swagger: "2.0"

info:

description: "This is a sample server Petstore server. You can find out more about Swagger at [http://swagger.io](http://swagger.io) or on [irc.freenode.net, #swagger](http://swagger.io/irc/). For this sample, you can use the api key `special-key` to test the authorization filters."

version: "1.0.0"

title: "Swagger Petstore"

termsOfService: "<http://swagger.io/terms/>"

contact:

email: "[apiteam@swagger.io](mailto:apiteam@swagger.io)"

license:

name: "Apache 2.0"

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

host: "user.swagger.io"

basePath: "/v2"

tags:

- name: "user"

description: "Operations about user"

schemes:

- "http"

paths:

/user:

post:

tags:

- "user"

summary: "Create user"

description: "This can only be done by the logged in user."

operationId: "createUser"

produces:

- "application/json"

parameters:

- in: "body"

name: "body"

description: "Created user object"

required: true

schema:

$ref: "#/definitions/User"

responses:

default:

description: "successful operation"

/allUsers:

post:

tags:

- "user"

summary: "Creates list of users with given input array"

description: ""

operationId: "createUsersWithArrayInput"

produces:

- "application/json"

parameters:

- in: "body"

name: "body"

description: "List of user object"

required: false

schema:

type: "array"

items:

$ref: "#/definitions/User"

responses:

200:

description: "successful operation"

schema:

type: "string"

headers:

X-Rate-Limit:

type: "integer"

format: "int32"

description: "ok"

X-Expires-After:

type: "string"

format: "date-time"

description: "date in UTC when token expires"

203:

description: "não-autorizado"

204:

description: "Nenhum conteúdo"

206:

description: "Conteúdo parcial"

400:

description: "Requisição inválida"

401:

description: "Não autorizado"

404:

description: "Não encontrado"

407:

description: "Autenticação de proxy necessária"

409:

description: "Conflito"

413:

description: "Entidade de solicitação muito grande"

/user/login:

get:

tags:

- "user"

summary: "Logs user into the system"

description: ""

operationId: "loginUser"

produces:

- "application/json"

parameters:

- name: "username"

in: "query"

description: "The user name for login"

required: true

type: "string"

- name: "password"

in: "query"

description: "The password for login in clear text"

required: true

type: "string"

responses:

200:

description: "successful operation"

schema:

type: "string"

headers:

X-Rate-Limit:

type: "integer"

format: "int32"

description: "calls per hour allowed by the user"

X-Expires-After:

type: "string"

format: "date-time"

description: "date in UTC when token expires"

400:

description: "Invalid username/password supplied"

/user/{username}:

get:

tags:

- "user"

summary: "Get user by user name"

description: ""

operationId: "getUserByName"

produces:

- "application/json"

parameters:

- name: "username"

in: "path"

description: "The name that needs to be fetched. Use user1 for testing. "

required: true

type: "string"

responses:

200:

description: "successful operation"

schema:

$ref: "#/definitions/User"

400:

description: "Invalid username supplied"

404:

description: "User not found"

put:

tags:

- "user"

summary: "Updated user"

description: "This can only be done by the logged in user."

operationId: "updateUser"

produces:

- "application/json"

parameters:

- name: "username"

in: "path"

description: "name that need to be updated"

required: true

type: "string"

- in: "body"

name: "body"

description: "Updated user object"

required: true

schema:

$ref: "#/definitions/User"

responses:

400:

description: "Invalid user supplied"

404:

description: "User not found"

delete:

tags:

- "user"

summary: "Delete user"

description: "This can only be done by the logged in user."

operationId: "deleteUser"

produces:

- "application/json"

parameters:

- name: "username"

in: "path"

description: "The name that needs to be deleted"

required: true

type: "string"

responses:

400:

description: "Invalid username supplied"

404:

description: "User not found"

definitions:

User:

type: "object"

properties:

id:

type: "integer"

format: "int64"

username:

type: "string"

firstName:

type: "string"

lastName:

type: "string"

email:

type: "string"

password:

type: "string"

phone:

type: "string"

userStatus:

type: "integer"

format: "int32"

description: "User Status"

externalDocs:

description: "Find out more about Swagger"

url: "<http://swagger.io>"