

NAZWA	OPIS	TYP	ARGUMENTY
addFlight	Dodaje do bazy lub uaktualnia istniejący już lot <i>Powodzenie – 201</i> <i>Niepowodzenie - 400</i>	POST (JSON)	departCode – <i>char[3]</i> - trzycyfrowy kod lotniska z którego wylatujemy destCode – <i>char[3]</i> - trzycyfrowy kod lotniska docelowego airlineName – <i>string</i> – nazwa linii lotniczej, wielkość znaków nie ma znaczenia, serwer wyszukuje po pierwszym członie nazwy departTime – <i>timestamp</i> – data i godzina wylotu <u>Przykładowy format -> "departTime": "2001-09-29 3:30"</u> , arTime – <i>timestamp</i> – data i godzina przylotu, format taki sam jak departTime flightTime – <i>time</i> – długość lotu price – <i>int</i> – cena lotu WSZYSTKIE POWYŻSZE ARGUMENTY OBOWIĄZKOWE <i>Przykładowe zapytanie -> documentation/server/examples/addFlightExample.png</i>
getListOfFlights	Zwraca wszystkie loty znajdujące się w bazie	GET	-----
getFlights	Zwraca wszystkie loty znajdujące się w bazie, zgodne z parametrami wyszukiwania	POST	depCode – <i>char[3]</i> - trzycyfrowy KOD lotniska, z którego wylatujemy destCode – <i>char[3]</i> - trzycyfrowy KOD lotniska docelowego startTime – <i>date</i> – data, od której poszukujemy lotów <u>Przykładowy format -> 2001-01-01</u> maxChanges – <i>int</i> z zakresu od 0 do 3– maksymalna liczba przesiadek WSZYSTKIE POWYŻSZE ARGUMENTY OBOWIĄZKOWE Parametry opcjonalne: orderBy – sortuje po: <ol style="list-style-type: none"> 1- Dacie i godzinie odlotu 2- Dacie i godzinie przylotu 3- Cenie 4- Długości całej podróży Możliwe składanie kilku pól do sortowania np. <i>orderBy="1,3,4"</i> <i>Domyślnie sortuje po dacie i godzinie odlotu</i> maxPrice – <i>int</i> – filtruje wyniki i przekazuje tylko te o nie większej cenie niż podana maxTime – <i>time</i> – maksymalny czas podróży <u>Przykładowy format -> 7:22</u> <i>Przykładowe zapytanie -> documentation/server/examples/getFlightsExample.png</i>
getAirlines	Zwraca wszystkie linie lotnicze zgodne z parametrem wyszukiwania	GET	name - <i>string</i> – ciąg znaków, od których mają zaczynać się wyszukiwane linie lotnicze, wielkość wpisanych znaków jest ignorowana - ARGUMENT OBOWIĄZKOWY

getListOfAirlines	Zwraca wszystkie linie lotnicze znajdujące się w bazie	GET	-----
getAirport	Zwraca wszystkie lotniska znajdujące się w bazie, zgodne z parametrem wyszukiwania	GET	name – <i>string</i> – ciąg znaków, od których mają zaczynać się wyszukiwane lotniska, wielkość wpisanych znaków jest ignorowana -- ARGUMENT OBOWIĄZKOWY limit – PARAMETR <i>OPCJONALNY</i> – <i>int</i> - ograniczenie ile wyników ma zwracać serwer sort – <i>OPCJONALNY</i> – <i>true/false</i> - czy zwracane wyniki mają być posortowane? (domyślnie: false)
getListOfAirports	Zwraca wszystkie lotniska znajdujące się w bazie	GET	-----
getListOfClasses	Zwraca wszystkie klasy znajdujące się w bazie	GET	-----
addClass	Metoda dodająca nową klasę do bazy <i>Powodzenie – 201</i> <i>Niepowodzenie - 400</i>	POST	name – <i>string</i> class – <i>string</i>
deleteClass	Metoda usuwająca wybraną klasę z bazy <i>Powodzenie – 201</i> <i>Niepowodzenie - 400</i>	DELETE	name – <i>string</i> class – <i>string</i> Aby dana klasa/klasy zostały usunięte muszą pokrywać się obie kolumny!!!
updateClass	Metoda edytująca wybraną klasę z bazy. Wyszukuje wiersza z pasującą nazwą i podmienia kolumnę Class <i>Powodzenie – 201</i> <i>Niepowodzenie - 400</i>	PUT	name – <i>string</i> newClass – <i>string</i> Aby dana klasa/klasy zostały zmodyfikowane musi się pokrywać nazwa klasy!!!
advancedSearch	Metoda umożliwiająca zaawansowane wyszukiwanie, aby użytkownik nie miał dostępu do innych metod niż Select metoda przyjmuje tylko argumenty występujące po frazie Select. JSON zwraca dokładnie taki sam jak w przypadku	POST	queryString – <i>string</i> – ciąg SQLowego zapytania z wyłączeniem frazy Select tzn. NIE PRZESYŁAMY queryString = "Select * from Flights" PRZESYŁAMY queryString = "* from Flights"

	getFlights		
executeQuery	Metoda obsługująca wyszukiwanie zaawansowane	POST	Executequery – <i>string</i>