl.sopkom.services.impl.HarmonogramServiceImpl](#).

6.20.2.6 findHarmonogramByPia()

```
List< Harmonogram > com.mklachl.sopkom.services.HarmonogramService.findHarmonogramByPia (   
    boolean pia )
```

Znajduje harmonogramy, które są aktywne w piątki

Parametry

<i>pia</i>	piątek
------------	--------

Zwroca

lista harmonogramów

Implementowany w [com.mklachl.sopkom.services.impl.HarmonogramServiceImpl](#).

6.20.2.7 findHarmonogramByPon()

```
List< Harmonogram > com.mklachl.sopkom.services.HarmonogramService.findHarmonogramByPon ( boolean pon )
```

Znajduje harmonogramy, które są aktywne w poniedziałki

Parametry

<i>pon</i>	poniedziałek
------------	--------------

Zwroca

lista harmonogramów

Implementowany w [com.mklachl.sopkom.services.impl.HarmonogramServiceImpl](#).

6.20.2.8 findHarmonogramBySob()

```
List< Harmonogram > com.mklachl.sopkom.services.HarmonogramService.findHarmonogramBySob ( boolean sob )
```

Znajduje harmonogramy, które są aktywne w soboty

Parametry

<i>sob</i>	sobota
------------	--------

Zwroca

lista harmonogramów

Implementowany w [com.mklachl.sopkom.services.impl.HarmonogramServiceImpl](#).

6.20.2.9 findHarmonogramBySro()

```
List< Harmonogram > com.mklachl.sopkom.services.HarmonogramService.findHarmonogramBySro ( boolean sro )
```

Znajduje harmonogramy, które są aktywne w środy

Parametry

<i>sro</i>	środa
------------	-------

Zwroca

lista harmonogramów

Implementowany w [com.mklachl.sopkom.services.impl.HarmonogramServiceImpl](#).

6.20.2.10 findHarmonogramByWto()

```
List< Harmonogram > com.mklachl.sopkom.services.HarmonogramService.findHarmonogramByWto ( boolean wto )
```

Znajduje harmonogramy, które są aktywne we wtorki

Parametry

<i>wto</i>	wtorek
------------	--------

Zwroca

lista harmonogramów

Implementowany w [com.mklachl.sopkom.services.impl.HarmonogramServiceImpl](#).

6.20.2.11 saveHarmonogram() [1/2]

```
Harmonogram com.mklachl.sopkom.services.HarmonogramService.saveHarmonogram ( Harmonogram harmonogram )
```

Zapisuje nowy harmonogram

Parametry

<i>harmonogram</i>	encja harmonogram
--------------------	-------------------

Zwroca

zapisany harmonogram

Implementowany w [com.mklachl.sopkom.services.impl.HarmonogramServiceImpl](#).

6.20.2.12 saveHarmonogram() [2/2]

```
Harmonogram com.mklachl.sopkom.services.HarmonogramService.saveHarmonogram (
    HarmonogramDto harmonogramDto )
```

Zapisuje nowy harmonogram na podstawie DTO

Parametry

<i>harmonogramDto</i>	DTO harmonogramu
-----------------------	------------------

Zwroca

zapisany harmonogram

Implementowany w [com.mklachl.sopkom.services.impl.HarmonogramServiceImpl](#).

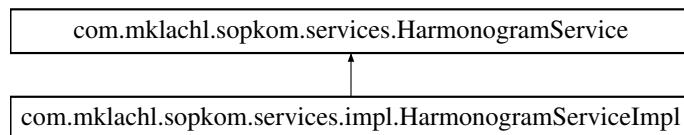
Dokumentacja dla tego interfejsu została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/se

6.21 Dokumentacja klasy

[com.mklachl.sopkom.services.impl.HarmonogramServiceImpl](#)

Diagram dziedziczenia dla [com.mklachl.sopkom.services.impl.HarmonogramServiceImpl](#)



Metody publiczne

- [HarmonogramServiceImpl \(HarmonogramRepository\) harmonogramRepository](#)
- [Harmonogram saveHarmonogram \(Harmonogram harmonogramDto\)](#)
- [Harmonogram saveHarmonogram \(HarmonogramDto harmonogramDto\)](#)
- [Optional< Harmonogram > findHarmonogramById \(Short id\)](#)
- [List< Harmonogram > findHarmonogramByNazwa \(String nazwa\)](#)
- [List< Harmonogram > findHarmonogramByPon \(boolean pon\)](#)
- [List< Harmonogram > findHarmonogramByWto \(boolean wto\)](#)
- [List< Harmonogram > findHarmonogramBySro \(boolean sro\)](#)
- [List< Harmonogram > findHarmonogramByCzw \(boolean czw\)](#)
- [List< Harmonogram > findHarmonogramByPia \(boolean pia\)](#)
- [List< Harmonogram > findHarmonogramBySob \(boolean sob\)](#)
- [List< Harmonogram > findHarmonogramByNie \(boolean nie\)](#)
- [List< Harmonogram > findHarmonogramByDodatkoweInflLike \(String dodatkoweInfl\)](#)

6.21.1 Opis szczegółowy

Definicja w linii 14 pliku [HarmonogramServiceImpl.java](#).

6.21.2 Dokumentacja konstruktora i destruktora

6.21.2.1 HarmonogramServiceImpl()

```
com.mklachl.sopkom.services.impl.HarmonogramServiceImpl.HarmonogramServiceImpl (   
    HarmonogramRepository harmonogramRepository )
```

Definicja w linii 17 pliku [HarmonogramServiceImpl.java](#).

6.21.3 Dokumentacja funkcji składowych

6.21.3.1 findHarmonogramByCzw()

```
List< Harmonogram > com.mklachl.sopkom.services.impl.HarmonogramServiceImpl.findHarmonogramByCzw (   
    boolean czw )
```

Znajduje harmonogramy, które są aktywne w czwartki

Parametry

<code>czw</code>	czwartek
------------------	----------

Zwraca

lista harmonogramów

Implementuje [com.mklachl.sopkom.services.HarmonogramService](#).

Definicja w linii 68 pliku [HarmonogramServiceImpl.java](#).

6.21.3.2 findHarmonogramByDodatkoweInfLike()

```
List< Harmonogram > com.mklachl.sopkom.services.impl.HarmonogramServiceImpl.findHarmonogramByDodatkoweInfLike (   
    String dodatkoweInf )
```

Znajduje harmonogramy, które pasują do podanych dodatkowych informacji

Parametry

<i>dodatkowelnf</i>	dodatkowe informacje
---------------------	----------------------

Zwroca

lista harmonogramów

Implementuje [com.mklachl.sopkom.services.HarmonogramService](#).

Definicja w linii 80 pliku [HarmonogramServiceImpl.java](#).

6.21.3.3 findHarmonogramById()

```
Optional< Harmonogram > com.mklachl.sopkom.services.impl.HarmonogramServiceImpl.findHarmonogramById ( Short id )
```

Znajduje harmonogram po ID

Parametry

<i>id</i>	ID harmonogramu
-----------	-----------------

Zwroca

Optional z harmonogramem

Implementuje [com.mklachl.sopkom.services.HarmonogramService](#).

Definicja w linii 53 pliku [HarmonogramServiceImpl.java](#).

6.21.3.4 findHarmonogramByNazwa()

```
List< Harmonogram > com.mklachl.sopkom.services.impl.HarmonogramServiceImpl.findHarmonogramByNazwa ( String nazwa )
```

Znajduje harmonogramy po nazwie

Parametry

<i>nazwa</i>	nazwa
--------------	-------

Zwrota

lista harmonogramów

Implementuje [com.mklachl.sopkom.services.HarmonogramService](#).

Definicja w linii [56](#) pliku [HarmonogramServiceImpl.java](#).

6.21.3.5 findHarmonogramByNie()

```
List< Harmonogram > com.mklachl.sopkom.services.impl.HarmonogramServiceImpl.findHarmonogramByNie (boolean nie )
```

Znajduje harmonogramy, które są aktywne w niedziele

Parametry

nie	niedziela
-----	-----------

Zwrota

lista harmonogramów

Implementuje [com.mklachl.sopkom.services.HarmonogramService](#).

Definicja w linii [77](#) pliku [HarmonogramServiceImpl.java](#).

6.21.3.6 findHarmonogramByPia()

```
List< Harmonogram > com.mklachl.sopkom.services.impl.HarmonogramServiceImpl.findHarmonogramByPia (boolean pia )
```

Znajduje harmonogramy, które są aktywne w piątki

Parametry

pia	piątek
-----	--------

Zwrota

lista harmonogramów

Implementuje [com.mklachl.sopkom.services.HarmonogramService](#).

Definicja w linii [71](#) pliku [HarmonogramServiceImpl.java](#).

6.21.3.7 findHarmonogramByPon()

```
List< Harmonogram > com.mklachl.sopkom.services.impl.HarmonogramServiceImpl.findHarmonogram→  
ByPon (   
    boolean pon )
```

Znajduje harmonogramy, które są aktywne w poniedziałki

Parametry

<i>pon</i>	poniedziałek
------------	--------------

Zwroca

lista harmonogramów

Implementuje [com.mklachl.sopkom.services.HarmonogramService](#).

Definicja w linii 59 pliku [HarmonogramServiceImpl.java](#).

6.21.3.8 findHarmonogramBySob()

```
List< Harmonogram > com.mklachl.sopkom.services.impl.HarmonogramServiceImpl.findHarmonogram→  
BySob (   
    boolean sob )
```

Znajduje harmonogramy, które są aktywne w soboty

Parametry

<i>sob</i>	sobota
------------	--------

Zwroca

lista harmonogramów

Implementuje [com.mklachl.sopkom.services.HarmonogramService](#).

Definicja w linii 74 pliku [HarmonogramServiceImpl.java](#).

6.21.3.9 findHarmonogramBySro()

```
List< Harmonogram > com.mklachl.sopkom.services.impl.HarmonogramServiceImpl.findHarmonogram→  
BySro (   
    boolean sro )
```

Znajduje harmonogramy, które są aktywne w środy

Parametry

sro	środa
-----	-------

Zwroca

lista harmonogramów

Implementuje [com.mklachl.sopkom.services.HarmonogramService](#).

Definicja w linii [65](#) pliku [HarmonogramServiceImpl.java](#).

6.21.3.10 findHarmonogramByWto()

```
List< Harmonogram > com.mklachl.sopkom.services.impl.HarmonogramServiceImpl.findHarmonogramByWto ( boolean wto )
```

Znajduje harmonogramy, które są aktywne we wtorki

Parametry

wto	wtorek
-----	--------

Zwroca

lista harmonogramów

Implementuje [com.mklachl.sopkom.services.HarmonogramService](#).

Definicja w linii [62](#) pliku [HarmonogramServiceImpl.java](#).

6.21.3.11 saveHarmonogram() [1/2]

```
Harmonogram com.mklachl.sopkom.services.impl.HarmonogramServiceImpl.saveHarmonogram ( Harmonogram harmonogram )
```

Zapisuje nowy harmonogram

Parametry

harmonogram	encja harmonogram
-------------	-------------------

Zwroca

zapisany harmonogram

Implementuje [com.mklachl.sopkom.services.HarmonogramService](#).

Definicja w linii [20](#) pliku [HarmonogramServiceImpl.java](#).

6.21.3.12 saveHarmonogram() [2/2]

```
Harmonogram com.mklachl.sopkom.services.impl.HarmonogramServiceImpl.saveHarmonogram (
    HarmonogramDto harmonogramDto )
```

Zapisuje nowy harmonogram na podstawie DTO

Parametry

<i>harmonogramDto</i>	DTO harmonogramu
-----------------------	------------------

Zwroca

zapisany harmonogram

Implementuje [com.mklachl.sopkom.services.HarmonogramService](#).

Definicja w linii [36](#) pliku [HarmonogramServiceImpl.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/se

6.22 Dokumentacja klasy com.mklachl.sopkom.helpers.Helpers

Statyczne metody publiczne

- static void [inject](#) (Model model)
- static [User getUserFromContext](#) ([UserService](#) service)

6.22.1 Opis szczegółowy

Definicja w linii [11](#) pliku [Helpers.java](#).

6.22.2 Dokumentacja funkcji składowych

6.22.2.1 getUserFromContext()

```
static User com.mklachl.sopkom.helpers.Helpers.getUserFromContext (
    UserService service ) [static]
```

Definicja w linii 25 pliku [Helpers.java](#).

6.22.2.2 inject()

```
static void com.mklachl.sopkom.helpers.Helpers.inject (
    Model model ) [static]
```

Definicja w linii 15 pliku [Helpers.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/helpers/Helpers.java

6.23 Dokumentacja klasy com.mklachl.sopkom.model.entity.Incydent

Metody publiczne

- Long [getIncydentId](#) ()
- void [setIncydentId](#) (Long incydentId)
- String [getTyp](#) ()
- void [setTyp](#) (String typ)
- Date [getDate](#) ()
- void [setDate](#) (Date date)
- Autobus [getAutobus](#) ()
- void [setAutobus](#) (Autobus autobus)
- Kierowca [getKierowca](#) ()
- void [setKierowca](#) (Kierowca kierowca)
- Przejazd [getPrzejazd](#) ()
- void [setPrzejazd](#) (Przejazd przejazd)
- String [getKrotko](#) ()
- void [setKrotko](#) (String krotko)
- String [getDodatkoweInformacje](#) ()
- void [setDodatkoweInformacje](#) (String dodatkoweInformacje)
- float [getKoszty](#) ()
- void [setKoszty](#) (float koszty)

6.23.1 Opis szczegółowy

Klasa reprezentująca encję [Incydent](#).

Definicja w linii 11 pliku [Incydent.java](#).

6.23.2 Dokumentacja funkcji składowych

6.23.2.1 getAutobus()

```
Autobus com.mklachl.sopkom.model.entity.Incydent.getAutobus ( )
```

Zwraca autobus związany z incydentem.

Zwraca

autobus [Autobus](#) związany z incydentem.

Definicja w linii [125](#) pliku [Incydent.java](#).

6.23.2.2 getDate()

```
Date com.mklachl.sopkom.model.entity.Incydent.getDate ( )
```

Zwraca datę incydentu.

Zwraca

date Data incydentu.

Definicja w linii [109](#) pliku [Incydent.java](#).

6.23.2.3 getDodatkoweInformacje()

```
String com.mklachl.sopkom.model.entity.Incydent.getDodatkoweInformacje ( )
```

Zwraca dodatkowe informacje dotyczące incydentu.

Zwraca

dodatkoweInformacje Dodatkowe informacje.

Definicja w linii [189](#) pliku [Incydent.java](#).

6.23.2.4 getIncydentId()

```
Long com.mklachl.sopkom.model.entity.Incydent.getIncydentId ( )
```

Zwraca identyfikator incydentu.

Zwraca

incydentId Identyfikator incydentu.

Definicja w linii 77 pliku [Incydent.java](#).

6.23.2.5 getKierowca()

```
Kierowca com.mklachl.sopkom.model.entity.Incydent.getKierowca ( )
```

Zwraca kierowcę związanego z incydentem.

Zwraca

kierowca [Kierowca](#) związany z incydentem.

Definicja w linii 141 pliku [Incydent.java](#).

6.23.2.6 getKoszty()

```
float com.mklachl.sopkom.model.entity.Incydent.getKoszty ( )
```

Zwraca koszty związane z incydentem.

Zwraca

koszty Koszty związane z incydentem.

Definicja w linii 205 pliku [Incydent.java](#).

6.23.2.7 getKrotko()

```
String com.mklachl.sopkom.model.entity.Incydent.getKrotko ( )
```

Zwraca krótki opis incydentu.

Zwraca

krotko Krótki opis incydentu.

Definicja w linii 173 pliku [Incydent.java](#).

6.23.2.8 getPrzejazd()

```
Przejazd com.mklachl.sopkom.model.entity.Incydent.getPrzejazd ( )
```

Zwraca przejazd związany z incydentem.

Zwrota

przejazd **Przejazd** związany z incydentem.

Definicja w linii 157 pliku [Incydent.java](#).

6.23.2.9 getTyp()

```
String com.mklachl.sopkom.model.entity.Incydent.getTyp ( )
```

Zwraca typ incydentu.

Zwrota

typ **Typ incydentu**.

Definicja w linii 93 pliku [Incydent.java](#).

6.23.2.10 setAutobus()

```
void com.mklachl.sopkom.model.entity.Incydent.setAutobus (
    Autobus autobus )
```

Ustawia autobus związany z incydentem.

Parametry

<i>autobus</i>	Autobus związany z incydentem.
----------------	---------------------------------------

Definicja w linii 133 pliku [Incydent.java](#).

6.23.2.11 setDate()

```
void com.mklachl.sopkom.model.entity.Incydent.setDate (
    Date date )
```

Ustawia datę incydentu.

Parametry

<i>date</i>	Data incydentu.
-------------	-----------------

Definicja w linii [117](#) pliku [Incydent.java](#).

6.23.2.12 setDodatkoweInformacje()

```
void com.mklachl.sopkom.model.entity.Incydent.setDodatkoweInformacje (
    String dodatkoweInformacje )
```

Ustawia dodatkowe informacje dotyczące incydentu.

Parametry

<i>dodatkoweInformacje</i>	Dodatkowe informacje.
----------------------------	-----------------------

Definicja w linii [197](#) pliku [Incydent.java](#).

6.23.2.13 setId()

```
void com.mklachl.sopkom.model.entity.Incydent.setId (
    Long id )
```

Ustawia identyfikator incydentu.

Parametry

<i>incydent</i> ↪ <i>id</i>	Identyfikator incydentu.
--------------------------------	--------------------------

Definicja w linii [85](#) pliku [Incydent.java](#).

6.23.2.14 setKierowca()

```
void com.mklachl.sopkom.model.entity.Incydent.setKierowca (
    Kierowca kierowca )
```

Ustawia kierowcę związanego z incydentem.

Parametry

<i>kierowca</i>	Kierowca związany z incydentem.
-----------------	---------------------------------

Definicja w linii [149](#) pliku [Incydent.java](#).

6.23.2.15 setKoszty()

```
void com.mklachl.sopkom.model.entity.Incydent.setKoszty (
    float koszty )
```

Ustawia koszty związane z incydentem.

Parametry

<i>koszty</i>	Koszty związane z incydentem.
---------------	-------------------------------

Definicja w linii [213](#) pliku [Incydent.java](#).

6.23.2.16 setKrotko()

```
void com.mklachl.sopkom.model.entity.Incydent.setKrotko (
    String krotko )
```

Ustawia krótki opis incydentu.

Parametry

<i>krotko</i>	Krótki opis incydentu.
---------------	------------------------

Definicja w linii [181](#) pliku [Incydent.java](#).

6.23.2.17 setPrzejazd()

```
void com.mklachl.sopkom.model.entity.Incydent.setPrzejazd (
    Przejazd przejazd )
```

Ustawia przejazd związany z incydentem.

Parametry

<i>przejazd</i>	Przejazd związany z incydentem.
-----------------	---------------------------------

Definicja w linii 165 pliku [Incydent.java](#).

6.23.2.18 setTyp()

```
void com.mklachl.sopkom.model.entity.Incydent.setTyp (
    String typ )
```

Ustawia typ incydentu.

Parametry

<i>typ</i>	Typ incydentu.
------------	----------------

Definicja w linii 101 pliku [Incydent.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.24 Dokumentacja klasy com.mklachl.sopkom.controller.IncydentController

Metody publiczne

- ResponseEntity<?> [saveIncydent](#) (@RequestBody IncydentDto input)
- ResponseEntity<?> [updateIncydent](#) (@PathVariable(name="id") Long id, @RequestBody IncydentDto input)
- ResponseEntity<?> [getIncydent](#) (@PathVariable(name="incydent_id") Long incydentId)
- ResponseEntity<?> [deleteIncydent](#) (@PathVariable(name="incydent_id") Long incydentId)
- ResponseEntity<?> [getIncydents](#) (@RequestParam(name="typ", required=false) String typ)
- ResponseEntity<?> [getIncydentsBeforeDate](#) (@RequestParam(name="date") @DateTimeFormat(pattern="dd. MM.yyyy") Date date)
- ResponseEntity<?> [getIncydentsAfterDate](#) (@RequestParam(name="date") @DateTimeFormat(pattern="dd. MM.yyyy") Date date)
- ResponseEntity<?> [getIncydentsBetweenDates](#) (@RequestParam(name="startDate") @DateTimeFormat(pattern="dd.MM.yyyy") Date startDate, @RequestParam(name="endDate") @DateTimeFormat(pattern="dd.MM.yyyy") Date endDate)
- ResponseEntity<?> [getIncydentsByAutobus](#) (@RequestParam(name="autobus_id") Long autobusId)
- ResponseEntity<?> [getIncydentsByKierowca](#) (@RequestParam(name="kierowca_id") Long kierowcaId)
- ResponseEntity<?> [getIncydentsByPrzejazd](#) (@RequestParam(name="przejazd_id") Long przejazdId)

Atrybuty publiczne

- [IncydentService](#) incydentService
- [IncydentRepository](#) incydentRepository

6.24.1 Opis szczegółowy

Kontroler dla zasobu Incydent. Obsługuje operacje tworzenia, pobierania, aktualizacji i usuwania incydentów.

Definicja w linii 28 pliku [IncydentController.java](#).

6.24.2 Dokumentacja funkcji składowych

6.24.2.1 deleteIncydent()

```
ResponseEntity<?> com.mklachl.sopkom.controller.IncydentController.deleteIncydent ( @PathVariable(name="incydent_id") Long incydentId )
```

Usuwa incydent po identyfikatorze.

Parametry

<i>incydent</i> ↪ <i>Id</i>	Identyfikator incydentu.
--------------------------------	--------------------------

Zwraca

Status operacji lub wiadomość o błędzie.

Definicja w linii 93 pliku [IncydentController.java](#).

6.24.2.2 getIncydent()

```
ResponseEntity<?> com.mklachl.sopkom.controller.IncydentController.getIncydent ( @PathVariable(name="incydent_id") Long incydentId )
```

Pobiera incydent po identyfikatorze.

Parametry

<i>incydent</i> ↪ <i>Id</i>	Identyfikator incydentu.
--------------------------------	--------------------------

Zwraca

Obiekt DTO incydentu lub wiadomość o błędzie.

Definicja w linii 77 pliku [IncydentController.java](#).

6.24.2.3 getIncydents()

```
ResponseEntity<?> com.mklachl.sopkom.controller.IncydentController.getIncydents ( @RequestParam(name="typ", required=false) String typ )
```

Pobiera wszystkie incydenty lub filtrowane po typie.

Parametry

<i>typ</i>	Typ incydentu (opcjonalny).
------------	-----------------------------

Zwroca

Lista obiektów DTO incydentów.

Definicja w linii 108 pliku [IncydentController.java](#).

6.24.2.4 getIncydentsAfterDate()

```
ResponseEntity<?> com.mklachl.sopkom.controller.IncydentController.getIncydentsAfterDate ( @RequestParam(name="date") @DateTimeFormat(pattern="dd.MM.yyyy") Date date )
```

Pobiera incydenty po określonej dacie.

Parametry

<i>date</i>	Data w formacie dd.MM.yyyy.
-------------	-----------------------------

Zwroca

Lista obiektów DTO incydentów.

Definicja w linii 138 pliku [IncydentController.java](#).

6.24.2.5 getIncydentsBeforeDate()

```
ResponseEntity<?> com.mklachl.sopkom.controller.IncydentController.getIncydentsBeforeDate ( @RequestParam(name="date") @DateTimeFormat(pattern="dd.MM.yyyy") Date date )
```

Pobiera incydenty przed określoną datą.

Parametry

<i>date</i>	Data w formacie dd.MM.yyyy.
-------------	-----------------------------

Zwrota

Lista obiektów DTO incydentów.

Definicja w linii 123 pliku [IncydentController.java](#).

6.24.2.6 getIncydentsBetweenDates()

```
ResponseEntity<?> com.mklachl.sopkom.controller.IncydentController.getIncydentsBetweenDates ( @RequestParam(name="startDate") @DateTimeFormat(pattern="dd.MM.yyyy") Date start↔ Date, @RequestParam(name="endDate") @DateTimeFormat(pattern="dd.MM.yyyy") Date endDate )
```

Pobiera incydenty pomiędzy dwiema datami.

Parametry

<i>startDate</i>	Data początkowa w formacie dd.MM.yyyy.
<i>endDate</i>	Data końcowa w formacie dd.MM.yyyy.

Zwrota

Lista obiektów DTO incydentów.

Definicja w linii 154 pliku [IncydentController.java](#).

6.24.2.7 getIncydentsByAutobus()

```
ResponseEntity<?> com.mklachl.sopkom.controller.IncydentController.getIncydentsByAutobus ( @RequestParam(name="autobus_id") Long autobusId )
```

Pobiera incydenty po identyfikatorze autobusu.

Parametry

<i>autobus↔ Id</i>	Identyfikator autobusu.
--------------------	-------------------------

Zwrota

Lista obiektów DTO incydentów.

Definicja w linii 170 pliku [IncydentController.java](#).

6.24.2.8 getIncydentsByKierowca()

```
ResponseEntity<?> com.mklachl.sopkom.controller.IncydentController.getIncydentsByKierowca ( @RequestParam(name="kierowca_id") Long kierowcaId )
```

Pobiera incydenty po identyfikatorze kierowcy.

Parametry

<i>kierowca</i> ↵ <i>Id</i>	Identyfikator kierowcy.
--------------------------------	-------------------------

Zwrota

Lista obiektów DTO incydentów.

Definicja w linii [187](#) pliku [IncydentController.java](#).

6.24.2.9 getIncydentsByPrzejazd()

```
ResponseEntity<?> com.mklachl.sopkom.controller.IncydentController.getIncydentsByPrzejazd ( @RequestParam(name="przejazd_id") Long przejazdId )
```

Pobiera incydenty po identyfikatorze przejazdu.

Parametry

<i>przejazd</i> ↵ <i>Id</i>	Identyfikator przejazdu.
--------------------------------	--------------------------

Zwrota

Lista obiektów DTO incydentów.

Definicja w linii [204](#) pliku [IncydentController.java](#).

6.24.2.10 saveIncydent()

```
ResponseEntity<?> com.mklachl.sopkom.controller.IncydentController.saveIncydent ( @RequestBody IncydentDto input )
```

Tworzy nowy incydent.

Parametry

<i>input</i>	Obiekt DTO incydentu.
--------------	-----------------------

Zwroca

Utworzony obiekt DTO incydentu lub wiadomość o błędzie.

Definicja w linii 42 pliku [IncydentController.java](#).

6.24.2.11 updateIncydent()

```
ResponseEntity<?> com.mklachl.sopkom.controller.IncydentController.updateIncydent ( @PathVariable(name="id") Long id, @RequestBody IncydentDto input )
```

Aktualizuje incydent po identyfikatorze.

Parametry

<i>id</i>	Identyfikator incydentu.
<i>input</i>	Obiekt DTO incydentu.

Zwroca

Status operacji lub wiadomość o błędzie.

Definicja w linii 61 pliku [IncydentController.java](#).

6.24.3 Dokumentacja atrybutów składowych

6.24.3.1 incydentRepository

```
IncydentRepository com.mklachl.sopkom.controller.IncydentController.incydentRepository
```

Definicja w linii 34 pliku [IncydentController.java](#).

6.24.3.2 incydentService

```
IncydentService com.mklachl.sopkom.controller.IncydentController.incydentService
```

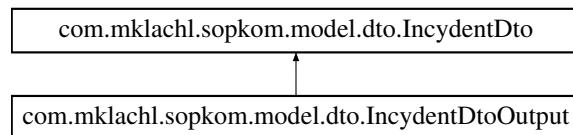
Definicja w linii 31 pliku [IncydentController.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/

6.25 Dokumentacja klasy com.mklachl.sopkom.model.dto.IncydentDto

Diagram dziedziczenia dla com.mklachl.sopkom.model.dto.IncydentDto



Metody publiczne

- `IncydentDto ()`
- `IncydentDto (Incydent incydent)`
- `String getKrotko ()`
- `void setKrotko (String krotko)`
- `Long getIncydentId ()`
- `void setIncydentId (Long incydentId)`
- `String getTyp ()`
- `void setTyp (String typ)`
- `Date getDate ()`
- `void setDate (Date date)`
- `Long getAutobusId ()`
- `void setAutobusId (Long autobusId)`
- `Long getKierowcaId ()`
- `void setKierowcaId (Long kierowcaId)`
- `Long getPrzejazdId ()`
- `void setPrzejazdId (Long przejazdId)`
- `String getDodatkoweInformacje ()`
- `void setDodatkoweInformacje (String dodatkoweInformacje)`
- `float getKoszty ()`
- `void setKoszty (float koszty)`

6.25.1 Opis szczegółowy

Data Transfer Object (DTO) dla encji Incydent.

Definicja w linii 10 pliku [IncydentDto.java](#).

6.25.2 Dokumentacja konstruktora i destruktora

6.25.2.1 IncydentDto() [1/2]

```
com.mklachl.sopkom.model.dto.IncydentDto.IncydentDto ( )
```

Konstruktor domyślny.

Definicja w linii 61 pliku [IncydentDto.java](#).

6.25.2.2 IncydentDto() [2/2]

```
com.mklachl.sopkom.model.dto.IncydentDto.IncydentDto (  
    Incydent incydent )
```

Konstruktor przyjmujący obiekt encji Incydent.

Parametry

<i>incydent</i>	Obiekt encji Incydent.
-----------------	------------------------

Definicja w linii [67](#) pliku [IncydentDto.java](#).

6.25.3 Dokumentacja funkcji składowych

6.25.3.1 getAutobusId()

```
Long com.mklachl.sopkom.model.dto.IncydentDto.getAutobusId ( )
```

Zwraca identyfikator autobusu związanego z incydentem.

Zwrota

autobusId Identyfikator autobusu.

Definicja w linii [149](#) pliku [IncydentDto.java](#).

6.25.3.2 getDate()

```
Date com.mklachl.sopkom.model.dto.IncydentDto.getDate ( )
```

Zwraca datę incydentu.

Zwrota

date Data incydentu.

Definicja w linii [133](#) pliku [IncydentDto.java](#).

6.25.3.3 getDodatkoweInformacje()

```
String com.mklachl.sopkom.model.dto.IncydentDto.getDodatkoweInformacje ( )
```

Zwraca dodatkowe informacje dotyczące incydentu.

Zwraca

dodatkoweInformacje Dodatkowe informacje dotyczące incydentu.

Definicja w linii 197 pliku [IncydentDto.java](#).

6.25.3.4 getIncydentId()

```
Long com.mklachl.sopkom.model.dto.IncydentDto.getIncydentId ( )
```

Zwraca identyfikator incydentu.

Zwraca

incydentId Identyfikator incydentu.

Definicja w linii 101 pliku [IncydentDto.java](#).

6.25.3.5 getKierowcaId()

```
Long com.mklachl.sopkom.model.dto.IncydentDto.getKierowcaId ( )
```

Zwraca identyfikator kierowcy związanego z incydentem.

Zwraca

kierowcaId Identyfikator kierowcy.

Definicja w linii 165 pliku [IncydentDto.java](#).

6.25.3.6 getKoszty()

```
float com.mklachl.sopkom.model.dto.IncydentDto.getKoszty ( )
```

Zwraca koszty związane z incydentem.

Zwraca

koszty Koszty związane z incydentem.

Definicja w linii 213 pliku [IncydentDto.java](#).

6.25.3.7 getKrotko()

```
String com.mklachl.sopkom.model.dto.IncydentDto.getKrotko ( )
```

Zwraca krótki opis incydentu.

Zwrota

krotko Krótki opis incydentu.

Definicja w linii [85](#) pliku [IncydentDto.java](#).

6.25.3.8 getPrzejazdId()

```
Long com.mklachl.sopkom.model.dto.IncydentDto.getPrzejazdId ( )
```

Zwraca identyfikator przejazdu związanego z incydentem.

Zwrota

przejazdId Identyfikator przejazdu.

Definicja w linii [181](#) pliku [IncydentDto.java](#).

6.25.3.9 getTyp()

```
String com.mklachl.sopkom.model.dto.IncydentDto.getTyp ( )
```

Zwraca typ incydentu.

Zwrota

typ Typ incydentu.

Definicja w linii [117](#) pliku [IncydentDto.java](#).

6.25.3.10 setAutobusId()

```
void com.mklachl.sopkom.model.dto.IncydentDto.setAutobusId ( Long autobusId )
```

Ustawia identyfikator autobusu związanego z incydentem.

Parametry

<i>autobus</i> <i>Id</i>	Identyfikator autobusu.
-----------------------------	-------------------------

Definicja w linii [157](#) pliku [IncydentDto.java](#).

6.25.3.11 setDate()

```
void com.mklachl.sopkom.model.dto.IncydentDto.setDate ( Date date )
```

Ustawia datę incydentu.

Parametry

<i>date</i>	Data incydentu.
-------------	-----------------

Definicja w linii [141](#) pliku [IncydentDto.java](#).

6.25.3.12 setDodatkoweInformacje()

```
void com.mklachl.sopkom.model.dto.IncydentDto.setDodatkoweInformacje ( String dodatkoweInformacje )
```

Ustawia dodatkowe informacje dotyczące incydentu.

Parametry

<i>dodatkoweInformacje</i>	Dodatkowe informacje dotyczące incydentu.
----------------------------	---

Definicja w linii [205](#) pliku [IncydentDto.java](#).

6.25.3.13 setIncydentId()

```
void com.mklachl.sopkom.model.dto.IncydentDto.setIncydentId ( Long incydentId )
```

Ustawia identyfikator incydentu.

Parametry

<i>incydent</i> ↪ <i>Id</i>	Identyfikator incydentu.
--------------------------------	--------------------------

Definicja w linii [109](#) pliku [IncydentDto.java](#).

6.25.3.14 setKierowcaId()

```
void com.mklachl.sopkom.model.dto.IncydentDto.setKierowcaId (
    Long kierowcaId )
```

Ustawia identyfikator kierowcy związanego z incydentem.

Parametry

<i>kierowca</i> ↪ <i>Id</i>	Identyfikator kierowcy.
--------------------------------	-------------------------

Definicja w linii [173](#) pliku [IncydentDto.java](#).

6.25.3.15 setKoszty()

```
void com.mklachl.sopkom.model.dto.IncydentDto.setKoszty (
    float koszty )
```

Ustawia koszty związane z incydentem.

Parametry

<i>koszty</i>	Koszty związane z incydentem.
---------------	-------------------------------

Definicja w linii [221](#) pliku [IncydentDto.java](#).

6.25.3.16 setKrotko()

```
void com.mklachl.sopkom.model.dto.IncydentDto.setKrotko (
    String krotko )
```

Ustawia krótki opis incydentu.

Parametry

<i>krotko</i>	Krótki opis incydentu.
---------------	------------------------

Definicja w linii [93](#) pliku [IncydentDto.java](#).

6.25.3.17 setPrzejazdId()

```
void com.mklachl.sopkom.model.dto.IncydentDto.setPrzejazdId ( Long przejazdId )
```

Ustawia identyfikator przejazdu związanego z incydentem.

Parametry

<i>przejazd</i> <i>Id</i>	Identyfikator przejazdu.
------------------------------	--------------------------

Definicja w linii [189](#) pliku [IncydentDto.java](#).

6.25.3.18 setTyp()

```
void com.mklachl.sopkom.model.dto.IncydentDto.setTyp ( String typ )
```

Ustawia typ incydentu.

Parametry

<i>typ</i>	Typ incydentu.
------------	----------------

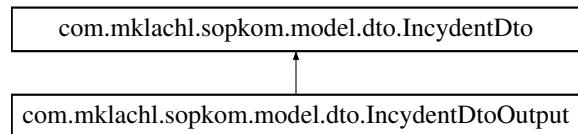
Definicja w linii [125](#) pliku [IncydentDto.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.26 Dokumentacja klasy**com.mklachl.sopkom.model.dto.IncydentDtoOutput**

Diagram dziedziczenia dla com.mklachl.sopkom.model.dto.IncydentDtoOutput



Metody publiczne

- [IncydentDtoOutput \(Incydent incydent\)](#)
- [KierowcaDto getKierowca \(\)](#)
- void [setKierowca \(KierowcaDto kierowca\)](#)
- [AutobusDto getAutobus \(\)](#)
- void [setAutobus \(AutobusDto autobus\)](#)
- [KursDto getKurs \(\)](#)
- void [setKurs \(KursDto kurs\)](#)

6.26.1 Opis szczegółowy

Data Transfer Object (DTO) rozszerzający [IncydentDto](#) o dodatkowe informacje.

Definicja w linii 11 pliku [IncydentDtoOutput.java](#).

6.26.2 Dokumentacja konstruktora i destruktora

6.26.2.1 IncydentDtoOutput()

```
com.mklachl.sopkom.model.dto.IncydentDtoOutput.IncydentDtoOutput (
    Incydent incydent )
```

Konstruktor przyjmujący obiekt encji [Incydent](#).

Parametry

<i>incydent</i>	Obiekt encji Incydent .
-----------------	---

Definicja w linii 38 pliku [IncydentDtoOutput.java](#).

6.26.3 Dokumentacja funkcji składowych

6.26.3.1 getAutobus()

```
AutobusDto com.mklachl.sopkom.model.dto.IncydentDtoOutput.getAutobus ( )
```

Zwraca DTO autobusu związanego z incydentem.

Zwrota

autobus DTO autobusu.

Definicja w linii 77 pliku [IncydentDtoOutput.java](#).

6.26.3.2 getKierowca()

```
KierowcaDto com.mklachl.sopkom.model.dto.IncydentDtoOutput.getKierowca ( )
```

Zwraca DTO kierowcy związanego z incydentem.

Zwrota

kierowca DTO kierowcy.

Definicja w linii 61 pliku [IncydentDtoOutput.java](#).

6.26.3.3 getKurs()

```
KursDto com.mklachl.sopkom.model.dto.IncydentDtoOutput.getKurs ( )
```

Zwraca DTO kursu związanego z incydentem.

Zwrota

kurs DTO kursu.

Definicja w linii 93 pliku [IncydentDtoOutput.java](#).

6.26.3.4 setAutobus()

```
void com.mklachl.sopkom.model.dto.IncydentDtoOutput.setAutobus (   
    AutobusDto autobus )
```

Ustawia DTO autobusu zowanego z incydentem.

Parametry

<i>autobus</i>	DTO autobusu.
----------------	---------------

Definicja w linii 85 pliku [IncydentDtoOutput.java](#).

6.26.3.5 setKierowca()

```
void com.mklachl.sopkom.model.dto.IncydentDtoOutput.setKierowca (
    KierowcaDto kierowca )
```

Ustawia DTO kierowcy związanego z incydentem.

Parametry

<i>kierowca</i>	DTO kierowcy.
-----------------	---------------

Definicja w linii 69 pliku [IncydentDtoOutput.java](#).

6.26.3.6 setKurs()

```
void com.mklachl.sopkom.model.dto.IncydentDtoOutput.setKurs (
    KursDto kurs )
```

Ustawia DTO kursu związanego z incydentem.

Parametry

<i>kurs</i>	DTO kursu.
-------------	------------

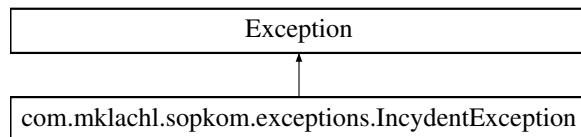
Definicja w linii 101 pliku [IncydentDtoOutput.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.27 Dokumentacja klasy**com.mklachl.sopkom.exceptions.IncydentException**

Diagram dziedziczenia dla com.mklachl.sopkom.exceptions.IncydentException



Metody publiczne

- [IncydentException \(String m\)](#)

6.27.1 Opis szczegółowy

Definicja w linii 3 pliku [IncydentException.java](#).

6.27.2 Dokumentacja konstruktora i destruktora

6.27.2.1 IncydentException()

```
com.mklachl.sopkom.exceptions.IncydentException.IncydentException (
    String m )
```

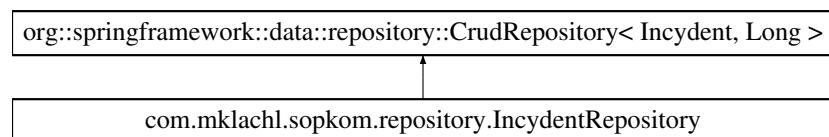
Definicja w linii 5 pliku [IncydentException.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/exc

6.28 Dokumentacja interfejsu com.mklachl.sopkom.repository.IncydentRepository

Diagram dziedziczenia dla com.mklachl.sopkom.repository.IncydentRepository



Metody publiczne

- List< [Incydent](#) > [findAllByTyp](#) (String typ)
- List< [Incydent](#) > [findAllByDateBefore](#) (Date date)
- List< [Incydent](#) > [findAllByDateAfter](#) (Date date)
- List< [Incydent](#) > [findAllByDateBetween](#) (Date date1, Date date2)
- List< [Incydent](#) > [findAllByDate](#) (Date date)
- List< [Incydent](#) > [findAllByAutobus](#) ([Autobus](#) autobus)
- List< [Incydent](#) > [findAllByKierowca](#) ([Kierowca](#) kierowca)
- List< [Incydent](#) > [findAllByPrzejazd](#) ([Przejazd](#) przejazd)
- List< [Incydent](#) > [findAllByDodatkoweInformacje](#) (String dodatkoweInformacje)
- List< [Incydent](#) > [findAllByKosztyLessThan](#) (float koszty)
- List< [Incydent](#) > [findAllByKosztyGreaterThanOrEqual](#) (float koszty)
- List< [Incydent](#) > [findAllByKoszty](#) (float koszty)
- List< [Incydent](#) > [findAll](#) ()

6.28.1 Opis szczegółowy

Interfejs repository dla encji Incydent

Definicja w linii 15 pliku [IncydentRepository.java](#).

6.28.2 Dokumentacja funkcji składowych

6.28.2.1 findAll()

```
List< Incydent > com.mklachl.sopkom.repository.IncydentRepository.findAll ( )
```

Znajduje wszystkie incydenty

Zwrota

Lista incydentów

6.28.2.2 findAllByAutobus()

```
List< Incydent > com.mklachl.sopkom.repository.IncydentRepository.findAllByAutobus (
    Autobus autobus )
```

Znajduje wszystkie incydenty związane z danym autobusem

Parametry

autobus	autobus
-------------------------	---------

Zwroca

Lista incydentów

6.28.2.3 findAllByDate()

```
List< Incident > com.mklachl.sopkom.repository.IncydentRepository.findAllByDate ( Date date )
```

Znajduje wszystkie incydenty w danym dniu

Parametry

<i>date</i>	data
-------------	------

Zwroca

Lista incydentów

6.28.2.4 findAllByDateAfter()

```
List< Incident > com.mklachl.sopkom.repository.IncydentRepository.findAllByDateAfter ( Date date )
```

Znajduje wszystkie incydenty po podanej dacie

Parametry

<i>date</i>	data
-------------	------

Zwroca

Lista incydentów

6.28.2.5 findAllByDateBefore()

```
List< Incident > com.mklachl.sopkom.repository.IncydentRepository.findAllByDateBefore ( Date date )
```

Znajduje wszystkie incydenty przed podaną datą

Parametry

<i>date</i>	data
-------------	------

Zwroca

Lista incydentów

6.28.2.6 findAllByDateBetween()

```
List< Incydent > com.mklachl.sopkom.repository.IncydentRepository.findAllByDateBetween (   
    Date date1,   
    Date date2 )
```

Znajduje wszystkie incydenty pomiędzy dwiema datami

Parametry

<i>date1</i>	data początkowa
<i>date2</i>	data końcowa

Zwroca

Lista incydentów

6.28.2.7 findAllByDodatkoweInformacje()

```
List< Incydent > com.mklachl.sopkom.repository.IncydentRepository.findAllByDodatkoweInformacje (   
     String dodatkoweInformacje )
```

Znajduje wszystkie incydenty z podanymi dodatkowymi informacjami

Parametry

<i>dodatkoweInformacje</i>	dodatkowe informacje
----------------------------	----------------------

Zwroca

Lista incydentów

6.28.2.8 findAllByKierowca()

```
List< Incydent > com.mklachl.sopkom.repository.IncydentRepository.findAllByKierowca (  
    Kierowca kierowca )
```

Znajduje wszystkie incydenty związane z danym kierowcą

Parametry

<i>kierowca</i>	<i>kierowca</i>
-----------------	-----------------

Zwrota

Lista incydentów

6.28.2.9 findAllByKoszty()

```
List< Incydent > com.mklachl.sopkom.repository.IncydentRepository.findAllByKoszty (   
    float koszty )
```

Znajduje wszystkie incydenty z dokładnymi kosztami

Parametry

<i>koszty</i>	<i>koszty</i>
---------------	---------------

Zwrota

Lista incydentów

6.28.2.10 findAllByKosztyGreaterThan()

```
List< Incydent > com.mklachl.sopkom.repository.IncydentRepository.findAllByKosztyGreaterThan (   
    float koszty )
```

Znajduje wszystkie incydenty z kosztami większymi niż podane

Parametry

<i>koszty</i>	<i>koszty</i>
---------------	---------------

Zwrota

Lista incydentów

6.28.2.11 findAllByKosztyLessThan()

```
List< Incydent > com.mklachl.sopkom.repository.IncydentRepository.findAllByKosztyLessThan ( float koszty )
```

Znajduje wszystkie incydenty z kosztami mniejszymi niż podane

Parametry

<i>koszty</i>	<i>koszty</i>
---------------	---------------

Zwroca

Lista incydentów

6.28.2.12 findAllByPrzejazd()

```
List< Incydent > com.mklachl.sopkom.repository.IncydentRepository.findAllByPrzejazd ( Przejazd przejazd )
```

Znajduje wszystkie incydenty związane z danym przejazdem

Parametry

<i>przejazd</i>	<i>przejazd</i>
-----------------	-----------------

Zwroca

Lista incydentów

6.28.2.13 findAllByTyp()

```
List< Incydent > com.mklachl.sopkom.repository.IncydentRepository.findAllByTyp ( String typ )
```

Znajduje wszystkie incydenty o określonym typie

Parametry

<i>typ</i>	<i>typ incydentu</i>
------------	----------------------

Zwraca

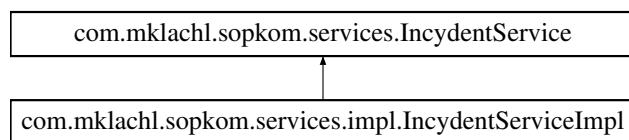
Lista incydentów

Dokumentacja dla tego interfejsu została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/repo

6.29 Dokumentacja interfejsu com.mklachl.sopkom.services.IncydentService

Diagram dziedziczenia dla com.mklachl.sopkom.services.IncydentService



Metody publiczne

- `Incydent saveIncydent (Incydent incydent)`
- `Incydent saveIncydent (IncydentDto incydentDto) throws IncydentException`
- `Incydent updateIncydent (long id, IncydentDto updatedIncydentDto) throws IncydentException`
- `Optional< Incydent > findIncydentById (long id)`
- `List< Incydent > findIncydentByTyp (String typ)`
- `List< Incydent > findIncydentByDateBefore (Date date)`
- `List< Incydent > findIncydentByDateAfter (Date date)`
- `List< Incydent > findIncydentByDateBetween (Date date1, Date date2)`
- `List< Incydent > findIncydentByDate (Date date)`
- `List< Incydent > findIncydentByAutobus (Autobus autobus)`
- `List< Incydent > findIncydentByKierowca (Kierowca kierowca)`
- `List< Incydent > findIncydentByPrzejazd (Przejazd przejazd)`
- `List< Incydent > findIncydentByDodatkoweInformacje (String dodatkoweInformacje)`
- `List< Incydent > findIncydentByKosztyLessThan (float koszty)`
- `List< Incydent > findIncydentByKosztyGreaterThanOrEqual (float koszty)`
- `List< Incydent > findIncydentByKoszty (float koszty)`

6.29.1 Opis szczegółowy

Interfejs serwisu dla obsługi incydentów

Definicja w linii 17 pliku [IncydentService.java](#).

6.29.2 Dokumentacja funkcji składowych

6.29.2.1 findIncydentByAutobus()

```
List< Incydent > com.mklachl.sopkom.services.IncydentService.findIncydentByAutobus ( Autobus autobus )
```

Znajduje incydenty dla podanego autobusu

Parametry

<i>autobus</i>	autobus
----------------	---------

Zwroca

lista incydentów

Implementowany w [com.mklachl.sopkom.services.impl.IncydentServiceImpl](#).

6.29.2.2 findIncydentByDate()

```
List< Incident > com.mklachl.sopkom.services.IncydentService.findIncydentByDate ( Date date )
```

Znajduje incydenty po dokładnej dacie

Parametry

<i>date</i>	data
-------------	------

Zwroca

lista incydentów

Implementowany w [com.mklachl.sopkom.services.impl.IncydentServiceImpl](#).

6.29.2.3 findIncydentByDateAfter()

```
List< Incident > com.mklachl.sopkom.services.IncydentService.findIncydentByDateAfter ( Date date )
```

Znajduje incydenty po podanej dacie

Parametry

<i>date</i>	data
-------------	------

Zwroca

lista incydentów

Implementowany w [com.mklachl.sopkom.services.impl.IncydentServiceImpl](#).

6.29.2.4 findIncydentByDateBefore()

```
List< Incydent > com.mklachl.sopkom.services.IncydentService.findIncydentByDateBefore ( Date date )
```

Znajduje incydenty przed podaną datą

Parametry

<i>date</i>	data
-------------	------

Zwroca

lista incydentów

Implementowany w [com.mklachl.sopkom.services.impl.IncydentServiceImpl](#).

6.29.2.5 findIncydentByDateBetween()

```
List< Incydent > com.mklachl.sopkom.services.IncydentService.findIncydentByDateBetween ( Date date1, Date date2 )
```

Znajduje incydenty w przedziale między dwiema datami

Parametry

<i>date1</i>	pierwsza data
<i>date2</i>	druga data

Zwroca

lista incydentów

Implementowany w [com.mklachl.sopkom.services.impl.IncydentServiceImpl](#).

6.29.2.6 findIncydentByDodatkoweInformacje()

```
List< Incydent > com.mklachl.sopkom.services.IncydentService.findIncydentByDodatkoweInformacje ( String dodatkoweInformacje )
```

Znajduje incydenty po dodatkowych informacjach

Parametry

<i>dodatkoweInformacje</i>	dodatkowe informacje
----------------------------	----------------------

Zwroca

lista incydentów

Implementowany w [com.mklachl.sopkom.services.impl.IncydentServiceImpl](#).

6.29.2.7 findIncydentById()

```
Optional< Incydent > com.mklachl.sopkom.services.IncydentService.findIncydentById ( long id )
```

Znajduje incydent po ID

Parametry

<i>id</i>	ID incydentu
-----------	--------------

Zwroca

Optional z incydentem

Implementowany w [com.mklachl.sopkom.services.impl.IncydentServiceImpl](#).

6.29.2.8 findIncydentByKierowca()

```
List< Incydent > com.mklachl.sopkom.services.IncydentService.findIncydentByKierowca ( Kierowca kierowca )
```

Znajduje incydenty dla podanego kierowcy

Parametry

<i>kierowca</i>	kierowca
-----------------	----------

Zwroca

lista incydentów

Implementowany w [com.mklachl.sopkom.services.impl.IncydentServiceImpl](#).

6.29.2.9 findIncydentByKoszty()

```
List< Incydent > com.mklachl.sopkom.services.IncydentService.findIncydentByKoszty ( float koszty )
```

Znajduje incydenty po dokładnych kosztach

Parametry

<i>koszty</i>	<i>koszty</i>
---------------	---------------

Zwroca

lista incydentów

Implementowany w [com.mklachl.sopkom.services.impl.IncydentServiceImpl](#).

6.29.2.10 findIncydentByKosztyGreaterThan()

```
List< Incydent > com.mklachl.sopkom.services.IncydentService.findIncydentByKosztyGreaterThan ( float koszty )
```

Znajduje incydenty, których koszty są większe niż podane

Parametry

<i>koszty</i>	<i>koszty</i>
---------------	---------------

Zwroca

lista incydentów

Implementowany w [com.mklachl.sopkom.services.impl.IncydentServiceImpl](#).

6.29.2.11 findIncydentByKosztyLessThan()

```
List< Incydent > com.mklachl.sopkom.services.IncydentService.findIncydentByKosztyLessThan ( float koszty )
```

Znajduje incydenty, których koszty są mniejsze niż podane

Parametry

<i>koszty</i>	<i>koszty</i>
---------------	---------------

Zwrota

lista incydentów

Implementowany w [com.mklachl.sopkom.services.impl.IncydentServiceImpl](#).

6.29.2.12 findIncydentByPrzejazd()

```
List< Incydent > com.mklachl.sopkom.services.IncydentService.findIncydentByPrzejazd ( Przejazd przejazd )
```

Znajduje incydenty dla podanego przejazdu

Parametry

przejazd	przejazd
----------	----------

Zwrota

lista incydentów

Implementowany w [com.mklachl.sopkom.services.impl.IncydentServiceImpl](#).

6.29.2.13 findIncydentByTyp()

```
List< Incydent > com.mklachl.sopkom.services.IncydentService.findIncydentByTyp ( String typ )
```

Znajduje incydenty po typie

Parametry

typ	typ
-----	-----

Zwrota

lista incydentów

Implementowany w [com.mklachl.sopkom.services.impl.IncydentServiceImpl](#).

6.29.2.14 saveIncydent() [1/2]

```
Incydent com.mklachl.sopkom.services.IncydentService.saveIncydent ( Incydent incydent )
```

Zapisuje nowy incydent

Parametry

<i>incydent</i>	encja incydent
-----------------	----------------

Zwroca

zapisany incydent

Implementowany w [com.mklachl.sopkom.services.impl.IncydentServiceImpl](#).

6.29.2.15 saveIncydent() [2/2]

```
Incydent com.mklachl.sopkom.services.IncydentService.saveIncydent (
    IncydentDto incydentDto ) throws IncydentException
```

Zapisuje nowy incydent na podstawie DTO

Parametry

<i>incydentDto</i>	DTO incydentu
--------------------	---------------

Zwroca

zapisany incydent

Wyjątki

<i>IncydentException</i>	wyjątek incydentu
--------------------------	-------------------

Implementowany w [com.mklachl.sopkom.services.impl.IncydentServiceImpl](#).

6.29.2.16 updateIncydent()

```
Incydent com.mklachl.sopkom.services.IncydentService.updateIncydent (
    long id,
    IncydentDto updatedIncydentDto ) throws IncydentException
```

Aktualizuje istniejący incydent na podstawie ID

Parametry

<i>id</i>	ID incydentu
<i>updatedIncydentDto</i>	zaktualizowany DTO incydentu

Zwraca

zaktualizowany incydent

Wyjątki

<i>IncydentException</i>	wyjątek incydentu
--------------------------	-------------------

Implementowany w [com.mklachl.sopkom.services.impl.IncydentServiceImpl](#).

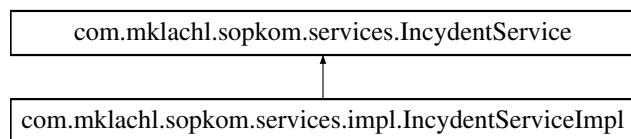
Dokumentacja dla tego interfejsu została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/ser...

6.30 Dokumentacja klasy

com.mklachl.sopkom.services.impl.IncydentServiceImpl

Diagram dziedziczenia dla com.mklachl.sopkom.services.impl.IncydentServiceImpl



Metody publiczne

- `IncydentServiceImpl (IncydentRepository incydentRepository, AutobusRepository autobusRepository, KierowcaRepository kierowcaRepository, PrzejazdRepository przejazdRepository)`
- `Incydent saveIncydent (Incydent incydentDto)`
- `Incydent saveIncydent (IncydentDto incydentDto) throws IncydentException`
- `Incydent updateIncydent (long id, IncydentDto updatedIncydentDto) throws IncydentException`
- `Optional< Incydent > findIncydentById (long id)`
- `List< Incydent > findIncydentByTyp (String typ)`
- `List< Incydent > findIncydentByDateBefore (Date date)`
- `List< Incydent > findIncydentByDateAfter (Date date)`
- `List< Incydent > findIncydentByDateBetween (Date date1, Date date2)`
- `List< Incydent > findIncydentByDate (Date date)`
- `List< Incydent > findIncydentByAutobus (Autobus autobus)`
- `List< Incydent > findIncydentByKierowca (Kierowca kierowca)`
- `List< Incydent > findIncydentByPrzejazd (Przejazd przejazd)`
- `List< Incydent > findIncydentByDodatkoweInformacje (String dodatkoweInformacje)`
- `List< Incydent > findIncydentByKosztyLessThan (float koszty)`
- `List< Incydent > findIncydentByKosztyGreaterThanOrEqual (float koszty)`
- `List< Incydent > findIncydentByKoszty (float koszty)`

6.30.1 Opis szczegółowy

Definicja w linii 25 pliku [IncydentServiceImpl.java](#).

6.30.2 Dokumentacja konstruktora i destruktora

6.30.2.1 IncydentServiceImpl()

```
com.mklachl.sopkom.services.impl.IncydentServiceImpl.IncydentServiceImpl (  
    IncydentRepository incydentRepository,  
    AutobusRepository autobusRepository,  
    KierowcaRepository kierowcaRepository,  
    PrzejazdRepository przejazdRepository )
```

Definicja w linii 33 pliku [IncydentServiceImpl.java](#).

6.30.3 Dokumentacja funkcji składowych

6.30.3.1 findIncydentByAutobus()

```
List< Incydent > com.mklachl.sopkom.services.impl.IncydentServiceImpl.findIncydentByAutobus (  
    Autobus autobus )
```

Znajduje incydenty dla podanego autobusu

Parametry

autobus	autobus
---------	---------

Zwrota

lista incydentów

Implementuje [com.mklachl.sopkom.services.IncydentService](#).

Definicja w linii 163 pliku [IncydentServiceImpl.java](#).

6.30.3.2 findIncydentByDate()

```
List< Incydent > com.mklachl.sopkom.services.impl.IncydentServiceImpl.findIncydentByDate (  
    Date date )
```

Znajduje incydenty po dokładnej dacie

Parametry

<i>date</i>	data
-------------	------

Zwroca

lista incydentów

Implementuje [com.mklachl.sopkom.services.IncydentService](#).

Definicja w linii [158](#) pliku [IncydentServiceImpl.java](#).

6.30.3.3 findIncydentByDateAfter()

```
List< Incydent > com.mklachl.sopkom.services.impl.IncydentServiceImpl.findIncydentByDateAfter  
(  
    Date date )
```

Znajduje incydenty po podanej dacie

Parametry

<i>date</i>	data
-------------	------

Zwroca

lista incydentów

Implementuje [com.mklachl.sopkom.services.IncydentService](#).

Definicja w linii [148](#) pliku [IncydentServiceImpl.java](#).

6.30.3.4 findIncydentByDateBefore()

```
List< Incydent > com.mklachl.sopkom.services.impl.IncydentServiceImpl.findIncydentByDateBefore  
(  
    Date date )
```

Znajduje incydenty przed podaną datą

Parametry

<i>date</i>	data
-------------	------

Zwroca

listę incydentów

Implementuje [com.mklachl.sopkom.services.IncydentService](#).

Definicja w linii 143 pliku [IncydentServiceImpl.java](#).

6.30.3.5 findIncydentByDateBetween()

```
List< Incydent > com.mklachl.sopkom.services.impl.IncydentServiceImpl.findIncydentByDateBetween ( Date date1, Date date2 )
```

Znajduje incydenty w przedziale między dwiema datami

Parametry

<i>date1</i>	pierwsza data
<i>date2</i>	druga data

Zwroca

listę incydentów

Implementuje [com.mklachl.sopkom.services.IncydentService](#).

Definicja w linii 153 pliku [IncydentServiceImpl.java](#).

6.30.3.6 findIncydentByDodatkoweInformacje()

```
List< Incydent > com.mklachl.sopkom.services.impl.IncydentServiceImpl.findIncydentByDodatkoweInformacje ( String dodatkoweInformacje )
```

Znajduje incydenty po dodatkowych informacjach

Parametry

<i>dodatkoweInformacje</i>	dodatkowe informacje
----------------------------	----------------------

Zwroca

listę incydentów

Implementuje [com.mklachl.sopkom.services.IncydentService](#).

Definicja w linii [178](#) pliku [IncydentServiceImpl.java](#).

6.30.3.7 `findIncydentById()`

```
Optional< Incydent > com.mklachl.sopkom.services.impl.IncydentServiceImpl.findIncydentById ( long id )
```

Znajduje incydent po ID

Parametry

<i>id</i>	ID incydentu
-----------	--------------

Zwroca

Optional z incydentem

Implementuje [com.mklachl.sopkom.services.IncydentService](#).

Definicja w linii [133](#) pliku [IncydentServiceImpl.java](#).

6.30.3.8 `findIncydentByKierowca()`

```
List< Incydent > com.mklachl.sopkom.services.impl.IncydentServiceImpl.findIncydentByKierowca ( Kierowca kierowca )
```

Znajduje incydenty dla podanego kierowcy

Parametry

<i>kierowca</i>	kierowca
-----------------	----------

Zwroca

lista incydentów

Implementuje [com.mklachl.sopkom.services.IncydentService](#).

Definicja w linii [168](#) pliku [IncydentServiceImpl.java](#).

6.30.3.9 findIncydentByKoszty()

```
List< Incydent > com.mklachl.sopkom.services.impl.IncydentServiceImpl.findIncydentByKoszty ( float koszty )
```

Znajduje incydenty po dokładnych kosztach

Parametry

<i>koszty</i>	<i>koszty</i>
---------------	---------------

Zwroca

lista incydentów

Implementuje [com.mklachl.sopkom.services.IncydentService](#).

Definicja w linii 193 pliku [IncydentServiceImpl.java](#).

6.30.3.10 findIncydentByKosztyGreaterThan()

```
List< Incydent > com.mklachl.sopkom.services.impl.IncydentServiceImpl.findIncydentByKosztyGreaterThan ( float koszty )
```

Znajduje incydenty, których koszty są większe niż podane

Parametry

<i>koszty</i>	<i>koszty</i>
---------------	---------------

Zwroca

lista incydentów

Implementuje [com.mklachl.sopkom.services.IncydentService](#).

Definicja w linii 188 pliku [IncydentServiceImpl.java](#).

6.30.3.11 findIncydentByKosztyLessThan()

```
List< Incydent > com.mklachl.sopkom.services.impl.IncydentServiceImpl.findIncydentByKosztyLessThan ( float koszty )
```

Znajduje incydenty, których koszty są mniejsze niż podane

Parametry

<i>koszty</i>	<i>koszty</i>
---------------	---------------

Zwroca

lista incydentów

Implementuje [com.mklachl.sopkom.services.IncydentService](#).

Definicja w linii [183](#) pliku [IncydentServiceImpl.java](#).

6.30.3.12 findIncydentByPrzejazd()

```
List< Incydent > com.mklachl.sopkom.services.impl.IncydentServiceImpl.findIncydentByPrzejazd ( Przejazd przejazd )
```

Znajduje incydenty dla podanego przejazdu

Parametry

<i>przejazd</i>	<i>przejazd</i>
-----------------	-----------------

Zwroca

lista incydentów

Implementuje [com.mklachl.sopkom.services.IncydentService](#).

Definicja w linii [173](#) pliku [IncydentServiceImpl.java](#).

6.30.3.13 findIncydentByTyp()

```
List< Incydent > com.mklachl.sopkom.services.impl.IncydentServiceImpl.findIncydentByTyp ( String typ )
```

Znajduje incydenty po typie

Parametry

<i>typ</i>	<i>typ</i>
------------	------------

Zwroca

listę incydentów

Implementuje [com.mklachl.sopkom.services.IncydentService](#).

Definicja w linii [138](#) pliku [IncydentServiceImpl.java](#).

6.30.3.14 saveIncydent() [1/2]

```
Incydent com.mklachl.sopkom.services.impl.IncydentServiceImpl.saveIncydent (
    Incydent incydent )
```

Zapisuje nowy incydent

Parametry

<i>incydent</i>	encja incydent
-----------------	----------------

Zwroca

zapisany incydent

Implementuje [com.mklachl.sopkom.services.IncydentService](#).

Definicja w linii [42](#) pliku [IncydentServiceImpl.java](#).

6.30.3.15 saveIncydent() [2/2]

```
Incydent com.mklachl.sopkom.services.impl.IncydentServiceImpl.saveIncydent (
    IncydentDto incydentDto ) throws IncydentException
```

Zapisuje nowy incydent na podstawie DTO

Parametry

<i>incydentDto</i>	DTO incydentu
--------------------	---------------

Zwroca

zapisany incydent

Wyjątki

<i>IncydentException</i>	wyjątek incydentu
--------------------------	-------------------

Implementuje [com.mklachl.sopkom.services.IncydentService](#).

Definicja w linii [57](#) pliku [IncydentServiceImpl.java](#).

6.30.3.16 updateIncydent()

```
Incydent com.mklachl.sopkom.services.impl.IncydentServiceImpl.updateIncydent (
    long id,
    IncydentDto updatedIncydentDto ) throws IncydentException
```

Aktualizuje istniejący incydent na podstawie ID

Parametry

<i>id</i>	ID incydentu
<i>updatedIncydentDto</i>	zaktualizowany DTO incydentu

Zwraca

zaktualizowany incydent

Wyjątki

<i>IncydentException</i>	wyjątek incydentu
--------------------------	-------------------

Implementuje [com.mklachl.sopkom.services.IncydentService](#).

Definicja w linii [89](#) pliku [IncydentServiceImpl.java](#).

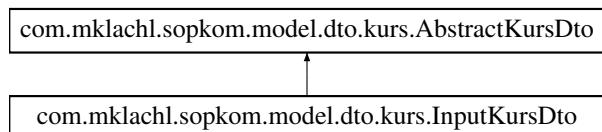
Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/ser...

6.31 Dokumentacja klasy

com.mklachl.sopkom.model.dto.kurs.InputKursDto

Diagram dziedziczenia dla **com.mklachl.sopkom.model.dto.kurs.InputKursDto**



Metody publiczne

- `InputKursDto ()`
- `List< InputPrzystanekWKursieDto > getPrzystanki ()`
- `void setPrzystanki (List< InputPrzystanekWKursieDto > przystanki)`

Atrybuty publiczne

- `List< InputPrzystanekWKursieDto > przystanki`

Dodatkowe Dziedziczone Składowe

6.31.1 Opis szczegółowy

Data Transfer Object (DTO) wejściowy dla encji Kurs.

Definicja w linii 8 pliku [InputKursDto.java](#).

6.31.2 Dokumentacja konstruktora i destruktora

6.31.2.1 InputKursDto()

```
com.mklachl.sopkom.model.dto.kurs.InputKursDto.InputKursDto ( )
```

Konstruktor domyślny.

Definicja w linii 18 pliku [InputKursDto.java](#).

6.31.3 Dokumentacja funkcji składowych

6.31.3.1 getPrzystanki()

```
List< InputPrzystanekWKursieDto > com.mklachl.sopkom.model.dto.kurs.InputKursDto.getPrzystanki  
( )
```

Zwraca listę przystanków w kursie.

Zwrota

przystanki Lista przystanków w kursie.

Definicja w linii 27 pliku [InputKursDto.java](#).

6.31.3.2 setPrzystanki()

```
void com.mklachl.sopkom.model.dto.kurs.InputKursDto.setPrzystanki  
(  
    List< InputPrzystanekWKursieDto > przystanki )
```

Ustawia listę przystanków w kursie.

Parametry

<code>przystanki</code>	Lista przystanków w kursie.
-------------------------	-----------------------------

Definicja w linii [35](#) pliku [InputKursDto.java](#).

6.31.4 Dokumentacja atrybutów składowych

6.31.4.1 przystanki

`List<InputPrzystanekWKursieDto> com.mklachl.sopkom.model.dto.kurs.InputKursDto.przystanki`

Lista przystanków w kursie.

Definicja w linii [13](#) pliku [InputKursDto.java](#).

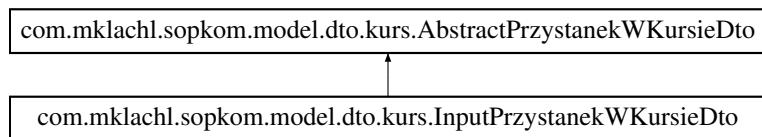
Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.32 Dokumentacja klasy

com.mklachl.sopkom.model.dto.kurs.InputPrzystanekWKursieDto

Diagram dziedziczenia dla com.mklachl.sopkom.model.dto.kurs.InputPrzystanekWKursieDto



Metody publiczne

- `InputPrzystanekWKursieDto ()`
- `InputPrzystanekWKursieDto (KursPrzystanekWlini kpw)`
- `PrzystanekDtoDlaLinia getPrzystanekWLini ()`
- `void setPrzystanekWLini (PrzystanekDtoDlaLinia przystanekWLini)`

Dodatkowe Dziedziczone Składowe

6.32.1 Opis szczegółowy

Data Transfer Object (DTO) wejściowy dla encji KursPrzystanekWlini.

Definicja w linii [9](#) pliku [InputPrzystanekWKursieDto.java](#).

6.32.2 Dokumentacja konstruktora i destruktora

6.32.2.1 InputPrzystanekWKursieDto() [1/2]

```
com.mklachl.sopkom.model.dto.kurs.InputPrzystanekWKursieDto.InputPrzystanekWKursieDto ( )
```

Konstruktor domyślny.

Definicja w linii 19 pliku [InputPrzystanekWKursieDto.java](#).

6.32.2.2 InputPrzystanekWKursieDto() [2/2]

```
com.mklachl.sopkom.model.dto.kurs.InputPrzystanekWKursieDto.InputPrzystanekWKursieDto ( KursPrzystanekWLini kpw )
```

Konstruktor przyjmujący obiekt encji KursPrzystanekWLini.

Parametry

<i>kpw</i>	Obiekt encji KursPrzystanekWLini.
------------	-----------------------------------

Definicja w linii 26 pliku [InputPrzystanekWKursieDto.java](#).

6.32.3 Dokumentacja funkcji składowych

6.32.3.1 getPrzystanekWLini()

```
PrzystanekDtoDlaLinia com.mklachl.sopkom.model.dto.kurs.InputPrzystanekWKursieDto.getPrzystanekWLini ( )
```

Zwraca DTO przystanku w linii.

Zwrota

przystanekWLini DTO przystanku w linii.

Definicja w linii 37 pliku [InputPrzystanekWKursieDto.java](#).

6.32.3.2 setPrzystanekWLini()

```
void com.mklachl.sopkom.model.dto.kurs.InputPrzystanekWKursieDto.setPrzystanekWLini ( PrzystanekDtoDlaLinia przystanekWLini )
```

Ustawia DTO przystanku w linii.

Parametry

<code>przystanekWLini</code>	DTO przystanku w linii.
------------------------------	-------------------------

Definicja w linii 45 pliku [InputPrzystanekWKursieDto.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.33 Dokumentacja klasy com.mklachl.sopkom.model.entity.Kierowca

Metody publiczne

- Long `getKierowcald` ()
- void `setKierowcald` (Long kierowcald)
- Date `getPrawoJazdyWazneDo` ()
- void `setPrawoJazdyWazneDo` (Date prawoJazdyWazneDo)
- String `getImie` ()
- void `setImie` (String imie)
- String `getNazwisko` ()
- void `setNazwisko` (String nazwisko)
- String `getPesel` ()
- void `setPesel` (String pesel)
- User `getUser` ()
- void `setUser` (User user)
- int `hashCode` ()
- boolean `equals` (Object obj)

6.33.1 Opis szczegółowy

Klasa reprezentująca encję [Kierowca](#).

Definicja w linii 21 pliku [Kierowca.java](#).

6.33.2 Dokumentacja funkcji składowych

6.33.2.1 equals()

```
boolean com.mklachl.sopkom.model.entity.Kierowca.equals (
    Object obj )
```

Porównuje bieżący obiekt [Kierowca](#) z innym obiektem.

Parametry

<i>obj</i>	Obiekt do porównania.
------------	-----------------------

Zwrota

boolean Wartość true, jeśli obiekty są równe, w przeciwnym razie false.

Definicja w linii [174](#) pliku [Kierowca.java](#).

6.33.2.2 getImie()

```
String com.mklachl.sopkom.model.entity.Kierowca.getImie ( )
```

Zwrota imię kierowcy.

Zwrota

imie Imię kierowcy.

Definicja w linii [99](#) pliku [Kierowca.java](#).

6.33.2.3 getKierowcaId()

```
Long com.mklachl.sopkom.model.entity.Kierowca.getKierowcaId ( )
```

Zwrota identyfikator kierowcy.

Zwrota

kierowcaId Identyfikator kierowcy.

Definicja w linii [67](#) pliku [Kierowca.java](#).

6.33.2.4 getNazwisko()

```
String com.mklachl.sopkom.model.entity.Kierowca.getNazwisko ( )
```

Zwrota nazwisko kierowcy.

Zwrota

nazwisko Nazwisko kierowcy.

Definicja w linii [115](#) pliku [Kierowca.java](#).

6.33.2.5 `getPesel()`

```
String com.mklachl.sopkom.model.entity.Kierowca.getPesel ( )
```

Zwraca PESEL kierowcy.

Zwraca

pesel PESEL kierowcy.

Definicja w linii 131 pliku [Kierowca.java](#).

6.33.2.6 `getPrawoJazdyWazneDo()`

```
Date com.mklachl.sopkom.model.entity.Kierowca.getPrawoJazdyWazneDo ( )
```

Zwraca datę ważności prawa jazdy kierowcy.

Zwraca

prawoJazdyWazneDo Data ważności prawa jazdy kierowcy.

Definicja w linii 83 pliku [Kierowca.java](#).

6.33.2.7 `getUser()`

```
User com.mklachl.sopkom.model.entity.Kierowca.getUser ( )
```

Zwraca użytkownika związanego z kierowcą.

Zwraca

user Użytkownik związany z kierowcą.

Definicja w linii 147 pliku [Kierowca.java](#).

6.33.2.8 `hashCode()`

```
int com.mklachl.sopkom.model.entity.Kierowca.hashCode ( )
```

Generuje hash code dla obiektu [Kierowca](#).

Zwraca

hashCode Hash code obiektu [Kierowca](#).

Definicja w linii 164 pliku [Kierowca.java](#).

6.33.2.9 `setImie()`

```
void com.mklachl.sopkom.model.entity.Kierowca.setImie ( String imie )
```

Ustawia imię kierowcy.

Parametry

<i>imie</i>	Imię kierowcy.
-------------	----------------

Definicja w linii [107](#) pliku [Kierowca.java](#).

6.33.2.10 setKierowcaId()

```
void com.mklachl.sopkom.model.entity.Kierowca.setKierowcaId (  
    Long kierowcaId )
```

Ustawia identyfikator kierowcy.

Parametry

<i>kierowca</i> ← <i>Id</i>	Identyfikator kierowcy.
--------------------------------	-------------------------

Definicja w linii [75](#) pliku [Kierowca.java](#).

6.33.2.11 setNazwisko()

```
void com.mklachl.sopkom.model.entity.Kierowca.setNazwisko (   
    String nazwisko )
```

Ustawia nazwisko kierowcy.

Parametry

<i>nazwisko</i>	Nazwisko kierowcy.
-----------------	--------------------

Definicja w linii [123](#) pliku [Kierowca.java](#).

6.33.2.12 setPesel()

```
void com.mklachl.sopkom.model.entity.Kierowca.setPesel (   
    String pesel )
```

Ustawia PESEL kierowcy.

Parametry

<i>pesel</i>	PESEL kierowcy.
--------------	-----------------

Definicja w linii [139](#) pliku [Kierowca.java](#).

6.33.2.13 setPrawoJazdyWazneDo()

```
void com.mklachl.sopkom.model.entity.Kierowca.setPrawoJazdyWazneDo ( Date prawoJazdyWazneDo )
```

Ustawia datę ważności prawa jazdy kierowcy.

Parametry

<i>prawoJazdyWazneDo</i>	Data ważności prawa jazdy kierowcy.
--------------------------	-------------------------------------

Definicja w linii [91](#) pliku [Kierowca.java](#).

6.33.2.14 setUser()

```
void com.mklachl.sopkom.model.entity.Kierowca.setUser ( User user )
```

Ustawia użytkownika związanego z kierowcą.

Parametry

<i>user</i>	Użytkownik związany z kierowcą.
-------------	---------------------------------

Definicja w linii [155](#) pliku [Kierowca.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.34 Dokumentacja klasy **com.mklachl.sopkom.controller.KierowcaController**

Metody publiczne

- [KierowcaController](#) ([KierowcaRepository](#) kierowcaRepository, [UserService](#) userService)

- ResponseEntity<?> `template ()`
- ResponseEntity< `KierowcaDto` > `saveKierowca (@RequestBody KierowcaDto kierowca)`
- ResponseEntity< `KierowcaDto` > `findKierowcaById (@PathVariable(name="id") Long id)`
- ResponseEntity<?> `deleteKierowca (@PathVariable("id") Long id)`
- ResponseEntity<?> `updateKierowca (@PathVariable("id") Long id, @RequestBody KierowcaDto kierowcaDto)`
- ResponseEntity< List< `KierowcaDto` > > `findKierowcaByPesel (@RequestParam(name="pesel", required=false) String pesel)`
- ResponseEntity< `KierowcaDto` > `findKierowcaByUser ()`
- ResponseEntity< List< `KierowcaDto` > > `findKierowcaByImieAndNazwisko (@RequestParam(name="imie") String imie, @RequestParam(name="nazwisko") String nazwisko)`
- ResponseEntity< List< `KierowcaDto` > > `findKierowcaByNazwisko (@RequestParam(name="nazwisko") String nazwisko)`
- ResponseEntity< List< `KierowcaDto` > > `findKierowcaByPrawoJazdyWazneDoBefore (@RequestParam(name="date") Date date)`
- ResponseEntity< List< `KierowcaDto` > > `findKierowcaByPrawoJazdyWazneDoAfter (@RequestParam(name="date") Date date)`
- ResponseEntity< List< `KierowcaDto` > > `findAll ()`

6.34.1 Opis szczegółowy

Kontroler dla zasobu Kierowca. Obsługuje operacje tworzenia, pobierania, aktualizacji i usuwania kierowców.

Definicja w linii 29 pliku `KierowcaController.java`.

6.34.2 Dokumentacja konstruktora i destruktora

6.34.2.1 KierowcaController()

```
com.mklachl.sopkom.controller.KierowcaController.KierowcaController (
    KierowcaRepository kierowcaRepository,
    UserService userService )
```

Konstruktor kontrolera.

Parametry

<code>kierowcaRepository</code>	Repozytorium kierowców.
<code>userService</code>	Serwis użytkowników.

Definicja w linii 41 pliku `KierowcaController.java`.

6.34.3 Dokumentacja funkcji składowych

6.34.3.1 deleteKierowca()

```
ResponseEntity<?> com.mklachl.sopkom.controller.KierowcaController.deleteKierowca ( @PathVariable("id") Long id )
```

Usuwa kierowcę po identyfikatorze.

Parametry

<i>id</i>	Identyfikator kierowcy.
-----------	-------------------------

Zwrota

Status operacji.

Definicja w linii [93](#) pliku [KierowcaController.java](#).

6.34.3.2 findAll()

```
ResponseEntity< List< KierowcaDto > > com.mklachl.sopkom.controller.KierowcaController.findAll ( )
```

Pobiera wszystkich kierowców.

Zwrota

Lista obiektów DTO kierowców.

Definicja w linii [230](#) pliku [KierowcaController.java](#).

6.34.3.3 findKierowcaById()

```
ResponseEntity< KierowcaDto > com.mklachl.sopkom.controller.KierowcaController.findKierowcaById ( @PathVariable(name="id") Long id )
```

Pobiera kierowcę po identyfikatorze.

Parametry

<i>id</i>	Identyfikator kierowcy.
-----------	-------------------------

Zwrota

Obiekt DTO kierowcy lub wiadomość o błędzie.

Definicja w linii 81 pliku [KierowcaController.java](#).

6.34.3.4 findKierowcaByImieAndNazwisko()

```
ResponseEntity< List< KierowcaDto > > com.mklachl.sopkom.controller.KierowcaController.find←
KierowcaByImieAndNazwisko (
    @RequestParam(name="imie") String imie,
    @RequestParam(name="nazwisko") String nazwisko )
```

Pobiera kierowców po imieniu i nazwisku.

Parametry

<i>imie</i>	Imię kierowcy.
<i>nazwisko</i>	Nazwisko kierowcy.

Zwraca

Lista obiektów DTO kierowców.

Definicja w linii 175 pliku [KierowcaController.java](#).

6.34.3.5 findKierowcaByNazwisko()

```
ResponseEntity< List< KierowcaDto > > com.mklachl.sopkom.controller.KierowcaController.find←
KierowcaByNazwisko (
    @RequestParam(name="nazwisko") String nazwisko )
```

Pobiera kierowców po nazwisku.

Parametry

<i>nazwisko</i>	Nazwisko kierowcy.
-----------------	--------------------

Zwraca

Lista obiektów DTO kierowców.

Definicja w linii 189 pliku [KierowcaController.java](#).

6.34.3.6 findKierowcaByPesel()

```
ResponseEntity< List< KierowcaDto > > com.mklachl.sopkom.controller.KierowcaController.find←
KierowcaByPesel (
    @RequestParam(name="pesel", required=false) String pesel )
```

Pobiera kierowców po numerze PESEL.

Parametry

<i>pesel</i>	Numer PESEL (opcjonalny).
--------------	---------------------------

Zwrota

Lista obiektów DTO kierowców.

Definicja w linii 135 pliku [KierowcaController.java](#).

6.34.3.7 findKierowcaByPrawoJazdyWazneDoAfter()

```
ResponseEntity< List< KierowcaDto > > com.mklachl.sopkom.controller.KierowcaController.find←
KierowcaByPrawoJazdyWazneDoAfter (
    @RequestParam(name="date") Date date )
```

Pobiera kierowców z prawem jazdy ważnym po określonej dacie.

Parametry

<i>date</i>	Data ważności prawa jazdy.
-------------	----------------------------

Zwrota

Lista obiektów DTO kierowców.

Definicja w linii 217 pliku [KierowcaController.java](#).

6.34.3.8 findKierowcaByPrawoJazdyWazneDoBefore()

```
ResponseEntity< List< KierowcaDto > > com.mklachl.sopkom.controller.KierowcaController.find←
KierowcaByPrawoJazdyWazneDoBefore (
    @RequestParam(name="date") Date date )
```

Pobiera kierowców z prawem jazdy ważnym do określonej daty.

Parametry

<i>date</i>	Data ważności prawa jazdy.
-------------	----------------------------

Zwrota

Lista obiektów DTO kierowców.

Definicja w linii [203](#) pliku [KierowcaController.java](#).

6.34.3.9 findKierowcaByUser()

```
ResponseEntity< KierowcaDto > com.mklachl.sopkom.controller.KierowcaController.findKierowca←
ByUser ( )
```

Pobiera kierowcę powiązanego z zalogowanym użytkownikiem.

Zwrota

Obiekt DTO kierowcy lub wiadomość o błędzie.

Definicja w linii [158](#) pliku [KierowcaController.java](#).

6.34.3.10 saveKierowca()

```
ResponseEntity< KierowcaDto > com.mklachl.sopkom.controller.KierowcaController.saveKierowca (
    @RequestBody KierowcaDto kierowca )
```

Tworzy nowego kierowcę.

Parametry

<i>kierowca</i>	Obiekt DTO kierowcy
-----------------	---------------------

Zwrota

Utworzony obiekt DTO kierowcy.

Definicja w linii [68](#) pliku [KierowcaController.java](#).

6.34.3.11 template()

```
ResponseEntity<?> com.mklachl.sopkom.controller.KierowcaController.template ( )
```

Szablon obiektu KierowcaDto.

Zwroca

Obiekt KierowcaDto.

Definicja w linii [52](#) pliku [KierowcaController.java](#).

6.34.3.12 updateKierowca()

```
ResponseEntity<?> com.mklachl.sopkom.controller.KierowcaController.updateKierowca ( @PathVariable("id") Long id, @RequestBody KierowcaDto kierowcaDto )
```

Aktualizuje kierowcę po identyfikatorze.

Parametry

<i>id</i>	Identyfikator kierowcy.
<i>kierowcaDto</i>	Obiekt DTO kierowcy.

Zwroca

Status operacji.

Definicja w linii [108](#) pliku [KierowcaController.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/controller/KierowcaController.java

6.35 Dokumentacja klasy com.mklachl.sopkom.model.dto.KierowcaDto

Metody publiczne

- [KierowcaDto \(\)](#)
- [KierowcaDto \(Kierowca kierowca\)](#)
- Long [getKierowcaId \(\)](#)
- Date [getPrawoJazdyWazneDo \(\)](#)
- String [getImie \(\)](#)
- String [getNazwisko \(\)](#)
- String [getPesel \(\)](#)

- `UserDto getUser ()`
- `void setKierowcald (Long kierowcald)`
- `void setPrawoJazdyWazneDo (Date prawoJazdyWazneDo)`
- `void setImie (String imie)`
- `void setNazwisko (String nazwisko)`
- `void setPesel (String pesel)`
- `void setUserDto (UserDto user)`
- `String toString ()`

6.35.1 Opis szczegółowy

Data Transfer Object (DTO) dla encji Kierowca.

Definicja w linii 10 pliku [KierowcaDto.java](#).

6.35.2 Dokumentacja konstruktora i destruktora

6.35.2.1 KierowcaDto() [1/2]

```
com.mklachl.sopkom.model.dto.KierowcaDto.KierowcaDto ( )
```

Konstruktor domyślny.

Definicja w linii 45 pliku [KierowcaDto.java](#).

6.35.2.2 KierowcaDto() [2/2]

```
com.mklachl.sopkom.model.dto.KierowcaDto.KierowcaDto (
    Kierowca kierowca )
```

Konstruktor przyjmujący obiekt encji Kierowca.

Parametry

<code>kierowca</code>	Obiekt encji Kierowca.
-----------------------	------------------------

Definicja w linii 51 pliku [KierowcaDto.java](#).

6.35.3 Dokumentacja funkcji składowych

6.35.3.1 getImie()

```
String com.mklachl.sopkom.model.dto.KierowcaDto.getImie ( )
```

Zwraca imię kierowcy.

Zwraca

 imie Imię kierowcy.

Definicja w linii [85](#) pliku [KierowcaDto.java](#).

6.35.3.2 getKierowcald()

```
Long com.mklachl.sopkom.model.dto.KierowcaDto.getKierowcald ( )
```

Zwraca identyfikator kierowcy.

Zwraca

 kierowcald Identyfikator kierowcy.

Definicja w linii [69](#) pliku [KierowcaDto.java](#).

6.35.3.3 getNazwisko()

```
String com.mklachl.sopkom.model.dto.KierowcaDto.getNazwisko ( )
```

Zwraca nazwisko kierowcy.

Zwraca

 nazwisko Nazwisko kierowcy.

Definicja w linii [93](#) pliku [KierowcaDto.java](#).

6.35.3.4 getPesel()

```
String com.mklachl.sopkom.model.dto.KierowcaDto.getPesel ( )
```

Zwraca PESEL kierowcy.

Zwraca

 pesel PESEL kierowcy.

Definicja w linii [101](#) pliku [KierowcaDto.java](#).

6.35.3.5 getPrawoJazdyWazneDo()

```
Date com.mklachl.sopkom.model.dto.KierowcaDto.getPrawoJazdyWazneDo ( )
```

Zwraca datę ważności prawa jazdy kierowcy.

Zwrota

prawoJazdyWazneDo Data ważności prawa jazdy kierowcy.

Definicja w linii 77 pliku [KierowcaDto.java](#).

6.35.3.6 getUser()

```
UserDto com.mklachl.sopkom.model.dto.KierowcaDto.getUser ( )
```

Zwraca DTO użytkownika związanego z kierowcą.

Zwrota

user DTO użytkownika.

Definicja w linii 109 pliku [KierowcaDto.java](#).

6.35.3.7 setImie()

```
void com.mklachl.sopkom.model.dto.KierowcaDto.setImie ( String imie )
```

Ustawia imię kierowcy.

Parametry

<i>imie</i>	Imię kierowcy.
-------------	----------------

Definicja w linii 135 pliku [KierowcaDto.java](#).

6.35.3.8 setKierowcaId()

```
void com.mklachl.sopkom.model.dto.KierowcaDto.setKierowcaId ( Long kierowcaId )
```

Ustawia identyfikator kierowcy.

Parametry

<i>kierowca</i> ↪ <i>Id</i>	Identyfikator kierowcy.
--------------------------------	-------------------------

Definicja w linii [119](#) pliku [KierowcaDto.java](#).

6.35.3.9 setNazwisko()

```
void com.mklachl.sopkom.model.dto.KierowcaDto.setNazwisko (
    String nazwisko )
```

Ustawia nazwisko kierowcy.

Parametry

<i>nazwisko</i>	Nazwisko kierowcy.
-----------------	--------------------

Definicja w linii [143](#) pliku [KierowcaDto.java](#).

6.35.3.10 setPesel()

```
void com.mklachl.sopkom.model.dto.KierowcaDto.setPesel (
    String pesel )
```

Ustawia PESEL kierowcy.

Parametry

<i>pesel</i>	PESEL kierowcy.
--------------	-----------------

Definicja w linii [151](#) pliku [KierowcaDto.java](#).

6.35.3.11 setPrawoJazdyWazneDo()

```
void com.mklachl.sopkom.model.dto.KierowcaDto.setPrawoJazdyWazneDo (
    Date prawoJazdyWazneDo )
```

Ustawia datę ważności prawa jazdy kierowcy.

Parametry

<i>prawoJazdyWazneDo</i>	Data ważności prawa jazdy kierowcy.
--------------------------	-------------------------------------

Definicja w linii 127 pliku [KierowcaDto.java](#).

6.35.3.12 setUserDto()

```
void com.mklachl.sopkom.model.dto.KierowcaDto.setUserDto (  
    UserDto user )
```

Ustawia DTO użytkownika związanego z kierowcą.

Parametry

<i>user</i>	DTO użytkownika.
-------------	------------------

Definicja w linii 159 pliku [KierowcaDto.java](#).

6.35.3.13 toString()

```
String com.mklachl.sopkom.model.dto.KierowcaDto.toString ( )
```

Zwraca reprezentację obiektu w postaci łańcucha znaków.

Zwrota

Reprezentacja obiektu w postaci łańcucha znaków.

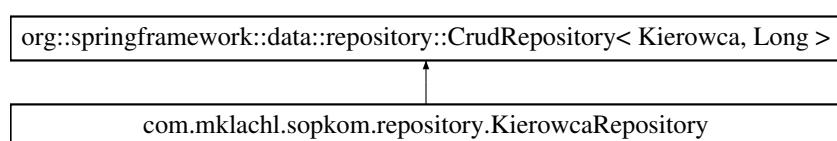
Definicja w linii 168 pliku [KierowcaDto.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.36 Dokumentacja interfejsu com.mklachl.sopkom.repository.KierowcaRepository

Diagram dziedziczenia dla com.mklachl.sopkom.repository.KierowcaRepository



Metody publiczne

- Kierowca `findByPesel` (String pesel)
- List< Kierowca > `findAllByPeselLike` (String pesel)
- Kierowca `findByUser` (User user)
- List< Kierowca > `findAllByImieAndNazwisko` (String imie, String nazwisko)
- List< Kierowca > `findAllByImieAndNazwiskoLike` (String imie, String nazwisko)
- List< Kierowca > `findAllByPrawoJazdyWazneDoBefore` (Date date)
- List< Kierowca > `findAllByPrawoJazdyWazneDoAfter` (Date date)
- List< Kierowca > `findAll` ()
- List< Kierowca > `findByNazwiskoLike` (String nazwisko)
- List< Kierowca > `findAllByAvailability` (@Param("startTime") Date startTime, @Param("endTime") Date endTime)

6.36.1 Opis szczegółowy

Interfejs repository dla encji Kierowca

Definicja w linii 15 pliku `KierowcaRepository.java`.

6.36.2 Dokumentacja funkcji składowych

6.36.2.1 `findAll()`

```
List< Kierowca > com.mklachl.sopkom.repository.KierowcaRepository.findAll ( )
```

Znajduje wszystkich kierowców

Zwraca

Lista kierowców

6.36.2.2 `findAllByAvailability()`

```
List< Kierowca > com.mklachl.sopkom.repository.KierowcaRepository.findAllByAvailability (
    @Param("startTime") Date startTime,
    @Param("endTime") Date endTime )
```

Znajduje wszystkich dostępnych kierowców w podanym przedziale czasowym

Parametry

<code>startTime</code>	czas rozpoczęcia
<code>endTime</code>	czas zakończenia

Zwrota

Lista kierowców

6.36.2.3 findAllByImieAndNazwisko()

```
List< Kierowca > com.mklachl.sopkom.repository.KierowcaRepository.findAllByImieAndNazwisko (  
    String imie,  
    String nazwisko )
```

Znajduje wszystkich kierowców o podanym imieniu i nazwisku

Parametry

<i>imie</i>	imię
<i>nazwisko</i>	nazwisko

Zwrota

Lista kierowców

6.36.2.4 findAllByImieAndNazwiskoLike()

```
List< Kierowca > com.mklachl.sopkom.repository.KierowcaRepository.findAllByImieAndNazwiskoLike (  
(  
    String imie,  
    String nazwisko )
```

Znajduje wszystkich kierowców, których imię i nazwisko pasuje do wzorca

Parametry

<i>imie</i>	imię
<i>nazwisko</i>	nazwisko

Zwrota

Lista kierowców

6.36.2.5 findAllByPeselLike()

```
List< Kierowca > com.mklachl.sopkom.repository.KierowcaRepository.findAllByPeselLike (  
    String pesel )
```

Znajduje wszystkich kierowców, których numer PESEL pasuje do wzorca

Parametry

<i>pesel</i>	numer PESEL
--------------	-------------

Zwroca

Lista kierowców

6.36.2.6 findAllByPrawoJazdyWazneDoAfter()

```
List< Kierowca > com.mklachl.sopkom.repository.KierowcaRepository.findAllByPrawoJazdyWazneDo→
After (
    Date date )
```

Znajduje wszystkich kierowców, których prawo jazdy jest ważne po podanej dacie

Parametry

<i>date</i>	data
-------------	------

Zwroca

Lista kierowców

6.36.2.7 findAllByPrawoJazdyWazneDoBefore()

```
List< Kierowca > com.mklachl.sopkom.repository.KierowcaRepository.findAllByPrawoJazdyWazneDo→
Before (
    Date date )
```

Znajduje wszystkich kierowców, których prawo jazdy jest ważne do podanej daty

Parametry

<i>date</i>	data
-------------	------

Zwroca

Lista kierowców

6.36.2.8 findByNazwiskoLike()

```
List< Kierowca > com.mklachl.sopkom.repository.KierowcaRepository.findByNazwiskoLike ( String nazwisko )
```

Znajduje wszystkich kierowców, których nazwisko pasuje do wzorca

Parametry

<i>nazwisko</i>	nazwisko
-----------------	----------

Zwrota

Lista kierowców

6.36.2.9 findByPesel()

```
Kierowca com.mklachl.sopkom.repository.KierowcaRepository.findByPesel ( String pesel )
```

Znajduje kierowcę po numerze PESEL

Parametry

<i>pesel</i>	numer PESEL
--------------	-------------

Zwrota

Kierowca

6.36.2.10 findByUser()

```
Kierowca com.mklachl.sopkom.repository.KierowcaRepository.findByUser ( User user )
```

Znajduje kierowcę powiązanego z danym użytkownikiem

Parametry

<i>user</i>	użytkownik
-------------	------------

Zwraca

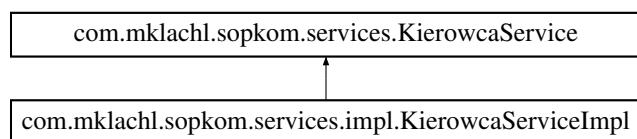
Kierowca

Dokumentacja dla tego interfejsu została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/rep

6.37 Dokumentacja interfejsu **com.mklachl.sopkom.services.KierowcaService**

Diagram dziedziczenia dla com.mklachl.sopkom.services.KierowcaService



Metody publiczne

- Kierowca saveKierowca (Kierowca kierowca)
- Kierowca saveKierowca (KierowcaDto kierowcaDto)
- Kierowca findKierowcaByPesel (String pesel)
- Kierowca findKierowcaByUser (User user)
- List< Kierowca > findKierowcaByImieAndNazwisko (String imie, String nazwisko)
- List< Kierowca > findKierowcaByPrawoJazdyWazneDoBefore (Date date)
- List< Kierowca > findKierowcaByPrawoJazdyWazneDoAfter (Date date)
- Optional< Kierowca > findKierowcaById (Long id)
- List< Kierowca > findKierowcaByAvailability (Date startTime, Date endTime)

6.37.1 Opis szczegółowy

Interfejs serwisu dla obsługi kierowców

Definicja w linii 15 pliku [KierowcaService.java](#).

6.37.2 Dokumentacja funkcji składowych

6.37.2.1 findKierowcaByAvailability()

```
List< Kierowca > com.mklachl.sopkom.services.KierowcaService.findKierowcaByAvailability ( Date startTime, Date endTime )
```

Znajduje dostępnych kierowców w podanym przedziale czasowym

Parametry

<i>startTime</i>	czas rozpoczęcia
<i>endTime</i>	czas zakończenia

Zwrota

lista dostępnych kierowców

Implementowany w [com.mklachl.sopkom.services.impl.KierowcaServiceImpl](#).

6.37.2.2 findKierowcaById()

```
Optional< Kierowca > com.mklachl.sopkom.services.KierowcaService.findKierowcaById ( Long id )
```

Znajduje kierowcę po ID

Parametry

<i>id</i>	ID kierowcy
-----------	-------------

Zwrota

Optional z kierowca

Implementowany w [com.mklachl.sopkom.services.impl.KierowcaServiceImpl](#).

6.37.2.3 findKierowcaByImieAndNazwisko()

```
List< Kierowca > com.mklachl.sopkom.services.KierowcaService.findKierowcaByImieAndNazwisko ( String imie, String nazwisko )
```

Znajduje kierowców po imieniu i nazwisku

Parametry

<i>imie</i>	imię
<i>nazwisko</i>	nazwisko

Zwrota

lista kierowców

Implementowany w [com.mklachl.sopkom.services.impl.KierowcaServiceImpl](#).

6.37.2.4 findKierowcaByPesel()

```
Kierowca com.mklachl.sopkom.services.KierowcaService.findKierowcaByPesel (
    String pesel )
```

Znajduje kierowcę po numerze PESEL

Parametry

<i>pesel</i>	numer PESEL
--------------	-------------

Zwrota

znaleziony kierowca

Implementowany w [com.mklachl.sopkom.services.impl.KierowcaServiceImpl](#).

6.37.2.5 findKierowcaByPrawoJazdyWazneDoAfter()

```
List< Kierowca > com.mklachl.sopkom.services.KierowcaService.findKierowcaByPrawoJazdyWazneDo←
After (
    Date date )
```

Znajduje kierowców, których prawo jazdy jest ważne po podanej dacie

Parametry

<i>date</i>	data
-------------	------

Zwrota

lista kierowców

Implementowany w [com.mklachl.sopkom.services.impl.KierowcaServiceImpl](#).

6.37.2.6 findKierowcaByPrawoJazdyWazneDoBefore()

```
List< Kierowca > com.mklachl.sopkom.services.KierowcaService.findKierowcaByPrawoJazdyWazneDo←
Before (
    Date date )
```

Znajduje kierowców, których prawo jazdy jest ważne do podanej daty

Parametry

<i>date</i>	data
-------------	------

Zwrota

lista kierowców

Implementowany w [com.mklachl.sopkom.services.impl.KierowcaServiceImpl](#).

6.37.2.7 findKierowcaByUser()

```
Kierowca com.mklachl.sopkom.services.KierowcaService.findKierowcaByUser (
    User user )
```

Znajduje kierowcę powiązanego z użytkownikiem

Parametry

<i>user</i>	użytkownik
-------------	------------

Zwrota

znaleziony kierowca

Implementowany w [com.mklachl.sopkom.services.impl.KierowcaServiceImpl](#).

6.37.2.8 saveKierowca() [1/2]

```
Kierowca com.mklachl.sopkom.services.KierowcaService.saveKierowca (
    Kierowca kierowca )
```

Zapisuje nowego kierowcę

Parametry

<i>kierowca</i>	encja kierowcy
-----------------	----------------

Zwrota

zapisany kierowca

Implementowany w [com.mklachl.sopkom.services.impl.KierowcaServiceImpl](#).

6.37.2.9 saveKierowca() [2/2]

```
Kierowca com.mklachl.sopkom.services.KierowcaService.saveKierowca (
    KierowcaDto kierowcaDto )
```

Zapisuje nowego kierowcę na podstawie DTO

Parametry

<i>kierowcaDto</i>	DTO kierowcy
--------------------	--------------

Zwrota

zapisany kierowca

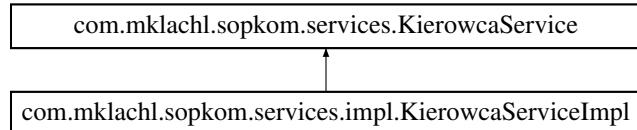
Implementowany w [com.mklachl.sopkom.services.impl.KierowcaServiceImpl](#).

Dokumentacja dla tego interfejsu została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/se

6.38 Dokumentacja klasy **com.mklachl.sopkom.services.impl.KierowcaServiceImpl**

Diagram dziedziczenia dla com.mklachl.sopkom.services.impl.KierowcaServiceImpl



Metody publiczne

- [KierowcaServiceImpl \(KierowcaRepository\) kierowcaRepository, UserService userService\)](#)
- [Kierowca saveKierowca \(Kierowca kierowcaDto\)](#)
- [Kierowca saveKierowca \(KierowcaDto kierowcaDto\)](#)
- [Kierowca findKierowcaByPesel \(String pesel\)](#)
- [Kierowca findKierowcaByUser \(User user\)](#)
- [List< Kierowca > findKierowcaByImieAndNazwisko \(String imie, String nazwisko\)](#)
- [List< Kierowca > findKierowcaByPrawoJazdyWazneDoBefore \(Date date\)](#)
- [List< Kierowca > findKierowcaByPrawoJazdyWazneDoAfter \(Date date\)](#)
- [Optional< Kierowca > findKierowcaById \(Long id\)](#)
- [List< Kierowca > findKierowcaByAvailability \(Date startTime, Date endTime\)](#)

6.38.1 Opis szczegółowy

Definicja w linii 16 pliku [KierowcaServiceImpl.java](#).

6.38.2 Dokumentacja konstruktora i destruktora

6.38.2.1 KierowcaServiceImpl()

```
com.mklachl.sopkom.services.impl.KierowcaServiceImpl.KierowcaServiceImpl (  
    KierowcaRepository kierowcaRepository,  
    UserService userService )
```

Definicja w linii 20 pliku [KierowcaServiceImpl.java](#).

6.38.3 Dokumentacja funkcji składowych

6.38.3.1 findKierowcaByAvailability()

```
List< Kierowca > com.mklachl.sopkom.services.impl.KierowcaServiceImpl.findKierowcaByAvailability  
(  
    Date startTime,  
    Date endTime )
```

Znajduje dostępnych kierowców w podanym przedziale czasowym

Parametry

startTime	czas rozpoczęcia
endTime	czas zakończenia

Zwrota

lista dostępnych kierowców

Implementuje [com.mklachl.sopkom.services.KierowcaService](#).

Definicja w linii 86 pliku [KierowcaServiceImpl.java](#).

6.38.3.2 findKierowcaById()

```
Optional< Kierowca > com.mklachl.sopkom.services.impl.KierowcaServiceImpl.findKierowcaById (Long id)
```

Znajduje kierowcę po ID

Parametry

<i>id</i>	ID kierowcy
-----------	-------------

Zwroca

Optional z kierowcą

Implementuje [com.mklachl.sopkom.services.KierowcaService](#).

Definicja w linii 81 pliku [KierowcaServiceImpl.java](#).

6.38.3.3 findKierowcaByImieAndNazwisko()

```
List< Kierowca > com.mklachl.sopkom.services.impl.KierowcaServiceImpl.findKierowcaByImieAnd←
Nazwisko (
    String imie,
    String nazwisko )
```

Znajduje kierowców po imieniu i nazwisku

Parametry

<i>imie</i>	imię
<i>nazwisko</i>	nazwisko

Zwroca

lista kierowców

Implementuje [com.mklachl.sopkom.services.KierowcaService](#).

