Документация REST API с использованием Swagger

Федотовский Павел, 14.05.2017

План

- Документация REST API – когда, что и зачем?
- Code First (Swashbuckle)
- Design First (Swagger Hub)
- Swagger
- Выводы

Документация REST API

Кому нужна?

- Разработчики
- Тестировщики
- Клиенты

Когда нужна?

- Для уже существующего API
- Создание нового API с нуля

Что такое хорошая документация для API?

- Примеры использования
- Согласованность
- Хорошая организация
- Удобство написания
- Генерация кода

Что нужно описывать?

- Информация о REST API
- Список ресурсов
- Операции над ресурсами
- Модели
- Параметры

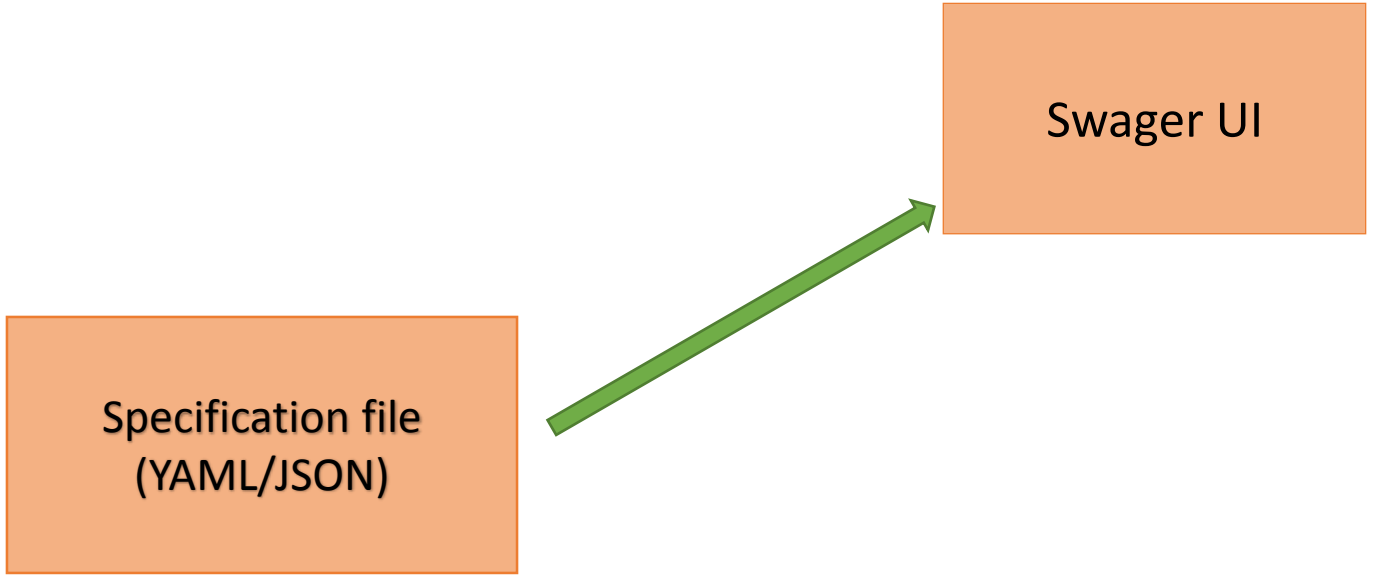
Стандарты спецификации API

apiblueprint

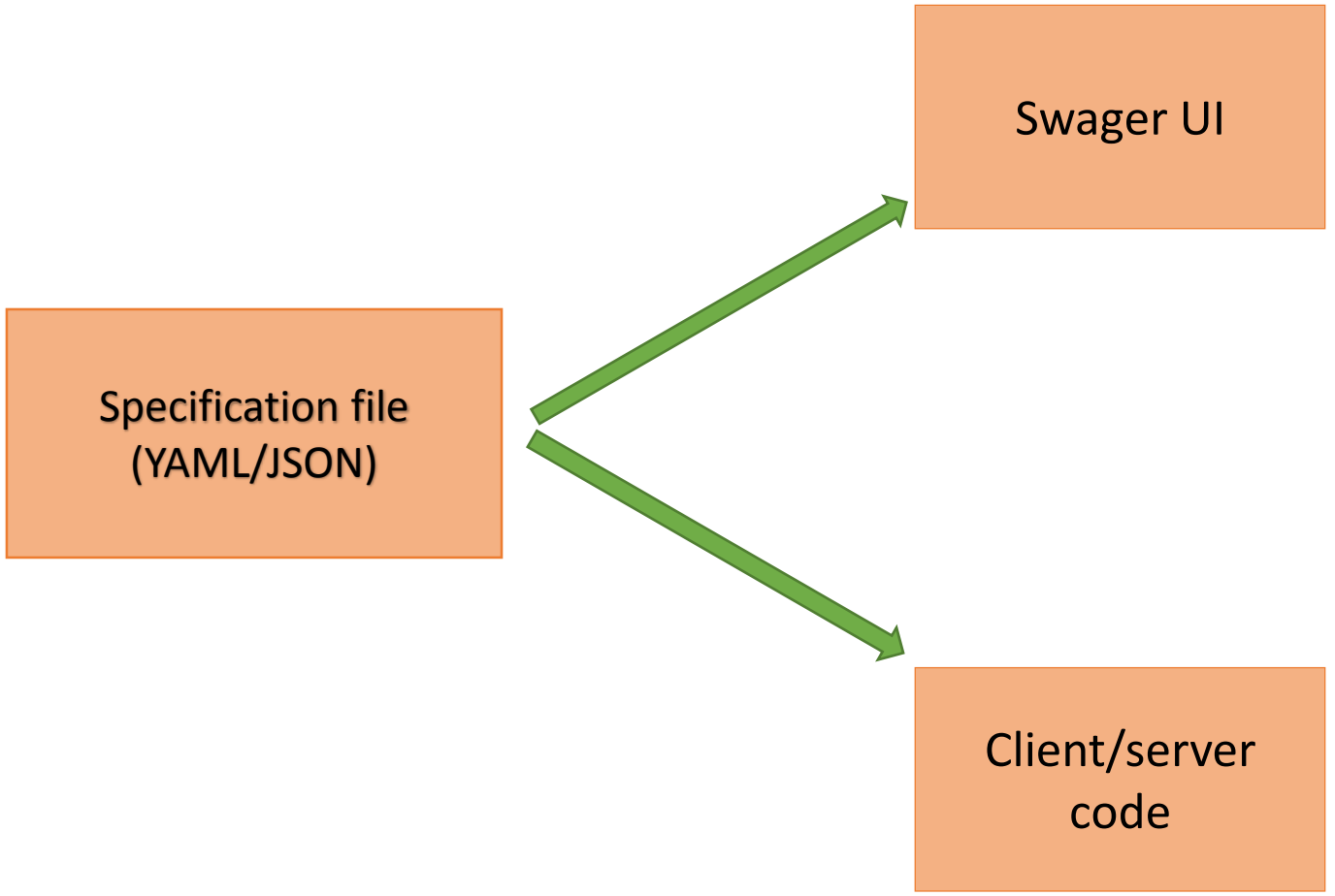
Swagger (OpenAPI)

Specification file
(YAML/JSON)

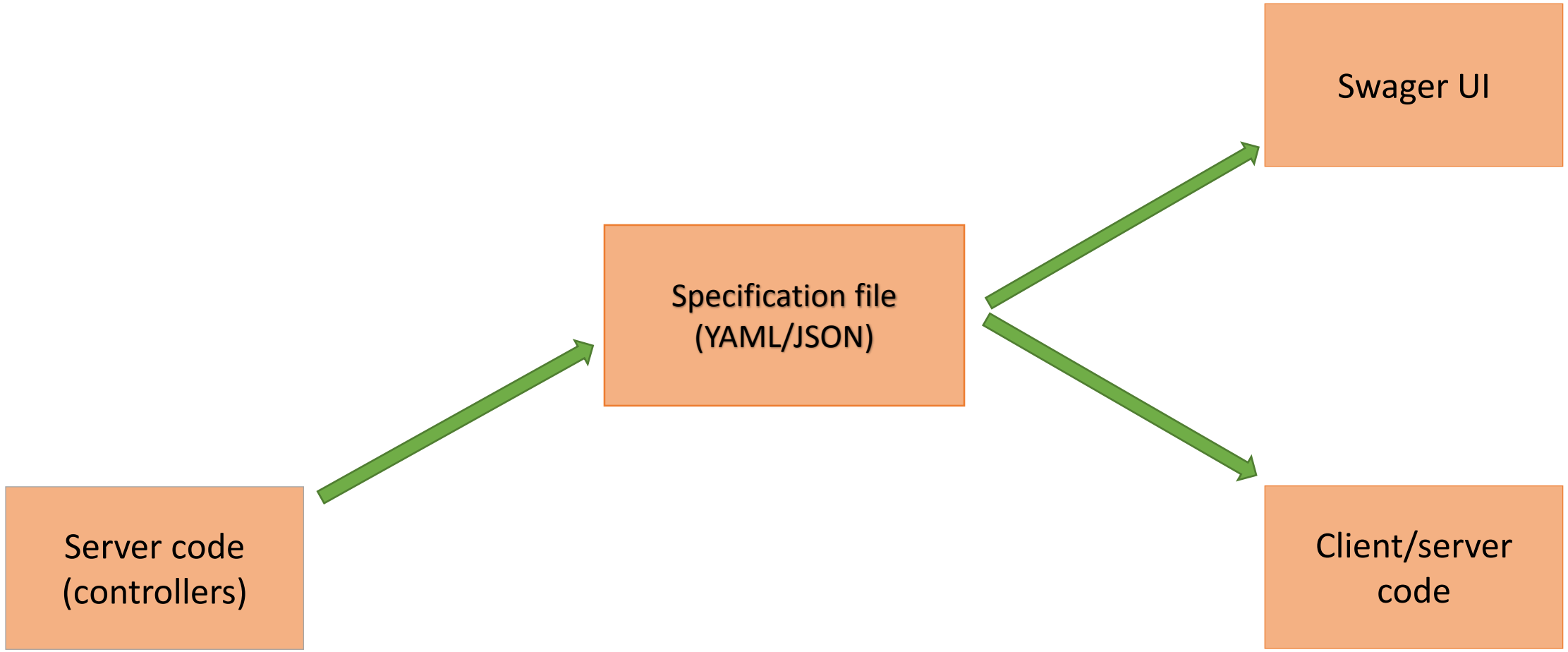
Swagger (OpenAPI)



