

AM1 - Travel Agency

Overview

Solution of homework for NI-AM1 subject.

Version information

Version : 1.0.0

Contact information

Contact Email : chvostom@fit.cvut.cz

URI scheme

Host : t-a.org

BasePath : /

Schemes : HTTP

Tags

- country : All services about countries
- customer : All services about customers
- location : All services about locations
- tour : All services about tours

Security

travelagency_auth

Type : oauth2

Flow : implicit

Token URL : <http://travelagency.swagger.io/oauth/dialog>

Name	Description
write:customers	modify customers in your account
read:customers	read your customers

Name	Description
write:tours	modify tour in your account
read:tours	read your tours
write:locations	modify locations in your account
read:locations	read your locations
write:countries	modify countries in your account
read:countries	read your countries

Paths

Add a new country to travel agency database

POST /country

Parameters

Type	Name	Description	Schema
Body	body <i>required</i>	Country object that needs to be added to the travel agency database	Country

Responses

HTTP Code	Description	Schema
201	Country created	No Content
400	Bad request	No Content
405	Invalid input	No Content

Consumes

- `application/json`
- `application/xml`

Produces

- `application/json`
- `application/xml`

Tags

- `country`

Security

Type	Name	Scopes
oauth2	travelagency_auth	write:countries,read:countries

Get all countries

GET /country

Responses

HTTP Code	Description	Schema
200	successful operation	< Country > array
400	Bad request	No Content

Consumes

- `application/json`
- `application/xml`

Produces

- `application/json`
- `application/xml`

Tags

- country

Security

Type	Name	Scopes
oauth2	travelagency_auth	read:countries

Update an existing country

PUT /country

Parameters

Type	Name	Description	Schema
Body	body <i>required</i>	Country object that needs to be updated in the travel agency database	Country

Responses

HTTP Code	Description	Schema
200	Country successfully updated	No Content
201	Country created	No Content
400	Invalid ID supplied	No Content
405	Validation exception	No Content

Consumes

- [application/json](#)
- [application/xml](#)

Produces

- [application/json](#)
- [application/xml](#)

Tags

- country

Security

Type	Name	Scopes
oauth2	travelagency_auth	write:countries,read:countries

Find country by ID

```
GET /country/{countryId}
```

Description

Returns a single country

Parameters

Type	Name	Description	Schema
Path	countryId <i>required</i>	ID of country to return	string

Responses

HTTP Code	Description	Schema
200	successful operation	Country
400	Invalid ID supplied	No Content
404	Country not found	No Content

Produces

- `application/json`
- `application/xml`

Tags

- country

Security

Type	Name	Scopes
oauth2	travelagency_auth	read:countries

Deletes a country

```
DELETE /country/{countryId}
```

Parameters

Type	Name	Description	Schema
Path	countryId <i>required</i>	Country id to delete	string

Responses

HTTP Code	Description	Schema
200	Country successfully deleted	No Content
400	Invalid ID supplied	No Content
404	Country not found	No Content

Produces

- `application/json`
- `application/xml`

Tags

- country

Security

Type	Name	Scopes
oauth2	travelagency_auth	write:countries,read:countries

Add a new customer to travel agency database

POST /customer

Parameters

Type	Name	Description	Schema
Body	body <i>required</i>	Customer object that needs to be added to the travel agency database	Customer

Responses

HTTP Code	Description	Schema
201	Customer successfully created	No Content
400	Bad request	No Content
405	Invalid input	No Content

Consumes

- `application/json`
- `application/xml`

Produces

- `application/json`
- `application/xml`

Tags

- customer

Security

Type	Name	Scopes
oauth2	travelagency_auth	write:customers,read:customers

Get all customers

GET /customer

Responses

HTTP Code	Description	Schema
200	successful operation	< Customer > array
400	Bad request	No Content

Consumes

- [application/json](#)
- [application/xml](#)

Produces

- [application/json](#)
- [application/xml](#)

Tags

- customer

Security

Type	Name	Scopes
oauth2	travelagency_auth	read:customers

Update an existing customer

PUT /customer

Parameters

Type	Name	Description	Schema
Body	body <i>required</i>	Customer object that needs to be updated in the travel agency database	Customer

