**SIMPLE API**

swagger: '2.0'

info:

description: This is a simple API

version: 1.0.0

title: Simple Inventory API

# put the contact info for your development or API team

contact:

email: you@your-company.com

license:

name: Apache 2.0

url: http://www.apache.org/licenses/LICENSE-2.0.html

# tags are used for organizing operations

tags:

- name: admins

description: Secured Admin-only calls

- name: developers

description: Operations available to regular developers

paths:

/inventory:

get:

tags:

- developers

summary: searches inventory

operationId: searchInventory

description: |

By passing in the appropriate options, you can search for

available inventory in the system

produces:

- application/json

parameters:

- in: query

name: searchString

description: pass an optional search string for looking up inventory

required: false

type: string

- in: query

name: skip

description: number of records to skip for pagination

type: integer

format: int32

minimum: 0

- in: query

name: limit

description: maximum number of records to return

type: integer

format: int32

minimum: 0

maximum: 50

responses:

200:

description: search results matching criteria

schema:

type: array

items:

$ref: '#/definitions/InventoryItem'

400:

description: bad input parameter

post:

tags:

- admins

summary: adds an inventory item

operationId: addInventory

description: Adds an item to the system

consumes:

- application/json

produces:

- application/json

parameters:

- in: body

name: inventoryItem

description: Inventory item to add

schema:

$ref: '#/definitions/InventoryItem'

responses:

201:

description: item created

400:

description: invalid input, object invalid

409:

description: an existing item already exists

definitions:

InventoryItem:

type: object

required:

- id

- name

- manufacturer

- releaseDate

properties:

id:

type: string

format: uuid

example: d290f1ee-6c54-4b01-90e6-d701748f0851

name:

type: string

example: Widget Adapter

releaseDate:

type: string

format: date-time

example: 2016-08-29T09:12:33.001Z

manufacturer:

$ref: '#/definitions/Manufacturer'

Manufacturer:

required:

- name

properties:

name:

type: string

example: ACME Corporation

homePage:

type: string

format: url

example: https://www.acme-corp.com

phone:

type: string

example: 408-867-5309

**CUSTOMER API - MY DEMO SECURITY DEFINITIONS**

securityDefinitions:

BasicAuth:

type: basic

ApiKeyAuth:

type: apiKey

in: header

name: X-API-Key

OAuth2:

type: oauth2

flow: accessCode

authorizationUrl: https://example.com/oauth/authorize

scopes:

read: Grants read access

write: Grants write access

**CUSTOMER API FULL DEMO (SWAGGER 2.0)**

swagger: '2.0'

info:

title: Customer API

description: This is the Customer API. It provides RESTful operations for Customer data.

version: 1.0.0

contact:

name: rohini priya

url: http://www.globalmantics.com

email: ragalapalli.rohinipriya@capgemini.com

license:

name: Apache '2.0'

url: http://apache.org/licenses/LICENSE-2.0.html

paths:

/customer:

get:

summary: reads a cutomer's data

operationId: getCustomer

description: "This operation provides a view of the customers data in JSON.

the operation used the customer id to identity the customer

in a query string."

produces:

- application/json

- application/xml

- text/xml

- text/cvs

parameters:

- in: query

name: customerId

description: pass a customer id to get customer data.

required: true

type: integer

responses:

200:

description: search results mathching criteria.

schema:

$ref: "#/definitions/Customer"

404:

description: employee with this id does not exist.

post:

summary: adds a new customer.

operationId: addCustomer

description: Add a new customer to the system.

produces:

- text/plain

consumes:

- application/json

parameters:

- in: body

name: body

description: The new customer data in Json format

required: true

schema:

$ref: "#/definitions/Customer"

responses:

'200':

description: "Successful operation"

schema:

type: integer

'405':

description: "Invalid input"

/customer/{customerId}:

put:

summary: update an existing customer

operationId: updateCustomer

description: updates an existing customer with new data

consumes:

- application/json

parameters:

- in: body

name: body

description: The updated customer data in JSON formar

required: true

schema:

$ref: "#/definitions/Customer"

- in: path

name: customerId

description: The Id of the customer to update.

required: true

type: integer

responses:

'200':

description: Success.