Definicja w linii 66 pliku [KierowcaServiceImpl.java](#).

6.38.3.4 findKierowcaByPesel()

```
Kierowca com.mklachl.sopkom.services.impl.KierowcaServiceImpl.findKierowcaByPesel (
    String pesel )
```

Znajduje kierowcę po numerze PESEL

Parametry

<i>pesel</i>	numer PESEL
--------------	-------------

Zwrota

znaleziony kierowca

Implementuje [com.mklachl.sopkom.services.KierowcaService](#).

Definicja w linii 56 pliku [KierowcaServiceImpl.java](#).

6.38.3.5 findKierowcaByPrawoJazdyWazneDoAfter()

```
List< Kierowca > com.mklachl.sopkom.services.impl.KierowcaServiceImpl.findKierowcaByPrawoJazdyWazneDoAfter ( Date date )
```

Znajduje kierowców, których prawo jazdy jest ważne po podanej dacie

Parametry

<i>date</i>	data
-------------	------

Zwrota

lista kierowców

Implementuje [com.mklachl.sopkom.services.KierowcaService](#).

Definicja w linii 76 pliku [KierowcaServiceImpl.java](#).

6.38.3.6 findKierowcaByPrawoJazdyWazneDoBefore()

```
List< Kierowca > com.mklachl.sopkom.services.impl.KierowcaServiceImpl.findKierowcaByPrawoJazdyWazneDoBefore ( Date date )
```

Znajduje kierowców, których prawo jazdy jest ważne do podanej daty

Parametry

<i>date</i>	data
-------------	------

Zwrota

lista kierowców

Implementuje [com.mklachl.sopkom.services.KierowcaService](#).

Definicja w linii 71 pliku [KierowcaServiceImpl.java](#).

6.38.3.7 findKierowcaByUser()

```
Kierowca com.mklachl.sopkom.services.impl.KierowcaServiceImpl.findKierowcaByUser (
    User user )
```

Znajduje kierowcę powiązanego z użytkownikiem

Parametry

<i>user</i>	użytkownik
-------------	------------

Zwrota

znaleziony kierowca

Implementuje [com.mklachl.sopkom.services.KierowcaService](#).

Definicja w linii 61 pliku [KierowcaServiceImpl.java](#).

6.38.3.8 saveKierowca() [1/2]

```
Kierowca com.mklachl.sopkom.services.impl.KierowcaServiceImpl.saveKierowca (
    Kierowca kierowca )
```

Zapisuje nowego kierowcę

Parametry

<i>kierowca</i>	encja kierowcy
-----------------	----------------

Zwrota

zapisany kierowca

Implementuje [com.mklachl.sopkom.services.KierowcaService](#).

Definicja w linii 26 pliku [KierowcaServiceImpl.java](#).

6.38.3.9 saveKierowca() [2/2]

```
Kierowca com.mklachl.sopkom.services.impl.KierowcaServiceImpl.saveKierowca (
    KierowcaDto kierowcaDto )
```

Zapisuje nowego kierowcę na podstawie DTO

Parametry

<i>kierowcaDto</i>	DTO kierowcy
--------------------	--------------

Zwraca

zapisany kierowca

Implementuje [com.mklachl.sopkom.services.KierowcaService](#).

Definicja w linii 38 pliku [KierowcaServiceImpl.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/ser...

6.39 Dokumentacja klasy com.mklachl.sopkom.model.entity.Kurs

Metody publiczne

- Long [getKursId \(\)](#)
- void [setKursId \(Long kursId\)](#)
- [Linia getLinia \(\)](#)
- void [setLinia \(Linia linia\)](#)
- [Harmonogram getHarmonogram \(\)](#)
- void [setHarmonogram \(Harmonogram harmonogram\)](#)
- Short [getKierunek \(\)](#)
- void [setKierunek \(Short kierunek\)](#)
- Short [getTypAutobusu \(\)](#)
- void [setTypAutobusu \(Short typAutobusu\)](#)
- List< [KursPrzystanekWlini >](#) [getKursPrzystanekWlinii \(\)](#)
- void [setKursPrzystanekWlinii \(List< KursPrzystanekWlini > kursPrzystanekWlinii\)](#)

6.39.1 Opis szczegółowy

Klasa reprezentująca encję [Kurs](#).

Definicja w linii 11 pliku [Kurs.java](#).

6.39.2 Dokumentacja funkcji składowych

6.39.2.1 getHarmonogram()

```
Harmonogram com.mklachl.sopkom.model.entity.Kurs.getHarmonogram ( )
```

Zwraca harmonogram związany z kursem.

Zwraca

harmonogram [Harmonogram](#) związany z kursem.

Definicja w linii [89](#) pliku [Kurs.java](#).

6.39.2.2 getKierunek()

```
Short com.mklachl.sopkom.model.entity.Kurs.getKierunek ( )
```

Zwraca kierunek kursu.

Zwraca

kierunek [Kierunek](#) kursu.

Definicja w linii [105](#) pliku [Kurs.java](#).

6.39.2.3 getKursId()

```
Long com.mklachl.sopkom.model.entity.Kurs.getKursId ( )
```

Zwraca identyfikator kursu.

Zwraca

kursId [Identyfikator](#) kursu.

Definicja w linii [57](#) pliku [Kurs.java](#).

6.39.2.4 getKursPrzystanekWlinii()

```
List< KursPrzystanekWlini > com.mklachl.sopkom.model.entity.Kurs.getKursPrzystanekWlinii ( )
```

Zwraca listę przystanków w linii związanych z kursem.

Zwraca

kursPrzystanekWlinii [Lista](#) przystanków w linii.

Definicja w linii [137](#) pliku [Kurs.java](#).

6.39.2.5 getLinia()

```
Linia com.mklachl.sopkom.model.entity.Kurs.getLinia ( )
```

Zwraca linię związaną z kursem.

Zwrota

linia [Linia](#) związana z kursem.

Definicja w linii [73](#) pliku [Kurs.java](#).

6.39.2.6 getTypAutobusu()

```
Short com.mklachl.sopkom.model.entity.Kurs.getTypAutobusu ( )
```

Zwraca typ autobusu używanego na kursie.

Zwrota

typAutobusu Typ autobusu.

Definicja w linii [121](#) pliku [Kurs.java](#).

6.39.2.7 setHarmonogram()

```
void com.mklachl.sopkom.model.entity.Kurs.setHarmonogram (   
    Harmonogram harmonogram )
```

Ustawia harmonogram związany z kursem.

Parametry

<i>harmonogram</i>	Harmonogram związany z kursem.
--------------------	--

Definicja w linii [97](#) pliku [Kurs.java](#).

6.39.2.8 setKierunek()

```
void com.mklachl.sopkom.model.entity.Kurs.setKierunek (   
    Short kierunek )
```

Ustawia kierunek kursu.

Parametry

<i>kierunek</i>	Kierunek kursu.
-----------------	-----------------

Definicja w linii [113](#) pliku [Kurs.java](#).

6.39.2.9 setKursId()

```
void com.mklachl.sopkom.model.entity.Kurs.setKursId (   
    Long kursId )
```

Ustawia identyfikator kursu.

Parametry

<i>kurs</i> ↪ <i>Id</i>	Identyfikator kursu.
----------------------------	----------------------

Definicja w linii [65](#) pliku [Kurs.java](#).

6.39.2.10 setKursPrzystanekWlinii()

```
void com.mklachl.sopkom.model.entity.Kurs.setKursPrzystanekWlinii (   
    List< KursPrzystanekWlinii > kursPrzystanekWlinii )
```

Ustawia listę przystanków w linii związanych z kursem.

Parametry

<i>kursPrzystanekWlinii</i>	Lista przystanków w linii.
-----------------------------	----------------------------

Definicja w linii [145](#) pliku [Kurs.java](#).

6.39.2.11 setLinia()

```
void com.mklachl.sopkom.model.entity.Kurs.setLinia (   
    Linia linia )
```

Ustawia linię związaną z kursem.

Parametry

<i>linia</i>	Linia związana z kursem.
--------------	--------------------------

Definicja w linii 81 pliku [Kurs.java](#).

6.39.2.12 setTypAutobusu()

```
void com.mklachl.sopkom.model.entity.Kurs.setTypAutobusu (
    Short typAutobusu )
```

Ustawia typ autobusu używanego na kursie.

Parametry

<i>typAutobusu</i>	Typ autobusu.
--------------------	---------------

Definicja w linii 129 pliku [Kurs.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.40 Dokumentacja klasy com.mklachl.sopkom.controller.KursController

Metody publiczne

- ResponseEntity<?> [getTemplateKurs](#) (@PathVariable(name="id") Long linia_id)
- ResponseEntity<?> [getTemplateKursInput](#) (@PathVariable(name="id") Long linia_id)
- ResponseEntity<?> [getAllKursy](#) (@RequestParam(name="linia_id", defaultValue="-1") Long linia_id)
- ResponseEntity<?> [getKurs](#) (@PathVariable(name="id") Long id)
- ResponseEntity<?> [addKurs](#) (@RequestBody [InputKursDto](#) inputKursDto)
- ResponseEntity<?> [updateKurs](#) (@PathVariable(name="id") Long id, @RequestBody [InputKursDto](#) inputKursDto)
- ResponseEntity<?> [deleteKurs](#) (@PathVariable(name="id") Long id)

Atrybuty publiczne

- [LiniaService](#) liniaService
- [LiniaRepository](#) liniaRepository
- [PrzystanekWliniRepository](#) przystanekWliniRepository
- [KursService](#) kursService
- [KursRepository](#) kursRepo

Statyczne atrybuty publiczne

- static List< KursDto > kursy = new ArrayList<KursDto>()

6.40.1 Opis szczegółowy

Kontroler dla zasobu Kurs. Obsługuje operacje tworzenia, pobierania, aktualizacji i usuwania kursów.

Definicja w linii 48 pliku [KursController.java](#).

6.40.2 Dokumentacja funkcji składowych

6.40.2.1 addKurs()

```
ResponseEntity<?> com.mklachl.sopkom.controller.KursController.addKurs (   
    @RequestBody InputKursDto inputKursDto )
```

Tworzy nowy kurs.

Parametry

<i>inputKursDto</i>	Obiekt InputKursDto.
---------------------	----------------------

Zwraca

Status operacji.

Definicja w linii 223 pliku [KursController.java](#).

6.40.2.2 deleteKurs()

```
ResponseEntity<?> com.mklachl.sopkom.controller.KursController.deleteKurs (   
    @PathVariable(name="id") Long id )
```

Usuwa kurs po identyfikatorze.

Parametry

<i>id</i>	Identyfikator kursu.
-----------	----------------------

Zwrota

Status operacji.

Definicja w linii 364 pliku KursController.java.

6.40.2.3 getAllKursy()

```
ResponseEntity<?> com.mklachl.sopkom.controller.KursController.getAllKursy ( @RequestParam(name="linia_id", defaultValue="-1") Long linia_id )
```

Pobiera wszystkie kursy lub filtrowane po identyfikatorze linii.

Parametry

<i>linia</i> <i>_id</i>	Identyfikator linii (opcjonalny).
----------------------------	-----------------------------------

Zwrota

Lista obiektów KursDto.

Definicja w linii 176 pliku KursController.java.

6.40.2.4 getKurs()

```
ResponseEntity<?> com.mklachl.sopkom.controller.KursController.getKurs ( @PathVariable(name="id") Long id )
```

Pobiera kurs po identyfikatorze.

Parametry

<i>id</i>	Identyfikator kursu.
-----------	----------------------

Zwrota

Obiekt KursDto.

Definicja w linii 213 pliku KursController.java.

6.40.2.5 getTemplateKurs()

```
ResponseEntity<?> com.mklachl.sopkom.controller.KursController.getTemplateKurs (   
    @PathVariable(name="id") Long linia_id )
```

Szablon obiektu KursDto.

Parametry

<i>linia</i> <i>_id</i>	Identyfikator linii.
----------------------------	----------------------

Zwroca

Obiekt KursDto.

Definicja w linii [78](#) pliku [KursController.java](#).

6.40.2.6 getTemplateKursInput()

```
ResponseEntity<?> com.mklachl.sopkom.controller.KursController.getTemplateKursInput (   
    @PathVariable(name="id") Long linia_id )
```

Szablon obiektu InputKursDto.

Parametry

<i>linia</i> <i>_id</i>	Identyfikator linii.
----------------------------	----------------------

Zwroca

Obiekt InputKursDto.

Definicja w linii [128](#) pliku [KursController.java](#).

6.40.2.7 updateKurs()

```
ResponseEntity<?> com.mklachl.sopkom.controller.KursController.updateKurs (   
    @PathVariable(name="id") Long id,   
    @RequestBody InputKursDto inputKursDto )
```

Aktualizuje kurs po identyfikatorze.

Parametry

<i>id</i>	Identyfikator kursu.
<i>inputKursDto</i>	Obiekt InputKursDto.

Zwraca

Status operacji.

Definicja w linii [293](#) pliku [KursController.java](#).

6.40.3 Dokumentacja atrybutów składowych

6.40.3.1 kursRepo

[KursRepository](#) com.mklachl.sopkom.controller.KursController.kursRepo

Definicja w linii [70](#) pliku [KursController.java](#).

6.40.3.2 kursService

[KursService](#) com.mklachl.sopkom.controller.KursController.kursService

Definicja w linii [67](#) pliku [KursController.java](#).

6.40.3.3 kursy

```
List<KursDto> com.mklachl.sopkom.controller.KursController.kursy = new ArrayList<KursDto>()  
[static]
```

Definicja w linii [50](#) pliku [KursController.java](#).

6.40.3.4 liniaRepository

[LiniaRepository](#) com.mklachl.sopkom.controller.KursController.liniaRepository

Definicja w linii [60](#) pliku [KursController.java](#).

6.40.3.5 liniaService

`LiniaService` com.mklachl.sopkom.controller.KursController.liniaService

Definicja w linii 57 pliku [KursController.java](#).

6.40.3.6 przystnekWliniRepository

`PrzystanekWliniRepository` com.mklachl.sopkom.controller.KursController.przystnekWliniRepository

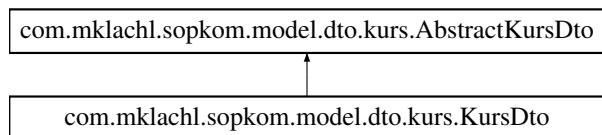
Definicja w linii 63 pliku [KursController.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/com

6.41 Dokumentacja klasy com.mklachl.sopkom.model.dto.kurs.KursDto

Diagram dziedziczenia dla com.mklachl.sopkom.model.dto.kurs.KursDto



Metody publiczne

- `KursDto ()`
- `KursDto (Kurs kurs)`
- `List< InputPrzystanekWKursieDto > getPrzystanki ()`
- `void setPrzystanki (List< InputPrzystanekWKursieDto > przystanki)`
- `HarmonogramDto getHarmonogram ()`
- `void setHarmonogram (HarmonogramDto harmonogram)`
- `LiniaDto getLinia ()`
- `void setLinia (LiniaDto linia)`
- `boolean getOdwracalny ()`
- `void setOdwracalny (boolean odwracalny)`

Dodatkowe Dziedziczone Składowe

6.41.1 Opis szczegółowy

Data Transfer Object (DTO) dla encji Kurs.

Definicja w linii 12 pliku [KursDto.java](#).

6.41.2 Dokumentacja konstruktora i destruktora

6.41.2.1 KursDto() [1/2]

```
com.mklachl.sopkom.model.dto.kurs.KursDto.KursDto( )
```

Konstruktor domyślny.

Definicja w linii 37 pliku [KursDto.java](#).

6.41.2.2 KursDto() [2/2]

```
com.mklachl.sopkom.model.dto.kurs.KursDto.KursDto(  
    Kurs kurs)
```

Konstruktor przyjmujący obiekt encji Kurs.

Parametry

<i>kurs</i>	Obiekt encji Kurs.
-------------	--------------------

Definicja w linii 44 pliku [KursDto.java](#).

6.41.3 Dokumentacja funkcji składowych

6.41.3.1 getHarmonogram()

```
HarmonogramDto com.mklachl.sopkom.model.dto.kurs.KursDto.getHarmonogram( )
```

Zwraca DTO harmonogramu związanego z kursem.

Zwroca

harmonogram DTO harmonogramu.

Definicja w linii 78 pliku [KursDto.java](#).

6.41.3.2 getLinia()

```
LiniaDto com.mklachl.sopkom.model.dto.kurs.KursDto.getLinia ( )
```

Zwraca DTO linii związanej z kursem.

Zwraca

linia DTO linii.

Definicja w linii [95](#) pliku [KursDto.java](#).

6.41.3.3 getOdwracalny()

```
boolean com.mklachl.sopkom.model.dto.kurs.KursDto.getOdwracalny ( )
```

Zwraca wartość flagi określającej, czy linia jest odwracalna.

Zwraca

odwracalny Wartość flagi odwracalności linii.

Definicja w linii [112](#) pliku [KursDto.java](#).

6.41.3.4 getPrzystanki()

```
List< InputPrzystanekWKursieDto > com.mklachl.sopkom.model.dto.kurs.KursDto.getPrzystanki ( )
```

Zwraca listę przystanków w kursie.

Zwraca

przystanki Lista przystanków w kursie.

Definicja w linii [62](#) pliku [KursDto.java](#).

6.41.3.5 setHarmonogram()

```
void com.mklachl.sopkom.model.dto.kurs.KursDto.setHarmonogram ( HarmonogramDto harmonogram )
```

Ustawia DTO harmonogramu związanego z kursem.

Parametry

<i>harmonogram</i>	DTO harmonogramu.
--------------------	-------------------

Definicja w linii 86 pliku [KursDto.java](#).

6.41.3.6 setLinia()

```
void com.mklachl.sopkom.model.dto.kurs.KursDto.setLinia (
    LiniaDto linia )
```

Ustawia DTO linii związanej z kursem.

Parametry

<i>linia</i>	DTO linii.
--------------	------------

Definicja w linii 103 pliku [KursDto.java](#).

6.41.3.7 setOdwracalny()

```
void com.mklachl.sopkom.model.dto.kurs.KursDto.setOdwracalny (
    boolean odwracalny )
```

Ustawia wartość flagi określającej, czy linia jest odwracalna.

Parametry

<i>odwracalny</i>	Wartość flagi odwracalności linii.
-------------------	------------------------------------

Definicja w linii 120 pliku [KursDto.java](#).

6.41.3.8 setPrzystanki()

```
void com.mklachl.sopkom.model.dto.kurs.KursDto.setPrzystanki (
    List< InputPrzystanekWKursieDto > przystanki )
```

Ustawia listę przystanków w kursie.

Parametry

<i>przystanki</i>	Lista przystanków w kursie.
-------------------	-----------------------------

Definicja w linii 70 pliku [KursDto.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.42 Dokumentacja klasy com.mklachl.sopkom.model.dto.kurs.[OUTDATED.KursDtoOUTDATED](#)

Metody publiczne

- [KursDtoOUTDATED \(\)](#)
- [KursDtoOUTDATED \(Kurs kurs\)](#)
- Long [getKursId \(\)](#)
- void [setKursId \(Long kursId\)](#)
- [HarmonogramDto getHarmonogram \(\)](#)
- void [setHarmonogram \(HarmonogramDto harmonogram\)](#)
- [LiniaDto getLinia \(\)](#)
- void [setLinia \(LiniaDto linia\)](#)
- boolean [isOdwracalny \(\)](#)
- void [setOdwracalny \(boolean odwracalny\)](#)
- List< [PrzystenekWKursieDtoOUTDATED >](#) getPrzystanki ()
- void [setPrzystanki \(List< PrzystenekWKursieDtoOUTDATED > przystanki\)](#)
- Short [getKieruenk \(\)](#)
- void [setKieruenk \(Short kieruenk\)](#)

Atrybuty chronione

- [LiniaDto linia](#)
- Long [kursId](#)
- boolean [odwracalny](#)
- Short [kieruenk](#)
- [HarmonogramDto harmonogram](#)
- List< [PrzystenekWKursieDtoOUTDATED >](#) przystanki

6.42.1 Opis szczegółowy

Definicja w linii 10 pliku [KursDtoOUTDATED.java](#).

6.42.2 Dokumentacja konstruktora i destruktora

6.42.2.1 KursDtoOUTDATED() [1/2]

```
com.mklachl.sopkom.model.dto.kurs.OUTDATED.KursDtoOUTDATED.KursDtoOUTDATED ( )
```

Definicja w linii 23 pliku [KursDtoOUTDATED.java](#).

6.42.2.2 KursDtoOUTDATED() [2/2]

```
com.mklachl.sopkom.model.dto.kurs.OUTDATED.KursDtoOUTDATED.KursDtoOUTDATED (   
    Kurs kurs )
```

Definicja w linii 27 pliku [KursDtoOUTDATED.java](#).

6.42.3 Dokumentacja funkcji składowych

6.42.3.1 getHarmonogram()

```
HarmonogramDto com.mklachl.sopkom.model.dto.kurs.OUTDATED.KursDtoOUTDATED.getHarmonogram ( )
```

Definicja w linii 42 pliku [KursDtoOUTDATED.java](#).

6.42.3.2 getKieruenk()

```
Short com.mklachl.sopkom.model.dto.kurs.OUTDATED.KursDtoOUTDATED.getKieruenk ( )
```

Definicja w linii 74 pliku [KursDtoOUTDATED.java](#).

6.42.3.3 getKursId()

```
Long com.mklachl.sopkom.model.dto.kurs.OUTDATED.KursDtoOUTDATED.getKursId ( )
```

Definicja w linii 34 pliku [KursDtoOUTDATED.java](#).

6.42.3.4 getLinia()

```
LiniaDto com.mklachl.sopkom.model.dto.kurs.OUTDATED.KursDtoOUTDATED.getLinia ( )
```

Definicja w linii 50 pliku [KursDtoOUTDATED.java](#).

6.42.3.5 getPrzystanki()

```
List< PrzystenekWKursieDtoOUTDATED > com.mklachl.sopkom.model.dto.kurs.OUTDATED.KursDto←  
OUTDATED.getPrzystanki ( )
```

Definicja w linii 66 pliku [KursDtoOUTDATED.java](#).

6.42.3.6 isOdwracalny()

```
boolean com.mklachl.sopkom.model.dto.kurs.OUTDATED.KursDtoOUTDATED.isOdwracalny ( )
```

Definicja w linii 58 pliku [KursDtoOUTDATED.java](#).

6.42.3.7 setHarmonogram()

```
void com.mklachl.sopkom.model.dto.kurs.OUTDATED.KursDtoOUTDATED.setHarmonogram (  
    HarmonogramDto harmonogram )
```

Definicja w linii 46 pliku [KursDtoOUTDATED.java](#).

6.42.3.8 setKieruenk()

```
void com.mklachl.sopkom.model.dto.kurs.OUTDATED.KursDtoOUTDATED.setKieruenk (   
    Short kieruenk )
```

Definicja w linii 78 pliku [KursDtoOUTDATED.java](#).

6.42.3.9 setKursId()

```
void com.mklachl.sopkom.model.dto.kurs.OUTDATED.KursDtoOUTDATED.setKursId (   
    Long kursId )
```

Definicja w linii 38 pliku [KursDtoOUTDATED.java](#).

6.42.3.10 setLinia()

```
void com.mklachl.sopkom.model.dto.kurs.OUTDATED.KursDtoOUTDATED.setLinia (   
    LiniaDto linia )
```

Definicja w linii 54 pliku [KursDtoOUTDATED.java](#).

6.42.3.11 setOdwracalny()

```
void com.mklachl.sopkom.model.dto.kurs.OUTDATED.KursDtoOUTDATED.setOdwracalny ( boolean odwracalny )
```

Definicja w linii 62 pliku [KursDtoOUTDATED.java](#).

6.42.3.12 setPrzystanki()

```
void com.mklachl.sopkom.model.dto.kurs.OUTDATED.KursDtoOUTDATED.setPrzystanki ( List< PrzystenekWKursieDtoOUTDATED > przystanki )
```

Definicja w linii 70 pliku [KursDtoOUTDATED.java](#).

6.42.4 Dokumentacja atrybutów składowych

6.42.4.1 harmonogram

```
HarmonogramDto com.mklachl.sopkom.model.dto.kurs.OUTDATED.KursDtoOUTDATED.harmonogram [protected]
```

Definicja w linii 19 pliku [KursDtoOUTDATED.java](#).

6.42.4.2 kieruenk

```
Short com.mklachl.sopkom.model.dto.kurs.OUTDATED.KursDtoOUTDATED.kieruenk [protected]
```

Definicja w linii 17 pliku [KursDtoOUTDATED.java](#).

6.42.4.3 kursId

```
Long com.mklachl.sopkom.model.dto.kurs.OUTDATED.KursDtoOUTDATED.kursId [protected]
```

Definicja w linii 13 pliku [KursDtoOUTDATED.java](#).

6.42.4.4 linia

```
LiniaDto com.mklachl.sopkom.model.dto.kurs.UPDATED.KursDtoOUTDATED.linia [protected]
```

Definicja w linii 12 pliku [KursDtoOUTDATED.java](#).

6.42.4.5 odwracalny

```
boolean com.mklachl.sopkom.model.dto.kurs.UPDATED.KursDtoOUTDATED.odwracalny [protected]
```

Definicja w linii 15 pliku [KursDtoOUTDATED.java](#).

6.42.4.6 przystanki

```
List<PrzystenekWKursieDtoOUTDATED> com.mklachl.sopkom.model.dto.kurs.UPDATED.KursDtoOUTDATED.przystanki [protected]
```

Definicja w linii 21 pliku [KursDtoOUTDATED.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo
OUTDATED/KursDtoOUTDATED.java

6.43 Dokumentacja klasy

com.mklachl.sopkom.model.entity.KursPrzstanekWlini

Metody publiczne

- Long [getKursPrzstanekWliniId](#) ()
- void [setKursPrzstanekWliniId](#) (Long kursPrzstanekWliniId)
- Kurs [getKurs](#) ()
- void [setKurs](#) (Kurs kurs)
- PrzystanekWlini [getPrzystanekWlini](#) ()
- void [setPrzystanekWlini](#) (PrzystanekWlini przystanekWlini)
- Date [getGodzinna](#) ()
- void [setGodzinna](#) (Date godzinna)

6.43.1 Opis szczegółowy

Klasa reprezentująca encję [KursPrzstanekWlini](#).

Definicja w linii 11 pliku [KursPrzstanekWlini.java](#).

6.43.2 Dokumentacja funkcji składowych

6.43.2.1 getGodzinna()

```
Date com.mklachl.sopkom.model.entity.KursPrzystanekWlini.getGodzinna ( )
```

Zwraca godzinę przystanku w linii.

Zwrota

godzinna Godzina przystanku w linii.

Definicja w linii 94 pliku [KursPrzystanekWlini.java](#).

6.43.2.2 getKurs()

```
Kurs com.mklachl.sopkom.model.entity.KursPrzystanekWlini.getKurs ( )
```

Zwraca kurs związany z [KursPrzystanekWlini](#).

Zwrota

kurs [Kurs](#) związany z [KursPrzystanekWlini](#).

Definicja w linii 62 pliku [KursPrzystanekWlini.java](#).

6.43.2.3 getKursPrzystanekWliniId()

```
Long com.mklachl.sopkom.model.entity.KursPrzystanekWlini.getKursPrzystanekWliniId ( )
```

Zwraca identyfikator [KursPrzystanekWlini](#).

Zwrota

kursPrzystanekWliniId Identyfikator [KursPrzystanekWlini](#).

Definicja w linii 46 pliku [KursPrzystanekWlini.java](#).

6.43.2.4 `getPrzystanekWlini()`

```
PrzystanekWlini com.mklachl.sopkom.model.entity.KursPrzystanekWlini.getPrzystanekWlini ( )
```

Zwraca przystanek w linii związanego z [KursPrzystanekWlini](#).

Zwrota

przystanekWlini [Przystanek](#) w linii.

Definicja w linii [78](#) pliku [KursPrzystanekWlini.java](#).

6.43.2.5 `setGodzinna()`

```
void com.mklachl.sopkom.model.entity.KursPrzystanekWlini.setGodzinna ( Date godzinna )
```

Ustawia godzinę przystanku w linii.

Parametry

<i>godzinna</i>	Godzina przystanku w linii.
-----------------	-----------------------------

Definicja w linii 102 pliku [KursPrzystanekWlini.java](#).

6.43.2.6 setKurs()

```
void com.mklachl.sopkom.model.entity.KursPrzystanekWlini.setKurs (   
    Kurs kurs )
```

Ustawia kurs zвязany z [KursPrzystanekWlini](#).

Parametry

<i>kurs</i>	Kurs zвязany z KursPrzystanekWlini .
-------------	--

Definicja w linii 70 pliku [KursPrzystanekWlini.java](#).

6.43.2.7 setKursPrzystanekWliniId()

```
void com.mklachl.sopkom.model.entity.KursPrzystanekWlini.setKursPrzystanekWliniId (   
    Long kursPrzystanekWliniId )
```

Ustawia identyfikator [KursPrzystanekWlini](#).

Parametry

<i>kursPrzystanekWliniId</i>	Identyfikator KursPrzystanekWlini .
------------------------------	---

Definicja w linii 54 pliku [KursPrzystanekWlini.java](#).

6.43.2.8 setPrzystanekWlini()

```
void com.mklachl.sopkom.model.entity.KursPrzystanekWlini.setPrzystanekWlini (   
    PrzystanekWlini przystanekWlini )
```

Ustawia przystanek w linii zвязany z [KursPrzystanekWlini](#).

Parametry

<code>przystanekWlini</code>	<code>Przystanek</code> w linii.
------------------------------	----------------------------------

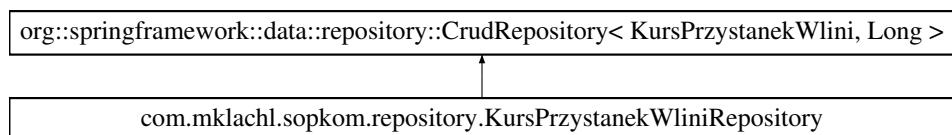
Definicja w linii 86 pliku [KursPrzystanekWlini.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.44 Dokumentacja interfejsu **com.mklachl.sopkom.repository.KursPrzystanekWliniRepository**

Diagram dziedziczenia dla com.mklachl.sopkom.repository.KursPrzystanekWliniRepository



Metody publiczne

- List< [KursPrzystanekWlini](#) > `findAllByKurs` ([Kurs](#) kurs)
- List< [KursPrzystanekWlini](#) > `findAllByPrzystanekWlini` ([PrzystanekWlini](#) przystanekWlini)
- List< [KursPrzystanekWlini](#) > `findAllByGodzinnaAfter` (Date date)
- List< [KursPrzystanekWlini](#) > `findAllByGodzinnaBefore` (Date date)
- List< [KursPrzystanekWlini](#) > `findAllByGodzinna` (Date date)
- List< [KursPrzystanekWlini](#) > `findAll` ()

6.44.1 Opis szczegółowy

Interfejs repository dla encji [KursPrzystanekWlini](#)

Definicja w linii 14 pliku [KursPrzystanekWliniRepository.java](#).

6.44.2 Dokumentacja funkcji składowych

6.44.2.1 findAll()

```
List< KursPrzystanekWlini > com.mklachl.sopkom.repository.KursPrzystanekWliniRepository.findAll( )
```

Znajduje wszystkie kursy przystanku

Zwrota

Lista kursów przystanku

6.44.2.2 findAllByGodzinna()

```
List< KursPrzystanekWlini > com.mklachl.sopkom.repository.KursPrzystanekWliniRepository.findAllByGodzinna( Date date )
```

Znajduje wszystkie kursy przystanku po dokładnej dacie

Parametry

<i>date</i>	data
-------------	------

Zwrota

Lista kursów przystanku

6.44.2.3 findAllByGodzinnaAfter()

```
List< KursPrzystanekWlini > com.mklachl.sopkom.repository.KursPrzystanekWliniRepository.findAllByGodzinnaAfter( Date date )
```

Znajduje wszystkie kursy przystanku po dacie późniejszej niż podana

Parametry

<i>date</i>	data
-------------	------

Zwrota

Lista kursów przystanku

6.44.2.4 findAllByGodzinnaBefore()

```
List< KursPrzystanekWlini > com.mklachl.sopkom.repository.KursPrzystanekWliniRepository.findAllByGodzinnaBefore ( Date date )
```

Znajduje wszystkie kursy przystanku po dacie wcześniejszej niż podana

Parametry

<i>date</i>	data
-------------	------

Zwroca

Lista kursów przystanku

6.44.2.5 findAllByKurs()

```
List< KursPrzystanekWlini > com.mklachl.sopkom.repository.KursPrzystanekWliniRepository.findAllByKurs ( Kurs kurs )
```

Znajduje wszystkie przystanki w kursie dla danego kursu

Parametry

<i>kurs</i>	kurs
-------------	------

Zwroca

Lista przystanków w kursie

6.44.2.6 findAllByPrzystanekWlini()

```
List< KursPrzystanekWlini > com.mklachl.sopkom.repository.KursPrzystanekWliniRepository.findAllByPrzystanekWlini ( PrzystanekWlini przystanekWlini )
```

Znajduje wszystkie kursy przystanku w danym przystanku w linii

Parametry

<i>przystanekWlini</i>	przystanek w linii
------------------------	--------------------

Zwrota

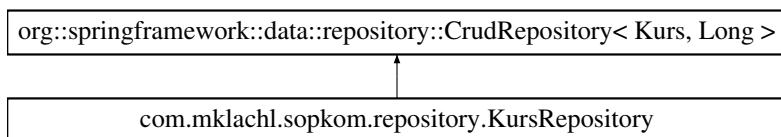
Lista kursów przystanku w linii

Dokumentacja dla tego interfejsu została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/repo

6.45 Dokumentacja interfejsu com.mklachl.sopkom.repository.KursRepository

Diagram dziedziczenia dla com.mklachl.sopkom.repository.KursRepository



Metody publiczne

- List< [Kurs](#) > [findAllByLinia](#) ([Linia](#) linia)
- List< [Kurs](#) > [findAllByHarmonogram](#) ([Harmonogram](#) harmonogram)
- List< [Kurs](#) > [findAllByKierunek](#) ([Short](#) kierunek)
- List< [Kurs](#) > [findAllByTypAutobusu](#) ([Short](#) typAutobusu)
- List< [Kurs](#) > [findAll](#) ()

6.45.1 Opis szczegółowy

Interfejs repository dla encji Kurs

Definicja w linii 13 pliku [KursRepository.java](#).

6.45.2 Dokumentacja funkcji składowych

6.45.2.1 findAll()

```
List< Kurs > com.mklachl.sopkom.repository.KursRepository.findAll ( )
```

Znajduje wszystkie kursy

Zwrota

Lista kursów

6.45.2.2 findAllByHarmonogram()

```
List< Kurs > com.mklachl.sopkom.repository.KursRepository.findAllByHarmonogram ( Harmonogram harmonogram )
```

Znajduje wszystkie kursy dla danego harmonogramu

Parametry

<i>harmonogram</i>	harmonogram
--------------------	-------------

Zwroca

Lista kursów

6.45.2.3 findAllByKierunek()

```
List< Kurs > com.mklachl.sopkom.repository.KursRepository.findAllByKierunek (  
    Short kierunek )
```

Znajduje wszystkie kursy o określonym kierunku

Parametry

<i>kierunek</i>	kierunek
-----------------	----------

Zwroca

Lista kursów

6.45.2.4 findAllByLinia()

```
List< Kurs > com.mklachl.sopkom.repository.KursRepository.findAllByLinia (   
    Linia linia )
```

Znajduje wszystkie kursy dla danej linii

Parametry

<i>linia</i>	linia
--------------	-------

Zwroca

Lista kursów

6.45.2.5 findAllByTypAutobusu()

```
List< Kurs > com.mklachl.sopkom.repository.KursRepository.findAllByTypAutobusu (   
    Short typAutobusu )
```

Znajduje wszystkie kursy o określonym typie autobusu

Parametry

<code>typAutobusu</code>	typ autobusu
--------------------------	--------------

Zwraca

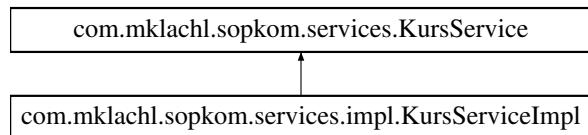
Lista kursów

Dokumentacja dla tego interfejsu została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/repo...

6.46 Dokumentacja interfejsu com.mklachl.sopkom.services.KursService

Diagram dziedziczenia dla com.mklachl.sopkom.services.KursService



Metody publiczne

- `Kurs saveKurs (Kurs kurs)`
- `Kurs saveKurs (InputKursDto kursDto) throws Exception`
- `Kurs updateKurs (InputKursDto kursDto) throws Exception`
- `List< Kurs > findKursByLinia (Linia linia)`
- `List< Kurs > findKursByHarmonogram (Harmonogram harmonogram)`
- `List< Kurs > findKursByKierunek (Short kierunek)`
- `List< Kurs > findKursByTypAutobusu (Short typAutobusu)`

6.46.1 Opis szczegółowy

Interfejs serwisu dla obsługi kursów

Definicja w linii 17 pliku [KursService.java](#).

6.46.2 Dokumentacja funkcji składowych

6.46.2.1 findKursByHarmonogram()

```
List< Kurs > com.mklachl.sopkom.services.KursService.findKursByHarmonogram (
    Harmonogram harmonogram )
```

Znajduje kursy po harmonogramie

Parametry

<i>harmonogram</i>	harmonogram
--------------------	-------------

Zwrota

lista kursów

Implementowany w [com.mklachl.sopkom.services.impl.KursServiceImpl](#).

6.46.2.2 findKursByKierunek()

```
List< Kurs > com.mklachl.sopkom.services.KursService.findKursByKierunek (  
    Short kierunek )
```

Znajduje kursy po kierunku

Parametry

<i>kierunek</i>	kierunek
-----------------	----------

Zwrota

lista kursów

Implementowany w [com.mklachl.sopkom.services.impl.KursServiceImpl](#).

6.46.2.3 findKursByLinia()

```
List< Kurs > com.mklachl.sopkom.services.KursService.findKursByLinia (   
    Linia linia )
```

Znajduje kursy po linii

Parametry

<i>linia</i>	linia
--------------	-------

Zwrota

lista kursów

Implementowany w [com.mklachl.sopkom.services.impl.KursServiceImpl](#).

6.46.2.4 findKursByTypAutobusu()

```
List< Kurs > com.mklachl.sopkom.services.KursService.findKursByTypAutobusu (  
    Short typAutobusu )
```

Znajduje kursy po typie autobusu

Parametry

<i>typAutobusu</i>	typ autobusu
--------------------	--------------

Zwroca

lista kursów

Implementowany w [com.mklachl.sopkom.services.impl.KursServiceImpl](#).

6.46.2.5 saveKurs() [1/2]

```
Kurs com.mklachl.sopkom.services.KursService.saveKurs (  
    InputKursDto kursDto ) throws Exception
```

Zapisuje nowy kurs na podstawie DTO

Parametry

<i>kursDto</i>	DTO kursu
----------------	-----------

Zwroca

zapisany kurs

Wyjątki

<i>Exception</i>	wyjątek ogólny
------------------	----------------

Implementowany w [com.mklachl.sopkom.services.impl.KursServiceImpl](#).

6.46.2.6 saveKurs() [2/2]

```
Kurs com.mklachl.sopkom.services.KursService.saveKurs (  
    Kurs kurs )
```

Zapisuje nowy kurs

Parametry

<i>kurs</i>	encja kurs
-------------	------------

Zwroca

zapisany kurs

Implementowany w [com.mklachl.sopkom.services.impl.KursServiceImpl](#).

6.46.2.7 updateKurs()

```
Kurs com.mklachl.sopkom.services.KursService.updateKurs (
    InputKursDto kursDto ) throws Exception
```

Aktualizuje istniejący kurs na podstawie DTO

Parametry

<i>kursDto</i>	DTO kursu
----------------	-----------

Zwroca

zaktualizowany kurs

Wyjątki

<i>Exception</i>	wyjątek ogólny
------------------	----------------

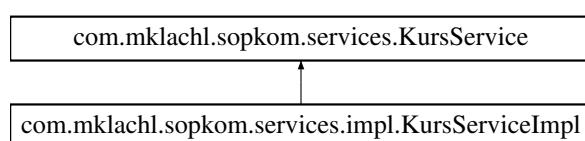
Implementowany w [com.mklachl.sopkom.services.impl.KursServiceImpl](#).

Dokumentacja dla tego interfejsu została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/se

6.47 Dokumentacja klasy com.mklachl.sopkom.services.impl.KursServiceImpl

Diagram dziedziczenia dla com.mklachl.sopkom.services.impl.KursServiceImpl



Metody publiczne

- KursServiceImpl (`KursRepository kursRepository, LiniaService liniaService, HarmonogramService harmonogramService, KursPrzystanekWliniRepository kpwlRepository, PrzystanekWliniRepository pwlRepository`)
- Kurs saveKurs (`Kurs kursDto`)
- Kurs saveKurs (`InputKursDto kursDto`) throws Exception
- Kurs updateKurs (`InputKursDto kursDto`) throws Exception
- List<`Kurs`> findKursByLinia (`Linia linia`)
- List<`Kurs`> findKursByHarmonogram (`Harmonogram harmonogram`)
- List<`Kurs`> findKursByKierunek (`Short kierunek`)
- List<`Kurs`> findKursByTypAutobusu (`Short typAutobusu`)

6.47.1 Opis szczegółowy

Definicja w linii 30 pliku `KursServiceImpl.java`.

6.47.2 Dokumentacja konstruktora i destruktora

6.47.2.1 KursServiceImpl()

```
com.mklachl.sopkom.services.impl.KursServiceImpl.KursServiceImpl (
    KursRepository kursRepository,
    LiniaService liniaService,
    HarmonogramService harmonogramService,
    KursPrzystanekWliniRepository kpwlRepository,
    PrzystanekWliniRepository pwlRepository )
```

Definicja w linii 43 pliku `KursServiceImpl.java`.

6.47.3 Dokumentacja funkcji składowych

6.47.3.1 findKursByHarmonogram()

```
List<Kurs> com.mklachl.sopkom.services.impl.KursServiceImpl.findKursByHarmonogram (
    Harmonogram harmonogram )
```

Znajduje kursy po harmonogramie

Parametry

<code>harmonogram</code>	harmonogram
--------------------------	-------------

Zwrota

lista kursów

Implementuje [com.mklachl.sopkom.services.KursService](#).

Definicja w linii 171 pliku [KursServiceImpl.java](#).

6.47.3.2 findKursByKierunek()

```
List< Kurs > com.mklachl.sopkom.services.impl.KursServiceImpl.findKursByKierunek (  
    Short kierunek )
```

Znajduje kursy po kierunku

Parametry

<i>kierunek</i>	<i>kierunek</i>
-----------------	-----------------

Zwrota

lista kursów

Implementuje [com.mklachl.sopkom.services.KursService](#).

Definicja w linii 174 pliku [KursServiceImpl.java](#).

6.47.3.3 findKursByLinia()

```
List< Kurs > com.mklachl.sopkom.services.impl.KursServiceImpl.findKursByLinia (   
    Linia linia )
```

Znajduje kursy po linii

Parametry

<i>linia</i>	<i>linia</i>
--------------	--------------

Zwrota

lista kursów

Implementuje [com.mklachl.sopkom.services.KursService](#).

Definicja w linii 168 pliku [KursServiceImpl.java](#).

6.47.3.4 findKursByTypAutobusu()

```
List< Kurs > com.mklachl.sopkom.services.impl.KursServiceImpl.findKursByTypAutobusu (  
    Short typAutobusu )
```

Znajduje kursy po typie autobusu

Parametry

<i>typAutobusu</i>	typ autobusu
--------------------	--------------

Zwrota

lista kursów

Implementuje [com.mklachl.sopkom.services.KursService](#).

Definicja w linii [177](#) pliku [KursServiceImpl.java](#).

6.47.3.5 saveKurs() [1/2]

```
Kurs com.mklachl.sopkom.services.impl.KursServiceImpl.saveKurs (  
    InputKursDto kursDto ) throws Exception
```

Zapisuje nowy kurs na podstawie DTO

Parametry

<i>kursDto</i>	DTO kursu
----------------	-----------

Zwrota

zapisany kurs

Wyjątki

<i>Exception</i>	wyjątek ogólny
------------------	----------------

Implementuje [com.mklachl.sopkom.services.KursService](#).

Definicja w linii [76](#) pliku [KursServiceImpl.java](#).

6.47.3.6 saveKurs() [2/2]

```
Kurs com.mklachl.sopkom.services.impl.KursServiceImpl.saveKurs (  
    Kurs kurs )
```

Zapisuje nowy kurs

Parametry

<i>kurs</i>	encja kurs
-------------	------------

Zwroca

zapisany kurs

Implementuje [com.mklachl.sopkom.services.KursService](#).

Definicja w linii 55 pliku [KursServiceImpl.java](#).

6.47.3.7 updateKurs()

```
Kurs com.mklachl.sopkom.services.impl.KursServiceImpl.updateKurs (
    InputKursDto kursDto ) throws Exception
```

Aktualizuje istniejący kurs na podstawie DTO

Parametry

<i>kursDto</i>	DTO kursu
----------------	-----------

Zwroca

zaktualizowany kurs

Wyjątki

<i>Exception</i>	wyjątek ogólny
------------------	----------------

Implementuje [com.mklachl.sopkom.services.KursService](#).

Definicja w linii 120 pliku [KursServiceImpl.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/se

6.48 Dokumentacja klasy com.mklachl.sopkom.model.entity.Linia

Metody publiczne

- boolean [odwracalna \(\)](#)
- Long [getLiniald \(\)](#)

- void [setLiniald](#) (Long liniald)
- String [getNumer](#) ()
- void [setNumer](#) (String numer)
- List< [PrzystanekWlini](#) > [getPrzystankiWlini](#) ()
- void [setPrzystankiWlini](#) (List< [PrzystanekWlini](#) > przystankiWlini)

6.48.1 Opis szczegółowy

Klasa reprezentująca encję [Linia](#).

Definicja w linii [11](#) pliku [Linia.java](#).

6.48.2 Dokumentacja funkcji składowych

6.48.2.1 [getLiniald\(\)](#)

```
Long com.mklachl.sopkom.model.entity.Linia.getLiniald ( )
```

Zwraca identyfikator linii.

Zwrota

liniald Identyfikator linii.

Definicja w linii [52](#) pliku [Linia.java](#).

6.48.2.2 [getNumer\(\)](#)

```
String com.mklachl.sopkom.model.entity.Linia.getNumer ( )
```

Zwraca numer linii.

Zwrota

numer Numer linii.

Definicja w linii [68](#) pliku [Linia.java](#).

6.48.2.3 getPrzystankiWlini()

```
List< PrzystanekWlini > com.mklachl.sopkom.model.entity.Linia.getPrzystankiWlini ( )
```

Zwraca listę przystanków w linii.

Zwrota

przystankiWlini Lista przystanków w linii.

Definicja w linii 84 pliku [Linia.java](#).

6.48.2.4 odwracalna()

```
boolean com.mklachl.sopkom.model.entity.Linia.odwracalna ( )
```

Sprawdza, czy linia jest odwracalna.

Zwrota

boolean Wartość true, jeśli linia jest odwracalna, w przeciwnym razie false.

Definicja w linii 38 pliku [Linia.java](#).

6.48.2.5 setLiniaId()

```
void com.mklachl.sopkom.model.entity.Linia.setLiniaId ( Long liniaId )
```

Ustawia identyfikator linii.

Parametry

<i>linia</i> ↵ <i>Id</i>	Identyfikator linii.
-----------------------------	----------------------

Definicja w linii 60 pliku [Linia.java](#).

6.48.2.6 setNumer()

```
void com.mklachl.sopkom.model.entity.Linia.setNumer ( String numer )
```

Ustawia numer linii.

Parametry

<i>numer</i>	Numer linii.
--------------	--------------

Definicja w linii [76](#) pliku [Linia.java](#).

6.48.2.7 setPrzystankiWlini()

```
void com.mklachl.sopkom.model.entity.Linia.setPrzystankiWlini (
    List< PrzystanekWlini > przystankiWlini )
```

Ustawia listę przystanków w linii.

Parametry

<i>przystankiWlini</i>	Lista przystanków w linii.
------------------------	----------------------------

Definicja w linii [92](#) pliku [Linia.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.49 Dokumentacja klasy com.mklachl.sopkom.controller.LiniaController

Metody publiczne

- ResponseEntity<?> [template](#) ()
- ResponseEntity<?> [templateInput](#) ()
- ResponseEntity<?> [addLinia](#) (@RequestBody [LiniaDtoInput](#) liniaDto)
- ResponseEntity<?> [getLinia](#) (@PathVariable("id") Long id)
- ResponseEntity<?> [deleteLinia](#) (@PathVariable("id") Long id)
- ResponseEntity<?> [updateLinia](#) (@PathVariable("id") Long id, @RequestBody [LiniaDtoInput](#) liniaDtoInput)
- ResponseEntity<?> [getAllPrzystanki](#) ()
- ResponseEntity<?> [getAllPrzystankiFull](#) ()
- ResponseEntity<?> [getPrzystanekFull](#) (@PathVariable("id") Long id)

6.49.1 Opis szczegółowy

Definicja w linii [25](#) pliku [LiniaController.java](#).

6.49.2 Dokumentacja funkcji składowych

6.49.2.1 addLinia()

```
ResponseEntity<?> com.mklachl.sopkom.controller.LiniaController.addLinia (   
    @RequestBody LiniaDtoInput liniaDto )
```

Endpoint do dodawania nowej linii.

Parametry

<i>liniaDto</i>	DTO linii do dodania
-----------------	----------------------

Zwrota

Dodany obiekt LiniaDtoOutput

Definicja w linii [100](#) pliku [LiniaController.java](#).

6.49.2.2 deleteLinia()

```
ResponseEntity<?> com.mklachl.sopkom.controller.LiniaController.deleteLinia ( @PathVariable("id") Long id )
```

Endpoint do usuwania linii według ID.

Parametry

<i>id</i>	ID linii
-----------	----------

Zwrota

Odpowiedź HTTP

Definicja w linii [135](#) pliku [LiniaController.java](#).

6.49.2.3 getAllPrzystanki()

```
ResponseEntity<?> com.mklachl.sopkom.controller.LiniaController.getAllPrzystanki ( )
```

Endpoint do pobierania wszystkich linii.

Zwrota

Lista wszystkich linii

Definicja w linii [169](#) pliku [LiniaController.java](#).

6.49.2.4 getAllPrzystankiFull()

```
ResponseEntity<?> com.mklachl.sopkom.controller.LiniaController.getAllPrzystankiFull ( )
```

Endpoint do pobierania wszystkich linii z pełnymi danymi.

Zwroca

Lista wszystkich linii z pełnymi danymi

Definicja w linii 183 pliku [LiniaController.java](#).

6.49.2.5 getLinia()

```
ResponseEntity<?> com.mklachl.sopkom.controller.LiniaController.getLinia ( @PathVariable("id") Long id )
```

Endpoint do pobierania linii według ID.

Parametry

<i>id</i>	ID linii
-----------	----------

Zwroca

Obiekt LiniaDtoOutput

Definicja w linii 119 pliku [LiniaController.java](#).

6.49.2.6 getPrzystanekFull()

```
ResponseEntity<?> com.mklachl.sopkom.controller.LiniaController.getPrzystanekFull ( @PathVariable("id") Long id )
```

Endpoint do pobierania pełnych danych linii według ID.

Parametry

<i>id</i>	ID linii
-----------	----------

Zwroca

Obiekt LiniaDtoInput

Definicja w linii 198 pliku [LiniaController.java](#).

6.49.2.7 template()

```
ResponseEntity<?> com.mklachl.sopkom.controller.LiniaController.template ( )
```

Endpoint zwracający szablonowy obiekt LiniaDtoOutput.

Zwroca

Szablonowy obiekt LiniaDtoOutput

Definicja w linii [37](#) pliku [LiniaController.java](#).

6.49.2.8 templateInput()

```
ResponseEntity<?> com.mklachl.sopkom.controller.LiniaController.templateInput ( )
```

Endpoint zwracający szablonowy obiekt LiniaDtoInput.

Zwroca

Szablonowy obiekt LiniaDtoInput

Definicja w linii [70](#) pliku [LiniaController.java](#).

6.49.2.9 updateLinia()

```
ResponseEntity<?> com.mklachl.sopkom.controller.LiniaController.updateLinia ( @PathVariable("id") Long id, @RequestBody LiniaDtoInput liniaDtoInput )
```

Endpoint do aktualizacji linii.

Parametry

<i>id</i>	ID linii
<i>liniaDtoInput</i>	DTO linii do aktualizacji

Zwroca

Zaktualizowany obiekt LiniaDtoOutput

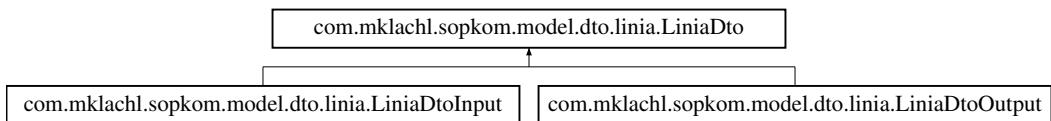
Definicja w linii [150](#) pliku [LiniaController.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/

6.50 Dokumentacja klasy com.mklachl.sopkom.model.dto.linia.LiniaDto

Diagram dziedziczenia dla com.mklachl.sopkom.model.dto.linia.LiniaDto



Metody publiczne

- `LiniaDto ()`
- `LiniaDto (Linia linia)`
- `Long getId ()`
- `void setId (Long id)`
- `String getNumer ()`
- `void setNumer (String numer)`

Atrybuty chronione

- `Long id`
- `String numer`

6.50.1 Opis szczegółowy

Data Transfer Object (DTO) dla encji Linia.

Definicja w linii 8 pliku [LiniaDto.java](#).

6.50.2 Dokumentacja konstruktora i destruktora

6.50.2.1 LiniaDto() [1/2]

```
com.mklachl.sopkom.model.dto.linia.LiniaDto.LiniaDto ( )
```

Konstruktor domyślny.

Definicja w linii 23 pliku [LiniaDto.java](#).

6.50.2.2 LiniaDto() [2/2]

```
com.mklachl.sopkom.model.dto.linia.LiniaDto.LiniaDto (
    Linia linia )
```

Konstruktor przyjmujący obiekt encji Linia.

Parametry

<i>linia</i>	Obiekt encji Linia.
--------------	---------------------

Definicja w linii [30](#) pliku [LiniaDto.java](#).

6.50.3 Dokumentacja funkcji składowych

6.50.3.1 getId()

```
Long com.mklachl.sopkom.model.dto.linia.LiniaDto.getId ( )
```

Zwroca identyfikator linii.

Zwroca

id Identyfikator linii.

Definicja w linii [41](#) pliku [LiniaDto.java](#).

6.50.3.2 getNumer()

```
String com.mklachl.sopkom.model.dto.linia.LiniaDto.getNumer ( )
```

Zwroca numer linii.

Zwroca

numer Numer linii.

Definicja w linii [57](#) pliku [LiniaDto.java](#).

6.50.3.3 setId()

```
void com.mklachl.sopkom.model.dto.linia.LiniaDto.setId ( Long id )
```

Ustawia identyfikator linii.

Parametry

<i>id</i>	Identyfikator linii.
-----------	----------------------

Definicja w linii [49](#) pliku [LiniaDto.java](#).

6.50.3.4 setNumer()

```
void com.mklachl.sopkom.model.dto.linia.LiniaDto.setNumer (
    String numer )
```

Ustawia numer linii.

Parametry

<i>numer</i>	Numer linii.
--------------	--------------

Definicja w linii [65](#) pliku [LiniaDto.java](#).

6.50.4 Dokumentacja atrybutów składowych

6.50.4.1 id

```
Long com.mklachl.sopkom.model.dto.linia.LiniaDto.id [protected]
```

Identyfikator linii.

Definicja w linii [13](#) pliku [LiniaDto.java](#).

6.50.4.2 numer

```
String com.mklachl.sopkom.model.dto.linia.LiniaDto.numer [protected]
```

Numer linii.

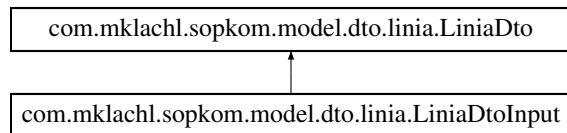
Definicja w linii [18](#) pliku [LiniaDto.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.51 Dokumentacja klasy com.mklachl.sopkom.model.dto.linia.LiniaDtoInput

Diagram dziedziczenia dla com.mklachl.sopkom.model.dto.linia.LiniaDtoInput



Metody publiczne

- [LiniaDtoInput \(\)](#)
- [LiniaDtoInput \(Linia linia\)](#)
- [List< PrzystanekDtoDlaLinialInput > getPrzystanki \(\)](#)
- [void setPrzystanki \(List< PrzystanekDtoDlaLinialInput > przystanki\)](#)

Dodatkowe Dziedziczone Składowe

6.51.1 Opis szczegółowy

Data Transfer Object (DTO) wejściowy dla encji Linia.

Definicja w linii 10 pliku [LiniaDtoInput.java](#).

6.51.2 Dokumentacja konstruktora i destruktora

6.51.2.1 LiniaDtoInput() [1/2]

```
com.mklachl.sopkom.model.dto.linia.LiniaDtoInput.LiniaDtoInput ( )
```

Konstruktor domyślny.

Definicja w linii 20 pliku [LiniaDtoInput.java](#).

6.51.2.2 LiniaDtoInput() [2/2]

```
com.mklachl.sopkom.model.dto.linia.LiniaDtoInput.LiniaDtoInput (
    Linia linia )
```

Konstruktor przyjmujący obiekt encji Linia.

Parametry

<i>linia</i>	Obiekt encji Linia.
--------------	---------------------

Definicja w linii [27](#) pliku [LiniaDtoInput.java](#).

6.51.3 Dokumentacja funkcji składowych

6.51.3.1 getPrzystanki()

```
List< PrzystanekDtoDlaLiniaInput > com.mklachl.sopkom.model.dto.linia.LiniaDtoInput.get←
Przystanki ( )
```

Zwraca listę przystanków związanych z linią.

Zwraca

przystanki Lista przystanków związanych z linią.

Definicja w linii [43](#) pliku [LiniaDtoInput.java](#).

6.51.3.2 setPrzystanki()

```
void com.mklachl.sopkom.model.dto.linia.LiniaDtoInput.setPrzystanki (
    List< PrzystanekDtoDlaLiniaInput > przystanki )
```

Ustawia listę przystanków związanych z linią.

Parametry

<i>przystanki</i>	Lista przystanków związanych z linią.
-------------------	---------------------------------------

Definicja w linii [51](#) pliku [LiniaDtoInput.java](#).

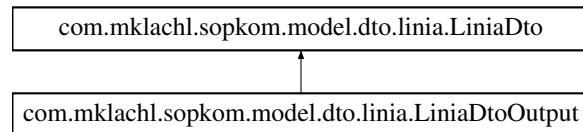
Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-Incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.52 Dokumentacja klasy

com.mklachl.sopkom.model.dto.linia.LiniaDtoOutput

Diagram dziedziczenia dla com.mklachl.sopkom.model.dto.linia.LiniaDtoOutput



Metody publiczne

- [LiniaDtoOutput \(\)](#)
- [LiniaDtoOutput \(Linia linia\)](#)
- boolean [isOdwracalna \(\)](#)
- void [setOdwracalna \(boolean odwracalna\)](#)
- List<[PrzystanekDtoDlaLinia](#)> [getPrzystanki \(\)](#)
- void [setPrzystanki \(List<\[PrzystanekDtoDlaLinia\]\(#\)> przystanki\)](#)

Atrybuty chronione

- boolean [odwracalna](#)

6.52.1 Opis szczegółowy

Data Transfer Object (DTO) wyjściowy dla encji Linia.

Definicja w linii 10 pliku [LiniaDtoOutput.java](#).

6.52.2 Dokumentacja konstruktora i destruktora

6.52.2.1 LiniaDtoOutput() [1/2]

```
com.mklachl.sopkom.model.dto.linia.LiniaDtoOutput.LiniaDtoOutput ( )
```

Konstruktor domyślny.

Definicja w linii 25 pliku [LiniaDtoOutput.java](#).

6.52.2.2 LiniaDtoOutput() [2/2]

```
com.mklachl.sopkom.model.dto.linia.LiniaDtoOutput.LiniaDtoOutput (
    Linia linia )
```

Konstruktor przyjmujący obiekt encji Linia.

Parametry

<i>linia</i>	Obiekt encji Linia.
--------------	---------------------

Definicja w linii [32](#) pliku [LiniaDtoOutput.java](#).

6.52.3 Dokumentacja funkcji składowych

6.52.3.1 getPrzystanki()

```
List< PrzystanekDtoDlaLinia > com.mklachl.sopkom.model.dto.linia.LiniaDtoOutput.getPrzystanki( )
```

Zwraca listę przystanków związanych z linią.

Zwrota

przystanki Lista przystanków związanych z linią.

Definicja w linii [69](#) pliku [LiniaDtoOutput.java](#).

6.52.3.2 isOdwracalna()

```
boolean com.mklachl.sopkom.model.dto.linia.LiniaDtoOutput.isOdwracalna( )
```

Zwraca wartość flagi określającej, czy linia jest odwracalna.

Zwrota

odwracalna Wartość flagi odwracalności linii.

Definicja w linii [53](#) pliku [LiniaDtoOutput.java](#).

6.52.3.3 setOdwracalna()

```
void com.mklachl.sopkom.model.dto.linia.LiniaDtoOutput.setOdwracalna( boolean odwracalna )
```

Ustawia wartość flagi określającej, czy linia jest odwracalna.

Parametry

<i>odwracalna</i>	Wartość flagi odwracalności linii.
-------------------	------------------------------------

Definicja w linii 61 pliku [LiniaDtoOutput.java](#).

6.52.3.4 setPrzystanki()

```
void com.mklachl.sopkom.model.dto.linia.LiniaDtoOutput.setPrzystanki (List< PrzystanekDtoDlaLinia > przystanki )
```

Ustawia listę przystanków związanych z linią.

Parametry

<i>przystanki</i>	Lista przystanków związanych z linią.
-------------------	---------------------------------------

Definicja w linii 77 pliku [LiniaDtoOutput.java](#).

6.52.4 Dokumentacja atrybutów składowych

6.52.4.1 odwracalna

```
boolean com.mklachl.sopkom.model.dto.linia.LiniaDtoOutput.odwracalna [protected]
```

Flaga określająca, czy linia jest odwracalna.

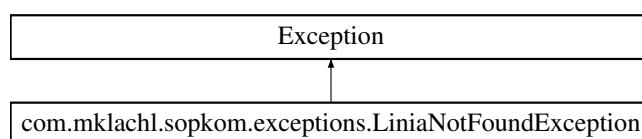
Definicja w linii 15 pliku [LiniaDtoOutput.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.53 Dokumentacja klasy com.mklachl.sopkom.exceptions.LiniaNotFoundException

Diagram dziedziczenia dla com.mklachl.sopkom.exceptions.LiniaNotFoundException



Metody publiczne

- [LiniaNotFoundException](#) (Long id)

6.53.1 Opis szczegółowy

Definicja w linii 3 pliku [LiniaNotFoundException.java](#).

6.53.2 Dokumentacja konstruktora i destruktora

6.53.2.1 LiniaNotFoundException()

```
com.mklachl.sopkom.exceptions.LiniaNotFoundException.LiniaNotFoundException (
    Long id )
```

Definicja w linii 5 pliku [LiniaNotFoundException.java](#).

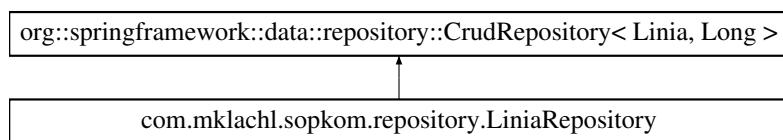
Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/exc

6.54 Dokumentacja interfejsu

com.mklachl.sopkom.repository.LiniaRepository

Diagram dziedziczenia dla com.mklachl.sopkom.repository.LiniaRepository



Metody publiczne

- Optional< [Linia](#) > [findById](#) (Long id)
- [Linia](#) [findByNumer](#) (String numer)
- List< [Linia](#) > [findAll](#) ()

6.54.1 Opis szczegółowy

Interfejs repository dla encji Linia

Definicja w linii 12 pliku [LiniaRepository.java](#).

6.54.2 Dokumentacja funkcji składowych

6.54.2.1 findAll()

```
List< Linia > com.mklachl.sopkom.repository.LiniaRepository.findAll ( )
```

Znajduje wszystkie linie

Zwroca

Lista linii

6.54.2.2 findById()

```
Optional< Linia > com.mklachl.sopkom.repository.LiniaRepository.findById ( Long id )
```

Znajduje linię po identyfikatorze

Parametry

<i>id</i>	identyfikator
-----------	---------------

Zwroca

Optional z linią

6.54.2.3 findByNumer()

```
Linia com.mklachl.sopkom.repository.LiniaRepository.findByNumer ( String numer )
```

Znajduje linię po numerze

Parametry

<i>numer</i>	numer
--------------	-------

Zwroca

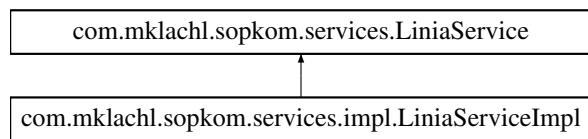
Linia

Dokumentacja dla tego interfejsu została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/rep...

6.55 Dokumentacja interfejsu **com.mklachl.sopkom.services.LiniaService**

Diagram dziedziczenia dla com.mklachl.sopkom.services.LiniaService



Metody publiczne

- [Linia saveLinia \(LiniaDtoInput liniaDto\) throws PrzystanekNotFoundException](#)
- [Linia patchLinia \(LiniaDtoInput liniaDto\) throws PrzystanekNotFoundException, LiniaNotFoundException](#)
- void [deleteLinia \(Long id\)](#)
- Optional< [Linia > findLiniaById \(Long id\)](#)
- [Linia findLiniaByNumer \(String numer\)](#)

6.55.1 Opis szczegółowy

Interfejs serwisu dla obsługi linii

Definicja w linii [14](#) pliku [LiniaService.java](#).

6.55.2 Dokumentacja funkcji składowych

6.55.2.1 deleteLinia()

```
void com.mklachl.sopkom.services.LiniaService.deleteLinia ( Long id )
```

Usuwa linię na podstawie ID

Parametry

<i>id</i>	ID linii
-----------	----------

Implementowany w [com.mklachl.sopkom.services.impl.LiniaServiceImpl](#).

6.55.2.2 findLiniaById()

```
Optional< Linia > com.mklachl.sopkom.services.LiniaService.findLiniaById ( Long id )
```

Znajduje linię po ID

Parametry

<i>id</i>	ID linii
-----------	----------

Zwroca

Optional z linią

Implementowany w [com.mklachl.sopkom.services.impl.LiniaServiceImpl](#).

6.55.2.3 findLiniaByNumer()

```
Linia com.mklachl.sopkom.services.LiniaService.findLiniaByNumer ( String numer )
```

Znajduje linię po numerze

Parametry

<i>numer</i>	numer linii
--------------	-------------

Zwroca

znaleziona linia

Implementowany w [com.mklachl.sopkom.services.impl.LiniaServiceImpl](#).

6.55.2.4 patchLinia()

```
Linia com.mklachl.sopkom.services.LiniaService.patchLinia ( LiniaDtoInput liniaDto ) throws PrzystanekNotFoundException, LiniaNotFoundException
```

Aktualizuje istniejącą linię na podstawie DTO

Parametry

<i>liniaDto</i>	DTO linii
-----------------	-----------

Zwrota

zaktualizowana linia

Wyjątki

<i>PrzystanekNotFoundException</i>	wyjątek braku przystanku
<i>LiniaNotFoundException</i>	wyjątek braku linii

Implementowany w [com.mklachl.sopkom.services.impl.LiniaServiceImpl](#).

6.55.2.5 saveLinia()

```
Linia com.mklachl.sopkom.services.LiniaService.saveLinia (
    LiniaDtoInput liniaDto ) throws PrzystanekNotFoundException
```

Zapisuje nową linię na podstawie DTO

Parametry

<i>liniaDto</i>	DTO linii
-----------------	-----------

Zwrota

zapisana linia

Wyjątki

<i>PrzystanekNotFoundException</i>	wyjątek braku przystanku
------------------------------------	--------------------------

Implementowany w [com.mklachl.sopkom.services.impl.LiniaServiceImpl](#).

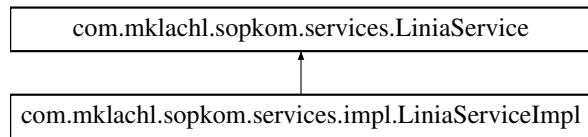
Dokumentacja dla tego interfejsu została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/ser...

6.56 Dokumentacja klasy

com.mklachl.sopkom.services.impl.LiniaServiceImpl

Diagram dziedziczenia dla com.mklachl.sopkom.services.impl.LiniaServiceImpl



Metody publiczne

- [Linia saveLinia](#) ([LiniaDtoInput](#) liniaDto) throws PrzystanekNotFoundException
- [Linia patchLinia](#) ([LiniaDtoInput](#) liniaDto) throws PrzystanekNotFoundException, LiniaNotFoundException
- void [deleteLinia](#) (Long id)
- Optional<[Linia](#)> [findLiniaById](#) (Long id)
- [Linia findLiniaByNumer](#) (String numer)

6.56.1 Opis szczegółowy

Definicja w linii 27 pliku [LiniaServiceImpl.java](#).

6.56.2 Dokumentacja funkcji składowych

6.56.2.1 deleteLinia()

```
void com.mklachl.sopkom.services.impl.LiniaServiceImpl.deleteLinia (
    Long id )
```

Usuwa linię na podstawie ID

Parametry

<i>id</i>	ID linii
-----------	----------

Implementuje [com.mklachl.sopkom.services.LiniaService](#).

Definicja w linii 140 pliku [LiniaServiceImpl.java](#).

6.56.2.2 findLiniaById()

```
Optional<Linia> com.mklachl.sopkom.services.impl.LiniaServiceImpl.findLiniaById (
    Long id )
```

Znajduje linię po ID

Parametry

<i>id</i>	ID linii
-----------	----------

Zwraca

Optional z linią

Implementuje [com.mklachl.sopkom.services.LiniaService](#).

Definicja w linii [145](#) pliku [LiniaServiceImpl.java](#).

6.56.2.3 findLiniaByNumer()

```
Linia com.mklachl.sopkom.services.impl.LiniaServiceImpl.findLiniaByNumer (
    String numer )
```

Znajduje linię po numerze

Parametry

<i>numer</i>	numer linii
--------------	-------------

Zwraca

znaleziona linia

Implementuje [com.mklachl.sopkom.services.LiniaService](#).

Definicja w linii [150](#) pliku [LiniaServiceImpl.java](#).

6.56.2.4 patchLinia()

```
Linia com.mklachl.sopkom.services.impl.LiniaServiceImpl.patchLinia (
    LiniaDtoInput liniaDto ) throws PrzystanekNotFoundException, LiniaNotFoundException
```

Aktualizuje istniejącą linię na podstawie DTO

Parametry

<i>liniaDto</i>	DTO linii
-----------------	-----------

Zwraca

zaktualizowana linia

Wyjątki

<i>PrzystanekNotFoundException</i>	wyjątek braku przystanku
<i>LiniaNotFoundException</i>	wyjątek braku linii

Implementuje [com.mklachl.sopkom.services.LiniaService](#).

Definicja w linii [88](#) pliku [LiniaServiceImpl.java](#).

6.56.2.5 saveLinia()

```
Linia com.mklachl.sopkom.services.impl.LiniaServiceImpl.saveLinia (
    LiniaDtoInput liniaDto ) throws PrzystanekNotFoundException
```

Zapisuje nową linię na podstawie DTO

Parametry

<i>liniaDto</i>	DTO linii
-----------------	-----------

Zwraca

zapisana linia

Wyjątki

<i>PrzystanekNotFoundException</i>	wyjątek braku przystanku
------------------------------------	--------------------------

Implementuje [com.mklachl.sopkom.services.LiniaService](#).

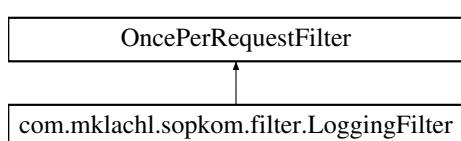
Definicja w linii [45](#) pliku [LiniaServiceImpl.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/ser...

6.57 Dokumentacja klasy com.mklachl.sopkom.filter.LoggingFilter

Diagram dziedziczenia dla com.mklachl.sopkom.filter.LoggingFilter



Metody chronione

- void [doFilterInternal](#) (HttpServletRequest request, HttpServletResponse response, FilterChain filterChain) throws ServletException, IOException

6.57.1 Opis szczegółowy

Definicja w linii 14 pliku [LoggingFilter.java](#).

6.57.2 Dokumentacja funkcji składowych

6.57.2.1 doFilterInternal()

```
void com.mklachl.sopkom.filter.LoggingFilter.doFilterInternal (
    HttpServletRequest request,
    HttpServletResponse response,
    FilterChain filterChain ) throws ServletException, IOException [protected]
```

Definicja w linii 16 pliku [LoggingFilter.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/filter/LoggingFilter.java

6.58 Dokumentacja klasy

com.mklachl.sopkom.controller.LoginController

Metody publiczne

- String [getLoginPage](#) ()
- String [getLogoutPage](#) ()

6.58.1 Opis szczegółowy

Definicja w linii 7 pliku [LoginController.java](#).

6.58.2 Dokumentacja funkcji składowych

6.58.2.1 getLoginPage()

```
String com.mklachl.sopkom.controller.LoginController.getLoginPage ( )
```

Endpoint do wyświetlania strony logowania.

Zwroca

Nazwa widoku strony logowania

Definicja w linii 14 pliku [LoginController.java](#).

6.58.2.2 getLogoutPage()

```
String com.mklachl.sopkom.controller.LoginController.getLogoutPage ( )
```

Endpoint do wyświetlania strony głównej po wylogowaniu.

Zwroca

Nazwa widoku strony głównej

Definicja w linii 23 pliku [LoginController.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/com

6.59 Dokumentacja klasy com.mklachl.sopkom.model.dto.Message

Metody publiczne

- [Message \(\)](#)
- [Message \(String message\)](#)
- [Message SetMessageInt \(String message\)](#)

Statyczne metody publiczne

- static [Message SetMessage \(String message\)](#)

Atrybuty publiczne

- String [message](#)

6.59.1 Opis szczegółowy

Klasa reprezentująca wiadomość.

Definicja w linii 6 pliku [Message.java](#).

6.59.2 Dokumentacja konstruktora i destruktora

6.59.2.1 Message() [1/2]

```
com.mklachl.sopkom.model.dto.Message.Message ( )
```

Konstruktor domyślny.

Definicja w linii 16 pliku [Message.java](#).

6.59.2.2 Message() [2/2]

```
com.mklachl.sopkom.model.dto.Message.Message (
    String message )
```

Konstruktor przyjmujący treść wiadomości.

Parametry

<i>message</i>	Treść wiadomości.
----------------	-------------------

Definicja w linii 23 pliku [Message.java](#).

6.59.3 Dokumentacja funkcji składowych

6.59.3.1 SetMessage()

```
static Message com.mklachl.sopkom.model.dto.Message.SetMessage (
    String message ) [static]
```

Ustawia treść wiadomości i zwraca nowy obiekt [Message](#).

Parametry

<i>message</i>	Treść wiadomości.
----------------	-------------------

Zwroca

Nowy obiekt [Message](#).

Definicja w linii [42](#) pliku [Message.java](#).

6.59.3.2 SetMessageInt()

```
Message com.mklachl.sopkom.model.dto.Message.SetMessageInt (
    String message )
```

Ustawia treść wiadomości i zwraca obiekt [Message](#).

Parametry

<i>message</i>	Treść wiadomości.
----------------	-------------------

Zwroca

Obiekt [Message](#).

Definicja w linii [32](#) pliku [Message.java](#).

6.59.4 Dokumentacja atrybutów składowych**6.59.4.1 message**

```
String com.mklachl.sopkom.model.dto.Message.message
```

Treść wiadomości.

Definicja w linii [11](#) pliku [Message.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.60 Dokumentacja klasy com.mklachl.sopkom.topdf.Pdf

Komponenty

- class \$

Metody publiczne

- TreeSet< \$ > [getAllByPrzystanekId](#) (Long id)
- int [getLiczbaLiniiNaPrzystanku](#) (Long id)
- File [genRozklad](#) (Long id) throws Exception
- void [genruj](#) (Long id) throws Exception

Atrybuty publiczne

- KursRepository kursRepository
- PrzystanekRepository przystanekRepos
- PrzystanekWliniRepository przystanekWliniRepos
- LiniaRepository liniaRepos

6.60.1 Opis szczegółowy

Definicja w linii [31](#) pliku Pdf.java.

6.60.2 Dokumentacja funkcji składowych

6.60.2.1 [genRozklad\(\)](#)

```
File com.mklachl.sopkom.topdf.Pdf.genRozklad (
    Long id ) throws Exception
```

Definicja w linii [238](#) pliku Pdf.java.

6.60.2.2 [genruj\(\)](#)

```
void com.mklachl.sopkom.topdf.Pdf.genruj (
    Long id ) throws Exception
```

Definicja w linii [304](#) pliku Pdf.java.

6.60.2.3 getAllByPrzystanekId()

```
TreeSet< $ > com.mklachl.sopkom.topdf.Pdf.getAllByPrzystanekId (  
    Long id )
```

Definicja w linii [207](#) pliku Pdf.java.

6.60.2.4 getLiczbaLiniiNaPrzystanku()

```
int com.mklachl.sopkom.topdf.Pdf.getLiczbaLiniiNaPrzystanku (   
    Long id )
```

Definicja w linii [225](#) pliku Pdf.java.

6.60.3 Dokumentacja atrybutów składowych

6.60.3.1 kursRepository

```
KursRepository com.mklachl.sopkom.topdf.Pdf.kursRepository
```

Definicja w linii [195](#) pliku Pdf.java.

6.60.3.2 liniaRepos

```
LiniaRepository com.mklachl.sopkom.topdf.Pdf.liniaRepos
```

Definicja w linii [204](#) pliku Pdf.java.

6.60.3.3 przystanekRepos

```
PrzystanekRepository com.mklachl.sopkom.topdf.Pdf.przystanekRepos
```

Definicja w linii [198](#) pliku Pdf.java.

6.60.3.4 przystanekWliniRepos

`PrzystanekWliniRepository com.mklachl.sopkom.topdf.Pdf.przystanekWliniRepos`

Definicja w linii 201 pliku [Pdf.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/top

6.61 Dokumentacja klasy

com.mklachl.sopkom.model.dto.przejazd.PKPWLDtoUpdate

Metody publiczne

- `PKPWLDtoUpdate()`
- `Long getId()`
- `void setId(Long id)`
- `Date getDate()`
- `void setDate(Date date)`

6.61.1 Opis szczegółowy

Data Transfer Object (DTO) dla aktualizacji encji PrzejazdKursPrzystanekWLini.

Definicja w linii 9 pliku [PKPWLDtoUpdate.java](#).

6.61.2 Dokumentacja konstruktora i destruktora

6.61.2.1 PKPWLDtoUpdate()

`com.mklachl.sopkom.model.dto.przejazd.PKPWLDtoUpdate.PKPWLDtoUpdate()`

Konstruktor domyślny.

Definicja w linii 25 pliku [PKPWLDtoUpdate.java](#).

6.61.3 Dokumentacja funkcji składowych

6.61.3.1 getDate()

```
Date com.mklachl.sopkom.model.dto.przejazd.PKPWLDtoUpdate.getDate()
```

Zwraca datę.

Zwraca

date Data.

Definicja w linii 50 pliku [PKPWLDtoUpdate.java](#).

6.61.3.2 getId()

```
Long com.mklachl.sopkom.model.dto.przejazd.PKPWLDtoUpdate.getId()
```

Zwraca identyfikator encji.

Zwraca

id Identyfikator encji.

Definicja w linii 34 pliku [PKPWLDtoUpdate.java](#).

6.61.3.3 setDate()

```
void com.mklachl.sopkom.model.dto.przejazd.PKPWLDtoUpdate.setDate(
    Date date)
```

Ustawia datę.

Parametry

<i>date</i>	Data.
-------------	-------

Definicja w linii 58 pliku [PKPWLDtoUpdate.java](#).

6.61.3.4 setId()

```
void com.mklachl.sopkom.model.dto.przejazd.PKPWLDtoUpdate.setId(
    Long id)
```

Ustawia identyfikator encji.

Parametry

<i>id</i>	Identyfikator encji.
-----------	----------------------

Definicja w linii 42 pliku [PKPWLDtoUpdate.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo...

6.62 Dokumentacja klasy com.mklachl.sopkom.model.entity.Przejazd

Metody publiczne

- Long [getPrzejazdId](#) ()
- void [setPrzejazdId](#) (Long przejazdId)
- [Kierowca getKierowca](#) ()
- void [setKierowca](#) ([Kierowca](#) kierowca)
- [Autobus getAutobus](#) ()
- void [setAutobus](#) ([Autobus](#) autobus)
- [Kurs getKurs](#) ()
- void [setKurs](#) ([Kurs](#) kurs)
- float [getSpalanie](#) ()
- void [setSpalanie](#) (float spalanie)
- float [getCenaZaLitr](#) ()
- void [setCenaZaLitr](#) (float cenaZaLitr)
- float [getDlugoscTrasy](#) ()
- void [setDlugoscTrasy](#) (float dlugoscTrasy)
- Date [getDataStartu](#) ()
- void [setDataStartu](#) (Date dataStartu)
- Date [getDataKonca](#) ()
- void [setDataKonca](#) (Date dataKonca)
- Date [getData](#) ()
- void [setData](#) (Date data)
- List< [PrzejazdKursPrzystanekWlini](#) > [getPrzejazdKursPrzystanekWlini](#) ()
- void [setPrzejazdKursPrzystanekWlini](#) (List< [PrzejazdKursPrzystanekWlini](#) > przejazdKursPrzystanekWlini)
- List< [PrzejazdBilet](#) > [getPrzejazdBilet](#) ()
- void [setPrzejazdBilet](#) (List< [PrzejazdBilet](#) > PrzejazdBilet)

6.62.1 Opis szczegółowy

Klasa reprezentująca encję [Przejazd](#).

Definicja w linii 12 pliku [Przejazd.java](#).

6.62.2 Dokumentacja funkcji składowych

6.62.2.1 getAutobus()

```
Autobus com.mklachl.sopkom.model.entity.Przejazd.getAutobus ( )
```

Zwraca autobus używany w przejeździe.

Zwraca

autobus [Autobus](#) używany w przejeździe.

Definicja w linii [128](#) pliku [Przejazd.java](#).

6.62.2.2 getCenaZaLitru()

```
float com.mklachl.sopkom.model.entity.Przejazd.getCenaZaLitru ( )
```

Zwraca cenę za litr paliwa podczas przejazdu.

Zwraca

cenaZaLitru Cena za litr paliwa podczas przejazdu.

Definicja w linii [176](#) pliku [Przejazd.java](#).

6.62.2.3 getData()

```
Date com.mklachl.sopkom.model.entity.Przejazd.getData ( )
```

Zwraca datę przejazdu.

Zwraca

data Data przejazdu.

Definicja w linii [240](#) pliku [Przejazd.java](#).

6.62.2.4 getDataKonca()

```
Date com.mklachl.sopkom.model.entity.Przejazd.getDataKonca ( )
```

Zwraca datę zakończenia przejazdu.

Zwraca

dataKonca Data zakończenia przejazdu.

Definicja w linii [224](#) pliku [Przejazd.java](#).

6.62.2.5 getDataStartu()

```
Date com.mklachl.sopkom.model.entity.Przejazd.getDataStartu ( )
```

Zwraca datę rozpoczęcia przejazdu.

Zwraca

dataStartu Data rozpoczęcia przejazdu.

Definicja w linii 208 pliku [Przejazd.java](#).

6.62.2.6 getDlugoscTrasy()

```
float com.mklachl.sopkom.model.entity.Przejazd.getDlugoscTrasy ( )
```

Zwraca długość trasy przejazdu.

Zwraca

dlugoscTrasy Długość trasy przejazdu.

Definicja w linii 192 pliku [Przejazd.java](#).

6.62.2.7 getKierowca()

```
Kierowca com.mklachl.sopkom.model.entity.Przejazd.getKierowca ( )
```

Zwraca kierowcę związanego z przejazdem.

Zwraca

kierowca [Kierowca](#) związany z przejazdem.

Definicja w linii 112 pliku [Przejazd.java](#).

6.62.2.8 getKurs()

```
Kurs com.mklachl.sopkom.model.entity.Przejazd.getKurs ( )
```

Zwraca kurs związany z przejazdem.

Zwraca

kurs [Kurs](#) związany z przejazdem.

Definicja w linii 144 pliku [Przejazd.java](#).

6.62.2.9 getPrzejazdBilet()

```
List< PrzejazdBilet > com.mklachl.sopkom.model.entity.Przejazd.getPrzejazdBilet ( )
```

Zwraca listę biletów związanych z przejazdem.

Zwrota

PrzejazdBilet Lista biletów związanych z przejazdem.

Definicja w linii 272 pliku [Przejazd.java](#).

6.62.2.10 getPrzejazdId()

```
Long com.mklachl.sopkom.model.entity.Przejazd.getPrzejazdId ( )
```

Zwraca identyfikator przejazdu.

Zwrota

przejazdId Identyfikator przejazdu.

Definicja w linii 96 pliku [Przejazd.java](#).

6.62.2.11 getPrzejazdKursPrzystanekWlini()

```
List< PrzejazdKursPrzystanekWlini > com.mklachl.sopkom.model.entity.Przejazd.getPrzejazdKurs←  
PrzystanekWlini ( )
```

Zwraca listę przystanków w linii związanych z przejazdem.

Zwrota

przejazdKursPrzystanekWlini Lista przystanków w linii związanych z przejazdem.

Definicja w linii 256 pliku [Przejazd.java](#).

6.62.2.12 getSpalanie()

```
float com.mklachl.sopkom.model.entity.Przejazd.getSpalanie ( )
```

Zwraca spalanie autobusu podczas przejazdu.

Zwrota

spalanie Spalanie autobusu podczas przejazdu.

Definicja w linii 160 pliku [Przejazd.java](#).

6.62.2.13 setAutobus()

```
void com.mklachl.sopkom.model.entity.Przejazd.setAutobus (←  
    Autobus autobus )
```

Ustawia autobus używany w przejeździe.

Parametry

<i>autobus</i>	Autobus używany w przejeździe.
----------------	--------------------------------

Definicja w linii [136](#) pliku [Przejazd.java](#).

6.62.2.14 setCenaZaLitr()

```
void com.mklachl.sopkom.model.entity.Przejazd.setCenaZaLitr (
    float cenaZaLitr )
```

Ustawia cenę za litr paliwa podczas przejazdu.

Parametry

<i>cenaZaLitr</i>	Cena za litr paliwa podczas przejazdu.
-------------------	--

Definicja w linii [184](#) pliku [Przejazd.java](#).

6.62.2.15 setData()

```
void com.mklachl.sopkom.model.entity.Przejazd.setData (
    Date data )
```

Ustawia datę przejazdu.

Parametry

<i>data</i>	Data przejazdu.
-------------	-----------------

Definicja w linii [248](#) pliku [Przejazd.java](#).

6.62.2.16 setDataKonca()

```
void com.mklachl.sopkom.model.entity.Przejazd.setDataKonca (
    Date dataKonca )
```

Ustawia datę zakończenia przejazdu.

Parametry

<i>dataKonca</i>	Data zakończenia przejazdu.
------------------	-----------------------------

Definicja w linii 232 pliku [Przejazd.java](#).

6.62.2.17 setDataStartu()

```
void com.mklachl.sopkom.model.entity.Przejazd.setDataStartu ( Date dataStartu )
```

Ustawia datę rozpoczęcia przejazdu.

Parametry

<i>dataStartu</i>	Data rozpoczęcia przejazdu.
-------------------	-----------------------------

Definicja w linii 216 pliku [Przejazd.java](#).

6.62.2.18 setDlugoscTrasy()

```
void com.mklachl.sopkom.model.entity.Przejazd.setDlugoscTrasy ( float dlugoscTrasy )
```

Ustawia długość trasy przejazdu.

Parametry

<i>dlugoscTrasy</i>	Długość trasy przejazdu.
---------------------	--------------------------

Definicja w linii 200 pliku [Przejazd.java](#).

6.62.2.19 setKierowca()

```
void com.mklachl.sopkom.model.entity.Przejazd.setKierowca ( Kierowca kierowca )
```

Ustawia kierowcę związanego z przejazdem.

Parametry

<i>kierowca</i>	Kierowca związany z przejazdem.
-----------------	---------------------------------

Definicja w linii 120 pliku [Przejazd.java](#).

6.62.2.20 setKurs()

```
void com.mklachl.sopkom.model.entity.Przejazd.setKurs (
    Kurs kurs )
```

Ustawia kurs związany z przejazdem.

Parametry

<i>kurs</i>	Kurs związany z przejazdem.
-------------	-----------------------------

Definicja w linii [152](#) pliku [Przejazd.java](#).

6.62.2.21 setPrzejazdBilet()

```
void com.mklachl.sopkom.model.entity.Przejazd.setPrzejazdBilet (
    List< PrzejazdBilet > PrzejazdBilet )
```

Ustawia listę biletów związanych z przejazdem.

Parametry

<i>PrzejazdBilet</i>	Lista biletów związanych z przejazdem.
----------------------	--

Definicja w linii [280](#) pliku [Przejazd.java](#).

6.62.2.22 setPrzejazdId()

```
void com.mklachl.sopkom.model.entity.Przejazd.setPrzejazdId (
    Long przejazdId )
```

Ustawia identyfikator przejazdu.

Parametry

<i>przejazd</i> ↪ <i>Id</i>	Identyfikator przejazdu.
--------------------------------	--------------------------

Definicja w linii [104](#) pliku [Przejazd.java](#).

6.62.2.23 setPrzejazdKursPrzystanekWlini()

```
void com.mklachl.sopkom.model.entity.Przejazd.setPrzejazdKursPrzystanekWlini (
    List< PrzejazdKursPrzystanekWlini > przejazdKursPrzystanekWlini )
```

Ustawia listę przystanków w linii związanych z przejazdem.

Parametry

<i>przejazdKursPrzystanekWlini</i>	Lista przystanków w linii związanych z przejazdem.
------------------------------------	--

Definicja w linii [264](#) pliku [Przejazd.java](#).

6.62.2.24 setSpalanie()

```
void com.mklachl.sopkom.model.entity.Przejazd.setSpalanie (
    float spalanie )
```

Ustawia spalanie autobusu podczas przejazdu.

Parametry

<i>spalanie</i>	Spalanie autobusu podczas przejazdu.
-----------------	--------------------------------------

Definicja w linii [168](#) pliku [Przejazd.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.63 Dokumentacja klasy com.mklachl.sopkom.model.entity.PrzejazdBilet

Metody publiczne

- Long [getPrzejazdBileId](#) ()
- void [setPrzejazdBileId](#) (Long przejazdBileId)
- [Przejazd getPrzejazd](#) ()
- void [setPrzejazd](#) ([Przejazd](#) przejazd)
- [Bilet getBilet](#) ()
- void [setBilet](#) ([Bilet](#) bilet)
- float [getCenaBiletu](#) ()
- void [setCenaBiletu](#) (float cenaBiletu)

6.63.1 Opis szczegółowy

Klasa reprezentująca encję [PrzejazdBilet](#).

Definicja w linii [10](#) pliku [PrzejazdBilet.java](#).

6.63.2 Dokumentacja funkcji składowych

6.63.2.1 getBilet()

```
Bilet com.mklachl.sopkom.model.entity.PrzejazdBilet.getBilet ( )
```

Zwraca bilet związanego z przejazdem.

Zwrota

bilet [Bilet](#) związanego z przejazdem.

Definicja w linii [76](#) pliku [PrzejazdBilet.java](#).

6.63.2.2 getCenaBiletu()

```
float com.mklachl.sopkom.model.entity.PrzejazdBilet.getCenaBiletu ( )
```

Zwraca cenę biletu.

Zwrota

cenaBiletu Cena biletu.

Definicja w linii [92](#) pliku [PrzejazdBilet.java](#).

6.63.2.3 getPrzejazd()

```
Przejazd com.mklachl.sopkom.model.entity.PrzejazdBilet.getPrzejazd ( )
```

Zwraca przejazd związanego z biletem.

Zwrota

przejazd [Przejazd](#) związanego z biletem.

Definicja w linii [60](#) pliku [PrzejazdBilet.java](#).

6.63.2.4 getPrzejazdBileId()

```
Long com.mklachl.sopkom.model.entity.PrzejazdBilet.getPrzejazdBileId ( )
```

Zwraca identyfikator [PrzejazdBilet](#).

Zwraca

przejazdBileId Identyfikator [PrzejazdBilet](#).

Definicja w linii 44 pliku [PrzejazdBilet.java](#).

6.63.2.5 setBilet()

```
void com.mklachl.sopkom.model.entity.PrzejazdBilet.setBilet (   
    Bilet bilet )
```

Ustawia bilet związany z przejazdem.

Parametry

<i>bilet</i>	Bilet związany z przejazdem.
--------------	------------------------------

Definicja w linii 84 pliku [PrzejazdBilet.java](#).

6.63.2.6 setCenaBiletu()

```
void com.mklachl.sopkom.model.entity.PrzejazdBilet.setCenaBiletu (
    float cenaBiletu )
```

Ustawia cenę biletu.

Parametry

<i>cenaBiletu</i>	Cena biletu.
-------------------	--------------

Definicja w linii 100 pliku [PrzejazdBilet.java](#).

6.63.2.7 setPrzejazd()

```
void com.mklachl.sopkom.model.entity.PrzejazdBilet.setPrzejazd (
    Przejazd przejazd )
```

Ustawia przejazd związany z biletem.

Parametry

<i>przejazd</i>	Przejazd związany z biletem.
-----------------	------------------------------

Definicja w linii 68 pliku [PrzejazdBilet.java](#).

6.63.2.8 setPrzejazdBiletid()

```
void com.mklachl.sopkom.model.entity.PrzejazdBilet.setPrzejazdBiletid (
    Long przejazdBiletid )
```

Ustawia identyfikator [PrzejazdBilet](#).

Parametry

<code>przejazd<--> BiletId</code>	Identyfikator PrzejazdBilet .
---	---

Definicja w linii [52](#) pliku [PrzejazdBilet.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.64 Dokumentacja klasy com.mklachl.sopkom.model.dto.PrzejazdBiletDto

Metody publiczne

- `PrzejazdBiletDto ()`
- `PrzejazdBiletDto (PrzejazdBilet przejazdBilet)`
- `Long getPrzejazdBiletId ()`
- `void setPrzejazdBiletId (Long przejazdBiletId)`
- `Long getPrzejazdId ()`
- `void setPrzejazdId (Long przejazdId)`
- `Short getBiletId ()`
- `void setBiletId (Short biletId)`
- `float getCenaBiletu ()`
- `void setCenaBiletu (float cenaBiletu)`

6.64.1 Opis szczegółowy

Data Transfer Object (DTO) dla encji PrzejazdBilet.

Definicja w linii [8](#) pliku [PrzejazdBiletDto.java](#).

6.64.2 Dokumentacja konstruktora i destruktora

6.64.2.1 PrzejazdBiletDto() [1/2]

```
com.mklachl.sopkom.model.dto.PrzejazdBiletDto.PrzejazdBiletDto ( )
```

Konstruktor domyślny.

Definicja w linii [33](#) pliku [PrzejazdBiletDto.java](#).

6.64.2.2 PrzejazdBiletDto() [2/2]

```
com.mklachl.sopkom.model.dto.PrzejazdBiletDto.PrzejazdBiletDto (   
    PrzejazdBilet przejazdBilet )
```

Konstruktor przyjmujący obiekt encji PrzejazdBilet.

Parametry

<i>przejazdBilet</i>	Obiekt encji PrzejazdBilet.
----------------------	-----------------------------

Definicja w linii [40](#) pliku [PrzejazdBiletDto.java](#).

6.64.3 Dokumentacja funkcji składowych

6.64.3.1 `getBileId()`

```
Short com.mklachl.sopkom.model.dto.PrzejazdBiletDto.getBileId ()
```

Zwraca identyfikator biletu.

Zwraca

bileId Identyfikator biletu.

Definicja w linii [85](#) pliku [PrzejazdBiletDto.java](#).

6.64.3.2 `getCenaBiletu()`

```
float com.mklachl.sopkom.model.dto.PrzejazdBiletDto.getCenaBiletu ()
```

Zwraca cenę biletu.

Zwraca

cenaBiletu Cena biletu.

Definicja w linii [101](#) pliku [PrzejazdBiletDto.java](#).

6.64.3.3 `getPrzejazdBileId()`

```
Long com.mklachl.sopkom.model.dto.PrzejazdBiletDto.getPrzejazdBileId ()
```

Zwraca identyfikator PrzejazdBilet.

Zwraca

przejazdBileId Identyfikator PrzejazdBilet.

Definicja w linii [53](#) pliku [PrzejazdBiletDto.java](#).

6.64.3.4 getPrzejazdId()

```
Long com.mklachl.sopkom.model.dto.PrzejazdBiletDto.getPrzejazdId ( )
```

Zwraca identyfikator przejazdu.

Zwraca

przejazdId Identyfikator przejazdu.

Definicja w linii 69 pliku [PrzejazdBiletDto.java](#).

6.64.3.5 setBileId()

```
void com.mklachl.sopkom.model.dto.PrzejazdBiletDto.setBileId (  
    Short biletId )
```

Ustawia identyfikator biletu.

Parametry

<i>bilet</i> ↪	Identyfikator biletu.
<i>Id</i>	

Definicja w linii 93 pliku [PrzejazdBiletDto.java](#).

6.64.3.6 setCenaBiletu()

```
void com.mklachl.sopkom.model.dto.PrzejazdBiletDto.setCenaBiletu (   
    float cenaBiletu )
```

Ustawia cenę biletu.

Parametry

<i>cenaBiletu</i>	Cena biletu.
-------------------	--------------

Definicja w linii 109 pliku [PrzejazdBiletDto.java](#).

6.64.3.7 setPrzejazdBileId()

```
void com.mklachl.sopkom.model.dto.PrzejazdBiletDto.setPrzejazdBileId (   
    Long przejazdBileId )
```

Ustawia identyfikator PrzejazdBilet.

Parametry

<i>przejazd</i> ↪ Biletid	Identyfikator PrzejazdBilet.
------------------------------	------------------------------

Definicja w linii 61 pliku [PrzejazdBiletDto.java](#).

6.64.3.8 setPrzejazdId()

```
void com.mklachl.sopkom.model.dto.PrzejazdBiletDto.setPrzejazdId (
    Long przejazdId )
```

Ustawia identyfikator przejazdu.

Parametry

<i>przejazd</i> ↪ Id	Identyfikator przejazdu.
-------------------------	--------------------------

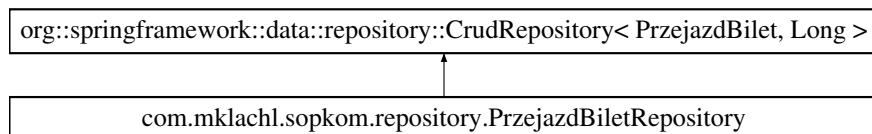
Definicja w linii 77 pliku [PrzejazdBiletDto.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.65 Dokumentacja interfejsu com.mklachl.sopkom.repository.PrzejazdBiletRepository

Diagram dziedziczenia dla com.mklachl.sopkom.repository.PrzejazdBiletRepository



Metody publiczne

- List< [PrzejazdBilet](#) > [findAllByPrzejazd](#) ([Przejazd](#) przejazd)
- List< [PrzejazdBilet](#) > [findAllByBilet](#) ([Bilet](#) bilet)
- List< [PrzejazdBilet](#) > [findAllByCenaBiletuGreaterThanOrEqual](#) (float cenaBiletu)
- List< [PrzejazdBilet](#) > [findAllByCenaBiletuLessThan](#) (float cenaBiletu)
- List< [PrzejazdBilet](#) > [findAllByCenaBiletu](#) (float cenaBiletu)
- List< [PrzejazdBilet](#) > [findAll](#) ()

6.65.1 Opis szczegółowy

Interfejs repository dla encji PrzejazdBilet

Definicja w linii 13 pliku [PrzejazdBiletRepository.java](#).

6.65.2 Dokumentacja funkcji składowych

6.65.2.1 findAll()

```
List< PrzejazdBilet > com.mklachl.sopkom.repository.PrzejazdBiletRepository.findAll ( )
```

Znajduje wszystkie bilety w przejeździe

Zwroca

Lista biletów w przejeździe

6.65.2.2 findAllByBilet()

```
List< PrzejazdBilet > com.mklachl.sopkom.repository.PrzejazdBiletRepository.findAllByBilet ( Bilet bilet )
```

Znajduje wszystkie bilety w przejeździe dla danego biletu

Parametry

bilet	bilet
-------	-------

Zwroca

Lista biletów w przejeździe

6.65.2.3 findAllByCenaBiletu()

```
List< PrzejazdBilet > com.mklachl.sopkom.repository.PrzejazdBiletRepository.findAllByCenaBiletu ( float cenaBiletu )
```

Znajduje wszystkie bilety w przejeździe o dokładnej cenie

Parametry

<i>cenaBiletu</i>	cena biletu
-------------------	-------------

Zwroca

Lista biletów w przejeździe

6.65.2.4 findAllByCenaBiletuGreaterThanOr()

```
List< PrzejazdBilet > com.mklachl.sopkom.repository.PrzejazdBiletRepository.findAllByCenaBiletuGreaterThanOr (
    float cenaBiletu )
```

Znajduje wszystkie bilety w przejeździe o cenie większej niż podana

Parametry

<i>cenaBiletu</i>	cena biletu
-------------------	-------------

Zwroca

Lista biletów w przejeździe

6.65.2.5 findAllByCenaBiletuLessThanOr()

```
List< PrzejazdBilet > com.mklachl.sopkom.repository.PrzejazdBiletRepository.findAllByCenaBiletuLessThanOr (
    float cenaBiletu )
```

Znajduje wszystkie bilety w przejeździe o cenie mniejszej niż podana

Parametry

<i>cenaBiletu</i>	cena biletu
-------------------	-------------

Zwroca

Lista biletów w przejeździe

6.65.2.6 findAllByPrzejazd()

```
List< PrzejazdBilet > com.mklachl.sopkom.repository.PrzejazdBiletRepository.findAllByPrzejazd
(
    Przejazd przejazd )
```

Znajduje wszystkie bilety w przejeździe dla danego przejazdu

Parametry

<code>przejazd</code>	<code>przejazd</code>
-----------------------	-----------------------

Zwraca

Lista biletów w przejeździe

Dokumentacja dla tego interfejsu została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/rep

6.66 Dokumentacja klasy

com.mklachl.sopkom.controller.PrzejazdController

Metody publiczne

- ResponseEntity<?> `savePrzejazd` (@RequestBody `PrzejazdDtoInput` input)
- ResponseEntity<?> `getPrzejazd` (@PathVariable(name="przejazd_id") Long przejazdid)
- ResponseEntity<?> `deletePrzejazd` (@PathVariable(name="przejazd_id") Long przejazdid)
- ResponseEntity<?> `getPrzejazdy` (@RequestParam(name="kurs_id", required=false) Long kursId)
- ResponseEntity<?> `gatAllAvailableDriversForKurs` (@RequestParam(name="kurs_id") Long kursId, @RequestParam(name="data") @DateTimeFormat(pattern="dd.MM.yyyy") Date date)
- ResponseEntity<?> `gatAllAvailableBusesForKurs` (@RequestParam(name="kurs_id") Long kursId, @RequestParam(name="data") @DateTimeFormat(pattern="dd.MM.yyyy") Date date)
- ResponseEntity<?> `getPrzyszlePrzejazdy` ()
- ResponseEntity<?> `gatAllAvailableDatesForKurs` (@RequestParam(name="kurs_id") Long kursId)

Atrybuty publiczne

- `HarmonogramRepository` harmonogramRepository
- `KursRepository` kursRepository
- `KursService` kursService
- `PrzejazdService` przejazdService
- `KierowcaRepository` kierowcaRepository
- `AutobusRepository` autobusRepository
- `PrzejazdRepository` przejazdRepository

6.66.1 Opis szczegółowy

Definicja w linii 36 pliku `PrzejazdController.java`.