Responses

HTTP Code	Description	Schema
200	Customer successfully updated	No Content
201	Customer created	No Content
400	Invalid ID supplied	No Content
405	Validation exception	No Content

Consumes

- `application/json`
- `application/xml`

Produces

- `application/json`
- `application/xml`

Tags

- customer

Security

Type	Name	Scopes
oauth2	travelagency_auth	write:customers,read:customers

Find customer by ID

```
GET /customer/{customerId}
```

Description

Returns a single customer

Parameters

Type	Name	Description	Schema
Path	customerId <i>required</i>	ID of customer to return	integer (int64)

Responses

HTTP Code	Description	Schema
200	successful operation	Customer
400	Invalid ID supplied	No Content
404	Customer not found	No Content

Produces

- `application/json`
- `application/xml`

Tags

- customer

Security

Type	Name	Scopes
oauth2	travelagency_auth	read:customers

Deletes a customer

```
DELETE /customer/{customerId}
```

Parameters

Type	Name	Description	Schema
Path	customerId <i>required</i>	Customer id to delete	integer (int64)

Responses

HTTP Code	Description	Schema
200	Customer successfully deleted	No Content
400	Invalid ID supplied	No Content
404	Customer not found	No Content

Produces

- `application/json`
- `application/xml`

Tags

- customer

Security

Type	Name	Scopes
oauth2	travelagency_auth	write:customers,read:customers

Add a new location to travel agency database

POST /location

Parameters

Type	Name	Description	Schema
Body	body <i>required</i>	Location object that needs to be added to the travel agency database	Location

Responses

HTTP Code	Description	Schema
201	Location created	No Content
400	Bad request	No Content
405	Invalid input	No Content

Consumes

- `application/json`
- `application/xml`

Produces

- `application/json`
- `application/xml`

Tags

- location

Security

Type	Name	Scopes
oauth2	travelagency_auth	write:locations,read:locations

Get all locations

GET /location

Responses

HTTP Code	Description	Schema
200	successful operation	< Location > array
400	Bad request	No Content

Consumes

- `application/json`
- `application/xml`

Produces

- `application/json`
- `application/xml`

Tags

- `location`

Security

Type	Name	Scopes
oauth2	travelagency_auth	read:locations

Update an existing location

PUT /location

Parameters

Type	Name	Description	Schema
Body	body <i>required</i>	Location object that needs to be updated in the travel agency database	Location

Responses

HTTP Code	Description	Schema
200	Location successfully updated	No Content
201	Location created	No Content
400	Invalid ID supplied	No Content
405	Validation exception	No Content

Consumes

- `application/json`
- `application/xml`

Produces

- `application/json`
- `application/xml`

Tags

- `location`

Security

Type	Name	Scopes
oauth2	travelagency_auth	write:locations,read:locations

Find location by ID

```
GET /location/{locationId}
```

Description

Returns a single location

Parameters

Type	Name	Description	Schema
Path	locationId <i>required</i>	ID of location to return	integer (int64)

Responses

HTTP Code	Description	Schema
200	successful operation	Location
400	Invalid ID supplied	No Content
404	Location not found	No Content

Produces

- `application/json`
- `application/xml`

Tags

- location

Security

Type	Name	Scopes
oauth2	travelagency_auth	read:locations

Deletes a location

```
DELETE /location/{locationId}
```

Parameters

Type	Name	Description	Schema
Path	locationId <i>required</i>	Location id to delete	integer (int64)

Responses

HTTP Code	Description	Schema
200	Location successfully deleted	No Content
400	Invalid ID supplied	No Content
404	Location not found	No Content

Produces

- `application/json`
- `application/xml`

Tags

- location

Security

Type	Name	Scopes
oauth2	travelagency_auth	write:locations,read:locations

Add a new tour to travel agency database

POST /tour

Parameters

Type	Name	Description	Schema
Body	body <i>required</i>	Tour object that needs to be added to the travel agency database	Tour

