Pagination, Sorting by some parameter, condition is applicable to each entities below I haven’t include them to each entity reduce document size

All Dates preferably should be in UTC

**Stewardesses: +**

GET: /api/stewardesses

Description: Returns all stewardesses

GET: /api/stewardesses?per\_page=5&page=4

Description: Returns 5 stewardesses from 15th to 20th

GET: /api/stewardesses/:id

Description: Returns stewardess by (:id)

GET: /api/stewardesses/:id/crews

Description: Returns crews in which there are stewardess with (:id)

GET: /api/stewardesses/:id/departures

Description: Returns departures of stewardess with (:id)

GET: /api/stewardesses/:id/flights

Description: Returns flights of stewardess with (:id)

GET: /api/stewardesses?name=lisa

Description: Returns all stewardesses that have ‘lisa’ substring in their first or last name

GET: /api/stewardesses?sort\_by=name&order\_by=asc

Description: Returns all stewardesses sorted by name ascending

GET: /api/stewardesses?sort\_by=dateOfBirth&order\_by=desc

Description: Returns all stewardesses sorted by name descending

GET: /api/stewardesses?dateOfBirth:lte:dateTime

:dateTime = 1993-01-20T00:00Z

Description: Returns stewardesses that were born after 1993-01-20

POST: /api/stewardesses

BODY:

{

"firstName": "Lisa", (string)

"lastname": "Osters", (string)

"dateOfBirdth": "1985-04-23T18:25Z" (DateTime)

}

PUT: /api/stewardesses/:id

BODY:

{

"firstName": "Lisa", (string)

"lastname": "Master", (string)

"dateOfBirdth": "1985-05-23T18:25Z" (DateTime)

}

Description: Fully Updates Stewardess by (:id),

PATCH: /api/stewardesses/:id

BODY:

{

"firstName": "Lexy" (string)

}

Description: Partially Updates Stewardess by (:id),

DELETE: /api/stewardesses/:id

Description: Deletes stewardess by (:id)

**Pilots: +**

GET: /api/pilots

GET: /api/pilots/:id

GET: /api/pilots?per\_page=5&page=4

GET: /api/pilots/:id/crews

GET: /api/pilots/:id/departures

GET: /api/pilots/:id/flights

GET: /api/pilots?name=alexey

GET: /api/pilots?sort\_by=name&order\_by=asc

GET: /api/pilots?sort\_by=dateOfBirth&order\_by=desc

GET: /api/pilots?dateOfBirth:lte:dateTime

:dateTime = 1990-01-20T00:00:00

GET: /api/pilots?experience:gte:timespan

:timespan = 1068:00:00

Description: Returns all pilots that have more than 3 years (1068 days) of experience

POST: /api/pilots

BODY:

{

"firstName ": "Boris", (string)

"lastName": "Albas", (string)

"dateOfBirth": "1985-04-23T18:25Z", (DateTime)

"experience": "1068:00:00" (TimeSpan)

}

Description: Creates Pilot. 1068 days of ‘Experience’ (approx 3 years), there are possibles breaks in Pilot’s career when he doesn't have flights practice

PUT: /api/pilots/:id

BODY:

{

"firstName": "Johan", (string)

"lastName": "Pogar", (string)

"dateOfBirth": "1984-05-23T18:25Z", (DateTime)

"experience": "1069:00:00" (TimeSpan)

}

PATCH: /api/pilots/:id

BODY:

{

"firstName": "Peter" (string)

}

DELETE: /api/pilots/:id

**Crews: +**

GET: /api/crews

GET: /api/crews/:id

GET: /api/crews/:id/pilot

GET: /api/crews/:id/stewardesses

GET: /api/crews?sort\_by=stewardesses.count&order\_by=desc

Description: Returns all Crews sorted by number of stewardesses in them, descending

POST: /api/crews

BODY:

{

"pilotId": 12, (int)

"stewardessesIds": [2, 4, 14] - ([] int)

}

(1 or more elements in the array)

PUT: /api/crews/:id

BODY:

{

"pilotId": 4, (int)

"stewardessesIds": [2, 4, 14, 23] - ([] int)

}

(1 or more elements in the array)

Description: Fully Updates crew by (:id). To add a single stewardesses into a crew we necessary have to include in array all previous stewardesses into request, because if we don’t do that we would have single new stewardess in the crew.

