



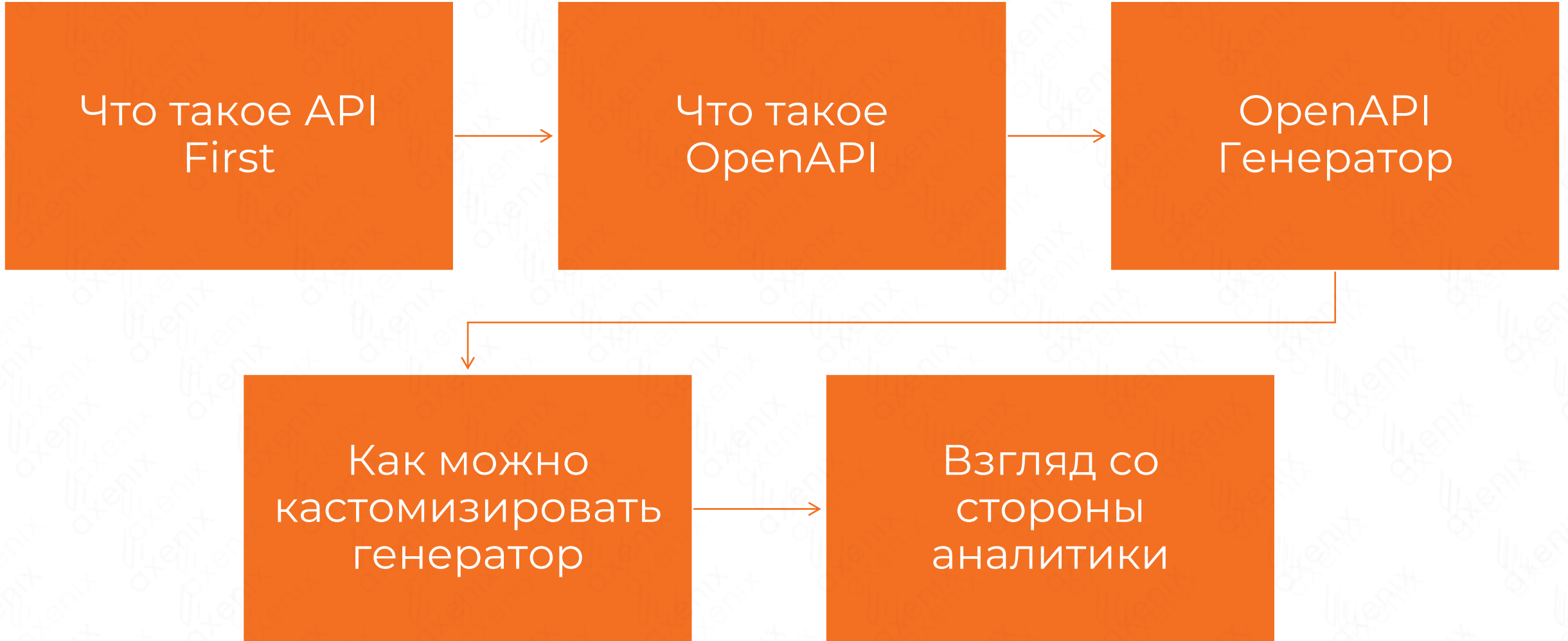
# OpenAPI: зачем и как его можно применить

OpenAPI, API First и OpenAPI





# План доклада





# Для кого доклад

## **Разработчик, Аналитик, Менеджер**

Любой, кто хочет ускорить и упростить работу в команде.

Наш герой — FE-разработчик Ваня, который устал читать docx.





## Савков Павел

Счастливый Java-разработчик, который подхватил проблему общения между проектами и внутри их, не могу мириться с несправедливостью

Вижу в OpenAPI выход из ситуации, поэтому делюсь с вами

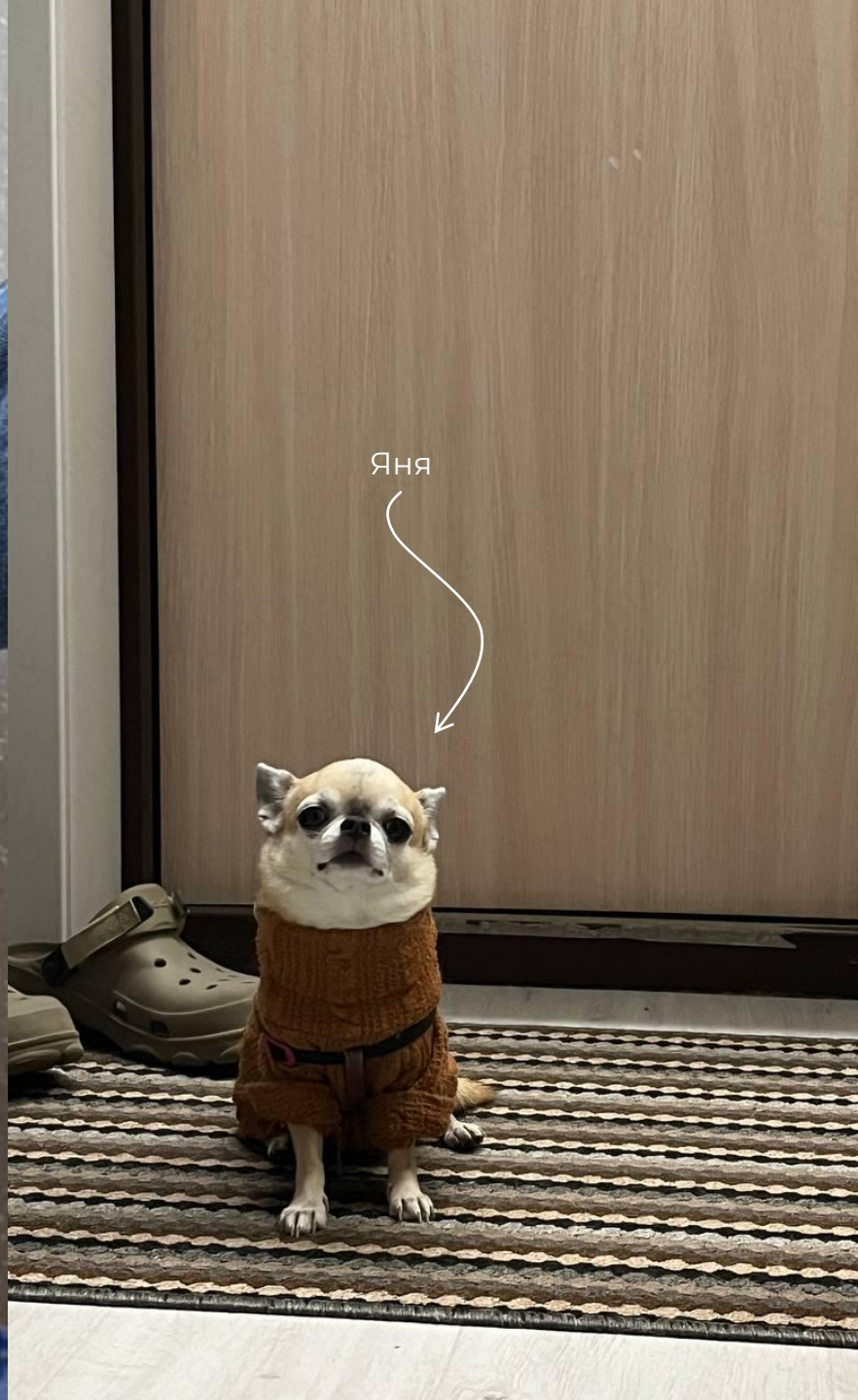
@daim\_dani

pavel.savkov@axenix.pro









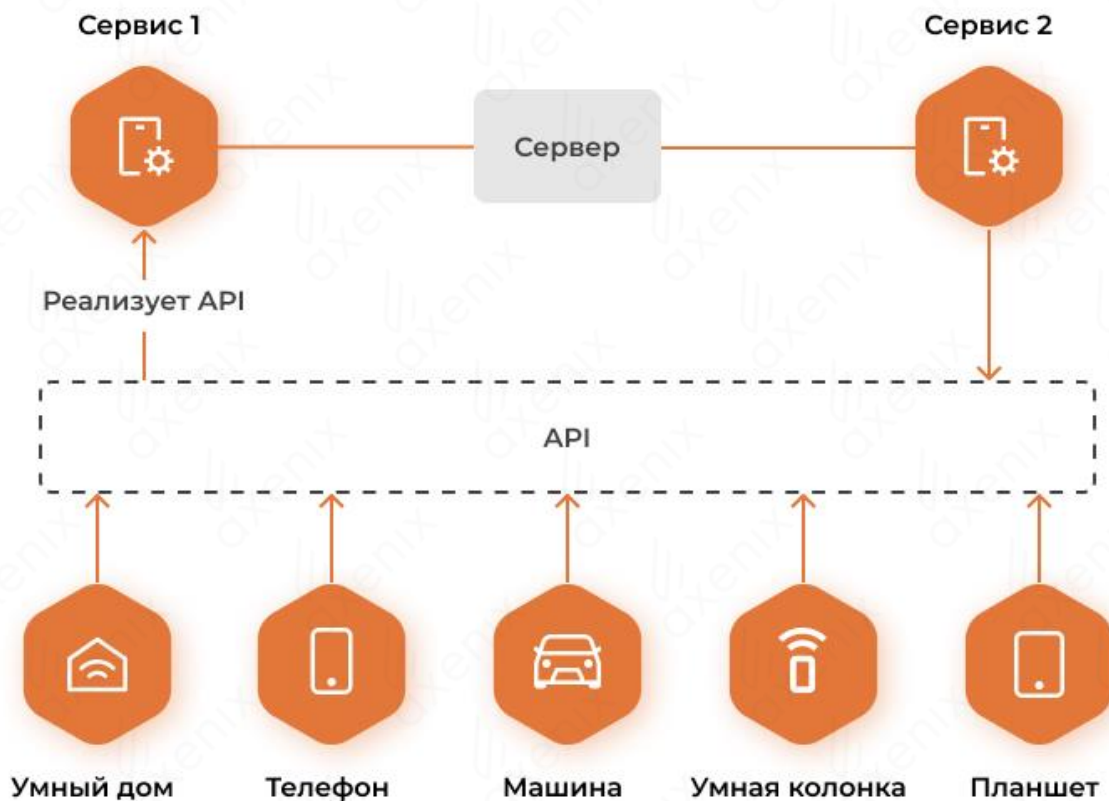
Яня





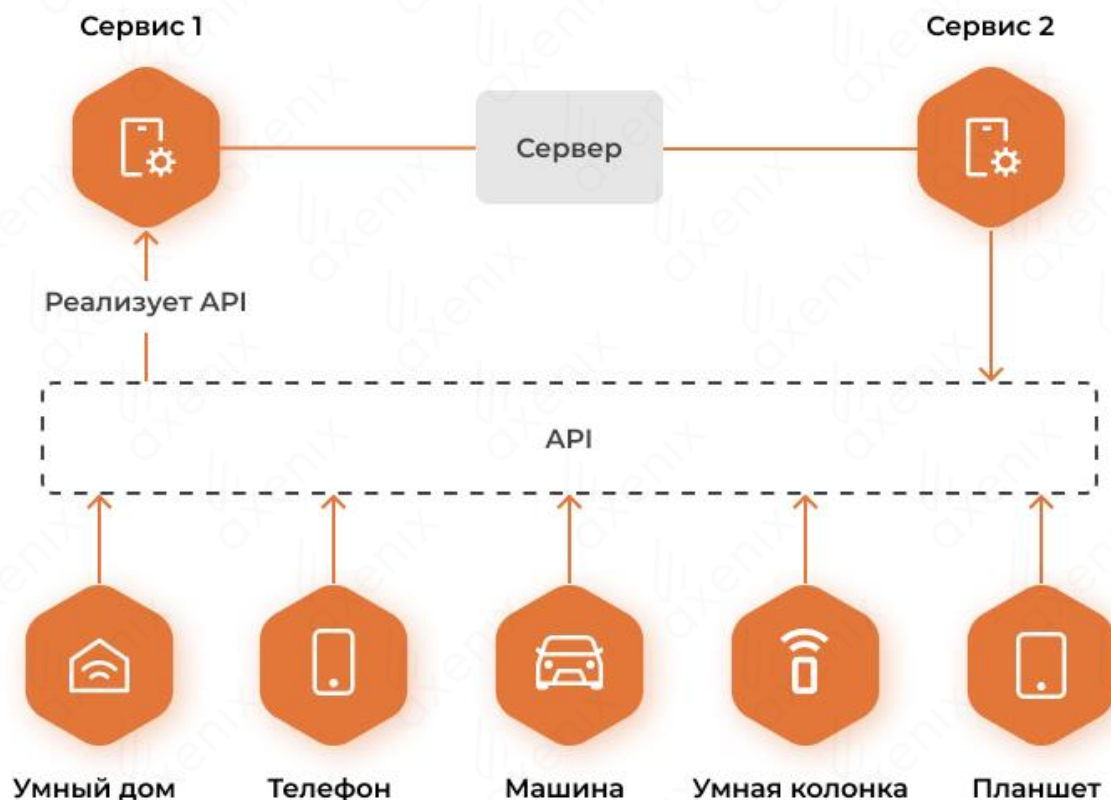
# Что такое API First

API – важная часть современного серверного приложения. Вы можете даже не знать количество клиентов вашего приложения.





