swagger: "2.0"

info:

description: "Ayurvedic plants website api by G12."

version: "1.0.0"

title: "E-Ayurvedic herbs"

contact:

email: "shitalshirole1111@gmail.com"

tags:

- name: "medicinalPlants"

description: "Everything about your ayurvedic plants"

- name: "store"

description: "Access to medicinal plants"

schemes:

- "https"

- "http"

paths:

/medicinalPlants:

post:

tags:

- "medicinalPlants"

summary: "Add a new plant information to the store"

description: ""

operationId: "addMedicinalPlants"

consumes:

- "application/json"

- "application/xml"

produces:

- "application/xml"

- "application/json"

parameters:

- in: "body"

name: "body"

description: "Medicinal plants object that needs to be added to the store"

required: true

schema:

$ref: "#/definitions/MedicinalPlant"

responses:

"405":

description: "Invalid input"

security:

- medicineStore\_auth:

- "write:plants"

- "read:plants"

put:

tags:

- "medicinalPlants"

summary: "Update an existing plant"

description: ""

operationId: "updatePlant"

consumes:

- "application/json"

- "application/xml"

produces:

- "application/xml"

- "application/json"

parameters:

- in: "body"

name: "body"

description: "mecinal plants object that needs to be added to the store"

required: true

schema:

$ref: "#/definitions/MedicinalPlant"

responses:

"400":

description: "Invalid ID supplied"

"404":

description: "Plant not found"

"405":

description: "Validation exception"

security:

- medicineStore\_auth:

- "write:plants"

- "read:plants"

/medicinalPlants/{plantName}:

get:

tags:

- "medicinalPlants"

summary: "Search plant by its name"

description: "Returns a single plant"

operationId: "getPlantByName"

produces:

- "application/xml"

- "application/json"

parameters:

- name: "plantName"

in: "path"

description: "name of plant to return"

required: true

type: "string"

responses:

"200":

description: "successful operation"

schema:

$ref: "#/definitions/MedicinalPlant"

"400":

description: "Invalid name supplied"

"404":

description: "Plant not found"

security:

- api\_key: []

delete:

tags:

- "medicinalPlants"

summary: "Deletes a plant"

description: ""

operationId: "deletePlant"

produces:

- "application/xml"

- "application/json"

parameters:

- name: "api\_key"

in: "header"

required: false

type: "string"

- name: "plantName"

in: "path"

description: "Plant name to delete"

required: true

type: "string"

responses:

"400":

description: "Invalid Name supplied"

"404":

description: "Plant not found"

security:

- medicineStore\_auth:

- "write:plants"

- "read:plants"

/store/inventory:

get:

tags:

- "store"

summary: "Show plants by status"

description: "Returns a map of status with respect to plants"

operationId: "getInventory"

produces:

- "application/json"

parameters: []

responses:

"200":

description: "successful operation"

schema:

type: "object"

additionalProperties:

type: "integer"

format: "int32"

security:

- api\_key: []

/store/information:

post:

tags:

- "store"

summary: "Place an information of a plant"

description: ""

operationId: "placeInformation"

produces:

- "application/xml"

- "application/json"

parameters:

- in: "body"

name: "body"

description: "request send for searching the plant"

required: true

schema:

$ref: "#/definitions/information"

responses:

"200":

description: "successful operation"

schema:

$ref: "#/definitions/information"

"400":

description: "Invalid plant"

/store/information/{plantName}:

get:

tags:

- "store"

summary: "Find medcinal plant by its name"

description: "For valid response try string names with word >= 1 and <= 10. Other values will generated exceptions"

operationId: "getPlantByitsName"

produces:

- "application/xml"

- "application/json"

parameters:

- name: "plantName"

in: "path"

description: "name of plant that needs to be fetched"

required: true

type: "string"

maximum: 10.0

minimum: 1.0

responses:

"200":

description: "successful operation"

schema:

$ref: "#/definitions/information"

"400":

description: "Invalid name supplied"

"404":

description: "Information not found"

securityDefinitions:

medicineStore\_auth:

type: "oauth2"

authorizationUrl: "http://ayurveda.com"

flow: "implicit"

scopes:

write:plants: "modify plants from your website"

read:plants: "read your medicinal plants information"

api\_key:

type: "apiKey"

name: "api\_key"

in: "header"

definitions:

information:

type: "object"

properties:

plantName:

type: "string"

place:

type: "string"

status:

type: "string"

description: "information Status"

enum:

- "need to upload"

- "uploading"

- "uploaded"

complete:

type: "string"

default: false

xml:

name: "information"

place:

type: "object"

properties:

name:

type: "string"

xml:

name: "places"

MedicinalPlant:

type: "object"

required:

- "name"

- "photoUrls"

properties:

place:

$ref: "#/definitions/place"

name:

type: "string"

example: "tulsi"

photoUrls:

type: "array"

xml:

name: "photoUrl"

wrapped: true

items:

type: "string"

tags:

type: "array"

xml:

name: "tag"

wrapped: true

items:

$ref: "#/definitions/MedicinalPlant"

status:

type: "string"

description: "plants status in the medicinal plants store"

enum:

- "available"

- "pending"

- "getting"

xml:

name: "Medicinal Plants"

externalDocs:

description: "Find out more about ayurveda"

url: http://ayurveda.com