6.66.2 Dokumentacja funkcji składowych

6.66.2.1 deletePrzejazd()

```
ResponseEntity<?> com.mklachl.sopkom.controller.PrzejazdController.deletePrzejazd ( @PathVariable(name="przejazd_id") Long przejazdid )
```

Endpoint do usuwania przejazdu według ID.

Parametry

przejazdid	ID przejazdu
------------	--------------

Zwroca

Odpowiedź HTTP

Definicja w linii [102](#) pliku [PrzejazdController.java](#).

6.66.2.2 getAllAvailableBusesForKurs()

```
ResponseEntity<?> com.mklachl.sopkom.controller.PrzejazdController.getAllAvailableBusesForKurs ( @RequestParam(name="kurs_id") Long kursId, @RequestParam(name="data") @DateTimeFormat(pattern="dd.MM.yyyy") Date date )
```

Endpoint do pobierania wszystkich dostępnych autobusów dla kursu w danym dniu.

Parametry

kurs ↳ Id	ID kursu
date	Data przejazdu

Zwroca

Lista dostępnych autobusów

Definicja w linii [198](#) pliku [PrzejazdController.java](#).

6.66.2.3 gatAllAvailableDatesForKurs()

```
ResponseEntity<?> com.mklachl.sopkom.controller.PrzejazdController.gatAllAvailableDatesForKurs
(
    @RequestParam(name="kurs_id") Long kursId )
```

Endpoint do pobierania wszystkich dostępnych dat dla kursu.

Parametry

<i>kurs</i> ↳ <i>Id</i>	ID kursu
-------------------------------	----------

Zwroca

Lista dostępnych dat

Definicja w linii 294 pliku [PrzejazdController.java](#).

6.66.2.4 gatAllAvailableDriversForKurs()

```
ResponseEntity<?> com.mklachl.sopkom.controller.PrzejazdController.gatAllAvailableDriversForKurs
(
    @RequestParam(name="kurs_id") Long kursId,
    @RequestParam(name="data") @DateTimeFormat(pattern="dd.MM.yyyy") Date date )
```

Endpoint do pobierania wszystkich dostępnych kierowców dla kursu w danym dniu.

Parametry

<i>kurs</i> ↳ <i>Id</i>	ID kursu
<i>date</i>	Data przejazdu

Zwroca

Lista dostępnych kierowców

Definicja w linii 149 pliku [PrzejazdController.java](#).

6.66.2.5 getPrzejazd()

```
ResponseEntity<?> com.mklachl.sopkom.controller.PrzejazdController.getPrzejazd (
    @PathVariable(name="przejazd_id") Long przejazdid )
```

Endpoint do pobierania przejazdu według ID.

Parametry

<i>przejazdid</i>	ID przejazdu
-------------------	--------------

Zwrota

Obiekt PrzejazdDtoOutput

Definicja w linii [80](#) pliku [PrzejazdController.java](#).

6.66.2.6 getPrzejazdy()

```
ResponseEntity<?> com.mklachl.sopkom.controller.PrzejazdController.getPrzejazdy ( @RequestParam(name="kurs_id", required=false) Long kursId )
```

Endpoint do pobierania przejazdów z opcjonalnym filtrowaniem według kursu.

Parametry

<i>kurs</i> <i>Id</i>	ID kursu
--------------------------	----------

Zwrota

Lista przejazdów

Definicja w linii [122](#) pliku [PrzejazdController.java](#).

6.66.2.7 getPrzyszłePrzejazdy()

```
ResponseEntity<?> com.mklachl.sopkom.controller.PrzejazdController.getPrzyszłePrzejazdy ( )
```

Endpoint do pobierania przyszłych przejazdów.

Zwrota

Lista przyszłych przejazdów

Definicja w linii [245](#) pliku [PrzejazdController.java](#).

6.66.2.8 savePrzejazd()

```
ResponseEntity<?> com.mklachl.sopkom.controller.PrzejazdController.savePrzejazd ( @RequestBody PrzejazdDtoInput input )
```

Endpoint do zapisywania nowego przejazdu.

Parametry

<i>input</i>	DTO przejazdu do zapisania
--------------	----------------------------

Zwroca

Odpowiedź HTTP

Definicja w linii [65](#) pliku [PrzejazdController.java](#).

6.66.3 Dokumentacja atrybutów składowych

6.66.3.1 autobusRepository

[AutobusRepository](#) com.mklachl.sopkom.controller.PrzejazdController.autobusRepository

Definicja w linii [54](#) pliku [PrzejazdController.java](#).

6.66.3.2 harmonogramRepository

[HarmonogramRepository](#) com.mklachl.sopkom.controller.PrzejazdController.harmonogramRepository

Definicja w linii [39](#) pliku [PrzejazdController.java](#).

6.66.3.3 kierowcaRepository

[KierowcaRepository](#) com.mklachl.sopkom.controller.PrzejazdController.kierowcaRepository

Definicja w linii [51](#) pliku [PrzejazdController.java](#).

6.66.3.4 kursRepository

[KursRepository](#) com.mklachl.sopkom.controller.PrzejazdController.kursRepository

Definicja w linii [42](#) pliku [PrzejazdController.java](#).

6.66.3.5 kursService

KursService com.mklachl.sopkom.controller.PrzejazdController.kursService

Definicja w linii 45 pliku [PrzejazdController.java](#).

6.66.3.6 przejazdRepository

PrzejazdRepository com.mklachl.sopkom.controller.PrzejazdController.przejazdRepository

Definicja w linii 57 pliku [PrzejazdController.java](#).

6.66.3.7 przejazdService

PrzejazdService com.mklachl.sopkom.controller.PrzejazdController.przejazdService

Definicja w linii 48 pliku [PrzejazdController.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/com

6.67 Dokumentacja klasy com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoInput

Metody publiczne

- PrzejazdDtoInput ()
- Long getKursId ()
- void setKursId (Long kursId)
- Long getKierowcaId ()
- void setKierowcaId (Long kierowcaId)
- Long getAutobusId ()
- void setAutobusId (Long autobusId)
- Date getDataPrzejazdu ()
- void setDataPrzejazdu (Date dataPrzejazdu)
- String toString ()

6.67.1 Opis szczegółowy

Data Transfer Object (DTO) wejściowy dla encji Przejazd.

Definicja w linii 9 pliku [PrzejazdDtoInput.java](#).

6.67.2 Dokumentacja konstruktora i destruktora

6.67.2.1 PrzejazdDtoInput()

```
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoInput.PrzejazdDtoInput( )
```

Konstruktor domyślny.

Definicja w linii [35](#) pliku [PrzejazdDtoInput.java](#).

6.67.3 Dokumentacja funkcji składowych

6.67.3.1 getAutobusId()

```
Long com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoInput.getAutobusId( )
```

Zwraca identyfikator autobusu.

Zwroca

autobusId Identyfikator autobusu.

Definicja w linii [75](#) pliku [PrzejazdDtoInput.java](#).

6.67.3.2 getDataPrzejazdu()

```
Date com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoInput.getDataPrzejazdu( )
```

Zwraca datę przejazdu.

Zwroca

dataPrzejazdu Data przejazdu.

Definicja w linii [91](#) pliku [PrzejazdDtoInput.java](#).

6.67.3.3 getKierowcId()

```
Long com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoInput.getKierowcaId ( )
```

Zwraca identyfikator kierowcy.

Zwraca

kierowcId Identyfikator kierowcy.

Definicja w linii 59 pliku [PrzejazdDtoInput.java](#).

6.67.3.4 getKursId()

```
Long com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoInput.getKursId ( )
```

Zwraca identyfikator kursu.

Zwraca

kursId Identyfikator kursu.

Definicja w linii 43 pliku [PrzejazdDtoInput.java](#).

6.67.3.5 setAutobusId()

```
void com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoInput.setAutobusId ( Long autobusId )
```

Ustawia identyfikator autobusu.

Parametry

<i>autobus</i> <i>Id</i>	Identyfikator autobusu.
-----------------------------	-------------------------

Definicja w linii 83 pliku [PrzejazdDtoInput.java](#).

6.67.3.6 setDataPrzejazdu()

```
void com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoInput.setDataPrzejazdu ( Date dataPrzejazdu )
```

Ustawia datę przejazdu.

Parametry

<i>dataPrzejazdu</i>	Data przejazdu.
----------------------	-----------------

Definicja w linii 99 pliku [PrzejazdDtoInput.java](#).

6.67.3.7 setKierowcaId()

```
void com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoInput.setKierowcaId (
    Long kierowcaId )
```

Ustawia identyfikator kierowcy.

Parametry

<i>kierowca</i> <i>Id</i>	Identyfikator kierowcy.
------------------------------	-------------------------

Definicja w linii 67 pliku [PrzejazdDtoInput.java](#).

6.67.3.8 setKursId()

```
void com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoInput.setKursId (
    Long kursId )
```

Ustawia identyfikator kursu.

Parametry

<i>kurs</i> <i>Id</i>	Identyfikator kursu.
--------------------------	----------------------

Definicja w linii 51 pliku [PrzejazdDtoInput.java](#).

6.67.3.9 toString()

```
String com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoInput.toString ( )
```

Definicja w linii 104 pliku [PrzejazdDtoInput.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.68 Dokumentacja klasy com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput

Metody publiczne

- `PrzejazdDtoOutput ()`
- `PrzejazdDtoOutput (Przejazd przejazd)`
- `PrzejazdDtoOutput (Przejazd przejazd, boolean a)`
- `Long getPrzejazdId ()`
- `void setPrzejazdId (Long przejazdId)`
- `Long getKursId ()`
- `void setKursId (Long kursId)`
- `String getLiniaNumer ()`
- `void setLiniaNumer (String liniaNumer)`
- `Date getDataStartu ()`
- `void setDataStartu (Date dataStartu)`
- `Date getDataKonca ()`
- `void setDataKonca (Date dataKonca)`
- `String getKierowcamiNazwisko ()`
- `void setKierowcamiNazwisko (String kierowcamiNazwisko)`
- `String getAutobusNazwa ()`
- `void setAutobusNazwa (String autobusNazwa)`
- `String getAutobusNumerRejestracyjny ()`
- `void setAutobusNumerRejestracyjny (String autobusNumerRejestracyjny)`
- `float getSpalanie ()`
- `void setSpalanie (float spalanie)`
- `float getCenaZaLitr ()`
- `void setCenaZaLitr (float cenaZaLitr)`
- `float getDlugoscTrasy ()`
- `void setDlugoscTrasy (float dlugoscTrasy)`
- `int getLiczbaBiletow ()`
- `void setLiczbaBiletow (int liczbaBiletow)`
- `String getStatus ()`
- `void setStatus (String status)`
- `List< PrzejazdKursPrzystanekWliniDto > getPrzystanki ()`
- `void setPrzystanki (List< PrzejazdKursPrzystanekWliniDto > przystanki)`
- `Date getData ()`
- `void setData (Date data)`
- `short getKierunek ()`
- `void setKierunek (short kierunek)`

6.68.1 Opis szczegółowy

Data Transfer Object (DTO) wyjściowy dla encji Przejazd.

Definicja w linii 13 pliku [PrzejazdDtoOutput.java](#).

6.68.2 Dokumentacja konstruktora i destruktora

6.68.2.1 PrzejazdDtoOutput() [1/3]

```
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput.PrzejazdDtoOutput ( )
```

Konstruktor domyślny.

Definicja w linii 101 pliku [PrzejazdDtoOutput.java](#).

6.68.2.2 PrzejazdDtoOutput() [2/3]

```
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput.PrzejazdDtoOutput (  
    Przejazd przejazd )
```

Konstruktor przyjmujący obiekt encji Przejazd.

Parametry

<i>przejazd</i>	Obiekt encji Przejazd.
-----------------	------------------------

Definicja w linii 107 pliku [PrzejazdDtoOutput.java](#).

6.68.2.3 PrzejazdDtoOutput() [3/3]

```
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput.PrzejazdDtoOutput (  
    Przejazd przejazd,  
    boolean a )
```

Konstruktor przyjmujący obiekt encji Przejazd oraz flagę.

Parametry

<i>przejazd</i>	Obiekt encji Przejazd.
<i>a</i>	Flaga.

Definicja w linii 154 pliku [PrzejazdDtoOutput.java](#).

6.68.3 Dokumentacja funkcji składowych

6.68.3.1 getAutobusNazwa()

```
String com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput.getAutobusNazwa ( )
```

Zwraca nazwę autobusu.

Zwrota

autobus Nazwa Nazwa autobusu.

Definicja w linii [276](#) pliku [PrzejazdDtoOutput.java](#).

6.68.3.2 getAutobusNumerRejestracyjny()

```
String com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput.getAutobusNumerRejestracyjny ( )
```

Zwrota numer rejestracyjny autobusu.

Zwrota

autobusNumerRejestracyjny Numer rejestracyjny autobusu.

Definicja w linii [292](#) pliku [PrzejazdDtoOutput.java](#).

6.68.3.3 getCenaZaLitr()

```
float com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput.getCenaZaLitr ( )
```

Zwrota cenę za litr paliwa.

Zwrota

cenaZaLitr Cena za litr paliwa.

Definicja w linii [324](#) pliku [PrzejazdDtoOutput.java](#).

6.68.3.4 getData()

```
Date com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput.getData ( )
```

Zwrota datę.

Zwrota

data Data.

Definicja w linii [404](#) pliku [PrzejazdDtoOutput.java](#).

6.68.3.5 **getDataKonca()**

```
Date com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput.getDataKonca( )
```

Zwraca datę końca.

Zwraca

dataKonca Data końca.

Definicja w linii [244](#) pliku [PrzejazdDtoOutput.java](#).

6.68.3.6 **getDataStartu()**

```
Date com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput.getDataStartu( )
```

Zwraca datę startu.

Zwraca

dataStartu Data startu.

Definicja w linii [228](#) pliku [PrzejazdDtoOutput.java](#).

6.68.3.7 **getDlugoscTrasy()**

```
float com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput.getDlugoscTrasy( )
```

Zwraca długość trasy.

Zwraca

dlugoscTrasy Długość trasy.

Definicja w linii [340](#) pliku [PrzejazdDtoOutput.java](#).

6.68.3.8 **getKierowcImieNazwisko()**

```
String com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput.getKierowcaImieNazwisko( )
```

Zwraca imię i nazwisko kierowcy.

Zwraca

kierowcImieNazwisko Imię i nazwisko kierowcy.

Definicja w linii [260](#) pliku [PrzejazdDtoOutput.java](#).

6.68.3.9 getKierunek()

```
short com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput.getKierunek ( )
```

Zwraca kierunek kursu.

Zwraca

kierunek Kierunek kursu.

Definicja w linii [420](#) pliku [PrzejazdDtoOutput.java](#).

6.68.3.10 getKursId()

```
Long com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput.getKursId ( )
```

Zwraca identyfikator kursu.

Zwraca

kursId Identyfikator kursu.

Definicja w linii [196](#) pliku [PrzejazdDtoOutput.java](#).

6.68.3.11 getLiczbaBiletow()

```
int com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput.getLiczbaBiletow ( )
```

Zwraca liczbę biletów.

Zwraca

liczbaBiletow Liczba biletów.

Definicja w linii [356](#) pliku [PrzejazdDtoOutput.java](#).

6.68.3.12 getLiniaNumer()

```
String com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput.getLiniaNumer ( )
```

Zwraca numer linii.

Zwraca

liniaNumer Numer linii.

Definicja w linii [212](#) pliku [PrzejazdDtoOutput.java](#).

6.68.3.13 getPrzejazdId()

```
Long com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput.getPrzejazdId ( )
```

Zwraca identyfikator przejazdu.

Zwraca

przejazdId Identyfikator przejazdu.

Definicja w linii [180](#) pliku [PrzejazdDtoOutput.java](#).

6.68.3.14 getPrzystanki()

```
List< PrzejazdKursPrzystanekWliniDto > com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput.getPrzystanki ( )
```

Zwraca listę przystanków w kursie.

Zwraca

przystanki Lista przystanków w kursie.

Definicja w linii [388](#) pliku [PrzejazdDtoOutput.java](#).

6.68.3.15 getSpalanie()

```
float com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput.getSpalanie ( )
```

Zwraca spalanie.

Zwraca

spalanie Spalanie.

Definicja w linii [308](#) pliku [PrzejazdDtoOutput.java](#).

6.68.3.16 getStatus()

```
String com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput.getStatus ( )
```

Zwraca status przejazdu.

Zwraca

status Status przejazdu.

Definicja w linii [372](#) pliku [PrzejazdDtoOutput.java](#).

6.68.3.17 setAutobusNazwa()

```
void com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput.setAutobusNazwa ( String autobusNazwa )
```

Ustawia nazwę autobusu.

Parametry

<i>autobusNazwa</i>	Nazwa autobusu.
---------------------	-----------------

Definicja w linii 284 pliku [PrzejazdDtoOutput.java](#).

6.68.3.18 setAutobusNumerRejestracyjny()

```
void com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput.setAutobusNumerRejestracyjny ( String autobusNumerRejestracyjny )
```

Ustawia numer rejestracyjny autobusu.

Parametry

<i>autobusNumerRejestracyjny</i>	Numer rejestracyjny autobusu.
----------------------------------	-------------------------------

Definicja w linii 300 pliku [PrzejazdDtoOutput.java](#).

6.68.3.19 setCenaZaLitr()

```
void com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput.setCenaZaLitr ( float cenaZaLitr )
```

Ustawia cenę za litr paliwa.

Parametry

<i>cenaZaLitr</i>	Cena za litr paliwa.
-------------------	----------------------

Definicja w linii 332 pliku [PrzejazdDtoOutput.java](#).

6.68.3.20 setData()

```
void com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput.setData ( Date data )
```

Ustawia datę.

Parametry

<i>data</i>	Data.
-------------	-------

Definicja w linii [412](#) pliku [PrzejazdDtoOutput.java](#).

6.68.3.21 setDataKonca()

```
void com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput.setDataKonca ( Date dataKonca )
```

Ustawia datę końca.

Parametry

<i>dataKonca</i>	Data końca.
------------------	-------------

Definicja w linii [252](#) pliku [PrzejazdDtoOutput.java](#).

6.68.3.22 setDataStartu()

```
void com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput.setDataStartu ( Date dataStartu )
```

Ustawia datę startu.

Parametry

<i>dataStartu</i>	Data startu.
-------------------	--------------

Definicja w linii [236](#) pliku [PrzejazdDtoOutput.java](#).

6.68.3.23 setDlugoscTrasy()

```
void com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput.setDlugoscTrasy ( float dlugoscTrasy )
```

Ustawia długość trasy.

Parametry

<i>dlugoscTrasy</i>	Długość trasy.
---------------------	----------------

Definicja w linii [348](#) pliku [PrzejazdDtoOutput.java](#).

6.68.3.24 setKierowcaImieNazwisko()

```
void com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput.setKierowcaImieNazwisko ( String kierowcaImieNazwisko )
```

Ustawia imię i nazwisko kierowcy.

Parametry

<i>kierowcaImieNazwisko</i>	Imię i nazwisko kierowcy.
-----------------------------	---------------------------

Definicja w linii [268](#) pliku [PrzejazdDtoOutput.java](#).

6.68.3.25 setKierunek()

```
void com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput.setKierunek ( short kierunek )
```

Ustawia kierunek kursu.

Parametry

<i>kierunek</i>	Kierunek kursu.
-----------------	-----------------

Definicja w linii [428](#) pliku [PrzejazdDtoOutput.java](#).

6.68.3.26 setKursId()

```
void com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput.setKursId ( Long kursId )
```

Ustawia identyfikator kursu.

Parametry

<i>kurs</i> ↗ <i>Id</i>	Identyfikator kursu.
----------------------------	----------------------

Definicja w linii [204](#) pliku [PrzejazdDtoOutput.java](#).

6.68.3.27 setLiczbaBiletow()

```
void com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput.setLiczbaBiletow ( int liczbaBiletow )
```

Ustawia liczbę biletów.

Parametry

<i>liczbaBiletow</i>	Liczba biletów.
----------------------	-----------------

Definicja w linii [364](#) pliku [PrzejazdDtoOutput.java](#).

6.68.3.28 setLiniaNumer()

```
void com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput.setLiniaNumer (
    String liniaNumer )
```

Ustawia numer linii.

Parametry

<i>liniaNumer</i>	Numer linii.
-------------------	--------------

Definicja w linii [220](#) pliku [PrzejazdDtoOutput.java](#).

6.68.3.29 setPrzejazdId()

```
void com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput.setPrzejazdId (
    Long przejazdId )
```

Ustawia identyfikator przejazdu.

Parametry

<i>przejazd</i> ↵ <i>Id</i>	Identyfikator przejazdu.
--------------------------------	--------------------------

Definicja w linii [188](#) pliku [PrzejazdDtoOutput.java](#).

6.68.3.30 setPrzystanki()

```
void com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput.setPrzystanki (
    List< PrzejazdKursPrzystanekWliniDto > przystanki )
```

Ustawia listę przystanków w kursie.

Parametry

<i>przystanki</i>	Lista przystanków w kursie.
-------------------	-----------------------------

Definicja w linii [396](#) pliku [PrzejazdDtoOutput.java](#).

6.68.3.31 setSpalanie()

```
void com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput.setSpalanie ( float spalanie )
```

Ustawia spalanie.

Parametry

<i>spalanie</i>	Spalanie.
-----------------	-----------

Definicja w linii [316](#) pliku [PrzejazdDtoOutput.java](#).

6.68.3.32 setStatus()

```
void com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput.setStatus ( String status )
```

Ustawia status przejazdu.

Parametry

<i>status</i>	Status przejazdu.
---------------	-------------------

Definicja w linii [380](#) pliku [PrzejazdDtoOutput.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.69 Dokumentacja klasy com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate

Metody publiczne

- [PrzejazdDtoUpdate \(\)](#)

- long `getPrzejazdId()`
- void `setPrzejazdId(long przejazdId)`
- double `getSpalanie()`
- void `setSpalanie(double spalanie)`
- double `getCenaZaLitr()`
- void `setCenaZaLitr(double cenaZaLitr)`
- double `getDlugoszTrasy()`
- void `setDlugoszTrasy(double dlugoszTrasy)`
- int `getLiczbaBiletowUlgowych()`
- void `setLiczbaBiletowUlgowych(int liczbaBiletowUlgowych)`
- int `getLiczbaBiletowNormalnych()`
- void `setLiczbaBiletowNormalnych(int liczbaBiletowNormalnych)`
- List<`PKPWLDtoUpdate`> `getPrzystanki()`
- void `setPrzystanki(List<PKPWLDtoUpdate> przystanki)`
- String `toString()`

6.69.1 Opis szczegółowy

Data Transfer Object (DTO) dla aktualizacji encji Przejazd.

Definicja w linii 8 pliku [PrzejazdDtoUpdate.java](#).

6.69.2 Dokumentacja konstruktora i destruktora

6.69.2.1 PrzejazdDtoUpdate()

```
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate.PrzejazdDtoUpdate( )
```

Konstruktor domyślny.

Definicja w linii 48 pliku [PrzejazdDtoUpdate.java](#).

6.69.3 Dokumentacja funkcji składowych

6.69.3.1 getCenaZaLitr()

```
double com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate.getCenaZaLitr( )
```

Zwraca cenę za litr paliwa.

Zwrota

cenaZaLitr Cena za litr paliwa.

Definicja w linii 88 pliku [PrzejazdDtoUpdate.java](#).

6.69.3.2 getDlugoszTrasy()

```
double com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate.getDlugoszTrasy ( )
```

Zwraca długość trasy.

Zwraca

dlugoszTrasy Długość trasy.

Definicja w linii 104 pliku [PrzejazdDtoUpdate.java](#).

6.69.3.3 getLiczbaBiletowNormalnych()

```
int com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate.getLiczbaBiletowNormalnych ( )
```

Zwraca liczbę biletów normalnych.

Zwraca

liczbaBiletowNormalnych Liczba biletów normalnych.

Definicja w linii 136 pliku [PrzejazdDtoUpdate.java](#).

6.69.3.4 getLiczbaBiletowUlgowych()

```
int com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate.getLiczbaBiletowUlgowych ( )
```

Zwraca liczbę biletów ulgowych.

Zwraca

liczbaBiletowUlgowych Liczba biletów ulgowych.

Definicja w linii 120 pliku [PrzejazdDtoUpdate.java](#).

6.69.3.5 getPrzejazdId()

```
long com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate.getPrzejazdId ( )
```

Zwraca identyfikator przejazdu.

Zwraca

przejazdId Identyfikator przejazdu.

Definicja w linii 56 pliku [PrzejazdDtoUpdate.java](#).

6.69.3.6 getPrzystanki()

```
List< PKPWLDtoUpdate > com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate.getPrzystanki ( )
```

Zwraca listę przystanków w kursie do aktualizacji.

Zwrota

przystanki Lista przystanków w kursie do aktualizacji.

Definicja w linii 152 pliku [PrzejazdDtoUpdate.java](#).

6.69.3.7 getSpalanie()

```
double com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate.getSpalanie ( )
```

Zwraca spalanie.

Zwrota

spalanie Spalanie.

Definicja w linii 72 pliku [PrzejazdDtoUpdate.java](#).

6.69.3.8 setCenaZaLitr()

```
void com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate.setCenaZaLitr ( double cenaZaLitr )
```

Ustawia cenę za litr paliwa.

Parametry

cenaZaLitr	Cena za litr paliwa.
------------	----------------------

Definicja w linii 96 pliku [PrzejazdDtoUpdate.java](#).

6.69.3.9 setDlugoszTrasy()

```
void com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate.setDlugoszTrasy ( double dlugoszTrasy )
```

Ustawia długość trasy.

Parametry

<i>dlugoszTrasy</i>	Długość trasy.
---------------------	----------------

Definicja w linii [112](#) pliku [PrzejazdDtoUpdate.java](#).

6.69.3.10 setLiczbaBiletowNormalnych()

```
void com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate.setLiczbaBiletowNormalnych ( int liczbaBiletowNormalnych )
```

Ustawia liczbę biletów normalnych.

Parametry

<i>liczbaBiletowNormalnych</i>	Liczba biletów normalnych.
--------------------------------	----------------------------

Definicja w linii [144](#) pliku [PrzejazdDtoUpdate.java](#).

6.69.3.11 setLiczbaBiletowUlgowych()

```
void com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate.setLiczbaBiletowUlgowych ( int liczbaBiletowUlgowych )
```

Ustawia liczbę biletów ulgowych.

Parametry

<i>liczbaBiletowUlgowych</i>	Liczba biletów ulgowych.
------------------------------	--------------------------

Definicja w linii [128](#) pliku [PrzejazdDtoUpdate.java](#).

6.69.3.12 setPrzejazdId()

```
void com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate.setPrzejazdId ( long przejazdId )
```

Ustawia identyfikator przejazdu.

Parametry

<i>przejazd</i> ↵ <i>Id</i>	Identyfikator przejazdu.
--------------------------------	--------------------------

Definicja w linii 64 pliku [PrzejazdDtoUpdate.java](#).

6.69.3.13 setPrzystanki()

```
void com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate.setPrzystanki (   
    List< PKPWLDtoUpdate > przystanki )
```

Ustawia listę przystanków w kursie do aktualizacji.

Parametry

<i>przystanki</i>	Lista przystanków w kursie do aktualizacji.
-------------------	---

Definicja w linii 160 pliku [PrzejazdDtoUpdate.java](#).

6.69.3.14 setSpalanie()

```
void com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate.setSpalanie (   
    double spalanie )
```

Ustawia spalanie.

Parametry

<i>spalanie</i>	Spalanie.
-----------------	-----------

Definicja w linii 80 pliku [PrzejazdDtoUpdate.java](#).

6.69.3.15 toString()

```
String com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate.toString ( )
```

Definicja w linii 165 pliku [PrzejazdDtoUpdate.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.70 Dokumentacja klasy com.mklachl.sopkom.model.entity.PrzejazdKursPrzystanekWlini

Metody publiczne

- Long [getPrzejazdKursPrzystanekWliniId\(\)](#)
- void [setPrzejazdKursPrzystanekWliniId\(Long przejazdKursPrzystanekWliniId\)](#)
- [Przejazd getPrzejazd\(\)](#)
- void [setPrzejazd\(Przejazd przejazd\)](#)
- [KursPrzystanekWlini getKursPrzystanekWlini\(\)](#)
- void [setKursPrzystanekWlini\(KursPrzystanekWlini kursPrzystanekWlini\)](#)
- Date [getRealnaGodzinna\(\)](#)
- void [setRealnaGodzinna\(Date realnaGodzinna\)](#)

6.70.1 Opis szczegółowy

Klasa reprezentująca encję [PrzejazdKursPrzystanekWlini](#).

Definicja w linii 11 pliku [PrzejazdKursPrzystanekWlini.java](#).

6.70.2 Dokumentacja funkcji składowych

6.70.2.1 [getKursPrzystanekWlini\(\)](#)

`KursPrzystanekWlini com.mklachl.sopkom.model.entity.PrzejazdKursPrzystanekWlini.getKursPrzystanekWlini()`

Zwraca [KursPrzystanekWlini](#) zвязany z [PrzejazdKursPrzystanekWlini](#).

Zwrota

[kursPrzystanekWlini KursPrzystanekWlini](#) zвязany z [PrzejazdKursPrzystanekWlini](#).

Definicja w linii 77 pliku [PrzejazdKursPrzystanekWlini.java](#).

6.70.2.2 [getPrzejazd\(\)](#)

`Przejazd com.mklachl.sopkom.model.entity.PrzejazdKursPrzystanekWlini.getPrzejazd()`

Zwraca przejazd zвязany z [KursPrzystanekWlini](#).

Zwrota

[przejazd Przejazd](#) zвязany z [KursPrzystanekWlini](#).

Definicja w linii 61 pliku [PrzejazdKursPrzystanekWlini.java](#).

6.70.2.3 getPrzejazdKursPrzystanekWliniId()

```
Long com.mklachl.sopkom.model.entity.PrzejazdKursPrzystanekWlini.getPrzejazdKursPrzystanekWliniId ( )
```

Zwraca identyfikator [PrzejazdKursPrzystanekWlini](#).

Zwrota

przejazdKursPrzystanekWliniId Identyfikator [PrzejazdKursPrzystanekWlini](#).

Definicja w linii 45 pliku [PrzejazdKursPrzystanekWlini.java](#).

6.70.2.4 getRealnaGodzinna()

```
Date com.mklachl.sopkom.model.entity.PrzejazdKursPrzystanekWlini.getRealnaGodzinna ( )
```

Zwraca realną godzinę przystanku w linii.

Zwrota

realnaGodzinna Realna godzina przystanku w linii.

Definicja w linii 93 pliku [PrzejazdKursPrzystanekWlini.java](#).

6.70.2.5 setKursPrzystanekWlini()

```
void com.mklachl.sopkom.model.entity.PrzejazdKursPrzystanekWlini.setKursPrzystanekWlini ( KursPrzystanekWlini kursPrzystanekWlini )
```

Ustawia [KursPrzystanekWlini](#) związany z [PrzejazdKursPrzystanekWlini](#).

Parametry

kursPrzystanekWlini	KursPrzystanekWlini związany z PrzejazdKursPrzystanekWlini .
-------------------------------------	--

Definicja w linii 85 pliku [PrzejazdKursPrzystanekWlini.java](#).

6.70.2.6 setPrzejazd()

```
void com.mklachl.sopkom.model.entity.PrzejazdKursPrzystanekWlini.setPrzejazd ( Przejazd przejazd )
```

Ustawia przejazd związany z [KursPrzystanekWlini](#).

Parametry

<i>przejazd</i>	Przejazd związany z KursPrzystanekWlini.
-----------------	--

Definicja w linii 69 pliku [PrzejazdKursPrzystanekWlini.java](#).

6.70.2.7 setPrzejazdKursPrzystanekWliniId()

```
void com.mklachl.sopkom.model.entity.PrzejazdKursPrzystanekWlini.setPrzejazdKursPrzystanekWliniId ( Long przejazdKursPrzystanekWliniId )
```

Ustawia identyfikator [PrzejazdKursPrzystanekWlini](#).

Parametry

<i>przejazdKursPrzystanekWliniId</i>	Identyfikator PrzejazdKursPrzystanekWlini .
--------------------------------------	---

Definicja w linii 53 pliku [PrzejazdKursPrzystanekWlini.java](#).

6.70.2.8 setRealnaGodzinna()

```
void com.mklachl.sopkom.model.entity.PrzejazdKursPrzystanekWlini.setRealnaGodzinna ( Date realnaGodzinna )
```

Ustawia realną godzinę przystanku w linii.

Parametry

<i>realnaGodzinna</i>	Realna godzina przystanku w linii.
-----------------------	------------------------------------

Definicja w linii 101 pliku [PrzejazdKursPrzystanekWlini.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.71 Dokumentacja klasy com.mklachl.sopkom.model.dto.PrzejazdKursPrzystanekWliniDto

Metody publiczne

- [PrzejazdKursPrzystanekWliniDto \(\)](#)

- `PrzejazdKursPrzystanekWliniDto` (`PrzejazdKursPrzystanekWlini` przejazdKursPrzystanekWlini)
- Long `getPrzejazdKursPrzystanekWliniId` ()
- void `setPrzejazdKursPrzystanekWliniId` (Long przejazdKursPrzystanekWliniId)
- int `getKolejnosc` ()
- void `setKolejnosc` (int kolejnosc)
- Date `getPrzewidywanaGodzinna` ()
- void `setPrzewidywanaGodzinna` (Date przewidywanaGodzinna)
- Date `getRealnaGodzinna` ()
- void `setRealnaGodzinna` (Date realnaGodzinna)
- String `getNazwa` ()
- void `setNazwa` (String nazwa)
- Long `getPrzystanekId` ()
- void `setPrzystanekId` (Long przystanekId)

6.71.1 Opis szczegółowy

Data Transfer Object (DTO) dla encji `PrzejazdKursPrzystanekWlini`.

Definicja w linii 10 pliku [PrzejazdKursPrzystanekWliniDto.java](#).

6.71.2 Dokumentacja konstruktora i destruktora

6.71.2.1 `PrzejazdKursPrzystanekWliniDto()` [1/2]

```
com.mklachl.sopkom.model.dto.PrzejazdKursPrzystanekWliniDto.PrzejazdKursPrzystanekWliniDto ( )
```

Konstruktor domyślny.

Definicja w linii 47 pliku [PrzejazdKursPrzystanekWliniDto.java](#).

6.71.2.2 `PrzejazdKursPrzystanekWliniDto()` [2/2]

```
com.mklachl.sopkom.model.dto.PrzejazdKursPrzystanekWliniDto.PrzejazdKursPrzystanekWliniDto (   
    PrzejazdKursPrzystanekWlini przejazdKursPrzystanekWlini )
```

Konstruktor przyjmujący obiekt encji `PrzejazdKursPrzystanekWlini`.

Parametry

<code>przejazdKursPrzystanekWlini</code>	Obiekt encji <code>PrzejazdKursPrzystanekWlini</code> .
--	---

Definicja w linii 54 pliku [PrzejazdKursPrzystanekWliniDto.java](#).

6.71.3 Dokumentacja funkcji składowych

6.71.3.1 getKolejnosc()

```
int com.mklachl.sopkom.model.dto.PrzejazdKursPrzystanekWliniDto.getKolejnosc ( )
```

Zwraca kolejność przystanku.

Zwrota

kolejnosc Kolejność przystanku.

Definicja w linii 85 pliku [PrzejazdKursPrzystanekWliniDto.java](#).

6.71.3.2 getNazwa()

```
String com.mklachl.sopkom.model.dto.PrzejazdKursPrzystanekWliniDto.getNazwa ( )
```

Zwraca nazwę przystanku.

Zwrota

nazwa Nazwa przystanku.

Definicja w linii 133 pliku [PrzejazdKursPrzystanekWliniDto.java](#).

6.71.3.3 getPrzejazdKursPrzystanekWliniId()

```
Long com.mklachl.sopkom.model.dto.PrzejazdKursPrzystanekWliniDto.getPrzejazdKursPrzystanekWliniId ( )
```

Zwraca identyfikator PrzejazdKursPrzystanekWlini.

Zwrota

przejazdKursPrzystanekWliniId Identyfikator PrzejazdKursPrzystanekWlini.

Definicja w linii 69 pliku [PrzejazdKursPrzystanekWliniDto.java](#).

6.71.3.4 getPrzewidywanaGodzinna()

```
Date com.mklachl.sopkom.model.dto.PrzejazdKursPrzystanekWliniDto.getPrzewidywanaGodzinna ( )
```

Zwraca przewidywaną godzinę przystanku.

Zwraca

przewidywanaGodzinna Przewidywana godzina przystanku.

Definicja w linii 101 pliku [PrzejazdKursPrzystanekWliniDto.java](#).

6.71.3.5 getPrzystanekId()

```
Long com.mklachl.sopkom.model.dto.PrzejazdKursPrzystanekWliniDto.getPrzystanekId ( )
```

Zwraca identyfikator przystanku.

Zwraca

przystanekId Identyfikator przystanku.

Definicja w linii 149 pliku [PrzejazdKursPrzystanekWliniDto.java](#).

6.71.3.6 getRealnaGodzinna()

```
Date com.mklachl.sopkom.model.dto.PrzejazdKursPrzystanekWliniDto.getRealnaGodzinna ( )
```

Zwraca realną godzinę przystanku.

Zwraca

realnaGodzinna Realna godzina przystanku.

Definicja w linii 117 pliku [PrzejazdKursPrzystanekWliniDto.java](#).

6.71.3.7 setKolejnosc()

```
void com.mklachl.sopkom.model.dto.PrzejazdKursPrzystanekWliniDto.setKolejnosc ( int kolejnosc )
```

Ustawia kolejność przystanku.

Parametry

kolejnosc	Kolejność przystanku.
-----------	-----------------------

Definicja w linii 93 pliku [PrzejazdKursPrzystanekWliniDto.java](#).

6.71.3.8 setNazwa()

```
void com.mklachl.sopkom.model.dto.PrzejazdKursPrzystanekWliniDto.setNazwa ( String nazwa )
```

Ustawia nazwę przystanku.

Parametry

nazwa	Nazwa przystanku.
-------	-------------------

Definicja w linii 141 pliku [PrzejazdKursPrzystanekWliniDto.java](#).

6.71.3.9 setPrzejazdKursPrzystanekWliniId()

```
void com.mklachl.sopkom.model.dto.PrzejazdKursPrzystanekWliniDto.setPrzejazdKursPrzystanekWliniId ( Long przejazdKursPrzystanekWliniId )
```

Ustawia identyfikator PrzejazdKursPrzystanekWlini.

Parametry

przejazdKursPrzystanekWliniId	Identyfikator PrzejazdKursPrzystanekWlini.
-------------------------------	--

Definicja w linii 77 pliku [PrzejazdKursPrzystanekWliniDto.java](#).

6.71.3.10 setPrzewidywanaGodzinna()

```
void com.mklachl.sopkom.model.dto.PrzejazdKursPrzystanekWliniDto.setPrzewidywanaGodzinna ( Date przewidywanaGodzinna )
```

Ustawia przewidywaną godzinę przystanku.

Parametry

<i>przewidywanaGodzinna</i>	Przewidywana godzina przystanku.
-----------------------------	----------------------------------

Definicja w linii 109 pliku [PrzejazdKursPrzystanekWliniDto.java](#).

6.71.3.11 setPrzystanekId()

```
void com.mklachl.sopkom.model.dto.PrzejazdKursPrzystanekWliniDto.setPrzystanekId ( Long przystanekId )
```

Ustawia identyfikator przystanku.

Parametry

<i>przystanek</i> ↪ <i>Id</i>	Identyfikator przystanku.
----------------------------------	---------------------------

Definicja w linii 157 pliku [PrzejazdKursPrzystanekWliniDto.java](#).

6.71.3.12 setRealnaGodzinna()

```
void com.mklachl.sopkom.model.dto.PrzejazdKursPrzystanekWliniDto.setRealnaGodzinna ( Date realnaGodzinna )
```

Ustawia realną godzinę przystanku.

Parametry

<i>realnaGodzinna</i>	Realna godzina przystanku.
-----------------------	----------------------------

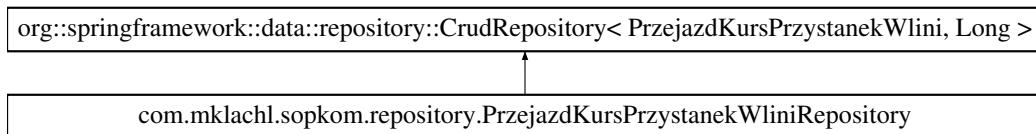
Definicja w linii 125 pliku [PrzejazdKursPrzystanekWliniDto.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.72 Dokumentacja interfejsu com.mklachl.sopkom.repository.Przejazd ↪ KursPrzystanekWliniRepository

Diagram dziedziczenia dla com.mklachl.sopkom.repository.PrzejazdKursPrzystanekWliniRepository



Metody publiczne

- List< [PrzejazdKursPrzystanekWlini](#) > [findAllByPrzejazd](#) ([Przejazd](#) przejazd)
- List< [PrzejazdKursPrzystanekWlini](#) > [findAllByKursPrzystanekWlini](#) ([KursPrzystanekWlini](#) kursPrzystanek←
Wlini)
- List< [PrzejazdKursPrzystanekWlini](#) > [findAllByRealnaGodzinna](#) (Date realnaGodzinna)
- List< [PrzejazdKursPrzystanekWlini](#) > [findAllByRealnaGodzinnaBefore](#) (Date realnaGodzinna)
- List< [PrzejazdKursPrzystanekWlini](#) > [findAllByRealnaGodzinnaAfter](#) (Date realnaGodzinna)
- List< [PrzejazdKursPrzystanekWlini](#) > [findAll](#) ()

6.72.1 Opis szczegółowy

Interfejs repository dla encji [PrzejazdKursPrzystanekWlini](#)

Definicja w linii 14 pliku [PrzejazdKursPrzystanekWliniRepository.java](#).

6.72.2 Dokumentacja funkcji składowych

6.72.2.1 findAll()

```
List< PrzejazdKursPrzystanekWlini > com.mklachl.sopkom.repository.PrzejazdKursPrzystanek←  
WliniRepository.findAll ( )
```

Znajduje wszystkie kursy przystanku w przejeździe

Zwrota

Lista kursów przystanku w przejeździe

6.72.2.2 findAllByKursPrzystanekWlini()

```
List< PrzejazdKursPrzystanekWlini > com.mklachl.sopkom.repository.PrzejazdKursPrzystanek←  
WliniRepository.findAllByKursPrzystanekWlini ( KursPrzystanekWlini kursPrzystanekWlini )
```

Znajduje wszystkie kursy przystanku w przejeździe dla danego kursu przystanku w linii

Parametry

<i>kursPrzystanekWlini</i>	kurs przystanku w linii
----------------------------	-------------------------

Zwroca

Lista kursów przystanku w przejeździe

6.72.2.3 findAllByPrzejazd()

```
List< PrzejazdKursPrzystanekWlini > com.mklachl.sopkom.repository.PrzejazdKursPrzystanek↔
WliniRepository.findAllByPrzejazd (
    Przejazd przejazd )
```

Znajduje wszystkie kursy przystanku w przejeździe dla danego przejazdu

Parametry

<i>przejazd</i>	przejazd
-----------------	----------

Zwroca

Lista kursów przystanku w przejeździe

6.72.2.4 findAllByRealnaGodzinna()

```
List< PrzejazdKursPrzystanekWlini > com.mklachl.sopkom.repository.PrzejazdKursPrzystanek↔
WliniRepository.findAllByRealnaGodzinna (
    Date realnaGodzinna )
```

Znajduje wszystkie kursy przystanku w przejeździe po rzeczywistej godzinie

Parametry

<i>realnaGodzinna</i>	rzeczywista godzina
-----------------------	---------------------

Zwroca

Lista kursów przystanku w przejeździe

6.72.2.5 findAllByRealnaGodzinnaAfter()

```
List< PrzejazdKursPrzystanekWlini > com.mklachl.sopkom.repository.PrzejazdKursPrzystanekWliniRepository.findAllByRealnaGodzinnaAfter ( Date realnaGodzinna )
```

Znajduje wszystkie kursy przystanku w przejeździe po rzeczywistej godzinie późniejszej niż podana

Parametry

realnaGodzinna	rzeczywista godzina
----------------	---------------------

Zwroca

Lista kursów przystanku w przejeździe

6.72.2.6 findAllByRealnaGodzinnaBefore()

```
List< PrzejazdKursPrzystanekWlini > com.mklachl.sopkom.repository.PrzejazdKursPrzystanekWliniRepository.findAllByRealnaGodzinnaBefore ( Date realnaGodzinna )
```

Znajduje wszystkie kursy przystanku w przejeździe po rzeczywistej godzinie wcześniejszej niż podana

Parametry

realnaGodzinna	rzeczywista godzina
----------------	---------------------

Zwroca

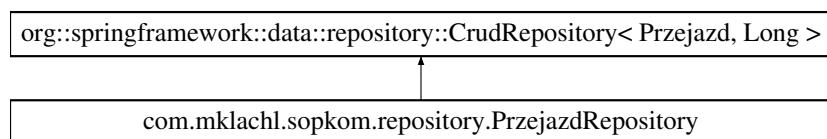
Lista kursów przystanku w przejeździe

Dokumentacja dla tego interfejsu została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/repo

6.73 Dokumentacja interfejsu com.mklachl.sopkom.repository.PrzejazdRepository

Diagram dziedziczenia dla com.mklachl.sopkom.repository.PrzejazdRepository



Metody publiczne

- List< [Przejazd](#) > [findAllByKierowca](#) ([Kierowca](#) kierowca)
- List< [Przejazd](#) > [findAllByAutobus](#) ([Autobus](#) autobus)
- List< [Przejazd](#) > [findAllByKurs](#) ([Kurs](#) kurs)
- List< [Przejazd](#) > [findAllBySpalanie](#) (float spalanie)
- List< [Przejazd](#) > [findAllBySpalanieLessThan](#) (float spalanie)
- List< [Przejazd](#) > [findAllBySpalanieGreaterThanOrEqual](#) (float spalanie)
- List< [Przejazd](#) > [findAllByCenaZaLitr](#) (float cenaZaLitr)
- List< [Przejazd](#) > [findAllByCenaZaLitrLessThan](#) (float cenaZaLitr)
- List< [Przejazd](#) > [findAllByCenaZaLitrGreaterThanOrEqual](#) (float cenaZaLitr)
- List< [Przejazd](#) > [findAllByDlugoscTrasy](#) (float dlugoscTrasy)
- List< [Przejazd](#) > [findAllByDlugoscTrasyLessThan](#) (float dlugoscTrasy)
- List< [Przejazd](#) > [findAllByDlugoscTrasyGreaterThanOrEqual](#) (float dlugoscTrasy)
- List< [Przejazd](#) > [findAllByDataStartu](#) (Date dataStartu)
- List< [Przejazd](#) > [findAllByDataStartuBefore](#) (Date dataStartu)
- List< [Przejazd](#) > [findAllByDataStartuAfter](#) (Date dataStartu)
- List< [Przejazd](#) > [findAllByDataKonca](#) (Date dataKonca)
- List< [Przejazd](#) > [findAllByDataKoncaBefore](#) (Date dataKonca)
- List< [Przejazd](#) > [findAllByDataKoncaAfter](#) (Date dataKonca)
- List< [Przejazd](#) > [findAllByData](#) (Date data)
- List< [Przejazd](#) > [findAllByKursAndData](#) (@Param("kurs") Kurs kurs, @Param("dataPrzejazdu") Date dataPrzejazdu)
- List< [Przejazd](#) > [findAllByData](#) (@Param("kurs") Kurs kurs, @Param("dataPrzejazdu") Date dataPrzejazdu)
- List< [Przejazd](#) > [findAll](#) ()

6.73.1 Opis szczegółowy

Interfejs repository dla encji Przejazd

Definicja w linii 17 pliku [PrzejazdRepository.java](#).

6.73.2 Dokumentacja funkcji składowych

6.73.2.1 findAll()

```
List< Przejazd > com.mklachl.sopkom.repository.PrzejazdRepository.findAll ( )
```

Znajduje wszystkie przejazdy

Zwrota

Lista przejazdów

6.73.2.2 findAllByAutobus()

```
List< Przejazd > com.mklachl.sopkom.repository.PrzejazdRepository.findAllByAutobus (
    Autobus autobus )
```

Znajduje wszystkie przejazdy dla danego autobusu

Parametry

<i>autobus</i>	autobus
----------------	---------

Zwroca

Lista przejazdów

6.73.2.3 findAllByCenaZaLitr()

```
List< Przejazd > com.mklachl.sopkom.repository.PrzejazdRepository.findAllByCenaZaLitr ( float cenaZaLitr )
```

Znajduje wszystkie przejazdy o określonej cenie za litr

Parametry

<i>cenaZaLitr</i>	cena za litr
-------------------	--------------

Zwroca

Lista przejazdów

6.73.2.4 findAllByCenaZaLitrGreaterThanOrEqual()

```
List< Przejazd > com.mklachl.sopkom.repository.PrzejazdRepository.findAllByCenaZaLitrGreaterThanOrEqual ( float cenaZaLitr )
```

Znajduje wszystkie przejazdy o cenie za litr większej niż podana

Parametry

<i>cenaZaLitr</i>	cena za litr
-------------------	--------------

Zwroca

Lista przejazdów

6.73.2.5 findAllByCenaZaLitrLessThan()

```
List< Przejazd > com.mklachl.sopkom.repository.PrzejazdRepository.findAllByCenaZaLitrLessThan  
(  
    float cenaZaLitr )
```

Znajduje wszystkie przejazdy o cenie za litr mniejszej niż podana

Parametry

cenaZaLitr	cena za litr
------------	--------------

Zwroca

Lista przejazdów

6.73.2.6 findAllByData() [1/2]

```
List< Przejazd > com.mklachl.sopkom.repository.PrzejazdRepository.findAllByData (   
    @Param("kurs") Kurs kurs,  
    @Param("dataPrzejazdu") Date dataPrzejazdu )
```

Znajduje wszystkie przejazdy dla danej daty przejazdu

Parametry

kurs	kurs
dataPrzejazdu	data przejazdu

Zwroca

Lista przejazdów

6.73.2.7 findAllByData() [2/2]

```
List< Przejazd > com.mklachl.sopkom.repository.PrzejazdRepository.findAllByData (   
    Date data )
```

Znajduje wszystkie przejazdy po dokładnej dacie

Parametry

data	data
------	------

Zwroca

Lista przejazdów

6.73.2.8 findAllByDataKonca()

```
List< Przejazd > com.mklachl.sopkom.repository.PrzejazdRepository.findAllByDataKonca ( Date dataKonca )
```

Znajduje wszystkie przejazdy po dacie zakończenia

Parametry

<i>dataKonca</i>	data zakończenia
------------------	------------------

Zwroca

Lista przejazdów

6.73.2.9 findAllByDataKoncaAfter()

```
List< Przejazd > com.mklachl.sopkom.repository.PrzejazdRepository.findAllByDataKoncaAfter ( Date dataKonca )
```

Znajduje wszystkie przejazdy po dacie zakończenia późniejszej niż podana

Parametry

<i>dataKonca</i>	data zakończenia
------------------	------------------

Zwroca

Lista przejazdów

6.73.2.10 findAllByDataKoncaBefore()

```
List< Przejazd > com.mklachl.sopkom.repository.PrzejazdRepository.findAllByDataKoncaBefore ( Date dataKonca )
```

Znajduje wszystkie przejazdy po dacie zakończenia wcześniejszej niż podana

Parametry

<code>dataKonca</code>	data zakończenia
------------------------	------------------

Zwrota

Lista przejazdów

6.73.2.11 findAllByDataStartu()

```
List< Przejazd > com.mklachl.sopkom.repository.PrzejazdRepository.findAllByDataStartu ( Date dataStartu )
```

Znajduje wszystkie przejazdy po dacie rozpoczęcia

Parametry

<code>dataStartu</code>	data rozpoczęcia
-------------------------	------------------

Zwrota

Lista przejazdów

6.73.2.12 findAllByDataStartuAfter()

```
List< Przejazd > com.mklachl.sopkom.repository.PrzejazdRepository.findAllByDataStartuAfter ( Date dataStartu )
```

Znajduje wszystkie przejazdy po dacie rozpoczęcia późniejszej niż podana

Parametry

<code>dataStartu</code>	data rozpoczęcia
-------------------------	------------------

Zwrota

Lista przejazdów

6.73.2.13 findAllByDataStartuBefore()

```
List< Przejazd > com.mklachl.sopkom.repository.PrzejazdRepository.findAllByDataStartuBefore ( Date dataStartu )
```

Znajduje wszystkie przejazdy po dacie rozpoczęcia wcześniejszej niż podana

Parametry

<code>dataStartu</code>	data rozpoczęcia
-------------------------	------------------

Zwroca

Lista przejazdów

6.73.2.14 `findAllByDlugoscTrasy()`

```
List< Przejazd > com.mklachl.sopkom.repository.PrzejazdRepository.findAllByDlugoscTrasy ( float dlugoscTrasy )
```

Znajduje wszystkie przejazdy o określonej długości trasy

Parametry

<code>dlugoscTrasy</code>	długość trasy
---------------------------	---------------

Zwroca

Lista przejazdów

6.73.2.15 `findAllByDlugoscTrasyGreaterThan()`

```
List< Przejazd > com.mklachl.sopkom.repository.PrzejazdRepository.findAllByDlugoscTrasy<-- GreaterThan ( float dlugoscTrasy )
```

Znajduje wszystkie przejazdy o długości trasy większej niż podana

Parametry

<code>dlugoscTrasy</code>	długość trasy
---------------------------	---------------

Zwroca

Lista przejazdów

6.73.2.16 findAllByDlugoscTrasyLessThan()

```
List< Przejazd > com.mklachl.sopkom.repository.PrzejazdRepository.findAllByDlugoscTrasyLessThan ( float dlugoscTrasy )
```

Znajduje wszystkie przejazdy o długości trasy mniejszej niż podana

Parametry

<i>dlugoscTrasy</i>	długość trasy
---------------------	---------------

Zwroca

Lista przejazdów

6.73.2.17 findAllByKierowca()

```
List< Przejazd > com.mklachl.sopkom.repository.PrzejazdRepository.findAllByKierowca ( Kierowca kierowca )
```

Znajduje wszystkie przejazdy dla danego kierowcy

Parametry

<i>kierowca</i>	kierowca
-----------------	----------

Zwroca

Lista przejazdów

6.73.2.18 findAllByKurs()

```
List< Przejazd > com.mklachl.sopkom.repository.PrzejazdRepository.findAllByKurs ( Kurs kurs )
```

Znajduje wszystkie przejazdy dla danego kursu

Parametry

<i>kurs</i>	kurs
-------------	------

Zwroca

Lista przejazdów

6.73.2.19 `findAllByKursAndData()`

```
List< Przejazd > com.mklachl.sopkom.repository.PrzejazdRepository.findAllByKursAndData (   
    @Param("kurs") Kurs kurs,  
    @Param("dataPrzejazdu") Date dataPrzejazdu )
```

Znajduje wszystkie przejazdy dla danego kursu i daty przejazdu

Parametry

<i>kurs</i>	kurs
<i>dataPrzejazdu</i>	data przejazdu

Zwroca

Lista przejazdów

6.73.2.20 `findAllBySpalanie()`

```
List< Przejazd > com.mklachl.sopkom.repository.PrzejazdRepository.findAllBySpalanie (   
    float spalanie )
```

Znajduje wszystkie przejazdy o określonym spalaniu

Parametry

<i>spalanie</i>	spalanie
-----------------	----------

Zwroca

Lista przejazdów

6.73.2.21 `findAllBySpalanieGreaterThanOrEqual()`

```
List< Przejazd > com.mklachl.sopkom.repository.PrzejazdRepository.findAllBySpalanieGreaterThanOrEqual (   
    float spalanie )
```

Znajduje wszystkie przejazdy o spalaniu większym niż podane

Parametry

<i>spalanie</i>	spalanie
-----------------	----------

Zwroca

Lista przejazdów

6.73.2.22 findAllBySpalanieLessThan()

```
List< Przejazd > com.mklachl.sopkom.repository.PrzejazdRepository.findAllBySpalanieLessThan ( float spalanie )
```

Znajduje wszystkie przejazdy o spalaniu mniejszym niż podane

Parametry

<i>spalanie</i>	spalanie
-----------------	----------

Zwroca

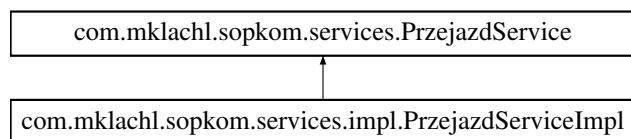
Lista przejazdów

Dokumentacja dla tego interfejsu została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/rep

6.74 Dokumentacja interfejsu com.mklachl.sopkom.services.PrzejazdService

Diagram dziedziczenia dla com.mklachl.sopkom.services.PrzejazdService



Metody publiczne

- Przejazd savePrzejazd (Przejazd przejazd)
- Przejazd savePrzejazd (PrzejazdDtoInput przejazdDto)
- Przejazd updatePrzejazd (PrzejazdDtoUpdate przejazdDto) throws Exception
- List< Przejazd > findPrzejazdByKierowca (Kierowca kierowca)

- List< [Przejazd](#) > [findPrzejazdByAutobus](#) ([Autobus](#) autobus)
- List< [Przejazd](#) > [findPrzejazdByKurs](#) ([Kurs](#) kurs)
- List< [Przejazd](#) > [findPrzejazdBySpalanie](#) (float spalanie)
- List< [Przejazd](#) > [findPrzejazdBySpalanieLessThan](#) (float spalanie)
- List< [Przejazd](#) > [findPrzejazdBySpalanieGreaterThan](#) (float spalanie)
- List< [Przejazd](#) > [findPrzejazdByCenaZaLit](#) (float cenaZaLit)
- List< [Przejazd](#) > [findPrzejazdByCenaZaLitLessThan](#) (float cenaZaLit)
- List< [Przejazd](#) > [findPrzejazdByCenaZaLitGreaterThan](#) (float cenaZaLit)
- List< [Przejazd](#) > [findPrzejazdByDlugoscTrasy](#) (float dlugoscTrasy)
- List< [Przejazd](#) > [findPrzejazdByDlugoscTrasyLessThan](#) (float dlugoscTrasy)
- List< [Przejazd](#) > [findPrzejazdByDlugoscTrasyGreaterThan](#) (float dlugoscTrasy)
- List< [Przejazd](#) > [findPrzejazdByDataStartu](#) (Date dataStartu)
- List< [Przejazd](#) > [findPrzejazdByDataStartuBefore](#) (Date dataStartu)
- List< [Przejazd](#) > [findPrzejazdByDataStartuAfter](#) (Date dataStartu)
- List< [Przejazd](#) > [findPrzejazdByDataKonca](#) (Date dataKonca)
- List< [Przejazd](#) > [findPrzejazdByDataKoncaBefore](#) (Date dataKonca)
- List< [Przejazd](#) > [findPrzejazdByDataKoncaAfter](#) (Date dataKonca)
- List< [Przejazd](#) > [findPrzejazdyByKursAndData](#) ([Kurs](#) kurs, Date data)
- Optional< [Przejazd](#) > [findPrzejazdById](#) (long id)
- List< Date > [findAllDatesForKurs](#) ([Kurs](#) kurs, int count)
- [Przejazd createDetachedPrzejazd](#) ([Kurs](#) kurs, Date dataPrzejazdu)

6.74.1 Opis szczegółowy

Interfejs serwisu dla obsługi przejazdów

Definicja w linii 17 pliku [PrzejazdService.java](#).

6.74.2 Dokumentacja funkcji składowych

6.74.2.1 [createDetachedPrzejazd\(\)](#)

```
Przejazd com.mklachl.sopkom.services.PrzejazdService.createDetachedPrzejazd (
    Kurs kurs,
    Date dataPrzejazdu )
```

Tworzy odłączony przejazd na podstawie kursu i daty przejazdu

Parametry

<i>kurs</i>	kurs
<i>dataPrzejazdu</i>	data przejazdu

Zwraca

odłączony przejazd

Implementowany w [com.mklachl.sopkom.services.impl.PrzejazdServiceImpl](#).

6.74.2.2 findAllDatesForKurs()

```
List< Date > com.mklachl.sopkom.services.PrzejazdService.findAllDatesForKurs (   
    Kurs kurs,  
    int count )
```

Znajduje wszystkie daty dla kursu

Parametry

<i>kurs</i>	kurs
<i>count</i>	liczba dat do znalezienia

Zwroca

lista dat

Implementowany w [com.mklachl.sopkom.services.impl.PrzejazdServiceImpl](#).

6.74.2.3 findPrzejazdByAutobus()

```
List< Przejazd > com.mklachl.sopkom.services.PrzejazdService.findPrzejazdByAutobus (   
    Autobus autobus )
```

Znajduje przejazdy po autobusie

Parametry

<i>autobus</i>	autobus
----------------	---------

Zwroca

lista przejazdów

Implementowany w [com.mklachl.sopkom.services.impl.PrzejazdServiceImpl](#).

6.74.2.4 findPrzejazdByCenaZaLitr()

```
List< Przejazd > com.mklachl.sopkom.services.PrzejazdService.findPrzejazdByCenaZaLitr (   
    float cenaZaLitr )
```

Znajduje przejazdy po cenie za litr paliwa

Parametry

<i>cenaZaLitr</i>	cena za litr
-------------------	--------------

Zwroca

lista przejazdów

Implementowany w [com.mklachl.sopkom.services.impl.PrzejazdServiceImpl](#).**6.74.2.5 findPrzejazdByCenaZaLitrGreaterThanOrEqual()**

```
List< Przejazd > com.mklachl.sopkom.services.PrzejazdService.findPrzejazdByCenaZaLitrGreaterThanOrEqual( float cenaZaLitr )
```

Znajduje przejazdy po cenie za litr paliwa większej niż podana wartość

Parametry

<i>cenaZaLitr</i>	cena za litr
-------------------	--------------

Zwroca

lista przejazdów

Implementowany w [com.mklachl.sopkom.services.impl.PrzejazdServiceImpl](#).**6.74.2.6 findPrzejazdByCenaZaLitrLessThan()**

```
List< Przejazd > com.mklachl.sopkom.services.PrzejazdService.findPrzejazdByCenaZaLitrLessThan( float cenaZaLitr )
```

Znajduje przejazdy po cenie za litr paliwa mniejszej niż podana wartość

Parametry

<i>cenaZaLitr</i>	cena za litr
-------------------	--------------

Zwroca

lista przejazdów

Implementowany w [com.mklachl.sopkom.services.impl.PrzejazdServiceImpl](#).

6.74.2.7 findPrzejazdByDataKonca()

```
List< Przejazd > com.mklachl.sopkom.services.PrzejazdService.findPrzejazdByDataKonca ( Date dataKonca )
```

Znajduje przejazdy po dacie zakończenia

Parametry

dataKonca	data zakończenia
-----------	------------------

Zwrota

lista przejazdów

Implementowany w [com.mklachl.sopkom.services.impl.PrzejazdServiceImpl](#).

6.74.2.8 findPrzejazdByDataKoncaAfter()

```
List< Przejazd > com.mklachl.sopkom.services.PrzejazdService.findPrzejazdByDataKoncaAfter ( Date dataKonca )
```

Znajduje przejazdy po dacie zakończenia późniejszej niż podana wartość

Parametry

dataKonca	data zakończenia
-----------	------------------

Zwrota

lista przejazdów

Implementowany w [com.mklachl.sopkom.services.impl.PrzejazdServiceImpl](#).

6.74.2.9 findPrzejazdByDataKoncaBefore()

```
List< Przejazd > com.mklachl.sopkom.services.PrzejazdService.findPrzejazdByDataKoncaBefore ( Date dataKonca )
```

Znajduje przejazdy po dacie zakończenia wcześniejszej niż podana wartość

Parametry

<i>dataKonca</i>	data zakończenia
------------------	------------------

Zwrota

lista przejazdów

Implementowany w [com.mklachl.sopkom.services.impl.PrzejazdServiceImpl](#).**6.74.2.10 findPrzejazdByDataStartu()**

```
List< Przejazd > com.mklachl.sopkom.services.PrzejazdService.findPrzejazdByDataStartu ( Date dataStartu )
```

Znajduje przejazdy po dacie rozpoczęcia

Parametry

<i>dataStartu</i>	data rozpoczęcia
-------------------	------------------

Zwrota

lista przejazdów

Implementowany w [com.mklachl.sopkom.services.impl.PrzejazdServiceImpl](#).**6.74.2.11 findPrzejazdByDataStartuAfter()**

```
List< Przejazd > com.mklachl.sopkom.services.PrzejazdService.findPrzejazdByDataStartuAfter ( Date dataStartu )
```

Znajduje przejazdy po dacie rozpoczęcia późniejszej niż podana wartość

Parametry

<i>dataStartu</i>	data rozpoczęcia
-------------------	------------------

Zwrota

lista przejazdów

Implementowany w [com.mklachl.sopkom.services.impl.PrzejazdServiceImpl](#).

6.74.2.12 findPrzejazdByDataStartuBefore()

```
List< Przejazd > com.mklachl.sopkom.services.PrzejazdService.findPrzejazdByDataStartuBefore ( Date dataStartu )
```

Znajduje przejazdy po dacie rozpoczęcia wcześniejszej niż podana wartość

Parametry

<i>dataStartu</i>	data rozpoczęcia
-------------------	------------------

Zwroca

lista przejazdów

Implementowany w [com.mklachl.sopkom.services.impl.PrzejazdServiceImpl](#).

6.74.2.13 findPrzejazdByDlugoscTrasy()

```
List< Przejazd > com.mklachl.sopkom.services.PrzejazdService.findPrzejazdByDlugoscTrasy ( float dlugoscTrasy )
```

Znajduje przejazdy po długości trasy

Parametry

<i>dlugoscTrasy</i>	długość trasy
---------------------	---------------

Zwroca

lista przejazdów

Implementowany w [com.mklachl.sopkom.services.impl.PrzejazdServiceImpl](#).

6.74.2.14 findPrzejazdByDlugoscTrasyGreaterThanOrEqual()

```
List< Przejazd > com.mklachl.sopkom.services.PrzejazdService.findPrzejazdByDlugoscTrasyGreaterThanOrEqual ( float dlugoscTrasy )
```

Znajduje przejazdy po długości trasy większej niż podana wartość

Parametry

<i>dlugoscTrasy</i>	długość trasy
---------------------	---------------

Zwrota

lista przejazdów

Implementowany w [com.mklachl.sopkom.services.impl.PrzejazdServiceImpl](#).

6.74.2.15 findPrzejazdByDlugoscTrasyLessThan()

```
List< Przejazd > com.mklachl.sopkom.services.PrzejazdService.findPrzejazdByDlugoscTrasyLessThan ( float dlugoscTrasy )
```

Znajduje przejazdy po długości trasy mniejszej niż podana wartość

Parametry

<i>dlugoscTrasy</i>	długość trasy
---------------------	---------------

Zwrota

lista przejazdów

Implementowany w [com.mklachl.sopkom.services.impl.PrzejazdServiceImpl](#).

6.74.2.16 findPrzejazdById()

```
Optional< Przejazd > com.mklachl.sopkom.services.PrzejazdService.findPrzejazdById ( long id )
```

Znajduje przejazd po ID

Parametry

<i>id</i>	ID przejazdu
-----------	--------------

Zwrota

Optional z przejazdem

Implementowany w [com.mklachl.sopkom.services.impl.PrzejazdServiceImpl](#).

6.74.2.17 findPrzejazdByKierowca()

```
List< Przejazd > com.mklachl.sopkom.services.PrzejazdService.findPrzejazdByKierowca (  
    Kierowca kierowca )
```

Znajduje przejazdy po kierowcy

Parametry

<i>kierowca</i>	<i>kierowca</i>
-----------------	-----------------

Zwrota

lista przejazdów

Implementowany w [com.mklachl.sopkom.services.impl.PrzejazdServiceImpl](#).

6.74.2.18 findPrzejazdByKurs()

```
List< Przejazd > com.mklachl.sopkom.services.PrzejazdService.findPrzejazdByKurs (   
    Kurs kurs )
```

Znajduje przejazdy po kursie

Parametry

<i>kurs</i>	<i>kurs</i>
-------------	-------------

Zwrota

lista przejazdów

Implementowany w [com.mklachl.sopkom.services.impl.PrzejazdServiceImpl](#).

6.74.2.19 findPrzejazdBySpalanie()

```
List< Przejazd > com.mklachl.sopkom.services.PrzejazdService.findPrzejazdBySpalanie (   
    float spalanie )
```

Znajduje przejazdy po spalaniu

Parametry

<i>spalanie</i>	<i>spalanie</i>
-----------------	-----------------

Zwroca

lista przejazdów

Implementowany w [com.mklachl.sopkom.services.impl.PrzejazdServiceImpl](#).

6.74.2.20 findPrzejazdBySpalanieGreaterThan()

```
List< Przejazd > com.mklachl.sopkom.services.PrzejazdService.findPrzejazdBySpalanieGreaterThan  
(  
    float spalanie )
```

Znajduje przejazdy po spalaniu większym niż podana wartość

Parametry

<i>spalanie</i>	spalanie
-----------------	----------

Zwroca

lista przejazdów

Implementowany w [com.mklachl.sopkom.services.impl.PrzejazdServiceImpl](#).

6.74.2.21 findPrzejazdBySpalanieLessThan()

```
List< Przejazd > com.mklachl.sopkom.services.PrzejazdService.findPrzejazdBySpalanieLessThan (   
    float spalanie )
```

Znajduje przejazdy po spalaniu mniejszym niż podana wartość

Parametry

<i>spalanie</i>	spalanie
-----------------	----------

Zwroca

lista przejazdów

Implementowany w [com.mklachl.sopkom.services.impl.PrzejazdServiceImpl](#).

6.74.2.22 findPrzejazdyByKursAndData()

```
List< Przejazd > com.mklachl.sopkom.services.PrzejazdService.findPrzejazdyByKursAndData (   
    Kurs kurs,  
    Date data )
```

Znajduje przejazdy po kursie i dacie

Parametry

<i>kurs</i>	kurs
<i>data</i>	data

Zwroca

lista przejazdów

Implementowany w [com.mklachl.sopkom.services.impl.PrzejazdServiceImpl](#).

6.74.2.23 savePrzejazd() [1/2]

```
Przejazd com.mklachl.sopkom.services.PrzejazdService.savePrzejazd (   
    Przejazd przejazd )
```

Zapisuje nowy przejazd

Parametry

<i>przejazd</i>	encja przejazdu
-----------------	-----------------

Zwroca

zapisany przejazd

Implementowany w [com.mklachl.sopkom.services.impl.PrzejazdServiceImpl](#).

6.74.2.24 savePrzejazd() [2/2]

```
Przejazd com.mklachl.sopkom.services.PrzejazdService.savePrzejazd (   
    PrzejazdDtoInput przejazdDto )
```

Zapisuje nowy przejazd na podstawie DTO

Parametry

<i>przejazdDto</i>	DTO przejazdu
--------------------	---------------

Zwroca

zapisany przejazd

Implementowany w [com.mklachl.sopkom.services.impl.PrzejazdServiceImpl](#).

6.74.2.25 updatePrzejazd()

```
Przejazd com.mklachl.sopkom.services.PrzejazdService.updatePrzejazd (
    PrzejazdDtoUpdate przejazdDto ) throws Exception
```

Aktualizuje przejazd na podstawie DTO

Parametry

<i>przejazdDto</i>	DTO przejazdu
--------------------	---------------

Zwroca

zaktualizowany przejazd

Wyjątki

<i>Exception</i>	wyjątek ogólny
------------------	----------------

Implementowany w [com.mklachl.sopkom.services.impl.PrzejazdServiceImpl](#).

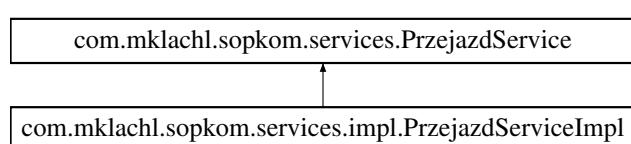
Dokumentacja dla tego interfejsu została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/se

6.75 Dokumentacja klasy

com.mklachl.sopkom.services.impl.PrzejazdServiceImpl

Diagram dziedziczenia dla com.mklachl.sopkom.services.impl.PrzejazdServiceImpl



Metody publiczne

- PrzejazdServiceImpl (KierowcaRepository kierowcaRepository, AutobusRepository autobusRepository, KursRepository kursRepository, PrzejazdRepository przejazdRepository, PrzejazdKursPrzystanekWliniRepository pkpwRepository, PrzejazdBiletRepository pbr, BiletRepository br)
- Przejazd savePrzejazd (Przejazd przejazdDto)
- List< Przejazd > findPrzejazdByKierowca (Kierowca kierowca)
- List< Przejazd > findPrzejazdByAutobus (Autobus autobus)
- List< Przejazd > findPrzejazdByKurs (Kurs kurs)
- List< Przejazd > findPrzejazdBySpalanie (float spalanie)
- List< Przejazd > findPrzejazdBySpalanieLessThan (float spalanie)
- List< Przejazd > findPrzejazdBySpalanieGreaterThan (float spalanie)
- List< Przejazd > findPrzejazdByCenaZaLitr (float cenaZaLitr)
- List< Przejazd > findPrzejazdByCenaZaLitrLessThan (float cenaZaLitr)
- List< Przejazd > findPrzejazdByCenaZaLitrGreaterThan (float cenaZaLitr)
- List< Przejazd > findPrzejazdByDlugoscTrasy (float dlugoscTrasy)
- List< Przejazd > findPrzejazdByDlugoscTrasyLessThan (float dlugoscTrasy)
- List< Przejazd > findPrzejazdByDlugoscTrasyGreaterThan (float dlugoscTrasy)
- List< Przejazd > findPrzejazdByDataStartu (Date dataStartu)
- List< Przejazd > findPrzejazdByDataStartuBefore (Date dataStartu)
- List< Przejazd > findPrzejazdByDataStartuAfter (Date dataStartu)
- List< Przejazd > findPrzejazdByDataKonca (Date dataKonca)
- List< Przejazd > findPrzejazdByDataKoncaBefore (Date dataKonca)
- List< Przejazd > findPrzejazdByDataKoncaAfter (Date dataKonca)
- Optional< Przejazd > findPrzejazdById (long id)
- List< Przejazd > findPrzejazdyByKursAndData (Kurs kurs, Date data)
- Przejazd createDetachedPrzejazd (Kurs kurs, Date dataPrzejazdu)
- Przejazd savePrzejazd (PrzejazdDtoInput przejazdDto)
- Przejazd updatePrzejazd (PrzejazdDtoUpdate przejazdDto) throws Exception
- List< Date > findAllDatesForKurs (Kurs kurs, int maxCount)

6.75.1 Opis szczegółowy

Definicja w linii 39 pliku [PrzejazdServiceImpl.java](#).

6.75.2 Dokumentacja konstruktora i destruktora

6.75.2.1 PrzejazdServiceImpl()

```
com.mklachl.sopkom.services.impl.PrzejazdServiceImpl.PrzejazdServiceImpl (
    KierowcaRepository kierowcaRepository,
    AutobusRepository autobusRepository,
    KursRepository kursRepository,
    PrzejazdRepository przejazdRepository,
    PrzejazdKursPrzystanekWliniRepository pkpwRepository,
    PrzejazdBiletRepository pbr,
    BiletRepository br )
```

Definicja w linii 56 pliku [PrzejazdServiceImpl.java](#).

6.75.3 Dokumentacja funkcji składowych

6.75.3.1 createDetachedPrzejazd()

```
Przejazd com.mklachl.sopkom.services.impl.PrzejazdServiceImpl.createDetachedPrzejazd (
    Kurs kurs,
    Date dataPrzejazdu )
```

Tworzy odłączony przejazd na podstawie kursu i daty przejazdu

Parametry

<i>kurs</i>	kurs
<i>dataPrzejazdu</i>	data przejazdu

Zwroca

odłączony przejazd

Implementuje [com.mklachl.sopkom.services.PrzejazdService](#).

Definicja w linii 171 pliku [PrzejazdServiceImpl.java](#).

6.75.3.2 findAllDatesForKurs()

```
List< Date > com.mklachl.sopkom.services.impl.PrzejazdServiceImpl.findAllDatesForKurs (
    Kurs kurs,
    int count )
```

Znajduje wszystkie daty dla kursu

Parametry

<i>kurs</i>	kurs
<i>count</i>	liczba dat do znalezienia

Zwroca

lista dat

Implementuje [com.mklachl.sopkom.services.PrzejazdService](#).

Definicja w linii 390 pliku [PrzejazdServiceImpl.java](#).

6.75.3.3 findPrzejazdByAutobus()

```
List< Przejazd > com.mklachl.sopkom.services.impl.PrzejazdServiceImpl.findPrzejazdByAutobus ( Autobus autobus )
```

Znajduje przejazdy po autobusie

Parametry

autobus	autobus
---------	---------

Zwroca

lista przejazdów

Implementuje [com.mklachl.sopkom.services.PrzejazdService](#).

Definicja w linii 114 pliku [PrzejazdServiceImpl.java](#).

6.75.3.4 findPrzejazdByCenaZaLitr()

```
List< Przejazd > com.mklachl.sopkom.services.impl.PrzejazdServiceImpl.findPrzejazdByCenaZaLitr ( float cenaZaLitr )
```

Znajduje przejazdy po cenie za litr paliwa

Parametry

cenaZaLitr	cena za litr
------------	--------------

Zwroca

lista przejazdów

Implementuje [com.mklachl.sopkom.services.PrzejazdService](#).

Definicja w linii 129 pliku [PrzejazdServiceImpl.java](#).

6.75.3.5 findPrzejazdByCenaZaLitrGreaterThanOrEqual()

```
List< Przejazd > com.mklachl.sopkom.services.impl.PrzejazdServiceImpl.findPrzejazdByCenaZaLitrGreaterThanOrEqual ( float cenaZaLitr )
```

Znajduje przejazdy po cenie za litr paliwa większej niż podana wartość

Parametry

<i>cenaZaLitr</i>	cena za litr
-------------------	--------------

Zwroca

lista przejazdów

Implementuje [com.mklachl.sopkom.services.PrzejazdService](#).Definicja w linii [135](#) pliku [PrzejazdServiceImpl.java](#).**6.75.3.6 findPrzejazdByCenaZaLitrLessThan()**

```
List< Przejazd > com.mklachl.sopkom.services.impl.PrzejazdServiceImpl.findPrzejazdByCenaZaLitrLessThan (
    float cenaZaLitr )
```

Znajduje przejazdy po cenie za litr paliwa mniejszej niż podana wartość

Parametry

<i>cenaZaLitr</i>	cena za litr
-------------------	--------------

Zwroca

lista przejazdów

Implementuje [com.mklachl.sopkom.services.PrzejazdService](#).Definicja w linii [132](#) pliku [PrzejazdServiceImpl.java](#).**6.75.3.7 findPrzejazdByDataKonca()**

```
List< Przejazd > com.mklachl.sopkom.services.impl.PrzejazdServiceImpl.findPrzejazdByDataKonca (
( Date dataKonca )
```

Znajduje przejazdy po dacie zakończenia

Parametry

<i>dataKonca</i>	data zakończenia
------------------	------------------

Zwrota

lista przejazdów

Implementuje [com.mklachl.sopkom.services.PrzejazdService](#).

Definicja w linii 156 pliku [PrzejazdServiceImpl.java](#).

6.75.3.8 findPrzejazdByDataKoncaAfter()

```
List< Przejazd > com.mklachl.sopkom.services.impl.PrzejazdServiceImpl.findPrzejazdByDataKoncaAfter (
    Date dataKonca )
```

Znajduje przejazdy po dacie zakończenia późniejszej niż podana wartość

Parametry

<code>dataKonca</code>	data zakończenia
------------------------	------------------

Zwrota

lista przejazdów

Implementuje [com.mklachl.sopkom.services.PrzejazdService](#).

Definicja w linii 162 pliku [PrzejazdServiceImpl.java](#).

6.75.3.9 findPrzejazdByDataKoncaBefore()

```
List< Przejazd > com.mklachl.sopkom.services.impl.PrzejazdServiceImpl.findPrzejazdByDataKoncaBefore (
    Date dataKonca )
```

Znajduje przejazdy po dacie zakończenia wcześniejszej niż podana wartość

Parametry

<code>dataKonca</code>	data zakończenia
------------------------	------------------

Zwrota

lista przejazdów

Implementuje [com.mklachl.sopkom.services.PrzejazdService](#).

Definicja w linii 159 pliku [PrzejazdServiceImpl.java](#).

6.75.3.10 findPrzejazdByDataStartu()

```
List< Przejazd > com.mklachl.sopkom.services.impl.PrzejazdServiceImpl.findPrzejazdByDataStartu
(
    Date dataStartu )
```

Znajduje przejazdy po dacie rozpoczęcia

Parametry

<i>dataStartu</i>	data rozpoczęcia
-------------------	------------------

Zwroca

lista przejazdów

Implementuje [com.mklachl.sopkom.services.PrzejazdService](#).

Definicja w linii 147 pliku [PrzejazdServiceImpl.java](#).

6.75.3.11 findPrzejazdByDataStartuAfter()

```
List< Przejazd > com.mklachl.sopkom.services.impl.PrzejazdServiceImpl.findPrzejazdByDataStartuAfter
(
    Date dataStartu )
```

Znajduje przejazdy po dacie rozpoczęcia późniejszej niż podana wartość

Parametry

<i>dataStartu</i>	data rozpoczęcia
-------------------	------------------

Zwroca

lista przejazdów

Implementuje [com.mklachl.sopkom.services.PrzejazdService](#).

Definicja w linii 153 pliku [PrzejazdServiceImpl.java](#).

6.75.3.12 findPrzejazdByDataStartuBefore()

```
List< Przejazd > com.mklachl.sopkom.services.impl.PrzejazdServiceImpl.findPrzejazdByDataStartuBefore
(
    Date dataStartu )
```

Znajduje przejazdy po dacie rozpoczęcia wcześniejszej niż podana wartość

Parametry

<i>dataStartu</i>	data rozpoczęcia
-------------------	------------------

Zwroca

lista przejazdów

Implementuje [com.mklachl.sopkom.services.PrzejazdService](#).Definicja w linii 150 pliku [PrzejazdServiceImpl.java](#).**6.75.3.13 findPrzejazdByDlugoscTrasy()**

```
List< Przejazd > com.mklachl.sopkom.services.impl.PrzejazdServiceImpl.findPrzejazdByDlugoscTrasy ( float dlugoscTrasy )
```

Znajduje przejazdy po długości trasy

Parametry

<i>dlugoscTrasy</i>	długość trasy
---------------------	---------------

Zwroca

lista przejazdów

Implementuje [com.mklachl.sopkom.services.PrzejazdService](#).Definicja w linii 138 pliku [PrzejazdServiceImpl.java](#).**6.75.3.14 findPrzejazdByDlugoscTrasyGreaterThanOrEqual()**

```
List< Przejazd > com.mklachl.sopkom.services.impl.PrzejazdServiceImpl.findPrzejazdByDlugoscTrasyGreaterThanOrEqual ( float dlugoscTrasy )
```

Znajduje przejazdy po długości trasy większej niż podana wartość

Parametry

<i>dlugoscTrasy</i>	długość trasy
---------------------	---------------

Zwrota

lista przejazdów

Implementuje [com.mklachl.sopkom.services.PrzejazdService](#).

Definicja w linii 144 pliku [PrzejazdServiceImpl.java](#).

6.75.3.15 findPrzejazdByDlugoscTrasyLessThan()

```
List< Przejazd > com.mklachl.sopkom.services.impl.PrzejazdServiceImpl.findPrzejazdByDlugoscTrasyLessThan ( float dlugoscTrasy )
```

Znajduje przejazdy po długości trasy mniejszej niż podana wartość

Parametry

<code>dlugoscTrasy</code>	długość trasy
---------------------------	---------------

Zwrota

lista przejazdów

Implementuje [com.mklachl.sopkom.services.PrzejazdService](#).

Definicja w linii 141 pliku [PrzejazdServiceImpl.java](#).

6.75.3.16 findPrzejazdById()

```
Optional< Przejazd > com.mklachl.sopkom.services.impl.PrzejazdServiceImpl.findPrzejazdById ( long id )
```

Znajduje przejazd po ID

Parametry

<code>id</code>	ID przejazdu
-----------------	--------------

Zwrota

Optional z przejazdem

Implementuje [com.mklachl.sopkom.services.PrzejazdService](#).

Definicja w linii 165 pliku [PrzejazdServiceImpl.java](#).

6.75.3.17 findPrzejazdByKierowca()

```
List< Przejazd > com.mklachl.sopkom.services.impl.PrzejazdServiceImpl.findPrzejazdByKierowca (  
    Kierowca kierowca )
```

Znajduje przejazdy po kierowcy

Parametry

<i>kierowca</i>	<i>kierowca</i>
-----------------	-----------------

Zwrota

lista przejazdów

Implementuje [com.mklachl.sopkom.services.PrzejazdService](#).

Definicja w linii 111 pliku [PrzejazdServiceImpl.java](#).

6.75.3.18 findPrzejazdByKurs()

```
List< Przejazd > com.mklachl.sopkom.services.impl.PrzejazdServiceImpl.findPrzejazdByKurs (   
    Kurs kurs )
```

Znajduje przejazdy po kursie

Parametry

<i>kurs</i>	<i>kurs</i>
-------------	-------------

Zwrota

lista przejazdów

Implementuje [com.mklachl.sopkom.services.PrzejazdService](#).

Definicja w linii 117 pliku [PrzejazdServiceImpl.java](#).

6.75.3.19 findPrzejazdBySpalanie()

```
List< Przejazd > com.mklachl.sopkom.services.impl.PrzejazdServiceImpl.findPrzejazdBySpalanie (   
    float spalanie )
```

Znajduje przejazdy po spalaniu

Parametry

<i>spalanie</i>	spalanie
-----------------	----------

Zwroca

lista przejazdów

Implementuje [com.mklachl.sopkom.services.PrzejazdService](#).

Definicja w linii 120 pliku [PrzejazdServiceImpl.java](#).

6.75.3.20 findPrzejazdBySpalanieGreaterThanOrEqual()

```
List< Przejazd > com.mklachl.sopkom.services.impl.PrzejazdServiceImpl.findPrzejazdBySpalanie<=  
GreaterThanOrEqual (   
    float spalanie )
```

Znajduje przejazdy po spalaniu większym niż podana wartość

Parametry

<i>spalanie</i>	spalanie
-----------------	----------

Zwroca

lista przejazdów

Implementuje [com.mklachl.sopkom.services.PrzejazdService](#).

Definicja w linii 126 pliku [PrzejazdServiceImpl.java](#).

6.75.3.21 findPrzejazdBySpalanieLessThan()

```
List< Przejazd > com.mklachl.sopkom.services.impl.PrzejazdServiceImpl.findPrzejazdBySpalanie<<  
LessThan (   
    float spalanie )
```

Znajduje przejazdy po spalaniu mniejszym niż podana wartość

Parametry

<i>spalanie</i>	spalanie
-----------------	----------

Zwrota

lista przejazdów

Implementuje [com.mklachl.sopkom.services.PrzejazdService](#).

Definicja w linii 123 pliku [PrzejazdServiceImpl.java](#).

6.75.3.22 findPrzejazdyByKursAndData()

```
List< Przejazd > com.mklachl.sopkom.services.impl.PrzejazdServiceImpl.findPrzejazdyByKursAnd→  
Data (   
    Kurs kurs,  
    Date data )
```

Znajduje przejazdy po kursie i dacie

Parametry

kurs	kurs
data	data

Zwrota

lista przejazdów

Implementuje [com.mklachl.sopkom.services.PrzejazdService](#).

Definicja w linii 168 pliku [PrzejazdServiceImpl.java](#).

6.75.3.23 savePrzejazd() [1/2]

```
Przejazd com.mklachl.sopkom.services.impl.PrzejazdServiceImpl.savePrzejazd (   
    Przejazd przejazd )
```

Zapisuje nowy przejazd

Parametry

przejazd	encja przejazdu
----------	-----------------

Zwrota

zapisany przejazd

Implementuje [com.mklachl.sopkom.services.PrzejazdService](#).

Definicja w linii 72 pliku [PrzejazdServiceImpl.java](#).

6.75.3.24 savePrzejazd() [2/2]

```
Przejazd com.mklachl.sopkom.services.impl.PrzejazdServiceImpl.savePrzejazd (
    PrzejazdDtoInput przejazdDto )
```

Zapisuje nowy przejazd na podstawie DTO

Parametry

<i>przejazdDto</i>	DTO przejazdu
--------------------	---------------

Zwrota

zapisany przejazd

Implementuje [com.mklachl.sopkom.services.PrzejazdService](#).

Definicja w linii 248 pliku [PrzejazdServiceImpl.java](#).

6.75.3.25 updatePrzejazd()

```
Przejazd com.mklachl.sopkom.services.impl.PrzejazdServiceImpl.updatePrzejazd (
    PrzejazdDtoUpdate przejazdDto ) throws Exception
```

Aktualizuje przejazd na podstawie DTO

Parametry

<i>przejazdDto</i>	DTO przejazdu
--------------------	---------------

Zwrota

zaktualizowany przejazd

Wyjątki

<i>Exception</i>	wyjątek ogólny
------------------	----------------

Implementuje [com.mklachl.sopkom.services.PrzejazdService](#).

Definicja w linii 336 pliku [PrzejazdServiceImpl.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/se

6.76 Dokumentacja klasy com.mklachl.sopkom.model.entity.Przystanek

Metody publiczne

- Long [getPrzystanekId \(\)](#)
- [Przystanek getPrzystanekOdwrótny \(\)](#)
- String [getNazwa \(\)](#)
- String [getKodPocztowy \(\)](#)
- String [getMiasto \(\)](#)
- String [getUlica \(\)](#)
- String [getDlugoscGeograficzna \(\)](#)
- String [getSzerokoscGeograficzna \(\)](#)
- List< [PrzystanekWlini > getPrzystankiWlini \(\)](#)
- void [setPrzystanekId \(Long przystanekId\)](#)
- void [setPrzystanekOdwrótny \(Przystanek przystanekOdwrótny\)](#)
- void [setNazwa \(String nazwa\)](#)
- void [setKodPocztowy \(String kodPocztowy\)](#)
- void [setMiasto \(String miasto\)](#)
- void [setUlica \(String ulica\)](#)
- void [setDlugoscGeograficzna \(String dlugoscGeograficzna\)](#)
- void [setSzerokoscGeograficzna \(String szerokoscGeograficzna\)](#)
- void [setPrzystankiWlini \(List< PrzystanekWlini > przystankiWlini\)](#)

6.76.1 Opis szczegółowy

Klasa reprezentująca encję [Przystanek](#).

Definicja w linii [11](#) pliku [Przystanek.java](#).

6.76.2 Dokumentacja funkcji składowych

6.76.2.1 [getDlugoscGeograficzna\(\)](#)

```
String com.mklachl.sopkom.model.entity.Przystanek.getDlugoscGeograficzna ( )
```

Zwraca długość geograficzną przystanku.

Zwrota

dlugoscGeograficzna Długość geograficzna przystanku.

Definicja w linii [124](#) pliku [Przystanek.java](#).

6.76.2.2 getKodPocztowy()

```
String com.mklachl.sopkom.model.entity.Przystanek.getKodPocztowy ( )
```

Zwraca kod pocztowy przystanku.

Zwraca

kodPocztowy Kod pocztowy przystanku.

Definicja w linii [100](#) pliku [Przystanek.java](#).

6.76.2.3 getMiasto()

```
String com.mklachl.sopkom.model.entity.Przystanek.getMiasto ( )
```

Zwraca miasto, w którym znajduje się przystanek.

Zwraca

miasto Miasto przystanku.

Definicja w linii [108](#) pliku [Przystanek.java](#).

6.76.2.4 getNazwa()

```
String com.mklachl.sopkom.model.entity.Przystanek.getNazwa ( )
```

Zwraca nazwę przystanku.

Zwraca

nazwa Nazwa przystanku.

Definicja w linii [92](#) pliku [Przystanek.java](#).

6.76.2.5 getPrzystanekId()

```
Long com.mklachl.sopkom.model.entity.Przystanek.getPrzystanekId ( )
```

Zwraca identyfikator przystanku.

Zwraca

przystanekId Identyfikator przystanku.

Definicja w linii [76](#) pliku [Przystanek.java](#).

6.76.2.6 getPrzystanekOdwrotny()

```
Przystanek com.mklachl.sopkom.model.entity.Przystanek.getPrzystanekOdwrotny ( )
```

Zwraca przystanek odwrotny.

Zwraca

przystanekOdwrotny [Przystanek](#) odwrotny.

Definicja w linii [84](#) pliku [Przystanek.java](#).

6.76.2.7 getPrzystankiWlini()

```
List< PrzystanekWlini > com.mklachl.sopkom.model.entity.Przystanek.getPrzystankiWlini ( )
```

Zwraca listę przystanków w linii związań z przystankiem.

Zwraca

przystankiWlini Lista przystanków w linii.

Definicja w linii [140](#) pliku [Przystanek.java](#).

6.76.2.8 getSzerokoscGeograficzna()

```
String com.mklachl.sopkom.model.entity.Przystanek.getSzerokoscGeograficzna ( )
```

Zwraca szerokość geograficzną przystanku.

Zwraca

szerokoscGeograficzna Szerokość geograficzna przystanku.

Definicja w linii [132](#) pliku [Przystanek.java](#).

6.76.2.9 getUlica()

```
String com.mklachl.sopkom.model.entity.Przystanek.getUlica ( )
```

Zwraca ulicę, na której znajduje się przystanek.

Zwraca

ulica Ulica przystanku.

Definicja w linii [116](#) pliku [Przystanek.java](#).

6.76.2.10 setDlugoscGeograficzna()

```
void com.mklachl.sopkom.model.entity.Przystanek.setDlugoscGeograficzna ( String dlugoscGeograficzna )
```

Ustawia długość geograficzną przystanku.

Parametry

<i>długośćGeograficzna</i>	Długość geograficzna przystanku.
----------------------------	----------------------------------

Definicja w linii [198](#) pliku [Przystanek.java](#).

6.76.2.11 setKodPocztowy()

```
void com.mklachl.sopkom.model.entity.Przystanek.setKodPocztowy (
    String kodPocztowy )
```

Ustawia kod pocztowy przystanku.

Parametry

<i>kodPocztowy</i>	Kod pocztowy przystanku.
--------------------	--------------------------

Definicja w linii [174](#) pliku [Przystanek.java](#).

6.76.2.12 setMiasto()

```
void com.mklachl.sopkom.model.entity.Przystanek.setMiasto (
    String miasto )
```

Ustawia miasto, w którym znajduje się przystanek.

Parametry

<i>miasto</i>	Miasto przystanku.
---------------	--------------------

Definicja w linii [182](#) pliku [Przystanek.java](#).

6.76.2.13 setNazwa()

```
void com.mklachl.sopkom.model.entity.Przystanek.setNazwa (
    String nazwa )
```

Ustawia nazwę przystanku.

Parametry

<i>nazwa</i>	Nazwa przystanku.
--------------	-------------------

Definicja w linii 166 pliku Przystanek.java.

6.76.2.14 setPrzystanekId()

```
void com.mklachl.sopkom.model.entity.Przystanek.setPrzystanekId (  
    Long przystanekId )
```

Ustawia identyfikator przystanku.

Parametry

<i>przystanek</i> ← <i>Id</i>	Identyfikator przystanku.
----------------------------------	---------------------------

Definicja w linii 150 pliku Przystanek.java.

6.76.2.15 setPrzystanekOdwrotny()

```
void com.mklachl.sopkom.model.entity.Przystanek.setPrzystanekOdwrotny (  
    Przystanek przystanekOdwrotny )
```

Ustawia przystanek odwrotny.

Parametry

<i>przystanekOdwrotny</i>	Przystanek odwrotny.
---------------------------	----------------------

Definicja w linii 158 pliku Przystanek.java.

6.76.2.16 setPrzystankiWlini()

```
void com.mklachl.sopkom.model.entity.Przystanek.setPrzystankiWlini (  
    List< PrzystanekWlini > przystankiWlini )
```

Ustawia listę przystanków w linii związanych z przystankiem.

Parametry

<i>przystankiWlini</i>	Lista przystanków w linii.
------------------------	----------------------------

Definicja w linii 214 pliku Przystanek.java.

6.76.2.17 setSzerokoscGeograficzna()

```
void com.mklachl.sopkom.model.entity.Przystanek.setSzerokoscGeograficzna (
    String szerokoscGeograficzna )
```

Ustawia szerokość geograficzną przystanku.

Parametry

szerokoscGeograficzna	Szerokość geograficzna przystanku.
-----------------------	------------------------------------

Definicja w linii 206 pliku [Przystanek.java](#).

6.76.2.18 setUlica()

```
void com.mklachl.sopkom.model.entity.Przystanek.setUlica (
    String ulica )
```

Ustawia ulicę, na której znajduje się przystanek.

Parametry

ulica	Ulica przystanku.
-------	-------------------

Definicja w linii 190 pliku [Przystanek.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.77 Dokumentacja klasy com.mklachl.sopkom.controller.PrzystanekController

Metody publiczne

- ResponseEntity<?> template ()
- ResponseEntity<?> addPrzystanek (@RequestBody PrzystanekDto przystanekDto)
- ResponseEntity<?> getPrzystanek (@PathVariable("id") Long id)
- ResponseEntity<?> deletePrzystanek (@PathVariable("id") Long id)
- ResponseEntity<?> updatePrzystanek (@PathVariable("id") Long id, @RequestBody PrzystanekDto przystanekDto)
- ResponseEntity<?> getAllPrzystanki ()
- ResponseEntity<?> getAllPrzystankiFull ()
- ResponseEntity<?> getPrzystanekFull (@PathVariable("id") Long id)

6.77.1 Opis szczegółowy

Definicja w linii 20 pliku [PrzystanekController.java](#).

6.77.2 Dokumentacja funkcji składowych

6.77.2.1 addPrzystanek()

```
ResponseEntity<?> com.mklachl.sopkom.controller.PrzystanekController.addPrzystanek ( @RequestBody PrzystanekDto przystanekDto )
```

Endpoint do zapisu nowego przystanku.

Parametry

<i>przystanekDto</i>	DTO przystanku do zapisania
----------------------	-----------------------------

Zwroca

ResponseEntity z zapisanym PrzystanekDto

Definicja w linii 57 pliku [PrzystanekController.java](#).

6.77.2.2 deletePrzystanek()

```
ResponseEntity<?> com.mklachl.sopkom.controller.PrzystanekController.deletePrzystanek ( @PathVariable("id") Long id )
```

Endpoint do usunięcia przystanku na podstawie ID.

Parametry

<i>id</i>	ID przystanku
-----------	---------------

Zwroca

ResponseEntity wskazujący wynik operacji

Definicja w linii 82 pliku [PrzystanekController.java](#).

6.77.2.3 getAllPrzystanki()

```
ResponseEntity<?> com.mklachl.sopkom.controller.PrzystanekController.getAllPrzystanki ( )
```

Endpoint do pobrania wszystkich przystanków.

Zwroca

ResponseEntity z listą PrzystanekDto

Definicja w linii [141](#) pliku [PrzystanekController.java](#).

6.77.2.4 getAllPrzystankiFull()

```
ResponseEntity<?> com.mklachl.sopkom.controller.PrzystanekController.getAllPrzystankiFull ( )
```

Endpoint do pobrania wszystkich przystanków z pełnymi informacjami.

Zwroca

ResponseEntity z listą PrzystanekDtoFull

Definicja w linii [154](#) pliku [PrzystanekController.java](#).

6.77.2.5 getPrzystanek()

```
ResponseEntity<?> com.mklachl.sopkom.controller.PrzystanekController.getPrzystanek ( @PathVariable("id") Long id )
```

Endpoint do pobrania przystanku na podstawie ID.

Parametry

<i>id</i>	ID przystanku
-----------	---------------

Zwroca

ResponseEntity z PrzystanekDto

Definicja w linii [68](#) pliku [PrzystanekController.java](#).

6.77.2.6 getPrzystanekFull()

```
ResponseEntity<?> com.mklachl.sopkom.controller.PrzystanekController.getPrzystanekFull ( @PathVariable("id") Long id )
```

Endpoint do pobrania przystanku z pełnymi informacjami na podstawie ID.

Parametry

<i>id</i>	ID przystanku
-----------	---------------

Zwroca

ResponseType z PrzystanekDtoFull

Definicja w linii 168 pliku [PrzystanekController.java](#).

6.77.2.7 template()

```
ResponseEntity<?> com.mklachl.sopkom.controller.PrzystanekController.template ( )
```

Endpoint do pobrania przykładowego obiektu PrzystanekDto.

Zwroca

ResponseType z przykładowym PrzystanekDto

Definicja w linii 33 pliku [PrzystanekController.java](#).

6.77.2.8 updatePrzystanek()

```
ResponseEntity<?> com.mklachl.sopkom.controller.PrzystanekController.updatePrzystanek ( @PathVariable("id") Long id, @RequestBody PrzystanekDto przystanekDto )
```

Endpoint do aktualizacji przystanku na podstawie ID.

Parametry

<i>id</i>	ID przystanku
<i>przystanekDto</i>	DTO przystanku z zaktualizowanymi polami

Zwraca

ResponseEntity z aktualizowanym PrzystanekDto

Definicja w linii 103 pliku [PrzystanekController.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/com/mklachl/sopkom/controller/PrzystanekController.java

6.78 Dokumentacja klasy `com.mklachl.sopkom.model.dto.PrzystanekDto`

Metody publiczne

- `PrzystanekDto ()`
- `PrzystanekDto (Przystanek przystanek)`
- `Long getPrzystanekId ()`
- `Long getPrzystanekOdwozny ()`
- `String getNazwa ()`
- `String getKodPocztowy ()`
- `String getMiasto ()`
- `String getUlica ()`
- `String getDlugoscGeograficzna ()`
- `String getSzerokoscGeograficzna ()`
- `void setPrzystanekId (Long przystanekId)`
- `void setPrzystanekOdwozny (Long przystanekOdwozny)`
- `String getPrzystanekOdwoznyNazwa ()`
- `void setPrzystanekOdwoznyNazwa (String przystanekOdwoznyNazwa)`
- `void setNazwa (String nazwa)`
- `void setKodPocztowy (String kodPocztowy)`
- `void setMiasto (String miasto)`
- `void setUlica (String ulica)`
- `void setDlugoscGeograficzna (String dlugoscGeograficzna)`
- `void setSzerokoscGeograficzna (String szerokoscGeograficzna)`

6.78.1 Opis szczegółowy

Data Transfer Object (DTO) dla encji Przystanek.

Definicja w linii 8 pliku [PrzystanekDto.java](#).

6.78.2 Dokumentacja konstruktora i destruktora

6.78.2.1 PrzystanekDto() [1/2]

```
com.mklachl.sopkom.model.dto.PrzystanekDto.PrzystanekDto( )
```

Konstruktor domyślny.

Definicja w linii 58 pliku [PrzystanekDto.java](#).

6.78.2.2 PrzystanekDto() [2/2]

```
com.mklachl.sopkom.model.dto.PrzystanekDto.PrzystanekDto(   
    Przystanek przystanek )
```

Konstruktor przyjmujący obiekt encji Przystanek.

Parametry

<i>przystanek</i>	Obiekt encji Przystanek.
-------------------	--------------------------

Definicja w linii 65 pliku [PrzystanekDto.java](#).

6.78.3 Dokumentacja funkcji składowych

6.78.3.1 getDlugoscGeograficzna()

```
String com.mklachl.sopkom.model.dto.PrzystanekDto.getDlugoscGeograficzna( )
```

Zwraca długość geograficzną przystanku.

Zwrota

dlugoscGeograficzna Długość geograficzna przystanku.