# Что такое API First



API – важная часть современного серверного приложения. Вы можете даже не знать количество клиентов вашего приложения.

API First – подход разработки, при котором API является ключевой частью вашего продукта. Сначала утверждаем API, и только потом начинаем его реализацию.



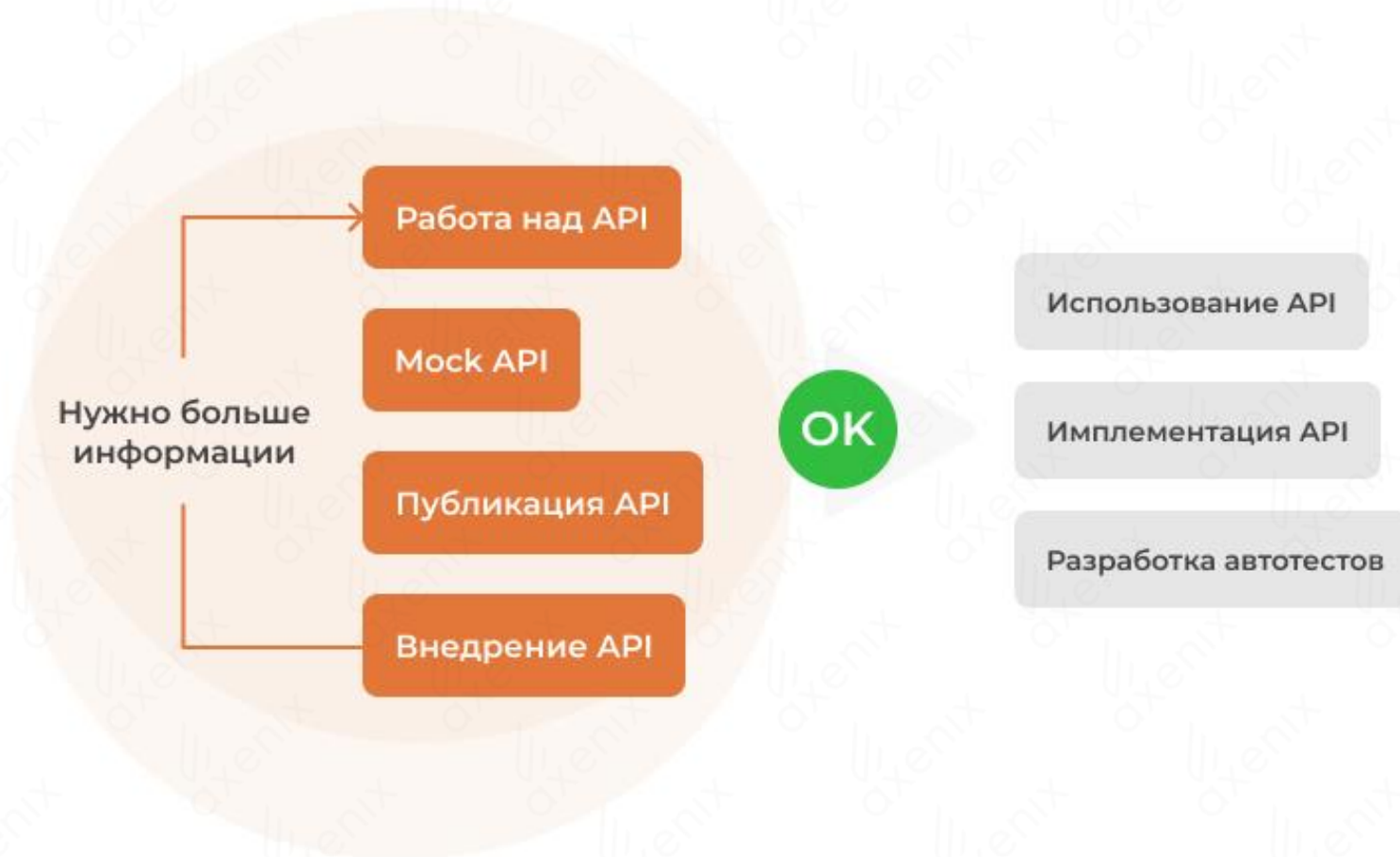


# Зачем нужен подход API First





# Зачем нужен подход API First



1. Ускоряет процесс разработки: многие процессы могут идти параллельно.





# Зачем нужен подход API First



1. Ускоряет процесс разработки: многие процессы могут идти параллельно.
2. Уменьшается риск ошибок, связанных с отличием моделей.



# Проблема согласования протоколов





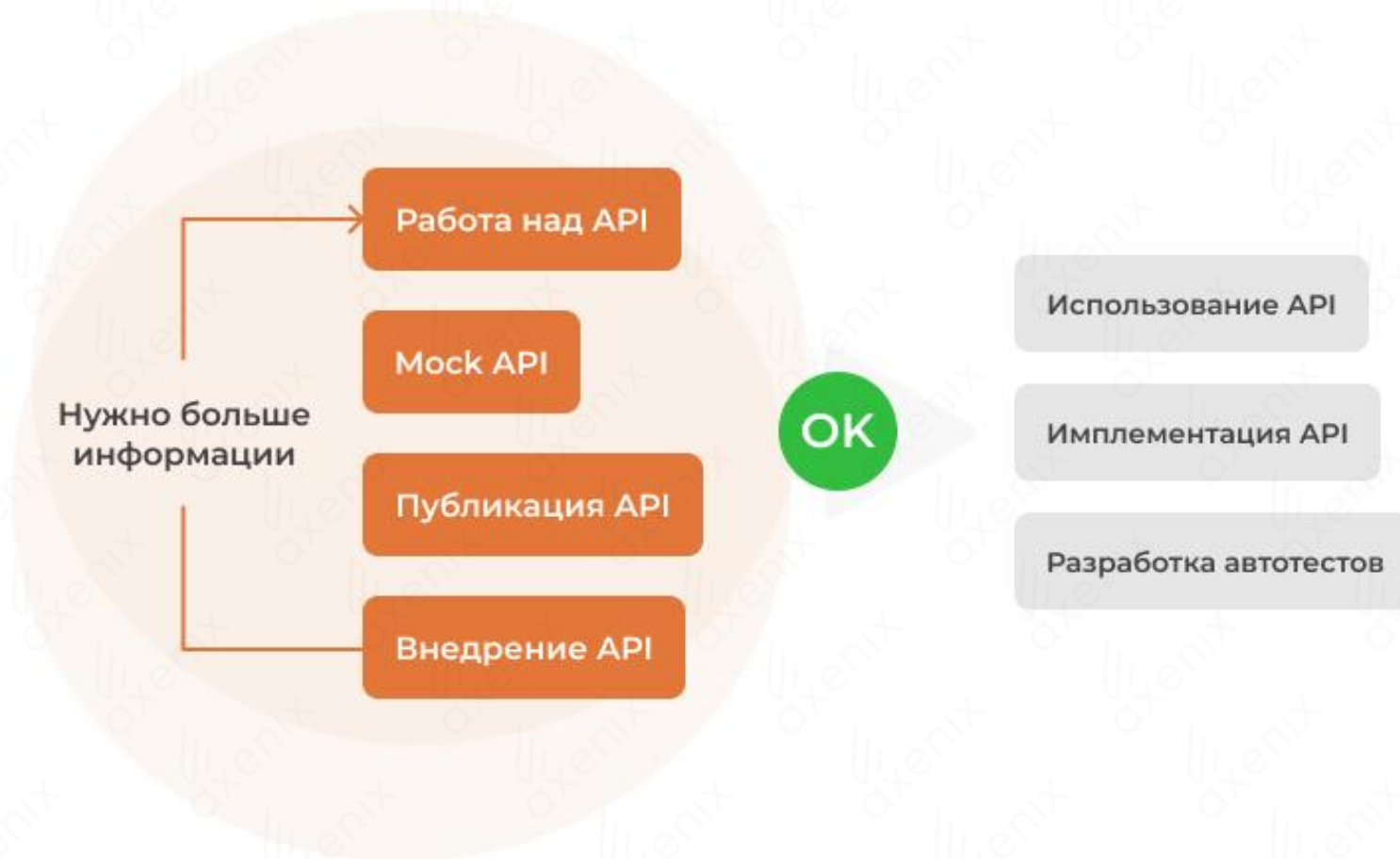


# Зачем нужен подход API First



1. Ускоряет процесс разработки: многие процессы могут идти параллельно.
2. Уменьшается риск ошибок, связанных с отличием моделей.
3. Улучшается качество проектирования API.

# Зачем нужен подход API First



1. Ускоряет процесс разработки: многие процессы могут идти параллельно.
2. Уменьшается риск ошибок, связанных с отличием моделей.
3. Улучшается качество проектирования API.
4. Появляются возможности генерации кода по спецификации API.





# OpenAPI – Что это?

**OpenAPI – это спецификация для описания HTTP API.**

1. Список доступных серверов.
2. URL, http метод.
3. Информация о схеме авторизации.
4. Описание headers, path params, request params и их обязательность.
5. Информация о статусах и ответах при этих статусах (разные body при разных статусах).
6. Описание полей request и response body.

```
servers:
  - url: http://petstore.swagger.io/v1
paths:
  /pets:
    get:
      summary: List all pets
      operationId: listPets
      tags: <1 item>
      security:
        - ApiKeyAuth: []
      parameters:
        - name: limit
          in: query # "path", "query", "cookie", "header"
          description: How many items to return at one time (max 100)
          required: false
          schema:
            type: integer
            format: int32
      responses:
        '200':
          description: A paged array of pets
          headers:
            x-next:
              description: A link to the next page of responses
              schema:
                type: string
          content:
            application/json:
              schema:
                $ref: "#/components/schemas/Pets"
```



# OpenAPI – Что это?

**OpenAPI – это спецификация для описания HTTP API.**

1. Список доступных серверов.
2. URL, http метод.
3. Информация о схеме авторизации.
4. Описание headers, path params, request params и их обязательность.
5. Информация о статусах и ответах при этих статусах (разные body при разных статусах).
6. Описание полей request и response body.

```
servers:
  - url: http://petstore.swagger.io/v1
paths:
  /pets:
    get:
      summary: List all pets
      operationId: listPets
      tags: <1 item>
      security:
        - ApiKeyAuth: []
      parameters:
        - name: limit
          in: query # "path", "query", "cookie", "header"
          description: How many items to return at one time (max 100)
          required: false
          schema:
            type: integer
            format: int32
      responses:
        '200':
          description: A paged array of pets
          headers:
            x-next:
              description: A link to the next page of responses
              schema:
                type: string
          content:
            application/json:
              schema:
                $ref: "#/components/schemas/Pets"
```





