

Доклад на тему:

Программные интерфейсы взаимодействия

Подготовил: Кравчук Дмитрий Вадимович

Команда аналитики: Мой МТС

Содержание

- Что такое синхронность и асинхронность
- Преимущества синхронности и асинхронности и способы реализации
- Программные интерфейсы взаимодействия REST/SOAP/Websocket
- Валидация данных



Глава 1

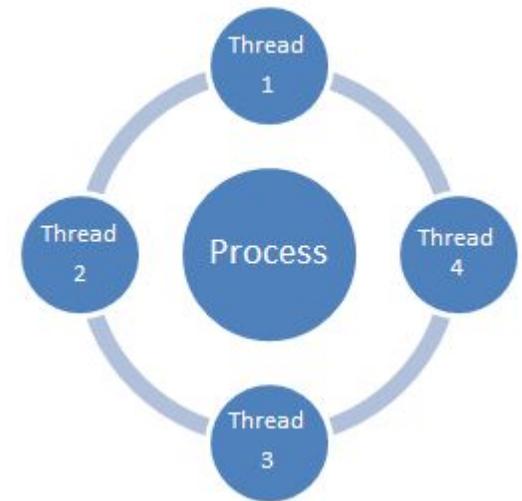
Синхронность и асинхронность

Отличие процесса от потока

Процесс - это программа запущенная в оперативной памяти компьютера (набор инструкций).

Свойства процесса: имеет выделенную оперативную память, адресное пространство, контекст.

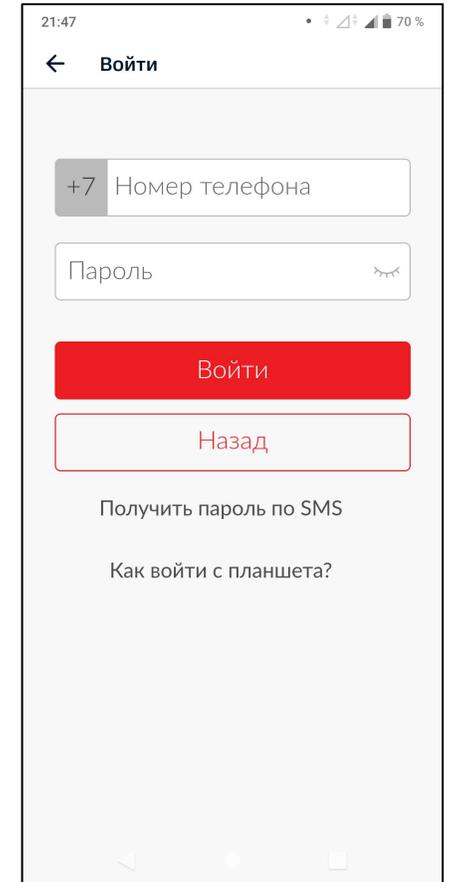
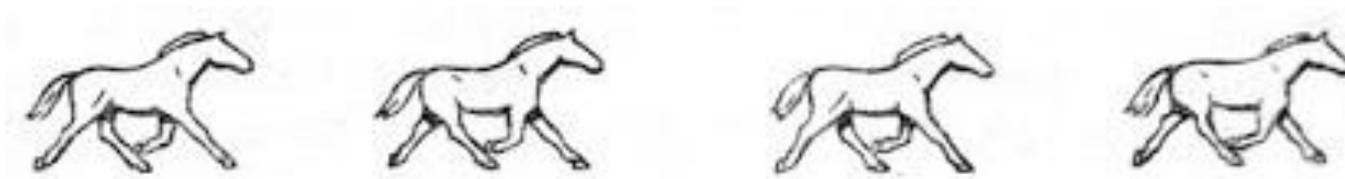
Поток - сущность ОС, процесс выполнения на процессоре набора инструкций (программного кода)



Синхронность

Синхронность – последовательное выполнение кода программы. Как правило, все действия выполняются в одном потоке и в одном процессе.

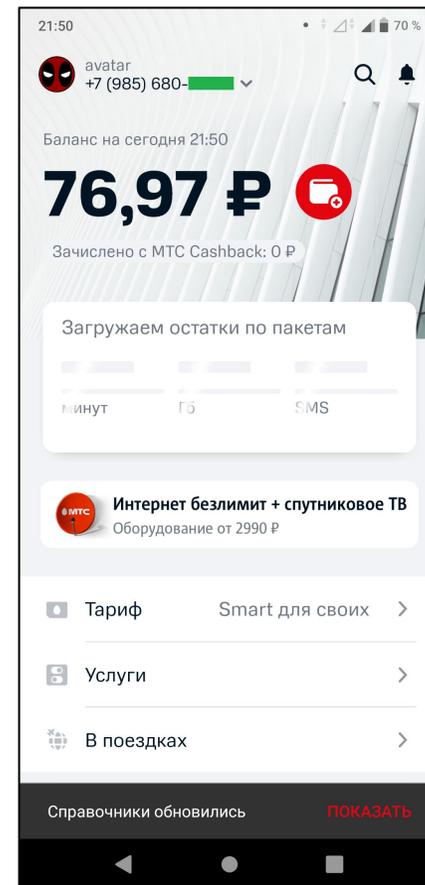
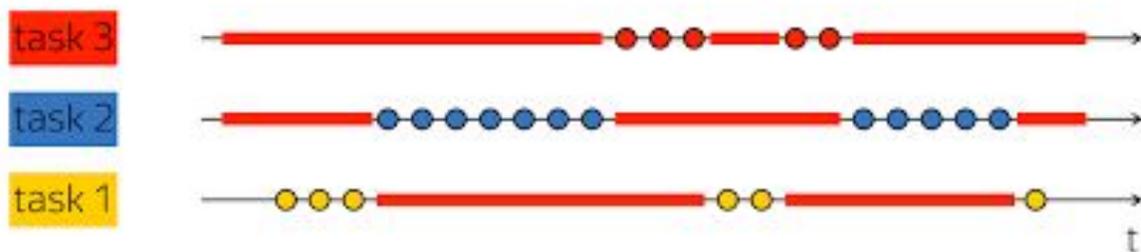
Пример: авторизация в приложении Мой МТС (на FE).
Метод 1.2 User token



