

ОГЛАВЛЕНИЕ

	стр.
ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ.....	2
ВВЕДЕНИЕ.....	2
1. ОБЗОРНАЯ ЧАСТЬ.....	4
1.1. Описание предметной области.....	4
1.1.1. Описание серверной архитектуры.....	4
1.1.2. Описание модели внедрения программных продуктов Agile Scrum	6
1.1.3. Схема ручного сопровождения API-документации	6
1.2. Исследование существующих технологий	6
1.2.1. Перечень функций, подлежащих автоматизации	6
1.2.2. Выбор и обоснование критериев качества	6
1.2.3. Анализ аналогов и прототипов	7
1.2.4. Сравнение аналогов и прототипов	7
2. РАСЧЕТНО-КОНСТРУКТОРСКАЯ ЧАСТЬ	7
2.1. Определение требований к системе	7
2.2. Разработка структуры автоматизированной системы	7
2.3. Разработка структуры интерфейса взаимодействия пользователя с	
системой.....	8
2.4. Разработка алгоритмов программных модулей	8
2.5. Разработка плана проведения тестирования.....	8
3. ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ	8
3.1. Реализация разработанных алгоритмов	8
3.2. Тестирование и отладка системы.....	8
3.3. Руководство пользователя	8
ЗАКЛЮЧЕНИЕ	8
СПИСОК ЛИТЕРАТУРЫ.....	9

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

1. REST API – это набор правил, по которым следует обращаться к серверу для отправки или получения данных.
2. Пользователь – **живой человек** (определение человека по Платону: «двуногое без перьев»)
3. Клиент – любое приложение которое делает запросы на сервер. Например, в роли клиента может выступать веб браузер, когда пользователь открывает веб-сайт.
4. АС – автоматизированная система
5. ПО – Программное обеспечение
6. API-документация – это техническая документация, в которой фиксируются инструкции о том, как использовать программное API.
7. UI – пользовательский интерфейс.
8. База данных (БД) – это совокупность систематизированных особым образом данных, находящаяся в памяти вычислительной системы. Для работы с БД используются специальные средства – системы управления базами данных (СУБД).

ВВЕДЕНИЕ

На сегодняшний день большинство крупных IT компаний для взаимодействия сервера и клиента используют REST API [1].

Компании вроде Яндекса, Google и т.п. Предоставляют открытые API методы своих сервисов чтобы разработчики могли интегрироваться с ними.

Например, при получении данных о пользователе, информация о котором храниться в БД (базе данных) на сервере необходимо указать путь до сервера (URI), идентификатор пользователя (ID) и метод (Method) по которому сервер поймет, что нужно сделать с ресурсом, в данном случае вернуть информацию

о пользователе. Данный процесс «общения» клиента и сервера, представлен на рисунке 1.1.