Definicja w linii 135 pliku [PrzystanekDto.java](#).

6.78.3.2 getKodPocztowy()

```
String com.mklachl.sopkom.model.dto.PrzystanekDto.getKodPocztowy( )
```

Zwraca kod pocztowy przystanku.

Zwrota

kodPocztowy Kod pocztowy przystanku.

Definicja w linii 111 pliku [PrzystanekDto.java](#).

6.78.3.3 getMiasto()

```
String com.mklachl.sopkom.model.dto.PrzystanekDto.getMiasto ( )
```

Zwraca miasto, w którym znajduje się przystanek.

Zwraca

miasto Miasto przystanku.

Definicja w linii 119 pliku [PrzystanekDto.java](#).

6.78.3.4 getNazwa()

```
String com.mklachl.sopkom.model.dto.PrzystanekDto.getNazwa ( )
```

Zwraca nazwę przystanku.

Zwraca

nazwa Nazwa przystanku.

Definicja w linii 103 pliku [PrzystanekDto.java](#).

6.78.3.5 getPrzystanekId()

```
Long com.mklachl.sopkom.model.dto.PrzystanekDto.getPrzystanekId ( )
```

Zwraca identyfikator przystanku.

Zwraca

przystanekId Identyfikator przystanku.

Definicja w linii 87 pliku [PrzystanekDto.java](#).

6.78.3.6 getPrzystanekOdwrotny()

```
Long com.mklachl.sopkom.model.dto.PrzystanekDto.getPrzystanekOdwrotny ( )
```

Zwraca identyfikator odwrotnego przystanku.

Zwraca

przystanekOdwrotnyId Identyfikator odwrotnego przystanku.

Definicja w linii 95 pliku [PrzystanekDto.java](#).

6.78.3.7 getPrzystanekOdwrotnyNazwa()

```
String com.mklachl.sopkom.model.dto.PrzystanekDto.getPrzystanekOdwrotnyNazwa ( )
```

Zwraca nazwę odwrotnego przystanku.

Zwraca

przystanekOdwrotnyNazwa Nazwa odwrotnego przystanku.

Definicja w linii [169](#) pliku [PrzystanekDto.java](#).

6.78.3.8 getSzerokoscGeograficzna()

```
String com.mklachl.sopkom.model.dto.PrzystanekDto.getSzerokoscGeograficzna ( )
```

Zwraca szerokość geograficzną przystanku.

Zwraca

szerokoscGeograficzna Szerokość geograficzna przystanku.

Definicja w linii [143](#) pliku [PrzystanekDto.java](#).

6.78.3.9 getUlica()

```
String com.mklachl.sopkom.model.dto.PrzystanekDto.getUlica ( )
```

Zwraca ulicę, na której znajduje się przystanek.

Zwraca

ulica Ulica przystanku.

Definicja w linii [127](#) pliku [PrzystanekDto.java](#).

6.78.3.10 setDlugoscGeograficzna()

```
void com.mklachl.sopkom.model.dto.PrzystanekDto.setDlugoscGeograficzna ( String dlugoscGeograficzna )
```

Ustawia długość geograficzną przystanku.

Parametry

<i>długośćGeograficzna</i>	Długość geograficzna przystanku.
----------------------------	----------------------------------

Definicja w linii [217](#) pliku [PrzystanekDto.java](#).

6.78.3.11 setKodPocztowy()

```
void com.mklachl.sopkom.model.dto.PrzystanekDto.setKodPocztowy (
    String kodPocztowy )
```

Ustawia kod pocztowy przystanku.

Parametry

<i>kodPocztowy</i>	Kod pocztowy przystanku.
--------------------	--------------------------

Definicja w linii [193](#) pliku [PrzystanekDto.java](#).

6.78.3.12 setMiasto()

```
void com.mklachl.sopkom.model.dto.PrzystanekDto.setMiasto (
    String miasto )
```

Ustawia miasto, w którym znajduje się przystanek.

Parametry

<i>miasto</i>	Miasto przystanku.
---------------	--------------------

Definicja w linii [201](#) pliku [PrzystanekDto.java](#).

6.78.3.13 setNazwa()

```
void com.mklachl.sopkom.model.dto.PrzystanekDto.setNazwa (
    String nazwa )
```

Ustawia nazwę przystanku.

Parametry

<i>nazwa</i>	Nazwa przystanku.
--------------	-------------------

Definicja w linii 185 pliku [PrzystanekDto.java](#).

6.78.3.14 setPrzystanekId()

```
void com.mklachl.sopkom.model.dto.PrzystanekDto.setPrzystanekId (  
    Long przystanekId )
```

Ustawia identyfikator przystanku.

Parametry

<i>przystanek</i> ← <i>Id</i>	Identyfikator przystanku.
----------------------------------	---------------------------

Definicja w linii 153 pliku [PrzystanekDto.java](#).

6.78.3.15 setPrzystanekOdwrotny()

```
void com.mklachl.sopkom.model.dto.PrzystanekDto.setPrzystanekOdwrotny (   
    Long przystanekOdwrotny )
```

Ustawia identyfikator odwrotnego przystanku.

Parametry

<i>przystanekOdwrotny</i>	Identyfikator odwrotnego przystanku.
---------------------------	--------------------------------------

Definicja w linii 161 pliku [PrzystanekDto.java](#).

6.78.3.16 setPrzystanekOdwrotnyNazwa()

```
void com.mklachl.sopkom.model.dto.PrzystanekDto.setPrzystanekOdwrotnyNazwa (   
    String przystanekOdwrotnyNazwa )
```

Ustawia nazwę odwrotnego przystanku.

Parametry

<i>przystanekOdwrotnyNazwa</i>	Nazwa odwrotnego przystanku.
--------------------------------	------------------------------

Definicja w linii 177 pliku [PrzystanekDto.java](#).

6.78.3.17 setSzerokoscGeograficzna()

```
void com.mklachl.sopkom.model.dto.PrzystanekDto.setSzerokoscGeograficzna ( String szerokoscGeograficzna )
```

Ustawia szerokość geograficzną przystanku.

Parametry

szerokoscGeograficzna	Szerokość geograficzna przystanku.
-----------------------	------------------------------------

Definicja w linii [225](#) pliku [PrzystanekDto.java](#).

6.78.3.18 setUlica()

```
void com.mklachl.sopkom.model.dto.PrzystanekDto.setUlica ( String ulica )
```

Ustawia ulicę, na której znajduje się przystanek.

Parametry

ulica	Ulica przystanku.
-------	-------------------

Definicja w linii [209](#) pliku [PrzystanekDto.java](#).

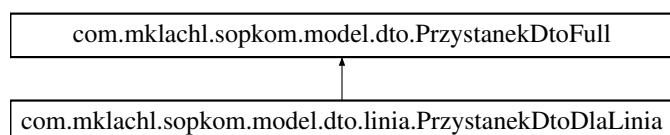
Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.79 Dokumentacja klasy

com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLinia

Diagram dziedziczenia dla com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLinia



Metody publiczne

- `PrzystanekDtoDlaLinia()`
- `PrzystanekDtoDlaLinia(PrzystanekWlini przystanekWlini)`
- `Short getKolejnosc()`
- `void setKolejnosc(Short kolejnosc)`
- `String getAdres()`
- `void setAdres(String adres)`
- `Long getPrzystanekwlinild()`
- `void setPrzystanekwlinild(Long przystanekwlinild)`

Dodatkowe Dziedziczone Składowe

6.79.1 Opis szczegółowy

Data Transfer Object (DTO) dla encji PrzystanekWlini, używany w kontekście linii.

Definicja w linii 9 pliku [PrzystanekDtoDlaLinia.java](#).

6.79.2 Dokumentacja konstruktora i destruktora

6.79.2.1 PrzystanekDtoDlaLinia() [1/2]

```
com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLinia.PrzystanekDtoDlaLinia( )
```

Konstruktor domyślny.

Definicja w linii 29 pliku [PrzystanekDtoDlaLinia.java](#).

6.79.2.2 PrzystanekDtoDlaLinia() [2/2]

```
com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLinia.PrzystanekDtoDlaLinia(   
    PrzystanekWlini przystanekWlini )
```

Konstruktor przyjmujący obiekt encji PrzystanekWlini.

Parametry

<code>przystanekWlini</code>	Obiekt encji PrzystanekWlini.
------------------------------	-------------------------------

Definicja w linii 36 pliku [PrzystanekDtoDlaLinia.java](#).

6.79.3 Dokumentacja funkcji składowych

6.79.3.1 getAdres()

```
String com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLinia.getAdres ( )
```

Zwraca adres przystanku.

Zwraca

 adres Adres przystanku.

Definicja w linii 65 pliku [PrzystanekDtoDlaLinia.java](#).

6.79.3.2 getKolejnosc()

```
Short com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLinia.getKolejnosc ( )
```

Zwraca kolejność przystanku w linii.

Zwraca

 kolejnosc Kolejność przystanku.

Definicja w linii 49 pliku [PrzystanekDtoDlaLinia.java](#).

6.79.3.3 getPrzystanekwlinId()

```
Long com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLinia.getPrzystanekwlinId ( )
```

Zwraca identyfikator przystanku w linii.

Zwraca

 przystanekwlinId Identyfikator przystanku w linii.

Definicja w linii 81 pliku [PrzystanekDtoDlaLinia.java](#).

6.79.3.4 setAdres()

```
void com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLinia.setAdres (   
    String adres )
```

Ustawia adres przystanku.

Parametry

<i>adres</i>	Adres przystanku.
--------------	-------------------

Definicja w linii 73 pliku [PrzystanekDtoDlaLinia.java](#).

6.79.3.5 setKolejnosc()

```
void com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLinia.setKolejnosc (
    Short kolejnosc )
```

Ustawia kolejność przystanku w linii.

Parametry

<i>kolejnosc</i>	Kolejność przystanku.
------------------	-----------------------

Definicja w linii 57 pliku [PrzystanekDtoDlaLinia.java](#).

6.79.3.6 setPrzystanekwlinild()

```
void com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLinia.setPrzystanekwliniId (
    Long przystanekwliniId )
```

Ustawia identyfikator przystanku w linii.

Parametry

<i>przystanekwlini_← Id</i>	Identyfikator przystanku w linii.
---	-----------------------------------

Definicja w linii 89 pliku [PrzystanekDtoDlaLinia.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.80 Dokumentacja klasy com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLinialInput

Metody publiczne

- [PrzystanekDtoDlaLinialInput \(\)](#)

- `PrzystanekDtoDlaLiniaInput` (`PrzystanekWlini` przystanek)
- `Long getPrzystanekId ()`
- `void setPrzystanekId (Long przystanekId)`
- `Short getKolejnosc ()`
- `void setKolejnosc (Short kolejnosc)`

6.80.1 Opis szczegółowy

Data Transfer Object (DTO) wejściowy dla encji `PrzystanekWlini`, używany w kontekście linii.

Definicja w linii 8 pliku [PrzystanekDtoDlaLiniaInput.java](#).

6.80.2 Dokumentacja konstruktora i destruktora

6.80.2.1 PrzystanekDtoDlaLiniaInput() [1/2]

```
com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLiniaInput.PrzystanekDtoDlaLiniaInput ( )
```

Konstruktor domyślny.

Definicja w linii 23 pliku [PrzystanekDtoDlaLiniaInput.java](#).

6.80.2.2 PrzystanekDtoDlaLiniaInput() [2/2]

```
com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLiniaInput.PrzystanekDtoDlaLiniaInput (  
    PrzystanekWlini przystanek )
```

Konstruktor przyjmujący obiekt encji `PrzystanekWlini`.

Parametry

<code>przystanek</code>	Obiekt encji <code>PrzystanekWlini</code> .
-------------------------	---

Definicja w linii 30 pliku [PrzystanekDtoDlaLiniaInput.java](#).

6.80.3 Dokumentacja funkcji składowych

6.80.3.1 getKolejnosc()

```
Short com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLiniaInput.getKolejnosc ( )
```

Zwraca kolejność przystanku w linii.

Zwrota

kolejnosc Kolejność przystanku.

Definicja w linii [57](#) pliku [PrzystanekDtoDlaLiniaInput.java](#).

6.80.3.2 getPrzystanekId()

```
Long com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLiniaInput.getPrzystanekId ( )
```

Zwraca identyfikator przystanku.

Zwrota

przystanekId Identyfikator przystanku.

Definicja w linii [41](#) pliku [PrzystanekDtoDlaLiniaInput.java](#).

6.80.3.3 setKolejnosc()

```
void com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLiniaInput.setKolejnosc (  
    Short kolejnosc )
```

Ustawia kolejność przystanku w linii.

Parametry

<i>kolejnosc</i>	Kolejność przystanku.
------------------	-----------------------

Definicja w linii [65](#) pliku [PrzystanekDtoDlaLiniaInput.java](#).

6.80.3.4 setPrzystanekId()

```
void com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLiniaInput.setPrzystanekId (   
    Long przystanekId )
```

Ustawia identyfikator przystanku.

Parametry

<i>przystanek</i>	Identyfikator przystanku.
<i>Id</i>	

Definicja w linii 49 pliku [PrzystanekDtoDlaLiniaInput.java](#).

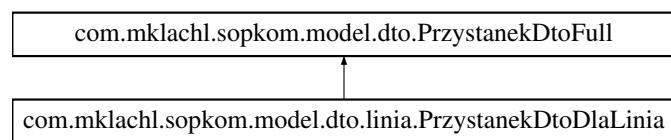
Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.81 Dokumentacja klasy

com.mklachl.sopkom.model.dto.PrzystanekDtoFull

Diagram dziedziczenia dla com.mklachl.sopkom.model.dto.PrzystanekDtoFull



Metody publiczne

- [PrzystanekDtoFull \(\)](#)
- [PrzystanekDtoFull \(Przystanek przystanek\)](#)
- Long [getPrzystanekId \(\)](#)
- [PrzystanekDtoFull getPrzystanekOdwozny \(\)](#)
- String [getNazwa \(\)](#)
- String [getKodPocztowy \(\)](#)
- String [getMiasto \(\)](#)
- String [getUlica \(\)](#)
- String [getDlugoscGeograficzna \(\)](#)
- String [getSzerokoscGeograficzna \(\)](#)
- void [setPrzystanekId \(Long przystanekId\)](#)
- void [setPrzystanekOdwozny \(PrzystanekDtoFull przystanekOdwozny\)](#)
- void [setNazwa \(String nazwa\)](#)
- void [setKodPocztowy \(String kodPocztowy\)](#)
- void [setMiasto \(String miasto\)](#)
- void [setUlica \(String ulica\)](#)
- void [setDlugoscGeograficzna \(String dlugoscGeograficzna\)](#)
- void [setSzerokoscGeograficzna \(String szerokoscGeograficzna\)](#)

Atrybuty chronione

- Long [przystanekId](#)
- [PrzystanekDtoFull przystanekOdwozny](#)
- String [nazwa](#)
- String [kodPocztowy](#)
- String [miasto](#)
- String [ulica](#)
- String [dlugoscGeograficzna](#)
- String [szerokoscGeograficzna](#)

6.81.1 Opis szczegółowy

Data Transfer Object (DTO) dla encji Przystanek z pełnymi informacjami.

Definicja w linii 8 pliku [PrzystanekDtoFull.java](#).

6.81.2 Dokumentacja konstruktora i destruktora

6.81.2.1 PrzystanekDtoFull() [1/2]

```
com.mklachl.sopkom.model.dto.PrzystanekDtoFull.PrzystanekDtoFull( )
```

Konstruktor domyślny.

Definicja w linii 53 pliku [PrzystanekDtoFull.java](#).

6.81.2.2 PrzystanekDtoFull() [2/2]

```
com.mklachl.sopkom.model.dto.PrzystanekDtoFull.PrzystanekDtoFull(   
    Przystanek przystanek )
```

Konstruktor przyjmujący obiekt encji Przystanek.

Parametry

<i>przystanek</i>	Obiekt encji Przystanek.
-------------------	--------------------------

Definicja w linii 60 pliku [PrzystanekDtoFull.java](#).

6.81.3 Dokumentacja funkcji składowych

6.81.3.1 getDlugoscGeograficzna()

```
String com.mklachl.sopkom.model.dto.PrzystanekDtoFull.getDlugoscGeograficzna( )
```

Zwraca długość geograficzną przystanku.

Zwrota

dlugoscGeograficzna Długość geograficzna przystanku.

Definicja w linii 129 pliku [PrzystanekDtoFull.java](#).

6.81.3.2 getKodPocztowy()

```
String com.mklachl.sopkom.model.dto.PrzystanekDtoFull.getKodPocztowy ( )
```

Zwraca kod pocztowy przystanku.

Zwraca

kodPocztowy Kod pocztowy przystanku.

Definicja w linii 105 pliku [PrzystanekDtoFull.java](#).

6.81.3.3 getMiasto()

```
String com.mklachl.sopkom.model.dto.PrzystanekDtoFull.getMiasto ( )
```

Zwraca miasto, w którym znajduje się przystanek.

Zwraca

miasto Miasto przystanku.

Definicja w linii 113 pliku [PrzystanekDtoFull.java](#).

6.81.3.4 getNazwa()

```
String com.mklachl.sopkom.model.dto.PrzystanekDtoFull.getNazwa ( )
```

Zwraca nazwę przystanku.

Zwraca

nazwa Nazwa przystanku.

Definicja w linii 97 pliku [PrzystanekDtoFull.java](#).

6.81.3.5 getPrzystanekId()

```
Long com.mklachl.sopkom.model.dto.PrzystanekDtoFull.getPrzystanekId ( )
```

Zwraca identyfikator przystanku.

Zwraca

przystanekId Identyfikator przystanku.

Definicja w linii 81 pliku [PrzystanekDtoFull.java](#).

6.81.3.6 getPrzystanekOdwrotny()

```
PrzystanekDtoFull com.mklachl.sopkom.model.dto.PrzystanekDtoFull.getPrzystanekOdwrotny ( )
```

Zwraca DTO odwrotnego przystanku.

Zwraca

przystanekOdwrotny DTO odwrotnego przystanku.

Definicja w linii 89 pliku [PrzystanekDtoFull.java](#).

6.81.3.7 getSzerokoscGeograficzna()

```
String com.mklachl.sopkom.model.dto.PrzystanekDtoFull.getSzerokoscGeograficzna ( )
```

Zwraca szerokość geograficzną przystanku.

Zwraca

szerokoscGeograficzna Szerokość geograficzna przystanku.

Definicja w linii 137 pliku [PrzystanekDtoFull.java](#).

6.81.3.8 getUlica()

```
String com.mklachl.sopkom.model.dto.PrzystanekDtoFull.getUlica ( )
```

Zwraca ulicę, na której znajduje się przystanek.

Zwraca

ulica Ulica przystanku.

Definicja w linii 121 pliku [PrzystanekDtoFull.java](#).

6.81.3.9 setDlugoscGeograficzna()

```
void com.mklachl.sopkom.model.dto.PrzystanekDtoFull.setDlugoscGeograficzna ( String dlugoscGeograficzna )
```

Ustawia długość geograficzną przystanku.

Parametry

<i>dlugoscGeograficzna</i>	Długość geograficzna przystanku.
----------------------------	----------------------------------

Definicja w linii [195](#) pliku [PrzystanekDtoFull.java](#).

6.81.3.10 setKodPocztowy()

```
void com.mklachl.sopkom.model.dto.PrzystanekDtoFull.setKodPocztowy (
    String kodPocztowy )
```

Ustawia kod pocztowy przystanku.

Parametry

<i>kodPocztowy</i>	Kod pocztowy przystanku.
--------------------	--------------------------

Definicja w linii [171](#) pliku [PrzystanekDtoFull.java](#).

6.81.3.11 setMiasto()

```
void com.mklachl.sopkom.model.dto.PrzystanekDtoFull.setMiasto (
    String miasto )
```

Ustawia miasto, w którym znajduje się przystanek.

Parametry

<i>miasto</i>	Miasto przystanku.
---------------	--------------------

Definicja w linii [179](#) pliku [PrzystanekDtoFull.java](#).

6.81.3.12 setNazwa()

```
void com.mklachl.sopkom.model.dto.PrzystanekDtoFull.setNazwa (
    String nazwa )
```

Ustawia nazwę przystanku.

Parametry

<i>nazwa</i>	Nazwa przystanku.
--------------	-------------------

Definicja w linii 163 pliku [PrzystanekDtoFull.java](#).

6.81.3.13 setPrzystanekId()

```
void com.mklachl.sopkom.model.dto.PrzystanekDtoFull.setPrzystanekId ( Long przystanekId )
```

Ustawia identyfikator przystanku.

Parametry

<i>przystanek</i> ← <i>Id</i>	Identyfikator przystanku.
----------------------------------	---------------------------

Definicja w linii 147 pliku [PrzystanekDtoFull.java](#).

6.81.3.14 setPrzystanekOdwrotny()

```
void com.mklachl.sopkom.model.dto.PrzystanekDtoFull.setPrzystanekOdwrotny ( PrzystanekDtoFull przystanekOdwrotny )
```

Ustawia DTO odwrotnego przystanku.

Parametry

<i>przystanekOdwrotny</i>	DTO odwrotnego przystanku.
---------------------------	----------------------------

Definicja w linii 155 pliku [PrzystanekDtoFull.java](#).

6.81.3.15 setSzerokoscGeograficzna()

```
void com.mklachl.sopkom.model.dto.PrzystanekDtoFull.setSzerokoscGeograficzna ( String szerokoscGeograficzna )
```

Ustawia szerokość geograficzną przystanku.

Parametry

<i>szerokoscGeograficzna</i>	Szerokość geograficzna przystanku.
------------------------------	------------------------------------

Definicja w linii 203 pliku [PrzystanekDtoFull.java](#).

6.81.3.16 setUlica()

```
void com.mklachl.sopkom.model.dto.PrzystanekDtoFull.setUlica (
    String ulica )
```

Ustawia ulicę, na której znajduje się przystanek.

Parametry

<i>ulica</i>	Ulica przystanku.
--------------	-------------------

Definicja w linii [187](#) pliku [PrzystanekDtoFull.java](#).

6.81.4 Dokumentacja atrybutów składowych

6.81.4.1 dlugoscGeograficzna

```
String com.mklachl.sopkom.model.dto.PrzystanekDtoFull.dlugoscGeograficzna [protected]
```

Długość geograficzna przystanku.

Definicja w linii [43](#) pliku [PrzystanekDtoFull.java](#).

6.81.4.2 kodPocztowy

```
String com.mklachl.sopkom.model.dto.PrzystanekDtoFull.kodPocztowy [protected]
```

Kod pocztowy przystanku.

Definicja w linii [28](#) pliku [PrzystanekDtoFull.java](#).

6.81.4.3 miasto

```
String com.mklachl.sopkom.model.dto.PrzystanekDtoFull.miasto [protected]
```

Miasto, w którym znajduje się przystanek.

Definicja w linii [33](#) pliku [PrzystanekDtoFull.java](#).

6.81.4.4 nazwa

```
String com.mklachl.sopkom.model.dto.PrzystanekDtoFull.nazwa [protected]
```

Nazwa przystanku.

Definicja w linii 23 pliku [PrzystanekDtoFull.java](#).

6.81.4.5 przystanekId

```
Long com.mklachl.sopkom.model.dto.PrzystanekDtoFull.przystanekId [protected]
```

Identyfikator przystanku.

Definicja w linii 13 pliku [PrzystanekDtoFull.java](#).

6.81.4.6 przystanekOdwrotny

```
PrzystanekDtoFull com.mklachl.sopkom.model.dto.PrzystanekDtoFull.przystanekOdwrotny [protected]
```

DTO odwrotnego przystanku.

Definicja w linii 18 pliku [PrzystanekDtoFull.java](#).

6.81.4.7 szerokoscGeograficzna

```
String com.mklachl.sopkom.model.dto.PrzystanekDtoFull.szerokoscGeograficzna [protected]
```

Szerokość geograficzna przystanku.

Definicja w linii 48 pliku [PrzystanekDtoFull.java](#).

6.81.4.8 ulica

```
String com.mklachl.sopkom.model.dto.PrzystanekDtoFull.ulica [protected]
```

Ulica, na której znajduje się przystanek.

Definicja w linii 38 pliku [PrzystanekDtoFull.java](#).

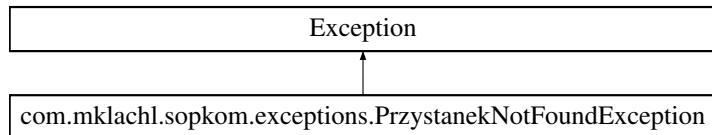
Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.82 Dokumentacja klasy

com.mklachl.sopkom.exceptions.PrzystanekNotFoundException

Diagram dziedziczenia dla com.mklachl.sopkom.exceptions.PrzystanekNotFoundException



Metody publiczne

- [PrzystanekNotFoundException](#) (Long liniald, Long przystanekid)

6.82.1 Opis szczegółowy

Definicja w linii 3 pliku [PrzystanekNotFoundException.java](#).

6.82.2 Dokumentacja konstruktora i destruktora

6.82.2.1 PrzystanekNotFoundException()

```
com.mklachl.sopkom.exceptions.PrzystanekNotFoundException.PrzystanekNotFoundException (
    Long liniald,
    Long przystanekid )
```

Definicja w linii 10 pliku [PrzystanekNotFoundException.java](#).

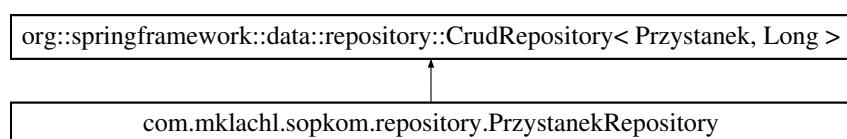
Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/exc

6.83 Dokumentacja interfejsu

com.mklachl.sopkom.repository.PrzystanekRepository

Diagram dziedziczenia dla com.mklachl.sopkom.repository.PrzystanekRepository



Metody publiczne

- List< Przystanek > findAllByPrzystanekOdwrótny (Przystanek przystanekOdwrótny)
- Przystanek findByNazwa (String nazwa)
- List< Przystanek > findAllByNazwaLike (String nazwa)
- List< Przystanek > findAllByKodPocztowy (String kodPocztowy)
- List< Przystanek > findAllByMiasto (String miasto)
- List< Przystanek > findAllByUlica (String ulica)
- List< Przystanek > findAllByUlicaLike (String ulica)
- List< Przystanek > findAllBySzerokoscGeograficzna (String szerokoscGeograficzna)
- List< Przystanek > findAllByDlugoscGeograficzna (String dlugoscGeograficzna)
- List< Przystanek > findAll ()

6.83.1 Opis szczegółowy

Interfejs repository dla encji Przystanek

Definicja w linii 12 pliku [PrzystanekRepository.java](#).

6.83.2 Dokumentacja funkcji składowych

6.83.2.1 findAll()

```
List< Przystanek > com.mklachl.sopkom.repository.PrzystanekRepository.findAll ( )
```

Znajduje wszystkie przystanki

Zwroca

Lista przystanków

6.83.2.2 findAllByDlugoscGeograficzna()

```
List< Przystanek > com.mklachl.sopkom.repository.PrzystanekRepository.findAllByDlugoscGeograficzna  
(  
    String dlugoscGeograficzna )
```

Znajduje wszystkie przystanki po długości geograficznej

Parametry

<i>dlugoscGeograficzna</i>	długość geograficzna
----------------------------	----------------------

Zwrota

Lista przystanków

6.83.2.3 findAllByKodPocztowy()

```
List< Przystanek > com.mklachl.sopkom.repository.PrzystanekRepository.findAllByKodPocztowy ( String kodPocztowy )
```

Znajduje wszystkie przystanki po kodzie pocztowym

Parametry

kodPocztowy	kod pocztowy
-------------	--------------

Zwrota

Lista przystanków

6.83.2.4 findAllByMiasto()

```
List< Przystanek > com.mklachl.sopkom.repository.PrzystanekRepository.findAllByMiasto ( String miasto )
```

Znajduje wszystkie przystanki po mieście

Parametry

miasto	miasto
--------	--------

Zwrota

Lista przystanków

6.83.2.5 findAllByNazwaLike()

```
List< Przystanek > com.mklachl.sopkom.repository.PrzystanekRepository.findAllByNazwaLike ( String nazwa )
```

Znajduje wszystkie przystanki, których nazwa pasuje do podanej frazy

Parametry

<i>nazwa</i>	nazwa
--------------	-------

Zwrota

Lista przystanków

6.83.2.6 findAllByPrzystanekOdwrotny()

```
List< Przystanek > com.mklachl.sopkom.repository.PrzystanekRepository.findAllByPrzystanekOdwrotny ( Przystanek przystanekOdwrotny )
```

Znajduje wszystkie przystanki dla danego przystanku odwrotnego

Parametry

<i>przystanekOdwrotny</i>	przystanek odwrotny
---------------------------	---------------------

Zwrota

Lista przystanków

6.83.2.7 findAllBySzerokoscGeograficzna()

```
List< Przystanek > com.mklachl.sopkom.repository.PrzystanekRepository.findAllBySzerokoscGeograficzna ( String szerokoscGeograficzna )
```

Znajduje wszystkie przystanki po szerokości geograficznej

Parametry

<i>szerokoscGeograficzna</i>	szerokość geograficzna
------------------------------	------------------------

Zwrota

Lista przystanków

6.83.2.8 findAllByUlica()

```
List< Przystanek > com.mklachl.sopkom.repository.PrzystanekRepository.findAllByUlica ( String ulica )
```

Znajduje wszystkie przystanki po ulicy

Parametry

<i>ulica</i>	ulica
--------------	-------

Zwrota

Lista przystanków

6.83.2.9 findAllByUlicaLike()

```
List< Przystanek > com.mklachl.sopkom.repository.PrzystanekRepository.findAllByUlicaLike ( String ulica )
```

Znajduje wszystkie przystanki, których ulica pasuje do podanej frazy

Parametry

<i>ulica</i>	ulica
--------------	-------

Zwrota

Lista przystanków

6.83.2.10 findByNazwa()

```
Przystanek com.mklachl.sopkom.repository.PrzystanekRepository.findByNazwa ( String nazwa )
```

Znajduje przystanek po nazwie

Parametry

<i>nazwa</i>	nazwa
--------------	-------

Zwrota

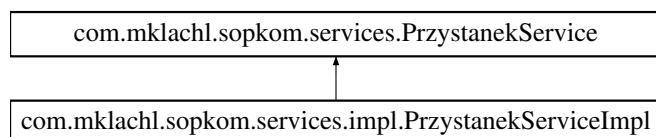
Przystanek

Dokumentacja dla tego interfejsu została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/repo...

6.84 Dokumentacja interfejsu com.mklachl.sopkom.services.PrzystanekService

Diagram dziedziczenia dla com.mklachl.sopkom.services.PrzystanekService



Metody publiczne

- Przystanek savePrzystanek (Przystanek przystanek)
- Przystanek savePrzystanek (PrzystanekDto przystanekDto)
- Przystanek findPrzystanekByNazwa (String nazwa)
- List< Przystanek > findPrzystanekByKodPocztowy (String kodPocztowy)
- List< Przystanek > findPrzystanekByMiasto (String miasto)
- List< Przystanek > findPrzystanekByUlica (String ulica)
- List< Przystanek > findPrzystanekBySzerokoscGeograficzna (String szerokoscGeograficzna)
- List< Przystanek > findPrzystanekByDlugoscGeograficzna (String dlugoscGeograficzna)
- Optional< Przystanek > findPrzystanekById (Long id)
- List< Przystanek > findPrzystanekByPrzystanekOdwrotny (Przystanek przystanekOdwrotny)

6.84.1 Opis szczegółowy

Interfejs serwisu dla obsługi przystanków

Definicja w linii 12 pliku [PrzystanekService.java](#).

6.84.2 Dokumentacja funkcji składowych

6.84.2.1 findPrzystanekByDlugoscGeograficzna()

```
List< Przystanek > com.mklachl.sopkom.services.PrzystanekService.findPrzystanekByDlugoscGeograficzna (String dlugoscGeograficzna )
```

Znajduje przystanki po długości geograficznej

Parametry

<i>długość geograficzna</i>	długość geograficzna
-----------------------------	----------------------

Zwroca

lista przystanków

Implementowany w [com.mklachl.sopkom.services.impl.PrzystanekServiceImpl](#).

6.84.2.2 findPrzystanekById()

```
Optional< Przystanek > com.mklachl.sopkom.services.PrzystanekService.findPrzystanekById ( Long id )
```

Znajduje przystanek po ID

Parametry

<i>id</i>	ID przystanku
-----------	---------------

Zwroca

Optional z przystankiem

Implementowany w [com.mklachl.sopkom.services.impl.PrzystanekServiceImpl](#).

6.84.2.3 findPrzystanekByKodPocztowy()

```
List< Przystanek > com.mklachl.sopkom.services.PrzystanekService.findPrzystanekByKodPocztowy ( String kodPocztowy )
```

Znajduje przystanki po kodzie pocztowym

Parametry

<i>kod pocztowy</i>	kod pocztowy
---------------------	--------------

Zwroca

lista przystanków

Implementowany w [com.mklachl.sopkom.services.impl.PrzystanekServiceImpl](#).

6.84.2.4 findPrzystanekByMiasto()

```
List< Przystanek > com.mklachl.sopkom.services.PrzystanekService.findPrzystanekByMiasto ( String miasto )
```

Znajduje przystanki po mieście

Parametry

<i>miasto</i>	miasto
---------------	--------

Zwroca

lista przystanków

Implementowany w [com.mklachl.sopkom.services.impl.PrzystanekServiceImpl](#).

6.84.2.5 findPrzystanekByNazwa()

```
Przystanek com.mklachl.sopkom.services.PrzystanekService.findPrzystanekByNazwa ( String nazwa )
```

Znajduje przystanek po nazwie

Parametry

<i>nazwa</i>	nazwa przystanku
--------------	------------------

Zwroca

znaleziony przystanek

Implementowany w [com.mklachl.sopkom.services.impl.PrzystanekServiceImpl](#).

6.84.2.6 findPrzystanekByPrzystanekOdwrotny()

```
List< Przystanek > com.mklachl.sopkom.services.PrzystanekService.findPrzystanekByPrzystanekOdwrotny ( Przystanek przystanekOdwrotny )
```

Znajduje przystanki po przystanku odwrotnym

Parametry

<i>przystanekOdwrotny</i>	przystanek odwrotny
---------------------------	---------------------

Zwrota

lista przystanków

Implementowany w [com.mklachl.sopkom.services.impl.PrzystanekServiceImpl](#).

6.84.2.7 findPrzystanekBySzerokoscGeograficzna()

```
List< Przystanek > com.mklachl.sopkom.services.PrzystanekService.findPrzystanekBySzerokoscGeograficzna ( String szerokoscGeograficzna )
```

Znajduje przystanki po szerokości geograficznej

Parametry

szerokoscGeograficzna	szerokość geograficzna
-----------------------	------------------------

Zwrota

lista przystanków

Implementowany w [com.mklachl.sopkom.services.impl.PrzystanekServiceImpl](#).

6.84.2.8 findPrzystanekByUlica()

```
List< Przystanek > com.mklachl.sopkom.services.PrzystanekService.findPrzystanekByUlica ( String ulica )
```

Znajduje przystanki po ulicy

Parametry

ulica	ulica
-------	-------

Zwrota

lista przystanków

Implementowany w [com.mklachl.sopkom.services.impl.PrzystanekServiceImpl](#).

6.84.2.9 savePrzystanek() [1/2]

```
Przystanek com.mklachl.sopkom.services.PrzystanekService.savePrzystanek (
    Przystanek przystanek )
```

Zapisuje nowy przystanek

Parametry

przystanek	encja przystanku
------------	------------------

Zwrota

zapisany przystanek

Implementowany w [com.mklachl.sopkom.services.impl.PrzystanekServiceImpl](#).

6.84.2.10 savePrzystanek() [2/2]

```
Przystanek com.mklachl.sopkom.services.PrzystanekService.savePrzystanek (
    PrzystanekDto przystanekDto )
```

Zapisuje nowy przystanek na podstawie DTO

Parametry

przystanekDto	DTO przystanku
---------------	----------------

Zwrota

zapisany przystanek

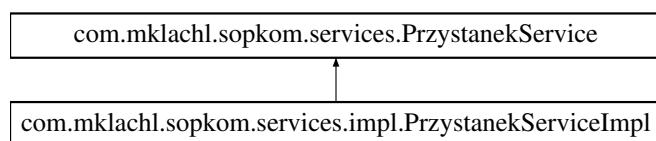
Implementowany w [com.mklachl.sopkom.services.impl.PrzystanekServiceImpl](#).

Dokumentacja dla tego interfejsu została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/ser...

6.85 Dokumentacja klasy com.mklachl.sopkom.services.impl.PrzystanekServiceImpl

Diagram dziedziczenia dla com.mklachl.sopkom.services.impl.PrzystanekServiceImpl



Metody publiczne

- PrzystanekServiceImpl ([PrzystanekRepository](#) przystanekRepository)
- Przystanek savePrzystanek ([Przystanek](#) przystanekDto)
- Przystanek savePrzystanek ([PrzystanekDto](#) przystanekDto)
- Przystanek findPrzystanekByNazwa (String nazwa)
- List<[Przystanek](#)> findPrzystanekByKodPocztowy (String kodPocztowy)
- List<[Przystanek](#)> findPrzystanekByMiasto (String miasto)
- List<[Przystanek](#)> findPrzystanekByUlica (String ulica)
- List<[Przystanek](#)> findPrzystanekBySzerokoscGeograficzna (String szerokoscGeograficzna)
- List<[Przystanek](#)> findPrzystanekByDlugoscGeograficzna (String dlugoscGeograficzna)
- Optional<[Przystanek](#)> findPrzystanekById (Long id)
- List<[Przystanek](#)> findPrzystanekByPrzystanekOdwrotny ([Przystanek](#) przystanekOdwrotny)

6.85.1 Opis szczegółowy

Definicja w linii 13 pliku [PrzystanekServiceImpl.java](#).

6.85.2 Dokumentacja konstruktora i destruktora

6.85.2.1 PrzystanekServiceImpl()

```
com.mklachl.sopkom.services.impl.PrzystanekServiceImpl.PrzystanekServiceImpl (
    PrzystanekRepository przystanekRepository )
```

Definicja w linii 16 pliku [PrzystanekServiceImpl.java](#).

6.85.3 Dokumentacja funkcji składowych

6.85.3.1 findPrzystanekByDlugoscGeograficzna()

```
List<Przystanek> com.mklachl.sopkom.services.impl.PrzystanekServiceImpl.findPrzystanekBy-
DlugoscGeograficzna (
    String dlugoscGeograficzna )
```

Znajduje przystanki po długości geograficznej

Parametry

<code>dlugoscGeograficzna</code>	długość geograficzna
----------------------------------	----------------------

Zwrota

lista przystanków

Implementuje [com.mklachl.sopkom.services.PrzystanekService](#).

Definicja w linii 89 pliku [PrzystanekServiceImpl.java](#).

6.85.3.2 findPrzystanekById()

```
Optional< Przystanek > com.mklachl.sopkom.services.impl.PrzystanekServiceImpl.findPrzystanekById ( Long id )
```

Znajduje przystanek po ID

Parametry

<i>id</i>	ID przystanku
-----------	---------------

Zwrota

Optional z przystankiem

Implementuje [com.mklachl.sopkom.services.PrzystanekService](#).

Definicja w linii 94 pliku [PrzystanekServiceImpl.java](#).

6.85.3.3 findPrzystanekByKodPocztowy()

```
List< Przystanek > com.mklachl.sopkom.services.impl.PrzystanekServiceImpl.findPrzystanekByKodPocztowy ( String kodPocztowy )
```

Znajduje przystanki po kodzie pocztowym

Parametry

<i>kodPocztowy</i>	kod pocztowy
--------------------	--------------

Zwrota

lista przystanków

Implementuje [com.mklachl.sopkom.services.PrzystanekService](#).

Definicja w linii 69 pliku [PrzystanekServiceImpl.java](#).

6.85.3.4 findPrzystanekByMiasto()

```
List< Przystanek > com.mklachl.sopkom.services.impl.PrzystanekServiceImpl.findPrzystanekBy->
Miasto (
        String miasto )
```

Znajduje przystanki po mieście

Parametry

<i>miasto</i>	miasto
---------------	--------

Zwroca

lista przystanków

Implementuje [com.mklachl.sopkom.services.PrzystanekService](#).

Definicja w linii 74 pliku [PrzystanekServiceImpl.java](#).

6.85.3.5 findPrzystanekByNazwa()

```
Przystanek com.mklachl.sopkom.services.impl.PrzystanekServiceImpl.findPrzystanekByNazwa (
        String nazwa )
```

Znajduje przystanek po nazwie

Parametry

<i>nazwa</i>	nazwa przystanku
--------------	------------------

Zwroca

znaleziony przystanek

Implementuje [com.mklachl.sopkom.services.PrzystanekService](#).

Definicja w linii 64 pliku [PrzystanekServiceImpl.java](#).

6.85.3.6 findPrzystanekByPrzystanekOdwrotny()

```
List< Przystanek > com.mklachl.sopkom.services.impl.PrzystanekServiceImpl.findPrzystanekBy->
PrzystanekOdwrotny (
        Przystanek przystanekOdwrotny )
```

Znajduje przystanki po przystanku odwrotnym

Parametry

<i>przystanekOdwrotny</i>	przystanek odwrotny
---------------------------	---------------------

Zwrota

lista przystanków

Implementuje [com.mklachl.sopkom.services.PrzystanekService](#).Definicja w linii 99 pliku [PrzystanekServiceImpl.java](#).**6.85.3.7 findPrzystanekBySzerokoscGeograficzna()**

```
List< Przystanek > com.mklachl.sopkom.services.impl.PrzystanekServiceImpl.findPrzystanekBy←
SzerokoscGeograficzna (
    String szerokoscGeograficzna )
```

Znajduje przystanki po szerokości geograficznej

Parametry

<i>szerokoscGeograficzna</i>	szerokość geograficzna
------------------------------	------------------------

Zwrota

lista przystanków

Implementuje [com.mklachl.sopkom.services.PrzystanekService](#).Definicja w linii 84 pliku [PrzystanekServiceImpl.java](#).**6.85.3.8 findPrzystanekByUlica()**

```
List< Przystanek > com.mklachl.sopkom.services.impl.PrzystanekServiceImpl.findPrzystanekBy←
Ulica (
    String ulica )
```

Znajduje przystanki po ulicy

Parametry

<i>ulica</i>	ulica
--------------	-------

Zwrota

lista przystanków

Implementuje [com.mklachl.sopkom.services.PrzystanekService](#).

Definicja w linii [79](#) pliku [PrzystanekServiceImpl.java](#).

6.85.3.9 savePrzystanek() [1/2]

```
Przystanek com.mklachl.sopkom.services.impl.PrzystanekServiceImpl.savePrzystanek (
    Przystanek przystanek )
```

Zapisuje nowy przystanek

Parametry

<i>przystanek</i>	encja przystanku
-------------------	------------------

Zwrota

zapisany przystanek

Implementuje [com.mklachl.sopkom.services.PrzystanekService](#).

Definicja w linii [19](#) pliku [PrzystanekServiceImpl.java](#).

6.85.3.10 savePrzystanek() [2/2]

```
Przystanek com.mklachl.sopkom.services.impl.PrzystanekServiceImpl.savePrzystanek (
    PrzystanekDto przystanekDto )
```

Zapisuje nowy przystanek na podstawie DTO

Parametry

<i>przystanekDto</i>	DTO przystanku
----------------------	----------------

Zwrota

zapisany przystanek

Implementuje [com.mklachl.sopkom.services.PrzystanekService](#).

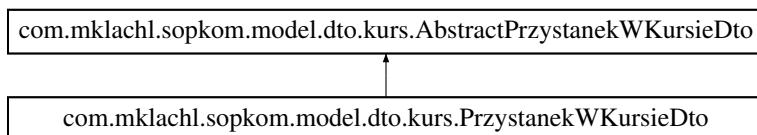
Definicja w linii [33](#) pliku [PrzystanekServiceImpl.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/se

6.86 Dokumentacja klasy com.mklachl.sopkom.model.dto.kurs.PrzystanekWKursieDto

Diagram dziedziczenia dla com.mklachl.sopkom.model.dto.kurs.PrzystanekWKursieDto



Metody publiczne

- [PrzystanekWKursieDto \(\)](#)
- [PrzystanekWKursieDto \(KursPrzystanekWlini kpw\)](#)
- [PrzystanekDtoDlaLinia getPrzystanekWLini \(\)](#)
- void [setPrzystanekWLini \(PrzystanekDtoDlaLinia przystanekWLini\)](#)

Dodatkowe Dziedziczone Składowe

6.86.1 Opis szczegółowy

Data Transfer Object (DTO) dla encji KursPrzystanekWlini.

Definicja w linii 9 pliku [PrzystanekWKursieDto.java](#).

6.86.2 Dokumentacja konstruktora i destruktora

6.86.2.1 PrzystanekWKursieDto() [1/2]

```
com.mklachl.sopkom.model.dto.kurs.PrzystanekWKursieDto.PrzystanekWKursieDto ( )
```

Konstruktor domyślny.

Definicja w linii 19 pliku [PrzystanekWKursieDto.java](#).

6.86.2.2 PrzystanekWKursieDto() [2/2]

```
com.mklachl.sopkom.model.dto.kurs.PrzystanekWKursieDto.PrzystanekWKursieDto (  
    KursPrzystanekWlini kpw )
```

Konstruktor przyjmujący obiekt encji KursPrzystanekWlini.

Parametry

<i>kpw</i>	Obiekt encji KursPrzystanekWLini.
------------	-----------------------------------

Definicja w linii 26 pliku [PrzystanekWKursieDto.java](#).

6.86.3 Dokumentacja funkcji składowych

6.86.3.1 getPrzystanekWLini()

```
PrzystanekDtoDlaLinia com.mklachl.sopkom.model.dto.kurs.PrzystanekWKursieDto.getPrzystanek←
WLini ( )
```

Zwraca DTO przystanku w linii.

Zwraca

przystanekWLini DTO przystanku w linii.

Definicja w linii 37 pliku [PrzystanekWKursieDto.java](#).

6.86.3.2 setPrzystanekWLini()

```
void com.mklachl.sopkom.model.dto.kurs.PrzystanekWKursieDto.setPrzystanekWLini (
    PrzystanekDtoDlaLinia przystanekWLini )
```

Ustawia DTO przystanku w linii.

Parametry

<i>przystanekWLini</i>	DTO przystanku w linii.
------------------------	-------------------------

Definicja w linii 45 pliku [PrzystanekWKursieDto.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.87 Dokumentacja klasy

com.mklachl.sopkom.model.entity.PrzystanekWLini

Metody publiczne

- Long [getPrzystanekWLiniId \(\)](#)

- `Przystanek getPrzystanek ()`
- `Linia getLinia ()`
- `Short getKolejnosc ()`
- `List< KursPrzystanekWlini > getKursPrzystanekWlinii ()`
- `void setPrzystanekWlinild (Long przystanekWlinild)`
- `void setPrzystanek (Przystanek przystanek)`
- `void setLinia (Linia linia)`
- `void setKolejnosc (Short kolejnosc)`
- `void setKursPrzystanekWlinii (List< KursPrzystanekWlini > kursPrzystanekWlinii)`

6.87.1 Opis szczegółowy

Klasa reprezentująca encję `PrzystanekWlini`.

Definicja w linii 14 pliku `PrzystanekWlini.java`.

6.87.2 Dokumentacja funkcji składowych

6.87.2.1 `getKolejnosc()`

`Short com.mklachl.sopkom.model.entity.PrzystanekWlini.getKolejnosc ()`

Zwrota kolejność przystanku w linii.

Zwrota

`kolejnosc` Kolejność przystanku w linii.

Definicja w linii 81 pliku `PrzystanekWlini.java`.

6.87.2.2 `getKursPrzystanekWlinii()`

`List< KursPrzystanekWlini > com.mklachl.sopkom.model.entity.PrzystanekWlini.getKursPrzystanekWlinii ()`

Zwrota listę `KursPrzystanekWlini` związanych z `PrzystanekWlini`.

Zwrota

`kursPrzystanekWlinii` Lista `KursPrzystanekWlini`.

Definicja w linii 89 pliku `PrzystanekWlini.java`.

6.87.2.3 getLinia()

```
Linia com.mklachl.sopkom.model.entity.PrzystanekWlini.getLinia ( )
```

Zwraca linię związaną z [PrzystanekWlini](#).

Zwrota

linia [Linia](#) związana z [PrzystanekWlini](#).

Definicja w linii [73](#) pliku [PrzystanekWlini.java](#).

6.87.2.4 getPrzystanek()

```
Przystanek com.mklachl.sopkom.model.entity.PrzystanekWlini.getPrzystanek ( )
```

Zwraca przystanek związały z [PrzystanekWlini](#).

Zwrota

przystanek [Przystanek](#) związały z [PrzystanekWlini](#).

Definicja w linii [65](#) pliku [PrzystanekWlini.java](#).

6.87.2.5 getPrzystanekWliniId()

```
Long com.mklachl.sopkom.model.entity.PrzystanekWlini.getPrzystanekWliniId ( )
```

Zwraca identyfikator [PrzystanekWlini](#).

Zwrota

przystanekWliniId Identyfikator [PrzystanekWlini](#).

Definicja w linii [57](#) pliku [PrzystanekWlini.java](#).

6.87.2.6 setKolejnosc()

```
void com.mklachl.sopkom.model.entity.PrzystanekWlini.setKolejnosc ( Short kolejnosc )
```

Ustawia kolejność przystanku w linii.

Parametry

<i>kolejnosc</i>	Kolejność przystanku w linii.
------------------	-------------------------------

Definicja w linii [123](#) pliku [PrzystanekWlini.java](#).

6.87.2.7 setKursPrzystanekWlinii()

```
void com.mklachl.sopkom.model.entity.PrzystanekWlini.setKursPrzystanekWlinii (List< KursPrzystanekWlini > kursPrzystanekWlinii )
```

Ustawia listę [KursPrzystanekWlini](#) związańnych z [PrzystanekWlini](#).

Parametry

<i>kursPrzystanekWlinii</i>	Lista KursPrzystanekWlini .
-----------------------------	---

Definicja w linii [131](#) pliku [PrzystanekWlini.java](#).

6.87.2.8 setLinia()

```
void com.mklachl.sopkom.model.entity.PrzystanekWlini.setLinia (Linia linia )
```

Ustawia linię związaną z [PrzystanekWlini](#).

Parametry

<i>linia</i>	Linia związana z PrzystanekWlini .
--------------	--

Definicja w linii [115](#) pliku [PrzystanekWlini.java](#).

6.87.2.9 setPrzystanek()

```
void com.mklachl.sopkom.model.entity.PrzystanekWlini.setPrzystanek (Przystanek przystanek )
```

Ustawia przystanek związańny z [PrzystanekWlini](#).

Parametry

<i>przystanek</i>	Przystanek związany z PrzystanekWlini .
-------------------	---

Definicja w linii 107 pliku [PrzystanekWlini.java](#).

6.87.2.10 setPrzystanekWliniId()

```
void com.mklachl.sopkom.model.entity.PrzystanekWlini.setPrzystanekWliniId (
    Long przystanekWliniId )
```

Ustawia identyfikator [PrzystanekWlini](#).

Parametry

<code>przystanek← WliniId</code>	Identyfikator PrzystanekWlini .
--------------------------------------	---

Definicja w linii 99 pliku [PrzystanekWlini.java](#).

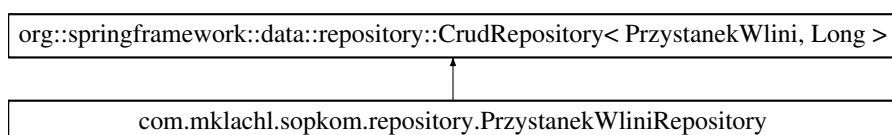
Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.88 Dokumentacja interfejsu

com.mklachl.sopkom.repository.PrzystanekWliniRepository

Diagram dziedziczenia dla com.mklachl.sopkom.repository.PrzystanekWliniRepository



Metody publiczne

- List< [PrzystanekWlini](#) > `findAllByPrzystanek (Przystanek przystanek)`
- List< [PrzystanekWlini](#) > `findAllByLinia (Linia linia)`
- List< [PrzystanekWlini](#) > `findAll` ()

6.88.1 Opis szczegółowy

Interfejs repository dla encji PrzystanekWlini

Definicja w linii 13 pliku [PrzystanekWliniRepository.java](#).

6.88.2 Dokumentacja funkcji składowych

6.88.2.1 findAll()

```
List< PrzystanekWlini > com.mklachl.sopkom.repository.PrzystanekWliniRepository.findAll ( )
```

Znajduje wszystkie przystanki w linii

Zwroca

Lista przystanków w linii

6.88.2.2 findAllByLinia()

```
List< PrzystanekWlini > com.mklachl.sopkom.repository.PrzystanekWliniRepository.findAllByLinia  
(  
    Linia linia )
```

Znajduje wszystkie przystanki w linii dla danej linii

Parametry

<i>linia</i>	<i>linia</i>
--------------	--------------

Zwroca

Lista przystanków w linii

6.88.2.3 findAllByPrzystanek()

```
List< PrzystanekWlini > com.mklachl.sopkom.repository.PrzystanekWliniRepository.findAllBy←  
Przystanek (←  
    Przystanek przystanek )
```

Znajduje wszystkie przystanki w linii dla danego przystanku

Parametry

<i>przystanek</i>	<i>przystanek</i>
-------------------	-------------------

Zwraca

Lista przystanków w linii

Dokumentacja dla tego interfejsu została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/rep

6.89 Dokumentacja klasy com.mklachl.sopkom.model.dto.kurs.[OUTDATED](#).PrzystenekWKursieDtoOUTDATED

Metody publiczne

- [PrzystenekWKursieDtoOUTDATED \(\)](#)
- [PrzystenekWKursieDtoOUTDATED \(KursPrzystanekWlini kwl\)](#)
- Long [getPrzystanekkurs_id \(\)](#)
- void [setPrzystanekkurs_id \(Long przystanekkurs_id\)](#)
- [PrzstanekDtoFull getPrzystanek \(\)](#)
- void [setPrzystanek \(PrzstanekDtoFull przystanek\)](#)
- Time [getGodzinna \(\)](#)
- void [setGodzinna \(Time t\)](#)

Atrybuty chronione

- Long [przystanekkurs_id](#)
- [PrzstanekDtoFull przystanek](#)
- short [kolejnosc](#)
- Time [godzinna](#)

6.89.1 Opis szczegółowy

Definicja w linii 9 pliku [PrzystenekWKursieDtoOUTDATED.java](#).

6.89.2 Dokumentacja konstruktora i destruktora

6.89.2.1 PrzystenekWKursieDtoOUTDATED() [1/2]

```
com.mklachl.sopkom.model.dto.kurs.OUTDATED.PrzystenekWKursieDtoOUTDATED.PrzystenekWKursieDtoOUTDATED ( )
```

Definicja w linii 19 pliku [PrzystenekWKursieDtoOUTDATED.java](#).

6.89.2.2 PrzystenekWKursieDtoOUTDATED() [2/2]

```
com.mklachl.sopkom.model.dto.kurs.UPDATED.PrzystenekWKursieDtoOUTDATED.PrzystenekWKursieDto←  
OUTDATED (   
    KursPrzystanekWlini kwl )
```

Definicja w linii [23](#) pliku [PrzystenekWKursieDtoOUTDATED.java](#).

6.89.3 Dokumentacja funkcji składowych**6.89.3.1 getGodzinna()**

```
Time com.mklachl.sopkom.model.dto.kurs.UPDATED.PrzystenekWKursieDtoOUTDATED.getGodzinna ( )
```

Definicja w linii [47](#) pliku [PrzystenekWKursieDtoOUTDATED.java](#).

6.89.3.2 getPrzystanek()

```
PrzystanekDtoFull com.mklachl.sopkom.model.dto.kurs.UPDATED.PrzystenekWKursieDtoOUTDATED.←  
getPrzystanek ( )
```

Definicja w linii [39](#) pliku [PrzystenekWKursieDtoOUTDATED.java](#).

6.89.3.3 getPrzystanekkurs_id()

```
Long com.mklachl.sopkom.model.dto.kurs.UPDATED.PrzystenekWKursieDtoOUTDATED.getPrzystanekkurs←  
_id ( )
```

Definicja w linii [31](#) pliku [PrzystenekWKursieDtoOUTDATED.java](#).

6.89.3.4 setGodzinna()

```
void com.mklachl.sopkom.model.dto.kurs.UPDATED.PrzystenekWKursieDtoOUTDATED.setGodzinna (   
    Time t )
```

Definicja w linii [51](#) pliku [PrzystenekWKursieDtoOUTDATED.java](#).

6.89.3.5 setPrzystanek()

```
void com.mklachl.sopkom.model.dto.kurs.UPDATED.PrzystenekWKursieDtoOUTDATED.setPrzystanek (←  
    PrzystanekDtoFull przystanek )
```

Definicja w linii 43 pliku [PrzystenekWKursieDtoOUTDATED.java](#).

6.89.3.6 setPrzystanekkurs_id()

```
void com.mklachl.sopkom.model.dto.kurs.UPDATED.PrzystenekWKursieDtoOUTDATED.setPrzystanekkurs←  
_id (←  
    Long przystanekkurs_id )
```

Definicja w linii 35 pliku [PrzystenekWKursieDtoOUTDATED.java](#).

6.89.4 Dokumentacja atrybutów składowych

6.89.4.1 godzinna

```
Time com.mklachl.sopkom.model.dto.kurs.UPDATED.PrzystenekWKursieDtoOUTDATED.godzinna [protected]
```

Definicja w linii 17 pliku [PrzystenekWKursieDtoOUTDATED.java](#).

6.89.4.2 kolejnosc

```
short com.mklachl.sopkom.model.dto.kurs.UPDATED.PrzystenekWKursieDtoOUTDATED.kolejnosc [protected]
```

Definicja w linii 15 pliku [PrzystenekWKursieDtoOUTDATED.java](#).

6.89.4.3 przystanek

```
PrzystanekDtoFull com.mklachl.sopkom.model.dto.kurs.UPDATED.PrzystenekWKursieDtoOUTDATED.←  
przystanek [protected]
```

Definicja w linii 13 pliku [PrzystenekWKursieDtoOUTDATED.java](#).

6.89.4.4 przystanekkurs_id

```
Long com.mklachl.sopkom.model.dto.kurs.OLDOUTDATED.PrzystenekWKursieDtoOUTDATED.przystanekkurs_id  
[protected]
```

Definicja w linii 11 pliku [PrzystenekWKursieDtoOUTDATED.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo
OLDOUTDATED/PrzystenekWKursieDtoOUTDATED.java

6.90 Dokumentacja klasy com.mklachl.sopkom.controller.RaportyController

Metody publiczne

- ResponseEntity< FileSystemResource > [generateRozklad](#) (@RequestParam(name="id") Long id) throws
FileNotFoundException, IOException

6.90.1 Opis szczegółowy

Definicja w linii 21 pliku [RaportyController.java](#).

6.90.2 Dokumentacja funkcji składowych

6.90.2.1 generateRozklad()

```
ResponseEntity< FileSystemResource > com.mklachl.sopkom.controller.RaportyController.generateRozklad  
(@RequestParam(name="id") Long id) throws FileNotFoundException, IOException
```

Endpoint do generowania raportu w formacie PDF.

Parametry

<i>id</i>	ID raportu
-----------	------------

Zwroca

ResponseType z plikiem PDF

Wyjątki

<code>FileNotFoundException</code>	
<code>IOException</code>	

Definicja w linii 35 pliku [RaportyController.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/

6.91 Dokumentacja wyliczenia com.mklachl.sopkom.model.entity.Role

Atrybuty publiczne

- `ADMIN`
- `MANAGER`
- `DISPATCHER`
- `DRIVER`

6.91.1 Opis szczegółowy

Enum reprezentujący role użytkowników.

Definicja w linii 6 pliku [Role.java](#).

6.91.2 Dokumentacja atrybutów składowych

6.91.2.1 ADMIN

`com.mklachl.sopkom.model.entity.Role.ADMIN`

Definicja w linii 7 pliku [Role.java](#).

6.91.2.2 DISPATCHER

`com.mklachl.sopkom.model.entity.Role.DISPATCHER`

Definicja w linii 9 pliku [Role.java](#).

6.91.2.3 DRIVER

com.mklachl.sopkom.model.entity.Role.DRIVER

Definicja w linii 11 pliku [Role.java](#).

6.91.2.4 MANAGER

com.mklachl.sopkom.model.entity.Role.MANAGER

Definicja w linii 8 pliku [Role.java](#).

Dokumentacja dla tego wyliczenia została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.92 Dokumentacja klasy com.mklachl.sopkom.controller.RozkladController

Metody publiczne

- ResponseEntity<?> [test](#) (@RequestParam(name="id") Long id, @RequestParam(name="date") @Date↔
TimeFormat(pattern="yyyy-MM-dd") Date date)

Atrybuty publiczne

- [PrzystanekRepository](#) repo
- [DateHelper](#) rozkladHelper

6.92.1 Opis szczegółowy

Definicja w linii 22 pliku [RozkladController.java](#).

6.92.2 Dokumentacja funkcji składowych

6.92.2.1 test()

```
ResponseEntity<?> com.mklachl.sopkom.controller.RozkladController.test (   
    @RequestParam(name="id") Long id,  
    @RequestParam(name="date") @DateTimeFormat(pattern="yyyy-MM-dd") Date date )
```

Endpoint do pobrania rozkładu dla przystanku w określonym dniu.

Parametry

<i>id</i>	ID przystanku
<i>date</i>	Data, dla której ma być wygenerowany rozkład

Zwraca

ResponseEntity z RozkladDto

Definicja w linii [37](#) pliku [RozkladController.java](#).

6.92.3 Dokumentacja atrybutów składowych

6.92.3.1 repo

`PrzystanekRepository com.mklachl.sopkom.controller.RozkladController.repo`

Definicja w linii [25](#) pliku [RozkladController.java](#).

6.92.3.2 rozkladHelper

`DateHelper com.mklachl.sopkom.controller.RozkladController.rozkladHelper`

Definicja w linii [28](#) pliku [RozkladController.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/

6.93 Dokumentacja klasy

com.mklachl.sopkom.model.dto.rozklad.RozkladDto

Komponenty

- class \$

Metody publiczne

- `RozkladDto (Przystanek p, KursDto kursDto, Date date)`
- `RozkladDto ()`
- `HashMap< String, TreeSet< $ > > getRozklad ()`
- `void setRozklad (HashMap< String, TreeSet< $ > > rozklad)`
- `RozkladDto addKurs (LiniaDto linia, Time godzinna, boolean odwrotna)`
- `PrzystanekDto getNormalny ()`
- `void setNormalny (PrzystanekDto normalny)`
- `Date getDate ()`
- `void setDate (Date date)`

6.93.1 Opis szczegółowy

Data Transfer Object (DTO) dla rozkładu jazdy.

Definicja w linii 17 pliku [RozkladDto.java](#).

6.93.2 Dokumentacja konstruktora i destruktora

6.93.2.1 RozkladDto() [1/2]

```
com.mklachl.sopkom.model.dto.rozklad.RozkladDto.RozkladDto (
    Przystanek p,
    KursDto kursDto,
    Date date )
```

Konstruktor przyjmujący obiekt Przystanek, KursDto oraz datę.

Parametry

<i>p</i>	Obiekt przystanku.
<i>kursDto</i>	Obiekt DTO kursu.
<i>date</i>	Data rozkładu.

Definicja w linii 88 pliku [RozkladDto.java](#).

6.93.2.2 RozkladDto() [2/2]

```
com.mklachl.sopkom.model.dto.rozklad.RozkladDto.RozkladDto ( )
```

Konstruktor domyślny.

Definicja w linii 96 pliku [RozkladDto.java](#).

6.93.3 Dokumentacja funkcji składowych

6.93.3.1 addKurs()

```
RozkladDto com.mklachl.sopkom.model.dto.rozklad.RozkladDto.addKurs (
    LiniaDto linia,
    Time godzinna,
    boolean odwrotna )
```

Dodaje kurs do rozkładu.

Parametry

<i>linia</i>	Obiekt DTO linii.
<i>godzinna</i>	Czas odjazdu.
<i>odwrotna</i>	Flaga odwrotności.

Zwroca

Obiekt [RozkladDto](#).

Definicja w linii [124](#) pliku [RozkladDto.java](#).

6.93.3.2 getDate()

```
Date com.mklachl.sopkom.model.dto.rozklad.RozkladDto.getDate ( )
```

Zwroca datę rozkładu.

Zwroca

Data rozkładu.

Definicja w linii [149](#) pliku [RozkladDto.java](#).

6.93.3.3 getNormalny()

```
PrzystanekDto com.mklachl.sopkom.model.dto.rozklad.RozkladDto.getNormalny ( )
```

Zwroca obiekt DTO przystanku.

Zwroca

Obiekt DTO przystanku.

Definicja w linii [133](#) pliku [RozkladDto.java](#).

6.93.3.4 getRozklad()

```
HashMap< String, TreeSet< $ > > com.mklachl.sopkom.model.dto.rozklad.RozkladDto.getRozklad ( )
```

Zwraca mapę rozkładu.

Zwraca

Mapa rozkładu.

Definicja w linii 105 pliku [RozkladDto.java](#).

6.93.3.5 setDate()

```
void com.mklachl.sopkom.model.dto.rozklad.RozkladDto.setDate ( Date date )
```

Ustawia datę rozkładu.

Parametry

<i>date</i>	Data rozkładu.
-------------	----------------

Definicja w linii 157 pliku [RozkladDto.java](#).

6.93.3.6 setNormalny()

```
void com.mklachl.sopkom.model.dto.rozklad.RozkladDto.setNormalny ( PrzystanekDto normalny )
```

Ustawia obiekt DTO przystanku.

Parametry

<i>normalny</i>	Obiekt DTO przystanku.
-----------------	------------------------

Definicja w linii 141 pliku [RozkladDto.java](#).

6.93.3.7 setRozklad()

```
void com.mklachl.sopkom.model.dto.rozklad.RozkladDto.setRozklad ( HashMap< String, TreeSet< $ > > rozklad )
```

Ustawia mapę rozkładu.

Parametry

<code>rozklad</code>	Mapa rozkładu.
----------------------	----------------

Definicja w linii 113 pliku [RozkladDto.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.94 Dokumentacja klasy com.mklachl.sopkom.model.dto.rozklad.RozkladKursDto

6.94.1 Opis szczegółowy

Data Transfer Object (DTO) dla kursu w rozkładzie jazdy.

Definicja w linii 6 pliku [RozkladKursDto.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.95 Dokumentacja klasy com.mklachl.sopkom.model.dto.rozklad.RozkladLiniaDto

6.95.1 Opis szczegółowy

Data Transfer Object (DTO) dla linii w rozkładzie jazdy.

Definicja w linii 6 pliku [RozkladLiniaDto.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.96 Dokumentacja klasy com.mklachl.sopkom.config.SecurityConfig

Metody publiczne

- AuthenticationManager [authenticationManager](#) (AuthenticationConfiguration authenticationConfiguration)
throws Exception
- SecurityFilterChain [securityFilterChain](#) (HttpSecurity http) throws Exception
- PasswordEncoder [passwordEncoder](#) ()

6.96.1 Opis szczegółowy

Definicja w linii 28 pliku [SecurityConfig.java](#).

6.96.2 Dokumentacja funkcji składowych

6.96.2.1 authenticationManager()

```
AuthenticationManager com.mklachl.sopkom.config.SecurityConfig.authenticationManager ( AuthenticationConfiguration authenticationConfiguration ) throws Exception
```

Definicja w linii 37 pliku [SecurityConfig.java](#).

6.96.2.2 passwordEncoder()

```
PasswordEncoder com.mklachl.sopkom.config.SecurityConfig.passwordEncoder ( )
```

Definicja w linii 79 pliku [SecurityConfig.java](#).

6.96.2.3 securityFilterChain()

```
SecurityFilterChain com.mklachl.sopkom.config.SecurityConfig.securityFilterChain ( HttpSecurity http ) throws Exception
```

Definicja w linii 42 pliku [SecurityConfig.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/

6.97 Dokumentacja klasy **com.mklachl.sopkom.controller.SimpleMessage**

Metody publiczne

- [SimpleMessage \(\)](#)
- [SimpleMessage \(String message\)](#)
- [String getMessage \(\)](#)
- [void setMessage \(String message\)](#)

Atrybuty publiczne

- String [message](#)

6.97.1 Opis szczegółowy

Klasa [SimpleMessage](#) służy do przechowywania prostych wiadomości tekstowych.

Definicja w linii [6](#) pliku [SimpleMessage.java](#).

6.97.2 Dokumentacja konstruktora i destruktora

6.97.2.1 SimpleMessage() [1/2]

```
com.mklachl.sopkom.controller.SimpleMessage.SimpleMessage( )
```

Konstruktor domyślny

Definicja w linii [12](#) pliku [SimpleMessage.java](#).

6.97.2.2 SimpleMessage() [2/2]

```
com.mklachl.sopkom.controller.SimpleMessage.SimpleMessage( String message )
```

Konstruktor inicjalizujący wiadomość

Parametry

<i>message</i>	wiadomość do ustawienia
----------------	-------------------------

Definicja w linii [20](#) pliku [SimpleMessage.java](#).

6.97.3 Dokumentacja funkcji składowych

6.97.3.1 getMessage()

```
String com.mklachl.sopkom.controller.SimpleMessage.getMessage( )
```

Pobiera wiadomość

Zwraca

wiadomość

Definicja w linii [29](#) pliku [SimpleMessage.java](#).

6.97.3.2 setMessage()

```
void com.mklachl.sopkom.controller.SimpleMessage.setMessage (
    String message )
```

Ustawia wiadomość

Parametry

<i>message</i>	wiadomość do ustawienia
----------------	-------------------------

Definicja w linii [37](#) pliku [SimpleMessage.java](#).

6.97.4 Dokumentacja atrybutów składowych

6.97.4.1 message

```
String com.mklachl.sopkom.controller.SimpleMessage.message
```

Wiadomość tekstowa

Definicja w linii [9](#) pliku [SimpleMessage.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/controller/

6.98 Dokumentacja klasy com.mklachl.sopkom.SOPKoMApplication

Statyczne metody publiczne

- static void [main](#) (String[] args)

6.98.1 Opis szczegółowy

Definicja w linii [7](#) pliku [SOPKoMApplication.java](#).

6.98.2 Dokumentacja funkcji składowych

6.98.2.1 main()

```
static void com.mklachl.sopkom.SOPKoMApplication.main (
    String[] args) [static]
```

Definicja w linii 9 pliku [SOPKoMApplication.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/SOPKoMApplication.java

6.99 Dokumentacja klasy com.mklachl.sopkom.model.entity.User

Metody publiczne

- [User \(\)](#)
- [User \(Long id, String name, String email, String password\)](#)
- [User \(Long id, String name, String email, String password, Set< Role > roles\)](#)
- Long [getId \(\)](#)
- void [setId \(Long id\)](#)
- String [getName \(\)](#)
- void [setName \(String name\)](#)
- String [getLogin \(\)](#)
- void [setLogin \(String login\)](#)
- String [getPassword \(\)](#)
- void [setPassword \(String password\)](#)
- Set< Role > [getRoles \(\)](#)
- void [setRoles \(Set< Role > roles\)](#)

6.99.1 Opis szczegółowy

Klasa reprezentująca encję [User](#).

Definicja w linii 11 pliku [User.java](#).

6.99.2 Dokumentacja konstruktora i destruktora

6.99.2.1 User() [1/3]

```
com.mklachl.sopkom.model.entity.User.User ( )
```

Konstruktor domyślny.

Definicja w linii 44 pliku [User.java](#).

6.99.2.2 User() [2/3]

```
com.mklachl.sopkom.model.entity.User.User (
    Long id,
    String name,
    String email,
    String password )
```

Konstruktor parametryczny.

Parametry

<i>id</i>	Identyfikator użytkownika.
<i>name</i>	Nazwa użytkownika.
<i>email</i>	Login użytkownika.
<i>password</i>	Hasło użytkownika.

Definicja w linii 54 pliku [User.java](#).

6.99.2.3 User() [3/3]

```
com.mklachl.sopkom.model.entity.User.User (
    Long id,
    String name,
    String email,
    String password,
    Set< Role > roles )
```

Konstruktor parametryczny z rolami.

Parametry

<i>id</i>	Identyfikator użytkownika.
<i>name</i>	Nazwa użytkownika.
<i>email</i>	Login użytkownika.
<i>password</i>	Hasło użytkownika.
<i>roles</i>	Role użytkownika.

Definicja w linii 69 pliku [User.java](#).

6.99.3 Dokumentacja funkcji składowych

6.99.3.1 getId()

```
Long com.mklachl.sopkom.model.entity.User.getId ( )
```

Zwraca identyfikator użytkownika.

Zwraca

id Identyfikator użytkownika.

Definicja w linii 81 pliku [User.java](#).

6.99.3.2 getLogin()

```
String com.mklachl.sopkom.model.entity.User.getLogin ( )
```

Zwraca login użytkownika.

Zwraca

login Login użytkownika.

Definicja w linii 113 pliku [User.java](#).

6.99.3.3 getName()

```
String com.mklachl.sopkom.model.entity.User.getName ( )
```

Zwraca nazwę użytkownika.

Zwraca

name Nazwa użytkownika.

Definicja w linii 97 pliku [User.java](#).

6.99.3.4 getPassword()

```
String com.mklachl.sopkom.model.entity.User.getPassword ( )
```

Zwraca hasło użytkownika.

Zwraca

password Hasło użytkownika.

Definicja w linii [129](#) pliku [User.java](#).

6.99.3.5 getRoles()

```
Set< Role > com.mklachl.sopkom.model.entity.User.getRoles ( )
```

Zwraca role użytkownika.

Zwraca

roles [Role](#) użytkownika.

Definicja w linii [145](#) pliku [User.java](#).

6.99.3.6 setId()

```
void com.mklachl.sopkom.model.entity.User.setId (   
    Long id )
```

Ustawia identyfikator użytkownika.

Parametry

<i>id</i>	Identyfikator użytkownika.
-----------	----------------------------

Definicja w linii [89](#) pliku [User.java](#).

6.99.3.7 setLogin()

```
void com.mklachl.sopkom.model.entity.User.setLogin (   
    String login )
```

Ustawia login użytkownika.

Parametry

<i>login</i>	Login użytkownika.
--------------	--------------------

Definicja w linii [121](#) pliku [User.java](#).

6.99.3.8 setName()

```
void com.mklachl.sopkom.model.entity.User.setName (
    String name )
```

Ustawia nazwę użytkownika.

Parametry

<i>name</i>	Nazwa użytkownika.
-------------	--------------------

Definicja w linii [105](#) pliku [User.java](#).

6.99.3.9 setPassword()

```
void com.mklachl.sopkom.model.entity.User.setPassword (
    String password )
```

Ustawia hasło użytkownika.

Parametry

<i>password</i>	Hasło użytkownika.
-----------------	--------------------

Definicja w linii [137](#) pliku [User.java](#).

6.99.3.10 setRoles()

```
void com.mklachl.sopkom.model.entity.User.setRoles (
    Set< Role > roles )
```

Ustawia role użytkownika.

Parametry

<i>roles</i>	Role użytkownika.
--------------	-----------------------------------

Definicja w linii [153](#) pliku [User.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.100 Dokumentacja klasy **com.mklachl.sopkom.controller.UserController**

Metody publiczne

- [UserDto getLoggedInUser \(\)](#)

6.100.1 Opis szczegółowy

Definicja w linii [17](#) pliku [UserController.java](#).

6.100.2 Dokumentacja funkcji składowych

6.100.2.1 [getLoggedInUser\(\)](#)

`UserDto com.mklachl.sopkom.controller.UserController.getLoggedInUser ()`

Endpoint do pobrania danych zalogowanego użytkownika.

Zwroca

UserDto z danymi użytkownika

Definicja w linii [27](#) pliku [UserController.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.101 Dokumentacja klasy com.mklachl.sopkom.model.dto.UserDto

Metody publiczne

- [UserDto \(\)](#)
- [UserDto \(User user\)](#)
- [Long getId \(\)](#)
- [void setId \(Long id\)](#)
- [String getName \(\)](#)
- [void setName \(String name\)](#)
- [String getEmail \(\)](#)
- [void setEmail \(String email\)](#)
- [String getPassword \(\)](#)
- [void setPassword \(String password\)](#)
- [Set< Role > getRoles \(\)](#)
- [void setRoles \(Set< Role > roles\)](#)
- [String getRole \(\)](#)
- [void setRole \(String role\)](#)

6.101.1 Opis szczegółowy

Data Transfer Object (DTO) dla encji User.

Definicja w linii 10 pliku [UserDto.java](#).

6.101.2 Dokumentacja konstruktora i destruktora

6.101.2.1 UserDto() [1/2]

```
com.mklachl.sopkom.model.dto.UserDto.UserDto ( )
```

Konstruktor domyślny.

Definicja w linii 45 pliku [UserDto.java](#).

6.101.2.2 UserDto() [2/2]

```
com.mklachl.sopkom.model.dto.UserDto.UserDto (
    User user )
```

Konstruktor przyjmujący obiekt encji User.

Parametry

<i>user</i>	Obiekt encji User.
-------------	--------------------

Definicja w linii [52](#) pliku [UserDto.java](#).

6.101.3 Dokumentacja funkcji składowych

6.101.3.1 getEmail()

```
String com.mklachl.sopkom.model.dto.UserDto.getEmail ( )
```

Zwraca email użytkownika.

Zwraca

email Email użytkownika.

Definicja w linii [102](#) pliku [UserDto.java](#).

6.101.3.2 getId()

```
Long com.mklachl.sopkom.model.dto.UserDto.getId ( )
```

Zwraca identyfikator użytkownika.

Zwraca

id Identyfikator użytkownika.

Definicja w linii [70](#) pliku [UserDto.java](#).

6.101.3.3 getName()

```
String com.mklachl.sopkom.model.dto.UserDto.getName ( )
```

Zwraca nazwę użytkownika.

Zwraca

name Nazwa użytkownika.

Definicja w linii [86](#) pliku [UserDto.java](#).

6.101.3.4 getPassword()

```
String com.mklachl.sopkom.model.dto.UserDto.getPassword ( )
```

Zwraca hasło użytkownika.

Zwraca

password Hasło użytkownika.

Definicja w linii 118 pliku [UserDto.java](#).

6.101.3.5 getRole()

```
String com.mklachl.sopkom.model.dto.UserDto.getRole ( )
```

Zwraca nazwę roli użytkownika.

Zwraca

role Nazwa roli użytkownika.

Definicja w linii 150 pliku [UserDto.java](#).

6.101.3.6 getRoles()

```
Set< Role > com.mklachl.sopkom.model.dto.UserDto.getRoles ( )
```

Zwraca role użytkownika.

Zwraca

roles Role użytkownika.

Definicja w linii 134 pliku [UserDto.java](#).

6.101.3.7 setEmail()

```
void com.mklachl.sopkom.model.dto.UserDto.setEmail (   
    String email )
```

Ustawia email użytkownika.

Parametry

<i>email</i>	Email użytkownika.
--------------	--------------------

Definicja w linii [110](#) pliku [UserDto.java](#).

6.101.3.8 setId()

```
void com.mklachl.sopkom.model.dto.UserDto.setId (
    Long id )
```

Ustawia identyfikator użytkownika.

Parametry

<i>id</i>	Identyfikator użytkownika.
-----------	----------------------------

Definicja w linii [78](#) pliku [UserDto.java](#).

6.101.3.9 setName()

```
void com.mklachl.sopkom.model.dto.UserDto.setName (
    String name )
```

Ustawia nazwę użytkownika.

Parametry

<i>name</i>	Nazwa użytkownika.
-------------	--------------------

Definicja w linii [94](#) pliku [UserDto.java](#).

6.101.3.10 setPassword()

```
void com.mklachl.sopkom.model.dto.UserDto.setPassword (
    String password )
```

Ustawia hasło użytkownika.

Parametry

<i>password</i>	Hasło użytkownika.
-----------------	--------------------

Definicja w linii 126 pliku [UserDto.java](#).

6.101.3.11 setRole()

```
void com.mklachl.sopkom.model.dto.UserDto.setRole (
    String role )
```

Ustawia nazwę roli użytkownika.

Parametry

<code>role</code>	Nazwa roli użytkownika.
-------------------	-------------------------

Definicja w linii 158 pliku [UserDto.java](#).

6.101.3.12 setRoles()

```
void com.mklachl.sopkom.model.dto.UserDto.setRoles (
    Set< Role > roles )
```

Ustawia role użytkownika.

Parametry

<code>roles</code>	Role użytkownika.
--------------------	-------------------

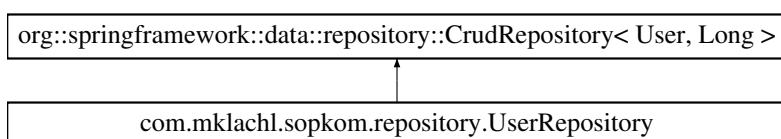
Definicja w linii 142 pliku [UserDto.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/mo

6.102 Dokumentacja interfejsu com.mklachl.sopkom.repository.UserRepository

Diagram dziedziczenia dla com.mklachl.sopkom.repository.UserRepository



Metody publiczne

- [User findByLogin \(String login\)](#)

6.102.1 Opis szczegółowy

Interfejs repository dla encji User

Definicja w linii 9 pliku [UserRepository.java](#).

6.102.2 Dokumentacja funkcji składowych

6.102.2.1 [findByLogin\(\)](#)

```
User com.mklachl.sopkom.repository.UserRepository.findByLogin (
    String login )
```

Znajduje użytkownika po loginie

Parametry

<i>login</i>	login
--------------	-------

Zwroca

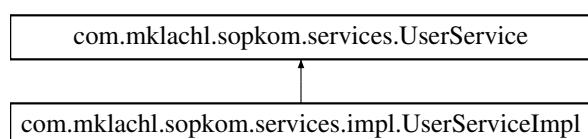
Użytkownik

Dokumentacja dla tego interfejsu została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/repo

6.103 Dokumentacja interfejsu [com.mklachl.sopkom.services.UserService](#)

Diagram dziedziczenia dla [com.mklachl.sopkom.services.UserService](#)



Metody publiczne

- void [saveUser \(User user\)](#)
- void [saveUser \(UserDto userDto\)](#)
- [User findUserByLogin \(String email\)](#)
- [User findUserById \(Long id\)](#)

6.103.1 Opis szczegółowy

Interfejs serwisu dla obsługi użytkowników

Definicja w linii 9 pliku [UserService.java](#).

6.103.2 Dokumentacja funkcji składowych

6.103.2.1 [findUserById\(\)](#)

```
User com.mklachl.sopkom.services.UserService.findUserById (
    Long id )
```

Znajduje użytkownika po ID

Parametry

<i>id</i>	ID użytkownika
-----------	----------------

Zwroca

znaleziony użytkownik

Implementowany w [com.mklachl.sopkom.services.impl.UserServiceImpl](#).

6.103.2.2 [findUserByLogin\(\)](#)

```
User com.mklachl.sopkom.services.UserService.findUserByLogin (
    String email )
```

Znajduje użytkownika po loginie

Parametry

<i>email</i>	login
--------------	-------

Zwraca

znaleziony użytkownik

Implementowany w [com.mklachl.sopkom.services.impl.UserServiceImpl](#).

6.103.2.3 saveUser() [1/2]

```
void com.mklachl.sopkom.services.UserService.saveUser (  
    User user )
```

Zapisuje nowego użytkownika

Parametry

user	encja użytkownika
------	-------------------

Implementowany w [com.mklachl.sopkom.services.impl.UserServiceImpl](#).

6.103.2.4 saveUser() [2/2]

```
void com.mklachl.sopkom.services.UserService.saveUser (  
    UserDto userDto )
```

Zapisuje nowego użytkownika na podstawie DTO

Parametry

userDto	DTO użytkownika
---------	-----------------

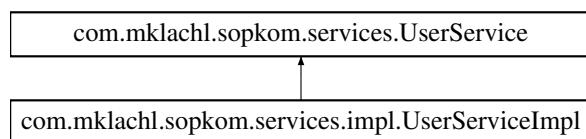
Implementowany w [com.mklachl.sopkom.services.impl.UserServiceImpl](#).

Dokumentacja dla tego interfejsu została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/se

6.104 Dokumentacja klasy **com.mklachl.sopkom.services.impl.UserServiceImpl**

Diagram dziedziczenia dla com.mklachl.sopkom.services.impl.UserServiceImpl



Metody publiczne

- `UserServiceImpl (UserRepository userRepository, PasswordEncoder passwordEncoder)`
- `void saveUser (UserDto userDto)`
- `void saveUser (User userDto)`
- `User findUserByLogin (String login)`
- `User findUserById (Long id)`

6.104.1 Opis szczegółowy

Definicja w linii 14 pliku [UserServiceImpl.java](#).

6.104.2 Dokumentacja konstruktora i destruktora

6.104.2.1 UserServiceImpl()

```
com.mklachl.sopkom.services.impl.UserServiceImpl.UserServiceImpl ( 
    UserRepository userRepository,
    PasswordEncoder passwordEncoder )
```

Definicja w linii 20 pliku [UserServiceImpl.java](#).

6.104.3 Dokumentacja funkcji składowych

6.104.3.1 findUserById()

```
User com.mklachl.sopkom.services.impl.UserServiceImpl.findUserById ( 
    Long id )
```

Znajduje użytkownika po ID

Parametry

<code>id</code>	ID użytkownika
-----------------	----------------

Zwroca

znaleziony użytkownik

Implementuje [com.mklachl.sopkom.services.UserService](#).

Definicja w linii 57 pliku [UserServiceImpl.java](#).

6.104.3.2 findUserByLogin()

```
User com.mklachl.sopkom.services.impl.UserServiceImpl.findUserByLogin (
    String email )
```

Znajduje użytkownika po loginie

Parametry

<i>email</i>	login
--------------	-------

Zwroca

znaleziony użytkownik

Implementuje [com.mklachl.sopkom.services.UserService](#).

Definicja w linii [51](#) pliku [UserServiceImpl.java](#).

6.104.3.3 saveUser() [1/2]

```
void com.mklachl.sopkom.services.impl.UserServiceImpl.saveUser (
    User user )
```

Zapisuje nowego użytkownika

Parametry

<i>user</i>	encja użytkownika
-------------	-------------------

Implementuje [com.mklachl.sopkom.services.UserService](#).

Definicja w linii [39](#) pliku [UserServiceImpl.java](#).

6.104.3.4 saveUser() [2/2]

```
void com.mklachl.sopkom.services.impl.UserServiceImpl.saveUser (
    UserDto userDto )
```

Zapisuje nowego użytkownika na podstawie DTO

Parametry

<i>userDto</i>	DTO użytkownika
----------------	-----------------

Implementuje [com.mklachl.sopkom.services.UserService](#).

Definicja w linii [26](#) pliku [UserServiceImpl.java](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/se

Rozdział 7

Dokumentacja plików

7.1 Dokumentacja pliku

X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/←
SOPKoM_Backend/src/main/java/com/mklachl/sopkom/config/←
SecurityConfig.java

Komponenty

- class [com.mklachl.sopkom.config.SecurityConfig](#)

Pakiety

- package [com.mklachl.sopkom.config](#)

7.2 SecurityConfig.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.config;
00002
00003 import java.util.List;
00004
00005 import org.springframework.beans.factory.annotation.Autowired;
00006 import org.springframework.beans.factory.annotation.Value;
00007 import org.springframework.context.annotation.Bean;
00008 import org.springframework.context.annotation.Configuration;
00009 import org.springframework.security.authentication.AuthenticationManager;
00010 import
00011     org.springframework.security.config.annotation.authentication.configuration.AuthenticationConfiguration;
00012 import org.springframework.security.config.annotation.method.configuration.EnableMethodSecurity;
00013 import org.springframework.security.config.annotation.web.builders.HttpSecurity;
00014 import org.springframework.security.config.annotation.web.configurers.AbstractHttpConfigurer;
00015 import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
00016 import org.springframework.security.crypto.password.PasswordEncoder;
00017 import org.springframework.security.web.SecurityFilterChain;
00018 import org.springframework.security.web.util.matcher.AntPathRequestMatcher;
00019 import org.springframework.web.cors.CorsConfiguration;
00020 import org.springframework.web.cors.CorsConfigurationSource;
00021 import org.springframework.web.cors.UrlBasedCorsConfigurationSource;
00022
00023 import com.mklachl.sopkom.security.CustomUserDetailsService;
00024
00025 @Configuration
00026 @EnableWebSecurity
```

```
00027 @EnableMethodSecurity
00028 public class SecurityConfig {
00029
00030     @Autowired
00031     private CustomUserDetailsService userDetailsService;
00032
00033     @Value("${frontend.url}")
00034     private String frontendURL;
00035
00036     @Bean
00037     public AuthenticationManager authenticationManager(AuthenticationConfiguration
00038         authenticationConfiguration) throws Exception {
00039         return authenticationConfiguration.getAuthenticationManager();
00040     }
00041
00042     @Bean
00043     public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {
00044         return http
00045             .authorizeHttpRequests(authorizeHttpRequests -> authorizeHttpRequests
00046                 .requestMatchers("/logout").permitAll()
00047                 // TODO testing, later for delete
00048                 .requestMatchers("/api/**").permitAll()
00049                 .requestMatchers("/error").permitAll()
00050                 .requestMatchers("/not_auth").permitAll()
00051
00052                 .anyRequest().authenticated())
00053             .formLogin(
00054                 form -> form
00055
00056                 .loginPage("/login")
00057                 .permitAll()
00058                 .loginProcessingUrl("/login")
00059                 .permitAll()
00060                 .defaultSuccessUrl(frontendURL, true)
00061
00062             ).logout(
00063                 logout -> logout
00064                     .logoutRequestMatcher(new AntPathRequestMatcher("/logout"))
00065                     .logoutSuccessUrl("/home")
00066                     .permitAll()
00067             )
00068
00069
00070         .cors(t -> t.configurationSource(corsConfiguration()))
00071
00072         // .csrf(AbstractHttpConfigurer::disable)
00073
00074         .csrf((csrf) -> csrf.disable())
00075         .build();
00076     }
00077
00078     @Bean
00079     public PasswordEncoder passwordEncoder() {
00080         return new BCryptPasswordEncoder();
00081     }
00082
00083     @Bean
00084     CorsConfigurationSource corsConfiguration(){
00085         CorsConfiguration configuration = new CorsConfiguration();
00086         configuration.setAllowedOrigins(List.of(frontendURL, "http://192.168.100.109:3000",
00087             "http://localhost:3000"));
00088         configuration.addAllowedHeader("*");
00089         configuration.addAllowedMethod("*");
00090         configuration.setAllowCredentials(true);
00091         UrlBasedCorsConfigurationSource urlBasedCorsConfigurationSource = new
00092             UrlBasedCorsConfigurationSource();
00093         urlBasedCorsConfigurationSource.registerCorsConfiguration("/**", configuration);
00094         return urlBasedCorsConfigurationSource;
00095     }
00096 }
```

7.3 Dokumentacja pliku

X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/ ←
SOPKoM_Backend/src/main/java/com/mklachl/sopkom/controller/ ←
AutobusController.java

Komponenty

- class [com.mklachl.sopkom.controller.AutobusController](#)

Pakiety

- package [com.mklachl.sopkom.controller](#)

7.4 AutobusController.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.controller;
00002
00003 import java.util.ArrayList;
00004 import java.util.Date;
00005 import java.util.List;
00006 import java.util.Optional;
00007
00008 import org.apache.logging.log4j.util.Strings;
00009 import org.springframework.beans.factory.annotation.Autowired;
00010 import org.springframework.http.HttpStatus;
00011 import org.springframework.http.MediaType;
00012 import org.springframework.http.ResponseEntity;
00013 import org.springframework.web.bind.annotation.*;
00014
00015 import com.mklachl.sopkom.model.dto.AutobusDto;
00016 import com.mklachl.sopkom.model.entity.Autobus;
00017 import com.mklachl.sopkom.repository.AutobusRepository;
00018 import com.mklachl.sopkom.services.AutobusService;
00019
00020 /**
00021 * Kontroler dla zasobu Autobus.
00022 * Obsługuje operacje tworzenia, pobierania, aktualizacji i usuwania autobusów.
00023 */
00024 @RestController
00025 @RequestMapping("api/autobus")
00026 public class AutobusController {
00027
00028     @Autowired
00029     AutobusService autobusService;
00030
00031     @Autowired
00032     AutobusRepository autobusRepo;
00033
00034     @GetMapping(path = {"test"}, produces = {MediaType.TEXT_PLAIN_VALUE})
00035     public ResponseEntity<String> test() {
00036         return new ResponseEntity<>("test_autobus", HttpStatus.OK);
00037     }
00038
00039     @GetMapping(path = {"template"}, produces = {MediaType.APPLICATION_JSON_VALUE})
00040     public ResponseEntity<?> template() {
00041         AutobusDto dto = new AutobusDto();
00042         dto.setAutobusId(Long.valueOf(10));
00043         dto.setNumerRejestracyjny("SK55555");
00044         dto.setPrzebieg(41247.12f);
00045         dto.setPrzegladWaznyDo(new Date());
00046         dto.setStatus("OK");
00047         return new ResponseEntity<>(dto, HttpStatus.OK);
00048     }
00049
00050     @PostMapping("/save")
00051     public ResponseEntity<?> addAutobus(@RequestBody AutobusDto autobusDto) {
00052         var autobus = autobusService.saveAutobus(autobusDto);
00053         return new ResponseEntity<>(new AutobusDto(autobus), HttpStatus.OK);
00054     }
00055 }
```

```

00068
00074     @GetMapping(path = "/{id}", produces = {MediaType.APPLICATION_JSON_VALUE})
00075     public ResponseEntity<?> getAutobus(@PathVariable("id") Long id) {
00076         var autobus = autobusService.findAutobusById(id.longValue());
00077         if (autobus.isEmpty()) {
00078             return new ResponseEntity<>(new SimpleMessage("not found"), HttpStatus.NOT_FOUND);
00079         }
00080         return new ResponseEntity<>(new AutobusDto(autobus.get()), HttpStatus.OK);
00081     }
00082
00088     @DeleteMapping("/{id}")
00089     public ResponseEntity<?> deleteAutobus(@PathVariable("id") Long id) {
00090         if (autobusRepo.findById(id).isEmpty()) {
00091             return new ResponseEntity<>(new SimpleMessage("no content"), HttpStatus.OK);
00092         }
00093         autobusRepo.deleteById(id);
00094         return new ResponseEntity<>(new SimpleMessage("deleted"), HttpStatus.OK);
00095     }
00096
00103     @PatchMapping("/{id}")
00104     public ResponseEntity<?> updateAutobus(@PathVariable("id") Long id, @RequestBody AutobusDto
00105     autobusDto) {
00106         Optional<Autobus> autobus = autobusRepo.findById(id);
00107         if (autobus.isEmpty()) {
00108             return new ResponseEntity<>(new SimpleMessage("not found"), HttpStatus.NOT_FOUND);
00109         }
00110         Autobus in = autobus.get();
00111         in.setNumerRejestracyjny(autobusDto.getNumerRejestracyjny());
00112         in.setStatus(autobusDto.getStatus());
00113         in.setPrzebieg(autobusDto.getPrzebieg());
00114         in.setPrzegladWaznyDo(autobusDto.getPrzegladWaznyDo());
00115         in = autobusRepo.save(in);
00116         return new ResponseEntity<>(new SimpleMessage("patched"), HttpStatus.OK);
00117     }
00118
00122     @GetMapping(path = {"", "/"})
00123     public ResponseEntity<?> getAllBuses() {
00124         List<AutobusDto> list = new ArrayList<>();
00125         autobusRepo.findAll().forEach(autobus -> list.add(new AutobusDto(autobus)));
00126         return new ResponseEntity<>(list, HttpStatus.OK);
00127     }
00128
00134     @GetMapping(path = {"/filtered"})
00135     public ResponseEntity<?> getAllBusesFilteredStatus(@RequestParam(name = "status", required =
00136     false) String status) {
00137         List<AutobusDto> list = new ArrayList<>();
00138         if (status == null || status.isBlank() || status.equals("ALL")) {
00139             autobusRepo.findAll().forEach(autobus -> list.add(new AutobusDto(autobus)));
00140         } else {
00141             autobusRepo.findByStatusLike("%" + status + "%").forEach(autobus -> list.add(new
00142             AutobusDto(autobus)));
00143         }
00144     }

```

7.5 Dokumentacja pliku

X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/←
 SOPKoM_Backend/src/main/java/com/mklachl/sopkom/controller/←
Controller.java

Komponenty

- class [com.mklachl.sopkom.controller.Controller](#)

Pakiety

- package [com.mklachl.sopkom.controller](#)

7.6 Controller.java

[Idź do dokumentacji tego pliku.](#)

```

00001 package com.mklachl.sopkom.controller;
00002
00003 import org.springframework.http.HttpStatus;
00004 import org.springframework.http.MediaType;
00005 import org.springframework.http.ResponseEntity;
00006 import org.springframework.web.bind.annotation.GetMapping;
00007
00011 @org.springframework.stereotype.Controller
00012 public class Controller {
00013
00018     @GetMapping(path = "/not_auth", produces = MediaType.TEXT_PLAIN_VALUE)
00019     public ResponseEntity<String> getString() {
00020         return new ResponseEntity<>("hello not auth", HttpStatus.OK);
00021     }
00022
00027     @GetMapping(path = "/get2")
00028     public ResponseEntity<String> getString2() {
00029         return new ResponseEntity<String>("authenticated only", HttpStatus.OK);
00030     }
00031 }
```

7.7 Dokumentacja pliku

X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/←
 SOPKoM_Backend/src/main/java/com/mklachl/sopkom/controller/←
HarmonogramController.java

Komponenty

- class [com.mklachl.sopkom.controller.HarmonogramController](#)

Pakiety

- package [com.mklachl.sopkom.controller](#)

7.8 HarmonogramController.java

[Idź do dokumentacji tego pliku.](#)

```

00001 package com.mklachl.sopkom.controller;
00002
00003 import java.util.ArrayList;
00004 import java.util.List;
00005 import java.util.stream.Collectors;
00006
00007 import org.springframework.beans.factory.annotation.Autowired;
00008 import org.springframework.http.ResponseEntity;
00009 import org.springframework.web.bind.annotation.*;
00010
00011 import com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto;
00012 import com.mklachl.sopkom.repository.HarmonogramRepository;
00013 import com.mklachl.sopkom.services.HarmonogramService;
00014
00019 @RestController
00020 @RequestMapping("api/harmonogram")
00021 public class HarmonogramController {
00022
00023     @Autowired
00024     HarmonogramService service;
00025
00026     @Autowired
00027     HarmonogramRepository repository;
00028 }
```

```

00034     @PostMapping
00035     public ResponseEntity<HarmonogramDto> createHarmonogram(@RequestBody HarmonogramDto harmonogram) {
00036         return ResponseEntity.ok(harmonogram);
00037     }
00038
00043     @GetMapping
00044     public ResponseEntity<List<HarmonogramDto>> getAllHarmonograms() {
00045         List<HarmonogramDto> dtoList = repository.findAll().stream()
00046             .map(HarmonogramDto::new)
00047             .collect(Collectors.toList());
00048         return ResponseEntity.ok(dtoList);
00049     }
00050
00056     @GetMapping("/{id}")
00057     public ResponseEntity<HarmonogramDto> getHarmonogramById(@PathVariable(name = "id") Short id) {
00058         var harmonogram = repository.findById(id).get();
00059         if (harmonogram != null) {
00060             return ResponseEntity.ok(new HarmonogramDto(harmonogram));
00061         } else {
00062             return ResponseEntity.notFound().build();
00063         }
00064     }
00065
00071     @DeleteMapping("/{id}")
00072     public ResponseEntity<Void> deleteHarmonogram(@PathVariable Short id) {
00073         repository.deleteById(id);
00074         return ResponseEntity.ok().build();
00075     }
00076 }
```

7.9 Dokumentacja pliku

X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/ ↵
SOPKoM_Backend/src/main/java/com/mklachl/sopkom/controller/ ↵
IncydentController.java

Komponenty

- class [com.mklachl.sopkom.controller.IncydentController](#)

Pakietы

- package [com.mklachl.sopkom.controller](#)

7.10 IncydentController.java

[Idź do dokumentacji tego pliku.](#)

```

00001 package com.mklachl.sopkom.controller;
00002
00003 import java.util.Date;
00004 import java.util.List;
00005 import java.util.stream.Collectors;
00006
00007 import org.springframework.beans.factory.annotation.Autowired;
00008 import org.springframework.format.annotation.DateTimeFormat;
00009 import org.springframework.http.ResponseEntity;
00010 import org.springframework.web.bind.annotation.*;
00011
00012 import com.mklachl.sopkom.exceptions.IncydentException;
00013 import com.mklachl.sopkom.model.dto.IncydentDto;
00014 import com.mklachl.sopkom.model.dto.IncydentDtoOutput;
00015 import com.mklachl.sopkom.model.entity.Autobus;
00016 import com.mklachl.sopkom.model.entity.Incydent;
00017 import com.mklachl.sopkom.model.entity.Kierowca;
00018 import com.mklachl.sopkom.model.entity.Przejazd;
00019 import com.mklachl.sopkom.repository.IncydentRepository;
00020 import com.mklachl.sopkom.services.IncydentService;
```

```

00021
00026 @RestController
00027 @RequestMapping("/api/incydent")
00028 public class IncydentController {
00029
00030     @Autowired
00031     public IncydentService incydentService;
00032
00033     @Autowired
00034     public IncydentRepository incydentRepository;
00035
00041     @PostMapping("/{save}", "/save/")
00042     public ResponseEntity<?> saveIncydent(@RequestBody IncydentDto input) {
00043         try {
00044             var i = incydentService.saveIncydent(input);
00045             return ResponseEntity.ok(new IncydentDto(i));
00046         } catch (IncydentException e) {
00047             e.printStackTrace();
00048             return ResponseEntity.notFound().eTag(e.getMessage()).build();
00049         } catch (Exception e) {
00050             return ResponseEntity.internalServerError().eTag(e.getMessage()).build();
00051         }
00052     }
00053
00060     @PatchMapping("{"/{id}}", "{/id}/")
00061     public ResponseEntity<?> updateIncydent(@PathVariable(name = "id") Long id, @RequestBody
00062     IncydentDto input) {
00063         try {
00064             incydentService.updateIncydent(id, input);
00065             return ResponseEntity.ok().build();
00066         } catch (IncydentException e) {
00067             e.printStackTrace();
00068             return ResponseEntity.notFound().eTag(e.getMessage()).build();
00069         }
00070     }
00076     @GetMapping("{"/{incydent_id}}", "{/incydent_id}/")
00077     public ResponseEntity<?> getIncydent(@PathVariable(name = "incydent_id") Long incydentId) {
00078         try {
00079             var incydent = incydentService.findIncydentById(incydentId).orElseThrow();
00080             IncydentDtoOutput output = new IncydentDtoOutput(incydent);
00081             return ResponseEntity.ok(output);
00082         } catch (Exception e) {
00083             return ResponseEntity.notFound().build();
00084         }
00085     }
00086
00092     @DeleteMapping("{"/{incydent_id}}", "{/incydent_id}/")
00093     public ResponseEntity<?> deleteIncydent(@PathVariable(name = "incydent_id") Long incydentId) {
00094         try {
00095             incydentRepository.deleteById(incydentId);
00096             return ResponseEntity.ok("deleted");
00097         } catch (Exception e) {
00098             return ResponseEntity.notFound().build();
00099         }
00100     }
00107     @GetMapping()
00108     public ResponseEntity<?> getIncydents(@RequestParam(name = "typ", required = false) String typ) {
00109         try {
00110             List<Incydent> incydents = typ == null ? incydentRepository.findAll() :
00111                 incydentRepository.findAllByTyp(typ);
00112             return ResponseEntity.ok(incydents.stream().map(IncydentDtoOutput::new).collect(Collectors.toList()));
00113         } catch (Exception e) {
00114             return ResponseEntity.notFound().build();
00115         }
00116     }
00122     @GetMapping("/beforeDate")
00123     public ResponseEntity<?> getIncydentsBeforeDate(@RequestParam(name = "date") @DateTimeFormat(pattern = "dd.MM.yyyy") Date date) {
00124         try {
00125             var incydents = incydentRepository.findAllByDateBefore(date);
00126             return
00127                 ResponseEntity.ok(incydents.stream().map(IncydentDtoOutput::new).collect(Collectors.toList()));
00128         } catch (Exception e) {
00129             return ResponseEntity.notFound().build();
00130         }
00131     }
00137     @GetMapping("/afterDate")
00138     public ResponseEntity<?> getIncydentsAfterDate(@RequestParam(name = "date") @DateTimeFormat(pattern = "dd.MM.yyyy") Date date) {
00139         try {
00140             var incydents = incydentRepository.findAllByDateAfter(date);
00141             return

```

```

    ResponseEntity.ok(incydents.stream().map(IncydentDtoOutput::new).collect(Collectors.toList())));
00142     } catch (Exception e) {
00143         return ResponseEntity.notFound().build();
00144     }
00145 }
00146
00153 @GetMapping("/betweenDates")
00154 public ResponseEntity<?> getIncydentsBetweenDates(@RequestParam(name = "startDate")
00155 @DateTimeFormat(pattern = "dd.MM.yyyy") Date startDate,
00156                                     @RequestParam(name = "endDate")
00157 @DateTimeFormat(pattern = "dd.MM.yyyy") Date endDate) {
00158     try {
00159         var incydents = incydentRepository.findAllByDateBetween(startDate, endDate);
00160         return
00161     }
00162 }
00163
00169 @GetMapping("/byAutobus")
00170 public ResponseEntity<?> getIncydentsByAutobus(@RequestParam(name = "autobus_id") Long autobusId)
00171 {
00172     try {
00173         Autobus autobus = new Autobus(); // Assuming Autobus is an entity and you have a way to
00174         get it by ID
00175         autobus.setAutobusId(autobusId); // Set the ID to the Autobus object
00176         var incydents = incydentRepository.findAllByAutobus(autobus);
00177         return
00178     }
00179 }
00180
00186 @GetMapping("/byKierowca")
00187 public ResponseEntity<?> getIncydentsByKierowca(@RequestParam(name = "kierowca_id") Long
00188 kierowcaId) {
00189     try {
00190         Kierowca kierowca = new Kierowca(); // Assuming Kierowca is an entity and you have a way
00191         to get it by ID
00192         kierowca.setKierowcaId(kierowcaId); // Set the ID to the Kierowca object
00193         var incydents = incydentRepository.findAllByKierowca(kierowca);
00194         return
00195     }
00196 }
00197
00203 @GetMapping("/byPrzejazd")
00204 public ResponseEntity<?> getIncydentsByPrzejazd(@RequestParam(name = "przejazd_id") Long
00205 przejazdId) {
00206     try {
00207         Przejazd przejazd = new Przejazd(); // Assuming Przejazd is an entity and you have a way
00208         to get it by ID
00209         przejazd.setPrzejazdId(przejazdId); // Set the ID to the Przejazd object
00210         var incydents = incydentRepository.findAllByPrzejazd(przejazd);
00211         return
00212     }
00213 }
00214 }

```

7.11 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/controller/Kierowca← Controller.java

Komponenty

- class [com.mklachl.sopkom.controller.KierowcaController](#)

Pakiety

- package com.mklachl.sopkom.controller

7.12 KierowcaController.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.controller;
00002
00003 import java.util.Date;
00004 import java.util.List;
00005 import java.util.Optional;
00006 import java.util.stream.Collectors;
00007
00008 import org.springframework.beans.factory.annotation.Autowired;
00009 import org.springframework.http.HttpStatus;
00010 import org.springframework.http.MediaType;
00011 import org.springframework.http.ResponseEntity;
00012 import org.springframework.web.bind.annotation.*;
00013
00014 import com.mklachl.sopkom.helpers.Helpers;
00015 import com.mklachl.sopkom.model.dto.KierowcaDto;
00016 import com.mklachl.sopkom.model.entity.Kierowca;
00017 import com.mklachl.sopkom.model.entity.User;
00018 import com.mklachl.sopkom.repository.KierowcaRepository;
00019 import com.mklachl.sopkom.services.KierowcaService;
00020 import com.mklachl.sopkom.services.UserService;
00021 import com.mklachl.sopkom.services.impl.KierowcaServiceImpl;
00022
00023 @RestController
00024 @RequestMapping("/api/kierowca")
00025 public class KierowcaController {
00026
00027     KierowcaService kierowcaService;
00028     UserService userService;
00029
00030     private KierowcaRepository kierowcaRepository;
00031
00032     public KierowcaController(KierowcaRepository kierowcaRepository, UserService userService) {
00033         this.kierowcaRepository = kierowcaRepository;
00034         kierowcaService = new KierowcaServiceImpl(kierowcaRepository, userService);
00035         this.userService = userService;
00036     }
00037
00038     @GetMapping(path = {"template"}, produces = {MediaType.APPLICATION_JSON_VALUE})
00039     public ResponseEntity<KierowcaDto> template(@RequestHeader("Content-Type") String type) {
00040         KierowcaDto dto = new KierowcaDto();
00041         dto.setKierowcaId(Long.valueOf(12));
00042         dto.setNazwisko("Kowalski");
00043         dto.setImie("Jan");
00044         dto.setPrawoJazdyWazneDo(new Date());
00045         dto.setPesel("93050203761");
00046         return new ResponseEntity<>(dto, HttpStatus.OK);
00047     }
00048
00049     @PostMapping("/save")
00050     public ResponseEntity<KierowcaDto> saveKierowca(@RequestBody KierowcaDto kierowca) {
00051         System.out.println(kierowca);
00052         Kierowca savedKierowca = kierowcaService.saveKierowca(kierowca);
00053         KierowcaDto kierowcaDto = new KierowcaDto(savedKierowca);
00054         return new ResponseEntity<>(kierowcaDto, HttpStatus.CREATED);
00055     }
00056
00057     @GetMapping(path = "/{id}", produces = {MediaType.APPLICATION_JSON_VALUE})
00058     public ResponseEntity<KierowcaDto> findKierowcaById(@PathVariable(name = "id") Long id) {
00059         Optional<Kierowca> optionalKierowca = kierowcaService.findKierowcaById(id);
00060         return optionalKierowca.map(kierowca -> ResponseEntity.ok(new KierowcaDto(kierowca)))
00061             .orElse(ResponseEntity.notFound().build());
00062     }
00063
00064     @DeleteMapping("/{id}")
00065     public ResponseEntity<?> deleteKierowca(@PathVariable("id") Long id) {
00066         if (kierowcaRepository.findById(id).isEmpty()) {
00067             return new ResponseEntity<>(new SimpleMessage("no content"), HttpStatus.OK);
00068         }
00069         kierowcaRepository.deleteById(id);
00070         return new ResponseEntity<>(new SimpleMessage("deleted"), HttpStatus.OK);
00071     }
00072
00073     @PatchMapping("/{id}")
00074     public ResponseEntity<?> patchKierowca(@PathVariable("id") Long id, @RequestBody Kierowca kierowca) {
00075         Kierowca updatedKierowca = kierowcaService.patchKierowca(id, kierowca);
00076         KierowcaDto kierowcaDto = new KierowcaDto(updatedKierowca);
00077         return new ResponseEntity<>(kierowcaDto, HttpStatus.OK);
00078     }
00079 }
```

```

00108     public ResponseEntity<?> updateKierowca(@PathVariable("id") Long id, @RequestBody KierowcaDto
00109         kierowcaDto) {
00110             Optional<Kierowca> keirowcaOp = kierowcaRepository.findById(id);
00111             if (keirowcaOp.isEmpty()) {
00112                 return new ResponseEntity<?>(new SimpleMessage("not found"), HttpStatus.NOT_FOUND);
00113             }
00114
00115             Kierowca kierowca = keirowcaOp.get();
00116             kierowca.setPrawoJazdyWazneDo(kierowcaDto.getPrawoJazdyWazneDo());
00117             kierowca.setImie(kierowcaDto.getImie());
00118             kierowca.setNazwisko(kierowcaDto.getNazwisko());
00119             kierowca.setPesel(kierowcaDto.getPesel());
00120
00121             if (kierowcaDto.getUser() != null) {
00122                 User userIn = userService.findUserById(kierowcaDto.getUser().getId());
00123                 kierowca.setUser(userIn);
00124             }
00125
00126             kierowca = kierowcaRepository.save(kierowca);
00127             return new ResponseEntity<?>(new SimpleMessage("patched"), HttpStatus.OK);
00128         }
00129
00130     @GetMapping("/byPesel")
00131     public ResponseEntity<List<KierowcaDto>> findKierowcaByPesel(@RequestParam(name = "pesel", required
00132 = false) String pesel) {
00133         List<KierowcaDto> kierowcyDto;
00134
00135         if (pesel == null || pesel.isBlank() || pesel.equals("ALL")) {
00136             kierowcyDto = kierowcaRepository.findAll()
00137                 .stream()
00138                 .map(KierowcaDto::new)
00139                 .collect(Collectors.toList());
00140         } else {
00141             kierowcyDto = kierowcaRepository.findAllByPeselLike("%" + pesel + "%")
00142                 .stream()
00143                 .map(KierowcaDto::new)
00144                 .collect(Collectors.toList());
00145         }
00146
00147         return new ResponseEntity<?>(kierowcyDto, HttpStatus.OK);
00148     }
00149
00150
00151     @GetMapping("/byUser/")
00152     public ResponseEntity<KierowcaDto> findKierowcaByUser() {
00153         Kierowca kierowca =
00154             kierowcaService.findKierowcaByUser(Helpers.getUserFromContext(userService));
00155         if (kierowca != null) {
00156             KierowcaDto kierowcaDto = new KierowcaDto(kierowca);
00157             return ResponseEntity.ok(kierowcaDto);
00158         } else {
00159             return ResponseEntity.notFound().build();
00160         }
00161     }
00162
00163     @GetMapping("/byImieAndNazwisko")
00164     public ResponseEntity<List<KierowcaDto>> findKierowcaByImieAndNazwisko(@RequestParam(name = "imie")
00165     String imie, @RequestParam(name = "nazwisko") String nazwisko) {
00166         List<Kierowca> kierowcy = kierowcaService.findKierowcaByImieAndNazwisko(imie, nazwisko);
00167         List<KierowcaDto> kierowcyDto = kierowcy.stream()
00168             .map(KierowcaDto::new)
00169             .collect(Collectors.toList());
00170         return ResponseEntity.ok(kierowcyDto);
00171     }
00172
00173     @GetMapping("/byNazwisko")
00174     public ResponseEntity<List<KierowcaDto>> findKierowcaByNazwisko(@RequestParam(name = "nazwisko")
00175     String nazwisko) {
00176         List<Kierowca> kierowcy = kierowcaRepository.findByNazwiskoLike("%" + nazwisko + "%");
00177         List<KierowcaDto> kierowcyDto = kierowcy.stream()
00178             .map(KierowcaDto::new)
00179             .collect(Collectors.toList());
00180         return ResponseEntity.ok(kierowcyDto);
00181     }
00182
00183     @GetMapping("/byPrawoJazdyWazneDoBefore")
00184     public ResponseEntity<List<KierowcaDto>> findKierowcaByPrawoJazdyWazneDoBefore(@RequestParam(name =
00185     "date") Date date) {
00186         List<Kierowca> kierowcy = kierowcaService.findKierowcaByPrawoJazdyWazneDoBefore(date);
00187         List<KierowcaDto> kierowcyDto = kierowcy.stream()
00188             .map(KierowcaDto::new)
00189             .collect(Collectors.toList());
00190         return ResponseEntity.ok(kierowcyDto);
00191     }
00192
00193     @GetMapping("/byPrawoJazdyWazneDoAfter")
00194     public ResponseEntity<List<KierowcaDto>> findKierowcaByPrawoJazdyWazneDoAfter(@RequestParam(name =
00195     "date") Date date) {

```

```
00218     List<Kierowca> kierowcy = kierowcaService.findKierowcaByPrawoJazdyWazneDoAfter(date);  
00219     List<KierowcaDto> kierowcyDto = kierowcy.stream()  
00220         .map(KierowcaDto::new)  
00221         .collect(Collectors.toList());  
00222     return ResponseEntity.ok(kierowcyDto);  
00223 }  
00224  
00225 @GetMapping(path = {"/", ""})  
00226 public ResponseEntity<List<KierowcaDto>> findAll() {  
00227     List<Kierowca> kierowcy = kierowcaRepository.findAll();  
00228     List<KierowcaDto> kierowcyDto = kierowcy.stream()  
00229         .map(KierowcaDto::new)  
00230         .collect(Collectors.toList());  
00231     return ResponseEntity.ok(kierowcyDto);  
00232 }  
00233 }  
00234 }
```

7.13 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_ ← Backend/src/main/java/com/mklachl/sopkom/controller/KursController.java ←

Komponenty

- class [com.mklachl.sopkom.controller.KursController](#)

Pakiety

- package [com.mklachl.sopkom.controller](#)

7.14 KursController.java

Idź do dokumentacji tego pliku.