# OpenAPI – Что это?

**OpenAPI – это спецификация для описания HTTP API.**

1. Список доступных серверов.
2. URL, http метод.
3. Информация о схеме авторизации.
4. Описание headers, path params, request params и их обязательность.
5. Информация о статусах и ответах при этих статусах (разные body при разных статусах).
6. Описание полей request и response body.

```
servers:
  - url: http://petstore.swagger.io/v1
paths:
  /pets:
    get:
      summary: List all pets
      operationId: listPets
      tags: <1 item>
      security:
        - ApiKeyAuth: []
      parameters:
        - name: limit
          in: query # "path", "query", "cookie", "header"
          description: How many items to return at one time (max 100)
          required: false
          schema:
            type: integer
            format: int32
      responses:
        '200':
          description: A paged array of pets
          headers:
            x-next:
              description: A link to the next page of responses
              schema:
                type: string
          content:
            application/json:
              schema:
                $ref: "#/components/schemas/Pets"
```



# OpenAPI – Что это?

**OpenAPI – это спецификация для описания HTTP API.**

1. Список доступных серверов.
2. URL, http метод.
3. Информация о схеме авторизации.
4. Описание headers, path params, request params и их обязательность.
5. Информация о статусах и ответах при этих статусах (разные body при разных статусах).
6. Описание полей request и response body.

```
servers:
  - url: http://petstore.swagger.io/v1
paths:
  /pets:
    get:
      summary: List all pets
      operationId: listPets
      tags: <1 item>
      security:
        - ApiKeyAuth: []
      parameters:
        - name: limit
          in: query # "path", "query", "cookie", "header"
          description: How many items to return at one time (max 100)
          required: false
          schema:
            type: integer
            format: int32
      responses:
        '200':
          description: A paged array of pets
          headers:
            x-next:
              description: A link to the next page of responses
              schema:
                type: string
          content:
            application/json:
              schema:
                $ref: "#/components/schemas/Pets"
```



# OpenAPI – Что это?

**OpenAPI – это спецификация для описания HTTP API.**

1. Список доступных серверов.
2. URL, http метод.
3. Информация о схеме авторизации.
4. Описание headers, path params, request params и их обязательность.
5. Информация о статусах и ответах при этих статусах (разные body при разных статусах).
6. Описание полей request и response body.

```
servers:
  - url: http://petstore.swagger.io/v1
paths:
  /pets:
    get:
      summary: List all pets
      operationId: listPets
      tags: <1 item>
      security:
        - ApiKeyAuth: []
      parameters:
        - name: limit
          in: query # "path", "query", "cookie", "header"
          description: How many items to return at one time (max 100)
          required: false
          schema:
            type: integer
            format: int32
      responses:
        '200':
          description: A paged array of pets
          headers:
            x-next:
              description: A link to the next page of responses
              schema:
                type: string
          content:
            application/json:
              schema:
                $ref: "#/components/schemas/Pets"
```





# OpenAPI – Что это?

**OpenAPI – это спецификация для описания HTTP API.**

1. Список доступных серверов.
2. URL, http метод.
3. Информация о схеме авторизации.
4. Описание headers, path params, request params и их обязательность.
5. Информация о статусах и ответах при этих статусах (разные body при разных статусах).
6. Описание полей request и response body.

```
servers:
  - url: http://petstore.swagger.io/v1
paths:
  /pets:
    get:
      summary: List all pets
      operationId: listPets
      tags: <1 item>
      security:
        - ApiKeyAuth: []
      parameters:
        - name: limit
          in: query # "path", "query", "cookie", "header"
          description: How many items to return at one time (max 100)
          required: false
          schema:
            type: integer
            format: int32
      responses:
        '200':
          description: A paged array of pets
          headers:
            x-next:
              description: A link to the next page of responses
              schema:
                type: string
          content:
            application/json:
              schema:
                $ref: "#/components/schemas/Pets"
```



# OpenAPI – Что это?

**OpenAPI – это спецификация для описания HTTP API.**

1. Список доступных серверов.
2. URL, http метод.
3. Информация о схеме авторизации.
4. Описание headers, path params, request params и их обязательность.
5. Информация о статусах и ответах при этих статусах (разные body при разных статусах).
6. Описание полей request и response body.

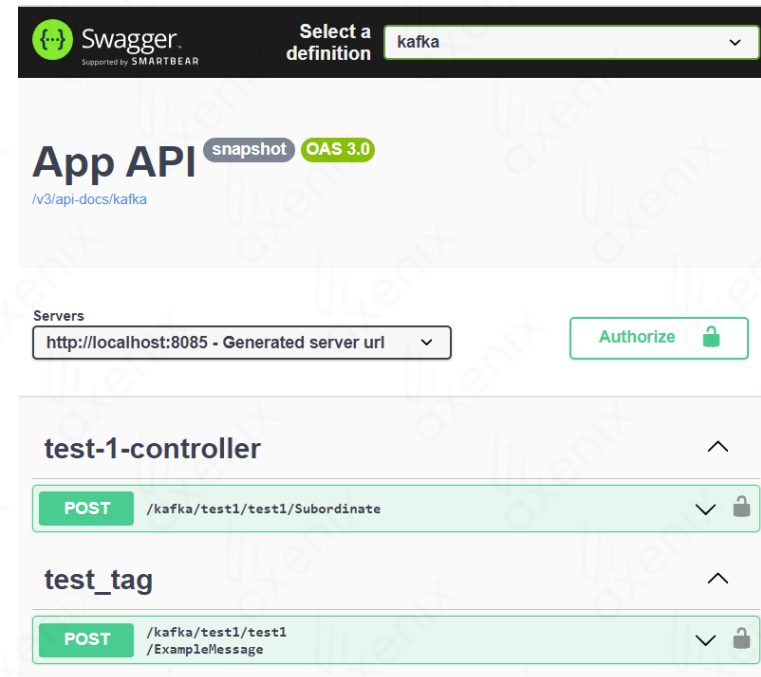
```
servers:
  - url: http://petstore.swagger.io/v1
paths:
  /pets:
    get:
      summary: List all pets
      operationId: listPets
      tags: <1 item>
      security:
        - ApiKeyAuth: []
      parameters:
        - name: limit
          in: query # "path", "query", "cookie", "header"
          description: How many items to return at one time (max 100)
          required: false
          schema:
            type: integer
            format: int32
      responses:
        '200':
          description: A paged array of pets
          headers:
            x-next:
              description: A link to the next page of responses
              schema:
                type: string
          content:
            application/json:
              schema:
                $ref: "#/components/schemas/Pets"
```



# OpenAPI – Что это?



OpenAPI



Swagger UI





# Как начать работать по API First подходу

## В начале разработки

1. Разработать (или доработать) спецификацию в формате OpenAPI.
2. Генерировать по ней код.



OpenAPI Generator

## В процессе разработки

0. Инициировать первую версию спецификации в формате OpenAPI





## Подробнее про API First

Подробнее про API First  
рассказано в нашей статье  
на Habr





# OpenAPI Generator

```
paths:
  /pets:
    get:
      summary: List all pets
      operationId: listPets
      tags: <1 item>
      security:
        - ApiKeyAuth: []
      parameters:
        - name: limit
          in: query # "path", "query", "cookie", "header"
          description: How many items to return at one time (max 100)
          required: false
          schema:
            type: integer
            format: int32
      responses:
        '200':
          description: A paged array of pets
          headers:
            x-next:
              description: A link to the next page of responses
              schema:
                type: string
          content:
            application/json:
              schema:
                $ref: "#/components/schemas/Pets"
```

OpenAPI Specification



OpenAPI Generator:  
генерация кода по  
спецификации API



Application  
code





# OpenAPI Generator

```
versions:
- url: http://petstore.swagger.io/v1

paths:
  /pets:
    get:
      summary: List all pets
      operationId: listPets
      tags: <1 item>
      security:
      - ApiKeyAuth: []
      parameters:
      - name: limit
        in: query # "path", "query", "cookie", "header"
        description: How many items to return at one time (max 100)
        required: false
        schema:
          type: integer
          format: int32
      responses:
        '200':
          description: A paged array of pets
          headers:
            x-next:
              description: A link to the next page of responses
              schema:
                type: string
          content:
            application/json:
              schema:
                $ref: "#/components/schemas/Pets"
```

OpenAPI Specification



OpenAPI Generator:  
генерация кода по  
спецификации API



Application  
code



# OpenAPI Generator

```

- url: http://petstore.swagger.io/v1
paths:
  /pets:
    get:
      summary: List all pets
      operationId: listPets
      tags: <1 item>
      security:
        - ApiKeyAuth: []
      parameters:
        - name: limit
          in: query # "path", "query", "cookie", "header"
          description: How many items to return at one time (max 100)
          required: false
          schema:
            type: integer
            format: int32
      responses:
        '200':
          description: A paged array of pets
          headers:
            x-next:
              description: A link to the next page of responses
              schema:
                type: string
          content:
            application/json:
              schema:
                $ref: "#/components/schemas/Pets"

```

OpenAPI Specification



OpenAPI Generator:  
генерация кода по  
спецификации API



Application  
code



# OpenAPI Generator

```
openapi: 3.0.1
info:
  title: Swagger Petstore
  version: 1.0.0
servers:
  - url: http://petstore.swagger.io/v1
paths:
  /pets:
    get:
      summary: List all pets
      operationId: listPets
      tags:
        - <1 item>
      security:
        - ApiKeyAuth: []
      parameters:
        - name: limit
          in: query # "path", "query", "cookie", "header"
          description: How many items to return at one time (max 100)
          required: false
          schema:
            type: integer
            format: int32
      responses:
        '200':
          description: A paged array of pets
          headers:
            x-next:
              description: A link to the next page of responses
              schema:
                type: string
          content:
            application/json:
              schema:
                $ref: "#/components/schemas/Pets"
```

OpenAPI Specification



OpenAPI Generator:  
генерация кода по  
спецификации API



Application  
code





# OpenAPI Generator

```
openapi: 3.0.1
info:
  title: Swagger Petstore
  version: 1.0.0
servers:
  - url: http://petstore.swagger.io/v1
paths:
  /pets:
    get:
      summary: List all pets
      operationId: listPets
      tags:
        - <1 item>
      security:
        - ApiKeyAuth: []
      parameters:
        - name: limit
          in: query # "path", "query", "cookie", "header"
          description: How many items to return at one time (max 100)
          required: false
          schema:
            type: integer
            format: int32
      responses:
        '200':
          description: A paged array of pets
          headers:
            x-next:
              description: A link to the next page of responses
              schema:
                type: string
          content:
            application/json:
              schema:
                $ref: "#/components/schemas/Pets"
```

OpenAPI Specification



OpenAPI Generator:  
генерация кода по  
спецификации API



Application  
code



# OpenAPI Generator

```
openapi: 3.0.1
info:
  title: Swagger Petstore
  version: 1.0.0
servers:
  - url: http://petstore.swagger.io/v1
paths:
  /pets:
    get:
      summary: List all pets
      operationId: listPets
      tags:
        - <1 item>
      security:
        - ApiKeyAuth: []
      parameters:
        - name: limit
          in: query # "path", "query", "cookie", "header"
          description: How many items to return at one time (max 100)
          required: false
          schema:
            type: integer
            format: int32
      responses:
        '200':
          description: A paged array of pets
          headers:
            x-next:
              description: A link to the next page of responses
              schema:
                type: string
          content:
            application/json:
              schema:
                $ref: "#/components/schemas/Pets"
```

OpenAPI Specification



OpenAPI Generator:  
генерация кода по  
спецификации API



Application  
code



# OpenAPI Generator. Из чего состоит.

---





# OpenAPI Generator. Из чего состоит.



Swagger-парсер







# OpenAPI Generator. Из чего состоит.



Swagger-парсер

Набор mustache-файлов



# OpenAPI Generator. Mustache.

Общедоступный репозиторий



openapi generator



All

Videos

Images

Shopping

Forums

News

Web

More

Tools



GitHub

<https://github.com> > [OpenAPITools](#) > [openapi-generator](#)

## OpenAPITools/openapi-generator

**OpenAPI Generator** allows generation of API client libraries (SDK generation), server stubs, documentation and configuration automatically given an OpenAPI Spec.

