МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Национальный исследовательский Томский государственный университет
Томский государственный университет систем управления и радиоэлектроники
Болгарская Академия наук
ООО «Научно исследовательское предприятие «Лазерные технологии»

ИННОВАТИКА-2019

СБОРНИК МАТЕРИАЛОВ

XV Международной школы-конференции студентов, аспирантов и молодых ученых
25–27 апреля 2019 г.
г. Томск, Россия

Под редакцией А.Н. Солдатова, С.Л. Минькова

Scientific & Technical Translations
STT
ИЗДАТЕЛЬСТВО
Томск – 2019

ГЕНЕРАЦИЯ API-КЛИЕНТОВ НА ОСНОВЕ OPENAPI

А.Д. Никифоров, С.И. Самохина
Национальный исследовательский Томский государственный университет
nikart28@gmail.com

OPENAPI BASED API-CLIENTS GENERATION

A.D. Nikiforov, S.I. Samohina
National Research Tomsk State University

The relevance of compositing in the automation of generating SDK libraries for interacting with third-party REST API, which will be distributed to develop new services. Decreased developer time to do the same type of work. There are more opportunities to test the newly created API.

Keywords: swagger, OpenAPI, SDK, API, QA, web, development

В современном web-программировании отдельное место заняла разработка API (Application Programming Interface) – это язык, регламентированный способ общения одной компьютерной программы с другой для исполнения какой-нибудь общей задачи, когда одна программа выполняет запросы другой [1]. Если говорить о web-программировании, то также к аббревиатуре API необходимо добавить аббревиатуру REST (Representational State Transfer) – это архитектурное решение взаимодействия компонентов информационной системы расположенных в сети Интернет. REST определяет ряд правил, которые должны соблюдаться при проектировании программного продукта [4].

В процессе разработки новых интернет-приложений программисту очень часто приходится обращаться к чужим API, использовать их интерфейс и внедрять в свою программу. При интеграции с платформой, предлагающей API REST, разработчик может использовать сгенерированную или написанную человеком специальную библиотеку – SDK, также у программиста есть возможность самому писать HTTP-запросы для взаимодействия с чужим API.

Выбирая из этих двух вариантов, с уверенностью можно сказать, что первый – наиболее удобный, так как необходимо только обратиться к уже существующим методам и классам, чтобы вызвать функцию определенного API. Второй вариант тоже используется при отсутствии уже сделанных библиотек или же при незнании популярных языков программирования, на которых обычно и делаются API-клиенты [2].

Как же сделать библиотеку или SDK, которую можно будет подключить в свой проект?

Существует несколько путей решения этой проблемы, например, создание библиотеки с необходимыми классами, методами, вручную – то есть за всю специфику подключения к API и генерации всех классов отвечает программист-разработчик. Этот подход имеет ряд преимуществ, самое главное из которых – это, конечно, то, что человек может учесть все нюансы уже созданного API и реализовать их в SDK, которая будет распространятся для разработки новых сервисов. Но в таком подходе есть и большой минус – время, затрачиваемое на разработку таких пакетов.

Для того чтобы не заниматься одними и теми же действиями, были придуманы генераторы кода, которые на основе документации к API, могут сгенерировать код на выбранном языке программирования, с помощью которого можно работать с существующим API.

Таким образом, имея в арсенале код-генератор и правильно сгенерированную документацию API можно создать SDK, который можно свободно распространять и облегчать работу со своим интерфейсом другим разработчикам. Но возникает один вопрос, где взять документацию API, как сформулировать общий регламент оформления и сделать его доступным для большинства код-генераторов? На помощь в этой ситуации пришла спецификация OpenAPI.

Спецификация OpenAPI – это формат описания API или язык определения API. В принципе, файл спецификации OpenAPI позволяет описывать API, включая (среди прочего): общую информацию об API, доступные пути (/ресурсы), доступные операции по каждому пути (get / resources), вход / выход для каждой операции.