```
00001 package com.mklachl.sopkom.controller;  
00002  
00003 import java.sql.Time;  
00004 import java.text.SimpleDateFormat;  
00005 import java.time.LocalDate;  
00006 import java.util.ArrayList;  
00007 import java.util.Calendar;  
00008 import java.util.Date;  
00009 import java.util.List;  
00010 import java.util.Locale;  
00011 import java.util.Optional;  
00012 import java.util.TreeSet;  
00013  
00014 import org.springframework.beans.factory.annotation.Autowired;  
00015 import org.springframework.http.ResponseEntity;  
00016 import org.springframework.web.bind.annotation.DeleteMapping;  
00017 import org.springframework.web.bind.annotation.GetMapping;  
00018 import org.springframework.web.bind.annotation.PatchMapping;  
00019 import org.springframework.web.bind.annotation.PathVariable;  
00020 import org.springframework.web.bind.annotation.PostMapping;  
00021 import org.springframework.web.bind.annotation.RequestBody;  
00022 import org.springframework.web.bind.annotation.RequestMapping;  
00023 import org.springframework.web.bind.annotation.RequestParam;  
00024 import org.springframework.web.bind.annotation.RestController;  
00025  
00026 import com.fasterxml.jackson.databind.ObjectMapper;  
00027 import com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto;  
00028 import com.mklachl.sopkom.model.dto.kurs.InputKursDto;  
00029 import com.mklachl.sopkom.model.dto.kurs.InputPrzystanekWKursieDto;  
00030 import com.mklachl.sopkom.model.dto.kurs.KursDto;  
00031 import com.mklachl.sopkom.model.dto.linia.LiniaDto;  
00032 import com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLinia;  
00033 import com.mklachl.sopkom.model.entity.Kurs;
```

```
00034 import com.mklachl.sopkom.model.entity.Linia;
00035 import com.mklachl.sopkom.repository.KursRepository;
00036 import com.mklachl.sopkom.repository.LiniaRepository;
00037 import com.mklachl.sopkom.repository.PrzystanekWliniRepository;
00038 import com.mklachl.sopkom.services.KursService;
00039 import com.mklachl.sopkom.services.LiniaService;
00040
00041
00042 @RestController
00043 @RequestMapping(path = "/api/kurs")
00044 public class KursController {
00045
00046     public static List<KursDto> kursy = new ArrayList<KursDto>();
00047
00048     long currIndex = Long.valueOf(1);
00049     long currIndexPrzystanki = Long.valueOf(1);
00050
00051
00052     @Autowired
00053     public LiniaService liniaService;
00054
00055     @Autowired
00056     public LiniaRepository liniaRepository;
00057
00058     @Autowired
00059     public PrzystanekWliniRepository przystanekWliniRepository;
00060
00061     @Autowired
00062     public KursService kursService;
00063
00064     @Autowired
00065     public KursRepository kursRepo;
00066
00067     @GetMapping("/template/{id}")
00068     public ResponseEntity<?> getTemplateKurs(@PathVariable(name = "id") Long linia_id) {
00069
00070         KursDto dto = new KursDto();
00071
00072         Linia lIn = liniaRepository.findById(linia_id).orElseThrow();
00073
00074         dto.setHarmonogram(new HarmonogramDto());
00075
00076         dto.setLinia(new LiniaDto(lIn));
00077
00078         dto.setOdwracalny(lIn.odwracalna());
00079
00080         dto.setKierunek((short) 0);
00081
00082         dto.setKursId(1L);
00083
00084         List<InputPrzystanekWKursieDto> pk = new ArrayList<InputPrzystanekWKursieDto>();
00085
00086         int h = 12;
00087
00088         Time start = Time.valueOf(h + ":00:00");
00089         h++;
00090
00091         for(int i = 0; i < lIn.getPrzystankiWlini().size(); i++) {
00092
00093             InputPrzystanekWKursieDto pkwDto = new InputPrzystanekWKursieDto();
00094
00095             pkwDto.setGodzinna(start);
00096             start = Time.valueOf(h + ":00:00");
00097             h++;
00098
00099             pkwDto.setPrzystanekwKursieId((Long) (long)(i+1));
00100             pkwDto.setPrzystanekWLini(new PrzystanekDtoDlaLinia(lIn.getPrzystankiWlini().get(i)));
00101
00102             pk.add(pkwDto);
00103         }
00104
00105         dto.setPrzystanki(pk);
00106         dto.setTyp_autobusu((short) 0);
00107
00108
00109         return ResponseEntity.ok(dto);
00110     }
00111
00112     @GetMapping("/template/input/{id}")
00113     public ResponseEntity<?> getTemplateKursInput(@PathVariable(name = "id") Long linia_id) {
00114
00115         InputKursDto dto = new InputKursDto();
00116
00117         Linia lIn = liniaRepository.findById(linia_id).orElseThrow();
00118
00119         dto.setHarmonogramId((short) 1);
```

```
00135         dto.setLiniaId(linia_id);
00136
00137         dto.setKierunek((short)0);
00138
00139         dto.setKursId(11);
00140
00141     List<InputPrzystanekWKursieDto> pk = new ArrayList<InputPrzystanekWKursieDto>();
00142
00143     int h = 12;
00144
00145     Time start = Time.valueOf(h + ":00:00");
00146     h++;
00147
00148     for(int i = 0; i < lIn.getPrzystankiWlini().size(); i++) {
00149
00150         InputPrzystanekWKursieDto pkwDto = new InputPrzystanekWKursieDto();
00151
00152         pkwDto.setGodzinna(start);
00153         start = Time.valueOf(h + ":00:00");
00154         h++;
00155
00156         pkwDto.setPrzystanekwKursieId((Long) (long)(i+1));
00157         pkwDto.setPrzystanekWLini(new PrzystanekDtoDlaLinia(lIn.getPrzystankiWlini().get(i)));
00158
00159         pk.add(pkwDto);
00160     }
00161
00162     dto.setPrzystanki(pk);
00163     dto.setTyp_ autobusu((short)0);
00164
00165
00166     return ResponseEntity.ok(dto);
00167 }
00168
00169 @GetMapping(path = "", "/")
00170 public ResponseEntity<?> getAllKursy(@RequestParam(name="linia_id", defaultValue = "-1") Long
00171 linia_id) {
00172
00173     // if -1 then ret all
00174     System.out.println(linia_id);
00175     List<Kurs> kursy = null;
00176     var ret = new ArrayList<KursDto>();
00177     if(linia_id.equals(-1)) {
00178
00179         kursy = kursRepo.findAll();
00180
00181     } else {
00182         Optional<Linia> op = liniaService.findLiniaById(linia_id);
00183
00184         if(op.isEmpty()) {
00185             return ResponseEntity.notFound().build();
00186         }
00187
00188         kursy = kursService.findKursByLinia(op.get());
00189     }
00190
00191     for(var kurs : kursy) {
00192
00193         ret.add(new KursDto(kurs));
00194     }
00195
00196
00197     return ResponseEntity.ok(ret);
00198 }
00199
00200
00201
00202
00203
00204     return ResponseEntity.ok(ret);
00205 }
00206
00207
00208 @GetMapping("/{id}")
00209 public ResponseEntity<?> getKurs(@PathVariable(name="id") Long id) {
00210
00211     return ResponseEntity.ok(new KursDto(kursRepo.findById(id).get()));
00212 }
00213
00214 @PostMapping("/save")
00215 public ResponseEntity<?> addKurs(@RequestBody InputKursDto inputKursDto) {
00216
00217     // ah yes, legacy
00218     /*ObjectMapper mapper = new ObjectMapper();
00219
00220     Optional<Linia> liniaIn = liniaService.findLiniaById(inputKursDto.getLiniaId());
00221
00222     if(liniaIn.isEmpty()) {
00223         return ResponseEntity.notFound().build();
00224     }
00225
00226     HarmonogramDto dto =
00227     HarmonogramController.harmonograms.get(inputKursDto.getHarmonogramId()-1);
```

```

00235
00236
00237
00238     KursDto finale = new KursDto();
00239     finale.setHarmonogram(dto);
00240     finale.setLinia(new LiniaDto(liniaIn.get()));
00241     finale.setOdwracalny(liniaIn.get().odwracalna());
00242     finale.setKierunek(finale.getOdwracalny() ? inputKursDto.getKierunek() : 0);
00243     finale.setKursId(Long.valueOf(currIndex));
00244     currIndex+= 1;
00245     finale.setTyp_autobusu(inputKursDto.getTyp_autobusu());
00246
00247     var przystanki = new ArrayList<InputPrzystanekWKursieDto>();
00248
00249     for(int i = 0; i < inputKursDto.getPrzystanki().size(); i++) {
00250         var in = inputKursDto.getPrzystanki().get(i);
00251         InputPrzystanekWKursieDto out = new InputPrzystanekWKursieDto();
00252
00253         out.setGodzinna(in.getGodzinna());
00254         out.setPrzystanekWKursieId(Long.valueOf(currIndexPrzystanki));
00255         currIndexPrzystanki++;
00256
00257         var pwl =
00258             przystnekWLiniRepository.findById(in.getPrzystanekWLini().getPrzystanekwliniId());
00259
00260         if(pwl.isEmpty()) {
00261             return ResponseEntity.notFound().build();
00262         }
00263
00264         out.setPrzystanekWLini(new PrzystanekDtoDlaLinia(pwl.get()));
00265
00266         przystanki.add(out);
00267     }
00268     finale.setPrzystanki(przystanki);
00269
00270     if(kursy.add(finale)) {
00271         return ResponseEntity.ok().body(inputKursDto);
00272     }*/
00273
00274     try {
00275         kursService.saveKurs(inputKursDto);
00276
00277         return ResponseEntity.ok().build();
00278     } catch (Exception e) {
00279         e.printStackTrace();
00280     }
00281
00282     return ResponseEntity.badRequest().build();
00283 }
00284
00285 @PatchMapping("/{id}")
00286 public ResponseEntity<?> updateKurs(@PathVariable(name="id") Long id, @RequestBody InputKursDto
00287 inputKursDto) {
00288
00289     /*
00290     Optional<Linia> liniaIn = liniaService.findLiniaById(inputKursDto.getLiniaId());
00291
00292     if(liniaIn.isEmpty()) {
00293         return ResponseEntity.notFound().build();
00294     }
00295
00296     HarmonogramDto dto =
00297 HarmonogramController.harmonograms.get(inputKursDto.getHarmonogramId()-1);
00298
00299     KursDto finale = getById(id);
00300
00301     if(finale == null) {
00302         return ResponseEntity.noContent().build();
00303     }
00304
00305     finale.setHarmonogram(dto);
00306     finale.setLinia(new LiniaDto(liniaIn.get()));
00307     finale.setOdwracalny(liniaIn.get().odwracalna());
00308     finale.setKierunek(finale.getOdwracalny() ? inputKursDto.getKierunek() : 0);
00309     finale.setTyp_autobusu(inputKursDto.getTyp_autobusu());
00310
00311     var przystanki = new ArrayList<InputPrzystanekWKursieDto>();
00312
00313     finale.setPrzystanki(przystanki);
00314
00315     for(int i = 0; i < inputKursDto.getPrzystanki().size(); i++) {
00316         var in = inputKursDto.getPrzystanki().get(i);
00317         InputPrzystanekWKursieDto out = new InputPrzystanekWKursieDto();
00318
00319         out.setGodzinna(in.getGodzinna());
00320         out.setPrzystanekWKursieId(Long.valueOf(currIndexPrzystanki));
00321         currIndexPrzystanki++;
00322
00323         out.setLinia(new LiniaDto(liniaIn.get()));
00324         out.setOdwracalny(liniaIn.get().odwracalna());
00325         out.setKierunek(finale.getOdwracalny() ? inputKursDto.getKierunek() : 0);
00326         out.setTyp_autobusu(inputKursDto.getTyp_autobusu());
00327
00328         przystanki.add(out);
00329     }
00330
00331     finale.setPrzystanki(przystanki);
00332
00333     try {
00334         kursService.saveKurs(inputKursDto);
00335
00336         return ResponseEntity.ok().body(finale);
00337     } catch (Exception e) {
00338         e.printStackTrace();
00339     }
00340
00341     return ResponseEntity.badRequest().build();
00342 }
```

```
00325         out.setGodzinna(in.getGodzinna());
00326         out.setPrzystanekwKursieId(currIndexPrzystanki++);
00327
00328         var pwl =
00329             przystnekWliniRepository.findById(in.getPrzystanekWLini().getPrzystanekwliniId());
00330
00331         if(pwl.isEmpty()) {
00332             return ResponseEntity.notFound().build();
00333         }
00334
00335         out.setPrzystanekWLini(new PrzystanekDtoDlaLinia(pwl.get()));
00336
00337         przystanki.add(out);
00338     }
00339     finale.setPrzystanki(przystanki);
00340
00341     deleteById(finale.getKursId());
00342
00343     if(kursy.add(finale)) {
00344         return ResponseEntity.ok().body(inputKursDto);
00345     }
00346     8*/
00347
00348     try {
00349         kursService.updateKurs(inputKursDto);
00350     } catch (Exception e) {
00351         ResponseEntity.notFound();
00352         e.printStackTrace();
00353     }
00354
00355
00356     return ResponseEntity.ok().body(inputKursDto);
00357 }
00363 @DeleteMapping("/{id}")
00364 public ResponseEntity<?> deleteKurs(@PathVariable(name="id") Long id) {
00365
00366     kursRepo.deleteById(id);
00367
00368     return ResponseEntity.ok().build();
00369 }
00370
00371
00372 }
```

7.15 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_ ← Backend/src/main/java/com/mklachl/sopkom/controller/LiniaController.java ←

Komponenty

- class [com.mklachl.sopkom.controller.LiniaController](#)

Pakiety

- package [com.mklachl.sopkom.controller](#)

7.16 LiniaController.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.controller;
00002
00003 import java.util.ArrayList;
00004 import java.util.List;
00005 import java.util.Optional;
```

```

00006
00007 import org.springframework.beans.factory.annotation.Autowired;
00008 import org.springframework.http.HttpStatus;
00009 import org.springframework.http.MediaType;
00010 import org.springframework.http.ResponseEntity;
00011 import org.springframework.web.bind.annotation.*;
00012
00013 import com.mklachl.sopkom.exceptions.LiniaNotFoundException;
00014 import com.mklachl.sopkom.exceptions.PrzystanekNotFoundException;
00015 import com.mklachl.sopkom.model.dto.linia.LiniaDtoInput;
00016 import com.mklachl.sopkom.model.dto.linia.LiniaDtoOutput;
00017 import com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLinia;
00018 import com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLiniaInput;
00019 import com.mklachl.sopkom.model.entity.Linia;
00020 import com.mklachl.sopkom.repository.LiniaRepository;
00021 import com.mklachl.sopkom.services.LiniaService;
00022
00023 @RestController
00024 @RequestMapping("/api/linia")
00025 public class LiniaController {
00026     @Autowired
00027     LiniaService liniaService;
00028
00029     @Autowired
00030     LiniaRepository liniaRepository;
00031
00032     @GetMapping(path = "template", produces = {MediaType.APPLICATION_JSON_VALUE})
00033     public ResponseEntity<?> template() {
00034         LiniaDtoOutput liniaDto = new LiniaDtoOutput();
00035
00036         liniaDto.setNumer("M500");
00037         liniaDto.setId(Long.valueOf(1));
00038
00039         List<PrzystanekDtoDlaLinia> list = new ArrayList<PrzystanekDtoDlaLinia>();
00040
00041         for(Short i = 0; i < 5; i++) {
00042             PrzystanekDtoDlaLinia przystanek = new PrzystanekDtoDlaLinia();
00043             przystanek.setKodPocztowy("44-100");
00044             przystanek.setDlugoscGeograficzna((22.123 + i) + "");
00045             przystanek.setSzerokoscGeograficzna((50.000 - i) + "");
00046             przystanek.setKolejnosc(i);
00047             przystanek.setMiasto("Gliwce");
00048             przystanek.setUlica("Ulica testowa " + i);
00049             przystanek.setPrzystanekId(Long.valueOf(i));
00050
00051             list.add(przystanek);
00052
00053         }
00054
00055         liniaDto.setPrzystanki(list);
00056
00057         return new ResponseEntity<>(liniaDto, HttpStatus.OK);
00058     }
00059
00060     @GetMapping(path = "templateInput", produces = {MediaType.APPLICATION_JSON_VALUE})
00061     public ResponseEntity<?> templateInput() {
00062         LiniaDtoInput liniaDto = new LiniaDtoInput();
00063
00064         liniaDto.setNumer("M500");
00065         liniaDto.setId(Long.valueOf(1));
00066
00067         List<PrzystanekDtoDlaLiniaInput> list = new ArrayList<PrzystanekDtoDlaLiniaInput>();
00068
00069         for(Short i = 0; i < 5; i++) {
00070             PrzystanekDtoDlaLiniaInput przystanek = new PrzystanekDtoDlaLiniaInput();
00071
00072             przystanek.setKolejnosc(i);
00073             przystanek.setPrzystanekId(Long.valueOf(i));
00074
00075             list.add(przystanek);
00076
00077         }
00078
00079         liniaDto.setPrzystanki(list);
00080
00081         return new ResponseEntity<>(liniaDto, HttpStatus.OK);
00082     }
00083
00084     @PostMapping("/save")
00085     public ResponseEntity<?> addLinia(@RequestBody LiniaDtoInput liniaDto){
00086
00087         Linia linia;
00088         try {
00089             linia = liniaService.saveLinia(liniaDto);
00090
00091         }
00092
00093     }
00094
00095 }
```

```
00106     } catch (PrzystanekNotFoundException e) {
00107         return new ResponseEntity<>(e, HttpStatus.NOT_FOUND);
00108     }
00109 }
00110 return new ResponseEntity<>(new LiniaDtoOutput(linia), HttpStatus.OK);
00111 }
00112
00113 @GetMapping("/{id}")
00114 public ResponseEntity<?> getLinia(@PathVariable("id") Long id){
00115     Optional<Linia> linia = liniaRepository.findById(id);
00116
00117     if(linia.isEmpty()) {
00118         return new ResponseEntity<>(HttpStatus.NOT_FOUND);
00119     }
00120
00121     return new ResponseEntity<>(new LiniaDtoOutput(linia.get()), HttpStatus.OK);
00122 }
00123
00124 @DeleteMapping("/{id}")
00125 public ResponseEntity<?> deleteLinia(@PathVariable("id") Long id) {
00126     if(liniaRepository.findById(id).isEmpty()) {
00127         return new ResponseEntity<>(HttpStatus.NO_CONTENT);
00128     }
00129     liniaRepository.deleteById(id);
00130     return new ResponseEntity<>(HttpStatus.OK);
00131 }
00132
00133 @PatchMapping("/{id}")
00134 public ResponseEntity<?> updateLinia(@PathVariable("id") Long id, @RequestBody LiniaDtoInput
liniaDtoInput) {
00135
00136     Linia updated;
00137     liniaDtoInput.setId(id);
00138     try {
00139         updated = liniaService.patchLinia(liniaDtoInput);
00140     } catch (PrzystanekNotFoundException | LiniaNotFoundException e) {
00141
00142         return new ResponseEntity<>(e, HttpStatus.NOT_FOUND);
00143     }
00144
00145     return new ResponseEntity<>(new LiniaDtoOutput(updated), HttpStatus.OK);
00146 }
00147
00148 @GetMapping
00149 public ResponseEntity<?> getAllPrzystanki() {
00150     List<LiniaDtoOutput> list = new ArrayList<>();
00151
00152     liniaRepository.findAll().forEach(linia -> {
00153         list.add(new LiniaDtoOutput(linia));
00154     });
00155     return new ResponseEntity<>(list, HttpStatus.OK);
00156 }
00157
00158 @GetMapping("/input")
00159 public ResponseEntity<?> getAllPrzystankiFull() {
00160     List<LiniaDtoInput> list = new ArrayList<>();
00161
00162     liniaRepository.findAll().forEach(linia -> {
00163         list.add(new LiniaDtoInput(linia));
00164     });
00165     return new ResponseEntity<>(list, HttpStatus.OK);
00166 }
00167
00168 @GetMapping("/input/{id}")
00169 public ResponseEntity<?> getPrzystanekFull(@PathVariable("id") Long id){
00170     Optional<Linia> linia = liniaRepository.findById(id);
00171
00172     if(linia.isEmpty()) {
00173         return new ResponseEntity<>(HttpStatus.NOT_FOUND);
00174     }
00175
00176     return new ResponseEntity<>(new LiniaDtoInput(linia.get()), HttpStatus.OK);
00177 }
00178
00179 }
```

7.17 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ ← Backend/src/main/java/com/mklachl/sopkom/controller/Login ← Controller.java

Komponenty

- class [com.mklachl.sopkom.controller.LoginController](#)

Pakiety

- package [com.mklachl.sopkom.controller](#)

7.18 LoginController.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.controller;
00002
00003 import org.springframework.stereotype.Controller;
00004 import org.springframework.web.bind.annotation.GetMapping;
00005
00006 @Controller
00007 public class LoginController {
00008
00013     @GetMapping("/login")
00014     public String getLoginPage(){
00015         return "login";
00016     }
00017
00022     @GetMapping("/home")
00023     public String getLogoutPage(){
00024         return "home";
00025     }
00026 }
```

7.19 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ ← Backend/src/main/java/com/mklachl/sopkom/controller/Przejazd ← Controller.java

Komponenty

- class [com.mklachl.sopkom.controller.PrzejazdController](#)

Pakiety

- package [com.mklachl.sopkom.controller](#)

7.20 PrzejazdController.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.controller;
00002
00003 import java.text.SimpleDateFormat;
00004 import java.util.ArrayList;
00005 import java.util.Calendar;
00006 import java.util.Comparator;
00007 import java.util.Date;
00008 import java.util.LinkedList;
00009 import java.util.List;
00010 import java.util.Optional;
00011 import java.util.stream.Collectors;
00012
00013 import org.springframework.beans.factory.annotation.Autowired;
00014 import org.springframework.format.annotation.DateTimeFormat;
00015 import org.springframework.http.ResponseEntity;
00016 import org.springframework.web.bind.annotation.*;
00017
00018 import com.mklachl.sopkom.model.dto.AutobusDto;
00019 import com.mklachl.sopkom.model.dto.KierowcaDto;
00020 import com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoInput;
00021 import com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput;
00022 import com.mklachl.sopkom.model.entity.Harmonogram;
00023 import com.mklachl.sopkom.model.entity.Kurs;
00024 import com.mklachl.sopkom.model.entity.Przejazd;
00025 import com.mklachl.sopkom.raporty.rozkład.DateHelper;
00026 import com.mklachl.sopkom.repository.AutobusRepository;
00027 import com.mklachl.sopkom.repository.HarmonogramRepository;
00028 import com.mklachl.sopkom.repository.KierowcaRepository;
00029 import com.mklachl.sopkom.repository.KursRepository;
00030 import com.mklachl.sopkom.repository.PrzejazdRepository;
00031 import com.mklachl.sopkom.services.KursService;
00032 import com.mklachl.sopkom.services.PrzejazdService;
00033
00034 @RestController
00035 @RequestMapping("/api/przejazd")
00036 public class PrzejazdController {
00037
00038     @Autowired
00039     public HarmonogramRepository harmonogramRepository;
00040
00041     @Autowired
00042     public KursRepository kursRepository;
00043
00044     @Autowired
00045     public KursService kursService;
00046
00047     @Autowired
00048     public PrzejazdService przejazdService;
00049
00050     @Autowired
00051     public KierowcaRepository kierowcaRepository;
00052
00053     @Autowired
00054     public AutobusRepository autobusRepository;
00055
00056     @Autowired
00057     public PrzejazdRepository przejazdRepository;
00058
00059     @PostMapping({"/save", "/save/"})
00060     public ResponseEntity<?> savePrzejazd(@RequestBody PrzejazdDtoInput input) {
00061
00062         System.out.println(input);
00063
00064         przejazdService.savePrzejazd(input);
00065
00066         return ResponseEntity.ok().build();
00067     }
00068
00069     @GetMapping("/{przejazd_id}", "{przejazd_id}/")
00070     public ResponseEntity<?> getPrzejazd(@PathVariable(name="przejazd_id") Long przejazdid) {
00071
00072         try {
00073             var p = przejazdService.findPrzejazdById(przejazdid).orElseThrow();
00074
00075             PrzejazdDtoOutput output = new PrzejazdDtoOutput(p);
00076
00077             return ResponseEntity.ok(output);
00078         } catch (Exception e) {
00079             return ResponseEntity.notFound().build();
00080         }
00081     }
00082 }
```

```

00093
00094     }
00095
00101     @DeleteMapping("/{przejazd_id}", "/{przejazd_id}/")
00102     public ResponseEntity<?> deletePrzejazd(@PathVariable(name="przejazd_id") Long przejazdid){
00103
00104         try {
00105             przejazdRepository.deleteById(przejazdid);
00106
00107             return ResponseEntity.ok("deleted");
00108
00109         } catch (Exception e) {
00110
00111             return ResponseEntity.notFound().build();
00112         }
00113
00114     }
00115
00121     @GetMapping()
00122     public ResponseEntity<?> getPrzejazdy(@RequestParam(name="kurs_id", required = false) Long
00123     kursId){
00124         try {
00125             List<Przejazd> przejazdy = null;
00126
00127             if(kursId == null) {
00128                 przejazdy = przejazdRepository.findAll();
00129             } else {
00130
00131                 przejazdy =
00132                     przejazdRepository.findAllByKurs(kursRepository.findById(kursId).orElseThrow());
00133             }
00134
00135         } catch (Exception e) {
00136
00137             return ResponseEntity.notFound().build();
00138         }
00139
00140     }
00141
00148     @GetMapping("/allDrivers")
00149     public ResponseEntity<?> getAllAvailableDriversForKurs(@RequestParam(name="kurs_id") Long kursId ,
00150     @RequestParam(name="data") @DateTimeFormat(pattern = "dd.MM.yyyy") Date date){
00151
00152         System.out.println(date);
00153
00154         Kurs kurs;
00155         try {
00156             kurs = kursRepository.findById(kursId).orElseThrow();
00157
00158         } catch (Exception e) {
00159             return ResponseEntity.notFound().build();
00160         }
00161
00162         Date startRaw;
00163         Date endRaw;
00164
00165         if(kurs.getKierunek().shortValue() == 0) {
00166
00167             startRaw = DateHelper.copyTimeComponents(
00168                 kurs.getKursPrzystanekWlinii().get(0).getGodzinna(),
00169                 date);
00170
00171             endRaw = DateHelper.copyTimeComponents(
00172                 kurs.getKursPrzystanekWlinii().get(kurs.getKursPrzystanekWlinii().size()-1).getGodzinna(),
00173                 date);
00174
00175         } else {
00176             startRaw = DateHelper.copyTimeComponents(
00177
00178                 kurs.getKursPrzystanekWlinii().get(kurs.getKursPrzystanekWlinii().size()-1).getGodzinna(),
00179                 date);
00180
00181             endRaw = DateHelper.copyTimeComponents(
00182                 kurs.getKursPrzystanekWlinii().get(0).getGodzinna(),
00183                 date);
00184
00185         }
00186
00187         var temp = kierowcaRepository.findAllByAvailability(startRaw,
00188         endRaw).stream().map(KierowcaDto::new).collect(Collectors.toList());
00189
00190         return ResponseEntity.ok(temp);

```

```
00189     }
00190
00197     @GetMapping("/allBuses")
00198     public ResponseEntity<?> getAllAvailableBusesForKurs(@RequestParam(name="kurs_id") Long kursId ,
00199         @RequestParam(name="data") @DateTimeFormat(pattern = "dd.MM.yyyy") Date date) {
00200         System.out.println(date);
00201
00202         Kurs kurs;
00203         try {
00204             kurs = kursRepository.findById(kursId).orElseThrow();
00205
00206         } catch (Exception e) {
00207             return ResponseEntity.notFound().build();
00208         }
00209
00210         Date startRaw;
00211         Date endRaw;
00212
00213         if(kurs.getKierunek().shortValue() == 0) {
00214
00215             startRaw = DateHelper.copyTimeComponents(
00216                 kurs.getKursPrzystanekWlinii().get(0).getGodzinna(),
00217                 date);
00218
00219             endRaw = DateHelper.copyTimeComponents(
00220
00221             kurs.getKursPrzystanekWlinii().get(kurs.getKursPrzystanekWlinii().size()-1).getGodzinna(),
00222
00223
00224         } else {
00225             startRaw = DateHelper.copyTimeComponents(
00226
00227             kurs.getKursPrzystanekWlinii().get(kurs.getKursPrzystanekWlinii().size()-1).getGodzinna(),
00228
00229             endRaw = DateHelper.copyTimeComponents(
00230                 kurs.getKursPrzystanekWlinii().get(0).getGodzinna(),
00231                 date);
00232
00233         }
00234
00235         var temp = autobusRepository.findAllByAvailable(startRaw, endRaw).stream().filter((item) -> {
00236             return item.getStatus().equals("OK"));
00237         }).map(AutobusDto::new).collect(Collectors.toList());
00238
00239         return ResponseEntity.ok(temp);
00240     }
00241
00242     @GetMapping("/{przyszle}", "/przyszle/")
00243     public ResponseEntity<?> getPrzyszlePrzejazdy() {
00244
00245         List<PrzejazdDtoOutput> przyszlePrzejazdy = new ArrayList<PrzejazdDtoOutput>();
00246
00247         for(Kurs kurs : kursRepository.findAll()) {
00248
00249             List<Date> datesForKurs = przejazdService.findAllDatesForKurs(kurs, 30);
00250
00251             Date godzinna_startu = kurs.getKierunek() == 0 ?
00252                 kurs.getKursPrzystanekWlinii().get(0).getGodzinna() :
00253                 kurs.getKursPrzystanekWlinii().get(0).getGodzinna();
00254
00255             List<PrzejazdDtoOutput> a = datesForKurs.stream()
00256                 .filter((item) -> {
00257                     Calendar c = Calendar.getInstance();
00258                     c.setTime(godzinna_startu);
00259
00260                     Date startWithFullDate = DateHelper.copyTimeComponents(godzinna_startu, item);
00261                     System.out.println(new Date());
00262                     System.out.println(startWithFullDate);
00263
00264                     return startWithFullDate.after(new Date());
00265
00266                 })
00267                 .map((date) -> new PrzejazdDtoOutput(przejazdService.createDetachedPrzejazd(kurs,
00268
00269             date), true)).collect(Collectors.toList());
00270
00271             przyszlePrzejazdy.addAll(a);
00272
00273         }
00274
00275         przyszlePrzejazdy.sort(new Comparator<PrzejazdDtoOutput>() {
00276
00277             @Override
00278             public int compare(PrzejazdDtoOutput o1, PrzejazdDtoOutput o2) {
00279
00280                 return o1.getData().compareTo(o2.getData());
00281             }
00282         });
00283     }
00284 }
```

```

00281         }
00282     });
00283 
00284     return ResponseEntity.ok(przyszlePrzejazdy);
00285 }
00286 }
00287 
00288 @GetMapping("/allDates")
00289 public ResponseEntity<?> getAllAvailableDatesForKurs(@RequestParam(name="kurs_id") Long kursId) {
00290     Kurs kurs;
00291     try {
00292         kurs = kursRepository.findById(kursId).orElseThrow();
00293     } catch (Exception e) {
00294         return ResponseEntity.notFound().build();
00295     }
00296 
00297     Harmonogram harmongram = kurs.getHarmonogram();
00298 
00299     var result = DateHelper.getAllDatesForHarmonogram(harmongram);
00300     var copy = new LinkedList<Date>(result);
00301 
00302     var data = przejazdService.findPrzejazdByKurs(kurs);
00303 
00304     System.out.println(data.size());
00305     SimpleDateFormat format = new SimpleDateFormat("dd.MM.yyyy");
00306 
00307     List<String> dates = data.stream()
00308         .map(item -> format.format(item.getDataStartu()))
00309         .collect(Collectors.toList());
00310 
00311     List<String> list = copy.stream().map(item ->
00312         format.format(item)).collect(Collectors.toList());
00313 
00314     list.removeAll(dates);
00315 
00316     return ResponseEntity.ok(list);
00317 }
00318 }
00319 }
00320 }
00321 }
00322 }
00323 }
00324 }

```

7.21 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/controller/← PrzystanekController.java

Komponenty

- class [com.mklachl.sopkom.controller.PrzystanekController](#)

Pakietы

- package [com.mklachl.sopkom.controller](#)

7.22 PrzystanekController.java

[Idź do dokumentacji tego pliku.](#)

```

00001 package com.mklachl.sopkom.controller;
00002 
00003 import com.mklachl.sopkom.model.dto.PrzystanekDto;
00004 import com.mklachl.sopkom.model.dto.PrzystanekDtoFull;
00005 import com.mklachl.sopkom.model.entity.Przystanek;
00006 import com.mklachl.sopkom.repository.PrzystanekRepository;
00007 import com.mklachl.sopkom.services.PrzystanekService;
00008 import org.springframework.beans.factory.annotation.Autowired;
00009 import org.springframework.http.HttpStatus;

```

```
00010 import org.springframework.http.MediaType;
00011 import org.springframework.http.ResponseEntity;
00012 import org.springframework.web.bind.annotation.*;
00013
00014 import java.util.ArrayList;
00015 import java.util.List;
00016 import java.util.Optional;
00017
00018 @RestController
00019 @RequestMapping("/api/przystanek")
00020 public class PrzystanekController {
00021
00022     @Autowired
00023     PrzystanekService przystanekService;
00024
00025     @Autowired
00026     PrzystanekRepository przystanekRepo;
00027
00028     @GetMapping(path = "template", produces = {MediaType.APPLICATION_JSON_VALUE})
00029     public ResponseEntity<?> template() {
00030         PrzystanekDto dto = new PrzystanekDto();
00031
00032         dto.setPrzystanekId(Long.valueOf(1));
00033         dto.setNazwa("Centralny");
00034         dto.setKodPocztowy("00-950");
00035         dto.setMiasto("Warszawa");
00036         dto.setUlica("Jana Pawła II");
00037         dto.setDlugoscGeograficzna("21.0122");
00038         dto.setSzerokoscGeograficzna("52.2297");
00039
00040         dto.setPrzystanekOdwrzony(Long.valueOf(12));
00041         dto.setPrzystanekOdwrzonyNazwa("Przystanek odwrzony test (nie wymagane)");
00042
00043         // Zwrócenie przykładowego DTO w odpowiedzi HTTP
00044         return new ResponseEntity<>(dto, HttpStatus.OK);
00045     }
00046
00047     @PostMapping("/save")
00048     public ResponseEntity<?> addPrzystanek(@RequestBody PrzystanekDto przystanekDto) {
00049         Przystanek przystanek = przystanekService.savePrzystanek(przystanekDto);
00050         return new ResponseEntity<?>(new PrzystanekDto(przystanek), HttpStatus.OK);
00051     }
00052
00053     @GetMapping("/{id}")
00054     public ResponseEntity<?> getPrzystanek(@PathVariable("id") Long id) {
00055         Optional<Przystanek> przystanek = przystanekService.findPrzystanekById(id);
00056         if(przystanek.isEmpty()) {
00057             return new ResponseEntity<?>(HttpStatus.NOT_FOUND);
00058         }
00059         return new ResponseEntity<?>(new PrzystanekDto(przystanek.get()), HttpStatus.OK);
00060     }
00061
00062     @DeleteMapping("/{id}")
00063     public ResponseEntity<?> deletePrzystanek(@PathVariable("id") Long id) {
00064         if(przystanekRepo.findById(id).isEmpty()) {
00065             return new ResponseEntity<?>(HttpStatus.NO_CONTENT);
00066         }
00067
00068         try {
00069             przystanekRepo.deleteById(id);
00070             return new ResponseEntity<?>(HttpStatus.OK);
00071
00072         } catch (Exception e) {
00073             return new ResponseEntity<?>(new SimpleMessage(e.getMessage()), HttpStatus.FORBIDDEN);
00074         }
00075     }
00076
00077     @PatchMapping("/{id}")
00078     public ResponseEntity<?> updatePrzystanek(@PathVariable("id") Long id, @RequestBody PrzystanekDto
00079     przystanekDto) {
00080         Optional<Przystanek> currentPrzystanek = przystanekRepo.findById(id);
00081         if(currentPrzystanek.isEmpty()) {
00082             return new ResponseEntity<?>(HttpStatus.NOT_FOUND);
00083         }
00084
00085         Przystanek przystanek = currentPrzystanek.get();
00086
00087         przystanek.setNazwa(przystanekDto.getNazwa());
00088         przystanek.setKodPocztowy(przystanekDto.getKodPocztowy());
00089         przystanek.setMiasto(przystanekDto.getMiasto());
00090         przystanek.setUlica(przystanekDto.getUlica());
00091         przystanek.setDlugoscGeograficzna(przystanekDto.getDlugoscGeograficzna());
00092         przystanek.setSzerokoscGeograficzna(przystanekDto.getSzerokoscGeograficzna());
00093
00094         if(przystanekDto.getPrzystanekOdwrzony() != null) {
00095             if(przystanekDto.getPrzystanekOdwrzony().longValue() == -1) {
```

```

00121         przystanek.setPrzystanekOdwrotny(null);
00122     } else {
00123         var przystanekOdwrotny =
00124             przystanekRepo.findById(przystanekDto.getPrzystanekOdwrotny());
00125         if(przystanekOdwrotny.isPresent()) {
00126             przystanek.setPrzystanekOdwrotny(przystanekOdwrotny.get());
00127         } else {
00128             System.out.println("no przystanek odwrotny present!");
00129         }
00130     }
00131
00132     przystanek = przystanekRepo.save(przystanek);
00133     return new ResponseEntity<>(new PrzystanekDto(przystanek), HttpStatus.OK);
00134 }
00135
00140 @GetMapping
00141 public ResponseEntity<?> getAllPrzystanki() {
00142     List<PrzystanekDto> list = new ArrayList<>();
00143     przystanekRepo.findAll().forEach(przystanek -> {
00144         list.add(new PrzystanekDto(przystanek));
00145     });
00146     return new ResponseEntity<?>(list, HttpStatus.OK);
00147 }
00148
00153 @GetMapping("/full")
00154 public ResponseEntity<?> getAllPrzystankiFull() {
00155     List<PrzystanekDtoFull> list = new ArrayList<>();
00156     przystanekRepo.findAll().forEach(przystanek -> {
00157         list.add(new PrzystanekDtoFull(przystanek));
00158     });
00159     return new ResponseEntity<?>(list, HttpStatus.OK);
00160 }
00161
00167 @GetMapping("/full/{id}")
00168 public ResponseEntity<?> getPrzystanekFull(@PathVariable("id") Long id){
00169     Optional<Przystanek> przystanek = przystanekService.findPrzystanekById(id);
00170     if(przystanek.isEmpty()) {
00171         return new ResponseEntity<?>(HttpStatus.NOT_FOUND);
00172     }
00173     return new ResponseEntity<?>(new PrzystanekDtoFull(przystanek.get()), HttpStatus.OK);
00174 }
00175 }

```

7.23 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/controller/Raporty← Controller.java

Komponenty

- class [com.mklachl.sopkom.controller.RaportyController](#)

Pakiety

- package [com.mklachl.sopkom.controller](#)

7.24 RaportyController.java

[Idź do dokumentacji tego pliku.](#)

```

00001 package com.mklachl.sopkom.controller;
00002
00003 import java.io.File;
00004 import java.io.FileNotFoundException;
00005 import java.io.IOException;
00006

```

```
00007 import org.springframework.beans.factory.annotation.Autowired;
00008 import org.springframework.core.io.FileSystemResource;
00009 import org.springframework.http.MediaType;
00010 import org.springframework.http.ResponseEntity;
00011 import org.springframework.web.bind.annotation.RequestMapping;
00012 import org.springframework.web.bind.annotation.RequestMethod;
00013 import org.springframework.web.bind.annotation.RequestParam;
00014 import org.springframework.web.bind.annotation.ResponseBody;
00015 import org.springframework.web.bind.annotation.RestController;
00016
00017 import com.mklachl.sopkom.topdf.Pdf;
00018
00019 @RestController
00020 @RequestMapping("/api/raporty")
00021 public class RaportyController {
00022
00023     @Autowired
00024     Pdf pdfService;
00025
00033     @RequestMapping(value = "/rozklad", method = RequestMethod.GET)
00034     @ResponseBody
00035     public ResponseEntity<FileSystemResource> generateRozklad(@RequestParam(name="id") Long id) throws
00036         FileNotFoundException, IOException {
00037         File fileIn;
00038         try {
00039             fileIn = pdfService.genRozklad(id);
00040             System.out.println(fileIn.getAbsolutePath());
00041             return ResponseEntity.ok()
00042                 .contentLength(fileIn.length())
00043                 .contentType(MediaType.parseMediaType(MediaType.APPLICATION_PDF_VALUE))
00044                 .body(new FileSystemResource(fileIn.getAbsolutePath()));
00044     } catch (Exception e) {
00045         e.printStackTrace();
00046     }
00047     return ResponseEntity.notFound().build();
00048 }
00049 }
```

7.25 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_↔ Backend/src/main/java/com/mklachl/sopkom/controller/Rozklad↔ Controller.java

Komponenty

- class [com.mklachl.sopkom.controller.RozkladController](#)

Pakiety

- package [com.mklachl.sopkom.controller](#)

7.26 RozkladController.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.controller;
00002
00003 import java.util.Date;
00004
00005 import org.springframework.beans.factory.annotation.Autowired;
00006 import org.springframework.format.annotation.DateTimeFormat;
00007 import org.springframework.http.HttpStatus;
00008 import org.springframework.http.ResponseEntity;
00009 import org.springframework.web.bind.annotation.GetMapping;
00010 import org.springframework.web.bind.annotation.RequestMapping;
00011 import org.springframework.web.bind.annotation.RequestParam;
00012 import org.springframework.web.bind.annotation.RestController;
```

```

00013
00014 import com.mklachl.sopkom.model.dto.PrzystanekDto;
00015 import com.mklachl.sopkom.model.dto.rozklad.RozkladDto;
00016 import com.mklachl.sopkom.model.entity.Przystanek;
00017 import com.mklachl.sopkom.raporty.rozklad.DateHelper;
00018 import com.mklachl.sopkom.repository.PrzystanekRepository;
00019
00020 @RestController
00021 @RequestMapping("/api/rozklad")
00022 public class RozkladController {
00023
00024     @Autowired
00025     public PrzystanekRepository repo;
00026
00027     @Autowired
00028     public DateHelper rozkladHelper;
00029
00030     @GetMapping()
00031     public ResponseEntity<?> test(@RequestParam(name="id") Long id, @RequestParam(name="date")
00032         @DateTimeFormat(pattern = "yyyy-MM-dd") Date date){
00033
00034         Przystanek normalny = repo.findById(id).get();
00035
00036         RozkladDto dto = new RozkladDto();
00037         dto.setNormalny(new PrzystanekDto(normalny));
00038         dto.setDate(date);
00039
00040         var vat = rozkladHelper.getAllByPrzystanekId(id, date);
00041
00042         for(var $$$ : vat) {
00043             dto.addKurs($$$.getLinia(), $$$.getT(), $$$.isOdw());
00044         }
00045
00046         return new ResponseEntity<>(dto, HttpStatus.OK);
00047     }
00048 }
00049 }
```

7.27 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/controller/Simple← Message.java

Komponenty

- class [com.mklachl.sopkom.controller.SimpleMessage](#)

Pakiety

- package [com.mklachl.sopkom.controller](#)

7.28 SimpleMessage.java

[Idź do dokumentacji tego pliku.](#)

```

00001 package com.mklachl.sopkom.controller;
00002
00006 public class SimpleMessage {
00007
00009     public String message;
00010
00012     public SimpleMessage() {
00013         super();
00014     }
00015
00020     public SimpleMessage(String message) {
00021         super();
00022         this.message = message;
```

```
00023     }
00024
00025     public String getMessage() {
00026         return message;
00027     }
00028
00029     public void setMessage(String message) {
00030         this.message = message;
00031     }
00032
00033 }
```

7.29 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_ ← Backend/src/main/java/com/mklachl/sopkom/controller/UserController.java ←

Komponenty

- class [com.mklachl.sopkom.controller.UserController](#)

Pakiety

- package [com.mklachl.sopkom.controller](#)

7.30 UserController.java

Idź do dokumentacji tego pliku.

```
00001 package com.mklachl.sopkom.controller;
00002
00003 import org.springframework.beans.factory.annotation.Autowired;
00004 import org.springframework.web.bind.annotation.CrossOrigin;
00005 import org.springframework.web.bind.annotation.GetMapping;
00006 import org.springframework.web.bind.annotation.RequestMapping;
00007 import org.springframework.web.bind.annotation.RestController;
00008
00009 import com.mklachl.sopkom.helpers.Helpers;
00010 import com.mklachl.sopkom.model.dto.UserDto;
00011 import com.mklachl.sopkom.model.entity.User;
00012 import com.mklachl.sopkom.services.UserService;
00013
00014 @RestController
00015 @RequestMapping("/user")
00016 @CrossOrigin(origins = {"*"}, maxAge = 3600)
00017 public class UserController {
00018
00019     @Autowired
00020     private UserService userService;
00021
00022     @GetMapping
00023     public UserDto getLoggedInUser() {
00024         User user = Helpers.getUserFromContext(userService);
00025
00026         if (user == null) {
00027             throw new NullPointerException("critical exception!!!!");
00028         }
00029
00030         return new UserDto(user);
00031     }
00032 }
```

7.31 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/exceptions/Icydent← Exception.java

Komponenty

- class [com.mklachl.sopkom.exceptions.IcydentException](#)

Pakiety

- package [com.mklachl.sopkom.exceptions](#)

7.32 IncydentException.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.exceptions;
00002
00003 public class IncydentException extends Exception {
00004
00005     public IncydentException(String m) {
00006         super(m);
00007     }
00008
00012     private static final long serialVersionUID = -5933687264500607949L;
00013
00014 }
```

7.33 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/exceptions/LiniaNot← FoundException.java

Komponenty

- class [com.mklachl.sopkom.exceptions.LiniaNotFoundException](#)

Pakiety

- package [com.mklachl.sopkom.exceptions](#)

7.34 LiniaNotFoundException.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.exceptions;
00002
00003 public class LiniaNotFoundException extends Exception {
00004
00005     public LiniaNotFoundException(Long id) {
00006         super("could not found linia s" + id);
00007     }
00008
00012     private static final long serialVersionUID = -5933687264500607949L;
00013
00014 }
```

7.35 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ ← Backend/src/main/java/com/mklachl/sopkom/exceptions/ ← PrzystanekNotFoundException.java

Komponenty

- class [com.mklachl.sopkom.exceptions.PrzystanekNotFoundException](#)

Pakiety

- package [com.mklachl.sopkom.exceptions](#)

7.36 PrzystanekNotFoundException.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.exceptions;  
00002  
00003 public class PrzystanekNotFoundException extends Exception {  
00004     private static final long serialVersionUID = 7349300203842523635L;  
00005  
00010     public PrzystanekNotFoundException(Long liniaId, Long przystanekid) {  
00011         super("cannot found przystane with id:" + przystanekid + " , that was meant to be for linia "  
00012             + liniaId);  
00013     }  
00014  
00015  
00016 }
```

7.37 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ ← Backend/src/main/java/com/mklachl/sopkom/filter/Logging ← Filter.java

Komponenty

- class [com.mklachl.sopkom.filter.LoggingFilter](#)

Pakiety

- package [com.mklachl.sopkom.filter](#)

7.38 LoggingFilter.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.filter;
00002
00003 import java.io.IOException;
00004
00005 import org.springframework.stereotype.Component;
00006 import org.springframework.web.filter.OncePerRequestFilter;
00007
00008 import jakarta.servlet.FilterChain;
00009 import jakarta.servlet.ServletException;
00010 import jakarta.servlet.http.HttpServletRequest;
00011 import jakarta.servlet.http.HttpServletResponse;
00012
00013 @Component
00014 public class LoggingFilter extends OncePerRequestFilter {
00015     @Override
00016     protected void doFilterInternal(HttpServletRequest request, HttpServletResponse response,
00017         FilterChain filterChain) throws ServletException, IOException {
00018         System.out.printf("Handling incoming request: %s, %s\n",
00019             request.getRequestURI(),
00020             request.getMethod());
00021         try {
00022             filterChain.doFilter(request, response);
00023         } catch (Exception e) {
00024             System.out.println(e.getMessage());
00025         }
00026     }
00027 }
```

7.39 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/helpers/Helpers.java

Komponenty

- class [com.mklachl.sopkom.helpers.Helpers](#)

Pakiety

- package [com.mklachl.sopkom.helpers](#)

7.40 Helpers.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.helpers;
00002
00003 import org.springframework.security.core.Authentication;
00004 import org.springframework.security.core.context.SecurityContext;
00005 import org.springframework.security.core.context.SecurityContextHolder;
00006 import org.springframework.ui.Model;
00007
00008 import com.mklachl.sopkom.model.entity.User;
00009 import com.mklachl.sopkom.services.UserService;
00010
00011 public final class Helpers {
00012
00013
00014
00015     public static void inject(Model model) {
00016         SecurityContext context = SecurityContextHolder.getContext();
00017         Authentication auth = context.getAuthentication();
00018         model.addAttribute("logged", auth != null && !("anonymousUser").equals(auth.getName()));
00019     }
00020 }
```

```
00019     model.addAttribute("isAmator", auth != null && auth.getAuthorities().stream().anyMatch(role  
00020         -> role.getAuthority().equals("ROLE_AMATOR")));  
00021     model.addAttribute("isProfi", auth != null && auth.getAuthorities().stream().anyMatch(role ->  
00022         role.getAuthority().equals("ROLE_PROFIE")));  
00023     model.addAttribute("isAdmin", auth != null && auth.getAuthorities().stream().anyMatch(role ->  
00024         role.getAuthority().equals("ROLE_ADMIN")));  
00025 }  
00026  
00027     public static User getUserFromContext(UserService service) {  
00028         SecurityContext context = SecurityContextHolder.getContext();  
00029         Authentication auth = context.getAuthentication();  
00030         return service.findUserByLogin(auth.getName());  
00031     }
```

7.41 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ Backend/src/main/java/com/mklachl/sopkom/model/dto/Autobus_ Dto.java

Komponenty

- class [com.mklachl.sopkom.model.dto.AutobusDto](#)

Pakiety

- package [com.mklachl.sopkom.model.dto](#)

7.42 AutobusDto.java

Idź do dokumentacji tego pliku.

```
00001 package com.mklachl.sopkom.model.dto;  
00002  
00003 import com.mklachl.sopkom.model.entity.Autobus;  
00004 import java.util.Date;  
00005  
00009 public class AutobusDto {  
00010     private Long autbousId;  
00015     private String numerRejestracyjny;  
00020     private Date przegladWaznyDo;  
00025     private String status;  
00030     private float przebieg;  
00035  
00039     public AutobusDto() {  
00040     }  
00041  
00046     public AutobusDto(Autobus autobus) {  
00047         this.autbousId = autobus.getAutbousId();  
00048         this.numerRejestracyjny = autobus.getNumerRejestracyjny();  
00049         this.przegladWaznyDo = autobus.getPrzegladWaznyDo();  
00050         this.status = autobus.getStatus();  
00051         this.przebieg = autobus.getPrzebieg();  
00052     }  
00053  
00054     // Getters  
00055  
00060     public Long getAutbousId() {  
00061         return autbousId;  
00062     }
```

```

00063
00068     public String getNumerRejestracyjny() {
00069         return numerRejestracyjny;
00070     }
00071
00076     public Date getPrzegladWaznyDo() {
00077         return przegladWaznyDo;
00078     }
00079
00084     public String getStatus() {
00085         return status;
00086     }
00087
00092     public float getPrzebieg() {
00093         return przebieg;
00094     }
00095
00096     // Setters
00097
00102     public void setAutbousId(Long autbousId) {
00103         this.autbousId = autbousId;
00104     }
00105
00110     public void setNumerRejestracyjny(String numerRejestracyjny) {
00111         this.numerRejestracyjny = numerRejestracyjny;
00112     }
00113
00118     public void setPrzegladWaznyDo(Date przegladWaznyDo) {
00119         this.przegladWaznyDo = przegladWaznyDo;
00120     }
00121
00126     public void setStatus(String status) {
00127         this.status = status;
00128     }
00129
00134     public void setPrzebieg(float przebieg) {
00135         this.przebieg = przebieg;
00136     }
00137
00142     @Override
00143     public String toString() {
00144         return "AutobusDto [autbousId=" + autbousId + ", numerRejestracyjny=" + numerRejestracyjny
00145             + ", przegladWaznyDo=" + przegladWaznyDo + ", status=" + status + ", przebieg=" +
00146             przebieg + "]";
00147     }

```

7.43 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ ← Backend/src/main/java/com/mklachl/sopkom/model/dto/Bilet ← Dto.java

Komponenty

- class [com.mklachl.sopkom.model.dto.BiletDto](#)

Pakiety

- package [com.mklachl.sopkom.model.dto](#)

7.44 BiletDto.java

[Idź do dokumentacji tego pliku.](#)

```

00001 package com.mklachl.sopkom.model.dto;
00002
00003 import com.mklachl.sopkom.model.entity.Bilet;

```

```
00004  
00008 public class BiletDto {  
00009     private Short biletId;  
00014     private float cena;  
00019     private String nazwa;  
00024  
00028     public BiletDto() {  
00029     }  
00030  
00035     public BiletDto(Bilet bilet) {  
00036         this.biletId = bilet.getBiletId();  
00037         this.cena = bilet.getCena();  
00038         this.nazwa = bilet.getNazwa();  
00039     }  
00040  
00041     // Getters and Setters  
00042  
00047     public Short getBiletId() {  
00048         return biletId;  
00049     }  
00050  
00055     public void setBiletId(Short biletId) {  
00056         this.biletId = biletId;  
00057     }  
00058  
00063     public float getCena() {  
00064         return cena;  
00065     }  
00066  
00071     public void setCena(float cena) {  
00072         this.cena = cena;  
00073     }  
00074  
00079     public String getNazwa() {  
00080         return nazwa;  
00081     }  
00082  
00087     public void setNazwa(String nazwa) {  
00088         this.nazwa = nazwa;  
00089     }  
00090 }
```

7.45 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_ ← Backend/src/main/java/com/mklachl/sopkom/model/dto/Date← WrapperDto.java

Komponenty

- class [com.mklachl.sopkom.model.dto.DateWrapperDto](#)

Pakiety

- package [com.mklachl.sopkom.model.dto](#)

7.46 DateWrapperDto.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.model.dto;  
00002  
00003 import com.fasterxml.jackson.annotation.JsonFormat;  
00004 import java.util.Date;  
00005
```

```

00009 public class DateWrapperDto {
00010
00014     @JsonFormat(pattern = "dd.MM.yyyy")
00015     private Date date;
00016
00021     public DateWrapperDto(Date date) {
00022         this.date = date;
00023     }
00024
00028     public DateWrapperDto() {
00029     }
00030
00035     public Date getDate() {
00036         return date;
00037     }
00038
00043     public void setDate(Date date) {
00044         this.date = date;
00045     }
00046 }
```

7.47 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/model/dto/harmonogram/← HarmonogramDto.java

Komponenty

- class [com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto](#)

Pakiety

- package [com.mklachl.sopkom.model.dto.harmonogram](#)

7.48 HarmonogramDto.java

[Idź do dokumentacji tego pliku.](#)

```

00001 package com.mklachl.sopkom.model.dto.harmonogram;
00002
00003 import com.mklachl.sopkom.model.entity.Harmonogram;
00004
00008 public class HarmonogramDto {
00009
00013     private short harmonogramId;
00014
00018     private String nazwa;
00019
00023     private boolean pon;
00024
00028     private boolean wto;
00029
00033     private boolean sro;
00034
00038     private boolean czw;
00039
00043     private boolean pia;
00044
00048     private boolean sob;
00049
00053     private boolean nie;
00054
00058     private String dodatkoweInf;
00059
00063     public HarmonogramDto() {
00064     }
00065 }
```

```
00070     public HarmonogramDto(Harmonogram harmonogram) {
00071         this.harmonogramId = harmonogram.getHarmonogramId();
00072         this.nazwa = harmonogram.getNazwa();
00073         this.pon = harmonogram.isPon();
00074         this.wto = harmonogram.isWto();
00075         this.sro = harmonogram.isSro();
00076         this.czw = harmonogram.isCzw();
00077         this.pia = harmonogram.isPia();
00078         this.sob = harmonogram.isSob();
00079         this.nie = harmonogram.isNie();
00080         this.dodatkoweInf = harmonogram.getDodatkoweInf();
00081     }
00082
00083     // Getters and Setters
00084
00085     public short getHarmonogramId() {
00086         return harmonogramId;
00087     }
00088
00089     public void setHarmonogramId(short harmonogramId) {
00090         this.harmonogramId = harmonogramId;
00091     }
00092
00093     public String getNazwa() {
00094         return nazwa;
00095     }
00096
00097     public void setNazwa(String nazwa) {
00098         this.nazwa = nazwa;
00099     }
00100
00101     public boolean isPon() {
00102         return pon;
00103     }
00104
00105     public void setPon(boolean pon) {
00106         this.pon = pon;
00107     }
00108
00109     public boolean isWto() {
00110         return wto;
00111     }
00112
00113     public void setWto(boolean wto) {
00114         this.wto = wto;
00115     }
00116
00117     public boolean isSro() {
00118         return sro;
00119     }
00120
00121     public void setSro(boolean sro) {
00122         this.sro = sro;
00123     }
00124
00125     public boolean isCzw() {
00126         return czw;
00127     }
00128
00129     public void setCzw(boolean czw) {
00130         this.czw = czw;
00131     }
00132
00133     public boolean isPia() {
00134         return pia;
00135     }
00136
00137     public void setPia(boolean pia) {
00138         this.pia = pia;
00139     }
00140
00141     public boolean isSob() {
00142         return sob;
00143     }
00144
00145     public void setSob(boolean sob) {
00146         this.sob = sob;
00147     }
00148
00149     public boolean isNie() {
00150         return nie;
00151     }
00152
00153     public void setNie(boolean nie) {
00154         this.nie = nie;
00155     }
00156
00157
00158
00159
00160
00161
00162
00163
00164
00165
00166
00167
00168
00169
00170
00171
00172
00173
00174
00175
00176
00177
00178
00179
00180
00181
00182
00183
00184
00185
00186
00187
00188
00189
00190
00191
00192
00193
00194
00195
00196
00197
00198
00199
00200
00201
00202
00203
00204
00205
00206
00207
00208
00209
00210
00211
00212
00213
00214
00215
00216
00217
00218
00219
00220
00221
00222
00223
00224
00225
00226
00227
00228
```

```

00233     public String getDodatkoweInf() {
00234         return dodatkoweInf;
00235     }
00236
00241     public void setDodatkoweInf(String dodatkoweInf) {
00242         this.dodatkoweInf = dodatkoweInf;
00243     }
00244 }
```

7.49 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_↔ Backend/src/main/java/com/mklachl/sopkom/model/dto/Icydent↔ Dto.java

Komponenty

- class [com.mklachl.sopkom.model.dto.IcydentDto](#)

Pakiety

- package [com.mklachl.sopkom.model.dto](#)

7.50 IncydentDto.java

[Idź do dokumentacji tego pliku.](#)

```

00001 package com.mklachl.sopkom.model.dto;
00002
00003 import com.fasterxml.jackson.annotation.JsonFormat;
00004 import com.mklachl.sopkom.model.entity.Incydent;
00005 import java.util.Date;
00006
00010 public class IncydentDto {
00011
00015     private Long incydentId;
00016
00020     private String typ;
00021
00025     @JsonFormat(pattern = "dd.MM.yyyy")
00026     private Date date;
00027
00031     private Long autobusId;
00032
00036     private Long kierowcaId;
00037
00041     private Long przejazdId;
00042
00046     private String dodatkoweInformacje;
00047
00051     private String krotko;
00052
00056     private float koszty;
00057
00061     public IncydentDto() {}
00062
00067     public IncydentDto(Incydent incydent) {
00068         this.krotko = incydent.getKrotko();
00069         this.incydentId = incydent.getIncydentId();
00070         this.typ = incydent.getTyp();
00071         this.date = incydent.getDate();
00072         this.autobusId = incydent.getAutobus() != null ? incydent.getAutobus().getAutbousId() : null;
00073         this.kierowcaId = incydent.getKierowca() != null ? incydent.getKierowca().getKierowcaId() :
00074             null;
00074         this.przejazdId = incydent.getPrzejazd() != null ? incydent.getPrzejazd().getPrzejazdId() :
00075             null;
00075         this.dodatkoweInformacje = incydent.getDodatkoweInformacje();
00076         this.koszty = incydent.getKoszty();
```

```
00077     }
00078
00079     // Getters and Setters
00080
00085     public String getKrotko() {
00086         return krotko;
00087     }
00088
00093     public void setKrotko(String krotko) {
00094         this.krotko = krotko;
00095     }
00096
00101     public Long getIncydentId() {
00102         return incydentId;
00103     }
00104
00109     public void setIncydentId(Long incydentId) {
00110         this.incydentId = incydentId;
00111     }
00112
00117     public String getTyp() {
00118         return typ;
00119     }
00120
00125     public void setTyp(String typ) {
00126         this.typ = typ;
00127     }
00128
00133     public Date getDate() {
00134         return date;
00135     }
00136
00141     public void setDate(Date date) {
00142         this.date = date;
00143     }
00144
00149     public Long getAutobusId() {
00150         return autobusId;
00151     }
00152
00157     public void setAutobusId(Long autobusId) {
00158         this.autobusId = autobusId;
00159     }
00160
00165     public Long getKierowcaId() {
00166         return kierowcaId;
00167     }
00168
00173     public void setKierowcaId(Long kierowcaId) {
00174         this.kierowcaId = kierowcaId;
00175     }
00176
00181     public Long getPrzejazdId() {
00182         return przejazdId;
00183     }
00184
00189     public void setPrzejazdId(Long przejazdId) {
00190         this.przejazdId = przejazdId;
00191     }
00192
00197     public String getDodatkoweInformacje() {
00198         return dodatkoweInformacje;
00199     }
00200
00205     public void setDodatkoweInformacje(String dodatkoweInformacje) {
00206         this.dodatkoweInformacje = dodatkoweInformacje;
00207     }
00208
00213     public float getKoszty() {
00214         return koszty;
00215     }
00216
00221     public void setKoszty(float koszty) {
00222         this.koszty = koszty;
00223     }
00224 }
```

7.51 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/model/dto/Icydent← DtoOutput.java

Komponenty

- class [com.mklachl.sopkom.model.dto.IncydentDtoOutput](#)

Pakiety

- package [com.mklachl.sopkom.model.dto](#)

7.52 IncydentDtoOutput.java

[Idź do dokumentacji tego pliku.](#)

```

00001 package com.mklachl.sopkom.model.dto;
00002
00003 import java.util.Date;
00004 import com.fasterxml.jackson.annotation.JsonFormat;
00005 import com.mklachl.sopkom.model.dto.kurs.KursDto;
00006 import com.mklachl.sopkom.model.entity.Incydent;
00007
00011 public class IncydentDtoOutput extends IncydentDto {
00012
00016     private KierowcaDto kierowca;
00021     private AutobusDto autobus;
00022
00026     private KursDto kurs;
00027
00031     @JsonFormat(pattern = "EE MMM d y H:m:s ZZZ", locale = "en-US")
00032     private Date data;
00033
00038     public IncydentDtoOutput(Incydent incydent) {
00039         super(incydent);
00040
00041         if (incydent.getAutobus() != null) {
00042             this.autobus = new AutobusDto(incydent.getAutobus());
00043         }
00044
00045         if (incydent.getKierowca() != null) {
00046             this.kierowca = new KierowcaDto(incydent.getKierowca());
00047         }
00048
00049         if (incydent.getPrzejazd() != null) {
00050             this.kurs = new KursDto(incydent.getPrzejazd().getKurs());
00051             data = incydent.getPrzejazd().getData();
00052         }
00053     }
00054
00055     // Getters and setters for new fields
00056
00061     public KierowcaDto getKierowca() {
00062         return kierowca;
00063     }
00064
00069     public void setKierowca(KierowcaDto kierowca) {
00070         this.kierowca = kierowca;
00071     }
00072
00077     public AutobusDto getAutobus() {
00078         return autobus;
00079     }
00080
00085     public void setAutobus(AutobusDto autobus) {
00086         this.autobus = autobus;
00087     }
00088
00093     public KursDto getKurs() {

```

```
00094         return kurs;
00095     }
00096
00101     public void setKurs(KursDto kurs) {
00102         this.kurs = kurs;
00103     }
00104 }
```

7.53 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ ← Backend/src/main/java/com/mklachl/sopkom/model/dto/Kierowca← Dto.java

Komponenty

- class [com.mklachl.sopkom.model.dto.KierowcaDto](#)

Pakiety

- package [com.mklachl.sopkom.model.dto](#)

7.54 KierowcaDto.java

Idź do dokumentacji tego pliku.

```
00001 package com.mklachl.sopkom.model.dto;
00002
00003 import com.mklachl.sopkom.model.entity.Kierowca;
00004 import com.mklachl.sopkom.model.entity.User;
00005 import java.util.Date;
00006
00010 public class KierowcaDto {
00011
00015     private Long kierowcaId;
00016
00020     private Date prawoJazdyWazneDo;
00021
00025     private String imie;
00026
00030     private String nazwisko;
00031
00035     private String pesel;
00036
00040     private UserDto user;
00041
00045     public KierowcaDto() {}
00046
00051     public KierowcaDto(Kierowca kierowca) {
00052         this.kierowcaId = kierowca.getKierowcaId();
00053         this.prawoJazdyWazneDo = kierowca.getPrawoJazdyWazneDo();
00054         this.imie = kierowca.getImie();
00055         this.nazwisko = kierowca.getNazwisko();
00056         this.pesel = kierowca.getPesel();
00057
00058         if (kierowca.getUser() != null) {
00059             this.user = new UserDto(kierowca.getUser());
00060         }
00061     }
00062
00063     // Getters
00064
00069     public Long getKierowcaId() {
00070         return kierowcaId;
00071     }
00072
00077     public Date getPrawoJazdyWazneDo() {
00078         return prawoJazdyWazneDo;
```

```

00079     }
00080
00085     public String getImie() {
00086         return imie;
00087     }
00088
00093     public String getNazwisko() {
00094         return nazwisko;
00095     }
00096
00101     public String getPesel() {
00102         return pesel;
00103     }
00104
00109     public UserDto getUser() {
00110         return user;
00111     }
00112
00113 // Setters
00114
00119     public void setKierowcaId(Long kierowcaId) {
00120         this.kierowcaId = kierowcaId;
00121     }
00122
00127     public void setPrawoJazdyWazneDo(Date prawoJazdyWazneDo) {
00128         this.prawoJazdyWazneDo = prawoJazdyWazneDo;
00129     }
00130
00135     public void setImie(String imie) {
00136         this.imie = imie;
00137     }
00138
00143     public void setNazwisko(String nazwisko) {
00144         this.nazwisko = nazwisko;
00145     }
00146
00151     public void setPesel(String pesel) {
00152         this.pesel = pesel;
00153     }
00154
00159     public void setUserDto(UserDto user) {
00160         this.user = user;
00161     }
00162
00167     @Override
00168     public String toString() {
00169         return "KierowcaDto [kierowcaId=" + kierowcaId + ", prawoJazdyWazneDo=" + prawoJazdyWazneDo +
00170             ", imie=" + imie +
00171             ", nazwisko=" + nazwisko + ", pesel=" + pesel + ", user=" + user + "]";
00172     }

```

7.55 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/model/dto/kurs/← AbstractKursDto.java

Komponenty

- class [com.mklachl.sopkom.model.dto.kurs.AbstractKursDto](#)

Pakiety

- package [com.mklachl.sopkom.model.dto.kurs](#)

7.56 AbstractKursDto.java

Idź do dokumentacji tego pliku.

```
00001 package com.mklachl.sopkom.model.dto.kurs;
00002
00003 import com.mklachl.sopkom.model.entity.Kurs;
00004
00008 public abstract class AbstractKursDto {
00009
00013     protected Long kursId;
00014
00018     protected Long liniaId;
00019
00023     protected Short harmonogramId;
00024
00028     protected Short kierunek;
00029
00033     protected Short typ_autobusu;
00034
00039     public AbstractKursDto(Kurs kurs) {
00040         this.setKursId(kurs.getKursId());
00041         this.setKierunek(kurs.getKierunek());
00042         this.setTyp_autobusu(kurs.getTypAutobusu());
00043     }
00044
00048     public AbstractKursDto() {
00049     }
00050
00051     // Getters and Setters
00052
00057     public Long getKursId() {
00058         return kursId;
00059     }
00060
00065     public void setKursId(Long kursId) {
00066         this.kursId = kursId;
00067     }
00068
00073     public Long getLiniaId() {
00074         return liniaId;
00075     }
00076
00081     public void setLiniaId(Long liniaId) {
00082         this.liniaId = liniaId;
00083     }
00084
00089     public Short getHarmonogramId() {
00090         return harmonogramId;
00091     }
00092
00097     public void setHarmonogramId(Short harmonogramId) {
00098         this.harmonogramId = harmonogramId;
00099     }
00100
00105     public Short getKierunek() {
00106         return kierunek;
00107     }
00108
00113     public void setKierunek(Short kierunek) {
00114         this.kierunek = kierunek;
00115     }
00116
00121     public Short getTyp_autobusu() {
00122         return typ_autobusu;
00123     }
00124
00129     public void setTyp_autobusu(Short typ_autobusu) {
00130         this.typ_autobusu = typ_autobusu;
00131     }
00132 }
```

7.57 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ ↵ Backend/src/main/java/com/mklachl/sopkom/model/dto/kurs/ ↵ AbstractPrzystanekWKursieDto.java

Komponenty

- class [com.mklachl.sopkom.model.dto.kurs.AbstractPrzystanekWKursieDto](#)

Pakietы

- package [com.mklachl.sopkom.model.dto.kurs](#)

7.58 AbstractPrzystanekWKursieDto.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.model.dto.kurs;
00002
00003 import com.fasterxml.jackson.annotation.JsonFormat;
00004 import com.mklachl.sopkom.model.entity.KursPrzystanekWlini;
00005 import java.sql.Time;
00006
00010 public class AbstractPrzystanekWKursieDto {
00011
00015     protected Long przystanekwKursieId;
00016
00020     @JsonFormat(pattern = "hh:MM:ss")
00021     protected Time godzinna;
00022
00026     public AbstractPrzystanekWKursieDto() {
00027     }
00028
00033     public AbstractPrzystanekWKursieDto(KursPrzystanekWlini kpw) {
00034         this.godzinna = new Time(kpw.getGodzinna().getTime());
00035         this.przystanekwKursieId = kpw.getKursPrzystanekWliniId();
00036     }
00037
00038 // Getters and Setters
00039
00044     public Long getPrzystanekwKursieId() {
00045         return przystanekwKursieId;
00046     }
00047
00052     public void setPrzystanekwKursieId(Long przystanekwKursieId) {
00053         this.przystanekwKursieId = przystanekwKursieId;
00054     }
00055
00060     public Time getGodzinna() {
00061         return godzinna;
00062     }
00063
00068     public void setGodzinna(Time godzinna) {
00069         this.godzinna = godzinna;
00070     }
00071 }
```

7.59 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ ↵ Backend/src/main/java/com/mklachl/sopkom/model/dto/kurs/Input ↵ KursDto.java

Komponenty

- class [com.mklachl.sopkom.model.dto.kurs.InputKursDto](#)

Pakiety

- package com.mklachl.sopkom.model.dto.kurs

7.60 InputKursDto.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.model.dto.kurs;
00002
00003 import java.util.List;
00004
00008 public class InputKursDto extends AbstractKursDto {
00009
0013     public List<InputPrzystanekWKursieDto> przystanki;
0014
0018     public InputKursDto() {
0019 }
0020
0021     // Getters and Setters
0022
0027     public List<InputPrzystanekWKursieDto> getPrzystanki() {
0028         return przystanki;
0029     }
0030
0035     public void setPrzystanki(List<InputPrzystanekWKursieDto> przystanki) {
0036         this.przystanki = przystanki;
0037     }
0038 }
```

7.61 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/model/dto/kurs/Input← PrzystanekWKursieDto.java

Komponenty

- class com.mklachl.sopkom.model.dto.kurs.InputPrzystanekWKursieDto

Pakiety

- package com.mklachl.sopkom.model.dto.kurs

7.62 InputPrzystanekWKursieDto.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.model.dto.kurs;
00002
00003 import com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLinia;
00004 import com.mklachl.sopkom.model.entity.KursPrzystanekWLini;
00005
00009 public class InputPrzystanekWKursieDto extends AbstractPrzystanekWKursieDto {
00010
0014     private PrzystanekDtoDlaLinia przystanekWLini;
00015
0019     public InputPrzystanekWKursieDto() {
0020 }
0021
0026     public InputPrzystanekWKursieDto(KursPrzystanekWLini kpw) {
0027         super(kpw);
```

```

00028     this.przystanekWLini = new PrzystanekDtoDlaLinia(kpw.getPrzystanekWLini());
00029 }
00030
00031 // Getters and Setters
00032
00033 public PrzystanekDtoDlaLinia getPrzystanekWLini() {
00034     return przystanekWLini;
00035 }
00036
00037 public void setPrzystanekWLini(PrzystanekDtoDlaLinia przystanekWLini) {
00038     this.przystanekWLini = przystanekWLini;
00039 }
00040
00041 }
00042
00043 }
```

7.63 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/model/dto/kurs/Kurs← Dto.java

Komponenty

- class [com.mklachl.sopkom.model.dto.kurs.KursDto](#)

Pakiety

- package [com.mklachl.sopkom.model.dto.kurs](#)

7.64 KursDto.java

[Idź do dokumentacji tego pliku.](#)

```

00001 package com.mklachl.sopkom.model.dto.kurs;
00002
00003 import java.util.ArrayList;
00004 import java.util.List;
00005 import com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto;
00006 import com.mklachl.sopkom.model.dto.linia.LiniaDto;
00007 import com.mklachl.sopkom.model.entity.Kurs;
00008
00012 public class KursDto extends AbstractKursDto {
00013
00017     private List<InputPrzystanekWKursieDto> przystanki;
00018
00022     private HarmonogramDto harmonogram;
00023
00027     private LiniaDto linia;
00028
00032     private boolean odwracalny;
00033
00037     public KursDto() {
00038 }
00039
00044     public KursDto(Kurs kurs) {
00045         super(kurs);
00046         this.setLinia(new LiniaDto(kurs.getLinia()));
00047         this.setHarmonogram(new HarmonogramDto(kurs.getHarmonogram()));
00048         this.setOdwracalny(kurs.getLinia().odwracalna());
00049         this.przystanki = new ArrayList<>();
00050
00051         for (var k : kurs.getKursPrzystanekWLini()) {
00052             przystanki.add(new InputPrzystanekWKursieDto(k));
00053         }
00054     }
00055
00056 // Getters and Setters
00057
00062     public List<InputPrzystanekWKursieDto> getPrzystanki() {
```

```
00063     return przystanki;
00064 }
00065
00070 public void setPrzystanki(List<InputPrzystanekWKursieDto> przystanki) {
00071     this.przystanki = przystanki;
00072 }
00073
00078 public HarmonogramDto getHarmonogram() {
00079     return harmonogram;
00080 }
00081
00086 public void setHarmonogram(HarmonogramDto harmonogram) {
00087     this.setHarmonogramId(harmonogram.getHarmonogramId());
00088     this.harmonogram = harmonogram;
00089 }
00090
00095 public LiniaDto getLinia() {
00096     return linia;
00097 }
00098
00103 public void setLinia(LiniaDto linia) {
00104     this.linia = linia;
00105     this.setLiniaId(linia.getId());
00106 }
00107
00112 public boolean getOdwracalny() {
00113     return odwracalny;
00114 }
00115
00120 public void setOdwracalny(boolean odwracalny) {
00121     this.odwracalny = odwracalny;
00122 }
00123 }
```

7.65 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ ← Backend/src/main/java/com/mklachl/sopkom/model/dto/kurs/ ← OUTDATED/KursDtoOUTDATED.java

Komponenty

- class [com.mklachl.sopkom.model.dto.kurs.OUTDATED.KursDtoOUTDATED](#)

Pakiety

- package [com.mklachl.sopkom.model.dto.kurs.OUTDATED](#)

7.66 KursDtoOUTDATED.java

Idź do dokumentacji tego pliku.

```
00001 package com.mklachl.sopkom.model.dto.kurs.OUTDATED;
00002
00003 import java.util.List;
00004
00005 import com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto;
00006 import com.mklachl.sopkom.model.dto.linia.LiniaDto;
00007 import com.mklachl.sopkom.model.entity.Kurs;
00008 import com.mklachl.sopkom.model.entity.Linia;
00009
00010 public class KursDtoOUTDATED {
00011
00012     protected LiniaDto linia;
00013     protected Long kursId;
00014
00015     protected boolean odwracalny;
00016     // ZERO - normal // ONE - odwrony
```

```

00017     protected Short kieruenk;
00018
00019     protected HarmonogramDto harmonogram;
00020
00021     protected List<PrzystenekWKursieDtoOUTDATED> przystanki;
00022
00023     public KursDtoOUTDATED() {
00024
00025     }
00026
00027     public KursDtoOUTDATED(Kurs kurs) {
00028         Linia linia = kurs.getLinia();
00029         odwracalny = linia.odwracalna();
00030         kieruenk = kurs.getKierunek();
00031         throw new RuntimeException("NOT IMPLEMENTED");
00032     }
00033
00034     public Long getKursId() {
00035         return kursId;
00036     }
00037
00038     public void setKursId(Long kursId) {
00039         this.kursId = kursId;
00040     }
00041
00042     public HarmonogramDto getHarmonogram() {
00043         return harmonogram;
00044     }
00045
00046     public void setHarmonogram(HarmonogramDto harmonogram) {
00047         this.harmonogram = harmonogram;
00048     }
00049
00050     public LiniaDto getLinia() {
00051         return linia;
00052     }
00053
00054     public void setLinia(LiniaDto linia) {
00055         this.linia = linia;
00056     }
00057
00058     public boolean isOdwracalny() {
00059         return odwracalny;
00060     }
00061
00062     public void setOdwracalny(boolean odwracalny) {
00063         this.odwracalny = odwracalny;
00064     }
00065
00066     public List<PrzystenekWKursieDtoOUTDATED> getPrzystanki() {
00067         return przystanki;
00068     }
00069
00070     public void setPrzystanki(List<PrzystenekWKursieDtoOUTDATED> przystanki) {
00071         this.przystanki = przystanki;
00072     }
00073
00074     public Short getKieruenk() {
00075         return kieruenk;
00076     }
00077
00078     public void setKieruenk(Short kieruenk) {
00079         this.kieruenk = kieruenk;
00080     }
00081
00082
00083
00084
00085 }

```

7.67 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_[←](#) Backend/src/main/java/com/mklachl/sopkom/model/dto/kurs/[←](#) OUTDATED/PrzystenekWKursieDtoOUTDATED.java

Komponenty

- class [com.mklachl.sopkom.model.dto.kurs.OUTDATED.PrzystenekWKursieDtoOUTDATED](#)

Pakiety

- package com.mklachl.sopkom.model.dto.kurs.OUTDATED

7.68 PrzystenekWKursieDtoOUTDATED.java

Idź do dokumentacji tego pliku.

```
00001 package com.mklachl.sopkom.model.dto.kurs.OUTDATED;
00002
00003 import java.sql.Time;
00004
00005 import com.mklachl.sopkom.model.dto.PrzystanekDto;
00006 import com.mklachl.sopkom.model.dto.PrzystanekDtoFull;
00007 import com.mklachl.sopkom.model.entity.KursPrzystanekWlini;
00008
00009 public class PrzystenekWKursieDtoOUTDATED {
00010     protected Long przystanekkurs_id;
00012
00013     protected PrzystanekDtoFull przystanek;
00014
00015     protected short kolejnosc;
00016
00017     protected Time godzinna;
00018
00019     public PrzystenekWKursieDtoOUTDATED() {
00020
00021     }
00022
00023     public PrzystenekWKursieDtoOUTDATED(KursPrzystanekWlini kwl) {
00024         godzinna = new Time(kwl.getGodzinna().getTime());
00025         przystanek = new PrzystanekDtoFull(kwl.getPrzystanekWlini().getPrzystanek());
00026         kolejnosc = kwl.getPrzystanekWlini().getKolejnosc();
00027         przystanekkurs_id = Long.valueOf(9);
00028
00029     }
00030
00031     public Long getPrzystanekkurs_id() {
00032         return przystanekkurs_id;
00033     }
00034
00035     public void setPrzystanekkurs_id(Long przystanekkurs_id) {
00036         this.przystanekkurs_id = przystanekkurs_id;
00037     }
00038
00039     public PrzystanekDtoFull getPrzystanek() {
00040         return przystanek;
00041     }
00042
00043     public void setPrzystanek(PrzystanekDtoFull przystanek) {
00044         this.przystanek = przystanek;
00045     }
00046
00047     public Time getGodzinna() {
00048         return godzinna;
00049     }
00050
00051     public void setGodzinna(Time t) {
00052         this.godzinna = t;
00053     }
00054
00055
00056
00057 }
```

7.69 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/model/dto/kurs/← PrzystenekWKursieDto.java

Komponenty

- class com.mklachl.sopkom.model.dto.kurs.PrzystenekWKursieDto

Pakiety

- package com.mklachl.sopkom.model.dto.kurs

7.70 PrzystanekWKursieDto.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.model.dto.kurs;
00002
00003 import com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLinia;
00004 import com.mklachl.sopkom.model.entity.KursPrzystanekWLini;
00005
00009 public class PrzystanekWKursieDto extends AbstractPrzystanekWKursieDto {
00010
00014     private PrzystanekDtoDlaLinia przystanekWLini;
00015
00019     public PrzystanekWKursieDto() {
00020     }
00021
00026     public PrzystanekWKursieDto(KursPrzystanekWLini kpw) {
00027         super(kpw);
00028         this.przystanekWLini = new PrzystanekDtoDlaLinia(kpw.getPrzystanekWLini());
00029     }
00030
00031     // Getters and Setters
00032
00037     public PrzystanekDtoDlaLinia getPrzystanekWLini() {
00038         return przystanekWLini;
00039     }
00040
00045     public void setPrzystanekWLini(PrzystanekDtoDlaLinia przystanekWLini) {
00046         this.przystanekWLini = przystanekWLini;
00047     }
00048 }
```

7.71 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_←Backend/src/main/java/com/mklachl/sopkom/model/dto/linia/Linia←Dto.java

Komponenty

- class com.mklachl.sopkom.model.dto.linia.LiniaDto

Pakiety

- package com.mklachl.sopkom.model.dto.linia

7.72 LiniaDto.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.model.dto.linia;
00002
00003 import com.mklachl.sopkom.model.entity.Linia;
00004
00008 public class LiniaDto {
00009
00013     protected Long id;
00014 }
```

```
00018     protected String numer;
00019
00023     public LiniaDto() {
00024     }
00025
00030     public LiniaDto(Linia linia) {
00031         this.id = linia.getId();
00032         this.numer = linia.getNumer();
00033     }
00034
00035     // Getters and Setters
00036
00041     public Long getId() {
00042         return id;
00043     }
00044
00049     public void setId(Long id) {
00050         this.id = id;
00051     }
00052
00057     public String getNumer() {
00058         return numer;
00059     }
00060
00065     public void setNumer(String numer) {
00066         this.numer = numer;
00067     }
00068 }
```

7.73 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_ ← Backend/src/main/java/com/mklachl/sopkom/model/dto/linia/LiniaDtoInput.java

Komponenty

- class [com.mklachl.sopkom.model.dto.linia.LiniaDtoInput](#)

Pakiety

- package [com.mklachl.sopkom.model.dto.linia](#)

7.74 LiniaDtoInput.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.model.dto.linia;
00002
00003 import com.mklachl.sopkom.model.entity.Linia;
00004 import java.util.ArrayList;
00005 import java.util.List;
00006
00010 public class LiniaDtoInput extends LiniaDto {
00011
00015     private List<PrzystanekDtoDlaLiniaInput> przystanki;
00016
00020     public LiniaDtoInput() {
00021     }
00022
00027     public LiniaDtoInput(Linia linia) {
00028         super(linia);
00029         this.przystanki = new ArrayList<>();
00030
00031         for (int i = 0; i < linia.getPrzystankiWlini().size(); i++) {
00032             PrzystanekDtoDlaLiniaInput dto = new
00033                 PrzystanekDtoDlaLiniaInput(linia.getPrzystankiWlini().get(i));
00034             this.przystanki.add(dto);
00035     }
00036 }
```

```

00034         }
00035     }
00036
00037     // Getters and Setters
00038
00043     public List<PrzystanekDtoDlaLiniaInput> getPrzystanki() {
00044         return przystanki;
00045     }
00046
00051     public void setPrzystanki(List<PrzystanekDtoDlaLiniaInput> przystanki) {
00052         this.przystanki = przystanki;
00053     }
00054 }
```

7.75 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/model/dto/linia/Linia← DtoOutput.java

Komponenty

- class [com.mklachl.sopkom.model.dto.linia.LiniaDtoOutput](#)

Pakiety

- package [com.mklachl.sopkom.model.dto.linia](#)

7.76 LiniaDtoOutput.java

[Idź do dokumentacji tego pliku.](#)

```

00001 package com.mklachl.sopkom.model.dto.linia;
00002
00003 import com.mklachl.sopkom.model.entity.Linia;
00004 import java.util.ArrayList;
00005 import java.util.List;
00006
00010 public class LiniaDtoOutput extends LiniaDto {
00011
00015     protected boolean odwracalna;
00016
00020     private List<PrzystanekDtoDlaLinia> przystanki;
00021
00025     public LiniaDtoOutput() {
00026     }
00027
00032     public LiniaDtoOutput(Linia linia) {
00033         super(linia);
00034         this.przystanki = new ArrayList<>();
00035         this.odwracalna = true;
00036
00037         for (int i = 0; i < linia.getPrzystankiWlini().size(); i++) {
00038             PrzystanekDtoDlaLinia dto = new PrzystanekDtoDlaLinia(linia.getPrzystankiWlini().get(i));
00039             this.przystanki.add(dto);
00040
00041             if (dto.getPrzystanekOdwrotny() == null) {
00042                 this.odwracalna = false;
00043             }
00044         }
00045     }
00046
00047     // Getters and Setters
00048
00053     public boolean isOdwracalna() {
00054         return odwracalna;
00055     }
00056 }
```

```
00061     public void setOdwracalna(boolean odwracalna) {
00062         this.odwracalna = odwracalna;
00063     }
00064
00069     public List<PrzystanekDtoDlaLinia> getPrzystanki() {
00070         return przystanki;
00071     }
00072
00077     public void setPrzystanki(List<PrzystanekDtoDlaLinia> przystanki) {
00078         this.przystanki = przystanki;
00079     }
00080 }
```

7.77 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_ ← Backend/src/main/java/com/mklachl/sopkom/model/dto/linia/ ← PrzystanekDtoDlaLinia.java

Komponenty

- class [com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLinia](#)

Pakiety

- package [com.mklachl.sopkom.model.dto.linia](#)

7.78 PrzystanekDtoDlaLinia.java

Idź do dokumentacji tego pliku.