Sign in

[Bin/Utils/openapi-generator.cli.sh](#)

[Releases 69](#)

[Issues](#)



# OpenAPI Generator. Mustache.

**Mustache – шаблон, по которому  
сгенерируется класс.**

[openapi-generator](#) / [modules](#) / [openapi-generator](#) / [src](#) / [main](#) / [resources](#) / [typescript-fetch](#) / [modelGenericInterfaces.mustache](#)



l0gicgate and Pierre Berube [typescript-fetch] - Fix Nullable Types (#18887)

Code

Blame

50 lines (49 loc) · 1.47 KB

```
1  /**
2   * {{#lambda.indented_star_1}}{{{unescapeDescription}}}{{/lambda.indented_star_1}}
3   * @export
4   * @interface {{classname}}
5   */
6   export interface {{classname}} {{#parent}}extends {{{.}}} {{/parent}}{
7     {{{#additionalPropertiesType}}}
8     [key: string]: {{{#additionalPropertiesType}}}{{{#hasVars}}} | any{{{/hasVars}}};
9     {{{/additionalPropertiesType}}}
10    {{{#vars}}}
11    /**
12     * {{#lambda.indented_star_4}}{{{unescapeDescription}}}{{/lambda.indented_star_4}}
13     * @type {{{=<% %>=}}}{{{<%&datatype%>}}}<%={{{ }}=}%>
14     * @memberof {{{classname}}}
15     {{{#deprecated}}}
16     * @deprecated|
17     {{{/deprecated}}}
18     */
19     {{{#isReadOnly}}}readonly {{{/isReadOnly}}}{{{name}}}{{{^required}}}{{{/required}}}: {{{#isEnum}}}{{{datatypeWithEnum}}}{{{#isNullable}}} | null{{{/isNu}}}
20    {{{/vars}}}
21    {{{#hasEnums}}}
22  }
```



# OpenAPI Generator. Mustache.

Mustache – шаблон, по которому  
сгенерируется класс.

## 1. Описание

[openapi-generator](#) / [modules](#) / [openapi-generator](#) / [src](#) / [main](#) / [resources](#) / [typescript-fetch](#) / [modelGenericInterfaces.mustache](#)



l0gicgate and Pierre Berube [typescript-fetch] - Fix Nullable Types (#18887)

Code

Blame

50 lines (49 loc) · 1.47 KB

```
1  /**
2   * {{#lambda.indented_star_1}}{{{unescapedDescription}}}{/lambda.indented_star_1}}
3   * @export
4   * @interface {{classname}}
5   */
6   export interface {{classname}} {{#parent}}extends {{{.}}} {/parent}{{
7   {{#additionalPropertiesType}}
8     [key: string]: {{{additionalPropertiesType}}}{{#hasVars}} | any{{/hasVars}};
9   {/additionalPropertiesType}}
10  {{#vars}}
11  /**
12   * {{#lambda.indented_star_4}}{{{unescapedDescription}}}{/lambda.indented_star_4}}
13   * @type {<%=>}<{{&datatype%}}<%=>{{ }}=>
14   * @memberof {{classname}}
15   {{#deprecated}}
16   * @deprecated
17   {/deprecated}}
18   */
19   {{#isReadOnly}}readonly {{/isReadOnly}}{{name}}{{^required}}{/required}: {{{isEnum}}}{{{datatypeWithEnum}}}{{#isNullable}} | null{{/isNu
20   {/vars}}
21   }{{#hasEnums}}
22
```





# OpenAPI Generator. Mustache.

**Mustache – шаблон, по которому сгенерируется класс.**

1. Описание
2. Объявление интерфейса и возможное наследство

[openapi-generator](#) / [modules](#) / [openapi-generator](#) / [src](#) / [main](#) / [resources](#) / [typescript-fetch](#) / [modelGenericInterfaces.mustache](#)



l0gicgate and Pierre Berube [typescript-fetch] - Fix Nullable Types (#18887)

Code

Blame

50 lines (49 loc) · 1.47 KB

```
1  /**
2   * {{#lambda.indented_star_1}}{{{unescapedDescription}}}{{/lambda.indented_star_1}}
3   * @export
4   * @interface {{classname}}
5   */
6  export interface {{classname}} {{#parent}}extends {{{.}}} {{/parent}}{
7    {{{#additionalPropertiesType}}}
8    [key: string]: {{{#additionalPropertiesType}}}{{{#hasVars}}} | any{{{/hasVars}}};
9    {{{/additionalPropertiesType}}}
10   {{{#vars}}}
11   /**
12    * {{#lambda.indented_star_4}}{{{unescapedDescription}}}{{/lambda.indented_star_4}}
13    * @type {{{=<% %>=}}}{{{<%&datatype%>}}}<%={{{ }}=}%>
14    * @memberof {{{classname}}}
15    {{{#deprecated}}}
16    * @deprecated|
17    {{{/deprecated}}}
18    */
19    {{{#isReadOnly}}}readonly {{{/isReadOnly}}}{{{name}}}{{{^required}}}{{{/required}}}: {{{#isEnum}}}{{{datatypeWithEnum}}}{{{#isNullable}}} | null{{{/isNu}}}
20   {{{/vars}}}
21   {{{#hasEnums}}}
22 }
```



# OpenAPI Generator. Mustache.

**Mustache – шаблон, по которому сгенерируется класс.**

1. Описание
2. Объявление интерфейса и возможное наследство
3. Поля и их описание

[openapi-generator](#) / [modules](#) / [openapi-generator](#) / [src](#) / [main](#) / [resources](#) / [typescript-fetch](#) / [modelGenericInterfaces.mustache](#)



Ilogicgate and Pierre Berube [typescript-fetch] - Fix Nullable Types (#18887)

Code

Blame

50 lines (49 loc) · 1.47 KB

```
1  /**
2   * {{#lambda.indented_star_1}}{{{unescapedDescription}}}{/lambda.indented_star_1}}
3   * @export
4   * @interface {{classname}}
5   */
6  export interface {{classname}} {{#parent}}extends {{{.}}} {/parent}{{
7    {{{additionalPropertiesType}}}
8    [key: string]: {{{additionalPropertiesType}}}{{#hasVars}} | any{{/hasVars}};
9    {{{additionalPropertiesType}}}
10   {{{#vars}}}
11   /**
12    * {{#lambda.indented_star_4}}{{{unescapedDescription}}}{/lambda.indented_star_4}}
13    * @type {{{=<% %>=}}}{{<%&datatype%>}}<%={{{ }}=}%>
14    * @memberof {{{classname}}}
15    {{{#deprecated}}}
16    * @deprecated|
17    {{{/deprecated}}}
18    */
19    {{{#isReadOnly}}}readonly {{{/isReadOnly}}}{{{name}}}{{^required}}{/{required}}: {{{#isEnum}}}{{{datatypeWithEnum}}}{{{#isNullable}}} | null{{/isNu
20    {{{/vars}}}
21    }{{{#hasEnums}}}
```



# OpenAPI Generator. Mustache.

User:

```
type: object
properties:
  id:
    type: integer
    format: int64
  login:
    type: string
  name:
    type: string
  surname:
    type: string
  image:
    type: string
```

[openapi-generator](#) / [modules](#) / [openapi-generator](#) / [src](#) / [main](#) / [resources](#) / [typescript-fetch](#) / [modelGenericInterfaces.mustache](#)



l0gicgate and Pierre Berube [typescript-fetch] - Fix Nullable Types (#18887)

Code

Blame

50 lines (49 loc) · 1.47 KB

```
1  /**
2   * {{#lambda.indented_star_1}}{{unescapedDescription}}{{/lambda.indented_star_1}}
3   * @export
4   * @interface {{classname}}
5   */
6   export interface {{classname}} {{#parent}}extends {{{.}}} {{/parent}}{
7     {{#additionalPropertiesType}}
8     [key: string]: {{{additionalPropertiesType}}}{{#hasVars}} | any{{{hasVars}}};
9     {{/additionalPropertiesType}}
10    {{#vars}}
11    /**
12     * {{#lambda.indented_star_4}}{{unescapedDescription}}{{/lambda.indented_star_4}}
13     * @type {{{=<% %>=}}}{{{<%&datatype%>}<%={{{ }}=}%}}
14     * @memberof {{classname}}
15     {{{deprecated}}}
16     * @deprecated
17     {{{/deprecated}}}
18     */
19     {{{#isReadOnly}}}readonly {{{/isReadOnly}}}{{{name}}}{{{^required}}}{{{/required}}}: {{{#isEnum}}}{{{datatypeWithEnum}}}{{{#isNullable}}} | null{{{/isNu
20    {{{/vars}}}
21    {{{#hasEnums}}}
22  }
```



# OpenAPI Generator. Mustache.

User:

type: **object**

properties:

id:

type: integer  
format: int64

login:

type: string

name:

type: string

surname:

type: string

image:

type: string

[openapi-generator](#) / [modules](#) / [openapi-generator](#) / [src](#) / [main](#) / [resources](#) / [typescript-fetch](#) / **modelGenericInterfaces.mustache**



l0gicgate and Pierre Berube [typescript-fetch] - Fix Nullable Types (#18887)

Code

Blame

50 lines (49 loc) · 1.47 KB

```
1  /**
2   * {{#lambda.indented_star_1}}{{{unescapeDescription}}}{{/lambda.indented_star_1}}
3   * @export
4   * @interface {{classname}}
5   */
6   export interface {{classname}} {{#parent}}extends {{{.}}} {{/parent}}{
7     {{{additionalPropertiesType}}}
8     [key: string]: {{{additionalPropertiesType}}}{{#hasVars}} | any{{{/hasVars}}};
9     {{{additionalPropertiesType}}}
10    {{{#vars}}}
11    /**
12     * {{#lambda.indented_star_4}}{{{unescapeDescription}}}{{/lambda.indented_star_4}}
13     * @type {{{=<% %>=}}}{{{<%&datatype%><%={{{ }}=}}}
14     * @memberof {{{classname}}}
15     {{{#deprecated}}}
16     * @deprecated
17     {{{/deprecated}}}
18     */
19     {{{#isReadOnly}}}readonly {{{/isReadOnly}}}{{{name}}}{{{^required}}}{{{/required}}}: {{{#isEnum}}}{{{datatypeWithEnum}}}{{{#isNullable}}} | null{{{/isNu}}}
20    {{{/vars}}}
21    {{{#hasEnums}}}
22  }
```





# OpenAPI Generator. Mustache.

User:

type: object

properties:

id:

type: integer

format: int64

login:

type: string

name:

type: string

surname:

type: string

image:

type: string

[openapi-generator](#) / [modules](#) / [openapi-generator](#) / [src](#) / [main](#) / [resources](#) / [typescript-fetch](#) / [modelGenericInterfaces.mustache](#)



Ilogicgate and Pierre Berube [typescript-fetch] - Fix Nullable Types (#18887)



Code

Blame

50 lines (49 loc) · 1.47 KB

```
1  /**
2   * {{#lambda.indented_star_1}}{{unescapedDescription}}{{/lambda.indented_star_1}}
3   * @export
4   * @interface {{classname}}
5   */
6  export interface {{classname}} {{#parent}}extends {{{.}}} {{/parent}}{
7    {{{#additionalPropertiesType}}}
8      [key: string]: {{{additionalPropertiesType}}}{{{#hasVars}}} | any{{{/hasVars}}};
9    {{{/additionalPropertiesType}}}
10   {{{#vars}}}
11   /**
12    * {{#lambda.indented_star_4}}{{unescapedDescription}}{{/lambda.indented_star_4}}
13    * @type {{{=<% %>=}}}{{{<%&datatype%><%={{{ }}=}}}
14    * @memberof {{{classname}}}
15    {{{#deprecated}}}
16    * @deprecated
17    {{{/deprecated}}}
18    */
19    {{{#isReadOnly}}}readonly {{{/isReadOnly}}}{{{name}}}{{{^required}}}{{{/required}}}: {{{#isEnum}}}{{{datatypeWithEnum}}}{{{#isNullable}}} | null{{{/isNu}}}
20    {{{/vars}}}
21    {{{#hasEnums}}}
22  }
```



