# Input

## Consumes

### API uses url for the parameters

consumes:

- application/x-www-form-urlencoded

### API uses as input the body

consumes:

- application/json

- application/xml

## Parameters

### Parameter from body

parameters:

- in: body

name: body

description: Pet object that needs to be added to the store

required: true

schema:

$ref: '#/definitions/Pet'

### Parameter from query parameter

parameters:

- name: tags

in: query

description: Tags to filter by

required: true

type: array

items:

type: string

collectionFormat: multi

### Parameter from path (goes with url encoded in the format; see 1.1.1)

parameters:

- name: petId

in: path

description: ID of pet to return

required: true

type: integer

format: int64

### Parameter from header

parameters:

- name: api\_key

in: header

required: false

type: string

### Paramter with basic validation

parameters:

- name: orderId

in: path

description: ID of pet that needs to be fetched

required: true

type: integer

maximum: 10

minimum: 1

format: int64

# Output

## Produces

### API places the output in the body

produces:

- application/xml

- application/json

## Response

### Response with an error code

responses:

'405':

description: Invalid input

### Response with an array

responses:

'200':

description: successful operation

schema:

type: array

items:

$ref: '#/definitions/Pet'

### Response with a custom type (model)

responses:

'200':

description: successful operation

schema:

$ref: '#/definitions/Pet'

### Response is an object

responses:

'200':

description: successful operation

schema:

type: object

additionalProperties:

type: integer

format: int32

# Security

On each path we can specify the properties of the security

securityDefinitions:

petstore\_auth:

type: oauth2

authorizationUrl: 'http://petstore.swagger.io/oauth/dialog'

flow: implicit

scopes:

'write:pets': modify pets in your account

'read:pets': read your pets

api\_key:

type: apiKey

name: api\_key

in: header

## Using OAuth2 (on a given path)

security:

- petstore\_auth:

- 'write:pets'

- 'read:pets'

## Using a shared Key (on a given path)

security:

- api\_key: []