```
00001 package com.mklachl.sopkom.model.dto.linia;
00002
00003 import com.mklachl.sopkom.model.dto.PrzystanekDtoFull;
00004 import com.mklachl.sopkom.model.entity.PrzystanekWlini;
00005
00009 public class PrzystanekDtoDlaLinia extends PrzystanekDtoFull {
00010
00014     private Short kolejnosc;
00015
00019     private String adres;
00020
00024     private Long przystanekwliniId;
00025
00029     public PrzystanekDtoDlaLinia() {
00030     }
00031
00036     public PrzystanekDtoDlaLinia(PrzystanekWlini przystanekWlini) {
00037         super(przystanekWlini.getPrzystanek());
00038         this.przystanekwliniId = przystanekWlini.getPrzystanekWliniId();
00039         this.kolejnosc = przystanekWlini.getKolejnosc();
00040         this.adres = przystanekWlini.getPrzystanek().getMiasto() + ", " +
przystanekWlini.getPrzystanek().getUlica();
00041     }
00042
00043     // Getters and Setters
00044
00049     public Short getKolejnosc() {
00050         return kolejnosc;
00051     }
00052
00057     public void setKolejnosc(Short kolejnosc) {
00058         this.kolejnosc = kolejnosc;
00059     }
00060
00065     public String getAdres() {
00066         return adres;
00067     }
```

```

00068
00073     public void setAdres(String adres) {
00074         this.adres = adres;
00075     }
00076
00081     public Long getPrzystanekWliniId() {
00082         return przystanekWliniId;
00083     }
00084
00089     public void setPrzystanekWliniId(Long przystanekWliniId) {
00090         this.przystanekWliniId = przystanekWliniId;
00091     }
00092 }
```

7.79 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/model/dto/linia/← PrzystanekDtoDlaLinialInput.java

Komponenty

- class [com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLinialInput](#)

Pakiety

- package [com.mklachl.sopkom.model.dto.linia](#)

7.80 PrzystanekDtoDlaLinialInput.java

[Idź do dokumentacji tego pliku.](#)

```

00001 package com.mklachl.sopkom.model.dto.linia;
00002
00003 import com.mklachl.sopkom.model.entity.PrzystanekWlini;
00004
00008 public class PrzystanekDtoDlaLiniaInput {
00009
00013     private Long przystanekId;
00014
00018     private Short kolejnosc;
00019
00023     public PrzystanekDtoDlaLiniaInput() {
00024     }
00025
00030     public PrzystanekDtoDlaLiniaInput(PrzystanekWlini przystanek) {
00031         this.przystanekId = przystanek.getPrzystanek().getPrzystanekId();
00032         this.kolejnosc = przystanek.getKolejnosc();
00033     }
00034
00035     // Getters and Setters
00036
00041     public Long getPrzystanekId() {
00042         return przystanekId;
00043     }
00044
00049     public void setPrzystanekId(Long przystanekId) {
00050         this.przystanekId = przystanekId;
00051     }
00052
00057     public Short getKolejnosc() {
00058         return kolejnosc;
00059     }
00060
00065     public void setKolejnosc(Short kolejnosc) {
00066         this.kolejnosc = kolejnosc;
00067     }
00068 }
```

7.81 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ ← Backend/src/main/java/com/mklachl/sopkom/model/dto/ ← Message.java

Komponenty

- class [com.mklachl.sopkom.model.dto.Message](#)

Pakiety

- package [com.mklachl.sopkom.model.dto](#)

7.82 Message.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.model.dto;  
00002  
00006 public class Message {  
00007     public String message;  
00011     public Message() {  
00012         }  
00018     public Message(String message) {  
00024         this.message = message;  
00025     }  
00026     public Message SetMessageInt(String message) {  
00033         this.message = message;  
00034         return this;  
00035     }  
00036     public static Message SetMessage(String message) {  
00043         return new Message().SetMessageInt(message);  
00044     }  
00045 }
```

7.83 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ ← Backend/src/main/java/com/mklachl/sopkom/model/dto/przejazd/ ← PKPWLDtoUpdate.java

Komponenty

- class [com.mklachl.sopkom.model.dto.przejazd.PKPWLDtoUpdate](#)

Pakiety

- package [com.mklachl.sopkom.model.dto.przejazd](#)

7.84 PKPWLDtoUpdate.java

[Idź do dokumentacji tego pliku.](#)

```

00001 package com.mklachl.sopkom.model.dto.przejazd;
00002
00003 import com.fasterxml.jackson.annotation.JsonFormat;
00004 import java.util.Date;
00005
00009 public class PKPWLDtoUpdate {
00010
00014     private Long id;
00015
00019     @JsonFormat(pattern = "HH:mm:ss, dd.MM.yyyy", timezone = "UTC")
00020     private Date date;
00021
00025     public PKPWLDtoUpdate() {
00026     }
00027
00028     // Getters and Setters
00029
00034     public Long getId() {
00035         return id;
00036     }
00037
00042     public void setId(Long id) {
00043         this.id = id;
00044     }
00045
00050     public Date getDate() {
00051         return date;
00052     }
00053
00058     public void setDate(Date date) {
00059         this.date = date;
00060     }
00061 }
```

7.85 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/model/dto/przejazd/← PrzejazdDtoInput.java

Komponenty

- class [com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoInput](#)

Pakiety

- package [com.mklachl.sopkom.model.dto.przejazd](#)

7.86 PrzejazdDtoInput.java

[Idź do dokumentacji tego pliku.](#)

```

00001 package com.mklachl.sopkom.model.dto.przejazd;
00002
00003 import java.util.Date;
00004 import com.fasterxml.jackson.annotation.JsonFormat;
00005
00009 public class PrzejazdDtoInput {
00010
00014     private Long kursId;
00015
00019     private Long kierowcaId;
00020 }
```

```
00024     private Long autobusId;
00025
00029     @JsonFormat(pattern = "dd.MM.yyyy")
00030     private Date dataPrzejazdu;
00031
00035     public PrzejazdDtoInput() {}
00036
00037     // Getters and Setters
00038
00043     public Long getKursId() {
00044         return kursId;
00045     }
00046
00051     public void setKursId(Long kursId) {
00052         this.kursId = kursId;
00053     }
00054
00059     public Long getKierowcaId() {
00060         return kierowcaId;
00061     }
00062
00067     public void setKierowcaId(Long kierowcaId) {
00068         this.kierowcaId = kierowcaId;
00069     }
00070
00075     public Long getAutobusId() {
00076         return autobusId;
00077     }
00078
00083     public void setAutobusId(Long autobusId) {
00084         this.autobusId = autobusId;
00085     }
00086
00091     public Date getDataPrzejazdu() {
00092         return dataPrzejazdu;
00093     }
00094
00099     public void setDataPrzejazdu(Date dataPrzejazdu) {
00100         this.dataPrzejazdu = dataPrzejazdu;
00101     }
00102
00103     @Override
00104     public String toString() {
00105         return "PrzejazdDtoInput [kursId=" + kursId + ", kierowcaId=" + kierowcaId + ", autobusId=" +
autobusId
00106                     + ", dataPrzejazdu=" + dataPrzejazdu + "]";
00107     }
00108 }
```

7.87 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_↔ Backend/src/main/java/com/mklachl/sopkom/model/dto/przejazd/↔ PrzejazdDtoOutput.java

Komponenty

- class [com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput](#)

Pakiety

- package [com.mklachl.sopkom.model.dto.przejazd](#)

7.88 PrzejazdDtoOutput.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.model.dto.przejazd;
```

```

00002
00003 import com.fasterxml.jackson.annotation.JsonFormat;
00004 import com.mklachl.sopkom.model.dto.PrzejazdKursPrzystanekWliniDto;
00005 import com.mklachl.sopkom.model.entity.Przejazd;
00006 import java.util.Date;
00007 import java.util.List;
00008 import java.util.stream.Collectors;
00009
0013 public class PrzejazdDtoOutput {
0014
0018     private Long przejazdId;
0019
0023     private Long kursId;
0024
0028     private String liniaNumer;
0029
0033     @JsonFormat(pattern = "EE MMM d y H:m:s ZZZ", locale = "en-US")
0034     private Date dataStartu;
0035
0039     @JsonFormat(pattern = "EE MMM d y H:m:s ZZZ", locale = "en-US")
0040     private Date dataKonca;
0041
0045     @JsonFormat(pattern = "EE MMM d y H:m:s ZZZ", locale = "en-US")
0046     private Date data;
0047
0051     private String kierowcaImieNazwisko;
0052
0056     private String autobusNazwa;
0057
0061     private String autobusNumerRejestracyjny;
0062
0066     private float spalanie;
0067
0071     private float cenaZaLitru;
0072
0076     private float dlugoscTrasy;
0077
0081     private int liczbaBiletow;
0082
0086     private String status;
0087
0091     private short kierunek;
0092
0096     private List<PrzejazdKursPrzystanekWliniDto> przystanki;
0097
00101    public PrzejazdDtoOutput() {}
00102
00107    public PrzejazdDtoOutput(Przejazd przejazd) {
00108        this.przejazdId = przejazd.getPrzejazdId();
00109        this.kursId = przejazd.getKurs().getKursId();
00110        this.liniaNumer = przejazd.getKurs().getLinia().getNumer();
00111        this.dataStartu = przejazd.getDataStartu();
00112        this.dataKonca = przejazd.getDataKonca();
00113        this.kierowcaImieNazwisko = przejazd.getKierowca().getImie() + " " +
przejazd.getKierowca().getNazwisko();
00114        this.autobusNazwa = przejazd.getAutobus().getNumerRejestracyjny();
00115        this.autobusNumerRejestracyjny = przejazd.getAutobus().getNumerRejestracyjny();
00116        this.spalanie = przejazd.getSpalanie();
00117        this.cenaZaLitru = przejazd.getCenaZaLitru();
00118        this.dlugoscTrasy = przejazd.getDlugoscTrasy();
00119        this.liczbaBiletow = przejazd.getPrzejazdBilet().size();
00120        this.kierunek = przejazd.getKurs().getKierunek().shortValue();
00121        Date now = new Date();
00122        this.data = przejazd.getData();
00123        this.przystanki = przejazd.getPrzejazdKursPrzystanekWlini().stream()
00124            .map(PrzejazdKursPrzystanekWliniDto::new).collect(Collectors.toList());
00125        this.status = "Zaplanowany";
00126
00127        int count = 0;
00128
00129        for (PrzejazdKursPrzystanekWliniDto przystanek : przystanki) {
00130            if (przystanek.getRealnaGodzinna() != null) {
00131                count++;
00132                this.status = "W trakcie";
00133            }
00134        }
00135
00136        if (count == przystanki.size()) {
00137            this.status = "Zakończony";
00138        }
00139
00140        if (now.after(przejazd.getDataKonca())) {
00141            if (status.equals("Zaplanowany")) {
00142                this.status = "Nie odbył się";
00143            } else if (status.equals("W trakcie")) {
00144                this.status = "Spóźniony";
00145            }
}

```

```
00146         }
00147     }
00148
00149     public PrzejazdDtoOutput(Przejazd przejazd, boolean a) {
00150         this.przejazdId = przejazd.getPrzejazdId();
00151         this.kursId = przejazd.getKurs().getKursId();
00152         this.liniaNumer = przejazd.getKurs().getLinia().getNumer();
00153         this.dataStartu = przejazd.getDataStartu();
00154         this.dataKonca = przejazd.getDataKonca();
00155         this.kierowcaImieNazwisko = null;
00156         this.autobusNazwa = null;
00157         this.autobusNumerRejestracyjny = null;
00158         this.spalanie = 0;
00159         this.cenaZalitr = 0;
00160         this.dlugoscTrasy = 0;
00161         this.liczbaBiletow = 0;
00162         this.kierunek = przejazd.getKurs().getKierunek().shortValue();
00163         this.data = przejazd.getData();
00164         this.przystanki = przejazd.getPrzejazdKursPrzystanekWlini().stream()
00165             .map(PrzejazdKursPrzystanekWliniDto::new).collect(Collectors.toList());
00166         this.status = "Do zaplanowania";
00167     }
00168
00169     // Getters and Setters
00170
00171     public Long getPrzejazdId() {
00172         return przejazdId;
00173     }
00174
00175     public void setPrzejazdId(Long przejazdId) {
00176         this.przejazdId = przejazdId;
00177     }
00178
00179     public Long getKursId() {
00180         return kursId;
00181     }
00182
00183     public void setKursId(Long kursId) {
00184         this.kursId = kursId;
00185     }
00186
00187     public String getLiniaNumer() {
00188         return liniaNumer;
00189     }
00190
00191     public void setLiniaNumer(String liniaNumer) {
00192         this.liniaNumer = liniaNumer;
00193     }
00194
00195     public Date getDataStartu() {
00196         return dataStartu;
00197     }
00198
00199     public void setDataStartu(Date dataStartu) {
00200         this.dataStartu = dataStartu;
00201     }
00202
00203     public Date getDataKonca() {
00204         return dataKonca;
00205     }
00206
00207     public void setDataKonca(Date dataKonca) {
00208         this.dataKonca = dataKonca;
00209     }
00210
00211     public String getKierowcaImieNazwisko() {
00212         return kierowcaImieNazwisko;
00213     }
00214
00215     public void setKierowcaImieNazwisko(String kierowcaImieNazwisko) {
00216         this.kierowcaImieNazwisko = kierowcaImieNazwisko;
00217     }
00218
00219     public String getAutobusNazwa() {
00220         return autobusNazwa;
00221     }
00222
00223     public void setAutobusNazwa(String autobusNazwa) {
00224         this.autobusNazwa = autobusNazwa;
00225     }
00226
00227     public String getAutobusNumerRejestracyjny() {
00228         return autobusNumerRejestracyjny;
00229     }
00230
00231     public void setAutobusNumerRejestracyjny(String autobusNumerRejestracyjny) {
00232         this.autobusNumerRejestracyjny = autobusNumerRejestracyjny;
```

```
00302     }
00303
00308     public float getSpalanie() {
00309         return spalanie;
00310     }
00311
00316     public void setSpalanie(float spalanie) {
00317         this.spalanie = spalanie;
00318     }
00319
00324     public float getCenaZaLitr() {
00325         return cenaZaLitr;
00326     }
00327
00332     public void setCenaZaLitr(float cenaZaLitr) {
00333         this.cenaZaLitr = cenaZaLitr;
00334     }
00335
00340     public float getDlugoscTrasy() {
00341         return dlugoscTrasy;
00342     }
00343
00348     public void setDlugoscTrasy(float dlugoscTrasy) {
00349         this.dlugoscTrasy = dlugoscTrasy;
00350     }
00351
00356     public int getLiczbaBiletow() {
00357         return liczbaBiletow;
00358     }
00359
00364     public void setLiczbaBiletow(int liczbaBiletow) {
00365         this.liczbaBiletow = liczbaBiletow;
00366     }
00367
00372     public String getStatus() {
00373         return status;
00374     }
00375
00380     public void setStatus(String status) {
00381         this.status = status;
00382     }
00383
00388     public List<PrzejazdKursPrzystanekWliniDto> getPrzystanki() {
00389         return przystanki;
00390     }
00391
00396     public void setPrzystanki(List<PrzejazdKursPrzystanekWliniDto> przystanki) {
00397         this.przystanki = przystanki;
00398     }
00399
00404     public Date getData() {
00405         return data;
00406     }
00407
00412     public void setData(Date data) {
00413         this.data = data;
00414     }
00415
00420     public short getKierunek() {
00421         return kierunek;
00422     }
00423
00428     public void setKierunek(short kierunek) {
00429         this.kierunek = kierunek;
00430     }
00431
00432
00433     /*public String getStatus() {
00434         return status;
00435     }
00436
00437     public void setStatus(String status) {
00438         this.status = status;
00439     }*/
00440 }
```

7.89 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ ← Backend/src/main/java/com/mklachl/sopkom/model/dto/przejazd/← PrzejazdDtoUpdate.java

Komponenty

- class [com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate](#)

Pakiety

- package [com.mklachl.sopkom.model.dto.przejazd](#)

7.90 PrzejazdDtoUpdate.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.model.dto.przejazd;
00002
00003 import java.util.List;
00004
00005 public class PrzejazdDtoUpdate {
00006
00007     private long przejazdId;
00008
00009     private double spalanie;
00010
00011     private double cenaZaLitr;
00012
00013     private double dlugoszTrasy;
00014
00015     private int liczbaBiletowUlgowych;
00016
00017     private int liczbaBiletowNormalnych;
00018
00019     private List<PKPWLDtoUpdate> przystanki;
00020
00021     public PrzejazdDtoUpdate() {}
00022
00023     // Getters and Setters
00024
00025     public long getPrzejazdId() {
00026         return przejazdId;
00027     }
00028
00029     public void setPrzejazdId(long przejazdId) {
00030         this.przejazdId = przejazdId;
00031     }
00032
00033     public double getSpalanie() {
00034         return spalanie;
00035     }
00036
00037     public void setSpalanie(double spalanie) {
00038         this.spalanie = spalanie;
00039     }
00040
00041     public double getCenaZaLitr() {
00042         return cenaZaLitr;
00043     }
00044
00045     public void setCenaZaLitr(double cenaZaLitr) {
00046         this.cenaZaLitr = cenaZaLitr;
00047     }
00048
00049     public double getDlugoszTrasy() {
00050         return dlugoszTrasy;
00051     }
00052
00053     public void setDlugoszTrasy(double dlugoszTrasy) {
00054         this.dlugoszTrasy = dlugoszTrasy;
00055     }
```

```

00114     }
00115
00120     public int getLiczbaBiletowUlgowych() {
00121         return liczbaBiletowUlgowych;
00122     }
00123
00128     public void setLiczbaBiletowUlgowych(int liczbaBiletowUlgowych) {
00129         this.liczbaBiletowUlgowych = liczbaBiletowUlgowych;
00130     }
00131
00136     public int getLiczbaBiletowNormalnych() {
00137         return liczbaBiletowNormalnych;
00138     }
00139
00144     public void setLiczbaBiletowNormalnych(int liczbaBiletowNormalnych) {
00145         this.liczbaBiletowNormalnych = liczbaBiletowNormalnych;
00146     }
00147
00152     public List<PKPWLDtoUpdate> getPrzystanki() {
00153         return przystanki;
00154     }
00155
00160     public void setPrzystanki(List<PKPWLDtoUpdate> przystanki) {
00161         this.przystanki = przystanki;
00162     }
00163
00164     @Override
00165     public String toString() {
00166         return "PrzejazdDtoUpdate [przejazdId=" + przejazdId + ", spalanie=" + spalanie + ",
00167             cenaZaLitru=" + cenaZaLitru
00168             + ", dlugoszTrasy=" + dlugoszTrasy + ", liczbaBiletowUlgowych=" +
00169             liczbaBiletowUlgowych
00170             + ", liczbaBiletowNormalnych=" + liczbaBiletowNormalnych + ", przystanki=" +
00171             przystanki + "]";
00170     }
00170 }

```

7.91 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/model/dto/Przejazd← BiletDto.java

Komponenty

- class [com.mklachl.sopkom.model.dto.PrzejazdBiletDto](#)

Pakiety

- package [com.mklachl.sopkom.model.dto](#)

7.92 PrzejazdBiletDto.java

[Idź do dokumentacji tego pliku.](#)

```

00001 package com.mklachl.sopkom.model.dto;
00002
00003 import com.mklachl.sopkom.model.entity.PrzejazdBilet;
00004
00008 public class PrzejazdBiletDto {
00009
00013     private Long przejazdBiletId;
00014
00018     private Long przejazdId;
00019
00023     private Short biletId;
00024
00028     private float cenaBiletu;

```

```
00029
00033     public PrzejazdBiletDto() {
00034     }
00035
00040     public PrzejazdBiletDto(PrzejazdBilet przejazdBilet) {
00041         this.przejazdBiletId = przejazdBilet.getPrzejazdBiletId();
00042         this.przejazdId = przejazdBilet.getPrzejazd().getPrzejazdId();
00043         this.biletId = przejazdBilet.getBilet().getBiletId();
00044         this.cenaBiletu = przejazdBilet.getCenaBiletu();
00045     }
00046
00047     // Getters and Setters
00048
00053     public Long getPrzejazdBiletId() {
00054         return przejazdBiletId;
00055     }
00056
00061     public void setPrzejazdBiletId(Long przejazdBiletId) {
00062         this.przejazdBiletId = przejazdBiletId;
00063     }
00064
00069     public Long getPrzejazdId() {
00070         return przejazdId;
00071     }
00072
00077     public void setPrzejazdId(Long przejazdId) {
00078         this.przejazdId = przejazdId;
00079     }
00080
00085     public Short getBiletId() {
00086         return biletId;
00087     }
00088
00093     public void setBiletId(Short biletId) {
00094         this.biletId = biletId;
00095     }
00096
00101     public float getCenaBiletu() {
00102         return cenaBiletu;
00103     }
00104
00109     public void setCenaBiletu(float cenaBiletu) {
00110         this.cenaBiletu = cenaBiletu;
00111     }
00112 }
```

7.93 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ ← Backend/src/main/java/com/mklachl/sopkom/model/dto/Przejazd← KursPrzystanekWliniDto.java

Komponenty

- class [com.mklachl.sopkom.model.dto.PrzejazdKursPrzystanekWliniDto](#)

Pakietы

- package [com.mklachl.sopkom.model.dto](#)

7.94 PrzejazdKursPrzystanekWliniDto.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.model.dto;
00002
00003 import com.fasterxml.jackson.annotation.JsonFormat;
00004 import com.mklachl.sopkom.model.entity.PrzejazdKursPrzystanekWlini;
```

```
00005 import java.util.Date;
00006
00010 public class PrzejazdKursPrzystanekWliniDto {
00011
00015     private Long przejazdKursPrzystanekWliniId;
00016
00020     private int kolejnosc;
00021
00025     @JsonFormat(pattern = "dd-MM-yyyy'T'HH:mm:ss -02:00")
00026     private Date przewidywanaGodzinna;
00027
00031     @JsonFormat(pattern = "EE MMM d y H:m:s ZZZ", locale = "en-US")
00032     private Date realnaGodzinna;
00033
00037     private String nazwa;
00038
00042     private Long przystanekId;
00043
00047     public PrzejazdKursPrzystanekWliniDto() {
00048     }
00049
00054     public PrzejazdKursPrzystanekWliniDto(PrzejazdKursPrzystanekWlini przejazdKursPrzystanekWlini) {
00055         this.przejazdKursPrzystanekWliniId =
00056             przejazdKursPrzystanekWlini.getPrzejazdKursPrzystanekWliniId();
00057         this.kolejnosc =
00058             przejazdKursPrzystanekWlini.getKursPrzystanekWlini().getPrzystanekWlini().getKolejnosc();
00059         this.przewidywanaGodzinna =
00060             przejazdKursPrzystanekWlini.getKursPrzystanekWlini().getGodzinna();
00061         this.realnaGodzinna =
00062             przejazdKursPrzystanekWlini.getKursPrzystanekWlini().getRealnaGodzinna();
00063         this.nazwa =
00064             przejazdKursPrzystanekWlini.getKursPrzystanekWlini().getPrzystanekWlini().getNazwa();
00065         this.przystanekId =
00066             przejazdKursPrzystanekWlini.getKursPrzystanekWlini().getPrzystanekWlini().getPrzystanek().getPrzystanekId();
00067     }
00068
00069     // Getters and Setters
00070
00071     public Long getPrzejazdKursPrzystanekWliniId() {
00072         return przejazdKursPrzystanekWliniId;
00073     }
00074
00075     public void setPrzejazdKursPrzystanekWliniId(Long przejazdKursPrzystanekWliniId) {
00076         this.przejazdKursPrzystanekWliniId = przejazdKursPrzystanekWliniId;
00077     }
00078
00079     public int getKolejnosc() {
00080         return kolejnosc;
00081     }
00082
00083     public void setKolejnosc(int kolejnosc) {
00084         this.kolejnosc = kolejnosc;
00085     }
00086
00087     public Date getPrzewidywanaGodzinna() {
00088         return przewidywanaGodzinna;
00089     }
00090
00091     public void setPrzewidywanaGodzinna(Date przewidywanaGodzinna) {
00092         this.przewidywanaGodzinna = przewidywanaGodzinna;
00093     }
00094
00095     public Date getRealnaGodzinna() {
00096         return realnaGodzinna;
00097     }
00098
00099     public void setRealnaGodzinna(Date realnaGodzinna) {
00100         this.realnaGodzinna = realnaGodzinna;
00101     }
00102
00103     public String getNazwa() {
00104         return nazwa;
00105     }
00106
00107     public void setNazwa(String nazwa) {
00108         this.nazwa = nazwa;
00109     }
00110
00111     public Long getPrzystanekId() {
00112         return przystanekId;
00113     }
00114
00115     public void setPrzystanekId(Long przystanekId) {
00116         this.przystanekId = przystanekId;
00117     }
00118 }
```

7.95 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ ← Backend/src/main/java/com/mklachl/sopkom/model/dto/ ← PrzystanekDto.java

Komponenty

- class [com.mklachl.sopkom.model.dto.PrzystanekDto](#)

Pakiety

- package [com.mklachl.sopkom.model.dto](#)

7.96 PrzystanekDto.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.model.dto;
00002
00003 import com.mklachl.sopkom.model.entity.Przystanek;
00004
00008 public class PrzystanekDto {
00009
00013     private Long przystanekId;
00014
00018     private Long przystanekOdwrotnyId;
00019
00023     private String przystanekOdwrotnyNazwa;
00024
00028     private String nazwa;
00029
00033     private String kodPocztowy;
00034
00038     private String miasto;
00039
00043     private String ulica;
00044
00048     private String dlugoscGeograficzna;
00049
00053     private String szerokoscGeograficzna;
00054
00058     public PrzystanekDto() {
00059     }
00060
00065     public PrzystanekDto(Przystanek przystanek) {
00066         this.przystanekId = przystanek.getPrzystanekId();
00067
00068         if (przystanek.getPrzystanekOdwrotny() != null) {
00069             this.przystanekOdwrotnyId = przystanek.getPrzystanekOdwrotny().getPrzystanekId();
00070             this.przystanekOdwrotnyNazwa = przystanek.getPrzystanekOdwrotny().getNazwa();
00071         }
00072
00073         this.nazwa = przystanek.getNazwa();
00074         this.kodPocztowy = przystanek.getKodPocztowy();
00075         this.miasto = przystanek.getMiasto();
00076         this.ulica = przystanek.getUlica();
00077         this.dlugoscGeograficzna = przystanek.getDlugoscGeograficzna();
00078         this.szerokoscGeograficzna = przystanek.getSzerokoscGeograficzna();
00079     }
00080
00081     // Getters
00082
00087     public Long getPrzystanekId() {
00088         return przystanekId;
00089     }
00090
00095     public Long getPrzystanekOdwrotny() {
00096         return przystanekOdwrotnyId;
00097     }
00103     public String getNazwa() {
```

```

00104         return nazwa;
00105     }
00106
00111     public String getKodPocztowy() {
00112         return kodPocztowy;
00113     }
00114
00119     public String getMiasto() {
00120         return miasto;
00121     }
00122
00127     public String getUlica() {
00128         return ulica;
00129     }
00130
00135     public String getDlugoscGeograficzna() {
00136         return dlugoscGeograficzna;
00137     }
00138
00143     public String getSzerokoscGeograficzna() {
00144         return szerokoscGeograficzna;
00145     }
00146
00147     // Setters
00148
00153     public void setPrzystanekId(Long przystanekId) {
00154         this.przystanekId = przystanekId;
00155     }
00156
00161     public void setPrzystanekOdwrotny(Long przystanekOdwrotny) {
00162         this.przystanekOdwrotnyId = przystanekOdwrotny;
00163     }
00164
00169     public String getPrzystanekOdwrotnyNazwa() {
00170         return przystanekOdwrotnyNazwa;
00171     }
00172
00177     public void setPrzystanekOdwrotnyNazwa(String przystanekOdwrotnyNazwa) {
00178         this.przystanekOdwrotnyNazwa = przystanekOdwrotnyNazwa;
00179     }
00180
00185     public void setNazwa(String nazwa) {
00186         this.nazwa = nazwa;
00187     }
00188
00193     public void setKodPocztowy(String kodPocztowy) {
00194         this.kodPocztowy = kodPocztowy;
00195     }
00196
00201     public void setMiasto(String miasto) {
00202         this.miasto = miasto;
00203     }
00204
00209     public void setUlica(String ulica) {
00210         this.ulica = ulica;
00211     }
00212
00217     public void setDlugoscGeograficzna(String dlugoscGeograficzna) {
00218         this.dlugoscGeograficzna = dlugoscGeograficzna;
00219     }
00220
00225     public void setSzerokoscGeograficzna(String szerokoscGeograficzna) {
00226         this.szerokoscGeograficzna = szerokoscGeograficzna;
00227     }
00228 }

```

7.97 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ ← Backend/src/main/java/com/mklachl/sopkom/model/dto/ ← PrzystanekDtoFull.java

Komponenty

- class com.mklachl.sopkom.model.dto.PrzystanekDtoFull

Pakiety

- package com.mklachl.sopkom.model.dto

7.98 PrzystanekDtoFull.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.model.dto;
00002
00003 import com.mklachl.sopkom.model.entity.Przystanek;
00004
00005 public class PrzystanekDtoFull {
00006
00013     protected Long przystanekId;
00014
00018     protected PrzystanekDtoFull przystanekOdwrotny;
00019
00023     protected String nazwa;
00024
00028     protected String kodPocztowy;
00029
00033     protected String miasto;
00034
00038     protected String ulica;
00039
00043     protected String dlugoscGeograficzna;
00044
00048     protected String szerokoscGeograficzna;
00049
00053     public PrzystanekDtoFull() {
00054         }
00055
00060     public PrzystanekDtoFull(Przystanek przystanek) {
00061         this.przystanekId = przystanek.getPrzystanekId();
00062
00063         if (przystanek.getPrzystanekOdwrotny() != null) {
00064             this.przystanekOdwrotny = new PrzystanekDtoFull(przystanek.getPrzystanekOdwrotny());
00065         }
00066
00067         this.nazwa = przystanek.getNazwa();
00068         this.kodPocztowy = przystanek.getKodPocztowy();
00069         this.miasto = przystanek.getMiasto();
00070         this.ulica = przystanek.getUlica();
00071         this.dlugoscGeograficzna = przystanek.getDlugoscGeograficzna();
00072         this.szerokoscGeograficzna = przystanek.getSzerokoscGeograficzna();
00073     }
00074
00075     // Getters
00076
00081     public Long getPrzystanekId() {
00082         return przystanekId;
00083     }
00084
00089     public PrzystanekDtoFull getPrzystanekOdwrotny() {
00090         return przystanekOdwrotny;
00091     }
00092
00097     public String getNazwa() {
00098         return nazwa;
00099     }
00100
00105     public String getKodPocztowy() {
00106         return kodPocztowy;
00107     }
00108
00113     public String getMiasto() {
00114         return miasto;
00115     }
00116
00121     public String getUlica() {
00122         return ulica;
00123     }
00124
00129     public String getDlugoscGeograficzna() {
00130         return dlugoscGeograficzna;
00131     }
00132
00137     public String getSzerokoscGeograficzna() {
00138         return szerokoscGeograficzna;
00139     }
```

```

00140
00141 // Setters
00142
00143     public void setPrzystanekId(Long przystanekId) {
00144         this.przystanekId = przystanekId;
00145     }
00146
00147     public void setPrzystanekOdwrotny(PrzystanekDtoFull przystanekOdwrotny) {
00148         this.przystanekOdwrotny = przystanekOdwrotny;
00149     }
00150
00151     public void setNazwa(String nazwa) {
00152         this.nazwa = nazwa;
00153     }
00154
00155     public void setKodPocztowy(String kodPocztowy) {
00156         this.kodPocztowy = kodPocztowy;
00157     }
00158
00159     public void setMiasto(String miasto) {
00160         this.miasto = miasto;
00161     }
00162
00163     public void setUlica(String ulica) {
00164         this.ulica = ulica;
00165     }
00166
00167     public void setDlugoscGeograficzna(String dlugoscGeograficzna) {
00168         this.dlugoscGeograficzna = dlugoscGeograficzna;
00169     }
00170
00171     public void setSzerokoscGeograficzna(String szerokoscGeograficzna) {
00172         this.szerokoscGeograficzna = szerokoscGeograficzna;
00173     }
00174
00175 }
00176
00177 }
```

7.99 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/model/dto/rozklad/← RozkladDto.java

Komponenty

- class [com.mklachl.sopkom.model.dto.rozklad.RozkladDto](#)
- class [com.mklachl.sopkom.model.dto.rozklad.RozkladDto.\\$](#)

Pakiety

- package [com.mklachl.sopkom.model.dto.rozklad](#)

7.100 RozkladDto.java

[Idź do dokumentacji tego pliku.](#)

```

00001 package com.mklachl.sopkom.model.dto.rozklad;
00002
00003 import java.sql.Time;
00004 import java.util.Date;
00005 import java.util.HashMap;
00006 import java.util.Objects;
00007 import java.util.TreeSet;
00008 import com.fasterxml.jackson.annotation.JsonFormat;
00009 import com.mklachl.sopkom.model.dto.PrzystanekDto;
00010 import com.mklachl.sopkom.model.dto.kurs.KursDto;
00011 import com.mklachl.sopkom.model.dto.linia.LiniaDto;
00012 import com.mklachl.sopkom.model.entity.Przystanek;
00013
```

```
00017 public class RozkladDto {
00018     public static class $ implements Comparable<$> {
00026         public Time time;
00027
00031         public boolean odw;
00032
00038         public $(Time time, boolean odw) {
00039             this.time = time;
00040             this.odw = odw;
00041         }
00042
00043         @Override
00044         public int hashCode() {
00045             return Objects.hash(odw, time);
00046         }
00047
00048         @Override
00049         public boolean equals(Object obj) {
00050             if (this == obj)
00051                 return true;
00052             if (obj == null)
00053                 return false;
00054             if (getClass() != obj.getClass())
00055                 return false;
00056             $ other = ($)
00057             obj;
00058             return odw == other.odw && Objects.equals(time, other.time);
00059         }
00060
00061         @Override
00062         public int compareTo($ o) {
00063             return this.time.compareTo(o.time);
00064         }
00065     }
00066
00069     private PrzystanekDto normalny;
00070
00074     @JsonFormat(pattern = "dd.MM.yyyy", timezone = "Europe/Warsaw")
00075     private Date date;
00076
00080     private HashMap<String, TreeSet<$> rozklad = new HashMap<>();
00081
00088     public RozkladDto(Przystanek p, KursDto kursDto, Date date) {
00089         this.normalny = new PrzystanekDto(p);
00090         this.date = date;
00091     }
00092
00096     public RozkladDto() {
00097     }
00098
00099     // Getters and Setters
00100
00105     public HashMap<String, TreeSet<$> getRozklad() {
00106         return rozklad;
00107     }
00108
00113     public void setRozklad(HashMap<String, TreeSet<$> rozklad) {
00114         this.rozklad = rozklad;
00115     }
00116
00124     public RozkladDto addKurs(LiniaDto linia, Time godzinna, boolean odwrotna) {
00125         rozklad.computeIfAbsent(linia.getNumer(), k -> new TreeSet<>()).add(new $(godzinna,
00126             odwrotna));
00127         return this;
00128     }
00133
00133     public PrzystanekDto getNormalny() {
00134         return normalny;
00135     }
00136
00141     public void setNormalny(PrzystanekDto normalny) {
00142         this.normalny = normalny;
00143     }
00144
00149     public Date getDate() {
00150         return date;
00151     }
00152
00157     public void setDate(Date date) {
00158         this.date = date;
00159     }
00160 }
```

7.101 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ Backend/src/main/java/com/mklachl/sopkom/model/dto/rozklad/ RozkladKursDto.java

Komponenty

- class [com.mklachl.sopkom.model.dto.rozklad.RozkladKursDto](#)

Pakiety

- package [com.mklachl.sopkom.model.dto.rozklad](#)

7.102 RozkladKursDto.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.model.dto.rozklad;  
00002  
00006 public class RozkladKursDto {  
00007  
00008 }
```

7.103 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ Backend/src/main/java/com/mklachl/sopkom/model/dto/rozklad/ RozkladLiniaDto.java

Komponenty

- class [com.mklachl.sopkom.model.dto.rozklad.RozkladLiniaDto](#)

Pakiety

- package [com.mklachl.sopkom.model.dto.rozklad](#)

7.104 RozkladLiniaDto.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.model.dto.rozklad;  
00002  
00006 public class RozkladLiniaDto {  
00007  
00008 }
```

7.105 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ ← Backend/src/main/java/com/mklachl/sopkom/model/dto/User ← Dto.java

Komponenty

- class [com.mklachl.sopkom.model.dto.UserDto](#)

Pakiety

- package [com.mklachl.sopkom.model.dto](#)

7.106 UserDto.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.model.dto;
00002
00003 import com.mklachl.sopkom.model.entity.Role;
00004 import com.mklachl.sopkom.model.entity.User;
00005 import java.util.Set;
00006
00010 public class UserDto {
00011
00015     private Long id;
00016
00020     private String name;
00021
00025     private String email;
00026
00030     private String password;
00031
00035     private Set<Role> roles;
00036
00040     private String role;
00041
00045     public UserDto() {
00046     }
00047
00052     public UserDto(User user) {
00053         this.id = user.getId();
00054         this.name = user.getName();
00055         this.email = user.getLogin();
00056         // this.password = user.getPassword();
00057         this.roles = user.getRoles();
00058
00059         if (this.roles.size() > 0) {
00060             role = this.roles.iterator().next().toString();
00061         }
00062     }
00063
00064     // Getters
00065
00070     public Long getId() {
00071         return id;
00072     }
00073
00078     public void setId(Long id) {
00079         this.id = id;
00080     }
00081
00086     public String getName() {
00087         return name;
00088     }
00089
00094     public void setName(String name) {
00095         this.name = name;
00096     }
00102     public String getEmail() {
```

```

00103     return email;
00104 }
00105
00110     public void setEmail(String email) {
00111         this.email = email;
00112     }
00113
00118     public String getPassword() {
00119         return password;
00120     }
00121
00126     public void setPassword(String password) {
00127         this.password = password;
00128     }
00129
00134     public Set<Role> getRoles() {
00135         return roles;
00136     }
00137
00142     public void setRoles(Set<Role> roles) {
00143         this.roles = roles;
00144     }
00145
00150     public String getRole() {
00151         return role;
00152     }
00153
00158     public void setRole(String role) {
00159         this.role = role;
00160     }
00161 }
```

7.107 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ Backend/src/main/java/com/mklachl/sopkom/model/entity/ Autobus.java

Komponenty

- class [com.mklachl.sopkom.model.entity.Autobus](#)

Pakiety

- package [com.mklachl.sopkom.model.entity](#)

7.108 Autobus.java

[Idź do dokumentacji tego pliku.](#)

```

00001 package com.mklachl.sopkom.model.entity;
00002
00003 import java.util.Date;
00004 import jakarta.persistence.Entity;
00005 import jakarta.persistence.GeneratedValue;
00006 import jakarta.persistence.GenerationType;
00007 import jakarta.persistence.Id;
00008 import jakarta.persistence.Table;
00009 import jakarta.persistence.Temporal;
00010 import jakarta.persistence.TemporalType;
00011 import jakarta.persistence.Column;
00012
00016 @Entity
00017 @Table(name = "autobusy")
00018 public class Autobus {
00019
00023     @Id
00024     @GeneratedValue(strategy = GenerationType.IDENTITY)
```

```
00025     @Column(name = "autbous_id")
00026     private Long autbousId;
00027
00031     @Column(name = "numer_rejestracyjny", nullable = false, length = 10)
00032     private String numerRejestracyjny;
00033
00037     @Column(name = "przegladwaznydo", nullable = false)
00038     @Temporal(TemporalType.DATE)
00039     private Date przegladWaznyDo;
00040
00044     @Column(name = "status", nullable = false, length = 20)
00045     private String status;
00046
00050     @Column(name = "przebieg", nullable = false)
00051     private float przebieg;
00052
00053     // Getters
00054
00059     public Long getAutbousID() {
00060         return autbousId;
00061     }
00062
00067     public String getNumerRejestracyjny() {
00068         return numerRejestracyjny;
00069     }
00070
00075     public Date getPrzegladWaznyDo() {
00076         return przegladWaznyDo;
00077     }
00078
00083     public String getStatus() {
00084         return status;
00085     }
00086
00091     public float getPrzebieg() {
00092         return przebieg;
00093     }
00094
00095     // Setters
00096
00101     public void setAutbousId(Long autbousId) {
00102         this.autbousId = autbousId;
00103     }
00104
00109     public void setNumerRejestracyjny(String numerRejestracyjny) {
00110         this.numerRejestracyjny = numerRejestracyjny;
00111     }
00112
00117     public void setPrzegladWaznyDo(Date przegladWaznyDo) {
00118         this.przegladWaznyDo = przegladWaznyDo;
00119     }
00120
00125     public void setStatus(String status) {
00126         this.status = status;
00127     }
00128
00133     public void setPrzebieg(float przebieg) {
00134         this.przebieg = przebieg;
00135     }
00136 }
```

7.109 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ ← Backend/src/main/java/com/mklachl/sopkom/model/entity/ ← Bilet.java

Komponenty

- class [com.mklachl.sopkom.model.entity.Bilet](#)

Pakietы

- package [com.mklachl.sopkom.model.entity](#)

7.110 Bilet.java

[Idź do dokumentacji tego pliku.](#)

```

00001 package com.mklachl.sopkom.model.entity;
00002
00003 import jakarta.persistence.*;
00004
00005 import java.util.List;
00006
00010 @Entity
00011 @Table(name = "bilety")
00012 public class Bilet {
00013
00017     @Id
00018     @GeneratedValue(strategy = GenerationType.IDENTITY)
00019     @Column(name = "bilet_id")
00020     private Short biletId;
00021
00025     @Column(name = "cena", nullable = false)
00026     private float cena;
00027
00031     @Column(name = "nazwa", nullable = false, length = 30)
00032     private String nazwa;
00033
00037     @OneToMany(mappedBy = "bilet")
00038     private List<PrzejazdBilet> PrzejazdBilet;
00039
00040     // Getters and Setters
00041
00046     public Short getBiletId() {
00047         return biletId;
00048     }
00049
00054     public void setBiletId(Short biletId) {
00055         this.biletId = biletId;
00056     }
00057
00062     public float getCena() {
00063         return cena;
00064     }
00065
00070     public void setCena(float cena) {
00071         this.cena = cena;
00072     }
00073
00078     public String getNazwa() {
00079         return nazwa;
00080     }
00081
00086     public void setNazwa(String nazwa) {
00087         this.nazwa = nazwa;
00088     }
00089
00094     public List<PrzejazdBilet> getPrzejazdBilet() {
00095         return PrzejazdBilet;
00096     }
00097
00102     public void setPrzejazdBilet(List<PrzejazdBilet> PrzejazdBilet) {
00103         this.PrzejazdBilet = PrzejazdBilet;
00104     }
00105 }
```

7.111 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/model/entity/← Harmonogram.java

Komponenty

- class [com.mklachl.sopkom.model.entity.Harmonogram](#)

Pakiety

- package com.mklachl.sopkom.model.entity

7.112 Harmonogram.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.model.entity;
00002
00003 import jakarta.persistence.Entity;
00004 import jakarta.persistence.GeneratedValue;
00005 import jakarta.persistence.GenerationType;
00006 import jakarta.persistence.Id;
00007 import jakarta.persistence.Table;
00008 import jakarta.persistence.Column;
00009
00010 @Entity
00011 @Table(name = "harmonogramy")
00012 public class Harmonogram {
00013
00014     @Id
00015     @GeneratedValue(strategy = GenerationType.IDENTITY)
00016     @Column(name = "hamonogram_id")
00017     private Short hamonogramId;
00018
00019     @Column(name = "nazwa", length = 50)
00020     private String nazwa;
00021
00022     @Column(name = "pon", nullable = false)
00023     private boolean pon;
00024
00025     @Column(name = "wto", nullable = false)
00026     private boolean wto;
00027
00028     @Column(name = "sro", nullable = false)
00029     private boolean sro;
00030
00031     @Column(name = "czw", nullable = false)
00032     private boolean czw;
00033
00034     @Column(name = "pia", nullable = false)
00035     private boolean pia;
00036
00037     @Column(name = "sob", nullable = false)
00038     private boolean sob;
00039
00040     @Column(name = "nie", nullable = false)
00041     private boolean nie;
00042
00043     @Column(name = "dodatkowe_inf", length = 200)
00044     private String dodatkoweInf;
00045
00046     public Short getHamonogramId() {
00047         return hamonogramId;
00048     }
00049
00050     public void setHamonogramId(Short hamonogramId) {
00051         this.hamonogramId = hamonogramId;
00052     }
00053
00054     public String getNazwa() {
00055         return nazwa;
00056     }
00057
00058     public void setNazwa(String nazwa) {
00059         this.nazwa = nazwa;
00060     }
00061
00062     public boolean isPon() {
00063         return pon;
00064     }
00065
00066     public void setPon(boolean pon) {
00067         this.pon = pon;
00068     }
00069
00070     public boolean isWto() {
00071         return wto;
00072     }
00073
00074     public void setWto(boolean wto) {
00075         this.wto = wto;
00076     }
00077
00078     public boolean isSro() {
00079         return sro;
00080     }
00081
00082     public void setSro(boolean sro) {
00083         this.sro = sro;
00084     }
00085
00086     public boolean isCzw() {
00087         return czw;
00088     }
00089
00090     public void setCzw(boolean czw) {
00091         this.czw = czw;
00092     }
00093
00094     public boolean isPia() {
00095         return pia;
00096     }
00097
00098     public void setPia(boolean pia) {
00099         this.pia = pia;
00100     }
00101
00102     public boolean isSob() {
00103         return sob;
00104     }
00105
00106     public void setSob(boolean sob) {
00107         this.sob = sob;
00108     }
00109
00110     public boolean isNie() {
00111         return nie;
00112     }
00113
00114     public void setNie(boolean nie) {
00115         this.nie = nie;
00116     }
00117
00118     public String getDodatkoweInf() {
00119         return dodatkoweInf;
00120     }
00121
00122     public void setDodatkoweInf(String dodatkoweInf) {
00123         this.dodatkoweInf = dodatkoweInf;
00124     }
00125
00126     public String getNazwy() {
00127         return nazwy;
00128     }
00129
00130     public void setNazwy(String nazwy) {
00131         this.nazwy = nazwy;
00132     }
00133
00134 }
```

```

00139     public void setWto(boolean wto) {
00140         this.wto = wto;
00141     }
00142
00147     public boolean isSro() {
00148         return sro;
00149     }
00150
00155     public void setSro(boolean sro) {
00156         this.sro = sro;
00157     }
00158
00163     public boolean isCzw() {
00164         return czw;
00165     }
00166
00171     public void setCzw(boolean czw) {
00172         this.czv = czw;
00173     }
00174
00179     public boolean isPia() {
00180         return pia;
00181     }
00182
00187     public void setPia(boolean pia) {
00188         this.pia = pia;
00189     }
00190
00195     public boolean isSob() {
00196         return sob;
00197     }
00198
00203     public void setSob(boolean sob) {
00204         this.sob = sob;
00205     }
00206
00211     public boolean isNie() {
00212         return nie;
00213     }
00214
00219     public void setNie(boolean nie) {
00220         this.nie = nie;
00221     }
00222
00227     public String getDodatkoweInf() {
00228         return dodatkoweInf;
00229     }
00230
00235     public void setDodatkoweInf(String dodatkoweInf) {
00236         this.dodatkoweInf = dodatkoweInf;
00237     }
00238 }

```

7.113 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_[←](#) Backend/src/main/java/com/mklachl/sopkom/model/entity/[←](#) Incydent.java

Komponenty

- class [com.mklachl.sopkom.model.entity.Incydent](#)

Pakiety

- package [com.mklachl.sopkom.model.entity](#)

7.114 Incydent.java

Idź do dokumentacji tego pliku.

```
00001 package com.mklachl.sopkom.model.entity;
00002
00003 import jakarta.persistence.*;
00004 import java.util.Date;
00005
00006 @Entity
00007 @Table(name = "incydenty")
00008 public class Incydent {
00009
00010     @Id
00011     @GeneratedValue(strategy = GenerationType.IDENTITY)
00012     @Column(name = "incydent_id")
00013     private Long incydentId;
00014
00015
00016     @Column(name = "typ", nullable = false, length = 50)
00017     private String typ;
00018
00019     @Column(name = "date", nullable = false)
00020     @Temporal(TemporalType.DATE)
00021     private Date date;
00022
00023     @ManyToOne
00024     @JoinColumn(name = "autobus_id", referencedColumnName = "autbous_id")
00025     private Autobus autobus;
00026
00027     @ManyToOne
00028     @JoinColumn(name = "kierowca_id", referencedColumnName = "kierowca_id")
00029     private Kierowca kierowca;
00030
00031     @ManyToOne
00032     @JoinColumn(name = "przejazd_id", referencedColumnName = "przejazd_id")
00033     private Przejazd przejazd;
00034
00035     @Column(name = "krotko", length = 60)
00036     private String krotko;
00037
00038     @Column(name = "dodatkowe_informacje", length = 800)
00039     private String dodatkoweInformacje;
00040
00041     @Column(name = "koszty")
00042     private float koszty;
00043
00044     public Long getIncydentId() {
00045         return incydentId;
00046     }
00047
00048     public void setIncydentId(Long incydentId) {
00049         this.incydentId = incydentId;
00050     }
00051
00052     public String getTyp() {
00053         return typ;
00054     }
00055
00056     public void setTyp(String typ) {
00057         this.typ = typ;
00058     }
00059
00060     public Date getDate() {
00061         return date;
00062     }
00063
00064     public void setDate(Date date) {
00065         this.date = date;
00066     }
00067
00068     public Autobus getAutobus() {
00069         return autobus;
00070     }
00071
00072     public void setAutobus(Autobus autobus) {
00073         this.autobus = autobus;
00074     }
00075
00076     public Kierowca getKierowca() {
00077         return kierowca;
00078     }
00079
00080     public void setKierowca(Kierowca kierowca) {
00081         this.kierowca = kierowca;
00082     }
00083
00084 }
```

```

00157     public Przejazd getPrzejazd() {
00158         return przejazd;
00159     }
00160
00165     public void setPrzejazd(Przejazd przejazd) {
00166         this.przejazd = przejazd;
00167     }
00168
00173     public String getKrotko() {
00174         return krotko;
00175     }
00176
00181     public void setKrotko(String krotko) {
00182         this.krotko = krotko;
00183     }
00184
00189     public String getDodatkoweInformacje() {
00190         return dodatkoweInformacje;
00191     }
00192
00197     public void setDodatkoweInformacje(String dodatkoweInformacje) {
00198         this.dodatkoweInformacje = dodatkoweInformacje;
00199     }
00200
00205     public float getKoszty() {
00206         return koszty;
00207     }
00208
00213     public void setKoszty(float koszty) {
00214         this.koszty = koszty;
00215     }
00216 }

```

7.115 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_↪ Backend/src/main/java/com/mklachl/sopkom/model/entity/↪ Kierowca.java

Komponenty

- class [com.mklachl.sopkom.model.entity.Kierowca](#)

Pakiety

- package [com.mklachl.sopkom.model.entity](#)

7.116 Kierowca.java

[Idź do dokumentacji tego pliku.](#)

```

00001 package com.mklachl.sopkom.model.entity;
00002
00003 import java.util.Date;
00004 import java.util.Objects;
00005 import jakarta.persistence.Entity;
00006 import jakarta.persistence.GeneratedValue;
00007 import jakarta.persistence.GenerationType;
00008 import jakarta.persistence.Id;
00009 import jakarta.persistence.Table;
00010 import jakarta.persistence.Temporal;
00011 import jakarta.persistence.TemporalType;
00012 import jakarta.persistence.Column;
00013 import jakarta.persistence.ManyToOne;
00014 import jakarta.persistence.JoinColumn;
00015
00019 @Entity
00020 @Table(name = "kierowcy")

```

```
00021 public class Kierowca {
00022     @Id
00023     @GeneratedValue(strategy = GenerationType.IDENTITY)
00024     @Column(name = "kierowca_id")
00025     private Long kierowcaId;
00026
00027     @Column(name = "prawojazdywaznedo", nullable = false)
00028     @Temporal(TemporalType.DATE)
00029     private Date prawojazdyWazneDo;
00030
00031     @Column(name = "imie", nullable = false, length = 30)
00032     private String imie;
00033
00034     @Column(name = "nazwisko", nullable = false, length = 30)
00035     private String nazwisko;
00036
00037     @Column(name = "pesel", nullable = false, length = 30)
00038     private String pesel;
00039
00040     @ManyToOne
00041     @JoinColumn(name = "user_id", referencedColumnName = "id")
00042     private User user;
00043
00044     public Long getKierowcaId() {
00045         return kierowcaId;
00046     }
00047
00048     public void setKierowcaId(Long kierowcaId) {
00049         this.kierowcaId = kierowcaId;
00050     }
00051
00052     public Date getPrawojazdyWazneDo() {
00053         return prawojazdyWazneDo;
00054     }
00055
00056     public void setPrawojazdyWazneDo(Date prawojazdyWazneDo) {
00057         this.prawojazdyWazneDo = prawojazdyWazneDo;
00058     }
00059
00060     public String getImie() {
00061         return imie;
00062     }
00063
00064     public void setImie(String imie) {
00065         this.imie = imie;
00066     }
00067
00068     public String getNazwisko() {
00069         return nazwisko;
00070     }
00071
00072     public void setNazwisko(String nazwisko) {
00073         this.nazwisko = nazwisko;
00074     }
00075
00076     public String getPesel() {
00077         return pesel;
00078     }
00079
00080     public void setPesel(String pesel) {
00081         this.pesel = pesel;
00082     }
00083
00084     public User getUser() {
00085         return user;
00086     }
00087
00088     public void setUser(User user) {
00089         this.user = user;
00090     }
00091
00092     @Override
00093     public int hashCode() {
00094         return Objects.hash(imie, kierowcaId, nazwisko, pesel, prawojazdyWazneDo, user);
00095     }
00096
00097     @Override
00098     public boolean equals(Object obj) {
00099         if (this == obj)
00100             return true;
00101         if (obj == null)
00102             return false;
00103         if (getClass() != obj.getClass())
00104             return false;
00105         Kierowca other = (Kierowca) obj;
00106         return Objects.equals(imie, other.imie) && Objects.equals(kierowcaId, other.kierowcaId)
```

```

00183             && Objects.equals(nazwisko, other.nazwisko) && Objects.equals(pesel, other.pesel)
00184             && Objects.equals(prawoJazdyWazneDo, other.prawoJazdyWazneDo);
00185     }
00186 }
```

7.117 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ ← Backend/src/main/java/com/mklachl/sopkom/model/entity/ ← Kurs.java

Komponenty

- class [com.mklachl.sopkom.model.entity.Kurs](#)

Pakietы

- package [com.mklachl.sopkom.model.entity](#)

7.118 Kurs.java

[Idź do dokumentacji tego pliku.](#)

```

00001 package com.mklachl.sopkom.model.entity;
00002
00003 import jakarta.persistence.*;
00004 import java.util.List;
00005
00009 @Entity
00010 @Table(name = "kursy")
00011 public class Kurs {
00012
00016     @Id
00017     @GeneratedValue(strategy = GenerationType.IDENTITY)
00018     @Column(name = "kurs_id")
00019     private Long kursId;
00020
00024     @ManyToOne
00025     @JoinColumn(name = "linia_id", referencedColumnName = "linia_id")
00026     private Linia linia;
00027
00031     @ManyToOne
00032     @JoinColumn(name = "harmonogram_id", referencedColumnName = "hamonogram_id")
00033     private Harmonogram harmonogram;
00034
00038     @Column(name = "kierunek", nullable = false)
00039     private Short kierunek;
00040
00044     @Column(name = "typ_autobusu", nullable = false)
00045     private Short typAutobusu;
00046
00050     @OneToMany(mappedBy = "kurs")
00051     private List<KursPrzystanekWlinii> kursPrzystanekWlinii;
00052
00057     public Long getKursId() {
00058         return kursId;
00059     }
00060
00065     public void setKursId(Long kursId) {
00066         this.kursId = kursId;
00067     }
00068
00073     public Linia getLinia() {
00074         return linia;
00075     }
00076
00081     public void setLinia(Linia linia) {
00082         this.linia = linia;
```

```
00083     }
00084
00085     public Harmonogram getHarmonogram() {
00086         return harmonogram;
00087     }
00088
00089     public void setHarmonogram(Harmonogram harmonogram) {
00090         this.harmonogram = harmonogram;
00091     }
00092
00093     public Short getKierunek() {
00094         return kierunek;
00095     }
00096
00097     public void setKierunek(Short kierunek) {
00098         this.kierunek = kierunek;
00099     }
00100
00101     public Short getTypAutobusu() {
00102         return typAutobusu;
00103     }
00104
00105     public void setTypAutobusu(Short typAutobusu) {
00106         this.typAutobusu = typAutobusu;
00107     }
00108
00109     public List<KursPrzystanekWlini> getKursPrzystanekWlinii() {
00110         return kursPrzystanekWlinii;
00111     }
00112
00113     public void setKursPrzystanekWlinii(List<KursPrzystanekWlini> kursPrzystanekWlinii) {
00114         this.kursPrzystanekWlinii = kursPrzystanekWlinii;
00115     }
00116
00117 }
```

7.119 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ ← Backend/src/main/java/com/mklachl/sopkom/model/entity/KursPrzystanekWlini.java

Komponenty

- class [com.mklachl.sopkom.model.entity.KursPrzystanekWlini](#)

Pakiety

- package [com.mklachl.sopkom.model.entity](#)

7.120 KursPrzystanekWlini.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.model.entity;
00002
00003 import jakarta.persistence.*;
00004 import java.util.Date;
00005
00006 @Entity
00007 @Table(name = "kursprzystanekwlini")
00008 public class KursPrzystanekWlini {
00009
00010     @Id
00011     @GeneratedValue(strategy = GenerationType.IDENTITY)
00012     @Column(name = "kursprzystankiwlini_id")
00013     private Long kursPrzystanekWliniId;
00014
00015     @ManyToOne
```

```

00025     @JoinColumn(name = "kurs_id", referencedColumnName = "kurs_id")
00026     private Kurs kurs;
00027
00028     @ManyToOne
00029     @JoinColumn(name = "przystanekwlini_id", referencedColumnName = "przystanekwlini_id")
00030     private PrzystanekWlini przystanekWlini;
00031
00032     @Column(name = "godzinna", nullable = false)
00033     @Temporal(TemporalType.TIME)
00034     private Date godzinna;
00035
00036     public Long getKursPrzystanekWliniId() {
00037         return kursPrzystanekWliniId;
00038     }
00039
00040     public void setKursPrzystanekWliniId(Long kursPrzystanekWliniId) {
00041         this.kursPrzystanekWliniId = kursPrzystanekWliniId;
00042     }
00043
00044     public Kurs getKurs() {
00045         return kurs;
00046     }
00047
00048     public void setKurs(Kurs kurs) {
00049         this.kurs = kurs;
00050     }
00051
00052     public PrzystanekWlini getPrzystanekWlini() {
00053         return przystanekWlini;
00054     }
00055
00056     public void setPrzystanekWlini(PrzystanekWlini przystanekWlini) {
00057         this.przystanekWlini = przystanekWlini;
00058     }
00059
00060     public Date getGodzinna() {
00061         return godzinna;
00062     }
00063
00064     public void setGodzinna(Date godzinna) {
00065         this.godzinna = godzinna;
00066     }
00067
00068 }

```

7.121 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/model/entity/← Linia.java

Komponenty

- class [com.mklachl.sopkom.model.entity.Linia](#)

Pakiety

- package [com.mklachl.sopkom.model.entity](#)

7.122 Linia.java

[Idź do dokumentacji tego pliku.](#)

```

00001 package com.mklachl.sopkom.model.entity;
00002
00003 import jakarta.persistence.*;
00004 import java.util.List;
00005
00006 @Entity

```

```
00010 @Table(name = "linie")
00011 public class Linia {
00012     @Id
00013     @GeneratedValue(strategy = GenerationType.IDENTITY)
00014     @Column(name = "linia_id")
00015     private Long liniaId;
00016
00017     @Column(name = "numer", nullable = false, length = 50)
00018     private String numer;
00019
00020     @OneToMany(mappedBy = "linia")
00021     private List<PrzystanekWlini> przystankiWlini;
00022
00023     @Transient
00024     public boolean odwracalna() {
00025         for (PrzystanekWlini przystanekWlini : this.przystankiWlini) {
00026             Przystanek przystanek = przystanekWlini.getPrzystanek();
00027             if (przystanek.getPrzystanekOdwrotny() == null) {
00028                 return false;
00029             }
00030         }
00031         return true;
00032     }
00033
00034     public Long getLiniaId() {
00035         return liniaId;
00036     }
00037
00038     public void setLiniaId(Long liniaId) {
00039         this.liniaId = liniaId;
00040     }
00041
00042     public String getNumer() {
00043         return numer;
00044     }
00045
00046     public void setNumer(String numer) {
00047         this.numer = numer;
00048     }
00049
00050     public List<PrzystanekWlini> getPrzystankiWlini() {
00051         return przystankiWlini;
00052     }
00053
00054     public void setPrzystankiWlini(List<PrzystanekWlini> przystankiWlini) {
00055         this.przystankiWlini = przystankiWlini;
00056     }
00057 }
```

7.123 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/model/entity/← Przejazd.java

Komponenty

- class [com.mklachl.sopkom.model.entity.Przejazd](#)

Pakiety

- package [com.mklachl.sopkom.model.entity](#)

7.124 Przejazd.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.model.entity;
00002
00003 import jakarta.persistence.*;
00004 import java.util.Date;
00005 import java.util.List;
00006
00010 @Entity
00011 @Table(name = "przejazdy")
00012 public class Przejazd {
00013
00017     @Id
00018     @GeneratedValue(strategy = GenerationType.IDENTITY)
00019     @Column(name = "przejazd_id")
00020     private Long przejazdId;
00021
00025     @ManyToOne
00026     @JoinColumn(name = "kierowca_id", referencedColumnName = "kierowca_id")
00027     private Kierowca kierowca;
00028
00032     @ManyToOne
00033     @JoinColumn(name = "autobus_id", referencedColumnName = "autobus_id")
00034     private Autobus autobus;
00035
00039     @ManyToOne
00040     @JoinColumn(name = "kurs_id", referencedColumnName = "kurs_id")
00041     private Kurs kurs;
00042
00046     @Column(name = "spalanie", nullable = false)
00047     private float spalanie;
00048
00052     @Column(name = "cena_za_litr", nullable = false)
00053     private float cenaZaLitr;
00054
00058     @Column(name = "dlugosc_trasy", nullable = false)
00059     private float dlugoscTrasy;
00060
00064     @Column(name = "data_startu", nullable = true)
00065     private Date dataStartu;
00066
00070     @Column(name = "data_konca", nullable = true)
00071     private Date dataKonca;
00072
00076     @Column(name = "data", nullable = true)
00077     @Temporal(TemporalType.DATE)
00078     private Date data;
00079
00083     @OneToMany(mappedBy = "przejazd")
00084     private List<PrzejazdKursPrzystanekWlini> przejazdKursPrzystanekWlini;
00085
00089     @OneToMany(mappedBy = "przejazd")
00090     private List<PrzejazdBilet> PrzejazdBilet;
00091
00096     public Long getPrzejazdId() {
00097         return przejazdId;
00098     }
00099
00104     public void setPrzejazdId(Long przejazdId) {
00105         this.przejazdId = przejazdId;
00106     }
00107
00112     public Kierowca getKierowca() {
00113         return kierowca;
00114     }
00115
00120     public void setKierowca(Kierowca kierowca) {
00121         this.kierowca = kierowca;
00122     }
00123
00128     public Autobus getAutobus() {
00129         return autobus;
00130     }
00131
00136     public void setAutobus(Autobus autobus) {
00137         this.autobus = autobus;
00138     }
00139
00144     public Kurs getKurs() {
00145         return kurs;
00146     }
00147
00152     public void setKurs(Kurs kurs) {
00153         this.kurs = kurs;
00154     }
00155
00160     public float getSpalanie() {
00161         return spalanie;
00162     }
```

```
00163
00168     public void setSpalanie(float spalanie) {
00169         this.spalanie = spalanie;
00170     }
00171
00176     public float getCenaZaLitru() {
00177         return cenaZaLitru;
00178     }
00179
00184     public void setCenaZaLitru(float cenaZaLitru) {
00185         this.cenaZaLitru = cenaZaLitru;
00186     }
00187
00192     public float getDlugoscTrasy() {
00193         return dlugoscTrasy;
00194     }
00195
00200     public void setDlugoscTrasy(float dlugoscTrasy) {
00201         this.dlugoscTrasy = dlugoscTrasy;
00202     }
00203
00208     public Date getDataStartu() {
00209         return dataStartu;
00210     }
00211
00216     public void setDataStartu(Date dataStartu) {
00217         this.dataStartu = dataStartu;
00218     }
00219
00224     public Date getDataKonca() {
00225         return dataKonca;
00226     }
00227
00232     public void setDataKonca(Date dataKonca) {
00233         this.dataKonca = dataKonca;
00234     }
00235
00240     public Date getData() {
00241         return data;
00242     }
00243
00248     public void setData(Date data) {
00249         this.data = data;
00250     }
00251
00256     public List<PrzejazdKursPrzystanekWlini> getPrzejazdKursPrzystanekWlini() {
00257         return przejazdKursPrzystanekWlini;
00258     }
00259
00264     public void setPrzejazdKursPrzystanekWlini(List<PrzejazdKursPrzystanekWlini>
00265         przejazdKursPrzystanekWlini) {
00266         this.przejazdKursPrzystanekWlini = przejazdKursPrzystanekWlini;
00267     }
00268
00272     public List<PrzejazdBilet> getPrzejazdBilet() {
00273         return PrzejazdBilet;
00274     }
00275
00280     public void setPrzejazdBilet(List<PrzejazdBilet> PrzejazdBilet) {
00281         this.PrzejazdBilet = PrzejazdBilet;
00282     }
00283 }
```

7.125 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ ← Backend/src/main/java/com/mklachl/sopkom/model/entity/← PrzejazdBilet.java

Komponenty

- class [com.mklachl.sopkom.model.entity.PrzejazdBilet](#)

Pakiety

- package [com.mklachl.sopkom.model.entity](#)

7.126 PrzejazdBilet.java

[Idź do dokumentacji tego pliku.](#)

```

00001 package com.mklachl.sopkom.model.entity;
00002
00003 import jakarta.persistence.*;
00004
00008 @Entity
00009 @Table(name = "przejazdbilet")
00010 public class PrzejazdBilet {
00011
00015     @Id
00016     @GeneratedValue(strategy = GenerationType.IDENTITY)
00017     @Column(name = "przejazdbilet_id")
00018     private Long przejazdBiletId;
00019
00023     @ManyToOne
00024     @JoinColumn(name = "przejazd_id", referencedColumnName = "przejazd_id")
00025     private Przejazd przejazd;
00026
00030     @ManyToOne
00031     @JoinColumn(name = "bilet_id", referencedColumnName = "bilet_id")
00032     private Bilet bilet;
00033
00037     @Column(name = "cena_biletu", nullable = false)
00038     private float cenaBiletu;
00039
00044     public Long getPrzejazdBiletId() {
00045         return przejazdBiletId;
00046     }
00047
00052     public void setPrzejazdBiletId(Long przejazdBiletId) {
00053         this.przejazdBiletId = przejazdBiletId;
00054     }
00055
00060     public Przejazd getPrzejazd() {
00061         return przejazd;
00062     }
00063
00068     public void setPrzejazd(Przejazd przejazd) {
00069         this.przejazd = przejazd;
00070     }
00076     public Bilet.getBilet() {
00077         return bilet;
00078     }
00079
00084     public void setBilet(Bilet bilet) {
00085         this.bilet = bilet;
00086     }
00087
00092     public float getCenaBiletu() {
00093         return cenaBiletu;
00094     }
00095
00100     public void setCenaBiletu(float cenaBiletu) {
00101         this.cenaBiletu = cenaBiletu;
00102     }
00103 }
```

7.127 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/model/entity/← PrzejazdKursPrzystanekWlini.java

Komponenty

- class [com.mklachl.sopkom.model.entity.PrzejazdKursPrzystanekWlini](#)

Pakiety

- package [com.mklachl.sopkom.model.entity](#)

7.128 PrzejazdKursPrzystanekWlini.java

[Idź do dokumentacji tego pliku.](#)

```

00001 package com.mklachl.sopkom.model.entity;
00002
00003 import jakarta.persistence.*;
00004 import java.util.Date;
00005
00009 @Entity
00010 @Table(name = "przejazdkursprzystanekwlini")
00011 public class PrzejazdKursPrzystanekWlini {
00012
00016     @Id
00017     @GeneratedValue(strategy = GenerationType.IDENTITY)
00018     @Column(name = "przejazdkursprzystanekwlini_id")
00019     private Long przejazdKursPrzystanekWliniId;
00020
00024     @ManyToOne
00025     @JoinColumn(name = "przejazd_id", referencedColumnName = "przejazd_id")
00026     private Przejazd przejazd;
00027
00028     @ManyToOne
00029     @JoinColumn(name = "kursprzystanekwlini_id", referencedColumnName = "kursprzystankiwlini_id")
00030     private KursPrzystanekWlini kursPrzystanekWlini;
00034
00038     @Column(name = "realnagodzinna")
00039     private Date realnaGodzinna;
00040
00045     public Long getPrzejazdKursPrzystanekWliniId() {
00046         return przejazdKursPrzystanekWliniId;
00047     }
00048
00053     public void setPrzejazdKursPrzystanekWliniId(Long przejazdKursPrzystanekWliniId) {
00054         this.przejazdKursPrzystanekWliniId = przejazdKursPrzystanekWliniId;
00055     }
00056
00061     public Przejazd getPrzejazd() {
00062         return przejazd;
00063     }
00064
00069     public void setPrzejazd(Przejazd przejazd) {
00070         this.przejazd = przejazd;
00071     }
00072
00077     public KursPrzystanekWlini getKursPrzystanekWlini() {
00078         return kursPrzystanekWlini;
00079     }
00080
00085     public void setKursPrzystanekWlini(KursPrzystanekWlini kursPrzystanekWlini) {
00086         this.kursPrzystanekWlini = kursPrzystanekWlini;
00087     }
00088
00093     public Date getRealnaGodzinna() {
00094         return realnaGodzinna;
00095     }
00096
00101     public void setRealnaGodzinna(Date realnaGodzinna) {
00102         this.realnaGodzinna = realnaGodzinna;
00103     }
00104 }
```

7.129 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/model/entity/← Przystanek.java

Komponenty

- class [com.mklachl.sopkom.model.entity.Przystanek](#)

Pakiety

- package [com.mklachl.sopkom.model.entity](#)

7.130 Przystanek.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.model.entity;
00002
00003 import jakarta.persistence.*;
00004 import java.util.List;
00005
00006 @Entity
00007 @Table(name = "przystanki")
00008 public class Przystanek {
00009
00010     @Id
00011     @GeneratedValue(strategy = GenerationType.IDENTITY)
00012     @Column(name = "przystanek_id")
00013     private Long przystanekId;
00014
00015
00016     @ManyToOne(fetch = FetchType.LAZY)
00017     @JoinColumn(name = "przystanek_odwrotny_id", referencedColumnName = "przystanek_id")
00018     private Przystanek przystanekOdwrotny;
00019
00020
00021     @Column(name = "nazwa", nullable = false, length = 50)
00022     private String nazwa;
00023
00024     @Column(name = "kod_pocztowy", nullable = false, length = 10)
00025     private String kodPocztowy;
00026
00027     @Column(name = "miasto", nullable = false, length = 50)
00028     private String miasto;
00029
00030     @Column(name = "ulica", nullable = false, length = 50)
00031     private String ulica;
00032
00033     @Column(name = "dlugosc_geograficzna", nullable = false, length = 20)
00034     private String dlugoscGeograficzna;
00035
00036     @Column(name = "szerokosc_geograficzna", nullable = false, length = 20)
00037     private String szerokoscGeograficzna;
00038
00039
00040     @OneToMany(mappedBy = "przystanek")
00041     private List<PrzystanekWlini> przystankiWlini;
00042
00043
00044 // Getters
00045
00046     public Long getPrzystanekId() {
00047         return przystanekId;
00048     }
00049
00050     public Przystanek getPrzystanekOdwrotny() {
00051         return przystanekOdwrotny;
00052     }
00053
00054     public String getNazwa() {
00055         return nazwa;
00056     }
00057
00058     public String getKodPocztowy() {
00059         return kodPocztowy;
00060     }
00061
00062     public String getMiasto() {
00063         return miasto;
00064     }
00065
00066     public String getUlica() {
00067         return ulica;
00068     }
00069
00070     public String getDlugoscGeograficzna() {
00071         return dlugoscGeograficzna;
00072     }
00073
00074     public String getSzerokoscGeograficzna() {
00075         return szerokoscGeograficzna;
00076     }
00077
00078     public List<PrzystanekWlini> getPrzystankiWlini() {
00079         return przystankiWlini;
00080     }
00081
00082
00083 // Setters
00084
00085     public void setPrzystanekId(Long przystanekId) {
00086         this.przystanekId = przystanekId;
00087     }
00088
00089
00090
00091
00092
00093
00094
00095
00096
00097
00098
00099
00100
00101
00102
00103
00104
00105
00106
00107
00108
00109
00110
00111
00112
00113
00114
00115
00116
00117
00118
00119
00120
00121
00122
00123
00124
00125
00126
00127
00128
00129
00130
00131
00132
00133
00134
00135
00136
00137
00138
00139
00140
00141
00142
00143
00144
00145
00146
00147
00148
00149
00150
00151
00152
```

```
00153
00158     public void setPrzystanekOdwrotny(Przystanek przystanekOdwrotny) {
00159         this.przystanekOdwrotny = przystanekOdwrotny;
00160     }
00161
00166     public void setNazwa(String nazwa) {
00167         this.nazwa = nazwa;
00168     }
00169
00174     public void setKodPocztowy(String kodPocztowy) {
00175         this.kodPocztowy = kodPocztowy;
00176     }
00177
00182     public void setMiasto(String miasto) {
00183         this.miasto = miasto;
00184     }
00185
00190     public void setUlica(String ulica) {
00191         this.ulica = ulica;
00192     }
00193
00198     public void setDlugoscGeograficzna(String dlugoscGeograficzna) {
00199         this.dlugoscGeograficzna = dlugoscGeograficzna;
00200     }
00201
00206     public void setSzerokoscGeograficzna(String szerokoscGeograficzna) {
00207         this.szerokoscGeograficzna = szerokoscGeograficzna;
00208     }
00209
00214     public void setPrzystankiWlini(List<PrzystanekWlini> przystankiWlini) {
00215         this.przystankiWlini = przystankiWlini;
00216     }
00217 }
```

7.131 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_ ← Backend/src/main/java/com/mklachl/sopkom/model/entity/← PrzystanekWlini.java

Komponenty

- class [com.mklachl.sopkom.model.entity.PrzystanekWlini](#)

Pakiety

- package [com.mklachl.sopkom.model.entity](#)

7.132 PrzystanekWlini.java

Idź do dokumentacji tego pliku.

```
00001 package com.mklachl.sopkom.model.entity;
00002
00003 import org.hibernate.annotations.OnDelete;
00004 import org.hibernate.annotations.OnDeleteAction;
00005
00006 import jakarta.persistence.*;
00007 import java.util.List;
00008
00012 @Entity
00013 @Table(name = "przystanekwlini")
00014 public class PrzystanekWlini {
00015
00019     @Id
00020     @GeneratedValue(strategy = GenerationType.IDENTITY)
00021     @Column(name = "przystanekwlini_id")
00022     private Long przystanekWliniId;
```

```

00023
00027     @ManyToOne(fetch = FetchType.LAZY)
00028     @JoinColumn(name = "przystanek_id", referencedColumnName = "przystanek_id")
00029     private Przystanek przystanek;
00030
00034     @ManyToOne(fetch = FetchType.LAZY)
00035     @JoinColumn(name = "linia_id", referencedColumnName = "linia_id")
00036     @OnDelete(action = OnDeleteAction.CASCADE)
00037     private Linia linia;
00038
00042     @Column(name = "kolejnosc", nullable = false)
00043     private Short kolejnosc;
00044
00048     @OneToMany(mappedBy = "przystanekWlini")
00049     private List<KursPrzystanekWlini> kursPrzystanekWlini;
00050
00051     // Getters
00052
00057     public Long getPrzystanekWliniId() {
00058         return przystanekWliniId;
00059     }
00060
00065     public Przystanek getPrzystanek() {
00066         return przystanek;
00067     }
00068
00073     public Linia getLinia() {
00074         return linia;
00075     }
00076
00081     public Short getKolejnosc() {
00082         return kolejnosc;
00083     }
00084
00089     public List<KursPrzystanekWlini> getKursPrzystanekWlini() {
00090         return kursPrzystanekWlini;
00091     }
00092
00093     // Setters
00094
00099     public void setPrzystanekWliniId(Long przystanekWliniId) {
00100         this.przystanekWliniId = przystanekWliniId;
00101     }
00102
00107     public void setPrzystanek(Przystanek przystanek) {
00108         this.przystanek = przystanek;
00109     }
00110
00115     public void setLinia(Linia linia) {
00116         this.linia = linia;
00117     }
00118
00123     public void setKolejnosc(Short kolejnosc) {
00124         this.kolejnosc = kolejnosc;
00125     }
00126
00131     public void setKursPrzystanekWlini(List<KursPrzystanekWlini> kursPrzystanekWlini) {
00132         this.kursPrzystanekWlini = kursPrzystanekWlini;
00133     }
00134 }

```

7.133 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ Backend/src/main/java/com/mklachl/sopkom/model/entity/ Role.java

Komponenty

- enum [com.mklachl.sopkom.model.entity.Role](#)

Pakiety

- package [com.mklachl.sopkom.model.entity](#)

7.134 Role.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.model.entity;
00002
00006 public enum Role {
00007     ADMIN,
00008     MANAGER,
00009     DISPATCHER,
00010     DRIVER
00011 }
```

7.135 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/model/entity/← User.java

Komponenty

- class [com.mklachl.sopkom.model.entity.User](#)

Pakiety

- package [com.mklachl.sopkom.model.entity](#)

7.136 User.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.model.entity;
00002
00003 import java.util.Set;
00004 import jakarta.persistence.*;
00005
00009 @Entity
00010 @Table(name="users")
00011 public class User {
00012
00016     @Id
00017     @GeneratedValue(strategy = GenerationType.IDENTITY)
00018     private Long id;
00019
00023     private String name;
00024
00028     private String login;
00029
00033     private String password;
00034
00038     @Enumerated(EnumType.STRING)
00039     private Set<Role> roles;
00040
00044     public User() {
00045
00046
00054     public User(Long id, String name, String email, String password) {
00055         this.id = id;
00056         this.name = name;
00057         this.login = email;
00058         this.password = password;
00059     }
00060
00069     public User(Long id, String name, String email, String password, Set<Role> roles) {
00070         this.id = id;
00071         this.name = name;
00072         this.login = email;
```

```
00073     this.password = password;
00074     this.roles = roles;
00075 }
00076
00081 public Long getId() {
00082     return id;
00083 }
00084
00089 public void setId(Long id) {
00090     this.id = id;
00091 }
00092
00097 public String getName() {
00098     return name;
00099 }
00100
00105 public void setName(String name) {
00106     this.name = name;
00107 }
00108
00113 public String getLogin() {
00114     return login;
00115 }
00116
00121 public void setLogin(String login) {
00122     this.login = login;
00123 }
00124
00129 public String getPassword() {
00130     return password;
00131 }
00132
00137 public void setPassword(String password) {
00138     this.password = password;
00139 }
00140
00145 public Set<Role> getRoles() {
00146     return roles;
00147 }
00148
00153 public void setRoles(Set<Role> roles) {
00154     this.roles = roles;
00155 }
00156 }
```

7.137 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_[←](#) Backend/src/main/java/com/mklachl/sopkom/raporty/rozkład/[←](#) DateHelper.java

7.138 DateHelper.java

[Idź do dokumentacji tego pliku.](#)

7.139 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_[←](#) Backend/src/main/java/com/mklachl/sopkom/repository/Autobus Repository.java

Komponenty

- interface [com.mklachl.sopkom.repository.AutobusRepository](#)

Pakiety

- package com.mklachl.sopkom.repository

7.140 AutobusRepository.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.repository;
00002
00003 import org.springframework.data.repository.CrudRepository;
00004 import com.mklachl.sopkom.model.entity.Autobus;
00005 import org.springframework.data.jpa.repository.Query;
00006 import org.springframework.data.repository.query.Param;
00007
00008 import java.util.Date;
00009 import java.util.List;
00010
00014 public interface AutobusRepository extends CrudRepository<Autobus, Long> {
00015
00021     Autobus findByNumerRejestracyjny(String numerRejestracyjny);
00022
00028     Autobus findByPrzegladWaznyDoBefore(Date date);
00029
00035     Autobus findByPrzegladWaznyDoAfter(Date date);
00036
00042     List<Autobus> findAllByStatus(String status);
00043
00049     List<Autobus> findAllByPrzebiegGreaterThanOrEqual(float przebieg);
00050
00056     List<Autobus> findAllByPrzebiegLessThan(float przebieg);
00057
00063     List<Autobus> findAllByPrzebieg(float przebieg);
00064
00070     List<Autobus> findByStatusLike(String status);
00071
00077     List<Autobus> findAllByNumerRejestracyjnyLike(String numerRejestracyjny);
00078
00083     List<Autobus> findAll();
00084
00091     @Query(value = "SELECT * FROM autobusy a WHERE a.autobus_id NOT IN (SELECT p.autobus_id FROM
00092         przejazdy p WHERE p.data_startu < :endTime AND p.data_konca > :startTime)", nativeQuery = true)
00092     List<Autobus> findAllByAvailable(@Param("startTime") Date startTime, @Param("endTime") Date
00093         endTime);
```

7.141 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/repository/Bilet← Repository.java

Komponenty

- interface com.mklachl.sopkom.repository.BiletRepository

Pakiety

- package com.mklachl.sopkom.repository

7.142 BiletRepository.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.repository;
00002
00003 import com.mklachl.sopkom.model.entity.Bilet;
00004 import org.springframework.data.repository.CrudRepository;
00005
00006 import java.util.List;
00007
00011 public interface BiletRepository extends CrudRepository<Bilet, Short> {
00012
00018     List<Bilet> findAllByCena(float cena);
00019
00025     List<Bilet> findAllByNazwa(String nazwa);
00026
00031     List<Bilet> findAll();
00032 }
```

7.143 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/repository/← HarmonogramRepository.java

Komponenty

- interface [com.mklachl.sopkom.repository.HarmonogramRepository](#)

Pakiety

- package [com.mklachl.sopkom.repository](#)

7.144 HarmonogramRepository.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.repository;
00002
00003 import com.mklachl.sopkom.model.entity.Harmonogram;
00004 import org.springframework.data.repository.CrudRepository;
00005
00006 import java.util.Date;
00007 import java.util.List;
00008
00012 public interface HarmonogramRepository extends CrudRepository<Harmonogram, Short> {
00013
00019     List<Harmonogram> findAllByNazwa(String nazwa);
00020
00026     List<Harmonogram> findAllByPon(boolean pon);
00027
00033     List<Harmonogram> findAllByWto(boolean wto);
00034
00040     List<Harmonogram> findAllBySro(boolean sro);
00041
00047     List<Harmonogram> findAllByCzw(boolean czw);
00048
00054     List<Harmonogram> findAllByPia(boolean pia);
00055
00061     List<Harmonogram> findAllBySob(boolean sob);
00062
00068     List<Harmonogram> findAllByNie(boolean nie);
00069
00075     List<Harmonogram> findAllByDodatkoweInfLike(String dodatkoweInf);
00076
00081     List<Harmonogram> findAll();
00082 }
```

7.145 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ ← Backend/src/main/java/com/mklachl/sopkom/repository/Icydent_ ← Repository.java

Komponenty

- interface [com.mklachl.sopkom.repository.IcydentRepository](#)

Pakietы

- package [com.mklachl.sopkom.repository](#)

7.146 IcydentRepository.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.repository;
00002
00003 import com.mklachl.sopkom.model.entity.Autobus;
00004 import com.mklachl.sopkom.model.entity.Incydent;
00005 import com.mklachl.sopkom.model.entity.Kierowca;
00006 import com.mklachl.sopkom.model.entity.Przejazd;
00007 import org.springframework.data.repository.CrudRepository;
00008
00009 import java.util.Date;
00010 import java.util.List;
00011
00015 public interface IcydentRepository extends CrudRepository<Incydent, Long> {
00016
00022     List<Incydent> findAllByTyp(String typ);
00023
00029     List<Incydent> findAllByDateBefore(Date date);
00030
00036     List<Incydent> findAllByDateAfter(Date date);
00037
00044     List<Incydent> findAllByDateBetween(Date date1, Date date2);
00045
00051     List<Incydent> findAllByDate(Date date);
00052
00058     List<Incydent> findAllByAutobus(Autobus autobus);
00059
00065     List<Incydent> findAllByKierowca(Kierowca kierowca);
00066
00072     List<Incydent> findAllByPrzejazd(Przejazd przejazd);
00073
00079     List<Incydent> findAllByDodatkoweInformacje(String dodatkoweInformacje);
00080
00086     List<Incydent> findAllByKosztyLessThan(float koszty);
00087
00093     List<Incydent> findAllByKosztyGreaterThanOrEqual(float koszty);
00094
00100    List<Incydent> findAllByKoszty(float koszty);
00101
00106    List<Incydent> findAll();
00107 }
```

7.147 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ ← Backend/src/main/java/com/mklachl/sopkom/repository/ ← KierowcaRepository.java

Komponenty

- interface [com.mklachl.sopkom.repository.KierowcaRepository](#)

Pakiety

- package [com.mklachl.sopkom.repository](#)

7.148 KierowcaRepository.java

[Idź do dokumentacji tego pliku.](#)

```

00001 package com.mklachl.sopkom.repository;
00002
00003 import org.springframework.data.jpa.repository.Query;
00004 import org.springframework.data.repository.CrudRepository;
00005 import com.mklachl.sopkom.model.entity.Kierowca;
00006 import com.mklachl.sopkom.model.entity.User;
00007 import org.springframework.data.repository.query.Param;
00008
00009 import java.util.Date;
00010 import java.util.List;
00011
00015 public interface KierowcaRepository extends CrudRepository<Kierowca, Long> {
00016
00022     Kierowca findByPesel(String pesel);
00023
00029     List<Kierowca> findAllByPeselLike(String pesel);
00030
00036     Kierowca findByUser(User user);
00037
00044     List<Kierowca> findAllByImieAndNazwisko(String imie, String nazwisko);
00045
00052     List<Kierowca> findAllByImieAndNazwiskoLike(String imie, String nazwisko);
00053
00059     List<Kierowca> findAllByPrawoJazdyWazneDoBefore(Date date);
00060
00066     List<Kierowca> findAllByPrawoJazdyWazneDoAfter(Date date);
00067
00072     List<Kierowca> findAll();
00073
00079     List<Kierowca> findByNazwiskoLike(String nazwisko);
00080
00087     @Query(value = "SELECT * FROM kierowcy k WHERE k.kierowca_id NOT IN (SELECT p.kierowca_id FROM
00088     przejazdy p WHERE p.data_startu < :endTime AND p.data_konca > :startTime)", nativeQuery = true)
00088     List<Kierowca> findAllByAvailability(@Param("startTime") Date startTime, @Param("endTime") Date
00089     endTime);
00089 }
```

7.149 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/repository/Kurs← PrzystanekWliniRepository.java

Komponenty

- interface [com.mklachl.sopkom.repository.KursPrzystanekWliniRepository](#)

Pakiety

- package [com.mklachl.sopkom.repository](#)

7.150 KursPrzystanekWliniRepository.java

[Idź do dokumentacji tego pliku.](#)

```

00001 package com.mklachl.sopkom.repository;
00002
00003 import com.mklachl.sopkom.model.entity.Kurs;
00004 import com.mklachl.sopkom.model.entity.PrzystanekWlini;
00005 import org.springframework.data.repository.CrudRepository;
00006 import com.mklachl.sopkom.model.entity.KursPrzystanekWlini;
00007
00008 import java.util.Date;
00009 import java.util.List;
00010
00014 public interface KursPrzystanekWliniRepository extends CrudRepository<KursPrzystanekWlini, Long> {
00015
00021     List<KursPrzystanekWlini> findAllByKurs(Kurs kurs);
00022
00028     List<KursPrzystanekWlini> findAllByPrzystanekWlini(PrzystanekWlini przystanekWlini);
00029
00035     List<KursPrzystanekWlini> findAllByGodzinnaAfter(Date date);
00036
00042     List<KursPrzystanekWlini> findAllByGodzinnaBefore(Date date);
00043
00049     List<KursPrzystanekWlini> findAllByGodzinna(Date date);
00050
00055     List<KursPrzystanekWlini> findAll();
00056 }
```

7.151 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/repository/Kurs← Repository.java

Komponenty

- interface [com.mklachl.sopkom.repository.KursRepository](#)

Pakiety

- package [com.mklachl.sopkom.repository](#)

7.152 KursRepository.java

[Idź do dokumentacji tego pliku.](#)

```

00001 package com.mklachl.sopkom.repository;
00002
00003 import com.mklachl.sopkom.model.entity.Harmonogram;
00004 import com.mklachl.sopkom.model.entity.Linia;
00005 import org.springframework.data.repository.CrudRepository;
00006 import com.mklachl.sopkom.model.entity.Kurs;
00007
00008 import java.util.List;
00009
00013 public interface KursRepository extends CrudRepository<Kurs, Long> {
00014
00020     List<Kurs> findAllByLinia(Linia linia);
00021
00027     List<Kurs> findAllByHarmonogram(Harmonogram harmonogram);
00028
00034     List<Kurs> findAllByKierunek(Short kierunek);
00035
00041     List<Kurs> findAllByTypAutobusu(Short typAutobusu);
00042
00047     List<Kurs> findAll();
00048 }
```

7.153 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_↪ Backend/src/main/java/com/mklachl/sopkom/repository/Linia↪ Repository.java

Komponenty

- interface [com.mklachl.sopkom.repository.LiniaRepository](#)

Pakiety

- package [com.mklachl.sopkom.repository](#)

7.154 LiniaRepository.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.repository;
00002
00003 import org.springframework.data.repository.CrudRepository;
00004 import com.mklachl.sopkom.model.entity.Linia;
00005
00006 import java.util.List;
00007 import java.util.Optional;
00008
00012 public interface LiniaRepository extends CrudRepository<Linia, Long> {
00013
00019     Optional<Linia> findById(Long id);
00020
00026     Linia findByNumer(String numer);
00027
00032     List<Linia> findAll();
00033 }
```

7.155 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_↪ Backend/src/main/java/com/mklachl/sopkom/repository/Przejazd↪ BiletRepository.java

Komponenty

- interface [com.mklachl.sopkom.repository.PrzejazdBiletRepository](#)

Pakiety

- package [com.mklachl.sopkom.repository](#)

7.156 PrzejazdBiletRepository.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.repository;
00002
00003 import com.mklachl.sopkom.model.entity.Bilet;
00004 import com.mklachl.sopkom.model.entity.Przejazd;
00005 import com.mklachl.sopkom.model.entity.PrzejazdBilet;
00006 import org.springframework.data.repository.CrudRepository;
00007
00008 import java.util.List;
00009
00013 public interface PrzejazdBiletRepository extends CrudRepository<PrzejazdBilet, Long> {
00014
00020     List<PrzejazdBilet> findAllByPrzejazd(Przejazd przejazd);
00021
00027     List<PrzejazdBilet> findAllByBilet(Bilet bilet);
00028
00034     List<PrzejazdBilet> findAllByCenaBiletuGreaterThanOrEqual(float cenaBiletu);
00035
00041     List<PrzejazdBilet> findAllByCenaBiletuLessThan(float cenaBiletu);
00042
00048     List<PrzejazdBilet> findAllByCenaBiletu(float cenaBiletu);
00049
00054     List<PrzejazdBilet> findAll();
00055 }
```

7.157 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/repository/Przejazd← KursPrzystanekWliniRepository.java

Komponenty

- interface [com.mklachl.sopkom.repository.PrzejazdKursPrzystanekWliniRepository](#)

Pakiety

- package [com.mklachl.sopkom.repository](#)

7.158 PrzejazdKursPrzystanekWliniRepository.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.repository;
00002
00003 import com.mklachl.sopkom.model.entity.KursPrzystanekWlini;
00004 import com.mklachl.sopkom.model.entity.Przejazd;
00005 import com.mklachl.sopkom.model.entity.PrzejazdKursPrzystanekWlini;
00006 import org.springframework.data.repository.CrudRepository;
00007
00008 import java.util.Date;
00009 import java.util.List;
00010
00014 public interface PrzejazdKursPrzystanekWliniRepository extends
    CrudRepository<PrzejazdKursPrzystanekWlini, Long> {
00015
00021     List<PrzejazdKursPrzystanekWlini> findAllByPrzejazd(Przejazd przejazd);
00022
00028     List<PrzejazdKursPrzystanekWlini> findAllByKursPrzystanekWlini(KursPrzystanekWlini
        kursPrzystanekWlini);
00029
00035     List<PrzejazdKursPrzystanekWlini> findAllByRealnaGodzinna(Date realnaGodzinna);
00036
00042     List<PrzejazdKursPrzystanekWlini> findAllByRealnaGodzinnaBefore(Date realnaGodzinna);
00043
00049     List<PrzejazdKursPrzystanekWlini> findAllByRealnaGodzinnaAfter(Date realnaGodzinna);
00050
00055     List<PrzejazdKursPrzystanekWlini> findAll();
00056 }
```

7.159 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ ↵ Backend/src/main/java/com/mklachl/sopkom/repository/Przejazd ↵ Repository.java

Komponenty

- interface [com.mklachl.sopkom.repository.PrzejazdRepository](#)

Pakiety

- package [com.mklachl.sopkom.repository](#)

7.160 PrzejazdRepository.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.repository;
00002
00003 import com.mklachl.sopkom.model.entity.Autobus;
00004 import com.mklachl.sopkom.model.entity.Kierowca;
00005 import com.mklachl.sopkom.model.entity.Kurs;
00006 import com.mklachl.sopkom.model.entity.Przejazd;
00007 import org.springframework.data.jpa.repository.Query;
00008 import org.springframework.data.repository.CrudRepository;
00009 import org.springframework.data.repository.query.Param;
00010
00011 import java.util.Date;
00012 import java.util.List;
00013
00017 public interface PrzejazdRepository extends CrudRepository<Przejazd, Long> {
00018
00024     List<Przejazd> findAllByKierowca(Kierowca kierowca);
00025
00031     List<Przejazd> findAllByAutobus(Autobus autobus);
00032
00038     List<Przejazd> findAllByKurs(Kurs kurs);
00039
00045     List<Przejazd> findAllBySpalanie(float spalanie);
00046
00052     List<Przejazd> findAllBySpalanieLessThan(float spalanie);
00053
00059     List<Przejazd> findAllBySpalanieGreaterThan(float spalanie);
00060
00066     List<Przejazd> findAllByCenaZaLitru(float cenaZaLitru);
00067
00073     List<Przejazd> findAllByCenaZaLitruLessThan(float cenaZaLitru);
00074
00080     List<Przejazd> findAllByCenaZaLitruGreaterThan(float cenaZaLitru);
00081
00087     List<Przejazd> findAllByDlugoscTrasy(float dlugoscTrasy);
00088
00094     List<Przejazd> findAllByDlugoscTrasyLessThan(float dlugoscTrasy);
00095
00101     List<Przejazd> findAllByDlugoscTrasyGreaterThan(float dlugoscTrasy);
00102
00108     List<Przejazd> findAllByDataStartu(Date dataStartu);
00109
00115     List<Przejazd> findAllByDataStartuBefore(Date dataStartu);
00116
00122     List<Przejazd> findAllByDataStartuAfter(Date dataStartu);
00123
00129     List<Przejazd> findAllByDataKonca(Date dataKonca);
00130
00136     List<Przejazd> findAllByDataKoncaBefore(Date dataKonca);
00137
00143     List<Przejazd> findAllByDataKoncaAfter(Date dataKonca);
00144
00150     List<Przejazd> findAllByData(Date data);
00151
```

```
00158     @Query(value = "SELECT * FROM przejazdy p WHERE p.kurs_id = :kursId AND (p.data_startu =  
00159         :dataPrzejazdu or p.data_konca = :dataPrzejazdu)", nativeQuery = true)  
00160     List<Przejazd> findAllByKursAndData(@Param("kurs") Kurs kurs, @Param("dataPrzejazdu") Date  
00161         dataPrzejazdu);  
00162     @Query(value = "SELECT * FROM przejazdy p WHERE p.kurs_id = :kursId AND (p.data_startu =  
00163         :dataPrzejazdu or p.data_konca = :dataPrzejazdu)", nativeQuery = true)  
00164     List<Przejazd> findAllByData(@Param("kurs") Kurs kurs, @Param("dataPrzejazdu") Date  
00165         dataPrzejazdu);  
00166     List<Przejazd> findAll();  
00175 }
```

7.161 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ ← Backend/src/main/java/com/mklachl/sopkom/repository/← PrzystanekRepository.java

Komponenty

- interface [com.mklachl.sopkom.repository.PrzystanekRepository](#)

Pakiety

- package [com.mklachl.sopkom.repository](#)

7.162 PrzystanekRepository.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.repository;  
00002  
00003 import java.util.List;  
00004  
00005 import org.springframework.data.repository.CrudRepository;  
00006  
00007 import com.mklachl.sopkom.model.entity.Przystanek;  
00008  
00012 public interface PrzystanekRepository extends CrudRepository<Przystanek, Long> {  
00013  
00019     List<Przystanek> findAllByPrzystanekOdwrotny(Przystanek przystanekOdwrotny);  
00020  
00026     Przystanek findByNazwa(String nazwa);  
00027  
00033     List<Przystanek> findAllByNazwaLike(String nazwa);  
00034  
00040     List<Przystanek> findAllByKodPocztowy(String kodPocztowy);  
00041  
00047     List<Przystanek> findAllByMiasto(String miasto);  
00048  
00054     List<Przystanek> findAllByUlica(String ulica);  
00055  
00061     List<Przystanek> findAllByUlicaLike(String ulica);  
00062  
00068     List<Przystanek> findAllBySzerokoscGeograficzna(String szerokoscGeograficzna);  
00069  
00075     List<Przystanek> findAllByDlugoscGeograficzna(String dlugoscGeograficzna);  
00076  
00081     List<Przystanek> findAll();  
00082 }
```

7.163 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_↪ Backend/src/main/java/com/mklachl/sopkom/repository/↪ PrzystanekWliniRepository.java

Komponenty

- interface [com.mklachl.sopkom.repository.PrzystanekWliniRepository](#)

Pakiety

- package [com.mklachl.sopkom.repository](#)

7.164 PrzystanekWliniRepository.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.repository;
00002
00003 import com.mklachl.sopkom.model.entity.Linia;
00004 import com.mklachl.sopkom.model.entity.Przystanek;
00005 import org.springframework.data.repository.CrudRepository;
00006 import com.mklachl.sopkom.model.entity.PrzystanekWlini;
00007
00008 import java.util.List;
00009
00013 public interface PrzystanekWliniRepository extends CrudRepository<PrzystanekWlini, Long> {
00014     List<PrzystanekWlini> findAllByPrzystanek(Przystanek przystanek);
00021
00027     List<PrzystanekWlini> findAllByLinia(Linia linia);
00028
00033     List<PrzystanekWlini> findAll();
00034 }
```

7.165 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_↪ Backend/src/main/java/com/mklachl/sopkom/repository/User↪ Repository.java

Komponenty

- interface [com.mklachl.sopkom.repository.UserRepository](#)

Pakiety

- package [com.mklachl.sopkom.repository](#)

7.166 UserRepository.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.repository;
00002
00003 import org.springframework.data.repository.CrudRepository;
00004 import com.mklachl.sopkom.model.entity.User;
00005
00009 public interface UserRepository extends CrudRepository<User, Long> {
00010
00016     User findByLogin(String login);
00017 }
```

7.167 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ ↵ Backend/src/main/java/com/mklachl/sopkom/security/Command ↵ LineRunnerTest.java

Komponenty

- class [com.mklachl.sopkom.security.CommandLineRunnerTest](#)

Pakiety

- package [com.mklachl.sopkom.security](#)

7.168 CommandLineRunnerTest.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.security;
00002
00003 import java.util.HashSet;
00004 import java.util.Set;
00005
00006 import org.springframework.beans.factory.annotation.Autowired;
00007 import org.springframework.boot.CommandLineRunner;
00008 import org.springframework.stereotype.Service;
00009 import org.springframework.transaction.annotation.Transactional;
00010
00011 import com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto;
00012 import com.mklachl.sopkom.model.entity.Bilet;
00013 import com.mklachl.sopkom.model.entity.Role;
00014 import com.mklachl.sopkom.model.entity.User;
00015 import com.mklachl.sopkom.repository.BiletRepository;
00016 import com.mklachl.sopkom.repository.HarmonogramRepository;
00017 import com.mklachl.sopkom.repository.UserRepository;
00018 import com.mklachl.sopkom.services.HarmonogramService;
00019 import com.mklachl.sopkom.services.UserService;
00020
00021 @Service
00022 public class CommandLineRunnerTest implements CommandLineRunner {
00023
00024     @Autowired
00025     public UserRepository userRepository;
00026
00027     @Autowired
00028     UserService userService;
00029
00030     @Autowired
00031     BiletRepository biletRepo;
00032
00033     public static final String ADMIN_LOGIN = "admin";
00034     public static final String ADMIN_NAME = "admin";
00035     public static final String ADMIN_PASS = "admin";
```

```

00036
00037     public static final String MANAGER_LOGIN = "manager";
00038     public static final String MANAGER_NAME = "manager";
00039     public static final String MANAGER_PASS = "manager";
00040
00041     @Autowired
00042     HarmonogramService heService;
00043
00044     @Autowired
00045     HarmonogramRepository heRepo;
00046
00047     @Override
00048     @Transactional
00049     public void run(String... args) throws Exception {
00050
00051         System.out.println("if you see this, then it worked :)");
00052         {
00053             User adminIn = userRepository.findByLogin(ADMIN_LOGIN);
00054
00055             if(adminIn == null) {
00056                 adminIn = new User();
00057                 adminIn.setLogin(ADMIN_LOGIN);
00058                 adminIn.setName(ADMIN_NAME);
00059                 adminIn.setPassword(ADMIN_PASS);
00060
00061                 Set<Role> adminRole = new HashSet<Role>();
00062                 adminRole.add(Role.ADMIN);
00063
00064                 adminIn.setRoles(adminRole);
00065
00066                 userRepository.deleteAll();
00067
00068                 userService.saveUser(adminIn);
00069
00070                 System.out.println("Admin account has been created!!!");
00071
00072         }
00073     }
00074
00075     User managerIn = userRepository.findByLogin(MANAGER_LOGIN);
00076
00077     if(managerIn == null) {
00078         managerIn = new User();
00079         managerIn.setLogin(MANAGER_LOGIN);
00080         managerIn.setName(MANAGER_NAME);
00081         managerIn.setPassword(MANAGER_PASS);
00082
00083         Set<Role> adminRole = new HashSet<Role>();
00084         adminRole.add(Role.MANAGER);
00085
00086         managerIn.setRoles(adminRole);
00087
00088         userService.saveUser(managerIn);
00089
00090         System.out.println("Manager account has been created!!!");
00091
00092     }
00093
00094     User dyspIn = userRepository.findByLogin("dyspozytor");
00095
00096     if(dyspIn != null) {
00097         userRepository.delete(dyspIn);
00098         dyspIn = null;
00099     }
00100
00101     if(dyspIn == null) {
00102         dyspIn = new User();
00103         dyspIn.setLogin("dyspozytor");
00104         dyspIn.setName("dyspozytor");
00105         dyspIn.setPassword("dyspozytor");
00106
00107         Set<Role> adminRole = new HashSet<Role>();
00108         adminRole.add(Role.DISPATCHER);
00109
00110         dyspIn.setRoles(adminRole);
00111
00112         userService.saveUser(dyspIn);
00113
00114         System.out.println("Dispatcher account has been created!!!");
00115
00116     }
00117
00118     if(heRepo.count() < 3){
00119
00120         {
00121             HarmonogramDto harmonogram = new HarmonogramDto();
00122

```

```
00123         // Set sample data
00124         harmonogram.setHarmonogramId((short) 1);
00125         harmonogram.setNazwa("Dni robocze");
00126         harmonogram.setPon(true);
00127         harmonogram.setWto(true);
00128         harmonogram.setSro(true);
00129         harmonogram.setCzw(true);
00130         harmonogram.setPia(true);
00131         harmonogram.setSob(false);
00132         harmonogram.setNie(false);
00133         harmonogram.setDodatkoweInf("Dni robocze tylko.");
00134
00135         heService.saveHarmonogram(harmonogram);
00136
00137     }
00138
00139     {
00140
00141         HarmonogramDto harmonogram = new HarmonogramDto();
00142
00143         // Set sample data
00144         harmonogram.setHarmonogramId((short) 2);
00145         harmonogram.setNazwa("Dni robocze i weekendy");
00146         harmonogram.setPon(true);
00147         harmonogram.setWto(true);
00148         harmonogram.setSro(true);
00149         harmonogram.setCzw(true);
00150         harmonogram.setPia(true);
00151         harmonogram.setSob(true);
00152         harmonogram.setNie(true);
00153         harmonogram.setDodatkoweInf("Cały tydzień.");
00154
00155         heService.saveHarmonogram(harmonogram);
00156     }
00157
00158     {
00159
00160         HarmonogramDto harmonogram = new HarmonogramDto();
00161
00162         // Set sample data
00163         harmonogram.setHarmonogramId((short) 3);
00164         harmonogram.setNazwa("weekendy");
00165         harmonogram.setPon(false);
00166         harmonogram.setWto(false);
00167         harmonogram.setSro(false);
00168         harmonogram.setCzw(false);
00169         harmonogram.setPia(false);
00170         harmonogram.setSob(true);
00171         harmonogram.setNie(true);
00172         harmonogram.setDodatkoweInf("Tylko weekendy.");
00173
00174         heService.saveHarmonogram(harmonogram);
00175     }
00176 }
00177 //System.out.println(biletRepo.findById((short)0));
00178
00179 }
00180 }
```

7.169 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ ← Backend/src/main/java/com/mklachl/sopkom/security/Custom← UserDetailsService.java

Komponenty

- class [com.mklachl.sopkom.security.CustomUserDetailsService](#)

Pakiety

- package [com.mklachl.sopkom.security](#)

7.170 CustomUserDetailsService.java

[Idź do dokumentacji tego pliku.](#)

```

00001 package com.mklachl.sopkom.security;
00002
00003
00004 import org.springframework.security.core.userdetails.UserDetails;
00005 import org.springframework.security.core.userdetails.UserDetailsService;
00006 import org.springframework.security.core.userdetails.UsernameNotFoundException;
00007 import org.springframework.stereotype.Service;
00008
00009 import com.mklachl.sopkom.model.entity.User;
00010 import com.mklachl.sopkom.repository.UserRepository;
00011
00012 @Service
00013 public class CustomUserDetailsService implements UserDetailsService {
00014
00015     private UserRepository userRepository;
00016
00017     public CustomUserDetailsService(UserRepository userRepository) {
00018         this.userRepository = userRepository;
00019     }
00020
00021     @Override
00022     public UserDetails loadUserByUsername(String login) throws UsernameNotFoundException {
00023         User user = userRepository.findByLogin(login);
00024         System.out.println(login);
00025
00026         if (user != null) {
00027             //code for granting the user privileges
00028             //List<GrantedAuthority> privileges = new LinkedList<>();
00029             //privileges.add(new SimpleGrantedAuthority("PROFI"));
00030             //return new org.springframework.security.core.userdetails.User(user.getEmail(),
00031             //        /user.getPassword(),
00032             //        //privileges);
00033
00034             return org.springframework.security.core.userdetails.User
00035                 .withUsername(user.getLogin())
00036                 .password(user.getPassword())
00037                 .roles(user.getRoles().stream().map(Enum::name).toArray(String[]::new))
00038                 .build();
00039
00040         } else{
00041             throw new UsernameNotFoundException("Invalid username or password.");
00042         }
00043     }
00044 }
00045 }
```

7.171 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/services/Autobus← Service.java

Komponenty

- interface [com.mklachl.sopkom.services.AutobusService](#)

Pakietы

- package [com.mklachl.sopkom.services](#)

7.172 AutobusService.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.services;
00002
00003 import com.mklachl.sopkom.model.entity.Autobus;
00004 import com.mklachl.sopkom.model.dto.AutobusDto;
00005 import org.springframework.data.repository.query.Param;
00006
00007 import java.util.Date;
00008 import java.util.List;
00009 import java.util.Optional;
00010
00014 public interface AutobusService {
00015     Autobus saveAutobus(Autobus autobus);
00022
00028     Autobus saveAutobus(AutobusDto autobusDto);
00029
00035     Autobus findAutobusByNumerRejestracjny(String numerRejestracjny);
00036
00042     Optional<Autobus> findAutobusById(long id);
00043
00049     Autobus findAutobusByPrzegladWaznyDoBefore(Date date);
00050
00056     Autobus findAutobusByPrzegladWaznyDoAfter(Date date);
00057
00063     List<Autobus> findAutobusByStatus(String status);
00064
00070     List<Autobus> findAutobusByPrzebiegGreaterThanOrEqual(float przebieg);
00071
00077     List<Autobus> findAutobusByPrzebiegLessThan(float przebieg);
00078
00084     List<Autobus> findAutobusByPrzebieg(float przebieg);
00085
00092     List<Autobus> findAutobusByAvailable(Date startTime, Date endTime);
00093 }
00094 }
```

7.173 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/services/← HarmonogramService.java

Komponenty

- interface [com.mklachl.sopkom.services.HarmonogramService](#)

Pakiety

- package [com.mklachl.sopkom.services](#)

7.174 HarmonogramService.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.services;
00002
00003 import com.mklachl.sopkom.model.entity.Harmonogram;
00004 import com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto;
00005
00006 import java.util.List;
00007 import java.util.Optional;
00008
00012 public interface HarmonogramService {
```

```

00013
00019     Harmonogram saveHarmonogram(Harmonogram harmonogram);
00020
00026     Harmonogram saveHarmonogram(HarmonogramDto harmonogramDto);
00027
00033     Optional<Harmonogram> findHarmonogramById(Short id);
00034
00040     List<Harmonogram> findHarmonogramByNazwa(String nazwa);
00041
00047     List<Harmonogram> findHarmonogramByPon(boolean pon);
00048
00054     List<Harmonogram> findHarmonogramByWto(boolean wto);
00055
00061     List<Harmonogram> findHarmonogramBySro(boolean sro);
00062
00068     List<Harmonogram> findHarmonogramByCzw(boolean czw);
00069
00075     List<Harmonogram> findHarmonogramByPia(boolean pia);
00076
00082     List<Harmonogram> findHarmonogramBySob(boolean sob);
00083
00089     List<Harmonogram> findHarmonogramByNie(boolean nie);
00090
00096     List<Harmonogram> findHarmonogramByDodatkoweInfLike(String dodatkoweInf);
00097
00098 }

```

7.175 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ Backend/src/main/java/com/mklachl/sopkom/services/impl/ AutobusServiceImpl.java

Komponenty

- class [com.mklachl.sopkom.services.impl.AutobusServiceImpl](#)

Pakiety

- package [com.mklachl.sopkom.services.impl](#)

7.176 AutobusServiceImpl.java

[Idź do dokumentacji tego pliku.](#)

```

00001 package com.mklachl.sopkom.services.impl;
00002
00003 import org.springframework.stereotype.Service;
00004 import com.mklachl.sopkom.model.entity.Autobus;
00005 import com.mklachl.sopkom.model.dto.AutobusDto;
00006 import com.mklachl.sopkom.repository.AutobusRepository;
00007 import com.mklachl.sopkom.services.AutobusService;
00008
00009 import java.util.Date;
00010 import java.util.List;
00011 import java.util.Optional;
00012
00013 @Service
00014 public class AutobusServiceImpl implements AutobusService {
00015     private AutobusRepository autobusRepository;
00016
00017     public AutobusServiceImpl(AutobusRepository autobusRepository) {
00018         this.autobusRepository = autobusRepository;
00019     }
00020
00021     public Autobus saveAutobus(AutobusDto autobusDto) {
00022         Autobus autobus = new Autobus();
00023         autobus.setNumerRejestracyjny(autobusDto.getNumerRejestracyjny());

```

```
00024     autobus.setPrzebieg(autobusDto.getPrzebieg());
00025     autobus.setPrzegladWaznyDo(autobusDto.getPrzegladWaznyDo());
00026     autobus.setStatus(autobusDto.getStatus());
00027 
00028     autobus = autobusRepository.save(autobus);
00029     return autobus;
00030 }
00031 
00032 @Override
00033 public Autobus saveAutobus(Autobus autobusDto) {
00034     Autobus autobus = new Autobus();
00035     autobus.setNumerRejestracyjny(autobusDto.getNumerRejestracyjny());
00036     autobus.setPrzebieg(autobusDto.getPrzebieg());
00037     autobus.setPrzegladWaznyDo(autobusDto.getPrzegladWaznyDo());
00038     autobus.setStatus(autobusDto.getStatus());
00039 
00040     autobus = autobusRepository.save(autobus);
00041     return autobus;
00042 }
00043 
00044 @Override
00045 public Autobus findAutobusByNumerRejestracyjny(String numerRejestracyjny){
00046     return autobusRepository.findByNumerRejestracyjny(numerRejestracyjny);
00047 }
00048 
00049 @Override
00050 public Optional<Autobus> findAutobusById(long id){
00051     return autobusRepository.findById(id);
00052 }
00053 
00054 @Override
00055 public Autobus findAutobusByPrzegladWaznyDoBefore(Date date){
00056     return autobusRepository.findByPrzegladWaznyDoBefore(date);
00057 }
00058 
00059 @Override
00060 public Autobus findAutobusByPrzegladWaznyDoAfter(Date date){
00061     return autobusRepository.findByPrzegladWaznyDoAfter(date);
00062 }
00063 
00064 @Override
00065 public List<Autobus> findAutobusByStatus(String status){
00066     return autobusRepository.findAllByStatus(status);
00067 }
00068 
00069 @Override
00070 public List<Autobus> findAutobusByPrzebiegGreaterThanOrEqual(float przebieg){
00071     return autobusRepository.findAllByPrzebiegGreaterThanOrEqual(przebieg);
00072 }
00073 
00074 @Override
00075 public List<Autobus> findAutobusByPrzebiegLessThan(float przebieg){
00076     return autobusRepository.findAllByPrzebiegLessThan(przebieg);
00077 }
00078 
00079 @Override
00080 public List<Autobus> findAutobusByPrzebieg(float przebieg){
00081     return autobusRepository.findAllByPrzebieg(przebieg);
00082 }
00083 
00084 @Override
00085 public List<Autobus> findAutobusByAvailable(Date startTime, Date endTime){ return
00086     autobusRepository.findAllByAvailable(startTime, endTime); }
00087 }
```

7.177 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ ← Backend/src/main/java/com/mklachl/sopkom/services/impl/ ← HarmonogramServiceImpl.java

Komponenty

- class [com.mklachl.sopkom.services.impl.HarmonogramServiceImpl](#)

Pakiety

- package com.mklachl.sopkom.services.impl

7.178 HarmonogramServiceImpl.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.services.impl;
00002
00003 import com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto;
00004 import com.mklachl.sopkom.model.entity.Harmonogram;
00005 import com.mklachl.sopkom.repository.HarmonogramRepository;
00006 import com.mklachl.sopkom.services.HarmonogramService;
00007
00008 import java.util.List;
00009 import java.util.Optional;
00010
00011 import org.springframework.stereotype.Service;
00012
00013 @Service
00014 public class HarmonogramServiceImpl implements HarmonogramService {
00015     private HarmonogramRepository harmonogramRepository;
00016
00017     public HarmonogramServiceImpl(HarmonogramRepository harmonogramRepository)
00018         {this.harmonogramRepository = harmonogramRepository;}
00019
00020     @Override
00021     public Harmonogram saveHarmonogram(Harmonogram harmonogramDto){
00022         Harmonogram harmonogram = new Harmonogram();
00023         harmonogram.setNazwa(harmonogramDto.getNazwa());
00024         harmonogram.setPon(harmonogramDto.isPon());
00025         harmonogram.setWto(harmonogramDto.isWto());
00026         harmonogram.setSro(harmonogramDto.isSro());
00027         harmonogram.setCzw(harmonogramDto.isCzw());
00028         harmonogram.setPia(harmonogramDto.isPia());
00029         harmonogram.setSob(harmonogramDto.isSob());
00030         harmonogram.setNie(harmonogramDto.isNie());
00031         harmonogram.setDodatkoweInf(harmonogramDto.getDodatkoweInf());
00032
00033         harmonogram = harmonogramRepository.save(harmonogram);
00034         return harmonogram;
00035     }
00036
00037     public Harmonogram saveHarmonogram(HarmonogramDto harmonogramDto){
00038         Harmonogram harmonogram = new Harmonogram();
00039         harmonogram.setNazwa(harmonogramDto.getNazwa());
00040         harmonogram.setPon(harmonogramDto.isPon());
00041         harmonogram.setWto(harmonogramDto.isWto());
00042         harmonogram.setSro(harmonogramDto.isSro());
00043         harmonogram.setCzw(harmonogramDto.isCzw());
00044         harmonogram.setPia(harmonogramDto.isPia());
00045         harmonogram.setSob(harmonogramDto.isSob());
00046         harmonogram.setNie(harmonogramDto.isNie());
00047         harmonogram.setDodatkoweInf(harmonogramDto.getDodatkoweInf());
00048
00049         harmonogram = harmonogramRepository.save(harmonogram);
00050         return harmonogram;
00051     }
00052
00053     @Override
00054     public Optional<Harmonogram> findHarmonogramById(Short id){return
00055         harmonogramRepository.findById(id);}
00056
00057     @Override
00058     public List<Harmonogram> findHarmonogramByNazwa(String nazwa){return
00059         harmonogramRepository.findAllByNazwa(nazwa);}
00060
00061     @Override
00062     public List<Harmonogram> findHarmonogramByPon(boolean pon){return
00063         harmonogramRepository.findAllByPon(pon);}
00064
00065     @Override
00066     public List<Harmonogram> findHarmonogramByWto(boolean wto){return
00067         harmonogramRepository.findAllByWto(wto);}
```

```
00068     public List<Harmonogram> findHarmonogramByCzw(boolean czw) {return  
00069         harmonogramRepository.findAllByCzw(czw);}  
00070     @Override  
00071     public List<Harmonogram> findHarmonogramByPia(boolean pia) {return  
00072         harmonogramRepository.findAllByPia(pia);}  
00073     @Override  
00074     public List<Harmonogram> findHarmonogramBySob(boolean sob) {return  
00075         harmonogramRepository.findAllBySob(sob);}  
00076     @Override  
00077     public List<Harmonogram> findHarmonogramByNie(boolean nie) {return  
00078         harmonogramRepository.findAllByNie(nie);}  
00079     @Override  
00080     public List<Harmonogram> findHarmonogramByDodatkoweInfLike(String dodatkoweInf) {return  
00081         harmonogramRepository.findAllByDodatkoweInfLike(dodatkoweInf);}
```

7.179 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ ← Backend/src/main/java/com/mklachl/sopkom/services/impl/ ← IIncidentServiceImpl.java

Komponenty

- class com.mklachl.sopkom.services.impl.IIncidentServiceImpl

Pakiety

- package com.mklachl.sopkom.services.impl

7.180 IncidentServiceImpl.java

Idź do dokumentacji tego pliku.