# OpenAPI Generator. Mustache.

User:

type: object

properties:

id:

type: integer  
format: int64

login:

type: string

name:

type: string

surname:

type: string

image:

type: string

[openapi-generator](#) / [modules](#) / [openapi-generator](#) / [src](#) / [main](#) / [resources](#) / [typescript-fetch](#) / [modelGenericInterfaces.mustache](#)



l0gicgate and Pierre Berube [typescript-fetch] - Fix Nullable Types (#18887) ✓

Code

Blame

50 lines (49 loc) · 1.47 KB

```
1  /**
2   * {{#lambda.indented_star_1}}{{unescapedDescription}}{{/lambda.indented_star_1}}
3   * @export
4   * @interface {{classname}}
5   */
6  export interface {{classname}} {{#parent}}extends {{{.}}} {{/parent}}{
7    {{#additionalPropertiesType}}
8      [key: string]: {{{additionalPropertiesType}}}{{#hasVars}} | any{{{hasVars}}};
9    {{{additionalPropertiesType}}}
10   {{#vars}}
11   /**
12    * {{#lambda.indented_star_4}}{{unescapedDescription}}{{/lambda.indented_star_4}}
13    * @type {{<%=>}}{{<%&datatype%>}<%={{ }}=}}
14    * @memberof {{classname}}
15    {{{deprecated}}}
16    * @deprecated
17    {{{/deprecated}}}
18    */
19    {{{isReadOnly}}}readonly {{{isReadOnly}}}{{name}}{{^required}}?{{/required}}: {{{isEnum}}}{{{datatypeWithEnum}}}{{{isNullable}}} | null{{{isNu
20   {{{/vars}}}
21   {{{#enums}}}
22
```



# OpenAPI Generator. Mustache.

Там выше стандартное описание

```

User.ts x
11  * https://openapi-generator.tech
12  * Do not edit the class manually.
13  */
14
15  import { exists, mapValues } from '../runtime';
16  /**
17   *
18   * @export
19   * @interface User
20   */
21  export interface User {
22      /**
23       *
24       * @type {number}
25       * @memberof User
26       */
27      id?: number;
28      /**
29       *
30       * @type {string}
31       * @memberof User
32       */
33      login?: string;

```

Ниже остальные поля



# OpenAPI Generator. Mustache.

User:

type: object

properties:

id:

type: integer

format: int64

login:

type: string

nullable: true

default: "Pasha"

required: true

name:

type: string

surname:

type: string

image:

type: string





# OpenAPI Generator. Mustache.

User:

type: object

properties:

id:

type: integer

format: int64

login:

type: string

nullable: true

default: "Pasha"

required: true

name:

type: string

surname:

type: string

image:

type: string



# OpenAPI Generator. Mustache.

User:

```
type: object
properties:
  id:
    type: integer
    format: int64
  login:
    type: string
    nullable: true
    default: "Pasha"
    required: true
  name:
    type: string
  surname:
    type: string
  image:
    type: string
```



# OpenAPI Generator. Mustache.

User:

```
type: object
properties:
  id:
    type: integer
    format: int64
  login:
    type: string
    nullable: true
    default: "Pasha"
    required: true
  name:
    type: string
  surname:
    type: string
  image:
    type: string
```

```
{{#vars}}
/**
 * {{#lambda.indented_star_4}}{{unescapedDescription}}
{{/lambda.indented_star_4}}
 * @type {{=<% %>=}}{{<%&datatype%><%={{ }}=%>}}
 * @memberof {{classname}}
{{#deprecated}}
 * @deprecated
{{/deprecated}}
*/
{{#isReadOnly}}readonly {{/isReadOnly}}{{name}}{{^required}}?{{/required}}:
{{#isEnum}}{{datatypeWithEnum}}{{#isNullable}} | null{{/isNullable}}
{{/isEnum}}{{^isEnum}}{{datatype}}{{#isNullable}} | null{{/isNullable}}
{{/isEnum}};
{{/vars}}
```



# OpenAPI Generator. Mustache.

User:

```
type: object
properties:
  id:
    type: integer
    format: int64
  login:
    type: string
    nullable: true
    default: "Pasha"
    required: true
  name:
    type: string
  surname:
    type: string
  image:
    type: string
```

```
{{#vars}}
/**
 * {{#lambda.indented_star_4}}{{{unescapedDescription}}}}
{{/lambda.indented_star_4}}
 * @type {=<% %>=}}<{{&datatype%}}>{{%={{ }}=}}%>
 * @memberof {{{classname}}}
{{{#deprecated}}}
 * @deprecated
{{{/deprecated}}}
 */
{{{#isReadOnly}}}readonly {{{/isReadOnly}}}{{{{name}}}{^required}}?{{{/required}}}:
{{{#isEnum}}}{{{{datatypeWithEnum}}}{^isNullable}} | null{{{/isNullable}}}
{{{/isEnum}}}{^isEnum}}<{{{datatype}}}>{{{#isNullable}} | null{{{/isNullable}}}
{{{/isEnum}}};
{{{/vars}}}
```



# OpenAPI Generator. Mustache.

User:

```
type: object
properties:
  id:
    type: integer
    format: int64
  login:
    type: string
    nullable: true
    default: "Pasha"
    required: true
  name:
    type: string
  surname:
    type: string
  image:
    type: string
```

```
{{#vars}}
/**
 * {{#lambda.indented_star_4}}{{{unescapedDescription}}}}
{{/lambda.indented_star_4}}
 * @type {<%=> %>=}{{<%&datatype%>}}<%={{ }}=%>
 * @memberof {{classname}}
{{#deprecated}}
 * @deprecated
{{/deprecated}}
*/
{{#isReadOnly}}readonly {{/isReadOnly}}{{name}}{{^required}}?{{/required}}:
{{#isEnum}}{{{datatypeWithEnum}}}{{#isNullable}} | null{{/isNullable}}
{{/isEnum}}{{^isEnum}}{{{datatype}}}{{#isNullable}} | null{{/isNullable}}
{{/isEnum}};
{{/vars}}
```





# OpenAPI Generator. Mustache.

User:

```
type: object
properties:
  id:
    type: integer
    format: int64
  login:
    type: string
    nullable: true
    default: "Pasha"
    required: true
  name:
    type: string
  surname:
    type: string
  image:
    type: string
```

```
{{#vars}}
/**
 * {{#lambda.indented_star_4}}{{{unescapedDescription}}}}
{{/lambda.indented_star_4}}
 * @type {<%=> %>=>}<{{datatype%}}>{{<%=> %>=>}}
 * @memberof {{classname}}
{{#deprecated}}
 * @deprecated
{{/deprecated}}
*/
{{#isReadOnly}}readonly {{/isReadOnly}}{{name}}<{{^required}}>{{/required}}<:
{{#isEnum}}{{{datatypeWithEnum}}}<{{#isNullable}} | null{{/isNullable}}>
{{/isEnum}}{{^isEnum}}<{{datatype}}>{{#isNullable}} | null{{/isNullable}}>
{{/isEnum}};
{{/vars}}
```



# OpenAPI Generator. Mustache.

User:

```
type: object
properties:
  id:
    type: integer
    format: int64
  login:
    type: string
    nullable: true
    default: "Pasha"
    required: true
  name:
    type: string
  surname:
    type: string
  image:
    type: string
```

```
{{#vars}}
/**
 * {{#lambda.indented_star_4}}{{unescapedDescription}}
{{/lambda.indented_star_4}}
 * @type {{=<% %>=}}{{<%&datatype%>}}<%={{ }}=%>
 * @memberof {{classname}}
{{#deprecated}}
 * @deprecated
{{/deprecated}}
 */
{{#isReadOnly}}readonly {{/isReadOnly}}{{name}}{{^required}}?{{/required}}:
{{#isEnum}}{{datatypeWithEnum}}{{#isNullable}} | null{{/isNullable}}
{{/isEnum}}{{^isEnum}}{{datatype}}{{#isNullable}} | null{{/isNullable}}
{{/isEnum}};
{{/vars}}
```



# OpenAPI Generator. Зачем

---

**Ну и что?**



# OpenAPI Generator. Кастомизация

---



# OpenAPI Generator. Кастомизация

components:

schemas:





# OpenAPI Generator. Кастомизация

```
components:  
  schemas:  
    Animal:  
      type: object
```



# OpenAPI Generator. Кастомизация

```
components:
  schemas:
    Animal:
      type: object
      discriminator:
```



# OpenAPI Generator. Кастомизация

```
components:
  schemas:
    Animal:
      type: object
      discriminator:
        propertyName: type
```



# OpenAPI Generator. Кастомизация

```
components:
  schemas:
    Animal:
      type: object
      discriminator:
        propertyName: type

      properties:
        type:
          $ref: "#/components/schemas/AnimalType"
```



# OpenAPI Generator. Кастомизация

```
components:
  schemas:
    Animal:
      type: object
      discriminator:
        propertyName: type

      properties:
        type:
          $ref: "#/components/schemas/AnimalType"
```

```
AnimalType:
  type: string
  enum:
    - DOG
    - WHALE
```





# OpenAPI Generator. Кастомизация

```
components:
  schemas:
    Animal:
      type: object
      discriminator:
        propertyName: type
        mapping:
          DOG: "#/components/schemas/Dog"
          WHALE: "#/components/schemas/Whale"
      properties:
        type:
          $ref: "#/components/schemas/AnimalType"
```

```
AnimalType:
  type: string
  enum:
    - DOG
    - WHALE
```



# OpenAPI Generator. Кастомизация

```
components:
  schemas:
    Animal:
      type: object
      discriminator:
        propertyName: type
        mapping:
          DOG: "#/components/schemas/Dog"
          WHALE: "#/components/schemas/Whale"
      properties:
        type:
          $ref: "#/components/schemas/AnimalType"
        id:
          type: integer
          format: int64
        name:
          type: string
```

```
AnimalType:
  type: string
  enum:
    - DOG
    - WHALE
```



# OpenAPI Generator. Кастомизация

---

```
Dog:  
  type: object
```



# OpenAPI Generator. Кастомизация

---

```
Dog:  
  type: object  
  allOf:  
    - $ref: "#/components/schemas/Animal"
```



# OpenAPI Generator. Кастомизация

Dog:

type: object

allOf:

- \$ref: "#/components/schemas/Animal"
- type: object

properties:

barkCount:

type: integer

treatsEaten:

type: integer





# OpenAPI Generator. Кастомизация

Dog:

type: object

allOf:

- \$ref: "#/components/schemas/Animal"
- type: object

properties:

barkCount:

type: integer

treatsEaten:

type: integer

required:

- barkCount



# OpenAPI Generator. Кастомизация

Dog:

```
type: object
allof:
  - $ref: "#/components/schemas/Animal"
  - type: object
    properties:
      barkCount:
        type: integer
      treatsEaten:
        type: integer
    required:
      - barkCount
```

Whale:

```
type: object
allof:
  - $ref: "#/components/schemas/Animal"
  - type: object
    properties:
      distanceSailed:
        type: integer
      weight:
        type: integer
        format: int64
    required:
      - weight
```

```
@JsonPropertyOrder({
    Animal.JSON_PROPERTY_TYPE,
    Animal.JSON_PROPERTY_ID,
    Animal.JSON_PROPERTY_NAME
})

@javax.annotation.Generated(value = "org.openapitools.codegen.languages.JavaClientCodegen", date = "2024-04-16T11:49:21.1")
@JsonTypeInfo(use = JsonTypeInfo.Id.NAME, include = JsonTypeInfo.As.EXISTING_PROPERTY, property = "type", visible = true)
@JsonSubTypes({
    @JsonSubTypes.Type(value = Dog.class, name = "DOG"),
    @JsonSubTypes.Type(value = Whale.class, name = "WHALE"),
})
```

```
public class Animal {
    2 usages
    public static final String JSON_PROPERTY_TYPE = "type";
    7 usages
    private AnimalType type;

    2 usages
    public static final String JSON_PROPERTY_ID = "id";
    7 usages
    private Long id;

    public static final String JSON_PROPERTY_NAME = "name";
    7 usages
    private String name;
```

```
@JsonPropertyOrder({
    Animal.JSON_PROPERTY_TYPE,
    Animal.JSON_PROPERTY_ID,
    Animal.JSON_PROPERTY_NAME
})
```

```
@javax.annotation.Generated(value = "org.openapitools.codegen.languages.JavaClientCodegen", date = "2024-04-16T11:49:21.1")
@JsonTypeInfo(use = JsonTypeInfo.Id.NAME, include = JsonTypeInfo.As.EXISTING_PROPERTY, property = "type", visible = true)
@JsonSubTypes({
    @JsonSubTypes.Type(value = Dog.class, name = "DOG"),
    @JsonSubTypes.Type(value = Whale.class, name = "WHALE"),
})
```

```
public class Animal {
```