'404':

description: Customer Not Found.

'500':

description: Internal Server Error.

delete:

summary: Delete an existing customer

operationId: deleteCustomer

description: Deletes an existing customer in this system.

produces:

- application/json

parameters:

- in: path

name: customerId

description: The Id of the customer to update.

required: true

type: integer

responses:

'200':

description: The customer was deleted.

schema:

$ref: "#/definitions/Customer"

'404':

description: Customer Not Found.

'500':

description: Internal Server Error.

security:

- BasicAuth: []

definitions:

Customer:

type: object

properties:

customerId:

type: integer

accountValue:

type: number

customerName:

type: string

active:

type: boolean

address:

type: object

properties:

name:

type: string

street:

type: string

city:

type: string

state:

type: string

zip:

type: string

contacts:

type: array

items:

type: object

properties:

name:

type: string

email:

type: string

phone:

type: string

host: api.globalmantics.com

basePath: /crm/v1

schemes:

- http

- https

securityDefinitions:

BasicAuth:

type: basic

security:

- BasicAuth: []

**CUSTOMER API FULL DEMO (OpenAPI 3.0)**

openapi: 3.0.0

info:

title: Customer API

description: This is the Customer API. It provides RESTful operations for Customer data.

version: 1.0.0-oas3

contact:

name: rohini priya

url: http://www.globalmantics.com

email: ragalapalli.rohinipriya@capgemini.com

license:

name: Apache '2.0'

url: http://apache.org/licenses/LICENSE-2.0.html

paths:

/customer:

get:

summary: reads a cutomer's data

operationId: getCustomer

description: >-

This operation provides a view of the customers data in JSON. the

operation used the customer id to identity the customer in a query

string.

parameters:

- in: query

name: customerId

description: pass a customer id to get customer data.

required: true

schema:

type: integer

responses:

'200':

description: search results mathching criteria.

content:

application/json:

schema:

$ref: '#/components/schemas/Customer'

application/xml:

schema:

$ref: '#/components/schemas/Customer'

text/xml:

schema:

$ref: '#/components/schemas/Customer'

text/cvs:

schema:

$ref: '#/components/schemas/Customer'

'404':

description: employee with this id does not exist.

post:

summary: adds a new customer.

operationId: addCustomer

description: Add a new customer to the system.

requestBody:

content:

application/json:

schema:

$ref: '#/components/schemas/Customer'

description: The new customer data in Json format

required: true

responses:

'200':

description: Successful operation

content:

text/plain:

schema:

type: integer

'405':

description: Invalid input

/customer/{customerId}:

put:

summary: update an existing customer

operationId: updateCustomer

description: updates an existing customer with new data

parameters:

- in: path

name: customerId

description: The Id of the customer to update.

required: true

schema:

type: integer

requestBody:

content:

application/json:

schema:

$ref: '#/components/schemas/Customer'

description: The updated customer data in JSON formar

required: true

responses:

'200':

description: Success.

'404':

description: Customer Not Found.

'500':

description: Internal Server Error.

delete:

summary: Delete an existing customer

operationId: deleteCustomer

description: Deletes an existing customer in this system.

parameters:

- in: path

name: customerId

description: The Id of the customer to update.

required: true

schema:

type: integer

responses:

'200':

description: The customer was deleted.

content:

application/json:

schema:

$ref: '#/components/schemas/Customer'

'404':

description: Customer Not Found.

'500':

description: Internal Server Error.

security:

- BasicAuth: []

security:

- BasicAuth: []

servers:

- url: https://api.globalmantics.com/crm/v1

description: Internal Production.

- url: https://staging.globalmantics.com/crm/v1

description: Internal Staging.

- url: https://development.globalmantics.com/crm/v1

description: Internal Development.