Спецификацию Open API Specification можно найти в репозитории github Open API Initiative. В этом документе описываются все аспекты спецификации Open API. Использование языка определения API, такого как спецификация OpenAPI, позволяет легко и быстро описать API. Это особенно полезно, когда идёт процесс разработки API. Будучи простым текстовым файлом, файл спецификации OpenAPI можно совместно использовать и управлять в любом VCS. После написания файла спецификации OpenAPI можно также использовать как: исходный материал для документации, спецификация для разработчиков, частичное или полное генерирование кода [3].

Файл спецификации Open API может быть написан либо в JSON, либо

в YAML (рис. 1).

Рис. 1. Пример описания API по спецификации OpenAPI в формате JSON

После того, как готова документация, из неё можно генерировать SDK, который потом будет использоваться разработчиками для создания API. Вот несколько примеров программ, разработанных специально для автоматического создания SDK: APIMATIC, Restlet Studio, Swagger Codegen, REST United, Autorest.

У каждой из этих программ есть свои особенности предоставления инструментов для генерации кода, но суть у всех одна и та же. Для примера возьмём программу Autorest – это инструмент для создания клиентских библиотек с использованием спецификационного файла, который

описывает REST API. Используя Autorest, есть возможность создавать клиентские библиотеки для следующих языков: C#, NodeJS, Python, Java, Ruby, Go.

Теперь необходимо взять OpenAPI файл и создать клиентскую библиотеку. Для этого необходимо предоставить autorest входной файл, выходную папку и язык, на котором необходимо создать клиентскую библиотеку. Например, на рисунке 2 показано как из командной строки windows можно начать генерацию на языке C#.

```
autorest --input-file=swagger.json --output-folder=generated_csharp
--csharp
```

Рис.2. Команда Autorest для создания кода

Аналогичным образом можно было бы генерировать клиентскую библиотеку для любого другого языка, заменяя --csharp, например, --python, --go или другим поддерживаемым языком. Если служба защищена и, например, требуется токен OAuth, необходимо также добавить команду --add-credentials. Это создаст несколько разные конструкторы для принятия клиентом учетных данных.

После проделанных манипуляций Autorest создаст файлы с кодом на выбранном языке, в которых будут обращения к API в соответствии с описанной документацией OpenAPI. Когда есть сгенерированный код, необходимо создать пользовательскую библиотеку, которую можно было бы подключить к проекту, в котором создаётся API.

В данный момент Autorest не создаёт файл csproj вместе с генерированным кодом, поэтому нужно вручную создать проект в Visual Studio типа class library с использованием технологии .net framework. После этого нужно добавить сгенерированные классы в созданный проект, после чего собрать его. Теперь в папке bin-проекта есть dll, которую можно добавить в nuget-пакет, выложить его на хостинг nuget.org. После этих действий сгенерированную библиотеку можно добавлять в любой проект.

Автоматическая генерация SDK достаточно сложная процедура; есть много вещей, которые необходимо учитывать при попытке автоматически генерировать код. Отсутствующие данные, такие как определения моделей, коды ошибок и информация аутентификации, могут привести к тому, что автоматически сгенерированный код будет иметь очень низкое качество, а в некоторых случаях код не будет работать. Дизайн API также

является основным фактором при попытке автоматически генерировать код. Нет единого, широко принятого, стандартизованного в отрасли стандарта проектирования для API. Не говоря уже о том, что некоторые API разработаны хорошо, а другие хуже, возвращая плохо отформатированный JSON, что приводит к неожиданным результатам.

Литература

1. Tilda API [Электронный ресурс]. – URL: http://help-ru.tilda.ws/api (дата обращения 11.03.2019)
2. API [Электронный ресурс]. – URL: https://ru.wikipedia.org/wiki/API (дата обращения 11.03.2019)
3. Swagger [Электронный ресурс]. – URL: https://swagger.io/docs/specification/about/ (дата обращения: 11.03.2019).
4. Mark Masse. REST API Design Rulebook. Published by O'Reilly Media, Inc., 2012. – 94 с.