2 usages

```
public static final String JSON_PROPERTY_TYPE = "type";
```

7 usages

```
private AnimalType type;
```

2 usages

```
public static final String JSON_PROPERTY_ID = "id";
```

7 usages

```
private Long id;
```

```
public static final String JSON_PROPERTY_NAME = "name";
```

7 usages

```
private String name;
```

```
@JsonPropertyOrder({
    Animal.JSON_PROPERTY_TYPE,
    Animal.JSON_PROPERTY_ID,
    Animal.JSON_PROPERTY_NAME
})
```

```
@javax.annotation.Generated(value = "org.openapitools.codegen.languages.JavaClientCodegen", date = "2024-04-16T11:49:21.1")
@JsonTypeInfo(use = JsonTypeInfo.Id.NAME, include = JsonTypeInfo.As.EXISTING_PROPERTY, property = "type", visible = true)
```

```
@JsonSubTypes({
    @JsonSubTypes.Type(value = Dog.class, name = "DOG"),
    @JsonSubTypes.Type(value = Whale.class, name = "WHALE"),
})
```

```
public class Animal {
```

2 usages

```
public static final String JSON_PROPERTY_TYPE = "type";
```

7 usages

```
private AnimalType type;
```

2 usages

```
public static final String JSON_PROPERTY_ID = "id";
```

7 usages

```
private Long id;
```

```
public static final String JSON_PROPERTY_NAME = "name";
```

7 usages

```
private String name;
```

```
@JsonPropertyOrder({
    Animal.JSON_PROPERTY_TYPE,
    Animal.JSON_PROPERTY_ID,
    Animal.JSON_PROPERTY_NAME
})
```

```
@javax.annotation.Generated(value = "org.openapitools.codegen.languages.JavaClientCodegen", date = "2024-04-16T11:49:21.1")
@JsonTypeInfo(use = JsonTypeInfo.Id.NAME, include = JsonTypeInfo.As.EXISTING_PROPERTY, property = "type", visible = true)
```

```
@JsonSubTypes({
    @JsonSubTypes.Type(value = Dog.class, name = "DOG"),
    @JsonSubTypes.Type(value = Whale.class, name = "WHALE"),
})
```

```
public class Animal {
```

2 usages

```
public static final String JSON_PROPERTY_TYPE = "type";
```

7 usages

```
private AnimalType type;
```

2 usages

```
public static final String JSON_PROPERTY_ID = "id";
```

7 usages

```
private Long id;
```

```
public static final String JSON_PROPERTY_NAME = "name";
```

7 usages

```
private String name;
```



```
@JsonPropertyOrder({
    Whale.JSON_PROPERTY_DISTANCE_SAILED,
    Whale.JSON_PROPERTY_WEIGHT
})
@javax.annotation.Generated(value = "org.openapitools.codegen.languages.JavaCli
public class Whale extends Animal {
    2 usages
    public static final String JSON_PROPERTY_DISTANCE_SAILED = "distanceSailed";
    7 usages
    private Integer distanceSailed;

    2 usages
    public static final String JSON_PROPERTY_WEIGHT = "weight";
    7 usages
    private Long weight;
```

```
@JsonPropertyOrder({
    Whale.JSON_PROPERTY_DISTANCE_SAILED,
    Whale.JSON_PROPERTY_WEIGHT
})
@javax.annotation.Generated(value = "org.openapitools.codegen.languages.JavaCli
public class Whale extends Animal {
    2 usages
    public static final String JSON_PROPERTY_DISTANCE_SAILED = "distanceSailed";
    7 usages
    private Integer distanceSailed;

    2 usages
    public static final String JSON_PROPERTY_WEIGHT = "weight";
    7 usages
    private Long weight;
```

```
/**
 *
 * @export
 * @interface Animal
 */
export interface Animal {
  /**
   *
   * @type {AnimalType}
   * @memberof Animal
   */
  type?: AnimalType;
  /**
   *
   * @type {number}
   * @memberof Animal
   */
  id?: number;
  /**
   *
   * @type {string}
   * @memberof Animal
   */
  name?: string;
}
```

```
/**
 *
 * @export
 * @interface Animal
 */
export interface Animal {
  /**
   *
   * @type {AnimalType}
   * @memberof Animal
   */
  type?: AnimalType;
  /**
   *
   * @type {number}
   * @memberof Animal
   */
  id?: number;
  /**
   *
   * @type {string}
   * @memberof Animal
   */
  name?: string;
}
```

```
/**
 *
 * @export
 * @interface Whale
 */
export interface Whale extends Animal {
  /**
   *
   * @type {number}
   * @memberof Whale
   */
  distanceSailed?: number;
  /**
   *
   * @type {number}
   * @memberof Whale
   */
  weight: number;
}
```

```

/**
 *
 * @export
 * @interface Animal
 */
export interface Animal {
    /**
     *
     * @type {AnimalType}
     * @memberof Animal
     */
    type?: AnimalType;
    /**
     *
     * @type {number}
     * @memberof Animal
     */
    id?: number;
    /**
     *
     * @type {string}
     * @memberof Animal
     */
    name?: string;
}

```

```

/**
 *
 * @export
 * @interface Whale
 */
export interface Whale extends Animal {
    /**
     *
     * @type {number}
     * @memberof Whale
     */
    distanceSailed?: number;
    /**
     *
     * @type {number}
     * @memberof Whale
     */
    weight: number;
}

```

```
export function AnimalFromJSONTyped(json: any, ignoreDiscriminator: boolean): Animal {
  if ((json === undefined) || (json === null)) {
    return json;
  }
  if (!ignoreDiscriminator) {
    if (json['type'] === 'DOG') {
      return DogFromJSONTyped(json, true);
    }
    if (json['type'] === 'WHALE') {
      return WhaleFromJSONTyped(json, true);
    }
  }
  return {
    'type': !exists(json, 'type') ? undefined : AnimalTypeFromJSON(json['type']),
    'id': !exists(json, 'id') ? undefined : json['id'],
    'name': !exists(json, 'name') ? undefined : json['name'],
  };
}
```



```
export function AnimalFromJSONTyped(json: any, ignoreDiscriminator: boolean): Animal {  
  if ((json === undefined) || (json === null)) {  
    return json;  
  }  
  if (!ignoreDiscriminator) {  
    if (json['type'] === 'DOG') {  
      return DogFromJSONTyped(json, true);  
    }  
    if (json['type'] === 'WHALE') {  
      return WhaleFromJSONTyped(json, true);  
    }  
  }  
  return {  
    'type': !exists(json, 'type') ? undefined : AnimalTypeFromJSON(json['type']),  
    'id': !exists(json, 'id') ? undefined : json['id'],  
    'name': !exists(json, 'name') ? undefined : json['name'],  
  };  
}
```

```
export function AnimalFromJSONTyped(json: any, ignoreDiscriminator: boolean): Animal {  
  if ((json === undefined) || (json === null)) {  
    return json;  
  }  
  if (!ignoreDiscriminator) {  
    if (json['type'] === 'DOG') {  
      return DogFromJSONTyped(json, true);  
    }  
    if (json['type'] === 'WHALE') {  
      return WhaleFromJSONTyped(json, true);  
    }  
  }  
  return {  
    'type': !exists(json, 'type') ? undefined : AnimalTypeFromJSON(json['type']),  
    'id': !exists(json, 'id') ? undefined : json['id'],  
    'name': !exists(json, 'name') ? undefined : json['name'],  
  };  
}
```

```
export function AnimalFromJSONTyped(json: any, ignoreDiscriminator: boolean): Animal {  
  if ((json === undefined) || (json === null)) {  
    return json;  
  }  
  if (!ignoreDiscriminator) {  
    if (json['type'] === 'DOG') {  
      return DogFromJSONTyped(json, true);  
    }  
    if (json['type'] === 'WHALE') {  
      return WhaleFromJSONTyped(json, true);  
    }  
  }  
  return {  
    'type': !exists(json, 'type') ? undefined : AnimalTypeFromJSON(json['type']),  
    'id': !exists(json, 'id') ? undefined : json['id'],  
    'name': !exists(json, 'name') ? undefined : json['name'],  
  };  
}
```

```
export function AnimalFromJSONTyped(json: any, ignoreDiscriminator: boolean): Animal {  
  if ((json === undefined) || (json === null)) {  
    return json;  
  }  
  if (!ignoreDiscriminator) {  
    if (json['type'] === 'DOG') {  
      return DogFromJSONTyped(json, true);  
    }  
    if (json['type'] === 'WHALE') {  
      return WhaleFromJSONTyped(json, true);  
    }  
  }  
  return {  
    'type': !exists(json, 'type') ? undefined : AnimalTypeFromJSON(json['type']),  
    'id': !exists(json, 'id') ? undefined : json['id'],  
    'name': !exists(json, 'name') ? undefined : json['name'],  
  };  
}
```

```
export function AnimalFromJSONTyped(json: any, ignoreDiscriminator: boolean): Animal {  
  if ((json === undefined) || (json === null)) {  
    return json;  
  }  
  if (!ignoreDiscriminator) {  
    if (json['type'] === 'DOG') {  
      return DogFromJSONTyped(json, true);  
    }  
    if (json['type'] === 'WHALE') {  
      return WhaleFromJSONTyped(json, true);  
    }  
  }  
  return {  
    'type': !exists(json, 'type') ? undefined : AnimalTypeFromJSON(json['type']),  
    'id': !exists(json, 'id') ? undefined : json['id'],  
    'name': !exists(json, 'name') ? undefined : json['name'],  
  };  
}
```



```
export function AnimalFromJSONTyped(json: any, ignoreDiscriminator: boolean): Animal {  
  if ((json === undefined) || (json === null)) {  
    return json;  
  }  
  if (!ignoreDiscriminator) {  
    if (json['type'] === 'DOG') {  
      return DogFromJSONTyped(json, true);  
    }  
    if (json['type'] === 'WHALE') {  
      return WhaleFromJSONTyped(json, true);  
    }  
  }  
  return {  
    'type': !exists(json, 'type') ? undefined : AnimalTypeFromJSON(json['type']),  
    'id': !exists(json, 'id') ? undefined : json['id'],  
    'name': !exists(json, 'name') ? undefined : json['name'],  
  };  
}
```



```
export function AnimalToJSON(value?: Animal | null): any {  
  if (value === undefined) {  
    return undefined;  
  }  
  if (value === null) {  
    return null;  
  }  
  return {  
  
    'type': AnimalTypeToJSON(value.type),  
    'id': value.id,  
    'name': value.name,  
  };  
}
```

```
export function AnimalToJSON(value?: Animal | null): any {  
  if (value === undefined) {  
    return undefined;  
  }  
  if (value === null) {  
    return null;  
  }  
  return {  
    'type': AnimalTypeToJSON(value.type),  
    'id': value.id,  
    'name': value.name,  
  };  
}
```

```
export function AnimalToJSON(value?: Animal | null): any {  
  if (value === undefined) {  
    return undefined;  
  }  
  if (value === null) {  
    return null;  
  }  
  return {  
    'type': AnimalTypeToJSON(value.type),  
    'id': value.id,  
    'name': value.name,  
  };  
}
```

```
export function AnimalToJSON(value?: Animal | null): any {  
  if (value === undefined) {  
    return undefined;  
  }  
  if (value === null) {  
    return null;  
  }  
  return {  
    'type': AnimalTypeToJSON(value.type),  
    'id': value.id,  
    'name': value.name,  
  };  
}
```

```
export function AnimalToJSON(value?: Animal | null): any {  
  if (value === undefined) {  
    return undefined;  
  }  
  if (value === null) {  
    return null;  
  }  
  return {  
    'type': AnimalTypeToJSON(value.type),  
    'id': value.id,  
    'name': value.name,  
  };  
}
```

?





