

# Создание документации MkDocs из OpenAPI схемы с помощью openapi-genetator

[openapi-genetator](#) (форк swagger-codegen) позволяет автоматически создавать клиентские библиотеки API (генерация SDK), документацию и различные конфигурационные файлы с помощью файла спецификации OpenAPI (поддерживаются как версия 2.0, так и 3.0). В настоящее время поддерживаются [следующие](#) языки/фреймворки.

Нас же интересует создание документации в формате Markdown.

Рассмотрим создание документации для проекта `warriors_project`.

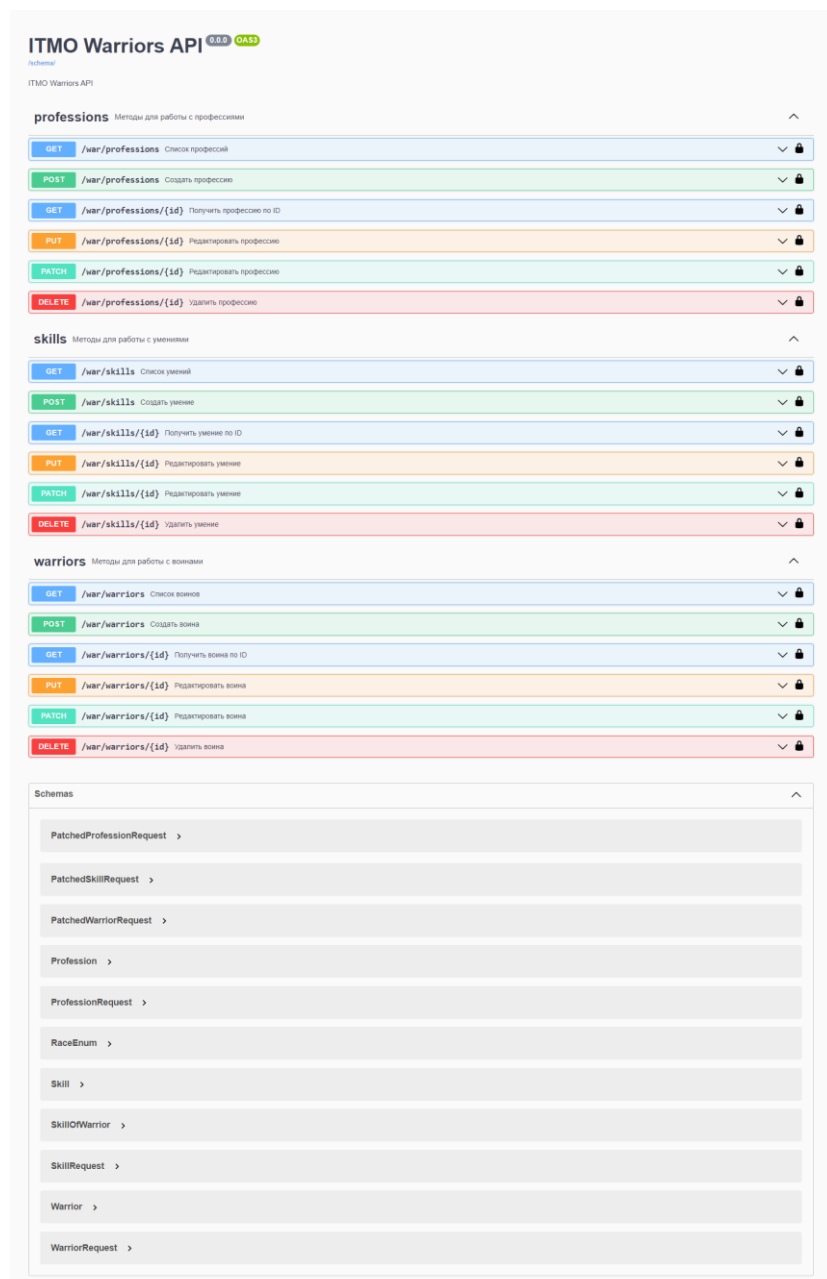


Рисунок 1 - текущая документация в Swagger

Генератор поставляется в качестве CLI приложения. Для запуска необходима Java.

Скачаем JAR пакет по [ссылке](#). На момент написания данного руководства, актуальная версия – [5.3.1](#).

Переместим скачанный файл в директорию с mkdocs и откроем терминал в данной папке.

```
(warriors_project)
Georgy@GEORGY-MSI MINGW64 ~/PycharmProjects/warriors_project
$ mkdocs new docs
INFO      - Creating project directory: docs
INFO      - Writing config file: docs\mkdocs.yml
INFO      - Writing initial docs: docs\docs\index.md
(warriors_project)
Georgy@GEORGY-MSI MINGW64 ~/PycharmProjects/warriors_project
$ cd docs
(warriors_project)
Georgy@GEORGY-MSI MINGW64 ~/PycharmProjects/warriors_project/docs
$ mv ~/Downloads/openapi-generator-cli-5.3.1.jar .
(warriors_project)
Georgy@GEORGY-MSI MINGW64 ~/PycharmProjects/warriors_project/docs
$ ls
docs/  mkdocs.yml  openapi-generator-cli-5.3.1.jar
(warriors_project)
Georgy@GEORGY-MSI MINGW64 ~/PycharmProjects/warriors_project/docs
$ |
```

Рисунок 2 - подготовка окружения

Запустим генератор со следующими параметрами (проект должен быть запущен):

```
java -jar openapi-generator-cli-5.3.1.jar generate -i <локальный
или сетевой путь к файлу схемы> -g markdown -o docs
```

generate – запуск генерации

-i <путь к файлу> – путь или ссылка на файл OpenAPI схемы в формате JSON. См.

