



# Use OpenAPI Generator to share your models between Flutter and your backend

By Bernos Guillaume  
Flutter Vikings 2022



## Who am I?

- Bad medium articles in 2017
- Creator of `location` package
- Tech Lead at Bam.tech in Paris





## Overview

1. What is OpenAPI?
2. How to install OpenAPI Generator?
3. How to use OpenAPI Generator?



# 1. What is OpenAPI?

It's a specification for describing APIs

Uses:

- Backend and frontend in a different language
- Easily create mocks
- Public API and need to document it automatically
- Automate parts of your workflow

```
/pet/{petId}:
  get:
    tags:
      - pet
    summary: Find pet by ID
    description: Returns a single pet
    operationId: getPetById
    produces:
      - application/json
      - application/xml
    parameters:
      - name: petId
        in: path
        description: ID of pet to return
        required: true
        type: integer
        format: int64
    responses:
      '200':
        description: successful operation
        schema:
          $ref: '#/definitions/Pet'
      '400':
        description: Invalid ID supplied
      '404':
        description: Pet not found
    security:
      - api_key: []
```

Demo OpenApi file

```
post:
  tags:
  - pet
  summary: Updates a pet in the store with form data
  description: ''
  operationId: updatePetWithForm
  consumes:
  - application/x-www-form-urlencoded
  produces:
  - application/json
  - application/xml
  parameters:
  - name: petId
    in: path
    description: ID of pet that needs to be updated
    required: true
    type: integer
    format: int64
  - name: name
    in: formData
    description: Updated name of the pet
    required: false
    type: string
  - name: status
    in: formData
    description: Updated status of the pet
    required: false
    type: string
  responses:
    '405':
      description: Invalid input
  security:
  - petstore_auth:
    - write:pets
    - read:pets
```

Demo OpenApi file

```
Pet:
  type: object
  required:
    - name
    - photoUrls
  properties:
    id:
      type: integer
      format: int64
    category:
      $ref: '#/definitions/Category'
    name:
      type: string
      example: doggie
    photoUrls:
      type: array
      xml:
        wrapped: true
      items:
        type: string
        xml:
          name: photoUrl
    tags:
      type: array
      xml:
        wrapped: true
      items:
        xml:
          name: tag
        $ref: '#/definitions/Tag'
    status:
      type: string
      description: pet status in the store
      enum:
        - available
        - pending
        - sold
  xml:
    name: Pet
```

Demo OpenApi file



# Thanks for watching!



BeGuillaume



<https://guillaume.bernos.dev>