```
00001 package com.mklachl.sopkom.services.impl;  
00002  
00003 import java.util.Date;  
00004 import java.util.List;  
00005 import java.util.Optional;  
00006  
00007 import org.springframework.beans.factory.annotation.Autowired;  
00008 import org.springframework.stereotype.Service;  
00009 import org.thymeleaf.util.DateUtils;  
00010  
00011 import com.mklachl.sopkom.exceptions.IncydentException;  
00012 import com.mklachl.sopkom.model.dto.IncydentDto;  
00013 import com.mklachl.sopkom.model.entity.Autobus;  
00014 import com.mklachl.sopkom.model.entity.Incydent;  
00015 import com.mklachl.sopkom.model.entity.Kierowca;  
00016 import com.mklachl.sopkom.model.entity.Przejazd;  
00017 import com.mklachl.sopkom.raporty.rozkład.DateHelper;  
00018 import com.mklachl.sopkom.repository.AutobusRepository;  
00019 import com.mklachl.sopkom.repository.IncydentRepository;  
00020 import com.mklachl.sopkom.repository.KierowcaRepository;  
00021 import com.mklachl.sopkom.repository.PrzejazdRepository;  
00022 import com.mklachl.sopkom.services.IncydentService;  
00023  
00024 @Service  
00025 public class IncydentServiceImpl implements IncydentService {  
00026  
00027     private final IncydentRepository incydentRepository;  
00028     private final AutobusRepository autobusRepository;  
00029     private final KierowcaRepository kierowcaRepository;  
00030     private final PrzejazdRepository przejazdRepository;
```

```

00031
00032     @Autowired
00033     public IncydentServiceImpl(IncidentRepository incidentRepository, AutobusRepository
00034                                         autobusRepository,
00035                                         KierowcaRepository kierowcaRepository, PrzejazdRepository
00036                                         przejazdRepository) {
00037         this.incidentRepository = incidentRepository;
00038         this.autobusRepository = autobusRepository;
00039         this.kierowcaRepository = kierowcaRepository;
00040         this.przejazdRepository = przejazdRepository;
00041     }
00042     @Override
00043     public Incident saveIncident(Incident incidentDto) {
00044         Incident incident = new Incident();
00045         incident.setKrotko(incidentDto.getKrotko());
00046         incident.setTyp(incidentDto.getTyp());
00047         incident.setDate(incidentDto.getDate());
00048         incident.setAutobus(incidentDto.getAutobus());
00049         incident.setKierowca(incidentDto.getKierowca());
00050         incident.setPrzejazd(incidentDto.getPrzejazd());
00051         incident.setDodatkoweInformacje(incidentDto.getDodatkoweInformacje());
00052         incident.setKoszty(incidentDto.getKoszty());
00053
00054         incident = incidentRepository.save(incident);
00055         return incident;
00056     }
00057     public Incident saveIncident(IncidentDto incidentDto) throws IncidentException {
00058         Incident incident = new Incident();
00059         incident.setTyp(incidentDto.getTyp());
00060         incident.setDate(incidentDto.getDate());
00061         incident.setKrotko(incidentDto.getKrotko());
00062
00063         if (incidentDto.getAutobusId() != null) {
00064             Autobus autobus = autobusRepository.findById(incidentDto.getAutobusId())
00065                 .orElseThrow(() -> new IncidentException("Autobus not found"));
00066             incident.setAutobus(autobus);
00067         }
00068
00069         if (incidentDto.getKierowcaId() != null) {
00070             Kierowca kierowca = kierowcaRepository.findById(incidentDto.getKierowcaId())
00071                 .orElseThrow(() -> new IncidentException("Kierowca not found"));
00072             incident.setKierowca(kierowca);
00073         }
00074
00075         if (incidentDto.getPrzejazdId() != null) {
00076             Przejazd przejazd = przejazdRepository.findById(incidentDto.getPrzejazdId())
00077                 .orElseThrow(() -> new IncidentException("Przejazd not found " +
00078                     incidentDto.getPrzejazdId()));
00079             incident.setPrzejazd(przejazd);
00080         }
00081         incident.setDodatkoweInformacje(incidentDto.getDodatkoweInformacje());
00082         incident.setKoszty(incidentDto.getKoszty());
00083
00084         incident = incidentRepository.save(incident);
00085         return incident;
00086     }
00087     @Override
00088     public Incident updateIncident(long id, IncidentDto updatedIncidentDto) throws IncidentException {
00089
00090         Incident incidentToUpdate = incidentRepository.findById(id)
00091             .orElseThrow(() -> new IncidentException("Incident not found"));
00092
00093         incidentToUpdate.setDodatkoweInformacje(updatedIncidentDto.getDodatkoweInformacje());
00094
00095         incidentToUpdate.setTyp(updatedIncidentDto.getTyp());
00096         incidentToUpdate.setDate(DateHelper.addDays(updatedIncidentDto.getDate(), 1));
00097         incidentToUpdate.setKrotko(updatedIncidentDto.getKrotko());
00098
00099
00100        if (updatedIncidentDto.getAutobusId() != null) {
00101            Autobus autobus = autobusRepository.findById(updatedIncidentDto.getAutobusId())
00102                .orElseThrow(() -> new IncidentException("Autobus not found"));
00103            incidentToUpdate.setAutobus(autobus);
00104        } else {
00105            incidentToUpdate.setAutobus(null);
00106        }
00107
00108        if (updatedIncidentDto.getKierowcaId() != null) {
00109            Kierowca kierowca = kierowcaRepository.findById(updatedIncidentDto.getKierowcaId())
00110                .orElseThrow(() -> new IncidentException("Kierowca not found"));
00111            incidentToUpdate.setKierowca(kierowca);
00112        } else {
00113            incidentToUpdate.setKierowca(null);
00114        }

```

```
00115         }
00116
00117     if (updatedIncydentDto.getPrzejazdId() != null) {
00118         Przejazd przejazd = przejazdRepository.findById(updatedIncydentDto.getPrzejazdId())
00119             .orElseThrow(() -> new IncydentException("Przejazd not found"));
00120         incydentToUpdate.setPrzejazd(przejazd);
00121     } else {
00122         incydentToUpdate.setPrzejazd(null);
00123     }
00124
00125     incydentToUpdate.setKoszty(updatedIncydentDto.getKoszty());
00126
00127     incydentToUpdate = incydentRepository.save(incydentToUpdate);
00128
00129     return incydentToUpdate;
00130 }
00131
00132 @Override
00133 public Optional<Incydent> findIncydentById(long id){
00134     return incydentRepository.findById(id);
00135 }
00136
00137 @Override
00138 public List<Incydent> findIncydentByTyp(String typ){
00139     return incydentRepository.findAllByTyp(typ);
00140 }
00141
00142 @Override
00143 public List<Incydent> findIncydentByDateBefore(Date date){
00144     return incydentRepository.findAllByDateBefore(date);
00145 }
00146
00147 @Override
00148 public List<Incydent> findIncydentByDateAfter(Date date){
00149     return incydentRepository.findAllByDateAfter(date);
00150 }
00151
00152 @Override
00153 public List<Incydent> findIncydentByDateBetween(Date date1, Date date2){
00154     return incydentRepository.findAllByDateBetween(date1, date2);
00155 }
00156
00157 @Override
00158 public List<Incydent> findIncydentByDate(Date date){
00159     return incydentRepository.findAllByDate(date);
00160 }
00161
00162 @Override
00163 public List<Incydent> findIncydentByAutobus(Autobus autobus){
00164     return incydentRepository.findAllByAutobus(autobus);
00165 }
00166
00167 @Override
00168 public List<Incydent> findIncydentByKierowca(Kierowca kierowca){
00169     return incydentRepository.findAllByKierowca(kierowca);
00170 }
00171
00172 @Override
00173 public List<Incydent> findIncydentByPrzejazd(Przejazd przejazd){
00174     return incydentRepository.findAllByPrzejazd(przejazd);
00175 }
00176
00177 @Override
00178 public List<Incydent> findIncydentByDodatkoweInformacje(String dodatkoweInformacje){
00179     return incydentRepository.findAllByDodatkoweInformacje(dodatkoweInformacje);
00180 }
00181
00182 @Override
00183 public List<Incydent> findIncydentByKosztyLessThan(float koszty){
00184     return incydentRepository.findAllByKosztyLessThan(koszty);
00185 }
00186
00187 @Override
00188 public List<Incydent> findIncydentByKosztyGreaterThanOrEqual(float koszty){
00189     return incydentRepository.findAllByKosztyGreaterThanOrEqual(koszty);
00190 }
00191
00192 @Override
00193 public List<Incydent> findIncydentByKoszty(float koszty){
00194     return incydentRepository.findAllByKoszty(koszty);
00195 }
00196 }
```

7.181 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ Backend/src/main/java/com/mklachl/sopkom/services/impl/ KierowcaServiceImpl.java

Komponenty

- class [com.mklachl.sopkom.services.impl.KierowcaServiceImpl](#)

Pakiety

- package [com.mklachl.sopkom.services.impl](#)

7.182 KierowcaServiceImpl.java

[Idź do dokumentacji tego pliku.](#)

```

00001 package com.mklachl.sopkom.services.impl;
00002
00003 import java.util.Date;
00004 import java.util.List;
00005 import java.util.Optional;
00006
00007 import com.mklachl.sopkom.model.dto.KierowcaDto;
00008 import com.mklachl.sopkom.model.dto.UserDto;
00009 import com.mklachl.sopkom.model.entity.Kierowca;
00010 import com.mklachl.sopkom.model.entity.User;
00011 import com.mklachl.sopkom.repository.KierowcaRepository;
00012 import com.mklachl.sopkom.services.KierowcaService;
00013 import com.mklachl.sopkom.services.UserService;
00014 import org.springframework.data.repository.query.Param;
00015
00016 public class KierowcaServiceImpl implements KierowcaService {
00017     private KierowcaRepository kierowcaRepository;
00018     private UserService userService;
00019
00020     public KierowcaServiceImpl(KierowcaRepository kierowcaRepository, UserService userService) {
00021         this.kierowcaRepository = kierowcaRepository;
00022         this.userService = userService;
00023     }
00024
00025     @Override
00026     public Kierowca saveKierowca(Kierowca kierowcaDto) {
00027         Kierowca kierowca = new Kierowca();
00028         kierowca.setPrawoJazdyWazneDo(kierowcaDto.getPrawoJazdyWazneDo());
00029         kierowca.setImie(kierowcaDto.getImie());
00030         kierowca.setNazwisko(kierowcaDto.getNazwisko());
00031         kierowca.setPesel(kierowcaDto.getPesel());
00032         kierowca.setUser(kierowcaDto.getUser());
00033
00034         kierowca=kierowcaRepository.save(kierowca);
00035         return kierowca;
00036     }
00037
00038     public Kierowca saveKierowca(KierowcaDto kierowcaDto) {
00039         Kierowca kierowca = new Kierowca();
00040         kierowca.setPrawoJazdyWazneDo(kierowcaDto.getPrawoJazdyWazneDo());
00041         kierowca.setImie(kierowcaDto.getImie());
00042         kierowca.setNazwisko(kierowcaDto.getNazwisko());
00043         kierowca.setPesel(kierowcaDto.getPesel());
00044
00045         if(kierowcaDto.getUser() != null) {
00046             User userIn = userService.findById(kierowcaDto.getUser().getId());
00047             kierowca.setUser(userIn);
00048         }
00049
00050         kierowca = kierowcaRepository.save(kierowca);
00051         return kierowca;
00052     }
00053 }
00054

```

```
00055     @Override
00056     public Kierowca findKierowcaByPesel(String pesel){
00057         return kierowcaRepository.findByPesel(pesel);
00058     }
00059
00060     @Override
00061     public Kierowca findKierowcaByUser(User user){
00062         return kierowcaRepository.findByUser(user);
00063     }
00064
00065     @Override
00066     public List<Kierowca> findKierowcaByImieAndNazwisko(String imie, String nazwisko){
00067         return kierowcaRepository.findAllByImieAndNazwisko(imie, nazwisko);
00068     }
00069
00070     @Override
00071     public List<Kierowca> findKierowcaByPrawoJazdyWazneDoBefore(Date date){
00072         return kierowcaRepository.findAllByPrawoJazdyWazneDoAfter(date);
00073     }
00074
00075     @Override
00076     public List<Kierowca> findKierowcaByPrawoJazdyWazneDoAfter(Date date){
00077         return kierowcaRepository.findAllByPrawoJazdyWazneDoAfter(date);
00078     }
00079
00080     @Override
00081     public Optional<Kierowca> findKierowcaById(Long id){
00082         return kierowcaRepository.findById(id);
00083     }
00084
00085     @Override
00086     public List<Kierowca> findKierowcaByAvailability(Date startTime, Date endTime){ return
00087         kierowcaRepository.findAllByAvailability(startTime, endTime); }
```

7.183 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ ← Backend/src/main/java/com/mklachl/sopkom/services/impl/Kurs← ServiceImpl.java

Komponenty

- class [com.mklachl.sopkom.services.impl.KursServiceImpl](#)

Pakiety

- package [com.mklachl.sopkom.services.impl](#)

7.184 KursServiceImpl.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.services.impl;
00002
00003 import java.text.SimpleDateFormat;
00004 import java.util.ArrayList;
00005 import java.util.Comparator;
00006 import java.util.Date;
00007 import java.util.LinkedList;
00008 import java.util.List;
00009 import java.util.Optional;
00010 import java.util.stream.Collectors;
00011
00012 import org.springframework.stereotype.Service;
00013
00014 import com.mklachl.sopkom.model.dto.kurs.AbstractKursDto;
00015 import com.mklachl.sopkom.model.dto.kurs.InputKursDto;
```

```
00016 import com.mklachl.sopkom.model.entity.Harmonogram;
00017 import com.mklachl.sopkom.model.entity.Kurs;
00018 import com.mklachl.sopkom.model.entity.KursPrzystanekWlini;
00019 import com.mklachl.sopkom.model.entity.Linia;
00020 import com.mklachl.sopkom.model.entity.PrzystanekWlini;
00021 import com.mklachl.sopkom.raporty.rozkład.DateHelper;
00022 import com.mklachl.sopkom.repository.KursPrzystanekWliniRepository;
00023 import com.mklachl.sopkom.repository.KursRepository;
00024 import com.mklachl.sopkom.repository.PrzystanekWliniRepository;
00025 import com.mklachl.sopkom.services.HarmonogramService;
00026 import com.mklachl.sopkom.services.KursService;
00027 import com.mklachl.sopkom.services.LiniaService;
00028
00029 @Service
00030 public class KursServiceImpl implements KursService {
00031
00032     private KursRepository kursRepository;
00033
00034     private LiniaService liniaService;
00035
00036     private HarmonogramService harmonogramService;
00037
00038     private KursPrzystanekWliniRepository kpwlRepository;
00039
00040     private PrzystanekWliniRepository pwlRepository;
00041
00042
00043     public KursServiceImpl(KursRepository kursRepository, LiniaService liniaService,
00044             HarmonogramService harmonogramService, KursPrzystanekWliniRepository kpwlRepository,
00045             PrzystanekWliniRepository pwlRepository) {
00046         super();
00047         this.kursRepository = kursRepository;
00048         this.liniaService = liniaService;
00049         this.harmonogramService = harmonogramService;
00050         this.kpwlRepository = kpwlRepository;
00051         this.pwlRepository = pwlRepository;
00052     }
00053
00054     @Override
00055     public Kurs saveKurs(Kurs kursDto){
00056         Kurs kurs = new Kurs();
00057         kurs.setLinia(kursDto.getLinia());
00058         kurs.setHarmonogram(kursDto.getHarmonogram());
00059         kurs.setKierunek(kursDto.getKierunek());
00060         kurs.setTypAutobusu(kursDto.getTypAutobusu());
00061
00062
00063
00064
00065
00066
00067
00068
00069
00070
00071         kurs = kursRepository.save(kurs);
00072         return kurs;
00073     }
00074
00075     @jakarta.transaction.Transactional(rollbackOn = {Exception.class})
00076     public Kurs saveKurs(InputKursDto kursDto) throws Exception{
00077         Kurs kurs = new Kurs();
00078
00079
00080         Optional<Linia> linia = liniaService.findLiniaById(kursDto.getLiniaId());
00081
00082         if(linia.isEmpty()) {
00083             throw new Exception("linia not found!");
00084         }
00085
00086         Optional<Harmonogram> harmonogram =
00087             harmonogramService.findHarmonogramById(kursDto.getHarmonogramId());
00088
00089         if(harmonogram.isEmpty()) {
00090             throw new Exception("harmonogram not found!");
00091         }
00092
00093         kurs.setLinia(linia.get());
00094         kurs.setHarmonogram(harmonogram.get());
00095         kurs.setKierunek(kursDto.getKierunek());
00096         kurs.setTypAutobusu(kursDto.getTyp_autobusu());
00097         kurs = kursRepository.save(kurs);
00098
00099         var data = new ArrayList<KursPrzystanekWlini>();
00100
00101         for(var prz: kursDto.getPrzystanki()) {
```

```
00102         KursPrzystanekWlini kpwl = new KursPrzystanekWlini();
00103         PrzystanekWlini pwl =
00104             pwlRepository.findById(prz.getPrzystanekWLini()).getPrzystanekwliniId().get();
00105         kpwl.setGodzinna(prz.getGodzinna());
00106         kpwl.setPrzystanekWLini(pwl);
00107         kpwl.setKurs(kurs);
00108         kpwl = kpwlRepository.save(kpwl);
00109         data.add(kpwl);
00110     }
00111 }
00112 kurs.setKursPrzystanekWlinii(data);
00113 kurs = kursRepository.save(kurs);
00115
00116     return kurs;
00117 }
00118
00119 @jakarta.transaction.Transactional(rollbackOn = {Exception.class})
00120 public Kurs updateKurs(InputKursDto kursDto) throws Exception {
00121
00122     var kursO = kursRepository.findById(kursDto.getKursId());
00123
00124     if(kursO.isEmpty()) {
00125         throw new Exception("kurs not found!");
00126     }
00127
00128     Optional<Linia> linia = liniaService.findLiniaById(kursDto.getLiniaId());
00129
00130     if(linia.isEmpty()) {
00131         throw new Exception("linia not found!");
00132     }
00133
00134     Optional<Harmonogram> harmonogram =
00135         harmonogramService.findHarmonogramById(kursDto.getHarmonogramId());
00136
00137     if(harmonogram.isEmpty()) {
00138         throw new Exception("harmonogram not found!");
00139     }
00140
00141     Kurs kurs = kursO.get();
00142
00143     kurs.setLinia(linia.get());
00144     kurs.setHarmonogram(harmonogram.get());
00145     kurs.setKierunek(kursDto.getKierunek());
00146     kurs.setTypAutobusu(kursDto.getTyp_autobusu());
00147     kurs = kursRepository.save(kurs);
00148
00149 //var data = new ArrayList<KursPrzystanekWlini>();
00150
00151     for(int i = 0; i < kursDto.getPrzystanki().size(); i++) {
00152         var dtoIn = kursDto.getPrzystanki().get(i);
00153         KursPrzystanekWlini kpwl = kurs.getKursPrzystanekWlinii().stream().filter((k) ->
00154             k.getKursPrzystanekWliniId().equals(dtoIn.getPrzystanekwKursieId())).findFirst().get();
00155
00156         kpwl.setGodzinna(dtoIn.getGodzinna());
00157         PrzystanekWlini pwl =
00158             pwlRepository.findById(dtoIn.getPrzystanekWLini().getPrzystanekwliniId()).get();
00159
00160         kpwl.setPrzystanekWLini(pwl);
00161     }
00162
00163
00164     return kursRepository.findById(kurs.getKursId()).get();
00165 }
00166
00167 @Override
00168 public List<Kurs> findKursByLinia(Linia linia){return kursRepository.findAllByLinia(linia);}
00169
00170 @Override
00171 public List<Kurs> findKursByHarmonogram(Harmonogram harmonogram){return
00172     kursRepository.findAllByHarmonogram(harmonogram);}
00173
00174 @Override
00175     public List<Kurs> findKursByKierunek(Short kierunek){return
00176     kursRepository.findAllByKierunek(kierunek);}
00177
00178 @Override
00179     public List<Kurs> findKursByTypAutobusu(Short typAutobusu){return
00180     kursRepository.findAllByTypAutobusu(typAutobusu);}
00181 }
```

7.185 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/services/impl/Linia← ServiceImpl.java

Komponenty

- class [com.mklachl.sopkom.services.impl.LiniaServiceImpl](#)

Pakiety

- package [com.mklachl.sopkom.services.impl](#)

7.186 LiniaServiceImpl.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.services.impl;
00002
00003 import com.mklachl.sopkom.model.entity.Linia;
00004 import com.mklachl.sopkom.model.entity.Przystanek;
00005 import com.mklachl.sopkom.model.entity.PrzystanekWlini;
00006 import com.mklachl.sopkom.exceptions.LiniaNotFoundException;
00007 import com.mklachl.sopkom.exceptions.PrzystanekNotFoundException;
00008 import com.mklachl.sopkom.model.dto.linia.LiniaDtoInput;
00009 import com.mklachl.sopkom.model.dto.linia.LiniaDtoOutput;
00010 import com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLiniaInput;
00011 import com.mklachl.sopkom.repository.LiniaRepository;
00012 import com.mklachl.sopkom.repository.PrzystanekRepository;
00013 import com.mklachl.sopkom.repository.PrzystanekWliniRepository;
00014 import com.mklachl.sopkom.services.LiniaService;
00015 import com.mklachl.sopkom.services.PrzystanekService;
00016
00017 import jakarta.transaction.Transactional;
00018
00019 import org.springframework.beans.factory.annotation.Autowired;
00020 import org.springframework.stereotype.Service;
00021
00022 import java.util.ArrayList;
00023 import java.util.List;
00024 import java.util.Optional;
00025
00026 @Service
00027 public class LiniaServiceImpl implements LiniaService {
00028     @Autowired
00029     private LiniaRepository liniaRepository;
00030
00031     @Autowired
00032     private PrzystanekRepository przystanekRepository;
00033
00034     @Autowired
00035     private PrzystanekWliniRepository przystanekWLiniRepository;
00036
00037
00038     @Autowired
00039     private PrzystanekService przystanekService;
00040
00041
00042
00043     @Override
00044     @Transactional(rollbackOn = {PrzystanekNotFoundException.class, LiniaNotFoundException.class})
00045     public Linia saveLinia(LiniaDtoInput liniaDto) throws PrzystanekNotFoundException {
00046         // Linia linia = mapToEntity(liniaDto);
00047         // linia = liniaRepository.save(linia);
00048
00049         Linia nowa = mapToEntity(liniaDto);
00050
00051         nowa = liniaRepository.save(nowa);
00052
00053         List<PrzystanekWlini> przystankiIn = new ArrayList<PrzystanekWlini>();
00054
```

```
00055
00056
00057     for(int i = 0; i < liniaDto.getPrzystanki().size(); i++) {
00058
00059         PrzystanekDtoDlaLiniaInput przystanekRaw = liniaDto.getPrzystanki().get(i);
00060
00061         var przystanek = przystanekService.findPrzystanekById(przystanekRaw.getPrzystanekId());
00062         if(przystanek.isPresent()) {
00063
00064             PrzystanekWlini lacznik = new PrzystanekWlini();
00065             lacznik.setKolejnosc(przystanekRaw.getKolejnosc());
00066             lacznik.setLinia(nowa);
00067             lacznik.setPrzystanek(przystanek.get());
00068
00069             przystanekRepository.save(przystanek.get());
00070
00071             przystankiIn.add(przystanekWLiniRepository.save(lacznik));
00072
00073         } else {
00074             throw new PrzystanekNotFoundException(nowa.getLiniaId(),
00075                 przystanekRaw.getPrzystanekId());
00076         }
00077     }
00078 }
00079
00080     nowa.setPrzystankiWlini(przystankiIn);
00081
00082     return liniaRepository.save(nowa);
00083 }
00084
00085
00086     @Override
00087     @Transactional(rollbackOn = {PrzystanekNotFoundException.class, LiniaNotFoundException.class})
00088     public Linia patchLinia(LiniaDtoInput liniaDto) throws PrzystanekNotFoundException,
00089     LiniaNotFoundException {
00090
00091         var ist = liniaRepository.findById(liniaDto.getId());
00092
00093
00094         if(ist.isEmpty()) {
00095             throw new LiniaNotFoundException(liniaDto.getId());
00096         }
00097
00098         Linia nowa = ist.get();
00099
00100         if(nowa.getPrzystankiWlini() != null) {
00101
00102             przystanekWLiniRepository.deleteAll(nowa.getPrzystankiWlini());
00103
00104             nowa.getPrzystankiWlini().clear();
00105             nowa = liniaRepository.save(nowa);
00106         }
00107
00108         List<PrzystanekWlini> przystankiIn = new ArrayList<PrzystanekWlini>();
00109
00110         for(int i = 0; i < liniaDto.getPrzystanki().size(); i++) {
00111
00112             PrzystanekDtoDlaLiniaInput przystanekRaw = liniaDto.getPrzystanki().get(i);
00113
00114             var przystanek = przystanekService.findPrzystanekById(przystanekRaw.getPrzystanekId());
00115             if(przystanek.isPresent()) {
00116
00117                 PrzystanekWlini lacznik = new PrzystanekWlini();
00118                 lacznik.setKolejnosc(przystanekRaw.getKolejnosc());
00119                 lacznik.setLinia(nowa);
00120                 lacznik.setPrzystanek(przystanek.get());
00121
00122                 przystanekRepository.save(przystanek.get());
00123
00124                 przystankiIn.add(przystanekWLiniRepository.save(lacznik));
00125
00126             } else {
00127                 throw new PrzystanekNotFoundException(nowa.getLiniaId(),
00128                     przystanekRaw.getPrzystanekId());
00129             }
00130         }
00131
00132         nowa.setPrzystankiWlini(przystankiIn);
00133         nowa.setNumer(liniaDto.getNumer());
00134
00135
00136         return liniaRepository.save(nowa);
00137     }
00138 }
```

```

00139     @Override
00140     public void deleteLinia(Long id) {
00141         liniaRepository.deleteById(id);
00142     }
00143
00144     @Override
00145     public Optional<Linia> findLiniaById(Long id) {
00146         return liniaRepository.findById(id);
00147     }
00148
00149     @Override
00150     public Linia findLiniaByNumer(String numer) {
00151         return liniaRepository.findByNumer(numer);
00152     }
00153
00154     private LiniaDtoOutput mapToDto(Linia linia) {
00155         LiniaDtoOutput dto = new LiniaDtoOutput();
00156         dto.setId(linia.getId());
00157         dto.setNumer(linia.getNumer());
00158         return dto;
00159     }
00160
00161     private Linia mapToEntity(LiniaDtoOutput dto) {
00162         Linia linia = new Linia();
00163         linia.setLiniaId(dto.getId());
00164         linia.setNumer(dto.getNumer());
00165         return linia;
00166     }
00167
00168
00169
00170     private Linia mapToEntity(LiniaDtoInput dto) {
00171         Linia linia = new Linia();
00172         linia.setLiniaId(dto.getId());
00173         linia.setNumer(dto.getNumer());
00174         return linia;
00175     }
00176 }
00177

```

7.187 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/services/impl/← PrzejazdServiceImpl.java

Komponenty

- class [com.mklachl.sopkom.services.impl.PrzejazdServiceImpl](#)

Pakiety

- package [com.mklachl.sopkom.services.impl](#)

7.188 PrzejazdServiceImpl.java

[Idź do dokumentacji tego pliku.](#)

```

00001 package com.mklachl.sopkom.services.impl;
00002
00003 import org.springframework.data.repository.query.Param;
00004 import org.springframework.stereotype.Service;
00005
00006 import com.mklachl.sopkom.exceptions.LiniaNotFoundException;
00007 import com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoInput;
00008 import com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate;
00009 import com.mklachl.sopkom.model.entity.Autobus;
00010 import com.mklachl.sopkom.model.entity.Harmonogram;

```

```
00011 import com.mklachl.sopkom.model.entity.Kierowca;
00012 import com.mklachl.sopkom.model.entity.Kurs;
00013 import com.mklachl.sopkom.model.entity.KursPrzystanekWlini;
00014 import com.mklachl.sopkom.model.entity.Przejazd;
00015 import com.mklachl.sopkom.model.entity.PrzejazdBilet;
00016 import com.mklachl.sopkom.model.entity.PrzejazdKursPrzystanekWlini;
00017 import com.mklachl.sopkom.raporty.rozklad.DateHelper;
00018 import com.mklachl.sopkom.repository.AutobusRepository;
00019 import com.mklachl.sopkom.repository.BiletRepository;
00020 import com.mklachl.sopkom.repository.KierowcaRepository;
00021 import com.mklachl.sopkom.repository.KursRepository;
00022 import com.mklachl.sopkom.repository.PrzejazdBiletRepository;
00023 import com.mklachl.sopkom.repository.PrzejazdKursPrzystanekWliniRepository;
00024 import com.mklachl.sopkom.repository.PrzejazdRepository;
00025 import com.mklachl.sopkom.services.PrzejazdService;
00026
00027 import jakarta.transaction.Transactional;
00028
00029 import java.text.SimpleDateFormat;
00030 import java.util.ArrayList;
00031 import java.util.Date;
00032 import java.util.LinkedList;
00033 import java.util.List;
00034 import java.util.Optional;
00035 import java.util.function.Supplier;
00036 import java.util.stream.Collectors;
00037
00038 @Service
00039 public class PrzejazdServiceImpl implements PrzejazdService {
00040
00041     private KierowcaRepository kierowcaRepository;
00042     private AutobusRepository autobusRepository;
00043     private KursRepository kursRepository;
00044
00045     private PrzejazdRepository przejazdRepository;
00046     private PrzejazdKursPrzystanekWliniRepository pkpwlRepository;
00047
00048     private PrzejazdBiletRepository pbr;
00049     private BiletRepository br;
00050
00051
00052
00053
00054
00055
00056     public PrzejazdServiceImpl(KierowcaRepository kierowcaRepository, AutobusRepository
autobusRepository,
00057             KursRepository kursRepository, PrzejazdRepository przejazdRepository,
00058             PrzejazdKursPrzystanekWliniRepository pkpwlRepository, PrzejazdBiletRepository pbr,
BiletRepository br) {
00059         super();
00060         this.kierowcaRepository = kierowcaRepository;
00061         this.autobusRepository = autobusRepository;
00062         this.kursRepository = kursRepository;
00063         this.przejazdRepository = przejazdRepository;
00064         this.pkpwlRepository = pkpwlRepository;
00065         this.pbr = pbr;
00066         this.br = br;
00067     }
00068
00069
00070
00071     @Override
00072     public Przejazd savePrzejazd(Przejazd przejazdDto) {
00073         Przejazd przejazd = new Przejazd();
00074
00075         przejazd.setKierowca(przejazdDto.getKierowca());
00076         przejazd.setAutobus(przejazdDto.getAutobus());
00077         przejazd.setKurs(przejazdDto.getKurs());
00078         przejazd.setSpalanie(przejazdDto.getSpalanie());
00079         przejazd.setCenaZaLitr(przejazdDto.getCenaZaLitr());
00080         przejazd.setDlugoscTrasy(przejazdDto.getDlugoscTrasy());
00081         przejazd.setDataStartu(przejazdDto.getDataStartu());
00082         przejazd.setDataKonca(przejazdDto.getDataKonca());
00083         przejazd.setPrzejazdKursPrzystanekWlini(przejazdDto.getPrzejazdKursPrzystanekWlini());
00084         przejazd.setPrzejazdBilet(przejazdDto.getPrzejazdBilet());
00085
00086         przejazd = przejazdRepository.save(przejazd);
00087         return przejazd;
00088     }
00089
00090
00091
00092     /*public Przejazd savePrzejazd(PrzejazdDto przejazdDto) {
00093         Przejazd przejazd = new Przejazd();
00094
00095         przejazd.setKierowca(przejazdDto.getKierowca());
```

```

00096     przejazd.setAutobus(przejazdDto.getAutobus());
00097     przejazd.setKurs(przejazdDto.getKurs());
00098     przejazd.setSpalanie(przejazdDto.getSpalanie());
00099     przejazd.setCenaZaLitr(przejazdDto.getCenaZaLitr());
00100     przejazd.setDlugoscTrasy(przejazdDto.getDlugoscTrasy());
00101     przejazd.setDataStartu(przejazdDto.getDataStartu());
00102     przejazd.setDataKonca(przejazdDto.getDataKonca());
00103     przejazd.setPrzejazdKursPrzystanekWlini(przejazdDto.getPrzejazdKursPrzystanekWlini());
00104     przejazd.setPrzejazdBilet(przejazdDto.getPrzejazdBilet());
00105
00106     przejazd = przejazdRepository.save(przejazd);
00107     return przejazd;
00108 }
00109
00110     @Override
00111     public List<Przejazd> findPrzejazdByKierowca(Kierowca kierowca){return
przejazdRepository.findAllByKierowca(kierowca);}
00112
00113     @Override
00114     public List<Przejazd> findPrzejazdByAutobus(Autobus autobus){return
przejazdRepository.findAllByAutobus(autobus);}
00115
00116     @Override
00117     public List<Przejazd> findPrzejazdByKurs(Kurs kurs){return
przejazdRepository.findAllByKurs(kurs);}
00118
00119     @Override
00120     public List<Przejazd> findPrzejazdBySpalanie(float spalanie){return
przejazdRepository.findAllBySpalanie(spalanie);}
00121
00122     @Override
00123     public List<Przejazd> findPrzejazdBySpalanieLessThan(float spalanie){return
przejazdRepository.findAllBySpalanieLessThan(spalanie);}
00124
00125     @Override
00126     public List<Przejazd> findPrzejazdBySpalanieGreaterThan(float spalanie){return
przejazdRepository.findAllBySpalanieGreaterThan(spalanie);}
00127
00128     @Override
00129     public List<Przejazd> findPrzejazdByCenaZaLitr(float cenaZaLitr){return
przejazdRepository.findAllByCenaZaLitr(cenaZaLitr);}
00130
00131     @Override
00132     public List<Przejazd> findPrzejazdByCenaZaLitrLessThan(float cenaZaLitr){return
przejazdRepository.findAllByCenaZaLitrLessThan(cenaZaLitr);}
00133
00134     @Override
00135     public List<Przejazd> findPrzejazdByCenaZaLitrGreaterThan(float cenaZaLitr){return
przejazdRepository.findAllByCenaZaLitrGreaterThan(cenaZaLitr);}
00136
00137     @Override
00138     public List<Przejazd> findPrzejazdByDlugoscTrasy(float dlugoscTrasy){return
przejazdRepository.findAllByDlugoscTrasy(dlugoscTrasy);}
00139
00140     @Override
00141     public List<Przejazd> findPrzejazdByDlugoscTrasyLessThan(float dlugoscTrasy){return
przejazdRepository.findAllByDlugoscTrasyLessThan(dlugoscTrasy);}
00142
00143     @Override
00144     public List<Przejazd> findPrzejazdByDlugoscTrasyGreaterThan(float dlugoscTrasy){return
przejazdRepository.findAllByDlugoscTrasyGreaterThan(dlugoscTrasy);}
00145
00146     @Override
00147     public List<Przejazd> findPrzejazdByDataStartu(Date dataStartu){return
przejazdRepository.findAllByDataStartu(dataStartu);}
00148
00149     @Override
00150     public List<Przejazd> findPrzejazdByDataStartuBefore(Date dataStartu){return
przejazdRepository.findAllByDataStartuBefore(dataStartu);}
00151
00152     @Override
00153     public List<Przejazd> findPrzejazdByDataStartuAfter(Date dataStartu){return
przejazdRepository.findAllByDataStartuAfter(dataStartu);}
00154
00155     @Override
00156     public List<Przejazd> findPrzejazdByDataKonca(Date dataKonca){return
przejazdRepository.findAllByDataKonca(dataKonca);}
00157
00158     @Override
00159     public List<Przejazd> findPrzejazdByDataKoncaBefore(Date dataKonca){return
przejazdRepository.findAllByDataKoncaBefore(dataKonca);}
00160
00161     @Override
00162     public List<Przejazd> findPrzejazdByDataKoncaAfter(Date dataKonca){return
przejazdRepository.findAllByDataKoncaAfter(dataKonca);}
00163
00164     @Override

```

```
00165     public Optional<Przejazd> findPrzejazdById(long id){return przejazdRepository.findById(id);}
00166
00167     @Override
00168     public List<Przejazd> findPrzejazdyByKursAndData(Kurs kurs, Date data){ return
00169         przejazdRepository.findAllByKursAndData(kurs, data); }
00170
00171     @Override
00172     public Przejazd createDetachedPrzejazd(Kurs kurs, Date dataPrzejazdu) {
00173
00174         Przejazd przejazd = new Przejazd();
00175
00176         przejazd.setKurs(kurs);
00177
00178         przejazd.setData(dataPrzejazdu);
00179
00180         var list = new ArrayList<PrzejazdKursPrzystanekWlini>();
00181
00182         var list2 = kurs.getKursPrzystanekWlini();
00183
00184         if(kurs.getKierunek().shortValue() == 0) {
00185
00185             Date startRaw = DateHelper.copyTimeComponents(
00186                 kurs.getKursPrzystanekWlini().get(0).getGodzinna(),
00187                 dataPrzejazdu);
00188
00189             przejazd.setDataStartu( DateHelper.addMinutesToDate(startRaw, -30));
00190
00191
00192             Date endRaw = DateHelper.copyTimeComponents(
00193                 kurs.getKursPrzystanekWlini().get(list2.size()-1).getGodzinna(),
00194                 dataPrzejazdu);
00195
00196             endRaw = DateHelper.addMinutesToDate(endRaw, 30);
00197
00198             if(endRaw.before(startRaw)) {
00199                 endRaw = DateHelper.addDays(endRaw, 1);
00200             }
00201
00202             przejazd.setDataKonca(endRaw);
00203
00204         } else {
00205
00205             Date startRaw = DateHelper.copyTimeComponents(
00206                 kurs.getKursPrzystanekWlini().get(list2.size()-1).getGodzinna(),
00207                 dataPrzejazdu);
00208
00209             przejazd.setDataStartu( DateHelper.addMinutesToDate(startRaw, -30));
00210
00211
00212             Date endRaw = DateHelper.copyTimeComponents(
00213                 kurs.getKursPrzystanekWlini().get(0).getGodzinna(),
00214                 dataPrzejazdu);
00215
00216             endRaw = DateHelper.addMinutesToDate(endRaw, 30);
00217
00218             if(endRaw.before(startRaw)) {
00219                 endRaw = DateHelper.addDays(endRaw, 1);
00220             }
00221
00222             przejazd.setDataKonca(endRaw);
00223
00224             przejazd.setDataKonca(DateHelper.addMinutesToDate(endRaw, 30));
00225         }
00226
00227
00228         for(int i = 0; i < list2.size(); i++) {
00229
00230             KursPrzystanekWlini kpwl = list2.get(i);
00231             PrzejazdKursPrzystanekWlini pkpwl = new PrzejazdKursPrzystanekWlini();
00232             pkpwl.setKursPrzystanekWlini(kpwl);
00233             pkpwl.setRealnaGodzinna(null);
00234             pkpwl.setPrzejazd(przejazd);
00235
00236
00237             list.add(pkpwl);
00238         }
00239
00240         przejazd.setPrzejazdKursPrzystanekWlini(list);
00241
00242
00243
00244         return przejazd;
00245     }
00246
00247     @Override
00248     public Przejazd savePrzejazd(PrzejazdDtoInput przejazdDto) {
00249
00250         Przejazd przejazd = new Przejazd();
```

```

00251
00252     przejazd.setKierowca(kierowcaRepository.findById(przejazdDto.getKierowcaId()).get());
00253     przejazd.setAutobus(autobusRepository.findById(przejazdDto.getAutobusId()).get());
00254
00255     Kurs kurs = kursRepository.findById(przejazdDto.getKursId()).get();
00256
00257     przejazd.setKurs(kurs);
00258
00259     przejazd.setData(przejazdDto.getDataPrzejazdu());
00260
00261
00262
00263     var list = new ArrayList<PrzejazdKursPrzystanekWlini>();
00264
00265     var list2 = kurs.getKursPrzystanekWlini();
00266
00267     if(kurs.getKierunek().shortValue() == 0) {
00268
00269         Date startRaw = DateHelper.copyTimeComponents(
00270             kurs.getKursPrzystanekWlini().get(0).getGodzinna(),
00271             przejazdDto.getDataPrzejazdu());
00272
00273         przejazd.setDataStartu( DateHelper.addMinutesToDate(startRaw, -30));
00274
00275
00276         Date endRaw = DateHelper.copyTimeComponents(
00277             kurs.getKursPrzystanekWlini().get(list2.size()-1).getGodzinna(),
00278             przejazdDto.getDataPrzejazdu());
00279
00280         endRaw = DateHelper.addMinutesToDate(endRaw, 30);
00281
00282         if(endRaw.before(startRaw)) {
00283             endRaw = DateHelper.addDays(endRaw, 1);
00284         }
00285
00286         przejazd.setDataKonca(endRaw);
00287
00288     } else {
00289         Date startRaw = DateHelper.copyTimeComponents(
00290             kurs.getKursPrzystanekWlini().get(list2.size()-1).getGodzinna(),
00291             przejazdDto.getDataPrzejazdu());
00292
00293         przejazd.setDataStartu( DateHelper.addMinutesToDate(startRaw, -30));
00294
00295
00296         Date endRaw = DateHelper.copyTimeComponents(
00297             kurs.getKursPrzystanekWlini().get(0).getGodzinna(),
00298             przejazdDto.getDataPrzejazdu());
00299
00300         endRaw = DateHelper.addMinutesToDate(endRaw, 30);
00301
00302         if(endRaw.before(startRaw)) {
00303             endRaw = DateHelper.addDays(endRaw, 1);
00304         }
00305
00306         przejazd.setDataKonca(endRaw);
00307
00308         przejazd.setDataKonca(DateHelper.addMinutesToDate(endRaw, 30));
00309     }
00310
00311     przejazdRepository.save(przejazd);
00312
00313     for(int i = 0; i < list2.size(); i++) {
00314
00315         KursPrzystanekWlini kpwl = list2.get(i);
00316         PrzejazdKursPrzystanekWlini pkpwl = new PrzejazdKursPrzystanekWlini();
00317         pkpwl.setKursPrzystanekWlini(kpwl);
00318         pkpwl.setRealnaGodzinna(null);
00319         pkpwl.setPrzejazd(przejazd);
00320
00321         pkpwl = pkpwlRepository.save(pkpwl);
00322         list.add(pkpwl);
00323
00324     }
00325
00326     przejazd.setPrzejazdKursPrzystanekWlini(list);
00327     przejazdRepository.save(przejazd);
00328
00329
00330     return przejazd;
00331 }
00332
00333
00334     @Override
00335     @Transactional(rollbackOn = {LiniaNotFoundException.class})
00336     public Przejazd updatePrzejazd(PrzejazdDtoUpdate przejazdDto) throws Exception {
00337

```

```
00338     Przejazd p = this.findPrzejazdById(przejazdDto.getPrzejazdId()).orElseThrow(new
00339         Supplier<LiniaNotFoundException>() {
00340             @Override
00341             public LiniaNotFoundException get() {
00342                 return new LiniaNotFoundException(przejazdDto.getPrzejazdId());
00343             }
00344         });
00345     );
00346
00347     p.setSpalanie((float)przejazdDto.getSpalanie());
00348     p.setCenaZaLitr((float)przejazdDto.getCenaZaLitr());
00349     p.setDlugoscTrasy((float)przejazdDto.getDlugoszTrasy());
00350
00351     for(int i = 0; i < przejazdDto.getPrzystanki().size(); i++) {
00352         var temp = przejazdDto.getPrzystanki().get(i);
00353         var main = pkpwlRepository.findById(temp.getId()).orElseThrow(() -> new
00354             LiniaNotFoundException(temp.getId()));
00355         main.setRealnaGodzinna(temp.getDate());
00356         pkpwlRepository.save(main);
00357     }
00358
00359     var list = new ArrayList<PrzejazdBilet>();
00360
00361     for(int i= 0; i < przejazdDto.getLiczbaBiletoWNormalnych(); i++) {
00362         PrzejazdBilet pb = new PrzejazdBilet();
00363         pb.setBilet(this.br.findById((short)0).get());
00364         pb.setCenaBiletu(this.br.findById((short)0).get().getCena());
00365         pb.setPrzejazd(p);
00366
00367         pb = pbr.save(pb);
00368
00369         list.add(pb);
00370     }
00371
00372     for(int i= 0; i < przejazdDto.getLiczbaBiletoWUlgowych(); i++) {
00373         PrzejazdBilet pb = new PrzejazdBilet();
00374         pb.setBilet(this.br.findById((short)1).get());
00375         pb.setCenaBiletu(this.br.findById((short)1).get().getCena());
00376         pb.setPrzejazd(p);
00377
00378         pb = pbr.save(pb);
00379
00380         list.add(pb);
00381     }
00382
00383     p.setPrzejazdBilet(list);
00384
00385     return przejazdRepository.save(p);
00386 }
00387
00388
00389     @Override
00390     public List<Date> findAllDatesForKurs(Kurs kurs, int maxCount) {
00391         Harmonogram harmongram = kurs.getHarmonogram();
00392
00393         var result = DateHelper.getAllDatesForHarmonogram(harmongram, maxCount);
00394         var copy = new LinkedList<Date>(result);
00395
00396         var data = this.findPrzejazdByKurs(kurs);
00397
00398         List<Date> dates = data.stream()
00399             .map(item -> item.getDataStartu())
00400             .collect(Collectors.toList());
00401
00402         copy.removeAll(dates);
00403         return copy;
00404     }
00405 }
00406
00407 }
```

7.189 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ Backend/src/main/java/com/mklachl/sopkom/services/impl/ PrzystanekServiceImpl.java

Komponenty

- class [com.mklachl.sopkom.services.impl.PrzystanekServiceImpl](#)

Pakiety

- package [com.mklachl.sopkom.services.impl](#)

7.190 PrzystanekServiceImpl.java

[Idź do dokumentacji tego pliku.](#)

```

00001 package com.mklachl.sopkom.services.impl;
00002
00003 import com.mklachl.sopkom.model.entity.Przystanek;
00004 import com.mklachl.sopkom.model.dto.PrzystanekDto;
00005 import com.mklachl.sopkom.repository.PrzystanekRepository;
00006 import com.mklachl.sopkom.services.PrzystanekService;
00007 import org.springframework.stereotype.Service;
00008
00009 import java.util.List;
00010 import java.util.Optional;
00011
00012 @Service
00013 public class PrzystanekServiceImpl implements PrzystanekService {
00014     private PrzystanekRepository przystanekRepository;
00015
00016     public PrzystanekServiceImpl(PrzystanekRepository przystanekRepository) {this.przystanekRepository
00017 = przystanekRepository;}
00018
00019     @Override
00020     public Przystanek savePrzystanek(Przystanek przystanekDto) {
00021         Przystanek przystanek = new Przystanek();
00022         przystanek.setPrzystanekOdwrótny(przystanekDto.getPrzystanekOdwrótny());
00023         przystanek.setNazwa(przystanekDto.getNazwa());
00024         przystanek.setKodPocztowy(przystanekDto.getKodPocztowy());
00025         przystanek.setMiasto(przystanekDto.getMiasto());
00026         przystanek.setUlica(przystanekDto.getUlica());
00027         przystanek.setDlugoscGeograficzna(przystanekDto.getDlugoscGeograficzna());
00028         przystanek.setSzerokoscGeograficzna(przystanekDto.getSzerokoscGeograficzna());
00029
00030         przystanek = przystanekRepository.save(przystanek);
00031         return przystanek;
00032     }
00033
00034     public Przystanek savePrzystanek(PrzystanekDto przystanekDto) {
00035         Przystanek przystanek = new Przystanek();
00036
00037         if(przystanekDto.getPrzystanekOdwrótny() != null) {
00038
00039             var przystanekOdwrótny =
00040                 przystanekRepository.findById(przystanekDto.getPrzystanekOdwrótny());
00041
00042             if(przystanekOdwrótny.isPresent()) {
00043                 przystanek.setPrzystanekOdwrótny(przystanekOdwrótny.get());
00044
00045             } else {
00046                 System.out.println("no przystanek odwrótny present!");
00047             }
00048
00049         }
00050
00051         przystanek.setNazwa(przystanekDto.getNazwa());

```

```
00053     przystanek.setKodPocztowy(przystanekDto.getKodPocztowy());
00054     przystanek.setMiasto(przystanekDto.getMiasto());
00055     przystanek.setUlica(przystanekDto.getUlica());
00056     przystanek.setDlugoscGeograficzna(przystanekDto.getDlugoscGeograficzna());
00057     przystanek.setSzerokoscGeograficzna(przystanekDto.getSzerokoscGeograficzna());
00058
00059     przystanek = przystanekRepository.save(przystanek);
00060     return przystanek;
00061 }
00062
00063 @Override
00064 public Przystanek findPrzystanekByNazwa(String nazwa){
00065     return przystanekRepository.findByNazwa(nazwa);
00066 }
00067
00068 @Override
00069 public List<Przystanek> findPrzystanekByKodPocztowy(String kodPocztowy){
00070     return przystanekRepository.findAllByKodPocztowy(kodPocztowy);
00071 }
00072
00073 @Override
00074 public List<Przystanek> findPrzystanekByMiasto(String miasto){
00075     return przystanekRepository.findAllByMiasto(miasto);
00076 }
00077
00078 @Override
00079 public List<Przystanek> findPrzystanekByUlica(String ulica){
00080     return przystanekRepository.findAllByUlica(ulica);
00081 }
00082
00083 @Override
00084 public List<Przystanek> findPrzystanekBySzerokoscGeograficzna(String szerokoscGeograficzna){
00085     return przystanekRepository.findAllBySzerokoscGeograficzna(szerokoscGeograficzna);
00086 }
00087
00088 @Override
00089 public List<Przystanek> findPrzystanekByDlugoscGeograficzna(String dlugoscGeograficzna){
00090     return przystanekRepository.findAllByDlugoscGeograficzna(dlugoscGeograficzna);
00091 }
00092
00093 @Override
00094 public Optional<Przystanek> findPrzystanekById(Long id){
00095     return przystanekRepository.findById(id);
00096 }
00097
00098 @Override
00099 public List<Przystanek> findPrzystanekByPrzystanekOdwrzony(Przystanek przystanekOdwrzony){
00100     return przystanekRepository.findAllByPrzystanekOdwrzony(przystanekOdwrzony);
00101 }
00102 }
00103 }
```

7.191 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ ← Backend/src/main/java/com/mklachl/sopkom/services/impl/User← ServiceImpl.java

Komponenty

- class [com.mklachl.sopkom.services.impl.UserServiceImpl](#)

Pakietы

- package [com.mklachl.sopkom.services.impl](#)

7.192 UserServiceImpl.java

[Idź do dokumentacji tego pliku.](#)

```

00001 package com.mklachl.sopkom.services.impl;
00002
00003
00004 import org.springframework.security.crypto.password.PasswordEncoder;
00005 import org.springframework.stereotype.Service;
00006
00007 import com.mklachl.sopkom.model.dto.UserDto;
00008 import com.mklachl.sopkom.model.entity.User;
00009 import com.mklachl.sopkom.repository.UserRepository;
00010 import com.mklachl.sopkom.services.UserService;
00011
00012
00013 @Service
00014 public class UserServiceImpl implements UserService {
00015
00016     private UserRepository userRepository;
00017
00018     private PasswordEncoder passwordEncoder;
00019
00020     public UserServiceImpl(UserRepository userRepository,
00021                           PasswordEncoder passwordEncoder) {
00022         this.userRepository = userRepository;
00023         this.passwordEncoder = passwordEncoder;
00024     }
00025
00026     public void saveUser(UserDto userDto) {
00027
00028         User user = new User();
00029         user.setName(userDto.getName());
00030         user.setLogin(userDto.getEmail());
00031         // we save the user with the appropiate privileges;
00032         user.setRoles(userDto.getRoles());
00033         // encrypt the password using spring security
00034         user.setPassword(passwordEncoder.encode(userDto.getPassword()));
00035         userRepository.save(user);
00036     }
00037
00038     @Override
00039     public void saveUser(User userDto) {
00040
00041         User user = new User();
00042         user.setName(userDto.getName());
00043         user.setLogin(userDto.getLogin());
00044         user.setRoles(userDto.getRoles());
00045
00046         user.setPassword(passwordEncoder.encode(userDto.getPassword()));
00047         userRepository.save(user);
00048     }
00049
00050     @Override
00051     public User findUserByLogin(String login) {
00052         return userRepository.findByLogin(login);
00053     }
00054
00055     @Override
00056     public User findUserById(Long id) {
00057
00058         return userRepository.findById(id).orElse(null);
00059     }
00060 }
00061 }
```

7.193 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/services/Icydent← Service.java

Komponenty

- interface [com.mklachl.sopkom.services.IcydentService](#)

Pakiety

- package com.mklachl.sopkom.services

7.194 IncydentService.java

Idź do dokumentacji tego pliku.

```
00001 package com.mklachl.sopkom.services;
00002
00003 import com.mklachl.sopkom.exceptions.IncydentException;
00004 import com.mklachl.sopkom.model.dto.IncydentDto;
00005 import com.mklachl.sopkom.model.entity.Autobus;
00006 import com.mklachl.sopkom.model.entity.Incydent;
00007 import com.mklachl.sopkom.model.entity.Kierowca;
00008 import com.mklachl.sopkom.model.entity.Przejazd;
00009
00010 import java.util.Date;
00011 import java.util.List;
00012 import java.util.Optional;
00013
00014 public interface IncydentService {
00015
00024     Incydent saveIncydent(Incydent incydent);
00025
00032     Incydent saveIncydent(IncydentDto incydentDto) throws IncydentException;
00033
00041     Incydent updateIncydent(long id, IncydentDto updatedIncydentDto) throws IncydentException;
00042
00048     Optional<Incydent> findIncydentById(long id);
00049
00055     List<Incydent> findIncydentByTyp(String typ);
00056
00062     List<Incydent> findIncydentByDateBefore(Date date);
00063
00069     List<Incydent> findIncydentByDateAfter(Date date);
00070
00077     List<Incydent> findIncydentByDateBetween(Date date1, Date date2);
00078
00084     List<Incydent> findIncydentByDate(Date date);
00085
00091     List<Incydent> findIncydentByAutobus(Autobus autobus);
00092
00098     List<Incydent> findIncydentByKierowca(Kierowca kierowca);
00099
00105     List<Incydent> findIncydentByPrzejazd(Przejazd przejazd);
00106
00112     List<Incydent> findIncydentByDodatkoweInformacje(String dodatkoweInformacje);
00113
00119     List<Incydent> findIncydentByKosztyLessThan(float koszty);
00120
00126     List<Incydent> findIncydentByKosztyGreaterThanOrEqual(float koszty);
00127
00133     List<Incydent> findIncydentByKoszty(float koszty);
00134 }
```

7.195 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/services/Kierowca← Service.java

Komponenty

- interface com.mklachl.sopkom.services.KierowcaService

Pakiety

- package com.mklachl.sopkom.services

7.196 KierowcaService.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.services;
00002
00003 import com.mklachl.sopkom.model.entity.Kierowca;
00004 import com.mklachl.sopkom.model.dto.KierowcaDto;
00005 import com.mklachl.sopkom.model.entity.User;
00006 import org.springframework.data.repository.query.Param;
00007
00008 import java.util.Date;
00009 import java.util.List;
00010 import java.util.Optional;
00011
00015 public interface KierowcaService {
00016
00022     Kierowca saveKierowca(Kierowca kierowca);
00023
00029     Kierowca saveKierowca(KierowcaDto kierowcaDto);
00030
00036     Kierowca findKierowcaByPesel(String pesel);
00037
00043     Kierowca findKierowcaByUser(User user);
00044
00051     List<Kierowca> findKierowcaByImieAndNazwisko(String imie, String nazwisko);
00052
00058     List<Kierowca> findKierowcaByPrawoJazdyWazneDoBefore(Date date);
00059
00065     List<Kierowca> findKierowcaByPrawoJazdyWazneDoAfter(Date date);
00066
00072     Optional<Kierowca> findKierowcaById(Long id);
00073
00080     List<Kierowca> findKierowcaByAvailability(Date startTime, Date endTime);
00081
00082 }
```

7.197 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/services/Kurs← Service.java

Komponenty

- interface [com.mklachl.sopkom.services.KursService](#)

Pakiety

- package [com.mklachl.sopkom.services](#)

7.198 KursService.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.services;
00002
00003 import com.mklachl.sopkom.model.dto.kurs.AbstractKursDto;
00004 import com.mklachl.sopkom.model.dto.kurs.InputKursDto;
00005 import com.mklachl.sopkom.model.dto.kurs.KursDto;
00006 import com.mklachl.sopkom.model.entity.Harmonogram;
00007 import com.mklachl.sopkom.model.entity.Kurs;
00008 import com.mklachl.sopkom.model.entity.Linia;
00009
00010 import java.util.Date;
00011 import java.util.List;
00012 import java.util.Optional;
```

```
00013
00017 public interface KursService {
00018
00024     Kurs saveKurs(Kurs kurs);
00025
00032     Kurs saveKurs(InputKursDto kursDto) throws Exception;
00033
00040     Kurs updateKurs(InputKursDto kursDto) throws Exception;
00041
00047     List<Kurs> findKursByLinia(Linia linia);
00048
00054     List<Kurs> findKursByHarmonogram(Harmonogram harmonogram);
00055
00061     List<Kurs> findKursByKierunek(Short kierunek);
00062
00068     List<Kurs> findKursByTypAutobusu(Short typAutobusu);
00069
00070 }
```

7.199 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ ← Backend/src/main/java/com/mklachl/sopkom/services/Linia← Service.java

Komponenty

- interface [com.mklachl.sopkom.services.LiniaService](#)

Pakiety

- package [com.mklachl.sopkom.services](#)

7.200 LiniaService.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.services;
00002
00003 import com.mklachl.sopkom.model.dto.linia.LiniaDtoInput;
00004 import com.mklachl.sopkom.model.dto.linia.LiniaDtoOutput;
00005 import com.mklachl.sopkom.model.entity.Linia;
00006 import com.mklachl.sopkom.exceptions.LiniaNotFoundException;
00007 import com.mklachl.sopkom.exceptions.PrzystanekNotFoundException;
00008
00009 import java.util.Optional;
00010
00014 public interface LiniaService {
00015
00022     Linia saveLinia(LiniaDtoInput liniaDto) throws PrzystanekNotFoundException;
00023
00031     Linia patchLinia(LiniaDtoInput liniaDto) throws PrzystanekNotFoundException,
00032         LiniaNotFoundException;
00033
00037     void deleteLinia(Long id);
00038
00044     Optional<Linia> findLiniaById(Long id);
00045
00051     Linia findLiniaByNumer(String numer);
00052 }
```

7.201 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ Backend/src/main/java/com/mklachl/sopkom/services/Przejazd Service.java

Komponenty

- interface [com.mklachl.sopkom.services.PrzejazdService](#)

Pakiety

- package [com.mklachl.sopkom.services](#)

7.202 PrzejazdService.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.services;
00002
00003 import com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoInput;
00004 import com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate;
00005 import com.mklachl.sopkom.model.entity.Autobus;
00006 import com.mklachl.sopkom.model.entity.Kierowca;
00007 import com.mklachl.sopkom.model.entity.Kurs;
00008 import com.mklachl.sopkom.model.entity.Przejazd;
00009
00010 import java.util.Date;
00011 import java.util.List;
00012 import java.util.Optional;
00013
00017 public interface PrzejazdService {
00018
00024     Przejazd savePrzejazd(Przejazd przejazd);
00025
00031     Przejazd savePrzejazd(PrzejazdDtoInput przejazdDto);
00032
00039     Przejazd updatePrzejazd(PrzejazdDtoUpdate przejazdDto) throws Exception;
00040
00046     List<Przejazd> findPrzejazdByKierowca(Kierowca kierowca);
00047
00053     List<Przejazd> findPrzejazdByAutobus(Autobus autobus);
00054
00060     List<Przejazd> findPrzejazdByKurs(Kurs kurs);
00061
00067     List<Przejazd> findPrzejazdBySpalanie(float spalanie);
00068
00074     List<Przejazd> findPrzejazdBySpalanieLessThan(float spalanie);
00075
00081     List<Przejazd> findPrzejazdBySpalanieGreaterThan(float spalanie);
00082
00088     List<Przejazd> findPrzejazdByCenaZaLitr(float cenaZaLitr);
00089
00095     List<Przejazd> findPrzejazdByCenaZaLitrLessThan(float cenaZaLitr);
00096
00102     List<Przejazd> findPrzejazdByCenaZaLitrGreaterThan(float cenaZaLitr);
00103
00109     List<Przejazd> findPrzejazdByDlugoscTrasy(float dlugoscTrasy);
00110
00116     List<Przejazd> findPrzejazdByDlugoscTrasyLessThan(float dlugoscTrasy);
00117
00123     List<Przejazd> findPrzejazdByDlugoscTrasyGreaterThan(float dlugoscTrasy);
00124
00130     List<Przejazd> findPrzejazdByDataStartu(Date dataStartu);
00131
00137     List<Przejazd> findPrzejazdByDataStartuBefore(Date dataStartu);
00138
00144     List<Przejazd> findPrzejazdByDataStartuAfter(Date dataStartu);
00145
00151     List<Przejazd> findPrzejazdByDataKonca(Date dataKonca);
00152
00158     List<Przejazd> findPrzejazdByDataKoncaBefore(Date dataKonca);
```

```
00159  
00165     List<Przejazd> findPrzejazdByDataKoncaAfter(Date dataKonca);  
00166  
00173     List<Przejazd> findPrzejazdyByKursAndData(Kurs kurs, Date data);  
00174  
00180     Optional<Przejazd> findPrzejazdById(long id);  
00181  
00188     List<Date> findAllDatesForKurs(Kurs kurs, int count);  
00189  
00196     Przejazd createDetachedPrzejazd(Kurs kurs, Date dataPrzejazdu);  
00197  
00198 }
```

7.203 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_ ← Backend/src/main/java/com/mklachl/sopkom/services/← PrzystanekService.java

Komponenty

- interface [com.mklachl.sopkom.services.PrzystanekService](#)

Pakiety

- package [com.mklachl.sopkom.services](#)

7.204 PrzystanekService.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.services;  
00002  
00003 import com.mklachl.sopkom.model.entity.Przystanek;  
00004 import com.mklachl.sopkom.model.dto.PrzystanekDto;  
00005  
00006 import java.util.List;  
00007 import java.util.Optional;  
00008  
00012 public interface PrzystanekService {  
00013  
00019     Przystanek savePrzystanek(Przystanek przystanek);  
00020  
00026     Przystanek savePrzystanek(PrzystanekDto przystanekDto);  
00027  
00033     Przystanek findPrzystanekByNazwa(String nazwa);  
00034  
00040     List<Przystanek> findPrzystanekByKodPocztowy(String kodPocztowy);  
00041  
00047     List<Przystanek> findPrzystanekByMiasto(String miasto);  
00048  
00054     List<Przystanek> findPrzystanekByUlica(String ulica);  
00055  
00061     List<Przystanek> findPrzystanekBySzerokoscGeograficzna(String szerokoscGeograficzna);  
00062  
00068     List<Przystanek> findPrzystanekByDlugoscGeograficzna(String dlugoscGeograficzna);  
00069  
00075     Optional<Przystanek> findPrzystanekById(Long id);  
00076  
00082     List<Przystanek> findPrzystanekByPrzystanekOdwrotny(Przystanek przystanekOdwrotny);  
00083 }
```

7.205 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_↳ Backend/src/main/java/com/mklachl/sopkom/services/User↳ Service.java

Komponenty

- interface [com.mklachl.sopkom.services.UserService](#)

Pakietы

- package [com.mklachl.sopkom.services](#)

7.206 UserService.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.services;
00002
00003 import com.mklachl.sopkom.model.dto.UserDto;
00004 import com.mklachl.sopkom.model.entity.User;
00005
00009 public interface UserService {
00010
00015     void saveUser(User user);
00016
00021     void saveUser(UserDto userDto);
00022
00028     User findUserByLogin(String email);
00029
00035     User findUserById(Long id);
00036
00037 }
```

7.207 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_↳ Backend/src/main/java/com/mklachl/sopkom/SOPKoMApplication.java

Komponenty

- class [com.mklachl.sopkom.SOPKoMApplication](#)

Pakietы

- package [com.mklachl.sopkom](#)

7.208 SOPKoMApplication.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom;
00002
00003 import org.springframework.boot.SpringApplication;
00004 import org.springframework.boot.autoconfigure.SpringBootApplication;
00005
00006 @SpringBootApplication
00007 public class SOPKoMApplication {
00008
00009     public static void main(String[] args) {
00010         SpringApplication.run(SOPKoMApplication.class, args);
00011     }
00012
00013 }
```

7.209 Dokumentacja pliku X:/java/TAB/Doxygen/2024_TAB_S9_LACH- incydenty-2-comments/SOPKoM_← Backend/src/main/java/com/mklachl/sopkom/topdf/Pdf.java

Komponenty

- class [com.mklachl.sopkom.topdf.Pdf](#)
- class [com.mklachl.sopkom.topdf.Pdf.\\$](#)

Pakiety

- package [com.mklachl.sopkom.topdf](#)

7.210 Pdf.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom.topdf;
00002
00003 import com.itextpdf.text.*;
00004 import com.itextpdf.text.pdf.BaseFont;
00005 import com.itextpdf.text.pdf.PdfPCell;
00006 import com.itextpdf.text.pdf.PdfPTable;
00007 import com.itextpdf.text.pdf.PdfWriter;
00008 import com.mklachl.sopkom.model.dto.linia.LiniaDto;
00009 import com.mklachl.sopkom.model.entity.*;
00010 import com.mklachl.sopkom.repository.KursRepository;
00011 import com.mklachl.sopkom.repository.LiniaRepository;
00012 import com.mklachl.sopkom.repository.PrzystanekRepository;
00013 import com.mklachl.sopkom.repository.PrzystanekWliniRepository;
00014 import com.mklachl.sopkom.services.PrzystanekService;
00015 import org.springframework.beans.factory.annotation.Autowired;
00016 import org.springframework.boot.CommandLineRunner;
00017 import org.springframework.stereotype.Service;
00018 import org.springframework.transaction.annotation.Transactional;
00019
00020 import java.io.File;
00021 import java.io.FileOutputStream;
00022 import java.io.IOException;
00023 import java.net.URISyntaxException;
00024 import java.sql.Time;
00025 import java.util.*;
00026 import java.util.List;
00027 import java.util.stream.Stream;
00028
00029 @Service
00030 //public class Pdf implements CommandLineRunner {
00031 public class Pdf{
00032
```

```

00033     static BaseFont BASE_FONT_PL;
00034     static File directoryBase;
00035     static File rozkladyDirectory;
00036     static {
00037
00038         try {
00039             BASE_FONT_PL = BaseFont.createFont(BaseFont.HELVETICA, BaseFont.CP1250,
00040             BaseFont.EMBEDDED);
00041         } catch (DocumentException e) {
00042             e.printStackTrace();
00043         } catch (IOException e) {
00044             e.printStackTrace();
00045         }
00046
00047         directoryBase = new File("raporty");
00048         if(!directoryBase.exists()) {
00049             directoryBase.mkdir();
00050         }
00051
00052         rozkladyDirectory = new File(directoryBase, "rozklady");
00053         if(!rozkladyDirectory.exists()) {
00054             rozkladyDirectory.mkdir();
00055         }
00056     }
00057 }
00058
00059     private void addTableHeader(PdfPTable table) {
00060         Stream.of("Linia", "Godzina")
00061             .forEach(columnTitle -> {
00062                 PdfPCell header = new PdfPCell();
00063                 header.setBackgroundColor(BaseColor.LIGHT_GRAY);
00064                 header.setBorderWidth(2);
00065                 header.setPhrase(new Phrase(columnTitle));
00066                 header.setHorizontalAlignment(Element.ALIGN_CENTER);
00067                 table.addCell(header);
00068             });
00069     }
00070
00071     private void addLinieHeader(PdfPTable table, Long id) {
00072         TreeSet<$> linieInfo = getAllByPrzystanekId(id);
00073         Set<String> uniqueLines = new HashSet<>();
00074
00075         for ($ liniaInfo : linieInfo) {
00076             String liniaNumer = liniaInfo.getLinia().getNumer();
00077             uniqueLines.add(liniaNumer);
00078         }
00079
00080         for (String liniaNumer : uniqueLines) {
00081             PdfPCell header = new PdfPCell();
00082             header.setBackgroundColor(BaseColor.LIGHT_GRAY);
00083             header.setBorderWidth(2);
00084             header.setPhrase(new Phrase(liniaNumer));
00085             header.setHorizontalAlignment(Element.ALIGN_CENTER);
00086             table.addCell(header);
00087         }
00088     }
00089 }
00090
00091     private void addLinieRows(PdfPTable table, Long id) throws URISyntaxException, IOException {
00092
00093         TreeSet<$> linieInfo = getAllByPrzystanekId(id);
00094         Set<String> uniqueLines = new HashSet<>();
00095
00096         for ($ liniaInfo : linieInfo) {
00097             String liniaNumer = liniaInfo.getLinia().getNumer();
00098             uniqueLines.add(liniaNumer);
00099         }
00100
00101         for (String liniaNumer : uniqueLines) {
00102             Linia l = liniaRepos.findById(liniaNumer);
00103
00104             List<PrzystanekWlini> p = przystanekWliniRepos.findAllByLinia(l);
00105             Phrase przystankiPhrase = new Phrase();
00106             for (PrzystanekWlini pp : p){
00107                 Chunk chunk = new Chunk(pp.getPrzystanek().getNazwa(), new Font(BASE_FONT_PL));
00108                 przystankiPhrase.add(chunk);
00109                 przystankiPhrase.add(chunk.NEWLINE);
00110             }
00111             PdfPCell przystankiCell = new PdfPCell(przystankiPhrase);
00112             przystankiCell.setHorizontalAlignment(Element.ALIGN_CENTER);
00113             przystankiCell.setPaddingTop(10);
00114             przystankiCell.setPaddingBottom(10);
00115             table.addCell(przystankiCell);
00116
00117         }
00118     }

```

```
00119
00120     }
00121     private void addCustomRows(PdfPTable table, Long id) throws URISyntaxException, IOException {
00122         var vat = getAllByPrzystanekId(id);
00123         Map<String, List<String>> linieGodzinyMap = new HashMap<>();
00124         Map<String, List<BaseColor>> linieKoloryMap = new HashMap<>();
00125         Map<String, List<Boolean>> linieOdwMap = new HashMap<>();
00126
00127         for (var $$$ : vat) {
00128             String liniaNumer = $$$.getLinia().getNumer();
00129             String godzinaStr = $$$.getT().toString();
00130             String formattedTime = godzinaStr.substring(0, 5);
00131             BaseColor textColor = getTextColor($$$.getHarmonogram().getNazwa());
00132             boolean isOdw = $$$.isOdw();
00133
00134             linieGodzinyMap.putIfAbsent(liniaNumer, new ArrayList<>());
00135             linieGodzinyMap.get(liniaNumer).add(formattedTime);
00136
00137             linieKoloryMap.putIfAbsent(liniaNumer, new ArrayList<>());
00138             linieKoloryMap.get(liniaNumer).add(textColor);
00139
00140             linieOdwMap.putIfAbsent(liniaNumer, new ArrayList<>());
00141             linieOdwMap.get(liniaNumer).add(isOdw);
00142         }
00143
00144         for (Map.Entry<String, List<String>> entry : linieGodzinyMap.entrySet()) {
00145             String liniaNumer = entry.getKey();
00146             List<String> godzinyList = entry.getValue();
00147             List<BaseColor> koloryList = linieKoloryMap.get(liniaNumer);
00148             List<Boolean> odwList = linieOdwMap.get(liniaNumer); // Lista z informacją o znaku
00149             podkreślenia
00150             PdfPCell liniaCell = new PdfPCell(new Phrase(liniaNumer));
00151             liniaCell.setHorizontalAlignment(Element.ALIGN_CENTER);
00152             liniaCell.setPaddingTop(10);
00153             liniaCell.setPaddingBottom(10);
00154             table.addCell(liniaCell);
00155
00156             Phrase godzinyPhrase = new Phrase();
00157             int i = 0;
00158             for (var godzina : godzinyList) {
00159                 BaseColor textColor = koloryList.get(i);
00160                 boolean isOdw = odwList.get(i);
00161
00162                 Font font = new Font(BASE_FONT_PL, 12, Font.NORMAL, textColor);
00163
00164                 if (isOdw) {
00165                     font.setStyle(Font.UNDERLINE);
00166                 }
00167
00168                 Chunk godzinaChunk = new Chunk(godzina, font);
00169                 godzinyPhrase.add(godzinaChunk);
00170                 godzinyPhrase.add(Chunk.SPACETABBING);
00171                 i++;
00172             }
00173
00174             PdfPCell godzinaCell = new PdfPCell(godzinyPhrase);
00175             godzinaCell.setHorizontalAlignment(Element.ALIGN_CENTER);
00176             godzinaCell.setPaddingTop(10);
00177             godzinaCell.setPaddingBottom(10);
00178             table.addCell(godzinaCell);
00179         }
00180     }
00181     private BaseColor getTextColor(String harmonogram) {
00182         if (harmonogram.equals("Dni robocze")) {
00183             return BaseColor.BLACK;
00184         } else if (harmonogram.equals("Dni robocze i weekendy")) {
00185             return BaseColor.BLUE;
00186         } else if (harmonogram.equals("weekendy")) {
00187             return BaseColor.RED;
00188         } else {
00189             return BaseColor.BLACK;
00190         }
00191     }
00192
00193
00194     @Autowired
00195     public KursRepository kursRepository;
00196
00197     @Autowired
00198     public PrzystanekRepository przystanekRepos;
00199
00200     @Autowired
00201     public PrzystanekWliniRepository przystanekWliniRepos;
00202
00203     @Autowired
00204     public LiniaRepository liniaRepos;
```

```

00205
00206
00207     public TreeSet<$> getAllByPrzystanekId(Long id) {
00208         TreeSet<$> data = new TreeSet<$>();
00209         List<Kurs> kursy = kursRepository.findAll();
00210
00211         for (Kurs kurs : kursy) {
00212             for (var kursprzystanek : kurs.getKursPrzystanekWlinii()) {
00213                 var przystanek = kursprzystanek.getPrzystanekWlini().getPrzystanek();
00214                 if (przystanek.getPrzystanekId().equals(id)) {
00215
00216                     data.add(new $ (kursprzystanek.getGodzinna(), new LiniaDto(kurs.getLinia())),
00217                     kurs.getKierunek().equals(Short.valueOf((short) 1)), kurs.getHarmonogram()));
00218                     continue;
00219                 }
00220             }
00221
00222         return data;
00223     }
00224
00225     public int getLiczbaLiniiNaPrzystanku(Long id) {
00226         TreeSet<$> linieInfo = getAllByPrzystanekId(id);
00227         Set<String> uniqueLines = new HashSet<$>();
00228
00229         for ($ liniaInfo : linieInfo) {
00230             String liniaNumer = liniaInfo.getLinia().getNumer();
00231             uniqueLines.add(liniaNumer);
00232         }
00233
00234         return uniqueLines.size();
00235     }
00236
00237
00238     public File genRozklad(Long id) throws Exception{
00239         String przystanekName = "nazwa";
00240         File result = new File(rozkladyDirectory, "rozklad_jazdy_"+przystanekName+".pdf");
00241
00242         //Long id = (long)1;
00243
00244         var przystanki = przystanekRepos.findAll();
00245         for(Przystanek p : przystanki) {
00246             if(p.getPrzystanekId().equals(id)) {
00247                 przystanekName = p.getNazwa();
00248             }
00249         }
00250
00251
00252         Document document = new Document();
00253         PdfWriter.getInstance(document, new FileOutputStream(result));
00254
00255         document.open();
00256
00257         Font font = new Font(BASE_FONT_PL, 18, Font.BOLD);
00258         Paragraph header = new Paragraph("Rozkład jazdy", font);
00259         header.setAlignment(Element.ALIGN_CENTER);
00260         document.add(header);
00261         document.add(new Paragraph(" "));
00262
00263         Paragraph przystanek = new Paragraph("Przystanek: " + przystanekName, font);
00264         przystanek.setAlignment(Element.ALIGN_CENTER);
00265         document.add(przystanek);
00266         document.add(new Paragraph(" "));
00267
00268         PdfPTable table = new PdfPTable(2);
00269         table.setWidths(new float[]{1, 3});
00270         addTableHeader(table);
00271         addCustomRows(table, id);
00272         document.add(table);
00273
00274         document.add(new Paragraph(" "));
00275
00276
00277         Paragraph legend = new Paragraph();
00278         legend.add(new Paragraph("Legenda: ", new Font(BASE_FONT_PL, 12, Font.BOLD)));
00279         legend.add(new Paragraph("Czarny - Dni robocze ", new Font(BASE_FONT_PL, 12, Font.NORMAL,
00280         BaseColor.BLACK)));
00280         legend.add(new Paragraph("Czerwony - Weekendy", new Font(BASE_FONT_PL, 12, Font.NORMAL,
00281         BaseColor.RED)));
00281         legend.add(new Paragraph("Niebieski - Dni robocze i weekendy", new Font(BASE_FONT_PL, 12,
00282         Font.NORMAL, BaseColor.BLUE)));
00282         legend.add(new Paragraph("Podkreślane - kierunek odwrotny", new Font(BASE_FONT_PL, 12,
00283         Font.UNDERLINE, BaseColor.BLACK)));
00284
00285         document.add(legend);
00286
00287         document.add(new Paragraph(" "));

```

```
00287     document.add(new Paragraph(" "));  
00288  
00289     int liczbaLiniNaPrzystanku= getLiczbaLiniiNaPrzystanku(id);  
00290     PdfPTable tableLinie = new PdfPTable(liczbaLiniNaPrzystanku);  
00291     addLinieHeader(tableLinie, id);  
00292     addLinieRows(tableLinie, id);  
00293     document.add(tableLinie);  
00294  
00295     document.close();  
00296  
00297  
00298     return result;  
00299 }  
00300  
00301 //{@Override  
00302 @Transactional  
00303 public void genruj(Long id) throws Exception {  
00304 //public void run(String... args) throws Exception {  
00305     String przystanekName = "nazwa";  
00306  
00307 //Long id = (long)1;  
00308  
00309 var przystanki = przystanekRepos.findAll();  
00310 for(Przystanek p : przystanki) {  
00311     if(p.getPrzystanekId().equals(id)) {  
00312         przystanekName = p.getNazwa();  
00313     }  
00314 }  
00315  
00316 Document document = new Document();  
00317 PdfWriter.getInstance(document, new FileOutputStream("rozklad_jazdy_"+przystanekName+".pdf"));  
00318  
00319 document.open();  
00320  
00321  
00322 Font font = new Font(Font.FontFamily.HELVETICA, 18, Font.BOLD);  
00323 Paragraph header = new Paragraph("Rozkład jazdy", font);  
00324 header.setAlignment(Element.ALIGN_CENTER);  
00325 document.add(header);  
00326 document.add(new Paragraph(" "));  
00327  
00328 Paragraph przystanek = new Paragraph("Przystanek: " + przystanekName, font);  
00329 przystanek.setAlignment(Element.ALIGN_CENTER);  
00330 document.add(przystanek);  
00331 document.add(new Paragraph(" "));  
00332  
00333 PdfPTable table = new PdfPTable(2);  
00334 table.setWidths(new float[]{1, 3});  
00335 addTableHeader(table);  
00336 addCustomRows(table, id);  
00337 document.add(table);  
00338  
00339 document.add(new Paragraph(" "));  
00340  
00341 Paragraph legend = new Paragraph();  
00342 legend.add(new Paragraph("Legenda: ", new Font(Font.FontFamily.HELVETICA, 12, Font.BOLD)));  
00343 legend.add(new Paragraph("Czarny - Dni robocze ", new Font(Font.FontFamily.HELVETICA, 12,  
Font.NORMAL, BaseColor.BLACK)));  
00344 legend.add(new Paragraph("Czerwony - Weekendy", new Font(Font.FontFamily.HELVETICA, 12,  
Font.NORMAL, BaseColor.RED)));  
00345 legend.add(new Paragraph("Niebieski - Dni robocze i weekendy", new  
Font(Font.FontFamily.HELVETICA, 12, Font.NORMAL, BaseColor.BLUE)));  
00346 document.add(legend);  
00347  
00348 document.add(new Paragraph(" "));  
00349 document.add(new Paragraph(" "));  
00350  
00351 int liczbaLiniNaPrzystanku= getLiczbaLiniiNaPrzystanku(id);  
00352 PdfPTable tableLinie = new PdfPTable(liczbaLiniNaPrzystanku);  
00353 addLinieHeader(tableLinie, id);  
00354 addLinieRows(tableLinie, id);  
00355 document.add(tableLinie);  
00356  
00357 document.close();  
00358 }  
00359  
00360  
00361 public static class $ implements Comparable<$>{  
00362     private Time t;  
00363     private LiniaDto linia;  
00364     private boolean odw;  
00365     private Harmonogram harmonogram;  
00366  
00367  
00368     public $(Time t, LiniaDto linia, boolean odw, Harmonogram harm) {  
00369         super();  
00370         this.setT(t);  
00371     }  
00372 }
```

```

00371     this.setLinia(linia);
00372     this.setOdw(odw);
00373     this.setHarmonogram(harm);
00374 }
00376
00377 public $(Date date, LiniaDto linia, boolean odw, Harmonogram harm) {
00378     this(new Time(date.getTime()), linia, odw, harm);
00379 }
00381
00382 @Override
00383 public int compareTo($ o) {
00384     return getT().compareTo(o.getT());
00385 }
00387
00388 public LiniaDto getLinia() {
00389     return linia;
00390 }
00391
00392 public void setLinia(LiniaDto linia) {
00393     this.linia = linia;
00394 }
00395
00396 public Time getT() {
00397     return t;
00398 }
00399
00400 public void setT(Time t) {
00401     this.t = t;
00402 }
00403
00404 public boolean isOdw() {
00405     return odw;
00406 }
00407
00408 public void setOdw(boolean odw) {
00409     this.odw = odw;
00410 }
00411
00412 public void setHarmonogram(Harmonogram h) {
00413     this.harmonogram = h;
00414 }
00415
00416 public Harmonogram getHarmonogram() {return harmonogram;}
00417
00418 }
00419 }
```

7.211 Dokumentacja pliku

X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-2-comments/SOPKoM_Backend/src/test/java/com/mklachl/sopkom/SOPKoMApplicationTests.java

Komponenty

- class [com.mklachl.sopkom.SOPKoMApplicationTests](#)

Pakiety

- package [com.mklachl.sopkom](#)

7.212 SOPKoMApplicationTests.java

[Idź do dokumentacji tego pliku.](#)

