

## Flightnode - Spring Boot REST API

### Cities

Endpoint	Method	Description
/api/cities	GET	<p>Get all cities</p> <p>Parameters (optional)</p> <p><b>show-airports</b> (bool), default=false</p> <p>Response codes: 200</p>
/api/city/{id}	GET	<p>Get city by ID</p> <p>Parameters (optional)</p> <p><b>show-airports</b> (bool), default=false</p> <p>Response codes: 200, 404 (not found)</p>
/api/city-search	GET	<p>Search cities</p> <p>Parameters (optional)</p> <p><b>name</b> (string), <b>substring match</b>, case-insensitive</p> <p><b>show-airports</b> (bool), default=false</p> <p>Response codes: 200</p>
/api/city	POST	<p>Create city (constraint: <b>name+state</b> must be unique)</p> <p>Example payload</p> <pre>{ "name": "Atlanta", "state": "GA", "population": 496461 }</pre> <p>Response codes: 201, 400 (bad input), 409 (duplicate)</p>
/api/cities	POST	<p>Create multiple cities (constraint: unique <b>name+state</b>)</p> <p>Example payload</p> <pre>[   { "name": "New York", "state": "NY", "population": 8804190 },   { "name": "Los Angeles", "state": "CA", "population": 3898747 } ]</pre> <p>Response codes: 201, 400 (bad input), 409 (duplicate)</p>
/api/city/{id}	PUT	<p>Update city by ID</p> <p>Example payload</p> <pre>{ "name": "Atlanta", "state": "GA", "population": 496461 }</pre> <p>Response codes: 200, 400, 404, 409 (duplicate)</p>
/api/city/{id}	DELETE	<p>Delete city by ID</p> <p>Response codes: 204, 404, 409 (has airports)</p>

## Airports

Endpoint	Method	Description
/api/airports	GET	Get all airports Response codes: 200
/api/airport/{id}	GET	Get airport by ID Response codes: 200, 404 (not found)
/api/airport-search	GET	Search airports Parameters (optional) <b>name</b> (string), substring match, case-insensitive <b>code</b> (string), substring match, case-insensitive Response codes: 200
/api/airport	POST	Create airport (constraint: <b>code</b> must be unique) Example payload <pre>{ "name": "Long Beach Airport", "code": "LGB", "city": { "id": 2 } }</pre> Response codes: 201, 400, 404 (city not found), 409 (code conflict)
/api/airports	POST	Create multiple airports (constraint: <b>code</b> must be unique) Example payload <pre>[ { "name": "Long Beach Airport", "code": "LGB", "city": { "id": 2 } }, { "name": "LaGuardia Airport", "code": "LGA", "city": { "id": 1 } } ]</pre> Response codes: 201, 400, 404 (city not found), 409 (code conflict)
/api/airport/{id}	PUT	Update airport by ID Example payload <pre>{ "name": "Long Beach Airport", "code": "LGB", "city": { "id": 2 } }</pre> Response codes: 200, 400, 404, 409
/api/airport/{id}	DELETE	Delete airport by ID Response codes: 204, 404, 409 (has flights attached)

### Example response

GET /api/airport/1	<pre>{  "id": 1,  "name": "John F. Kennedy International Airport",  "code": "JFK",  "city": {    "id": 1,    "name": "New York",    "state": "NY"  } }</pre>
--------------------	--

## Aircrafts

Endpoint	Method	Description
/api/aircraft	GET	Get all aircrafts Parameters (optional) <b>show-airports</b> (bool), default=false Response codes: 200
/api/aircraft/{id}	GET	Get aircraft by ID Parameters (optional) <b>show-airports</b> (bool), default=false Response codes: 200, 404 (not found)
/api/aircraft	POST	Create aircraft Example payload <pre>{ "type": "Airbus A320", "airlineName": "United", "capacity": 150 }</pre> Response codes: 201, 400 (bad input)
/api/aircrafts	POST	Create multiple aircrafts Example payload <pre>[ { "type": "Airbus A320", "airlineName": "United", "capacity": 150 }, { "type": "Boeing 737-900ER", "airlineName": "Delta", "capacity": 180 } ]</pre> Response codes: 201, 400 (bad input)
/api/aircraft/{id}	PUT	Update aircraft by ID Example payload <pre>{ "type": "Airbus A320", "airlineName": "United", "capacity": 150 }</pre> Response codes: 200, 400, 404, 409
/api/aircraft/{id}	DELETE	Delete aircraft by ID Response codes: 204, 404, 409 (has flights attached)

### Example response

GET /api/aircraft/1?show-airports=true

```
{
  "id": 1,
  "airlineName": "United Airlines",
  "capacity": 166,
  "type": "Boeing 737-800",
  "departureAirports": [
    {
      "id": 7,
      "name": "O'Hare International Airport",
      "code": "ORD"
    }
  ],
  "arrivalAirports": [
    {
      "id": 4,
      "name": "Los Angeles International Airport",
      "code": "LAX"
    }
  ]
}
```