PATCH: /api/crews/:id

BODY:

{

"pilotId": 4 typeof (int)

}

Description: Partially Updates crew by (:id). If we just want change the pilot of the crew

DELETE: /api/crews/:id

**PlaneTypes: +**

GET: /api/planeTypes

GET: /api/planeTypes/:id

GET: /api/planeTypes?planeModel=:modelName

:modelName = Business

GET: /api/planeTypes?maxNumberOfPlaces:gte:numOfPlaces

:numOfPlaces = 340

Description: Returns all plane types that have maximum Number of Places more than 340

GET: /api/planeTypes?sort\_by=maxCarryingCapacityKg&order\_by=asc

Description: Returns all plane types sorted my maximum carrying capacity in Kilograms (ascending)

POST: /api/planeTypes

BODY:

{

"planeModel": "Business Jet", (string)

"maxNumberOfPlaces": 330,(int),

"maxCarryingCapacityKg": 80 000 (int)

}

PUT: /api/planeTypes/:id

BODY:

{

"planeModel": "Business Jets", typeof (string)

"maxNumberOfPlaces": 331, typeof(int),

"maxCarryingCapacityKg": 80 080 typeof(int)

}

PATCH: /api/planeTypes/:id

BODY:

{

"maxCarryingCapacityKg": 80 180 typeof(int)

}

DELETE: /api/planeTypes/:id

**Planes: +**

GET: /api/planes

GET: /api/planes/:id

GET: /api/planes?planeTypeId=:typeId

:typeId = 2

GET: /api/planes?sort\_by=name&order\_by=asc

GET: /api/planes?sort\_by=creationDate&order\_by=desc

GET: /api/planes?lifeTime:gte:time

:lifeTime = "8950:00:00"

POST: /api/planes

BODY:

{

"name": "Boeing 737-300", (string)

"planeTypeId": 1, (int)

"creationDate": “1985-05-23T18:25Z”, (DateTime)

"lifeTime": "10950:00:00" (TimeSpan)

}

Hint: 10950 days - approx 30 years

PUT: /api/planes/:id

BODY:

{

"name": " Boeing 737-400 ", (string)

"planeTypeId": 2, (int)

"creationDate": “1985-05-23T18:25Z”, (DateTime)

"lifeTime": "11000:00:00" (TimeSpan)

}

PATCH: /api/planes/:id

BODY:

{

"name": "A380", typeof (string)

}

DELETE: /api/planes/:id

**Departures: +**

GET: /api/departures

GET: /api/departures/:id

GET: /api/departures/:id/flight

GET: /api/departures/:id/crew

GET: /api/departures/:id/plane

GET: /api/planes?departureTime:gte:time

:time = “2018-05-23T18:25Z”

POST: /api/departures

BODY:

{

"flightNumber": “AS78P5”, (GUID)

"departureTime": “2018-08-23T18:25Z”, (DateTime)

"CrewId": 4, (int)

"PlaneId": 2, (int)

}

PUT: /api/departures/:id

BODY:

{

"flightNumber": “AS78P5”, (GUID)

"departureTime": “2018-08-23T19:25Z”, (DateTime C#; Date JS)

"CrewId": 71, typeof(int)

"PlaneId": 1002, typeof(int)

}

PATCH: /api/departures/:id

BODY:

{

"departureTime": “2018-08-23T23:25Z”, (DateTime)

}

DELETE: /api/departure/:id

**Flights: +**

GET: /api/flights

GET: /api/flights/:number

GET: /api/flights?from=:departureFrom&to=:destination

:departureFrom = “Ukraine, Kyiv”

:destination = “Ukraine, Lviv”

POST: /api/flights

BODY:

{

"pointOfDeparture": “Ukraine, Lviv”, (string)

"departureTime": “2018-08-25T18:25Z”, (DateTime)

"destination": “USA, Las Vegas”, (string)

"arrivalTime": “2018-08-26T19:20Z”, (DateTime)

}

PUT: /api/flights/:number

BODY:

{

"pointOfDeparture": “Ukraine, Lviv”, (string)

"departureTime": “2018-08-25T18:25Z”, (DateTime)

"destination": “USA, New York”, (string)

"arrivalTime": “2018-08-26T10:20Z”, (DateTime)

}

PATCH: /api/flights/:number

BODY:

{

"arrivalTime": “2018-08-27T10:20Z”, (DateTime)

}

DELETE: /api/flights/:number

**Tickets: +**

GET: /api/tickets

GET: /api/tickets/:id

GET: /api/tickets/:id/flight

GET: /api/tickets?sort\_by=price&order\_by=desc

POST: /api/tickets

BODY:

{

"price": 49.99, (decimal)

"flightNumber": “AS78P5”, (GUID)

}

Hint: Money in $$$ by default

PUT: /api/tickets/:id

BODY:

{

"price": 45.99 , (decimal)

"flightNumber": “VB356P1”, (GUID)

}

PATCH: /api/tickets/:id

BODY:

{

"price": 100.00 , (decimal)

}

DELETE: /api/tickets/:id