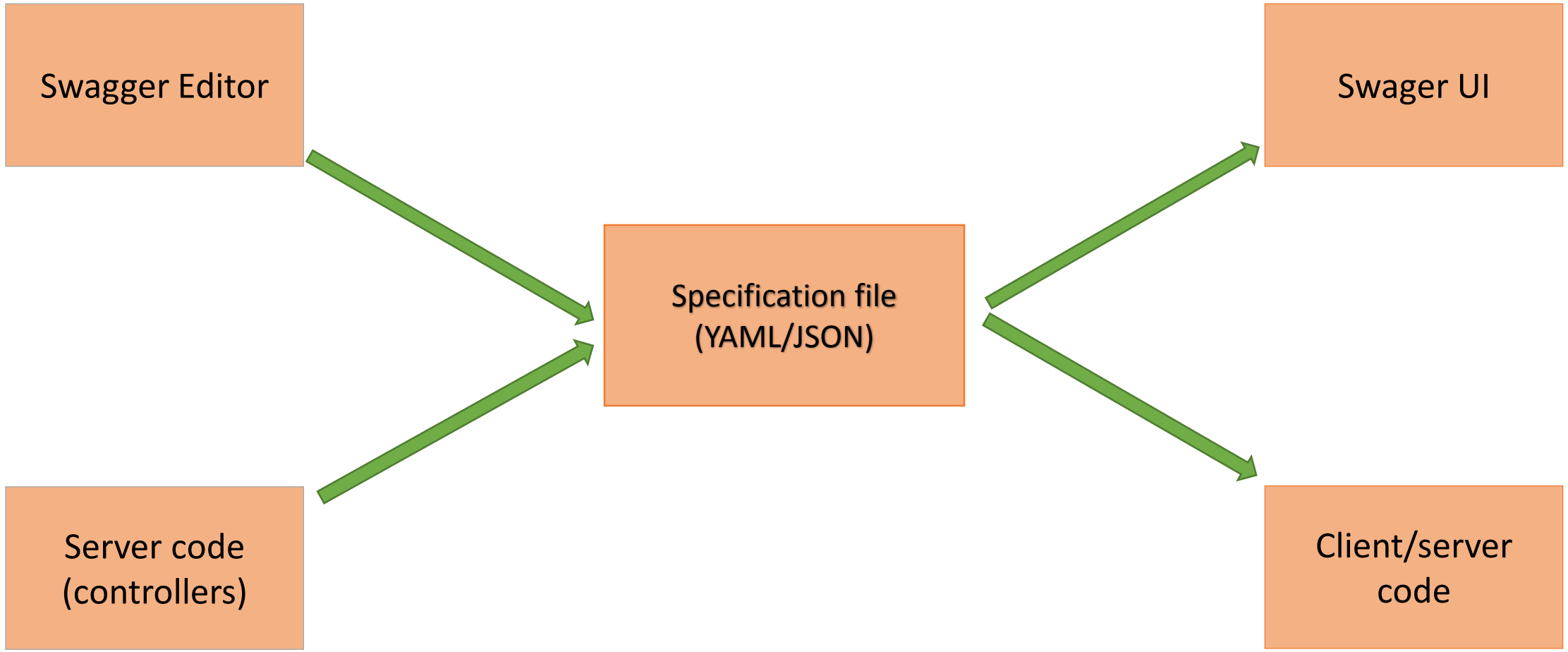
Swagger (OpenAPI)