## Flights

Endpoint	Method	Description
/api/flights	GET	Get all flights Response codes: 200
/api/flight/{id}	GET	Get flight by ID Parameters (optional) <b>show-passengers</b> (bool), default=false Response codes: 200, 404 (not found)
/api/flight-search	GET	Search flights Parameters (required) <b>starting-date</b> (yyyy-mm-dd) <b>departure-city</b> (string), substring match, case-insensitive <b>arrival-city</b> (string), substring match, case-insensitive Response codes: 200, 400
/api/flight	POST	Create flight Example payload <pre>{  "flightNumber": "DL2020",  "departureTime": "2024-10-30T15:50:00",  "arrivalTime": "2024-10-30T19:10:00",  "aircraft": { "id": 7 },  "departureAirport": { "id": 11 },  "arrivalAirport": { "id": 4 }}</pre> Response codes: 201, 400, 404, 409
/api/flights	POST	Create multiple flights Response codes: 201, 400, 404, 409
/api/aircraft/{id}	PUT	Update flight by ID Response codes: 200, 400, 404, 409 (has bookings)
/api/aircraft/{id}	DELETE	Delete flight by ID Response codes: 204, 404, 409 (has bookings)

### Example response

GET /api/flight/1	<pre>{  "id": 1,  "flightNumber": "AA101",  "departureTime": "2024-01-15T06:15:00",  "arrivalTime": "2024-01-15T09:30:00",  "aircraft": {    "id": 0,    "type": "Boeing 737-800",    "airlineName": "American Airlines",    "capacity": 160  },  "departureAirport": {    "id": 1,    "name": "John F. Kennedy International Airport",    "code": "JFK"  },  "arrivalAirport": {    "id": 12,    "name": "Seattle-Tacoma International Airport",    "code": "SEA"  }}</pre>
-------------------	--

## Passengers

Endpoint	Method	Description
/api/passengers	GET	Get all passengers Parameters (optional, any combination) <b>show-bookings</b> (bool), default=false <b>show-airports</b> (bool), default=false <b>show-aircrafts</b> (bool), default=false Response codes: 200
/api/passenger/{id}	GET	Get passenger by ID Parameters (optional, any combination) <b>show-bookings</b> (bool), default=false <b>show-airports</b> (bool), default=false <b>show-aircrafts</b> (bool), default=false Response codes: 200, 404 (not found)
/api/passenger-search	GET	Search passengers Parameters (optional, any combination) <b>first-name</b> (string), substring match, case-insensitive <b>last-name</b> (string), substring match, case-insensitive Response codes: 200
/api/passenger	POST	Create passenger Example payload <pre>{  "firstName": "Mia", "lastName": "Wright", "phoneNumber": "5034447890"}</pre> Response codes: 201, 400
/api/passengers	POST	Create multiple passengers Response codes: 201, 400
/api/passenger/{id}	PUT	Update passenger by ID Response codes: 200, 400, 404
/api/passenger/{id}	DELETE	Delete passenger by ID Response codes: 204, 404, 409 (has bookings)

### Example responses

GET /api/pasenger/9	<pre>{  "id": 9,  "firstName": "Isabella",  "lastName": "Brown",  "phoneNumber": "4801234567"}</pre>
---------------------	--

GET /api/pasenger/9?  
show-aircrafts=true

```
{
  "id": 9,
  "firstName": "Isabella",
  "lastName": "Brown",
  "phoneNumber": "4801234567",
  "aircrafts": [
    {
      "id": 4,
      "type": "Airbus A320",
      "airlineName": "United Airlines",
      "capacity": 150
    },
    {
      "id": 9,
      "type": "Boeing 737-800",
      "airlineName": "American Airlines",
      "capacity": 160
    },
    {
      "id": 11,
      "type": "Airbus A321neo",
      "airlineName": "American Airlines",
      "capacity": 196
    }
  ]
}
```

GET /api/pasenger/9?  
show-airports=true

```
{
  "id": 9,
  "firstName": "Isabella",
  "lastName": "Brown",
  "phoneNumber": "4801234567",
  "airports": [
    {
      "id": 1,
      "name": "John F. Kennedy International Airport",
      "code": "JFK"
    },
    {
      "id": 10,
      "name": "Dallas Love Field",
      "code": "DAL"
    },
    {
      "id": 11,
      "name": "Denver International Airport",
      "code": "DEN"
    },
    {
      "id": 16,
      "name": "Phoenix Sky Harbor International Airport",
      "code": "PHX"
    }
  ]
}
```

## Bookings

Endpoint	Method	Description
/api/book	POST	Create booking (register passenger for flight) Example payload <pre>{ "flightId": "18", "passengerId": "15" }</pre> Response codes: 200, 400, 404, 409
/api/bookings	POST	Create multiple bookings Example payload <pre>[   { "flightId": "1", "passengerId": "1" },   { "flightId": "10", "passengerId": "8" } ]</pre> Response codes: 200, 207(multi-status), 400, 404, 409
/api/book	DELETE	Delete booking Example payload <pre>{ "flightId": "18", "passengerId": "15" }</pre> Response codes: 204, 400, 404, 409

### Example payload and response

<p>POST /api/book</p> <p>Payload</p> <pre>{   "flightId": "11",   "passengerId": "5" }</pre>	<pre>{   "flight": {     "id": 11,     "flightNumber": "DL1111",     "departureTime": "2024-06-07T06:50:00",     "arrivalTime": "2024-06-07T10:10:00",     "departureAirportCode": "MIA",     "arrivalAirportCode": "ATL",     "aircraftType": "Airbus A350-900",     "airlineName": "Delta Air Lines"   },   "message": "Flight booked.",   "passenger": {     "id": 5,     "firstName": "Olivia",     "lastName": "Patel"   } }</pre>
--	---