Files



master



Go to file

- > typescript-axios
- ▼ typescript-fetch
  - ApiEntitiesRecord.mustache
  - ApiEntitiesReducer.mustache
  - ApiEntitiesSelectors.mustache
  - README.mustache
  - allSagas.mustache
  - apis.index.mustache
  - apis.mustache
  - gitignore
  - index.mustache
  - licenseInfo.mustache
  - modelEnum.mustache
  - modelEnumInterfaces.mustache
  - modelGeneric.mustache
  - modelGenericInterfaces.mustache
  - modelOneOf.mustache**
  - modelOneOfInterfaces.mustache

openapi-generator / modules / openapi-generator / src / main / resources / typescript-fetch / modelOneOf.mustache

↑ Top

Code

Blame

62 lines (56 loc) · 1.85 KB

Raw



```
6      {{{.}}}FromJSON,  
7      {{{.}}}FromJSONTyped,  
8      {{{.}}}ToJSON,  
9    } from './{{{.}}}{{importFileExtension}}';  
10    {{{/oneOf}}}   
11   
12    {{{/hasImports}}}   
13    {{{>modelOneOfInterfaces}}}   
14   
15    export function {{{classname}}}FromJSON(json: any): {{{classname}}} {  
16      return {{{classname}}}FromJSONTyped(json, false);  
17    }   
18   
19    export function {{{classname}}}FromJSONTyped(json: any, ignoreDiscriminator: boolean): {{{classname}}} {  
20      if (json == null) {  
21        return json;  
22      }   
23      {{{#discriminator}}}   
24        switch (json['{{{discriminator.propertyBaseName}}}']) {  
25          {{{#discriminator.mappedModels}}}   
26            case '{{{mappingName}}}':  
27              return Object.assign({}, {{{modelName}}}FromJSONTyped(json, true), { {{{discriminator.propertyName}}}: '{{{mappingName}}}' });  
28          {{{/discriminator.mappedModels}}}   
29          default:  
30            throw new Error(`No variant of {{{classname}}} exists with '{{{discriminator.propertyName}}}=${json['{{{discriminator.propertyName}}}']}`);  
31          }   
32      {{{/discriminator}}}   
33      {{{^discriminator}}}   
34        return {{{#oneOf}}}{{{.}}}FromJSONTyped(json, true){{{^-last}}} || {{{/-last}}}{{{/oneOf}}};  
35      {{{/discriminator}}}   
36    }   
37
```





Files



master



Go to file

- > typescript-axios
- ▼ typescript-fetch
  - ApiEntitiesRecord.mustache
  - ApiEntitiesReducer.mustache
  - ApiEntitiesSelectors.mustache
  - README.mustache
  - allSagas.mustache
  - apis.index.mustache
  - apis.mustache
  - gitignore
  - index.mustache
  - licenseInfo.mustache
  - modelEnum.mustache
  - modelEnumInterfaces.mustache
  - modelGeneric.mustache
  - modelGenericInterfaces.mustache
  - modelOneOf.mustache
  - modelOneOfInterfaces.mustache

openapi-generator / modules / openapi-generator / src / main / resources / typescript-fetch / modelOneOf.mustache

↑ Top

Code

Blame

62 lines (56 loc) · 1.85 KB

Raw



```
6      {{{.}}}FromJSON,  
7      {{{.}}}FromJSONTyped,  
8      {{{.}}}ToJSON,  
9    } from './{{{.}}}{{importFileExtension}}';  
10    {{{/oneOf}}}  
11  
12    {{{/hasImports}}}  
13    {{{>modelOneOfInterfaces}}}  
14  
15    export function {{{classname}}}FromJSON(json: any): {{{classname}}} {  
16      return {{{classname}}}FromJSONTyped(json, false);  
17    }  
18  
19    export function {{{classname}}}FromJSONTyped(json: any, ignoreDiscriminator: boolean): {{{classname}}} {  
20      if (json == null) {  
21        return json;  
22      }  
23      {{{#discriminator}}}  
24      switch (json['{{{discriminator.propertyBaseName}}}']) {  
25        {{{#discriminator.mappedModels}}}  
26          case '{{{mappingName}}}':  
27            return Object.assign({}, {{{modelName}}}FromJSONTyped(json, true), { {{{discriminator.propertyName}}}: '{{{mappingName}}}' });  
28        {{{/discriminator.mappedModels}}}  
29        default:  
30          throw new Error(`No variant of {{{classname}}} exists with '{{{discriminator.propertyName}}}=${json['{{{discriminator.propertyName}}}']}`);  
31        }  
32      {{{/discriminator}}}  
33      {{{^discriminator}}}  
34      return {{{#oneOf}}}{{{.}}}FromJSONTyped(json, true){{{^-last}}} || {{{/-last}}}{{{/oneOf}}};  
35    {{{/discriminator}}}  
36  }  
37
```



Files



master



Go to file

- > typescript-axios
- ▼ typescript-fetch
  - ApiEntitiesRecord.mustache
  - ApiEntitiesReducer.mustache
  - ApiEntitiesSelectors.mustache
  - README.mustache
  - allSagas.mustache
  - apis.index.mustache
  - apis.mustache
  - gitignore
  - index.mustache
  - licenseInfo.mustache
  - modelEnum.mustache
  - modelEnumInterfaces.mustache
  - modelGeneric.mustache
  - modelGenericInterfaces.mustache
  - modelOneOf.mustache**
  - modelOneOfInterfaces.mustache

openapi-generator / modules / openapi-generator / src / main / resources / typescript-fetch / modelOneOf.mustache

↑ Top

Code

Blame

62 lines (56 loc) · 1.85 KB

Raw



```
6      {{{.}}}FromJSON,  
7      {{{.}}}FromJSONTyped,  
8      {{{.}}}ToJSON,  
9    } from './{{{.}}}{{importFileExtension}}';  
10    {{{/oneOf}}}  
11  
12    {{{/hasImports}}}  
13    {{{>modelOneOfInterfaces}}}  
14  
15    export function {{{classname}}}FromJSON(json: any): {{{classname}}} {  
16      return {{{classname}}}FromJSONTyped(json, false);  
17    }  
18  
19    export function {{{classname}}}FromJSONTyped(json: any, ignoreDiscriminator: boolean): {{{classname}}} {  
20      if (json == null) {  
21        return json;  
22      }  
23      {{{#discriminator}}}  
24      switch (json['{{{discriminator.propertyBaseName}}}']) {  
25        {{{#discriminator.mappedModels}}}  
26          case '{{{mappingName}}}':  
27            return Object.assign({}, {{{modelName}}}FromJSONTyped(json, true), { {{{discriminator.propertyName}}}: '{{{mappingName}}}' });  
28        {{{/discriminator.mappedModels}}}  
29        default:  
30          throw new Error(`No variant of {{{classname}}} exists with '{{{discriminator.propertyName}}}=${json['{{{discriminator.propertyName}}}']}`);  
31        }  
32      {{{/discriminator}}}  
33      {{{^discriminator}}}  
34      return {{{#oneOf}}}{{{.}}}FromJSONTyped(json, true){{{^-last}}} || {{{/-last}}}{{{/oneOf}}};  
35    {{{/discriminator}}}  
36  }  
37
```

Project ▾

contract

src

main

resources

swagger

templates

JavaSpring

typescript-fetch

modelEnum.mustache

modelGeneric.mustache

modelOneOf.mustache

target

m pom.xml

modelOneOf.mustache ×

Plugins supporting \*.mustache files found.

Install Handlebars/Mustache plugin

Ignore extension

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

export function {{classname}}FromJSONTyped(json: any, ignoreDiscriminator: boolean): {{classname}} {

if ((json === undefined) || (json === null)) {

return json;

}

{{#discriminator}}

switch (json['{{discriminator.propertyName}}']) {

{{#discriminator.mappedModels}}

case '{{mappingName}}':

return {...{{modelName}}FromJSONTyped(json, true), {{discriminator.propertyName}}: '{{ma

{{/discriminator.mappedModels}}

default:

throw new Error(`No variant of {{classname}} exists with '{{discriminator.propertyName}}`

}

{{/discriminator}}

{{^discriminator}}

return { {{#oneOf}}...{{{.}}}FromJSONTyped(json, true){{^-last}}, {{/-last}}{{/oneOf}} };

{{/discriminator}}

}

export function {{classname}}ToJSON(value?: {{classname}} | null): any {

<Не обращаем сюда внимания>

}

}

Project ▾

contract

src

main

resources

swagger

templates

JavaSpring

typescript-fetch

modelEnum.mustache

modelGeneric.mustache

modelOneOf.mustache

target

pom.xml

modelOneOf.mustache ×

Plugins supporting \*.mustache files found.

[Install Handlebars/Mustache plugin](#) [Ignore extension](#)

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

export function {{classname}}FromJSONTyped(json: any, ignoreDiscriminator: boolean): {{classname}} {

if ((json === undefined) || (json === null)) {

return json;

}

{{#discriminator}}

switch (json['{{discriminator.propertyName}}']) {

{{#discriminator.mappedModels}}

case '{{mappingName}}':

return {...{{modelName}}FromJSONTyped(json, true), {{discriminator.propertyName}}: '{{ma

{{/discriminator.mappedModels}}

default:

throw new Error(`No variant of {{classname}} exists with '{{discriminator.propertyName}}`

}

{{/discriminator}}

{{^discriminator}}

return { {{#oneOf}}...{{.}}FromJSONTyped(json, true){{^-last}}, {{/-last}}{{/oneOf}} };

{{/discriminator}}

}

export function {{classname}}ToJSON(value?: {{classname}} | null): any {

<Не обращаем сюда внимания>

}

}

Project ▾

contract

src

main

resources

swagger

templates

JavaSpring

typescript-fetch

modelEnum.mustache

modelGeneric.mustache

modelOneOf.mustache

target

pom.xml

modelOneOf.mustache ×

Plugins supporting \*.mustache files found.

[Install Handlebars/Mustache plugin](#) [Ignore extension](#)

8

^ ▾

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

export function {{classname}}FromJSONTyped(json: any, ignoreDiscriminator: boolean): {{classname}} {

if ((json === undefined) || (json === null)) {

return json;

}

{{#discriminator}}

switch (json['{{discriminator.propertyName}}']) {

{{#discriminator.mappedModels}}

case '{{mappingName}}':

return {...{{modelName}}FromJSONTyped(json, true), {{discriminator.propertyName}}: '{{ma

{{/discriminator.mappedModels}}

default:

throw new Error(`No variant of {{classname}} exists with '{{discriminator.propertyName}}`

}

{{/discriminator}}

{{^discriminator}}

return { {{#oneOf}}...{{.}}FromJSONTyped(json, true){{^-last}}, {{/-last}}{{/oneOf}} };

{{/discriminator}}

}

export function {{classname}}ToJSON(value?: {{classname}} | null): any {

<Не обращаем сюда внимания>

}

}



Project ▾

▾ contract

▾ src

▾ main

▾ resources

&gt; swagger

▾ templates

&gt; JavaSpring

▾ typescript-fetch

≡ modelEnum.mustache

≡ modelGeneric.mustache

≡ modelOneOf.mustache

&gt; target

m pom.xml

≡ modelOneOf.mustache ×

❗ Plugins supporting \*.mustache files found.

[Install Handlebars/Mustache plugin](#)[Ignore extension](#)

✔ 8 ^ ▾

```
18  }
19
20  export function {{classname}}FromJSONTyped(json: any, ignoreDiscriminator: boolean): {{classname}} {
21      if ((json === undefined) || (json === null)) {
22          return json;
23      }
24      {{#discriminator}}
25          switch (json['{{discriminator.propertyName}}']) {
26              {{#discriminator.mappedModels}}
27                  case '{{mappingName}}':
28                      return {...{{modelName}}FromJSONTyped(json, true), {{discriminator.propertyName}}: '{{ma
29              {{/discriminator.mappedModels}}
30              default:
31                  throw new Error(`No variant of {{classname}} exists with '{{discriminator.propertyName}}
32              }
33      {{/discriminator}}
34      {{^discriminator}}
35          return { {{#oneOf}}...{{.}}FromJSONTyped(json, true){{^-last}}, {{/-last}}{{/oneOf}} };
36      {{/discriminator}}
37  }
38
39  export function {{classname}}ToJSON(value?: {{classname}} | null): any {
40      <Не обращаем сюда внимания>
41  }
42  }
43
```



Project ▾

▾ contract

▾ src

▾ main

▾ resources

> swagger

▾ templates

> JavaSpring

▾ typescript-fetch

≡ modelEnum.mustache

≡ modelGeneric.mustache

≡ modelOneOf.mustache

> target

m pom.xml

≡ modelOneOf.mustache ×

❗ Plugins supporting \*.mustache files found.