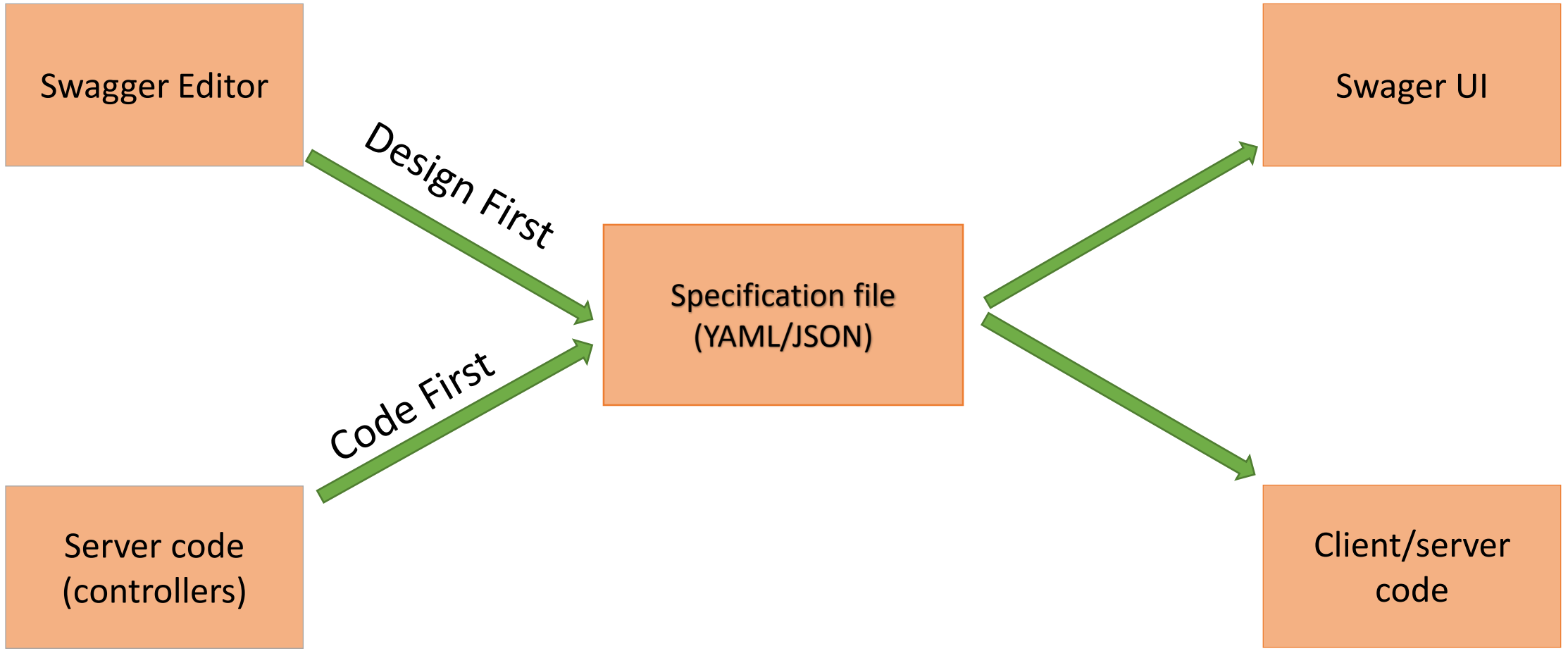
Swagger (OpenAPI)



Swagger (OpenAPI)



Подходы к документации



Swashbuckle (Code First)

- Можно сгенерировать Swagger спецификацию по коду
  - ASP.NET
  - ASP.NET Core
  - Azure Mobile backend
  - …

Swashbuckle (ASP.NET Core)

Install-Package Swashbuckle.AspNetCore

```
services.AddSwaggerGen(c => { c.SwaggerDoc("v1", new Info()});

app.UseSwagger().UseSwaggerUI(c => {
c.SwaggerEndpoint("/swagger/v1/swagger.json", "My API"); });
```


DEMO

Swashbuckle - возможности

- UI страница с описанием всех методов
  - Позволяет выполнять запросы к API!
- Подтягивает ваши XML комментарии
- Аттрибуты
  - HTTP коды
  - Значения по умолчанию
  - …
- Аутентификация
- Кастомизация UI