- url: https://{customerId}.globalmantics.com/crm/{version}

description: Customer Production

variables:

customerId:

default: demo

description: The Customer Id for subdomain

version:

default: v1

enum:

- v1

- v2

- v3-beta

components:

securitySchemes:

BasicAuth:

type: http

scheme: basic

schemas:

Customer:

type: object

properties:

customerId:

type: integer

accountValue:

type: number

customerName:

type: string

active:

type: boolean

address:

type: object

properties:

name:

type: string

street:

type: string

city:

type: string

state:

type: string

zip:

type: string

contacts:

type: array

items:

type: object

properties:

name:

type: string

email:

type: string

phone:

type: string

===========================================================================

**SWAGGER OPENAPI SWAPI DOCUMENT**

swagger: '2.0'

info:

description: Open API Document generated from SWAPI.

version: '0.1'

title: SWAPI

paths:

/api/people/:

get:

produces:

- application/json

parameters:

- name: search

in: query

required: false

type: string

x-example: r2

responses:

'200':

description: Definition generated from Swagger Inspector

schema:

$ref: '#/definitions/Model0'

/api/people:

get:

produces:

- application/json

parameters: []

responses:

'200':

description: Definition generated from Swagger Inspector

/api/people/2/:

get:

produces:

- application/json

parameters: []

responses:

'200':

description: Definition generated from Swagger Inspector

schema:

$ref: '#/definitions/Model1'

definitions:

Results:

properties:

name:

type: string

height:

type: string

mass:

type: string

hair\_color:

type: string

skin\_color:

type: string

eye\_color:

type: string

birth\_year:

type: string

gender:

type: string

homeworld:

type: string

films:

type: array

items:

type: string

species:

type: array

items:

type: string

vehicles:

type: array

starships:

type: array

created:

type: string

edited:

type: string

url:

type: string

Model0:

properties:

count:

type: integer

format: int32

next:

type: object

previous:

type: object

results:

type: array

items:

$ref: '#/definitions/Results'

Model1:

properties:

name:

type: string

height:

type: string

mass:

type: string

hair\_color:

type: string

skin\_color:

type: string

eye\_color:

type: string

birth\_year:

type: string

gender:

type: string

homeworld:

type: string

films:

type: array

items:

type: string

species:

type: array

items:

type: string

vehicles:

type: array

starships:

type: array

created:

type: string

edited:

type: string

url:

type: string

host: swapi.dev

schemes:

- http

- https

=================================================================

**SWAGGER OPENAPI SWAPI DOCUMENT(using basePath)**

swagger: '2.0'

info:

description: Open API Document generated from SWAPI.

version: '0.1'

title: SWAPI

paths:

/people/:

get:

produces:

- application/json

parameters:

- name: search

in: query

required: false

type: string

x-example: r2

responses:

'200':

description: Definition generated from Swagger Inspector

schema:

$ref: '#/definitions/Model0'

/people:

get:

produces:

- application/json

parameters: []

responses:

'200':

description: Definition generated from Swagger Inspector

/people/{characterId}:

get:

produces:

- application/json

parameters:

- in: path

name: characterId

required: true

type: integer

responses:

'200':

description: Definition generated from Swagger Inspector

schema:

$ref: '#/definitions/Model1'

definitions:

Results:

properties:

name:

type: string

height:

type: string

mass:

type: string

hair\_color:

type: string

skin\_color:

type: string

eye\_color:

type: string

birth\_year:

type: string

gender:

type: string

homeworld:

type: string

films:

type: array

items:

type: string

species:

type: array

items:

type: string

vehicles:

type: array

starships:

type: array

created:

type: string

edited:

type: string

url:

type: string

Model0:

properties:

count:

type: integer

format: int32

next:

type: object

previous:

type: object

results:

type: array

items:

$ref: '#/definitions/Results'

Model1:

properties:

name:

type: string

height:

type: string

mass:

type: string

hair\_color:

type: string

skin\_color:

type: string

eye\_color:

type: string

birth\_year:

type: string

gender:

type: string

homeworld:

type: string

films:

type: array

items:

type: string

species:

type: array

items:

type: string

vehicles:

type: array

starships:

type: array

created:

type: string

edited:

type: string

url:

type: string

host: swapi.dev

basePath: /api

schemes:

- http

- https