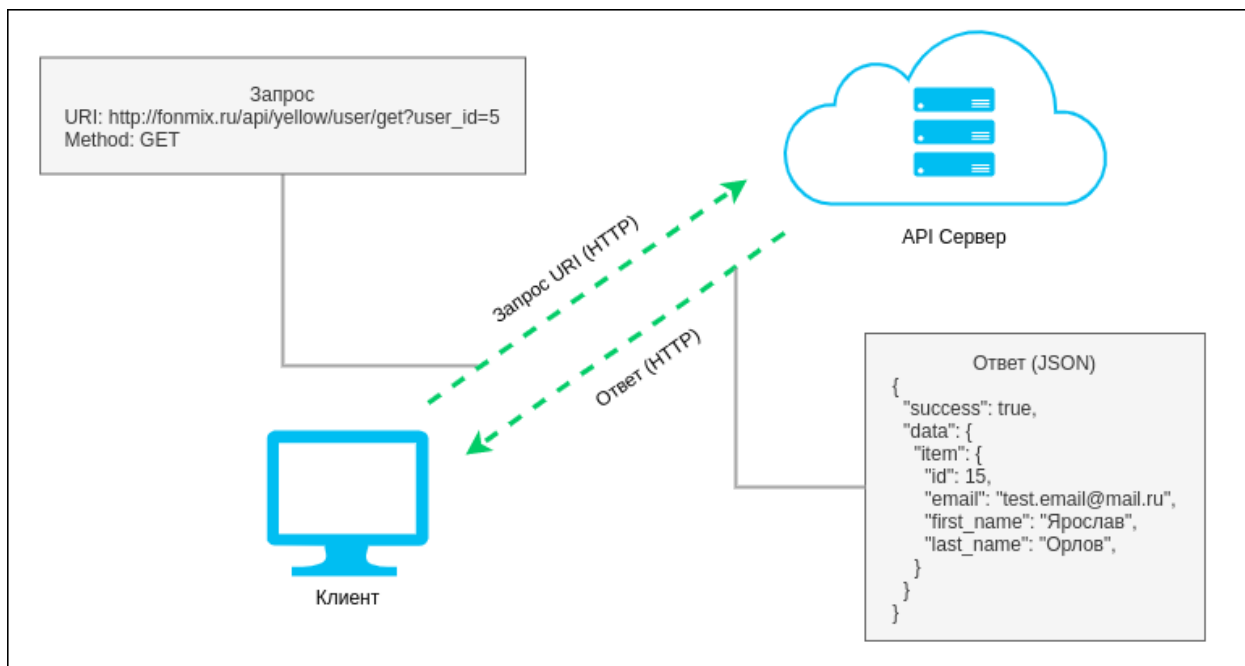


Рисунок 1.1 – Схема получения информации о пользователе

Помимо написания самих API методов необходимо написание подробной документации по ним, поскольку без нее попросту не удастся воспользоваться методом. А также не менее важно поддерживать документацию в актуальном состоянии поскольку если документация будет неправильная или устаревшая, то велика вероятность ошибок и в конечном итоге может сказываться на качестве и стоимости продуктов. Поэтому написание API-документации очень важная и актуальная тема.

API-документация представляет собой....

Что такое API-документация?

Компания ООО «ФорМакс» разрабатывает продукт Fonmix, серверная часть которого полностью базируется на технологии REST API, т.е. общение любого пользователя с сервером Fonmix осуществляется через REST API.

Основными клиентами [3] для сервера Fonmix являются:

- 1) Веб-сайт fonmix.ru – представляет собой веб интерфейс, в котором пользователи [2] могут управлять музыкой в своих заведениях: создавать плейлисты, составлять музыкальное расписание, добавлять рекламу в перерывах между песнями и т.п.
- 2) FM.Player – кроссплатформенный медиапроигрыватель разрабатываемый также в компании ООО «ФорМакс», с помощью которого воспроизводится медиа контент правообладателей.
- 3) Правообладатель – это исполнитель и изготовитель фонограмм, с которым заключается договор о дистрибуции контента и предоставлении отчетов об использовании.

Целью данной работы является создание системы автоматического сопровождения API-документации, позволяющей ускорить и повысить качество разработки. В соответствии с поставленной целью, работа над АС (автоматизированной системой) была разделена на несколько этапов, в рамках которых решались следующие задачи:

- анализ предметной области
- обзор и сравнение современных технологий по сопровождению API-документации
- выделение перечня функций, подлежащих автоматизации
- выфвыф

1. ОБЗОРНАЯ ЧАСТЬ

1.1. Описание предметной области

1.1.1. Описание серверной архитектуры

Серверная часть проекта Fonmix на разделена на микросервисы.

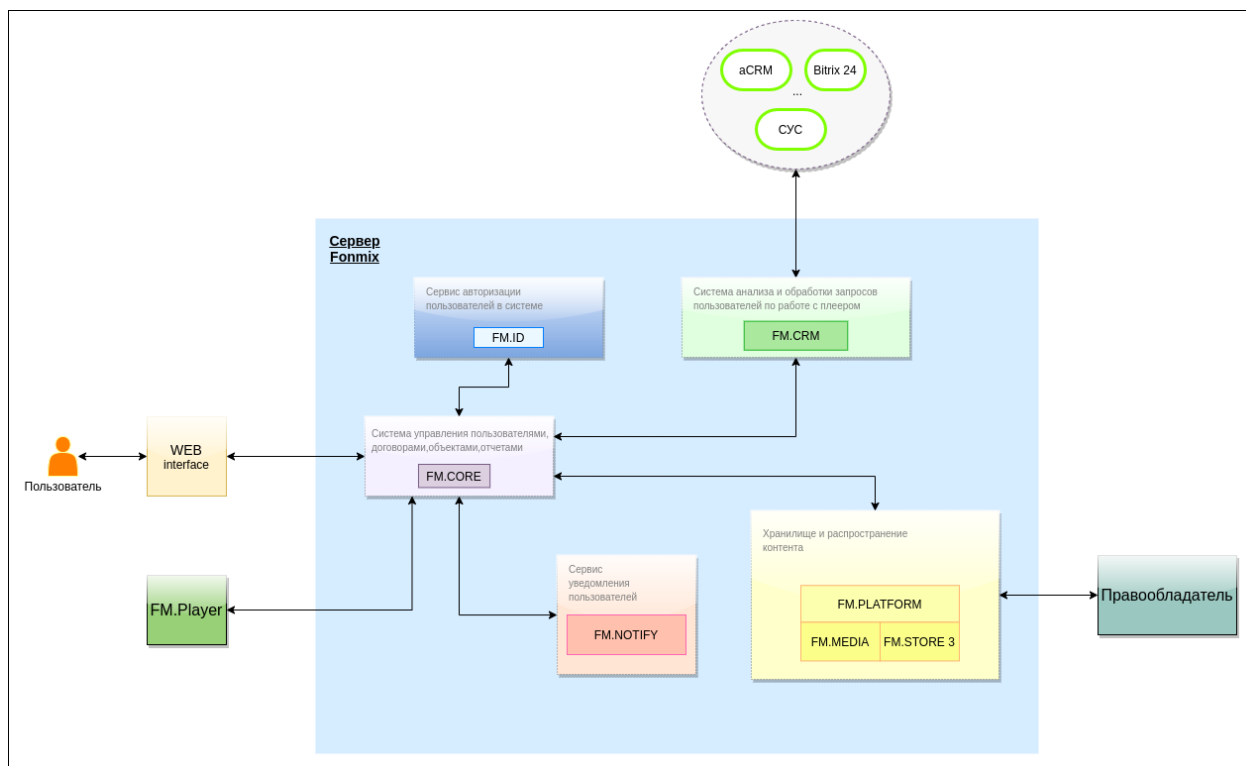
Микросервисная архитектура¹ – вариант сервис-ориентированной архитектуры программного обеспечения, направленный на взаимодействие насколько это возможно небольших, слабо связанных и легко изменяемых модулей – микросервисов.