Code First -> Design First

- Специфика: пишем много API с нуля
- Требования нечеткие, нужно уточнять -> совместная работа над спецификацией

Swagger Hub

- “облако” для хранения Swagger спецификаций
- Включает в себя основные Swagger тулы: UI, Editor, CodeGen, Validator
- Возможности
  - Версионирование API
  - Коллаборация
  - Генерация кода client/server
- Пример: https://app.swaggerhub.com/apis/swagger-hub/registry-api/1.0.41


DEMO

JSON vs YAML

```json
"info": {
    "description":"This is a Petstore service.",
    "version":"1.0.0",
    "title":"Swagger Petstore",
    "termsOfService":"http://swagger.io/terms/",
    "contact": {
        "email":"apiteam@swagger.io"
    },
    "license": {
        "name":"Apache 2.0"
    }
}
```

```yaml
info:
  description: This is a Petstore service
  version: 1.0.0
  title: Swagger Petstore
  termsOfService: 'http://swagger.io/terms/'
  contact:
    email: apiteam@swagger.io
  license:
    name: Apache 2.0
```

Описание ресурсов

```
paths:
  '/pet':
    post: ···
    put: ···
  '/pet/findByStatus':
    get: ···
  '/pet/findByTags':
    get: ···
  '/pet/{petId}':
    get: ···
    post: ···
    delete: ···
```

Описание модели

```yaml
Pet:
  title: a Pet
  description: A pet for sale in the pet store
  type: object
  properties:
    id:
      type: integer
      format: int64
    category:
      $ref: '#/definitions/Category'
    name:
      type: string
```

Описание параметров

```yaml
parameters:
  - name: tags
    in: query
    description: Tags to filter by
    required: true
    type: array
    items:
      type: string
    collectionFormat: csv
```

Описание ответов

```
responses:
  '200':
    description: successful operation
    schema:
      $ref: '#/definitions/Pet'
  '400':
    description: Invalid ID supplied
  '404':
    description: Pet not found
```

Authentication

- Basic authentication
- API key
- OAuth 2


DEMO

Описание enum

```
ResponseCode:
  type: string
  description: response code
  enum:
    - Return code
    - InternalError
    - Success
    - Information
    - NotFound
    - DownloadFileNotAllowed
    - FileNotFound
    - FileAlreadyExists
```

```
Code:
  type: integer
  description: |
      Return code.
      InternalError = -1,
      Success = 0,
      Information = 1,
      NotFound = 404,
      DownloadFileNotAllowed = 9005,
      FileNotFound = 9006,
      FileAlreadyExists = 9007.
```

Метаданные

- Хотим возвращать метаданные для каждого ответа, e.g.
  - apiVersion
  - id
  - data: Product
- Для каждой модели потребуется две модели в Swagger

Swagger misc

- Генерация кода работает криво
- Детерминированность
- Есть недостатки
  - Описание enum
  - Наследование (allOf) – ломает кодогенерацию
  - Многострочные комментарии
  - Баги ☺

Code First VS Design First

Code First (Swashbuckle)	Design First (Swagger Hub)
Плюсы	
➢ 1-к-1 соответствие с исходным кодом	➢ Совместная работа над дизайном
Минусы	
➢ Возможно придется допиливать документацию	➢ Нужно тестировать, что реализация API соответствует спецификации
Лучше подходит для уже существующих API	Лучше подходит для новых “нетривиальных” API

Как все работает у нас

- Все спецификации в Swagger Hub
- Каждое API по base path возвращает
  - Версия API
  - Ссылка на документацию в Swagger Hub

Кто пользуется Swagger?

- Рекомендуемый Microsoft способ описания API
- OpenAPI Initiative (https://www.openapis.org)
  - Входит в Linux Foundation

Swagger резюме

- Доступен всем – тестерам/разработчикам/пользователям
- Генерация кода (клиент/сервер) на множество языков
- Развитая экосистема

Полезные ссылки

- Swagger https://app.swaggerhub.com/help/tutorials/writing-swagger-definitions
- Swagger viewer for VS Code https://marketplace.visualstudio.com/items?itemName=Arjun.swagger-viewer
- Swashbuckle
  - https://github.com/domaindrivendev/Swashbuckle.AspNetCore
  - https://docs.microsoft.com/en-us/aspnet/core/tutorials/web-api-help-pages-using-swagger
- Utilities http://swagger.io/open-source-integrations/
- Swagger Hub https://swaggerhub.com

Контакты

- Email: pavel.fedotovsky@lanit-tercom.com
- Skype: pavel.v.fedotovsky