[Install Handlebars/Mustache plugin](#)

[Ignore extension](#)

✓ 8 ^ ▾

```
18
19
20 export function {{classname}}FromJSONTyped(json: any, ignoreDiscriminator: boolean): {{classname}} {
21     if ((json === undefined) || (json === null)) {
22         return json;
23     }
24     {{#discriminator}}
25     switch (json['{{discriminator.propertyName}}']) {
26     {{#discriminator.mappedModels}}
27         case '{{mappingName}}':
28             return {...{{modelName}}FromJSONTyped(json, true), {{discriminator.propertyName}}: '{{ma
29     {{/discriminator.mappedModels}}
30     default:
31         throw new Error(`No variant of {{classname}} exists with '{{discriminator.propertyName}}
32     }
33     {{/discriminator}}
34     {{^discriminator}}
35     return { {{#oneOf}}...{{.}}FromJSONTyped(json, true){{^-last}}, {{/-last}}{{/oneOf}} };
36     {{/discriminator}}
37 }
38
39 export function {{classname}}ToJSON(value?: {{classname}} | null): any {
40     <Не обращаем сюда внимания>
41 }
42 }
43
```

Project ▾

▾ contract

▾ src

▾ main

▾ resources

> swagger

▾ templates

> JavaSpring

▾ typescript-fetch

≡ modelEnum.mustache

≡ modelGeneric.mustache

≡ modelOneOf.mustache

> target

m pom.xml

≡ modelOneOf.mustache ×

❗ Plugins supporting \*.mustache files found.

[Install Handlebars/Mustache plugin](#)

[Ignore extension](#)

✓ 9 ^ ▾

```
33 {{/discriminator}}
34 {{^discriminator}}
35     return { {{#oneOf}}...{{.}}FromJSONTyped(json, true){{^-last}}, {{/-last}}{{/oneOf}} };
36 {{/discriminator}}
37 }
38
39 export function {{classname}}ToJSON(value?: {{classname}} | null): any {
40     if ((json === undefined) || (json === null)) {
41         return json;
42     }
43     {{#discriminator}}
44         switch (json['{{discriminator.propertyName}}']) {
45         {{#discriminator.mappedModels}}
46             case '{{mappingName}}':
47                 return {...{{modelName}}FromJSONTyped(json, true), {{discriminator.propertyName}}: '{{ma
48         {{/discriminator.mappedModels}}
49             default:
50                 throw new Error(`No variant of {{classname}} exists with '{{discriminator.propertyName}}
51         }
52     {{/discriminator}}
53     {{^discriminator}}
54         return { {{#oneOf}}...{{.}}FromJSONTyped(json, true){{^-last}}, {{/-last}}{{/oneOf}} };
55     {{/discriminator}}
56 }
57 }
58
```

Project ▾

▾ contract

▾ src

▾ main

▾ resources

> swagger

▾ templates

> JavaSpring

▾ typescript-fetch

≡ modelEnum.mustache

≡ modelGeneric.mustache

≡ modelOneOf.mustache

> target

m pom.xml

≡ modelOneOf.mustache ×

❗ Plugins supporting \*.mustache files found.

[Install Handlebars/Mustache plugin](#)

[Ignore extension](#)

✓ 9 ^ ▾

```
33 {{/discriminator}}
34 {{^discriminator}}
35     return { {{#oneOf}}...{{.}}FromJSONTyped(json, true){{^-last}}, {{/-last}}{{/oneOf}} };
36 {{/discriminator}}
37 }
38
39 export function {{classname}}ToJSON(value?: {{classname}} | null): any {
40     if ((json === undefined) || (json === null)) {
41         return json;
42     }
43     {{#discriminator}}
44         switch (json['{{discriminator.propertyName}}']) {
45         {{#discriminator.mappedModels}}
46             case '{{mappingName}}':
47                 return {...{{modelName}}FromJSONTyped(json, true), {{discriminator.propertyName}}: '{{ma
48         {{/discriminator.mappedModels}}
49         default:
50             throw new Error(`No variant of {{classname}} exists with '{{discriminator.propertyName}}
51         }
52     {{/discriminator}}
53     {{^discriminator}}
54         return { {{#oneOf}}...{{.}}FromJSONTyped(json, true){{^-last}}, {{/-last}}{{/oneOf}} };
55     {{/discriminator}}
56 }
57 }
58
```

Project ▾



▾ contract

▾ src

▾ main

▾ resources

&gt; swagger

▾ templates

&gt; JavaSpring

▾ typescript-fetch

≡ modelEnum.mustache

≡ modelGeneric.mustache

≡ modelOneOf.mustache

&gt; target

m pom.xml

≡ modelOneOf.mustache ×



❗ Plugins supporting \*.mustache files found.

[Install Handlebars/Mustache plugin](#)[Ignore extension](#)

✔ 9 ^ ▾

```
33 {{/discriminator}}
34 {{^discriminator}}
35     return { {{#oneOf}}...{{/.}}FromJSONTyped(json, true){{^-last}}, {{/-last}}{{/oneOf}} };
36 {{/discriminator}}
37 }
38
39 export function {{classname}}ToJSON(value?: {{classname}} | null): any {
40     if ((json === undefined) || (json === null)) {
41         return json;
42     }
43     {{#discriminator}}
44         switch (json['{{discriminator.propertyName}}']) {
45         {{#discriminator.mappedModels}}
46             case '{{mappingName}}':
47                 return {...{{modelName}}ToJSON(json, true), {{discriminator.propertyName}}: '{{mappingName}}'};
48         {{/discriminator.mappedModels}}
49         default:
50             throw new Error(`No variant of {{classname}} exists with '{{discriminator.propertyName}}'`)
51         }
52     {{/discriminator}}
53     {{^discriminator}}
54         return { {{#oneOf}}...{{/.}}FromJSONTyped(json, true){{^-last}}, {{/-last}}{{/oneOf}} };
55     {{/discriminator}}
56 }
57 }
58
```

```
export function AnimalToJSON(ignoreDiscriminator: boolean, value?: Animal | null): any {
  if (value === undefined) {
    return undefined;
  }
  if (value === null) {
    return null;
  }
  if (!ignoreDiscriminator) {
    if (json['type'] === 'DOG') {
      return DogToJSON(json, true);
    }
    if (json['type'] === 'WHALE') {
      return WhaleToJSON(json, true);
    }
  }
  return {
    'type': AnimalTypeToJSON(value.type),
    'id': value.id,
    'name': value.name,
  };
}
```

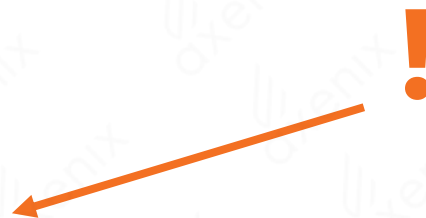
```
export function AnimalToJSON(ignoreDiscriminator: boolean, value?: Animal | null): any {  
  if (value === undefined) {  
    return undefined;  
  }  
  if (value === null) {  
    return null;  
  }  
  if (!ignoreDiscriminator) {  
    if (json['type'] === 'DOG') {  
      return DogToJSON(json, true);  
    }  
    if (json['type'] === 'WHALE') {  
      return WhaleToJSON(json, true);  
    }  
  }  
  return {  
    'type': AnimalTypeToJSON(value.type),  
    'id': value.id,  
    'name': value.name,  
  };  
}
```



```
export function AnimalToJSON(ignoreDiscriminator: boolean, value?: Animal | null): any {  
  if (value === undefined) {  
    return undefined;  
  }  
  if (value === null) {  
    return null;  
  }  
  if (!ignoreDiscriminator) {  
    if (json['type'] === 'DOG') {  
      return DogToJSON(json, true);  
    }  
    if (json['type'] === 'WHALE') {  
      return WhaleToJSON(json, true);  
    }  
  }  
  return {  
    'type': AnimalTypeToJSON(value.type),  
    'id': value.id,  
    'name': value.name,  
  };  
}
```



```
export function AnimalToJSON(ignoreDiscriminator: boolean, value?: Animal | null): any {  
  if (value === undefined) {  
    return undefined;  
  }  
  if (value === null) {  
    return null;  
  }  
  if (!ignoreDiscriminator) {  
    if (json['type'] === 'DOG') {  
      return DogToJSON(json, true);  
    }  
    if (json['type'] === 'WHALE') {  
      return WhaleToJSON(json, true);  
    }  
  }  
  return {  
    'type': AnimalTypeToJSON(value.type),  
    'id': value.id,  
    'name': value.name,  
  };  
}
```





# OpenAPI Выводы





# OpenAPI Выводы

1. OpenAPI генератор — мощнейший и кастомизируемый инструмент







# OpenAPI Выводы

1. OpenAPI генератор — мощнейший и кастомизируемый инструмент
2. Наличие уникальных правил вашего проекта не является поводом отказа от API First и генерации кода





# OpenAPI Выводы

1. OpenAPI генератор — мощнейший и кастомизируемый инструмент
2. Наличие уникальных правил вашего проекта не является поводом отказа от API First и генерации кода
3. Работа инструмента прозрачна и предсказуема







# OpenAPI Выводы

1. OpenAPI генератор — мощнейший и кастомизируемый инструмент
2. Наличие уникальных правил вашего проекта не является поводом отказа от API First и генерации кода
3. Работа инструмента прозрачна и предсказуема
4. Ускоряет разработку? Доволен ли Ваня?





# Оно действительно работает



# Оно действительно работает

---

1

Ускоряет процесс разработки: многие процессы могут идти параллельно.

Обеспечивает возможность демонстрации прототипа сервиса на ранних этапах разработки (моковый сервис).





# Оно действительно работает

---

1

Ускоряет процесс разработки: многие процессы могут идти параллельно.

2

Уменьшается риск ошибок, связанных с отличием моделей.



Обеспечивает возможность демонстрации прототипа сервиса на ранних этапах разработки (моковый сервис).

Особенно актуально при взаимодействии нескольких разработчиков или команд разработки.



# Оно действительно работает

1

Ускоряет процесс разработки: многие процессы могут идти параллельно.

2

Уменьшается риск ошибок, связанных с отличием моделей.

3

Улучшается качество проектирования API.



Обеспечивает возможность демонстрации прототипа сервиса на ранних этапах разработки (моковый сервис).

Особенно актуально при взаимодействии нескольких разработчиков или команд разработки.

Улучшается качество не только проектирования API, но и модели данных.



# Оно действительно работает

1

Ускоряет процесс разработки: многие процессы могут идти параллельно.

2

Уменьшается риск ошибок, связанных с отличием моделей.

3

Улучшается качество проектирования API.

4

Появляются возможности генерации кода по спецификации API.



Обеспечивает возможность демонстрации прототипа сервиса на ранних этапах разработки (моковый сервис).

Особенно актуально при взаимодействии нескольких разработчиков или команд разработки.

Улучшается качество не только проектирования API, но и модели данных.

## OpenAPI наблюдения

- Super-skilled аналитик VS вечно занятой бэкендер
- Не всем хватит спеки в OpenAPI
- Может потребоваться адаптация под особенности проекта





# Спасибо!

О компании ООО «Аксеникс Инновации» – российская компания, входит в группу «AXENIX», резидент Инновационного центра «Сколково», ведущая научно-технические исследования в области информационных технологий и предоставляющая на базе собственных разработок широкий спектр профессиональных услуг, направленных на цифровизацию бизнеса.

Подробнее на [axenix-innovation.pro](https://axenix-innovation.pro)