Responses

HTTP Code	Description	Schema
201	Tour created	No Content
400	Bad request	No Content
405	Invalid input	No Content

Consumes

- `application/json`
- `application/xml`

Produces

- `application/json`
- `application/xml`

Tags

- tour

Security

Type	Name	Scopes
oauth2	travelagency_auth	write:tours,read:tours

Get all tours

GET /tour

Responses

HTTP Code	Description	Schema
200	successful operation	< Tour > array
400	Bad request	No Content

Consumes

- [application/json](#)
- [application/xml](#)

Produces

- [application/json](#)
- [application/xml](#)

Tags

- tour

Security

Type	Name	Scopes
oauth2	travelagency_auth	read:tours

Update an existing tour

PUT /tour

Parameters

Type	Name	Description	Schema
Body	body <i>required</i>	Tour object that needs to be updated in the travel agency database	Tour

Responses

HTTP Code	Description	Schema
200	Tour successfully updated	No Content
201	Tour created	No Content
400	Invalid ID supplied	No Content
405	Validation exception	No Content

Consumes

- [application/json](#)
- [application/xml](#)

Produces

- [application/json](#)
- [application/xml](#)

Tags

- tour

Security

Type	Name	Scopes
oauth2	travelagency_auth	write:tours,read:tours

Finds tours by customer

GET /tour/findByCustomer

Parameters

Type	Name	Description	Schema
Query	customerId <i>required</i>	Customer value for filtering	integer (int64)

Responses

HTTP Code	Description	Schema
200	successful operation	< Tour > array
400	Invalid customer id value	No Content

Produces

- [application/json](#)
- [application/xml](#)

Tags

- [tour](#)

Security

Type	Name	Scopes
oauth2	travelagency_auth	read:tours

Finds tours by location

GET /tour/findByLocation

Parameters

Type	Name	Description	Schema
Query	locationId <i>required</i>	Location value for filtering	integer (int64)

Responses

HTTP Code	Description	Schema
200	successful operation	< Tour > array
400	Invalid location id value	No Content

Produces

- [application/json](#)
- [application/xml](#)

Tags

- [tour](#)

Security

Type	Name	Scopes
oauth2	travelagency_auth	read:tours

Find tour by ID

```
GET /tour/{tourId}
```

Description

Returns a single tour

Parameters

Type	Name	Description	Schema
Path	tourId <i>required</i>	ID of tour to return	integer (int64)

Responses

HTTP Code	Description	Schema
200	successful operation	Tour
400	Invalid ID supplied	No Content
404	Tour not found	No Content

Produces

- `application/json`
- `application/xml`

Tags

- `tour`

Security

Type	Name	Scopes
oauth2	travelagency_auth	read:tours

Deletes a tour

```
DELETE /tour/{tourId}
```

Parameters

Type	Name	Description	Schema
Path	tourId <i>required</i>	Tour id to delete	integer (int64)

Responses

HTTP Code	Description	Schema
200	Tour successfully deleted	No Content

HTTP Code	Description	Schema
400	Invalid ID supplied	No Content
404	Tour not found	No Content

Produces

- `application/json`
- `application/xml`

Tags

- `tour`

Security

Type	Name	Scopes
oauth2	travelagency_auth	write:tours,read:tours

Definitions

Country

Name	Schema
id <i>optional</i>	string
locations <i>optional</i>	< Location > array
name <i>optional</i>	string

Customer

Name	Schema
email <i>optional</i>	string

Name	Schema
firstname <i>optional</i>	string
id <i>optional</i>	integer (int64)
phone <i>optional</i>	string
surname <i>optional</i>	string

Location

Name	Schema
countryId <i>optional</i>	string
id <i>optional</i>	integer (int64)
title <i>optional</i>	string

Tour

Name	Schema
customers <i>optional</i>	< Customer > array
endDate <i>optional</i>	string (date-time)
id <i>optional</i>	integer (int64)
location <i>optional</i>	Location

Name	Schema
startDate <i>optional</i>	string (date-time)
transport <i>optional</i>	enum (car, bus, train, plane, boat)