Основными микросервисами являются:

- FM.Core – Основное ядро. Количество API методов 253
- FM.CRM – выф. Количество API методов 153
- FM.ID – dsa. Количество API методов 23
- FM.Notify – dsa. Количество API методов 34
- FM.Store – dsa. Количество API методов 15
- FM.Media – dsa. Количество API методов 36

Итого, общее количество методов 514

Схема взаимодействия клиентов и сервера представлена на рисунке 1.2.



¹ https://ru.wikipedia.org/wiki/Микросервисная_архитектура

Рисунок 1.2 – Схема взаимодействие клиентов и сервера

Исходя из этого можно сделать вывод что сопровождение API-документации в ручном режиме может занимать очень много времени.

1.1.2. Описание модели внедрения программных продуктов Agile Scrum

Текст

1.1.3. Схема ручного сопровождения API-документации

Текст

1.2. Исследование существующих технологий

1.2.1. Перечень функций, подлежащих автоматизации

Текст

1.2.2. Выбор и обоснование критериев качества

Для проведения сравнительного анализа аналогов и прототипов выбраны следующие критерии:

- 1) Публикация в confluence
- 2) Трудозатраты на изучение технологии
- 3) Потребность в дополнительном ПО
- 4) Настраиваемость системы
- 5) Кроссплатформенность системы

6)

7) живые примеры запросов/ответов

8) Интеграция с другими продуктами ?

1.2.3. Анализ аналогов и прототипов

Рассмотрим аналоги и прототипы с точки зрения выбранных критериев качества.

1.2.3.1. Swagger

Swagger представляет собой

1.2.4. Сравнение аналогов и прототипов

Текст

2. РАСЧЕТНО-КОНСТРУКТОРСКАЯ ЧАСТЬ

2.1. Определение требований к системе

Текст

2.2. Разработка структуры автоматизированной системы

Текст

2.3. Разработка структуры интерфейса взаимодействия пользователя с системой

Текст

2.4. Разработка алгоритмов программных модулей

Текст

2.5. Разработка плана проведения тестирования

Текст

3. ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ

3.1. Реализация разработанных алгоритмов

Текст

3.2. Тестирование и отладка системы

Текст

3.3. Руководство пользователя

Текст

ЗАКЛЮЧЕНИЕ

Что в итоге получилось.

СПИСОК ЛИТЕРАТУРЫ

1. Python Documentation – [Электронный ресурс]:
<https://docs.python.org/3/>
(Дата обращения: 28.09.2020)
2. Bootstrap Documentation – [Электронный ресурс]:
<https://getbootstrap.com/docs/3.3/> (Дата обращения: 28.09.2020)
3. PostgreSQL Database Documentation – [Электронный ресурс]:
<https://www.postgresql.org/docs/> (Дата обращения: 28.09.2020)
4. Скотт Б., Нейл Т. Проектирование веб-интерфейсов. – СПб.: Символ-Плюс, 2010. – 352 с.