```
00001 package com.mklachl.sopkom;
00002
00003 import org.junit.jupiter.api.Test;
00004 import org.springframework.boot.test.context.SpringBootTest;
00005
00006 @SpringBootTest
00007 class SOPKoMApplicationTests {
00008
00009     @Test
00010     void contextLoads() {
00011     }
00012
00013 }
```


Skorowidz

AbstractKursDto	securityFilterChain, 400
com.mklachl.sopkom.model.dto.kurs.AbstractKursDto	com.mklachl.sopkom.controller, 18
24	com.mklachl.sopkom.controller.AutobusController, 35
AbstractPrzystanekWKursieDto	addAutobus, 35
com.mklachl.sopkom.model.dto.kurs.AbstractPrzystanekWKursieDto	com.mklachl.sopkom.controller, 35
29	getAllBuses, 35
addAutobus	getAllBusesFilteredStatus, 36
com.mklachl.sopkom.controller.AutobusController,	getAutobus, 36
35	template, 37
addKurs	test, 37
com.mklachl.sopkom.controller.KursController, 188	updateAutobus, 37
com.mklachl.sopkom.model.dto.rozklad.RozkladDto, com.mklachl.sopkom.controller.Controller,	com.mklachl.sopkom.controller.Controller, 70
396	getString, 71
addLinia	getString2, 71
com.mklachl.sopkom.controller.LiniaController,	com.mklachl.sopkom.controller.HarmonogramController,
221	81
addPrzystanek	createHarmonogram, 81
com.mklachl.sopkom.controller.PrzystanekController,	deleteHarmonogram, 81
343	getAllHarmonograms, 83
ADMIN	getHarmonogramById, 83
com.mklachl.sopkom.model.entity.Role, 392	com.mklachl.sopkom.controller.IncydentController, 113
ADMIN_LOGIN	deleteIncydent, 114
com.mklachl.sopkom.security.CommandLineRunnerTest,	getIncydent, 114
69	getIncydents, 114
ADMIN_NAME	getIncydentsAfterDate, 115
com.mklachl.sopkom.security.CommandLineRunnerTest,	getIncydentsBeforeDate, 115
69	getIncydentsBetweenDates, 116
ADMIN_PASS	getIncydentsByAutobus, 116
com.mklachl.sopkom.security.CommandLineRunnerTest,	getIncydentsByKierowca, 116
69	getIncydentsByPrzejazd, 117
authenticationManager	incydentRepository, 118
com.mklachl.sopkom.config.SecurityConfig, 400	incydentService, 118
AutobusDto	saveIncydent, 117
com.mklachl.sopkom.model.dto.AutobusDto,	updateIncydent, 118
38,	com.mklachl.sopkom.controller.KierowcaController, 158
autobusRepository	deleteKierowca, 159
com.mklachl.sopkom.controller.PrzejazdController,	findAll, 160
272	findKierowcaById, 160
AutobusServiceImpl	findKierowcaByImieAndNazwisko, 161
com.mklachl.sopkom.services.impl.AutobusServiceImpl,	findKierowcaByNazwisko, 161
55	findKierowcaByPesel, 161
BiletDto	findKierowcaByPrawoJazdyWazneDoAfter, 162
com.mklachl.sopkom.model.dto.BiletDto, 64	findKierowcaByPrawoJazdyWazneDoBefore, 162
com.mklachl.sopkom, 17	findKierowcaByUser, 163
com.mklachl.sopkom.config, 17	KierowcaController, 159
com.mklachl.sopkom.config.SecurityConfig, 399	saveKierowca, 163
authenticationManager, 400	template, 163
passwordEncoder, 400	updateKierowca, 164
	com.mklachl.sopkom.controller.KursController, 187

addKurs, 188
 deleteKurs, 188
 getAllKursy, 189
 getKurs, 189
 getTemplateKurs, 189
 getTemplateKursInput, 190
 kursRepo, 191
 kursService, 191
 kurzy, 191
 liniaRepository, 191
 liniaService, 191
 przystnekWliniRepository, 192
 updateKurs, 190
com.mklachl.sopkom.controller.LiniaController, 221
 addLinia, 221
 deleteLinia, 223
 getAllPrzystanki, 223
 getAllPrzystankiFull, 223
 getLinia, 224
 getPrzystanekFull, 224
 template, 224
 templateInput, 225
 updateLinia, 225
com.mklachl.sopkom.controller.LoginController, 242
 getLoginPage, 242
 getLogoutPage, 243
com.mklachl.sopkom.controller.PrzejazdController, 268
 autobusRepository, 272
 deletePrzejazd, 269
 getAllAvaliableBusesForKurs, 269
 getAllAvaliableDatesForKurs, 269
 getAllAvaliableDriversForKurs, 270
 getPrzejazd, 270
 getPrzejazdy, 271
 getPrzyszlePrzejazdy, 271
 harmonogramRepository, 272
 kierowcaRepository, 272
 kursRepository, 272
 kursService, 272
 przejazdRepository, 273
 przejazdService, 273
 savePrzejazd, 271
com.mklachl.sopkom.controller.PrzystanekController, 342
 addPrzystanek, 343
 deletePrzystanek, 343
 getAllPrzystanki, 343
 getAllPrzystankiFull, 344
 getPrzystanek, 344
 getPrzystanekFull, 344
 template, 345
 updatePrzystanek, 345
com.mklachl.sopkom.controller.RaportyController, 391
 generateRozklad, 391
com.mklachl.sopkom.controller.RozkladController, 393
 repo, 394
 rozkladHelper, 394
 test, 393
com.mklachl.sopkom.controller.SimpleMessage, 400
 getMessage, 401
 message, 402
 setMessage, 402
 SimpleMessage, 401
com.mklachl.sopkom.controller.UserController, 408
 getLoggedInUser, 408
com.mklachl.sopkom.exceptions, 18
com.mklachl.sopkom.exceptions.IncydentException, 128
 IncydentException, 129
com.mklachl.sopkom.exceptions.LiniaNotFoundException, 233
 LiniaNotFoundException, 234
com.mklachl.sopkom.exceptions.PrzystanekNotFoundException, 366
 PrzystanekNotFoundException, 366
com.mklachl.sopkom.filter, 18
com.mklachl.sopkom.filter.LoggingFilter, 241
 doFilterInternal, 242
com.mklachl.sopkom.helpers, 18
com.mklachl.sopkom.helpers.Helpers, 106
 getUserFromContext, 106
 inject, 107
com.mklachl.sopkom.model.dto, 18
com.mklachl.sopkom.model.dto.AutobusDto, 38
 AutobusDto, 38, 39
 getAutbousId, 40
 getNumerRejestracyjny, 40
 getPrzebieg, 40
 getPrzegladWaznyDo, 40
 getStatus, 41
 setAutbousId, 41
 setNumerRejestracyjny, 41
 setPrzebieg, 42
 setPrzegladWaznyDo, 42
 setStatus, 42
 toString, 43
com.mklachl.sopkom.model.dto.BiletDto, 64
 BiletDto, 64
 getBiletid, 65
 getCena, 65
 getNazwa, 65
 setBiletid, 65
 setCena, 66
 setNazwa, 66
com.mklachl.sopkom.model.dto.DateWrapperDto, 72
 DateWrapperDto, 73
 getDate, 73
 setDate, 73
com.mklachl.sopkom.model.dto.harmonogram, 19
com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto, 84
 getDodatkoweInf, 85
 getHarmonogramId, 85
 getNazwa, 85
 HarmonogramDto, 84
 isCzw, 85

isNie, 86
isPia, 86
isPon, 86
isSob, 86
isSro, 87
isWto, 87
setCzw, 87
setDodatkowelnf, 88
setHarmonogramId, 88
setNazwa, 88
setNie, 89
setPia, 89
setPon, 89
setSob, 90
setSro, 90
setWto, 90
com.mklachl.sopkom.model.dto.IncydentDto, 119
getAutobusId, 120
getDate, 120
getDodatkowelnformacje, 120
getIncydentId, 121
getKierowcald, 121
getKoszty, 121
getKrotko, 121
getPrzejazdId, 122
getTyp, 122
IncydentDto, 119
setAutobusId, 122
setDate, 123
setDodatkowelnformacje, 123
setIncydentId, 123
setKierowcald, 124
setKoszty, 124
setKrotko, 124
setPrzejazdId, 125
setTyp, 125
com.mklachl.sopkom.model.dto.IncydentDtoOutput, 125
getAutobus, 126
getKierowca, 127
getKurs, 127
IncydentDtoOutput, 126
setAutobus, 127
setKierowca, 128
setKurs, 128
com.mklachl.sopkom.model.dto.KierowcaDto, 164
getImie, 165
getKierowcald, 166
getNazwisko, 166
getPesel, 166
getPrawoJazdyWazneDo, 166
getUser, 167
KierowcaDto, 165
setImie, 167
setKierowcald, 167
setNazwisko, 168
setPesel, 168
setPrawoJazdyWazneDo, 168
setUserDto, 169
toString, 169
com.mklachl.sopkom.model.dto.kurs, 19
com.mklachl.sopkom.model.dto.kurs.AbstractKursDto,
23
AbstractKursDto, 24
getHarmonogramId, 24
getKierunek, 24
getKursId, 25
getLiniald, 25
getTyp_autobusu, 25
harmonogramId, 27
kierunek, 27
kursId, 27
liniald, 28
setHarmonogramId, 25
setKierunek, 26
setKursId, 26
setLiniald, 26
setTyp_autobusu, 27
typ_autobusu, 28
com.mklachl.sopkom.model.dto.kurs.AbstractPrzystanekWKursieDto,
28
AbstractPrzystanekWKursieDto, 29
getGodzinna, 29
getPrzystanekwKursield, 30
godzinna, 31
przystanekwKursield, 31
setGodzinna, 30
setPrzystanekwKursield, 30
com.mklachl.sopkom.model.dto.kurs.InputKursDto, 150
getPrzystanki, 151
InputKursDto, 151
przystanki, 152
setPrzystanki, 151
com.mklachl.sopkom.model.dto.kurs.InputPrzystanekWKursieDto,
152
getPrzystanekWLini, 153
InputPrzystanekWKursieDto, 153
setPrzystanekWLini, 153
com.mklachl.sopkom.model.dto.kurs.KursDto, 192
getHarmonogram, 193
getLinia, 193
getOdwracalny, 194
getPrzystanki, 194
KursDto, 193
setHarmonogram, 194
setLinia, 195
setOdwracalny, 195
setPrzystanki, 195
com.mklachl.sopkom.model.dto.kurs.UPDATED, 19
com.mklachl.sopkom.model.dto.kurs.UPDATED.KursDtoUPDATED,
196
getHarmonogram, 197
getKieruenk, 197
getKursId, 197
getLinia, 197
getPrzystanki, 197
harmonogram, 199

isOdwracalny, 198
 Kieruenk, 199
 KursDtoOUTDATED, 196, 197
 kursId, 199
 linia, 199
 odwracalny, 200
 przystanki, 200
 setHarmonogram, 198
 setKieruenk, 198
 setKursId, 198
 setLinia, 198
 setOdwracalny, 198
 setPrzystanki, 199
 com.mklachl.sopkom.model.dto.kurs.OUTDATED.PrzystenekWKursieDtoOUTDATED,
 388
 getGodzinna, 389
 getPrzystanek, 389
 getPrzystanekkurs_id, 389
 godzinna, 390
 kolejnosc, 390
 przystanek, 390
 przystanekkurs_id, 390
 PrzystenekWKursieDtoOUTDATED, 388
 setGodzinna, 389
 setPrzystanek, 389
 setPrzystanekkurs_id, 390
 com.mklachl.sopkom.model.dto.kurs.PrzystanekWKursieDto,
 381
 getPrzystanekWLini, 382
 PrzystanekWKursieDto, 381
 setPrzystanekWLini, 382
 com.mklachl.sopkom.model.dto.linia, 20
 com.mklachl.sopkom.model.dto.linia.LiniaDto, 226
 getId, 227
 getNumer, 227
 id, 228
 LiniaDto, 226
 numer, 228
 setId, 227
 setNumer, 228
 com.mklachl.sopkom.model.dto.linia.LiniaDtoInput, 229
 getPrzystanki, 230
 LiniaDtoInput, 229
 setPrzystanki, 230
 com.mklachl.sopkom.model.dto.linia.LiniaDtoOutput,
 230
 getPrzystanki, 232
 isOdwracalna, 232
 LiniaDtoOutput, 231
 odwracalna, 233
 setOdwracalna, 232
 setPrzystanki, 233
 com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLinia,
 352
 getAdres, 354
 getKolejnosc, 354
 getPrzystanekwlinild, 354
 PrzystanekDtoDlaLinia, 353
 setAdres, 354
 setKolejnosc, 355
 setPrzystanekwlinild, 355
 com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLiniaInput,
 355
 getKolejnosc, 356
 getPrzystanekId, 357
 PrzystanekDtoDlaLiniaInput, 356
 setKolejnosc, 357
 setPrzystanekId, 357
 com.mklachl.sopkom.model.dto.Message, 243
 Message, 244
 message, 245
 SetMessageInt, 244
 SetMessageInt, 245
 com.mklachl.sopkom.model.dto.przejazd, 20
 com.mklachl.sopkom.model.dto.przejazd.PKPWLDtoUpdate,
 248
 getDate, 248
 getId, 249
 PKPWLDtoUpdate, 248
 setDate, 249
 setId, 249
 com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoInput,
 273
 getAutobusId, 274
 getDataPrzejazdu, 274
 getKierowcaId, 274
 getKursId, 275
 PrzejazdDtoInput, 274
 setAutobusId, 275
 setDataPrzejazdu, 275
 setKierowcaId, 276
 setKursId, 276
 toString, 276
 com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput,
 277
 getAutobusNazwa, 278
 getAutobusNumerRejestracyjny, 279
 getCenaZaLitru, 279
 getData, 279
 getDataKonca, 279
 getDataStartu, 280
 getDlugoscTrasy, 280
 getKierowcaImieNazwisko, 280
 getKierunek, 280
 getKursId, 281
 getLiczbaBiletow, 281
 getLiniaNumer, 281
 getPrzejazdId, 281
 getPrzystanki, 282
 getSpalanie, 282
 getStatus, 282
 PrzejazdDtoOutput, 277, 278
 setAutobusNazwa, 282
 setAutobusNumerRejestracyjny, 283
 setCenaZaLitru, 283
 setData, 283

setDataKonca, 284
setDataStartu, 284
setDlugoscTrasy, 284
setKierowcAcalmieNazwisko, 284
setKierunek, 285
setKursId, 285
setLiczbaBiletow, 285
setLiniaNumer, 286
setPrzejazdId, 286
setPrzystanki, 286
setSpalanie, 287
setStatus, 287
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate, 287
 getCenaZaLitr, 288
 getDlugoszTrasy, 288
 getLiczbaBiletowNormalnych, 289
 getLiczbaBiletowUlgowych, 289
 getPrzejazdId, 289
 getPrzystanki, 289
 getSpalanie, 290
 PrzejazdDtoUpdate, 288
 setCenaZaLitr, 290
 setDlugoszTrasy, 290
 setLiczbaBiletowNormalnych, 291
 setLiczbaBiletowUlgowych, 291
 setPrzejazdId, 291
 setPrzystanki, 292
 setSpalanie, 292
 toString, 292
com.mklachl.sopkom.model.dto.PrzejazdBiletDto, 261
 getBileId, 262
 getCenaBiletu, 262
 getPrzejazdBileId, 262
 getPrzejazdId, 262
 PrzejazdBiletDto, 261
 setBileId, 263
 setCenaBiletu, 263
 setPrzejazdBileId, 263
 setPrzejazdId, 265
com.mklachl.sopkom.model.dto.PrzejazdKursPrzystanekWliniDto, 295
 getKolejnosc, 297
 getNazwa, 297
 getPrzejazdKursPrzystanekWliniId, 297
 getPrzewidywanaGodzinna, 297
 getPrzystanekId, 298
 getRealnaGodzinna, 298
 PrzejazdKursPrzystanekWliniDto, 296
 setKolejnosc, 298
 setNazwa, 299
 setPrzejazdKursPrzystanekWliniId, 299
 setPrzewidywanaGodzinna, 299
 setPrzystanekId, 300
 setRealnaGodzinna, 300
com.mklachl.sopkom.model.dto.PrzystanekDto, 346
 getDlugoscGeograficzna, 347
 getKodPocztowy, 347
 getMiasto, 347
 getNazwa, 348
 getPrzystanekId, 348
 getPrzystanekOdwojny, 348
 getPrzystanekOdwojnyNazwa, 348
 getSzerokoscGeograficzna, 349
 getUlica, 349
 PrzystanekDto, 346, 347
 setDlugoscGeograficzna, 349
 setKodPocztowy, 350
 setMiasto, 350
 setNazwa, 350
 setPrzystanekId, 351
 setPrzystanekOdwojny, 351
 setPrzystanekOdwojnyNazwa, 351
 setSzerokoscGeograficzna, 351
 setUlica, 352
com.mklachl.sopkom.model.dto.PrzystanekDtoFull, 358
 dlugoscGeograficzna, 364
 getDlugoscGeograficzna, 359
 getKodPocztowy, 359
 getMiasto, 360
 getNazwa, 360
 getPrzystanekId, 360
 getPrzystanekOdwojny, 360
 getSzerokoscGeograficzna, 361
 getUlica, 361
 kodPocztowy, 364
 miasto, 364
 nazwa, 364
 PrzystanekDtoFull, 359
 przystanekId, 365
 przystanekOdwojny, 365
 setDlugoscGeograficzna, 361
 setKodPocztowy, 362
 setMiasto, 362
 setNazwa, 362
 setPrzystanekId, 363
 setPrzystanekOdwojny, 363
 setSzerokoscGeograficzna, 363
 setUlica, 363
 szerokoscGeograficzna, 365
 ulica, 365
com.mklachl.sopkom.model.dto.rozklad, 20
com.mklachl.sopkom.model.dto.rozklad.RozkladDto, 394
 addKurs, 396
 getDate, 396
 getNormalny, 396
 getRozklad, 396
 RozkladDto, 395
 setDate, 397
 setNormalny, 397
 setRozklad, 397
com.mklachl.sopkom.model.dto.rozklad.RozkladKursDto, 399
com.mklachl.sopkom.model.dto.rozklad.RozkladLiniaDto, 399

com.mklachl.sopkom.model.dto.UserDto, 409
 getEmail, 410
 getId, 410
 getName, 410
 getPassword, 410
 getRole, 411
 getRoles, 411
 setEmail, 411
 setId, 412
 setName, 412
 setPassword, 412
 setRole, 413
 setRoles, 413
 UserDto, 409

com.mklachl.sopkom.model.entity, 20

com.mklachl.sopkom.model.entity.Autobus, 31
 getAutbousId, 32
 getNumerRejestracyjny, 32
 getPrzebieg, 32
 getPrzegladWaznyDo, 32
 getStatus, 33
 setAutbousId, 33
 setNumerRejestracyjny, 33
 setPrzebieg, 34
 setPrzegladWaznyDo, 34
 setStatus, 34

com.mklachl.sopkom.model.entity.Bilet, 60
 getBileId, 61
 getCena, 61
 getNazwa, 61
 getPrzejazdBilet, 61
 setBileId, 62
 setCena, 63
 setNazwa, 63
 setPrzejazdBilet, 63

com.mklachl.sopkom.model.entity.Harmonogram, 74
 getDodatkowelnf, 75
 getHamonogramId, 75
 getNazwa, 75
 isCzw, 75
 isNie, 76
 isPia, 76
 isPon, 76
 isSob, 76
 isSro, 77
 isWto, 77
 setCzw, 77
 setDodatkowelnf, 78
 setHamonogramId, 78
 setNazwa, 78
 setNie, 79
 setPia, 79
 setPon, 79
 setSob, 80
 setSro, 80
 setWto, 80

com.mklachl.sopkom.model.entity.Incydent, 107
 getAutobus, 108

 getDate, 108
 getDodatkowelnformacje, 108
 getIncydentId, 108
 getKierowca, 109
 getKoszty, 109
 getKrotko, 109
 getPrzejazd, 109
 getTyp, 110
 setAutobus, 110
 setDate, 110
 setDodatkowelnformacje, 111
 setIncydentId, 111
 setKierowca, 111
 setKoszty, 112
 setKrotko, 112
 setPrzejazd, 112
 setTyp, 113

com.mklachl.sopkom.model.entity.Kierowca, 154
 equals, 154
 getImie, 155
 getKierowcald, 155
 getNazwisko, 155
 getPesel, 155
 getPrawoJazdyWazneDo, 156
 getUser, 156
 hashCode, 156
 getImie, 156
 setKierowcald, 157
 setNazwisko, 157
 setPesel, 157
 setPrawoJazdyWazneDo, 158
 setUser, 158

com.mklachl.sopkom.model.entity.Kurs, 183
 getHarmonogram, 183
 getKierunek, 184
 getKursId, 184
 getKursPrzystanekWlinii, 184
 getLinia, 184
 getTypAutobusu, 185
 setHarmonogram, 185
 setKierunek, 185
 setKursId, 186
 setKursPrzystanekWlinii, 186
 setLinia, 186
 setTypAutobusu, 187

com.mklachl.sopkom.model.entity.KursPrzystanekWlini, 200
 getGodzinna, 201
 getKurs, 201
 getKursPrzystanekWlinild, 201
 getPrzystanekWlini, 201
 setGodzinna, 202
 setKurs, 203
 setKursPrzystanekWlinild, 203
 setPrzystanekWlini, 203

com.mklachl.sopkom.model.entity.Linia, 218
 getLiniald, 219
 getNumer, 219

getPrzystankiWlini, 219
odwracalna, 220
setLiniald, 220
setNumer, 220
setPrzystankiWlini, 221
com.mklachl.sopkom.model.entity.Przejazd, 250
 getAutobus, 250
 getCenaZaLitr, 251
 getData, 251
 getDataKonca, 251
 getDataStartu, 251
 getDlugoscTrasy, 252
 getKierowca, 252
 getKurs, 252
 getPrzejazdBilet, 252
 getPrzejazdId, 253
 getPrzejazdKursPrzystanekWlini, 253
 getSpalanie, 253
 setAutobus, 253
 setCenaZaLitr, 254
 setData, 254
 setDataKonca, 254
 setDataStartu, 255
 setDlugoscTrasy, 255
 setKierowca, 255
 setKurs, 255
 setPrzejazdBilet, 256
 setPrzejazdId, 256
 setPrzejazdKursPrzystanekWlini, 256
 setSpalanie, 257
com.mklachl.sopkom.model.entity.PrzejazdBilet, 257
 getBilet, 258
 getCenaBiletu, 258
 getPrzejazd, 258
 getPrzejazdBiletid, 258
 setBilet, 259
 setCenaBiletu, 260
 setPrzejazd, 260
 setPrzejazdBiletid, 260
com.mklachl.sopkom.model.entity.PrzejazdKursPrzystanekWlini, 293
 getKursPrzystanekWlini, 293
 getPrzejazd, 293
 getPrzejazdKursPrzystanekWlinid, 293
 getRealnaGodzinna, 294
 setKursPrzystanekWlini, 294
 setPrzejazd, 294
 setPrzejazdKursPrzystanekWlinid, 295
 setRealnaGodzinna, 295
com.mklachl.sopkom.model.entity.Przystanek, 337
 getDlugoscGeograficzna, 337
 getKodPocztowy, 337
 getMiasto, 338
 getNazwa, 338
 getPrzystanekId, 338
 getPrzystanekOdwrotny, 338
 getPrzystankiWlini, 339
 getSzerokoscGeograficzna, 339
 getUlica, 339
 setDlugoscGeograficzna, 339
 setKodPocztowy, 340
 setMiasto, 340
 setNazwa, 340
 setPrzystanekId, 341
 setPrzystanekOdwrotny, 341
 setPrzystankiWlini, 341
 setSzerokoscGeograficzna, 341
 setUlica, 342
com.mklachl.sopkom.model.entity.PrzystanekWlini, 382
 getKolejnosc, 383
 getKursPrzystanekWlinii, 383
 getLinia, 383
 getPrzystanek, 384
 getPrzystanekWlinid, 384
 setKolejnosc, 384
 setKursPrzystanekWlinii, 385
 setLinia, 385
 setPrzystanek, 385
 setPrzystanekWlinid, 386
com.mklachl.sopkom.model.entity.Role, 392
 ADMIN, 392
 DISPATCHER, 392
 DRIVER, 392
 MANAGER, 393
com.mklachl.sopkom.model.entity.User, 403
 getId, 405
 getLogin, 405
 getName, 405
 getPassword, 405
 getRoles, 406
 setId, 406
 setLogin, 406
 setName, 407
 setPassword, 407
 setRoles, 407
 User, 403, 404
com.mklachl.sopkom.repository, 21
com.mklachl.sopkom.repository.AutobusRepository, 43
 findAll, 44
 findAllByAvailable, 44
 findAllByNumerRejestracyjnyLike, 44
 findAllByPrzebieg, 45
 findAllByPrzebiegGreaterThanOrEqual, 45
 findAllByPrzebiegLessThan, 45
 findAllByStatus, 47
 findByNumerRejestracyjny, 47
 findByPrzegladWaznyDoAfter, 47
 findByPrzegladWaznyDoBefore, 49
 findByStatusLike, 49
com.mklachl.sopkom.repository.BiletRepository, 66
 findAll, 67
 findAllByCena, 67
 findAllByNazwa, 68
com.mklachl.sopkom.repository.HarmonogramRepository, 91
 findAll, 91

findAllByCzw, 91
 findAllByDodatkoweInflLike, 92
 findAllByNazwa, 92
 findAllByNie, 92
 findAllByPia, 93
 findAllByPon, 93
 findAllBySob, 94
 findAllBySro, 94
 findAllByWto, 94
 com.mklachl.sopkom.repository.IncydentRepository,
 129
 findAll, 130
 findAllByAutobus, 130
 findAllByDate, 131
 findAllByDateAfter, 131
 findAllByDateBefore, 131
 findAllByDateBetween, 132
 findAllByDodatkoweInformacje, 132
 findAllByKierowca, 132
 findAllByKoszty, 133
 findAllByKosztyGreaterThan, 133
 findAllByKosztyLessThan, 134
 findAllByPrzejazd, 134
 findAllByTyp, 134
 com.mklachl.sopkom.repository.KierowcaRepository,
 169
 findAll, 170
 findAllByAvailability, 170
 findAllByImieAndNazwisko, 171
 findAllByImieAndNazwiskoLike, 171
 findAllByPeselLike, 171
 findAllByPrawoJazdyWazneDoAfter, 172
 findAllByPrawoJazdyWazneDoBefore, 172
 findByNazwiskoLike, 172
 findByPesel, 173
 findByUser, 173
 com.mklachl.sopkom.repository.KursPrzystanekWliniRepository,
 204
 findAll, 204
 findAllByGodzinna, 205
 findAllByGodzinnaAfter, 205
 findAllByGodzinnaBefore, 205
 findAllByKurs, 206
 findAllByPrzystanekWlini, 206
 com.mklachl.sopkom.repository.KursRepository, 207
 findAll, 207
 findAllByHarmonogram, 207
 findAllByKierunek, 208
 findAllByLinia, 208
 findAllByTypAutobusu, 208
 com.mklachl.sopkom.repository.LiniaRepository, 234
 findAll, 235
 findById, 235
 findByNumer, 235
 com.mklachl.sopkom.repository.PrzejazdBiletRepository,
 265
 findAll, 266
 findAllByBilet, 266
 findAllByCenaBiletu, 266
 findAllByCenaBiletuGreaterThanOrEqual, 267
 findAllByCenaBiletuLessThanOrEqual, 267
 findAllByPrzejazd, 267
 com.mklachl.sopkom.repository.PrzejazdKursPrzystanekWliniRepository,
 300
 findAll, 301
 findAllByKursPrzystanekWlini, 301
 findAllByPrzejazd, 302
 findAllByRealnaGodzinna, 302
 findAllByRealnaGodzinnaAfter, 302
 findAllByRealnaGodzinnaBefore, 303
 com.mklachl.sopkom.repository.PrzejazdRepository,
 303
 findAll, 304
 findAllByAutobus, 304
 findAllByCenaZaLitr, 305
 findAllByCenaZaLitrGreaterThanOrEqual, 305
 findAllByCenaZaLitrLessThanOrEqual, 305
 findAllByData, 306
 findAllByDataKonca, 307
 findAllByDataKoncaAfter, 307
 findAllByDataKoncaBefore, 307
 findAllByDataStartu, 308
 findAllByDataStartuAfter, 308
 findAllByDataStartuBefore, 308
 findAllByDlugoscTrasy, 310
 findAllByDlugoscTrasyGreaterThanOrEqual, 310
 findAllByDlugoscTrasyLessThanOrEqual, 310
 findAllByKierowca, 311
 findAllByKurs, 311
 findAllByKursAndData, 312
 findAllBySpalanie, 312
 findAllBySpalanieGreaterThanOrEqual, 312
 findAllBySpalanieLessThanOrEqual, 313
 com.mklachl.sopkom.repository.PrzystanekRepository,
 366
 findAll, 367
 findAllByDlugoscGeograficzna, 367
 findAllByKodPocztowy, 368
 findAllByMiasto, 368
 findAllByNazwaLike, 368
 findAllByPrzystanekOdwozny, 369
 findAllBySzerokoscGeograficzna, 369
 findAllByUlica, 369
 findAllByUlicaLike, 370
 findByNazwa, 370
 com.mklachl.sopkom.repository.PrzystanekWliniRepository,
 386
 findAll, 387
 findAllByLinia, 387
 findAllByPrzystanek, 387
 com.mklachl.sopkom.repository.UserRepository, 413
 findByLogin, 414
 com.mklachl.sopkom.security, 21
 com.mklachl.sopkom.security.CommandLineRunnerTest,
 68
 ADMIN_LOGIN, 69

ADMIN_NAME, 69
ADMIN_PASS, 69
MANAGER_LOGIN, 70
MANAGER_NAME, 70
MANAGER_PASS, 70
run, 69
userRepository, 70
com.mklachl.sopkom.security.CustomUserDetailsService,
 71
 CustomUserDetailsService, 72
 loadUserByUsername, 72
com.mklachl.sopkom.services, 21
com.mklachl.sopkom.services.AutobusService, 50
 findAutobusByAvailable, 50
 findAutobusById, 51
 findAutobusByNumerRejestracyjny, 51
 findAutobusByPrzebieg, 51
 findAutobusByPrzebiegGreaterThanOrEqual, 52
 findAutobusByPrzebiegLessThanOrEqual, 52
 findAutobusByPrzegladWaznyDoAfter, 52
 findAutobusByPrzegladWaznyDoBefore, 53
 findAutobusByStatus, 53
 saveAutobus, 54
com.mklachl.sopkom.services.HarmonogramService,
 95
 findHarmonogramByCzw, 95
 findHarmonogramByDodatkoweInfLike, 96
 findHarmonogramById, 96
 findHarmonogramByNazwa, 96
 findHarmonogramByNie, 97
 findHarmonogramByPia, 97
 findHarmonogramByPon, 98
 findHarmonogramBySob, 98
 findHarmonogramBySro, 98
 findHarmonogramByWto, 99
 saveHarmonogram, 99
com.mklachl.sopkom.services.impl, 22
com.mklachl.sopkom.services.impl.AutobusServiceImpl,
 55
 AutobusServiceImpl, 55
 findAutobusByAvailable, 55
 findAutobusById, 56
 findAutobusByNumerRejestracyjny, 56
 findAutobusByPrzebieg, 57
 findAutobusByPrzebiegGreaterThanOrEqual, 57
 findAutobusByPrzebiegLessThanOrEqual, 57
 findAutobusByPrzegladWaznyDoAfter, 58
 findAutobusByPrzegladWaznyDoBefore, 58
 findAutobusByStatus, 59
 saveAutobus, 59, 60
com.mklachl.sopkom.services.impl.HarmonogramServiceImpl,
 100
 findHarmonogramByCzw, 101
 findHarmonogramByDodatkoweInfLike, 101
 findHarmonogramById, 102
 findHarmonogramByNazwa, 102
 findHarmonogramByNie, 103
 findHarmonogramByPia, 103
findHarmonogramByPon, 103
findHarmonogramBySob, 104
findHarmonogramBySro, 104
findHarmonogramByWto, 105
HarmonogramServiceImpl, 101
saveHarmonogram, 105, 106
com.mklachl.sopkom.services.impl.IncydentServiceImpl,
 142
 findIncydentByAutobus, 143
 findIncydentByDate, 143
 findIncydentByDateAfter, 144
 findIncydentByDateBefore, 144
 findIncydentByDateBetween, 145
 findIncydentByDodatkoweInformacje, 145
 findIncydentById, 146
 findIncydentByKierowca, 146
 findIncydentByKoszty, 146
 findIncydentByKosztyGreaterThanOrEqual, 147
 findIncydentByKosztyLessThanOrEqual, 147
 findIncydentByPrzejazd, 148
 findIncydentByTyp, 148
 IncydentServiceImpl, 143
 saveIncydent, 149
 updateIncydent, 150
com.mklachl.sopkom.services.impl.KierowcaServiceImpl,
 178
 findKierowcaByAvailability, 179
 findKierowcaById, 179
 findKierowcaByImieAndNazwisko, 180
 findKierowcaByPesel, 180
 findKierowcaByPrawoJazdyWazneDoAfter, 181
 findKierowcaByPrawoJazdyWazneDoBefore, 181
 findKierowcaByUser, 181
 KierowcaServiceImpl, 179
 saveKierowca, 182
com.mklachl.sopkom.services.impl.KursServiceImpl,
 213
 findKursByHarmonogram, 214
 findKursByKierunek, 215
 findKursByLinia, 215
 findKursByTypAutobusu, 215
 KursServiceImpl, 214
 saveKurs, 216
 updateKurs, 218
com.mklachl.sopkom.services.impl.LiniaServiceImpl,
 238
 deleteLinia, 239
 findLiniaById, 239
 findLiniaByNumer, 240
 patchLinia, 240
 saveLinia, 241
com.mklachl.sopkom.services.impl.PrzejazdServiceImpl,
 324
 createDetachedPrzejazd, 326
 findAllDatesForKurs, 326
 findPrzejazdByAutobus, 326
 findPrzejazdByCenaZaLit, 327
 findPrzejazdByCenaZaLitGreaterThanOrEqual, 327

findPrzejazdByCenaZaLitrLessThan, 328
 findPrzejazdByDataKonca, 328
 findPrzejazdByDataKoncaAfter, 329
 findPrzejazdByDataKoncaBefore, 329
 findPrzejazdByDataStartu, 329
 findPrzejazdByDataStartuAfter, 330
 findPrzejazdByDataStartuBefore, 330
 findPrzejazdByDlugoscTrasy, 331
 findPrzejazdByDlugoscTrasyGreaterThan, 331
 findPrzejazdByDlugoscTrasyLessThan, 332
 findPrzejazdById, 332
 findPrzejazdByKierowca, 332
 findPrzejazdByKurs, 333
 findPrzejazdBySpalanie, 333
 findPrzejazdBySpalanieGreaterThan, 334
 findPrzejazdBySpalanieLessThan, 334
 findPrzejazdyByKursAndData, 335
 PrzejazdServiceImpl, 325
 savePrzejazd, 335, 336
 updatePrzejazd, 336
 com.mklachl.sopkom.services.impl.PrzystanekServiceImpl,
 375
 findPrzystanekByDlugoscGeograficzna, 376
 findPrzystanekById, 377
 findPrzystanekByKodPocztowy, 377
 findPrzystanekByMiasto, 377
 findPrzystanekByNazwa, 378
 findPrzystanekByPrzystanekOdwrotny, 378
 findPrzystanekBySzerokoscGeograficzna, 379
 findPrzystanekByUlica, 379
 PrzystanekServiceImpl, 376
 savePrzystanek, 380
 com.mklachl.sopkom.services.impl.UserServiceImpl,
 416
 findUserById, 417
 findUserByLogin, 417
 saveUser, 418
 UserService, 417
 com.mklachl.sopkom.services.IncydentService, 135
 findIncydentByAutobus, 135
 findIncydentByDate, 136
 findIncydentByDateAfter, 136
 findIncydentByDateBefore, 136
 findIncydentByDateBetween, 137
 findIncydentByDodatkoweInformacje, 137
 findIncydentById, 138
 findIncydentByKierowca, 138
 findIncydentByKoszty, 138
 findIncydentByKosztyGreaterThan, 139
 findIncydentByKosztyLessThan, 139
 findIncydentByPrzejazd, 140
 findIncydentByTyp, 140
 saveIncydent, 140, 141
 updateIncydent, 141
 com.mklachl.sopkom.services.KierowcaService, 174
 findKierowcaByAvailability, 174
 findKierowcaById, 175
 findKierowcaByImieAndNazwisko, 175
 findKierowcaByPesel, 176
 findKierowcaByPrawoJazdyWazneDoAfter, 176
 findKierowcaByPrawoJazdyWazneDoBefore, 176
 findKierowcaByUser, 177
 saveKierowca, 177
 com.mklachl.sopkom.services.KursService, 210
 findKursByHarmonogram, 210
 findKursByKierunek, 211
 findKursByLinia, 211
 findKursByTypAutobusu, 211
 saveKurs, 212
 updateKurs, 213
 com.mklachl.sopkom.services.LiniaService, 236
 deleteLinia, 236
 findLiniaById, 237
 findLiniaByNumer, 237
 patchLinia, 237
 saveLinia, 238
 com.mklachl.sopkom.services.PrzejazdService, 313
 createDetachedPrzejazd, 314
 findAllDatesForKurs, 315
 findPrzejazdByAutobus, 315
 findPrzejazdByCenaZaLitr, 315
 findPrzejazdByCenaZaLitrGreaterThan, 316
 findPrzejazdByCenaZaLitrLessThan, 316
 findPrzejazdByDataKonca, 317
 findPrzejazdByDataKoncaAfter, 317
 findPrzejazdByDataKoncaBefore, 317
 findPrzejazdByDataStartu, 318
 findPrzejazdByDataStartuAfter, 318
 findPrzejazdByDataStartuBefore, 318
 findPrzejazdByDlugoscTrasy, 319
 findPrzejazdByDlugoscTrasyGreaterThan, 319
 findPrzejazdByDlugoscTrasyLessThan, 320
 findPrzejazdById, 320
 findPrzejazdByKierowca, 320
 findPrzejazdByKurs, 321
 findPrzejazdBySpalanie, 321
 findPrzejazdBySpalanieGreaterThan, 322
 findPrzejazdBySpalanieLessThan, 322
 findPrzejazdyByKursAndData, 322
 savePrzejazd, 323
 updatePrzejazd, 324
 com.mklachl.sopkom.services.PrzystanekService, 371
 findPrzystanekByDlugoscGeograficzna, 371
 findPrzystanekById, 372
 findPrzystanekByKodPocztowy, 372
 findPrzystanekByMiasto, 372
 findPrzystanekByNazwa, 373
 findPrzystanekByPrzystanekOdwrotny, 373
 findPrzystanekBySzerokoscGeograficzna, 374
 findPrzystanekByUlica, 374
 savePrzystanek, 374, 375
 com.mklachl.sopkom.services.UserService, 414
 findUserById, 415
 findUserByLogin, 415
 saveUser, 416
 com.mklachl.sopkom.SOPKoMApplication, 402

main, 403
com.mklachl.sopkom.topdf, 22
com.mklachl.sopkom.topdf.Pdf, 246
genRozklad, 246
genruij, 246
getAllByPrzystanekId, 246
getLiczbaLiniiNaPrzystanku, 247
kursRepository, 247
liniaRepos, 247
przystanekRepos, 247
przystanekWliniRepos, 247
createDetachedPrzejazd
com.mklachl.sopkom.services.impl.PrzejazdServiceImpl, 326
com.mklachl.sopkom.services.PrzejazdService, 314
createHarmonogram
com.mklachl.sopkom.controller.HarmonogramController, 81
CustomUserDetailsService
com.mklachl.sopkom.security.CustomUserDetailsService, 72
DateWrapperDto
com.mklachl.sopkom.model.dto.DateWrapperDto, 73
deleteAutobus
com.mklachl.sopkom.controller.AutobusController, 35
deleteHarmonogram
com.mklachl.sopkom.controller.HarmonogramController, 81
deleteIncydent
com.mklachl.sopkom.controller.IncydentController, 114
deleteKierowca
com.mklachl.sopkom.controller.KierowcaController, 159
deleteKurs
com.mklachl.sopkom.controller.KursController, 188
deleteLinia
com.mklachl.sopkom.controller.LiniaController, 223
com.mklachl.sopkom.services.impl.LiniaServiceImpl, findAllByAvailable
com.mklachl.sopkom.services.LiniaService, 236
deletePrzejazd
com.mklachl.sopkom.controller.PrzejazdController, 269
deletePrzystanek
com.mklachl.sopkom.controller.PrzystanekController, 343
DISPATCHER
com.mklachl.sopkom.model.entity.Role, 392
dlugoscGeograficzna
com.mklachl.sopkom.model.dto.PrzystanekDtoFull, 364
doFilterInternal
com.mklachl.sopkom.filter.LoggingFilter, 242

DRIVER
com.mklachl.sopkom.model.entity.Role, 392
equals
com.mklachl.sopkom.model.entity.Kierowca, 154
findAll
com.mklachl.sopkom.controller.KierowcaController, 160
com.mklachl.sopkom.repository.AutobusRepository, 44
com.mklachl.sopkom.repository.BiletRepository, 67
com.mklachl.sopkom.repository.HarmonogramRepository, 91
com.mklachl.sopkom.repository.IncydentRepository, 130
com.mklachl.sopkom.repository.KierowcaRepository, 170
com.mklachl.sopkom.repository.KursPrzystanekWliniRepository, 204
com.mklachl.sopkom.repository.KursRepository, 207
com.mklachl.sopkom.repository.LiniaRepository, 235
com.mklachl.sopkom.repository.PrzejazdBiletRepository, 266
com.mklachl.sopkom.repository.PrzejazdKursPrzystanekWliniRepository, 301
com.mklachl.sopkom.repository.PrzejazdRepository, 304
com.mklachl.sopkom.repository.PrzystanekRepository, 367
com.mklachl.sopkom.repository.PrzystanekWliniRepository, 387
findAllByAutobus
com.mklachl.sopkom.repository.IncydentRepository, 130
com.mklachl.sopkom.repository.PrzejazdRepository, 304
findAllByAvailability
com.mklachl.sopkom.repository.KierowcaRepository, 170
findAllByAvailable
com.mklachl.sopkom.repository.AutobusRepository, 44
findAllByBilet
com.mklachl.sopkom.repository.PrzejazdBiletRepository, 266
findAllByCena
com.mklachl.sopkom.repository.BiletRepository, 67
findAllByCenaBiletu
com.mklachl.sopkom.repository.PrzejazdBiletRepository, 266
findAllByCenaBiletuGreaterThanOrEqual
com.mklachl.sopkom.repository.PrzejazdBiletRepository, 267
findAllByCenaBiletuLessThan

com.mklachl.sopkom.repository.PrzejazdBiletRepository,	310
267	findAllByDodatkoweInflLike
findAllByCenaZaLitr	com.mklachl.sopkom.repository.HarmonogramRepository,
305	92
com.mklachl.sopkom.repository.PrzejazdRepository,	findAllByDodatkoweInformacje
305	com.mklachl.sopkom.repository.IncydentRepository,
com.mklachl.sopkom.repository.PrzejazdRepository,	132
305	findAllByGodzinna
com.mklachl.sopkom.repository.PrzejazdRepository,	com.mklachl.sopkom.repository.KursPrzystanekWliniRepository,
305	205
com.mklachl.sopkom.repository.HarmonogramRepository,	findAllByGodzinnaAfter
91	com.mklachl.sopkom.repository.KursPrzystanekWliniRepository,
com.mklachl.sopkom.repository.HarmonogramRepository,	205
306	findAllByGodzinnaBefore
com.mklachl.sopkom.repository.PrzejazdRepository,	com.mklachl.sopkom.repository.KursPrzystanekWliniRepository,
306	205
com.mklachl.sopkom.repository.PrzejazdRepository,	findAllByHarmonogram
307	com.mklachl.sopkom.repository.KursRepository,
com.mklachl.sopkom.repository.PrzejazdRepository,	207
307	findAllByImieAndNazwisko
com.mklachl.sopkom.repository.PrzejazdRepository,	com.mklachl.sopkom.repository.KierowcaRepository,
307	171
com.mklachl.sopkom.repository.PrzejazdRepository,	findAllByImieAndNazwiskoLike
307	com.mklachl.sopkom.repository.KierowcaRepository,
com.mklachl.sopkom.repository.PrzejazdRepository,	171
308	findAllByKierowca
com.mklachl.sopkom.repository.PrzejazdRepository,	com.mklachl.sopkom.repository.IncydentRepository,
308	132
com.mklachl.sopkom.repository.PrzejazdRepository,	com.mklachl.sopkom.repository.PrzejazdRepository,
308	311
com.mklachl.sopkom.repository.PrzejazdRepository,	findAllByKierunek
308	com.mklachl.sopkom.repository.KursRepository,
com.mklachl.sopkom.repository.PrzejazdRepository,	208
308	findAllByKodPocztowy
com.mklachl.sopkom.repository.PrzejazdRepository,	com.mklachl.sopkom.repository.PrzystanekRepository,
308	368
com.mklachl.sopkom.repository.IncydentRepository,	findAllByKoszty
131	com.mklachl.sopkom.repository.IncydentRepository,
com.mklachl.sopkom.repository.IncydentRepository,	133
131	findAllByKosztyGreaterThan
com.mklachl.sopkom.repository.IncydentRepository,	com.mklachl.sopkom.repository.IncydentRepository,
131	133
com.mklachl.sopkom.repository.IncydentRepository,	findAllByKosztyLessThan
131	com.mklachl.sopkom.repository.IncydentRepository,
com.mklachl.sopkom.repository.IncydentRepository,	134
132	findAllByKurs
com.mklachl.sopkom.repository.IncydentRepository,	com.mklachl.sopkom.repository.KursPrzystanekWliniRepository,
132	206
com.mklachl.sopkom.repository.PrzystanekRepository,	com.mklachl.sopkom.repository.PrzejazdRepository,
367	311
com.mklachl.sopkom.repository.PrzejazdRepository,	findAllByKursAndData
310	com.mklachl.sopkom.repository.PrzejazdRepository,
com.mklachl.sopkom.repository.PrzejazdRepository,	312
310	findAllByKursPrzystanekWlini
com.mklachl.sopkom.repository.PrzejazdRepository,	com.mklachl.sopkom.repository.PrzejazdKursPrzystanekWliniRepository,
310	301
com.mklachl.sopkom.repository.PrzejazdRepository,	findAllByLinia
310	com.mklachl.sopkom.repository.KursRepository,

com.mklachl.sopkom.repository.PrzystanekWliniRepository, 206	com.mklachl.sopkom.repository.KursPrzystanekWliniRepository,
387	findAllByRealnaGodzinna
findAllByMiasto	com.mklachl.sopkom.repository.PrzejazdKursPrzystanekWliniRepository
com.mklachl.sopkom.repository.PrzystanekRepository, 302	findAllByRealnaGodzinnaAfter
368	com.mklachl.sopkom.repository.PrzejazdKursPrzystanekWliniRepository
findAllByNazwa	findAllByRealnaGodzinnaBefore
com.mklachl.sopkom.repository.BiletRepository, 302	com.mklachl.sopkom.repository.PrzejazdKursPrzystanekWliniRepository
68	findAllByRealnaGodzinnaBefore
com.mklachl.sopkom.repository.HarmonogramRepository, com.mklachl.sopkom.repository.PrzejazdKursPrzystanekWliniRepository	303
92	com.mklachl.sopkom.repository.HarmonogramRepository, 94
findAllByNazwaLike	findAllBySob
com.mklachl.sopkom.repository.PrzystanekRepository, 368	com.mklachl.sopkom.repository.HarmonogramRepository,
com.mklachl.sopkom.repository.HarmonogramRepository, com.mklachl.sopkom.repository.PrzejazdRepository, 312	findAllBySpalanie
92	com.mklachl.sopkom.repository.HarmonogramRepository, com.mklachl.sopkom.repository.PrzejazdRepository, 312
findAllByNumerRejestracyjnyLike	findAllBySpalanieGreaterThan
com.mklachl.sopkom.repository.AutobusRepository, 44	com.mklachl.sopkom.repository.PrzejazdRepository, 312
findAllByPeselLike	findAllBySpalanieLessThan
com.mklachl.sopkom.repository.KierowcaRepository, 171	com.mklachl.sopkom.repository.PrzejazdRepository, 313
findAllByPia	findAllBySro
com.mklachl.sopkom.repository.HarmonogramRepository, com.mklachl.sopkom.repository.HarmonogramRepository, 93	findAllByStatus
com.mklachl.sopkom.repository.HarmonogramRepository, com.mklachl.sopkom.repository.AutobusRepository, 93	47
findAllByPrawoJazdyWazneDoAfter	findAllBySzerokoscGeograficzna
com.mklachl.sopkom.repository.KierowcaRepository, 172	com.mklachl.sopkom.repository.PrzystanekRepository, 369
findAllByPrawoJazdyWazneDoBefore	findAllByTyp
com.mklachl.sopkom.repository.KierowcaRepository, 172	com.mklachl.sopkom.repository.IncydentRepository, 134
findAllByPrzebieg	findAllByTypAutobusu
com.mklachl.sopkom.repository.AutobusRepository, 45	com.mklachl.sopkom.repository.KursRepository, 208
findAllByPrzebiegGreaterThan	findAllByUlica
com.mklachl.sopkom.repository.AutobusRepository, 45	com.mklachl.sopkom.repository.PrzystanekRepository, 369
findAllByPrzebiegLessThan	findAllByUlicaLike
com.mklachl.sopkom.repository.AutobusRepository, 45	com.mklachl.sopkom.repository.PrzystanekRepository, 370
findAllByPrzejazd	findAllByWto
com.mklachl.sopkom.repository.IncydentRepository, 134	com.mklachl.sopkom.repository.HarmonogramRepository, 94
com.mklachl.sopkom.repository.PrzejazdBiletRepository, 267	findAllDatesForKurs
com.mklachl.sopkom.repository.PrzejazdKursPrzystanekWliniRepository, 302	com.mklachl.sopkom.services.impl.PrzejazdServiceImpl, 220
302	com.mklachl.sopkom.services.PrzejazdService, 315
findAllByPrzystanek	findAllAutobusByAvailable
com.mklachl.sopkom.repository.PrzystanekWliniRepository, 387	com.mklachl.sopkom.services.AutobusService, 50
findAllByPrzystanekOdwrotny	com.mklachl.sopkom.services.impl.AutobusServiceImpl, 55
com.mklachl.sopkom.repository.PrzystanekRepository, 369	findAutobusById
com.mklachl.sopkom.repository.PrzystanekWlini	com.mklachl.sopkom.services.AutobusService, 51

com.mklachl.sopkom.services.impl.AutobusServiceImpl,	com.mklachl.sopkom.repository.AutobusRepository,
56	49
findAutobusByNumerRejestracyjny	findByUser
com.mklachl.sopkom.services.AutobusService,	com.mklachl.sopkom.repository.KierowcaRepository,
com.mklachl.sopkom.services.impl.AutobusServiceImpl,	173
56	findHarmonogramByCzw
findAutobusByPrzebieg	com.mklachl.sopkom.services.HarmonogramService,
com.mklachl.sopkom.services.AutobusService,	95
com.mklachl.sopkom.services.impl.AutobusServiceImpl,	com.mklachl.sopkom.services.impl.HarmonogramServiceImpl,
57	101
findAutobusByPrzebiegGreaterThanOrEqual	findHarmonogramByDodatkoweInLike
com.mklachl.sopkom.services.AutobusService,	com.mklachl.sopkom.services.HarmonogramService,
com.mklachl.sopkom.services.impl.AutobusServiceImpl,	96
57	com.mklachl.sopkom.services.impl.HarmonogramServiceImpl,
findAutobusByPrzebiegLessThan	101
com.mklachl.sopkom.services.AutobusService,	findHarmonogramById
com.mklachl.sopkom.services.impl.AutobusServiceImpl,	com.mklachl.sopkom.services.HarmonogramService,
57	96
findAutobusByPrzegladWaznyDoAfter	com.mklachl.sopkom.services.impl.HarmonogramServiceImpl,
com.mklachl.sopkom.services.AutobusService,	102
com.mklachl.sopkom.services.impl.AutobusServiceImpl,	findHarmonogramByNazwa
58	com.mklachl.sopkom.services.HarmonogramService,
findAutobusByPrzegladWaznyDoBefore	96
com.mklachl.sopkom.services.AutobusService,	com.mklachl.sopkom.services.impl.HarmonogramServiceImpl,
com.mklachl.sopkom.services.impl.AutobusServiceImpl,	102
58	findHarmonogramByNie
findAutobusByStatus	com.mklachl.sopkom.services.HarmonogramService,
com.mklachl.sopkom.services.AutobusService,	97
com.mklachl.sopkom.services.impl.AutobusServiceImpl,	com.mklachl.sopkom.services.impl.HarmonogramServiceImpl,
59	103
findById	findHarmonogramByPia
com.mklachl.sopkom.repository.LiniaRepository,	com.mklachl.sopkom.services.HarmonogramService,
235	97
findByLogin	com.mklachl.sopkom.services.impl.HarmonogramServiceImpl,
com.mklachl.sopkom.repository.UserRepository,	103
414	findHarmonogramByPon
findByNazwa	com.mklachl.sopkom.services.HarmonogramService,
com.mklachl.sopkom.repository.PrzystanekRepository,	98
370	com.mklachl.sopkom.services.impl.HarmonogramServiceImpl,
findByNazwiskoLike	103
com.mklachl.sopkom.repository.KierowcaRepository,	findHarmonogramBySob
172	com.mklachl.sopkom.services.HarmonogramService,
findByNumer	98
com.mklachl.sopkom.repository.LiniaRepository,	com.mklachl.sopkom.services.impl.HarmonogramServiceImpl,
235	104
findByNumerRejestracyjny	findHarmonogramBySro
com.mklachl.sopkom.repository.AutobusRepository,	com.mklachl.sopkom.services.HarmonogramService,
47	98
findByPesel	com.mklachl.sopkom.services.impl.HarmonogramServiceImpl,
com.mklachl.sopkom.repository.KierowcaRepository,	104
173	findHarmonogramByWto
findByPrzegladWaznyDoAfter	com.mklachl.sopkom.services.HarmonogramService,
com.mklachl.sopkom.repository.AutobusRepository,	99
47	com.mklachl.sopkom.services.impl.HarmonogramServiceImpl,
findByPrzegladWaznyDoBefore	105
com.mklachl.sopkom.repository.AutobusRepository,	findIncydentByAutobus
49	com.mklachl.sopkom.services.impl.IncydentServiceImpl,
findByStatusLike	143

com.mklachl.sopkom.services.IncydentService,	com.mklachl.sopkom.services.impl.IncydentServiceImpl,
135	148
findIncydentByDate	com.mklachl.sopkom.services.IncydentService,
	com.mklachl.sopkom.services.impl.IncydentServiceImpl,
143	140
com.mklachl.sopkom.services.IncydentService,	findKierowcaByAvailability
136	com.mklachl.sopkom.services.impl.KierowcaServiceImpl,
	179
findIncydentByDateAfter	com.mklachl.sopkom.services.KierowcaService,
	com.mklachl.sopkom.services.impl.IncydentServiceImpl,
144	174
com.mklachl.sopkom.services.IncydentService,	findKierowcaById
136	com.mklachl.sopkom.controller.KierowcaController,
	160
findIncydentByDateBefore	com.mklachl.sopkom.services.impl.KierowcaServiceImpl,
	179
com.mklachl.sopkom.services.impl.IncydentServiceImpl,	com.mklachl.sopkom.services.KierowcaService,
144	175
com.mklachl.sopkom.services.IncydentService,	findKierowcaByImieAndNazwisko
136	com.mklachl.sopkom.controller.KierowcaController,
	161
findIncydentByDateBetween	com.mklachl.sopkom.services.impl.KierowcaServiceImpl,
	180
com.mklachl.sopkom.services.impl.IncydentServiceImpl,	com.mklachl.sopkom.services.KierowcaService,
145	175
com.mklachl.sopkom.services.IncydentService,	findKierowcaByNazwisko
137	com.mklachl.sopkom.controller.KierowcaController,
	161
findIncydentByDodatkoweInformacje	com.mklachl.sopkom.services.impl.KierowcaController,
	161
com.mklachl.sopkom.services.impl.IncydentServiceImpl,	findKierowcaByPesel
145	com.mklachl.sopkom.controller.KierowcaController,
com.mklachl.sopkom.services.IncydentService,	161
137	com.mklachl.sopkom.services.impl.KierowcaServiceImpl,
	180
findIncydentById	com.mklachl.sopkom.services.KierowcaService,
	176
com.mklachl.sopkom.services.impl.IncydentServiceImpl,	findKierowcaByPrawoJazdyWazneDoAfter
146	com.mklachl.sopkom.controller.KierowcaController,
com.mklachl.sopkom.services.IncydentService,	162
138	com.mklachl.sopkom.services.impl.KierowcaServiceImpl,
	181
findIncydentByKoszty	com.mklachl.sopkom.services.KierowcaService,
	176
com.mklachl.sopkom.services.impl.IncydentServiceImpl,	findKierowcaByPrawoJazdyWazneDoBefore
146	com.mklachl.sopkom.controller.KierowcaController,
com.mklachl.sopkom.services.IncydentService,	162
138	com.mklachl.sopkom.services.impl.KierowcaServiceImpl,
	181
findIncydentByKosztyGreaterThan	com.mklachl.sopkom.services.KierowcaService,
	176
com.mklachl.sopkom.services.impl.IncydentServiceImpl,	findKierowcaByUser
147	com.mklachl.sopkom.controller.KierowcaController,
com.mklachl.sopkom.services.IncydentService,	163
139	com.mklachl.sopkom.services.impl.KierowcaServiceImpl,
	181
findIncydentByKosztyLessThan	com.mklachl.sopkom.services.KierowcaService,
	177
com.mklachl.sopkom.services.impl.IncydentServiceImpl,	findKursByHarmonogram
147	com.mklachl.sopkom.services.impl.KursServiceImpl,
com.mklachl.sopkom.services.IncydentService,	214
140	com.mklachl.sopkom.services.KursService, 210
findIncydentByTyp	

findKursByKierunek com.mklachl.sopkom.services.impl.KursServiceImpl, 215 com.mklachl.sopkom.services.KursService, 211	com.mklachl.sopkom.services.PrzejazdService, 318 findPrzejazdByDataStartuAfter com.mklachl.sopkom.services.impl.PrzejazdServiceImpl, 330
findKursByLinia com.mklachl.sopkom.services.impl.KursServiceImpl, 215 com.mklachl.sopkom.services.KursService, 211	com.mklachl.sopkom.services.PrzejazdService, 318 findPrzejazdByDataStartuBefore com.mklachl.sopkom.services.impl.PrzejazdServiceImpl, 330
findKursByTypAutobusu com.mklachl.sopkom.services.impl.KursServiceImpl, 215 com.mklachl.sopkom.services.KursService, 211	com.mklachl.sopkom.services.PrzejazdService, 318
findLiniaById com.mklachl.sopkom.services.impl.LiniaServiceImpl, 239 com.mklachl.sopkom.services.LiniaService, 237	findPrzejazdByDlugoscTrasy com.mklachl.sopkom.services.impl.PrzejazdServiceImpl, 331 com.mklachl.sopkom.services.PrzejazdService, 319
findLiniaByNumer com.mklachl.sopkom.services.impl.LiniaServiceImpl, 240 com.mklachl.sopkom.services.LiniaService, 237	findPrzejazdByDlugoscTrasyGreaterThan com.mklachl.sopkom.services.impl.PrzejazdServiceImpl, 331
findPrzejazdByAutobus com.mklachl.sopkom.services.impl.PrzejazdServiceImpl, 326 com.mklachl.sopkom.services.PrzejazdService, 315	com.mklachl.sopkom.services.PrzejazdService, 319 findPrzejazdByDlugoscTrasyLessThan com.mklachl.sopkom.services.impl.PrzejazdServiceImpl, 332
findPrzejazdByCenaZaLitr com.mklachl.sopkom.services.impl.PrzejazdServiceImpl, 327 com.mklachl.sopkom.services.PrzejazdService, 315	com.mklachl.sopkom.services.PrzejazdService, 320 findPrzejazdById com.mklachl.sopkom.services.impl.PrzejazdServiceImpl, 332
findPrzejazdByCenaZaLitrGreaterThan com.mklachl.sopkom.services.impl.PrzejazdServiceImpl, 327 com.mklachl.sopkom.services.PrzejazdService, 316	com.mklachl.sopkom.services.PrzejazdService, 320 findPrzejazdByKierowca com.mklachl.sopkom.services.impl.PrzejazdServiceImpl, 332
findPrzejazdByCenaZaLitrLessThan com.mklachl.sopkom.services.impl.PrzejazdServiceImpl, 328 com.mklachl.sopkom.services.PrzejazdService, 316	com.mklachl.sopkom.services.PrzejazdService, 320 findPrzejazdByKurs com.mklachl.sopkom.services.impl.PrzejazdServiceImpl, 333
findPrzejazdByDataKonca com.mklachl.sopkom.services.impl.PrzejazdServiceImpl, 328 com.mklachl.sopkom.services.PrzejazdService, 317	com.mklachl.sopkom.services.PrzejazdService, 321 findPrzejazdBySpalanie com.mklachl.sopkom.services.impl.PrzejazdServiceImpl, 333
findPrzejazdByDataKoncaAfter com.mklachl.sopkom.services.impl.PrzejazdServiceImpl, 329 com.mklachl.sopkom.services.PrzejazdService, 317	com.mklachl.sopkom.services.PrzejazdService, 321 findPrzejazdBySpalanieGreaterThan com.mklachl.sopkom.services.impl.PrzejazdServiceImpl, 334
findPrzejazdByDataKoncaBefore com.mklachl.sopkom.services.impl.PrzejazdServiceImpl, 329 com.mklachl.sopkom.services.PrzejazdService, 317	com.mklachl.sopkom.services.PrzejazdService, 322 findPrzejazdBySpalanieLessThan com.mklachl.sopkom.services.impl.PrzejazdServiceImpl, 334
findPrzejazdByDataStartu com.mklachl.sopkom.services.impl.PrzejazdServiceImpl, 329	com.mklachl.sopkom.services.PrzejazdService, 322 findPrzejazdyByKursAndData

com.mklachl.sopkom.services.impl.PrzejazdServiceImpl	getAllAvailableDriversForKurs	com.mklachl.sopkom.controller.PrzejazdController,
335	270	
com.mklachl.sopkom.services.PrzejazdService,	generateRozklad	com.mklachl.sopkom.controller.RaportyController,
322	genRozklad	com.mklachl.sopkom.topdf.Pdf, 246
findPrzystanekByDlugoscGeograficzna	com.mklachl.sopkom.services.impl.PrzystanekServiceImpl,	391
376	genruj	com.mklachl.sopkom.topdf.Pdf, 246
com.mklachl.sopkom.services.PrzystanekService,	getAdres	com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLinia,
371	getAllBuses	354
findPrzystanekById	com.mklachl.sopkom.services.impl.PrzystanekServiceImpl	36
377	getAllBusesFilteredStatus	com.mklachl.sopkom.controller.AutobusController,
com.mklachl.sopkom.services.PrzystanekService,	com.mklachl.sopkom.services.PrzystanekService,	36
372	getAllByPrzystanekId	com.mklachl.sopkom.controller.HarmonogramController,
findPrzystanekByMiasto	com.mklachl.sopkom.services.impl.PrzystanekServiceImpl	246
377	getAllHarmonograms	83
com.mklachl.sopkom.services.PrzystanekService,	getAutobus	com.mklachl.sopkom.controller.KursController, 189
372	getAutobusId	com.mklachl.sopkom.controller.LiniaController,
findPrzystanekByNazwa	com.mklachl.sopkom.services.impl.PrzystanekServiceImpl	223
378	getAutobusId	com.mklachl.sopkom.controller.PrzystanekController,
com.mklachl.sopkom.services.PrzystanekService,	getAutobusId	343
373	getAutobusId	com.mklachl.sopkom.controller.LiniaController,
findPrzystanekByPrzystanekOdwrotny	com.mklachl.sopkom.services.impl.PrzystanekServiceImpl	223
378	getAutobusId	com.mklachl.sopkom.controller.PrzystanekController,
com.mklachl.sopkom.services.PrzystanekService,	getAutobusId	343
373	getAutobusId	com.mklachl.sopkom.controller.LiniaController,
findPrzystanekBySzerokoscGeograficzna	com.mklachl.sopkom.services.impl.PrzystanekServiceImpl,	40
379	getAutobusId	com.mklachl.sopkom.model.entity.Autobus, 32
com.mklachl.sopkom.services.PrzystanekService,	getAutobus	com.mklachl.sopkom.controller.PrzystanekController,
374	getAutobus	com.mklachl.sopkom.model.entity.Incydent, 108
findPrzystanekByUlica	com.mklachl.sopkom.services.impl.PrzystanekServiceImpl	250
379	getAutobus	com.mklachl.sopkom.model.entity.Incydent, 120
com.mklachl.sopkom.services.PrzystanekService,	getAutobus	com.mklachl.sopkom.model.entity.Przejazd, 274
374	getAutobus	com.mklachl.sopkom.model.entity.Przejazd, 278
findUserById	com.mklachl.sopkom.services.impl.UserServiceImpl,	getAutobusId
417	getAutobus	com.mklachl.sopkom.model.dto.IncydentDtoOutput,
com.mklachl.sopkom.services.UserService,	getAutobus	126
415	getAutobus	com.mklachl.sopkom.model.entity.Incydent, 108
findUserByLogin	com.mklachl.sopkom.services.impl.UserServiceImpl,	getAutobus
417	getAutobus	com.mklachl.sopkom.model.entity.Przejazd, 250
com.mklachl.sopkom.services.UserService,	getAutobus	getAutobusId
415	getAutobus	com.mklachl.sopkom.model.dto.IncydentDtoOutput,
gatAllAvailableBusesForKurs	com.mklachl.sopkom.controller.PrzejazdController,	279
269	getAutobus	com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput,
gatAllAvailableDatesForKurs	com.mklachl.sopkom.controller.PrzejazdController,	getAutobusNumerRejestracyjny
269	getAutobus	com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput,
com.mklachl.sopkom.model.entity.PrzejazdBilet,	getBilet	279
269	getBilet	com.mklachl.sopkom.model.entity.PrzejazdBilet,

258
getBileId
 com.mklachl.sopkom.model.dto.BiletDto, 65
 com.mklachl.sopkom.model.dto.PrzejazdBiletDto,
 262
 com.mklachl.sopkom.model.entity.Bilet, 61
getCena
 com.mklachl.sopkom.model.dto.BiletDto, 65
 com.mklachl.sopkom.model.entity.Bilet, 61
getCenaBiletu
 com.mklachl.sopkom.model.dto.PrzejazdBiletDto,
 262
 com.mklachl.sopkom.model.entity.PrzejazdBilet,
 258
getCenaZaLitr
 com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput, 201
 279
 com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate, mklachl.sopkom.model.entity.Harmonogram,
 288
 com.mklachl.sopkom.model.entity.Przejazd, 251
getData
 com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput, mklachl.sopkom.model.dto.kurs.OUTDATED.KursDtoOUTDATED
 279
 com.mklachl.sopkom.model.entity.Przejazd, 251
getDataKonca
 com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput, mklachl.sopkom.controller.HarmonogramController,
 279
 com.mklachl.sopkom.model.entity.Przejazd, 251
getDataPrzejazdu
 com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoInput, 85
 274
getDataStartu
 com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput,
 280
 com.mklachl.sopkom.model.entity.Przejazd, 251
getDate
 com.mklachl.sopkom.model.dto.DateWrapperDto,
 73
 com.mklachl.sopkom.model.dto.IncydentDto, 120
 com.mklachl.sopkom.model.dto.przejazd.PKPWLDtoUpdate, mklachl.sopkom.model.dto.kierowca.KierowcaDto,
 248
 com.mklachl.sopkom.model.dto.rozklad.RozkladDto, getIncydent
 396
 com.mklachl.sopkom.model.entity.Incydent, 108
getDlugoscGeograficzna
 com.mklachl.sopkom.model.dto.PrzystanekDto,
 347
 com.mklachl.sopkom.model.dto.PrzystanekDtoFull,
 359
 com.mklachl.sopkom.model.entity.Przystanek, 337
getDlugoscTrasy
 com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput, mklachl.sopkom.controller.IncydentController,
 280
 com.mklachl.sopkom.model.entity.Przejazd, 252
getDlugoszTrasy
 com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate, 115
 288
getDodatkoweInf
 com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto,
 85
 com.mklachl.sopkom.model.entity.Harmonogram,
 75
getDodatkoweInformacje
 com.mklachl.sopkom.model.dto.IncydentDto, 120
 com.mklachl.sopkom.model.entity.Incydent, 108
getEmail
 com.mklachl.sopkom.model.dto.UserDto, 410
getGodzinna
 com.mklachl.sopkom.model.dto.kurs.AbstractPrzystanekWKursieDto,
 29
 com.mklachl.sopkom.model.dto.kurs.OUTDATED.PrzystenekWKursie
 389
 com.mklachl.sopkom.model.entity.KursPrzystanekWlini,
 201
getHamonogramId
 com.mklachl.sopkom.model.entity.Kurs, 183
getHamonogram
 com.mklachl.sopkom.model.dto.kurs.KursDto, 193
 197
 com.mklachl.sopkom.model.entity.Kurs, 183
getHamonogramById
 com.mklachl.sopkom.controller.HarmonogramController,
 83
getHamonogramId
 com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto,
 85
 com.mklachl.sopkom.model.dto.kurs.AbstractKursDto,
 24
 com.mklachl.sopkom.model.dto.linia.LiniaDto, 227
 com.mklachl.sopkom.model.dto.przejazd.PKPWLDtoUpdate,
 249
 com.mklachl.sopkom.model.dto.UserDto, 410
 com.mklachl.sopkom.model.entity.User, 405
getImie
 mklachl.sopkom.model.dto.kierowca.KierowcaDto, 165
 com.mklachl.sopkom.model.entity.Kierowca, 155
getIncydent
 com.mklachl.sopkom.controller.IncydentController,
 114
getIncydentId
 com.mklachl.sopkom.model.dto.IncydentDto, 121
 com.mklachl.sopkom.model.entity.Incydent, 108
getIncydents
 com.mklachl.sopkom.controller.IncydentController,
 114
getIncydentsAfterDate
 mklachl.sopkom.controller.IncydentController,
 115
getIncydentsBeforeDate
 com.mklachl.sopkom.controller.IncydentController,
 115
getIncydentsBetweenDates
 com.mklachl.sopkom.controller.IncydentController,

getIncydentsByAutobus	116	com.mklachl.sopkom.model.entity.KursPrzystanekWlini,
com.mklachl.sopkom.controller.IncydentController,	116	201
getIncydentsByKierowca	116	com.mklachl.sopkom.model.entity.Przejazd, 252
com.mklachl.sopkom.controller.IncydentController,	116	getKursId
getIncydentsByPrzejazd	117	com.mklachl.sopkom.model.dto.kurs.AbstractKursDto,
com.mklachl.sopkom.controller.IncydentController,	117	25
getKierowca	127	com.mklachl.sopkom.model.dto.kurs.OUTDATED.KursDtoOUTDATED
com.mklachl.sopkom.model.dto.IncydentDtoOutput,	127	197
com.mklachl.sopkom.model.entity.Incydent, 109		com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoInput,
com.mklachl.sopkom.model.entity.Przejazd, 252		275
getKierowcaId	127	com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput,
com.mklachl.sopkom.model.dto.IncydentDto, 121		281
com.mklachl.sopkom.model.dto.KierowcaDto, 166		com.mklachl.sopkom.model.entity.Kurs, 184
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoInput		getKursPrzystanekWlini
com.mklachl.sopkom.model.entity.Kierowca, 155		com.mklachl.sopkom.model.entity.PrzejazdKursPrzystanekWlini,
getKierowcaIdNazwisko	280	293
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput,	201	getKursPrzystanekWlinii
getKieruenk	197	com.mklachl.sopkom.model.entity.Kurs, 184
com.mklachl.sopkom.model.dto.kurs.OUTDATED.KursDtoOUTDATED,		getKursPrzystanekWlinild
getKierunek	24	com.mklachl.sopkom.model.entity.KursPrzystanekWlini,
com.mklachl.sopkom.model.dto.kurs.AbstractKursDto,	24	383
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput,	280	getKursPrzystanekWlinild
com.mklachl.sopkom.model.entity.Kurs, 184		com.mklachl.sopkom.model.entity.KursPrzystanekWlini,
getKodPocztowy	347	getLiczbaBiletow
com.mklachl.sopkom.model.dto.PrzystanekDto,	347	com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput,
com.mklachl.sopkom.model.dto.PrzystanekDtoFull,	359	281
com.mklachl.sopkom.model.entity.Przystanek, 337		getLiczbaBiletowNormalnych
getKolejnosc	354	com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate,
com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLinia	354	289
com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLiniaInput	356	getLiczbaBiletowUlgowych
com.mklachl.sopkom.model.dto.PrzejazdKursPrzystanekWlini	297	com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate,
com.mklachl.sopkom.model.entity.PrzystanekWlini,	383	289
getKoszty	121	getLiczbaLiniiNaPrzystanku
com.mklachl.sopkom.model.dto.IncydentDto, 121		com.mklachl.sopkom.topdf.Pdf, 247
com.mklachl.sopkom.model.entity.Incydent, 109		getLinia
getKrotko	121	com.mklachl.sopkom.controller.LiniaController,
com.mklachl.sopkom.model.dto.IncydentDto, 121		224
com.mklachl.sopkom.model.entity.Incydent, 109		com.mklachl.sopkom.model.dto.kurs.KursDto, 193
getKurs	189	com.mklachl.sopkom.model.dto.kurs.OUTDATED.KursDtoOUTDATED,
com.mklachl.sopkom.controller.KursController, 189		197
com.mklachl.sopkom.model.dto.IncydentDtoOutput,	127	getLiniaId
		com.mklachl.sopkom.model.dto.kurs.AbstractKursDto,
		25
		com.mklachl.sopkom.model.entity.Linia, 219
		getLiniaNumer
		com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput,
		281
		getLoggedInUser
		com.mklachl.sopkom.controller.UserController,
		408
		getLogin
		com.mklachl.sopkom.model.entity.User, 405
		getLoginPage
		com.mklachl.sopkom.controller.LoginController,
		242

getLogoutPage	com.mklachl.sopkom.controller.LoginController, 243	getPrzejazd	com.mklachl.sopkom.controller.PrzejazdController, 270
getMessage	com.mklachl.sopkom.controller.SimpleMessage, 401	com.mklachl.sopkom.model.entity.Incydent, 109	com.mklachl.sopkom.model.entity.PrzejazdBilet, 258
getMiasto	com.mklachl.sopkom.model.dto.PrzystanekDto, 347	com.mklachl.sopkom.model.entity.PrzejazdKursPrzystanekWlini, 293	com.mklachl.sopkom.model.entity.PrzejazdBiletKursPrzystanekWliniDto, 293
getName	com.mklachl.sopkom.model.dto.UserDto, 410	getPrzejazdBilet	com.mklachl.sopkom.model.entity.Bilet, 61
	com.mklachl.sopkom.model.entity.User, 405	com.mklachl.sopkom.model.entity.Przejazd, 252	com.mklachl.sopkom.model.entity.PrzejazdBiletDto, 262
getNazwa	com.mklachl.sopkom.model.dto.BiletDto, 65	getPrzejazdBiletId	com.mklachl.sopkom.model.dto.PrzejazdBiletDto, 262
	com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto, 85	com.mklachl.sopkom.model.entity.PrzejazdBilet, 258	com.mklachl.sopkom.model.entity.PrzejazdBiletDto, 262
	com.mklachl.sopkom.model.dto.PrzejazdKursPrzystanekWliniDto, 297	getPrzejazdId	com.mklachl.sopkom.model.dto.IncydentDto, 122
	com.mklachl.sopkom.model.dto.PrzystanekDto, 348	com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput, 281	com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate, 289
	com.mklachl.sopkom.model.dto.PrzystanekDtoFull, 360	com.mklachl.sopkom.model.dto.PrzejazdBiletDto, 262	com.mklachl.sopkom.model.entity.Przejazd, 253
	com.mklachl.sopkom.model.entity.Bilet, 61	getPrzejazdKursPrzystanekWlini	com.mklachl.sopkom.model.entity.Przejazd, 253
	com.mklachl.sopkom.model.entity.Harmonogram, 75	getPrzejazdKursPrzystanekWliniId	com.mklachl.sopkom.model.dto.PrzejazdKursPrzystanekWliniDto, 297
	com.mklachl.sopkom.model.entity.Przystanek, 338	com.mklachl.sopkom.model.entity.PrzejazdKursPrzystanekWliniDto, 293	com.mklachl.sopkom.model.entity.PrzejazdKursPrzystanekWliniDto, 293
getNazwisko	com.mklachl.sopkom.model.dto.KierowcaDto, 166	getPrzejazdy	com.mklachl.sopkom.controller.PrzejazdController, 271
	com.mklachl.sopkom.model.entity.Kierowca, 155	getPrzewidywanaGodzinna	com.mklachl.sopkom.model.dto.PrzejazdKursPrzystanekWliniDto, 297
getNormalny	com.mklachl.sopkom.model.dto.rozklad.RozkladDto, 396	getPrzystanek	com.mklachl.sopkom.controller.PrzystanekController, 344
getNumer	com.mklachl.sopkom.model.dto.linia.LiniaDto, 227	com.mklachl.sopkom.model.dto.kurs.UPDATED.PrzystenekWKursieDto, 389	com.mklachl.sopkom.model.entity.PrzystanekWlini, 384
	com.mklachl.sopkom.model.entity.Linia, 219	getPrzystanekFull	com.mklachl.sopkom.controller.LiniaController, 224
getNumerRejestracyjny	com.mklachl.sopkom.model.dto.AutobusDto, 40	com.mklachl.sopkom.controller.PrzystanekController, 344	com.mklachl.sopkom.controller.PrzystanekController, 344
	com.mklachl.sopkom.model.entity.Autobus, 32	getPrzystanekId	com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLiniaInput, 357
getOdwracalny	com.mklachl.sopkom.model.dto.kurs.KursDto, 194	com.mklachl.sopkom.model.dto.PrzejazdKursPrzystanekWliniDto, 298	com.mklachl.sopkom.model.entity.PrzystanekWliniDto, 298
getPassword	com.mklachl.sopkom.model.dto.UserDto, 410	com.mklachl.sopkom.model.dto.PrzystanekDto, 348	com.mklachl.sopkom.model.dto.PrzystanekDtoFull, 348
	com.mklachl.sopkom.model.entity.User, 405		
getPesel	com.mklachl.sopkom.model.dto.KierowcaDto, 166		
	com.mklachl.sopkom.model.entity.Kierowca, 155		
getPrawoJazdyWazneDo	com.mklachl.sopkom.model.dto.KierowcaDto, 166		
	com.mklachl.sopkom.model.entity.Kierowca, 156		
getPrzebieg	com.mklachl.sopkom.model.dto.AutobusDto, 40		
	com.mklachl.sopkom.model.entity.Autobus, 32		
getPrzegladWaznyDo	com.mklachl.sopkom.model.dto.AutobusDto, 40		
	com.mklachl.sopkom.model.entity.Autobus, 32		

getPrzystanekId	360	getRoles	com.mklachl.sopkom.model.dto.UserDto, 411
getPrzystanekKurs_id	com.mklachl.sopkom.model.entity.Przystanek, 338	getRole	com.mklachl.sopkom.model.entity.User, 406
getPrzystanekKurs_id	com.mklachl.sopkom.model.dto.kurs.OLDATED.PrzystanekDto, 389	getRole_M4	com.mklachl.sopkom.model.dto.rozklad.RozkladDto, 396
getPrzystanekOdwrotny	com.mklachl.sopkom.model.dto.PrzystanekDto, 348	getSpalanie	com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput, 282
getPrzystanekOdwrotny	com.mklachl.sopkom.model.dto.PrzystanekDtoFull, 360	getSpalanie	com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate, 290
getPrzystanekOdwrotny	com.mklachl.sopkom.model.entity.Przystanek, 338	getSpalanie	com.mklachl.sopkom.model.entity.Przejazd, 253
getPrzystanekOdwrotnyNazwa	com.mklachl.sopkom.model.dto.PrzystanekDto, 348	getStatus	com.mklachl.sopkom.model.dto.AutobusDto, 41
getPrzystanekOdwrotnyNazwa	com.mklachl.sopkom.model.entity.Przystanek, 338	getStatus	com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput, 280
getPrzystanekWkursield	com.mklachl.sopkom.model.dto.kurs.AbstractPrzystanekWKursieDto, 30	getString	com.mklachl.sopkom.controller.Controller, 71
getPrzystanekWkursield	com.mklachl.sopkom.model.dto.kurs.InputPrzystanekWKursieDto, 153	getString	com.mklachl.sopkom.controller.Controller, 71
getPrzystanekWLini	com.mklachl.sopkom.model.dto.kurs.PrzystanekWKursieDto, 382	getSzerokoscGeograficzna	com.mklachl.sopkom.model.dto.PrzystanekDto, 349
getPrzystanekWLini	com.mklachl.sopkom.model.entity.KursPrzystanekWLini, 201	getSzerokoscGeograficzna	com.mklachl.sopkom.model.dto.PrzystanekDtoFull, 361
getPrzystanekWLini	com.mklachl.sopkom.model.entity.PrzystanekWLini, 384	getTemplateKurs	com.mklachl.sopkom.model.entity.Przystanek, 339
getPrzystanekWLini	com.mklachl.sopkom.model.dto.linia.PrzystanekDto, 354	getTemplateKurs	com.mklachl.sopkom.controller.KursController, 189
getPrzystanki	com.mklachl.sopkom.model.dto.kurs.InputKursDto, 151	getTemplateKursInput	com.mklachl.sopkom.controller.KursController, 190
getPrzystanki	com.mklachl.sopkom.model.dto.kurs.KursDto, 194	getTyp	com.mklachl.sopkom.model.dto.IncydentDto, 122
getPrzystanki	com.mklachl.sopkom.model.dto.kurs.OLDATED.KursDto, 197	getTyp	com.mklachl.sopkom.model.entity.Incydent, 110
getPrzystanki	com.mklachl.sopkom.model.dto.linia.LiniaDtoInput, 230	getTypAutobusu	com.mklachl.sopkom.model.entity.Kurs, 185
getPrzystanki	com.mklachl.sopkom.model.dto.linia.LiniaDtoOutput, 232	getTypAutobusu	com.mklachl.sopkom.model.dto.PrzystanekDto, 349
getPrzystanki	com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput, 282	getTypAutobusu	com.mklachl.sopkom.model.dto.PrzystanekDtoFull, 361
getPrzystanki	com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate, 289	getTypAutobusu	com.mklachl.sopkom.model.entity.Przystanek, 339
getPrzystankiWLini	com.mklachl.sopkom.model.entity.Linia, 219	getUser	com.mklachl.sopkom.model.dto.KierowcaDto, 167
getPrzystankiWLini	com.mklachl.sopkom.model.entity.Przystanek, 339	getUser	com.mklachl.sopkom.model.entity.Kierowca, 156
getPrzyszlePrzejazdy	com.mklachl.sopkom.controller.PrzejazdController, 271	getUserFromContext	com.mklachl.sopkom.helpers.Helpers, 106
getRealnaGodzinna	com.mklachl.sopkom.model.dto.PrzejazdKursPrzystanekWLini, 298	godzinna	com.mklachl.sopkom.model.dto.kurs.AbstractPrzystanekWKursieDto, 301
getRealnaGodzinna	com.mklachl.sopkom.model.entity.PrzejazdKursPrzystanekWLini, 294	godzinna	com.mklachl.sopkom.model.dto.kurs.OLDATED.PrzystanekWKursieDto, 390
getRole	com.mklachl.sopkom.model.dto.UserDto, 411	harmonogram	com.mklachl.sopkom.model.dto.kurs.OLDATED.KursDto, 199

HarmonogramDto	com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto
com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto	86
com.mklachl.sopkom.model.entity.Harmonogram,	76
harmonogramId	
com.mklachl.sopkom.model.dto.kurs.AbstractKursDtoIsPon	
27	
harmonogramRepository	com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto
com.mklachl.sopkom.controller.PrzejazdController,	86
272	
HarmonogramServiceImpl	com.mklachl.sopkom.model.entity.Harmonogram,
com.mklachl.sopkom.services.impl.HarmonogramServiceImpl	76
101	
hashCode	isSob
com.mklachl.sopkom.model.entity.Kierowca,	com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto
156	86
id	com.mklachl.sopkom.model.entity.Harmonogram,
com.mklachl.sopkom.model.dto.linia.LiniaDto,	76
228	
IncydentDto	isSro
com.mklachl.sopkom.model.dto.IncydentDto,	com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto
119	87
IncydentDtoOutput	com.mklachl.sopkom.model.entity.Harmonogram,
com.mklachl.sopkom.model.dto.IncydentDtoOutput,	77
126	
IncydentException	isWto
com.mklachl.sopkom.exceptions.IncydentException,	com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto
129	87
incydentRepository	com.mklachl.sopkom.model.entity.Harmonogram,
com.mklachl.sopkom.controller.IncydentController,	77
118	
incydentService	KierowcaController
com.mklachl.sopkom.controller.IncydentController,	com.mklachl.sopkom.controller.KierowcaController,
118	159
IncydentServiceImpl	KierowcaDto
com.mklachl.sopkom.services.impl.IncydentServiceImpl,	com.mklachl.sopkom.model.dto.KierowcaDto,
143	165
inject	kierowcaRepository
com.mklachl.sopkom.helpers.Helpers,	com.mklachl.sopkom.controller.PrzejazdController,
107	272
InputKursDto	KierowcaServiceImpl
com.mklachl.sopkom.model.dto.kurs.InputKursDto,	com.mklachl.sopkom.services.impl.KierowcaServiceImpl,
151	179
InputPrzystanekWKursieDto	kieruenk
com.mklachl.sopkom.model.dto.kurs.InputPrzystanekWKursieDto	com.mklachl.sopkom.model.dto.kurs.OUTDATED.KursDtoOUTDATED
153	199
isCzw	kierunek
com.mklachl.sopkom.model.dto.harmonogram.Harmonogram	com.mklachl.sopkom.model.dto.kurs.AbstractKursDto,
85	27
com.mklachl.sopkom.model.entity.Harmonogram,	com.mklachl.sopkom.model.dto.PrzystanekDtoFull,
75	364
isNie	kodPocztowy
com.mklachl.sopkom.model.dto.harmonogram.Harmonogram	com.mklachl.sopkom.model.dto.PrzystanekDtoFull,
86	364
com.mklachl.sopkom.model.entity.Harmonogram,	com.mklachl.sopkom.model.dto.kurs.OUTDATED.KursDtoOUTDATED
76	390
isOdwracalna	KognatDto
com.mklachl.sopkom.model.dto.linia.LiniaDtoOutput,	com.mklachl.sopkom.model.dto.kurs.KursDto,
kursId	193
232	
isOdwracalny	KursDtoOUTDATED
com.mklachl.sopkom.model.dto.kurs.OUTDATED.KursDtoOUTDATED	com.mklachl.sopkom.model.dto.kurs.OUTDATED.KursDtoOUTDATED
198	199
isPia	kursRepo

com.mklachl.sopkom.controller.KursController, 191
kursRepository
com.mklachl.sopkom.controller.PrzejazdController, 272
com.mklachl.sopkom.topdf.Pdf, 247
kursService
com.mklachl.sopkom.controller.KursController, 191
com.mklachl.sopkom.controller.PrzejazdController, 272
KursServiceImpl
com.mklachl.sopkom.services.impl.KursServiceImpl, numer 214
kursy
com.mklachl.sopkom.controller.KursController, 191
linia
com.mklachl.sopkom.model.dto.kurs.OLDATED.KursDto, 199
LiniaDto
com.mklachl.sopkom.model.dto.linia.LiniaDto, 226
LiniaDtoInput
com.mklachl.sopkom.model.dto.linia.LiniaDtoInput, 229
LiniaDtoOutput
com.mklachl.sopkom.model.dto.linia.LiniaDtoOutput, 231
liniald
com.mklachl.sopkom.model.dto.kurs.AbstractKursDto, 28
LiniaNotFoundException
com.mklachl.sopkom.exceptions.LiniaNotFoundException, 234
liniaRepos
com.mklachl.sopkom.topdf.Pdf, 247
liniaRepository
com.mklachl.sopkom.controller.KursController, 191
liniaService
com.mklachl.sopkom.controller.KursController, 191
loadUserByUsername
com.mklachl.sopkom.security.CustomUserDetailsService, 72
main
com.mklachl.sopkom.SOPKOApplication, 403
MANAGER
com.mklachl.sopkom.model.entity.Role, 393
MANAGER_LOGIN
com.mklachl.sopkom.security.CommandLineRunnerTest, 70
MANAGER_NAME
com.mklachl.sopkom.security.CommandLineRunnerTest, 70
MANAGER_PASS
com.mklachl.sopkom.security.CommandLineRunnerTest, 70
Message
com.mklachl.sopkom.model.dto.Message, 244
message

com.mklachl.sopkom.controller.SimpleMessage, 402
com.mklachl.sopkom.model.dto.Message, 245
miasto
com.mklachl.sopkom.model.dto.PrzystanekDtoFull, 364
nazwa
com.mklachl.sopkom.model.dto.PrzystanekDtoFull, 364
numer
com.mklachl.sopkom.model.dto.linia.LiniaDto, 228
odwracalna
com.mklachl.sopkom.model.dto.linia.LiniaDtoOutput, 233
ODATED, 220
odwracalny
com.mklachl.sopkom.model.dto.kurs.OLDATED.KursDtoOUTDATED, 200
passwordEncoder
com.mklachl.sopkom.config.SecurityConfig, 400
patchLinia
com.mklachl.sopkom.services.impl.LiniaServiceImpl, 240
com.mklachl.sopkom.services.LiniaService, 237
PKPWLDtoUpdate
com.mklachl.sopkom.model.dto.przejazd.PKPWLDTUpdate, 248
PrzejazdBiletDto
com.mklachl.sopkom.model.dto.PrzejazdBiletDto, 261
PrzejazdDtoInput
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoInput, 274
PrzejazdDtoOutput
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput, 277, 278
PrzejazdDtoUpdate
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate, 288
PrzejazdKursPrzystanekWliniDto
com.mklachl.sopkom.model.dto.PrzejazdKursPrzystanekWliniDto, 296
przejazdRepository
com.mklachl.sopkom.controller.PrzejazdController, 273
przejazdService
com.mklachl.sopkom.controller.PrzejazdController, 273
PrzejazdServiceImpl
com.mklachl.sopkom.services.impl.PrzejazdServiceImpl, 325
przystanek
com.mklachl.sopkom.model.dto.kurs.OLDATED.PrzystenekWKursie, 390
PrzystanekDto

com.mklachl.sopkom.model.dto.PrzystanekDto, 346, 347
PrzystanekDtoDlaLinia
 com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLinia, 59, 60
 353
PrzystanekDtoDlaLinialInput
 com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLinialInpu99
 356
PrzystanekDtoFull
 com.mklachl.sopkom.model.dto.PrzystanekDtoFull, 359
przystanekId
 com.mklachl.sopkom.model.dto.PrzystanekDtoFull, 365
przystanekkurs_id
 com.mklachl.sopkom.model.dto.kurs.OLDATED.PrzystenekWKursieDto, 390
PrzystanekNotFoundException
 com.mklachl.sopkom.exceptions.PrzystanekNotFoundException, 163
 366
przystanekOdwrotny
 com.mklachl.sopkom.model.dto.PrzystanekDtoFull, 365
przystanekRepos
 com.mklachl.sopkom.topdf.Pdf, 247
PrzystanekServiceImpl
 com.mklachl.sopkom.services.impl.PrzystanekServiceImpl, 212
 376
PrzystanekWKursieDto
 com.mklachl.sopkom.model.dto.kurs.PrzystanekWKursieDto, 241
 381
przystanekWkursield
 com.mklachl.sopkom.model.dto.kurs.AbstractPrzystanekWKursieD271
 31
przystanekWliniRepos
 com.mklachl.sopkom.topdf.Pdf, 247
przystanki
 com.mklachl.sopkom.model.dto.kurs.InputKursDto, 152
 com.mklachl.sopkom.model.dto.kurs.OLDATED.KursDtoOutD380, 200
PrzystenekWKursieDtoOUTDATED
 com.mklachl.sopkom.model.dto.kurs.OLDATED.PrzystenekWKursieD388, 374, 375
przystnekWliniRepository
 com.mklachl.sopkom.controller.KursController, 192
repo
 com.mklachl.sopkom.controller.RozkladController, 394
RozkladDto
 com.mklachl.sopkom.model.dto.rozklad.RozkladDto, 395
rozkladHelper
 com.mklachl.sopkom.controller.RozkladController, 394
run
 com.mklachl.sopkom.security.CommandLineRunnerTest, 69
saveAutobus
 com.mklachl.sopkom.services.AutobusService, 54
 com.mklachl.sopkom.services.impl.AutobusServiceImpl,
saveHarmonogram
 com.mklachl.sopkom.services.HarmonogramService, 119
 com.mklachl.sopkom.services.impl.HarmonogramServiceImpl, 105, 106
saveIncydent
 com.mklachl.sopkom.controller.IncydentController, 117
com.mklachl.sopkom.services.impl.IncydentServiceImpl, 149
com.mklachl.sopkom.services.IncydentService, 143, 144
saveKierowca
 com.mklachl.sopkom.controller.KierowcaController, 163
com.mklachl.sopkom.services.impl.KierowcaServiceImpl, 182
com.mklachl.sopkom.services.KierowcaService, 177
saveKurs
 com.mklachl.sopkom.services.impl.KursServiceImpl, 216
com.mklachl.sopkom.services.KursService, 212
saveLinia
 com.mklachl.sopkom.services.impl.LiniaServiceImpl, 241
com.mklachl.sopkom.services.LiniaService, 238
savePrzejazd
 com.mklachl.sopkom.controller.PrzejazdController, 271
com.mklachl.sopkom.services.impl.PrzejazdServiceImpl, 335, 336
com.mklachl.sopkom.services.PrzejazdService, 323
savePrzystanek
 com.mklachl.sopkom.services.impl.PrzystanekService, 374, 375
saveUser
 com.mklachl.sopkom.services.impl.UserService, 418
com.mklachl.sopkom.services.UserService, 416
securityFilterChain
 com.mklachl.sopkom.config.SecurityConfig, 400
setAdres
 com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLinia, 354
setAutbousId
 com.mklachl.sopkom.model.dto.AutobusDto, 41
 com.mklachl.sopkom.model.entity.Autobus, 33
setAutobus
 com.mklachl.sopkom.model.dto.IncydentDtoOutput, 127

com.mklachl.sopkom.model.entity.Incydent, 110	com.mklachl.sopkom.model.dto.przejazd.PKPWLDtoUpdate, 249
com.mklachl.sopkom.model.entity.Przejazd, 253	com.mklachl.sopkom.model.dto.rozklad.RozkladDto, 397
setAutobusId	com.mklachl.sopkom.model.entity.Incydent, 110
com.mklachl.sopkom.model.dto.IncydentDto, 122	setDlugoscGeograficzna
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoInput, 275	com.mklachl.sopkom.model.dto.PrzystanekDto,
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput, 282	com.mklachl.sopkom.model.dto.PrzystanekDtoFull, 361
setAutobusNazwa	com.mklachl.sopkom.model.entity.Przystanek, 339
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput, 282	setDlugoscTrasy
setAutobusNumerRejestracyjny	com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput, 284
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput, 283	com.mklachl.sopkom.model.entity.Przejazd, 255
setBilet	setDlugoszTrasy
com.mklachl.sopkom.model.entity.PrzejazdBilet, 259	com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate, 290
setBiletid	setDodatkoweInfnf
com.mklachl.sopkom.model.dto.BiletDto, 65	com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto, 88
com.mklachl.sopkom.model.dto.PrzejazdBiletDto, 263	com.mklachl.sopkom.model.entity.Harmonogram, 78
com.mklachl.sopkom.model.entity.Bilet, 62	setDodatkoweInformacje
setCena	com.mklachl.sopkom.model.dto.IncydentDto, 123
com.mklachl.sopkom.model.dto.BiletDto, 66	com.mklachl.sopkom.model.entity.Incydent, 111
com.mklachl.sopkom.model.entity.Bilet, 63	setEmail
setCenaBiletu	com.mklachl.sopkom.model.dto.UserDto, 411
com.mklachl.sopkom.model.dto.PrzejazdBiletDto, 263	setGodzinna
com.mklachl.sopkom.model.entity.PrzejazdBilet, 260	com.mklachl.sopkom.model.dto.kurs.AbstractPrzystanekWKursieDto, 30
setCenaZaLitr	com.mklachl.sopkom.model.dto.kurs.PrzystanekWKursieDto, 389
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput, 283	com.mklachl.sopkom.model.entity.KursPrzystanekWlini, 202
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate, 290	setHarmonogramId
com.mklachl.sopkom.model.entity.Przejazd, 254	com.mklachl.sopkom.model.entity.Harmonogram, 78
setData	setHarmonogram
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput, 283	com.mklachl.sopkom.model.dto.kurs.KursDto, 194
com.mklachl.sopkom.model.entity.Przejazd, 254	com.mklachl.sopkom.model.dto.kurs.OUTDATED.KursDtoOUTDATED, 198
setDataKonca	com.mklachl.sopkom.model.entity.Kurs, 185
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput, 284	setHarmonogramId
com.mklachl.sopkom.model.entity.Przejazd, 254	com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto, 88
setDataPrzejazdu	com.mklachl.sopkom.model.dto.kurs.AbstractKursDto, 25
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput, 275	setImlie
setDataStartu	com.mklachl.sopkom.model.dto.linia.LiniaDto, 227
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput, 284	com.mklachl.sopkom.model.dto.przejazd.PKPWLDtoUpdate, 249
com.mklachl.sopkom.model.entity.Przejazd, 255	com.mklachl.sopkom.model.dto.UserDto, 412
setData	com.mklachl.sopkom.model.entity.User, 406
com.mklachl.sopkom.model.dto.DateWrapperDto, 73	setImlie
com.mklachl.sopkom.model.dto.IncydentDto, 123	com.mklachl.sopkom.model.dto.KierowcaDto, 167
	com.mklachl.sopkom.model.entity.Kierowca, 156

com.mklachl.sopkom.model.dto.IncydentDto, 123	276
com.mklachl.sopkom.model.entity.Incydent, 111	com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput, 285
setKierowca	com.mklachl.sopkom.model.entity.Kurs, 186
com.mklachl.sopkom.model.dto.IncydentDtoOutput, 128	setKursPrzystanekWlini
com.mklachl.sopkom.model.entity.Incydent, 111	com.mklachl.sopkom.model.entity.PrzejazdKursPrzystanekWlini, 294
com.mklachl.sopkom.model.entity.Przejazd, 255	setKursPrzystanekWlinii
setKierowcaId	com.mklachl.sopkom.model.entity.Kurs, 186
com.mklachl.sopkom.model.dto.IncydentDto, 124	com.mklachl.sopkom.model.entity.PrzystanekWlini,
com.mklachl.sopkom.model.dto.KierowcaDto, 167	setKursPrzystanekWlinild
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoInput, 276	com.mklachl.sopkom.model.entity.KursPrzystanekWlini, 203
com.mklachl.sopkom.model.entity.Kierowca, 157	com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput, 285
setKierowcaImieNazwisko	setKursPrzystanekWliniNormalnych
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoInput, 284	com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate, 291
setKieruenk	setLinia
com.mklachl.sopkom.model.dto.kurs.OUTDATED.KursDtosetLiczbaBiletoUlgowych	com.mklachl.sopkom.model.dto.kurs.KursDto, 195
198	com.mklachl.sopkom.model.dto.kurs.OUTDATED.KursDtoOUTDATED
setKierunek	198
com.mklachl.sopkom.model.dto.kurs.AbstractKursDtosetLiczbaBiletoUlgowych	com.mklachl.sopkom.model.entity.Kurs, 186
26	com.mklachl.sopkom.model.entity.PrzystanekWlini, 385
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput, 285	setLiniald
com.mklachl.sopkom.model.entity.Kurs, 185	com.mklachl.sopkom.model.dto.kurs.AbstractKursDto, 26
setKodPocztowy	com.mklachl.sopkom.model.entity.Linia, 220
com.mklachl.sopkom.model.dto.PrzystanekDto, 350	setLinialdPartner
com.mklachl.sopkom.model.dto.PrzystanekDtoFull, 362	com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput, 286
com.mklachl.sopkom.model.entity.Przystanek, 340	setLogin
setKolejnosc	com.mklachl.sopkom.model.entity.User, 406
com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLinia, 355	setMessage
com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLinia, 357	com.mklachl.sopkom.model.dto.Message, 244
com.mklachl.sopkom.model.dto.PrzejazdKursPrzystanekWlini, 298	setMessage
com.mklachl.sopkom.model.entity.PrzystanekWlini, 384	com.mklachl.sopkom.controller.SimpleMessage, 402
setKoszty	SetMessageInt
com.mklachl.sopkom.model.dto.IncydentDto, 124	com.mklachl.sopkom.model.dto.Message, 245
com.mklachl.sopkom.model.entity.Incydent, 112	setMiasto
setKrotko	com.mklachl.sopkom.model.dto.PrzystanekDto, 350
com.mklachl.sopkom.model.dto.IncydentDto, 124	com.mklachl.sopkom.model.dto.PrzystanekDtoFull, 362
com.mklachl.sopkom.model.entity.Incydent, 112	com.mklachl.sopkom.model.entity.Przystanek, 340
setKurs	setName
com.mklachl.sopkom.model.dto.IncydentDtoOutput, 128	com.mklachl.sopkom.model.dto.UserDto, 412
com.mklachl.sopkom.model.entity.KursPrzystanekWlini, 203	com.mklachl.sopkom.model.entity.User, 407
com.mklachl.sopkom.model.entity.Przejazd, 255	setOUTDATED,
setKursId	com.mklachl.sopkom.model.dto.BiletDto, 66
com.mklachl.sopkom.model.dto.kurs.AbstractKursDto, 26	com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto,
com.mklachl.sopkom.model.dto.kurs.OUTDATED.KursDtosetLiczbaBiletoUlgowych, 198	
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoInput, 200	

88	com.mklachl.sopkom.model.dto.AutobusDto, 42
299	com.mklachl.sopkom.model.entity.Autobus, 34
350	setPrzejazd com.mklachl.sopkom.model.entity.Incydent, 112
362	com.mklachl.sopkom.model.entity.PrzejazdBilet, 260
350	com.mklachl.sopkom.model.entity.PrzejazdKursPrzystanekWlini, 294
78	setPrzejazdBilet com.mklachl.sopkom.model.entity.Bilet, 63
340	com.mklachl.sopkom.model.entity.Przejazd, 256
setNazwisko	setPrzejazdBiletId com.mklachl.sopkom.model.dto.PrzejazdBiletDto, 263
168	com.mklachl.sopkom.model.entity.PrzejazdBilet, 260
157	com.mklachl.sopkom.model.entity.Kierowca, 157
setNie	setPrzejazdId com.mklachl.sopkom.model.dto.IncydentDto, 125
89	com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput, 286
79	com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate, 291
setNormalny	com.mklachl.sopkom.model.dto.PrzejazdBiletDto, 265
397	com.mklachl.sopkom.model.entity.Przejazd, 256
setNumer	setPrzejazdKursPrzystanekWlini com.mklachl.sopkom.model.entity.Przejazd, 256
228	setPrzejazdKursPrzystanekWlinild com.mklachl.sopkom.model.dto.PrzejazdKursPrzystanekWliniDto, 299
220	com.mklachl.sopkom.model.entity.PrzejazdKursPrzystanekWlini, 295
41	setPrzewidywanaGodzinna com.mklachl.sopkom.model.dto.PrzejazdKursPrzystanekWliniDto, 299
33	setPrzystanek com.mklachl.sopkom.model.dto.kurs.OUTDATED.KursDto, 389
setOdwracalna	com.mklachl.sopkom.model.entity.PrzystanekWlini, 385
232	setPrzystanekDtl com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLiniaInput, 357
setOdwracalny	com.mklachl.sopkom.model.dto.PrzejazdKursPrzystanekWliniDto, 300
195	com.mklachl.sopkom.model.entity.PrzystanekDto, 351
198	com.mklachl.sopkom.model.dto.linia.PrzystanekDtoFull, 363
setPassword	com.mklachl.sopkom.model.entity.Przystanek, 341
412	setPrzystanekkurs_id com.mklachl.sopkom.model.dto.kurs.OUTDATED.PrzystenekWKursie, 390
407	setPrzystanekOdwrotny com.mklachl.sopkom.model.dto.PrzystanekDto, 351
setPesel	com.mklachl.sopkom.model.dto.PrzystanekDtoFull, 363
168	com.mklachl.sopkom.model.entity.KierowcaDto, 168
157	com.mklachl.sopkom.model.entity.Kierowca, 157
setPia	com.mklachl.sopkom.model.entity.Harmonogram, 79
89	setPrzystanek com.mklachl.sopkom.model.dto.kurs.OUTDATED.PrzystenekWKursie, 389
79	com.mklachl.sopkom.model.entity.Harmonogram, 79
setPon	com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDtl 89
89	com.mklachl.sopkom.model.entity.Harmonogram, 79
79	setPrzystanekDtl com.mklachl.sopkom.model.dto.PrzejazdKursPrzystanekWliniDto, 300
setPrawoJazdyWazneDo	com.mklachl.sopkom.model.entity.Przystanek, 341
168	setPrzystanekkurs_id com.mklachl.sopkom.model.dto.kurs.OUTDATED.PrzystenekWKursie, 390
158	setPrzystanekOdwrotny com.mklachl.sopkom.model.dto.PrzystanekDto, 351
setPrzebieg	com.mklachl.sopkom.model.dto.PrzystanekDtoFull, 363
42	com.mklachl.sopkom.model.entity.Autobus, 34
34	com.mklachl.sopkom.model.entity.Autobus, 34
setPrzegladWaznyDo	com.mklachl.sopkom.model.entity.Kierowca, 157

com.mklachl.sopkom.model.entity.Przystanek, 341	287
setPrzystanekOdwrotnyNazwa	com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate, 292
com.mklachl.sopkom.model.dto.PrzystanekDto, 351	com.mklachl.sopkom.model.entity.Przejazd, 257
setPrzystanekWkursield	setSro
com.mklachl.sopkom.model.dto.kurs.AbstractPrzystanekWkursield, 30	KursPrzystanekWkursield, 90
setPrzystanekWLini	com.mklachl.sopkom.model.entity.Harmonogram, 80
com.mklachl.sopkom.model.dto.kurs.InputPrzystanekWKursieDto, 153	setStatus
com.mklachl.sopkom.model.dto.kurs.PrzystanekWKursieDto, 382	com.mklachl.sopkom.model.dto.AutobusDto, 42
setPrzystanekWlini	com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput, 287
com.mklachl.sopkom.model.entity.KursPrzystanekWlini, 203	com.mklachl.sopkom.model.entity.Autobus, 34
setPrzystanekWlinild	setSzerokoscGeograficzna
com.mklachl.sopkom.model.entity.PrzystanekWlini, 386	com.mklachl.sopkom.model.dto.PrzystanekDto, 351
setPrzystanekwliniid	com.mklachl.sopkom.model.dto.PrzystanekDtoFull, 363
com.mklachl.sopkom.model.dto.linia.PrzystanekDtoDlaLinia, 355	com.mklachl.sopkom.model.entity.Przystanek, 341
setPrzystanki	setTyp
com.mklachl.sopkom.model.dto.kurs.InputKursDto, 151	com.mklachl.sopkom.model.dto.IncydentDto, 125
com.mklachl.sopkom.model.dto.kurs.KursDto, 195	com.mklachl.sopkom.model.entity.Incydent, 113
com.mklachl.sopkom.model.dto.kurs.OutOfRange, 199	setTyp_autobusu
com.mklachl.sopkom.model.dto.linia.LiniaDtoInput, 230	com.mklachl.sopkom.model.dto.kurs.AbstractKursDto, 207
com.mklachl.sopkom.model.dto.linia.LiniaDtoOutput, 233	setTypAutobusu
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput, 286	com.mklachl.sopkom.model.entity.Kurs, 187
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate, 292	setUlica
setPrzystankiWlini	com.mklachl.sopkom.model.dto.PrzystanekDto, 352
com.mklachl.sopkom.model.entity.Linia, 221	com.mklachl.sopkom.model.entity.Kierowca, 158
com.mklachl.sopkom.model.entity.Przystanek, 341	setUserDto
setRealnaGodzinna	com.mklachl.sopkom.model.dto.KierowcaDto, 169
com.mklachl.sopkom.model.dto.PrzejazdKursPrzystanekWlini, 300	setWto
com.mklachl.sopkom.model.entity.PrzejazdKursPrzystanekWlini, 295	com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto, 90
setRole	SimpleMessage
com.mklachl.sopkom.model.dto.UserDto, 413	com.mklachl.sopkom.controller.SimpleMessage, 401
setRoles	szerokoscGeograficzna
com.mklachl.sopkom.model.dto.UserDto, 413	com.mklachl.sopkom.model.dto.PrzystanekDtoFull, 365
com.mklachl.sopkom.model.entity.User, 407	template
setRozklad	com.mklachl.sopkom.controller.AutobusController, 37
com.mklachl.sopkom.model.dto.rozklad.RozkladDto, 397	com.mklachl.sopkom.controller.KierowcaController, 163
setSob	com.mklachl.sopkom.controller.LiniaController, 224
com.mklachl.sopkom.model.dto.harmonogram.HarmonogramDto, 90	com.mklachl.sopkom.controller.PrzystanekController, 345
com.mklachl.sopkom.model.entity.Harmonogram, 80	
setSpalanie	
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoOutput, 345	

templateInput		com.mklachl.sopkom.services.impl.UserServiceImpl, 417
com.mklachl.sopkom.controller.LiniaController, 225		
test		X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sop
com.mklachl.sopkom.controller.AutobusController, 37		421
com.mklachl.sopkom.controller.RozkladController, 393		X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sop 423
toString		X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sop 424, 425
com.mklachl.sopkom.model.dto.AutobusDto, 43		X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sop 276
com.mklachl.sopkom.model.dto.KierowcaDto, 169		X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sop 292
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoInput, 276		X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sop 425
com.mklachl.sopkom.model.dto.przejazd.PrzejazdDtoUpdate,		X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sop 426
typ_autobusu		X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sop 428, 429
ulica		X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sop 365
updateAutobus		X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sop 37
updateIncydent		X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sop 118
com.mklachl.sopkom.controller.IncydentController, 150		X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sop 435
com.mklachl.sopkom.services.impl.IncydentServiceImpl,		X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sop 438
com.mklachl.sopkom.services.IncydentService, 141		X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sop 438, 439
updateKierowca		X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sop 164
updateKurs		X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sop 442
com.mklachl.sopkom.controller.KursController, 190		X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sop 218
com.mklachl.sopkom.services.impl.KursServiceImpl,		X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sop 444
com.mklachl.sopkom.services.KursService, 213		X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sop 445
updateLinia		X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sop 225
com.mklachl.sopkom.controller.LiniaController, 336		X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sop 446
com.mklachl.sopkom.services.impl.PrzejazdServiceImpl, 324		X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sop 447
updatePrzystanek		X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sop 345
User		X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sop 448
com.mklachl.sopkom.model.entity.User, 403, 404		
UserDto		X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sop 449
userRepository		X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sop 70
com.mklachl.sopkom.security.CommandLineRunnerTest,		X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sop
UserServiceImpl		

2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/model/entity/Kierowca.java,
496 X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-
X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/model/entity/Kurs.java,
2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/model/entity/KursPrzystanekWlini.java,
498 X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-
X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/model/entity/Linia.java,
2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/model/entity/Przejazd.java,
499 X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-
X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/model/entity/PrzejazdBilet.java,
2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/model/entity/PrzejazdKursPrzystanekWlini.java,
500 X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-
X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/model/entity/Przystanek.java,
2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/model/entity/Role.java,
501 X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-
X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/model/entity/User.java,
2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/repository/RaportyRozkladDateHelper.java,
502 X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-
X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/repository/AutobusRepository.java,
2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/repository/BiletRepository.java,
503, 504 X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-
X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/repository/HarmonogramRepository.java,
2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/repository/IcydentRepository.java,
505, 506 X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-
X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/repository/KierowcaRepository.java,
2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/repository/KursPrzystanekWliniRepository.java,
507 X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-
X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/repository/KursRepository.java,
2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/repository/LiniaRepository.java,
508, 509 X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-
X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/repository/PrzejazdRepository.java,
2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/repository/PrzystanekWliniRepository.java,
510 X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-
X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/repository/RoleRepository.java,
2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/repository/UserRepository.java,
511, 512 X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-
X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/repository/PrzejazdBiletRepository.java,
2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/repository/PrzystanekWliniRepository.java,
513 X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-
X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/repository/PrzejazdKursPrzystanekWliniRepository.java,
2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/repository/PrzejazdRepository.java,
514, 515 X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-
X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/repository/PrzejazdKursRepository.java,
2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/repository/PrzejazdLiniaRepository.java,
516 X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-
X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty- 2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/repository/PrzejazdLiniaRepository.java,

X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-
2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/services/IcydentService.java,
[546, 547](#)

X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-
2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/services/KierowcaService.java,
[547, 548](#)

X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-
2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/services/KursService.java,
[548](#)

X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-
2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/services/LiniaService.java,
[549](#)

X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-
2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/services/PrzejazdService.java,
[550](#)

X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-
2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/services/PrzystanekService.java,
[551](#)

X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-
2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/services/UserService.java,
[552](#)

X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-
2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/SOPKoMApplication.java,
[552, 553](#)

X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-
2-comments/SOPKoM_Backend/src/main/java/com/mklachl/sopkom/topdf/Pdf.java,
[553](#)

X:/java/TAB/Doxygen/2024_TAB_S9_LACH-incydenty-
2-comments/SOPKoM_Backend/src/test/java/com/mklachl/sopkom/SOPKoMApplicationTests.java,
[558](#)