-g markdown – выбран генератор *Markdown* документации

-o docs – сохранить документацию в папку *docs*

В случае использования *drf\_yasg* (OpenAPI 2.0) ссылкой будет являться путь указанный для *schema\_view.without\_ui*.

В случае использования *drf-spectacular* (OpenAPI 3.0) ссылкой будет являться путь указанный для *SpectacularAPIView*.

```

20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('schema/', SpectacularAPIView.as_view(), name='schema'),
23     path('swagger/', SpectacularSwaggerView.as_view(url_name='schema'), name='swagger-ui'),
24     path('war/', include('warriors_app.urls')),
25 ]

```

Рисунок 3 - путь к API схеме: <http://127.0.0.1:8000/schema/>

В моем случае команда выглядит так:

```
java -jar openapi-generator-cli-5.3.1.jar generate -i
http://127.0.0.1:8000/schema/ -g markdown -o docs
```

```

(warriors_project)
Georgy@GEORGY-MSI MINGW64 ~/PycharmProjects/warriors_project/docs
$ java -jar openapi-generator-cli-5.3.1.jar generate -i http://127.0.0.1:8000/schema/ -g markdown -o docs
[main] INFO o.o.codegen.DefaultGenerator - Generating with dryRun=false
[main] INFO o.o.c.ignore.CodegenIgnoreProcessor - No .openapi-generator-ignore file found.
[main] INFO o.o.codegen.DefaultGenerator - OpenAPI Generator: markdown (documentation)
[main] INFO o.o.codegen.DefaultGenerator - Generator 'markdown' is considered beta.
[main] INFO o.o.codegen.TemplateManager - writing file C:\Users\Georgy\PycharmProjects\warriors_project\docs\docs\README.md
[main] INFO o.o.codegen.TemplateManager - writing file C:\Users\Georgy\PycharmProjects\warriors_project\docs\docs\.openapi-generator-ignore
[main] INFO o.o.codegen.TemplateManager - writing file C:\Users\Georgy\PycharmProjects\warriors_project\docs\docs\.openapi-generator\VERSION
[main] INFO o.o.codegen.TemplateManager - writing file C:\Users\Georgy\PycharmProjects\warriors_project\docs\docs\.openapi-generator\FILES
#####
# Thanks for using OpenAPI Generator.                                     #
# Please consider donation to help us maintain this project ?           #
# https://opencollective.com/openapi_generator/donate                     #
#####
(warriors_project)
Georgy@GEORGY-MSI MINGW64 ~/PycharmProjects/warriors_project/docs
$ |

```

Рисунок 4 - выполнение генерации (часть вывода опущена)

Папка с mkdocs теперь выглядит так:

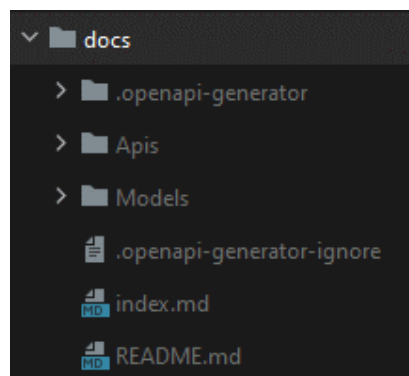


Рисунок 5 - сгенерированные файлы

*.openapi-generator* и *.openapi-generator-ignore* – служебные файлы генератора, можно удалить.

*README.md* – файл главной страницы.

*index.md*, созданный командой *mkdocs new*, **нужно удалить** (в противном случае, файл *README.md* будет проигнорирован).

*Apis* – эндпоинты проекта.

*Models* – модели проекта.

Теперь наша документация выглядит так:

My Docs

Documentation for ITMO Warriors API

Apis

Models

Q Search

Previous

Next

Documentation for ITMO Warriors API

Documentation for API Endpoints

Documentation for Models

Documentation for Authorization

Documentation for ITMO Warriors API

Documentation for API Endpoints

All URLs are relative to `http://localhost`

Class	Method	HTTP request	Description
ProfessionsApi	professionsCreate	POST /war/professions	Создать профессию
ProfessionsApi	professionsDestroy	DELETE /war/professions/{id}	Удалить профессию
ProfessionsApi	professionsList	GET /war/professions	Список профессий
ProfessionsApi	professionsPartialUpdate	PATCH /war/professions/{id}	Редактировать профессию
ProfessionsApi	professionsRetrieve	GET /war/professions/{id}	Получить профессию по ID
ProfessionsApi	professionsUpdate	PUT /war/professions/{id}	Редактировать профессию
SkillsApi	skillsCreate	POST /war/skills	Создать умение
SkillsApi	skillsDestroy	DELETE /war/skills/{id}	Удалить умение
SkillsApi	skillsList	GET /war/skills	Список умений
SkillsApi	skillsPartialUpdate	PATCH /war/skills/{id}	Редактировать умение
SkillsApi	skillsRetrieve	GET /war/skills/{id}	Получить умение по ID
SkillsApi	skillsUpdate	PUT /war/skills/{id}	Редактировать умение

Рисунок 6 - документация в формате Markdown