Асинхронность

Асинхронность – не совпадение чего-либо во времени.
В ИТ – выполнение кода в нескольких потоках, процессах.

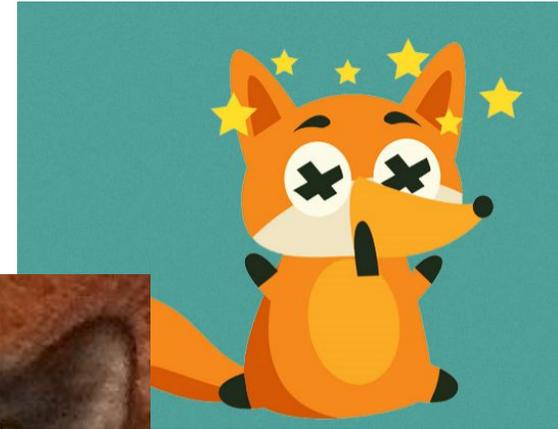
Пример: авторизация в приложении Мой МТС (на FE).
Метод 2.7 Получение счетчиков для главного экрана,
2.9 Баланс (Foris) и др.



Асинхронность

Способы реализации асинхронности:

- 1) Callback (функция обратного вызова)
- 2) Корутины
- 3) Futures (футуры)
- 4) Promises (обещания)



Преимущества и недостатки sync и async

SYNC (Синхронное API)

Преимущества:

- 1) Легко писать и читать код

Недостатки:

- 1) Неэффективное использование ресурсов
- 2) Блокировка UI

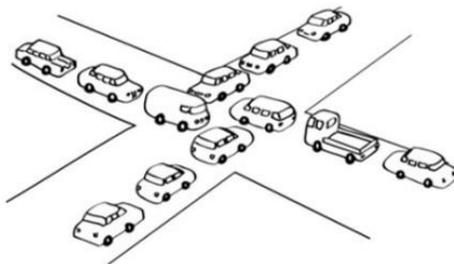
ASYNC (Асинхронное API)

Преимущества:

- 1) Эффективное использование ресурсов
- 2) Не блокируется UI

Недостатки:

- 1) Тяжело читать и писать код
- 2) Deadlock – взаимная блокировка



Глава 2

Программные интерфейсы взаимодействия



SOAP

SOAP (Simple object access protocol) – используется для обмена структурированными сообщениями в формате xml.

XML (eXtensible markup language) – расширяемый язык разметки
XSD (XML scheme definition) – язык описания структуры XML документа.

Пример SOAP: Foris IL





Пример XML и XSD

XML - документ

```
<?xml version="1.0" encoding="utf-8"?>
<country>
  <country_name>France</country_name>
  <population>59.7</population>
</country>
```

XSD

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="country">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="country_name" type="xs:string"/>
        <xs:element name="population" type="xs:decimal"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Из чего состоит SOAP-XML

1. Envelope (конверт) – корневой элемент
2. Header (заголовок) – содержит атрибуты сообщения
3. Body (тело) – содержит сообщение
4. Fault – информация об ошибках, которые произошли при обработке сообщений.

WSDL – язык описания веб-сервисов

```
<?xml version="1.0" encoding="UTF-8" ?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:head="http://schemas.sitel.ru/FORIS/IL/Head" ?>
  <soapenv:Header>
    <head:SubscriberInformation ?>
      <!--Optional-->
      <head:MessageId ?>/head:MessageId
      <!--Optional-->
      <head:ReplyToURL ?>/head:ReplyToURL
      <!--Optional-->
      <head:SubscriberName ?>/head:SubscriberName
      <!--Optional-->
      <head:Login ?>/head:Login
      <!--Optional-->
      <head:OperatorType ?>/head:OperatorType
      <!--Optional-->
      <head:ParentMessageId ?>/head:ParentMessageId
      <!--Optional-->
      <head:BusinessProcessCode ?>/head:BusinessProcessCode
    </head:SubscriberInformation>
  </soapenv:Header>
  <soapenv:Body>
    <res:GetCounters ?>
      <!--Optional-->
      <res:request ?>
        <!--Optional-->
        <res:Language ?>
          <!--Optional-->
          <sieb:Action ?>/sieb:Action
          <!--Optional-->
          <sieb:Id ?>
            <!--Optional-->
            <sieb:ForisId ?>/sieb:ForisId
            <!--Optional-->
            <sieb:ForisLocalId ?>/sieb:ForisLocalId
          </sieb:Id>
          <!--Optional-->
          <sieb:Name ?>/sieb:Name
          <!--Optional-->
          <sieb:GlobalCode ?>/sieb:GlobalCode
          <!--Optional-->
          <sieb:RegionCode ?>/sieb:RegionCode
        </res:Language>
        <!--Optional-->
        <res:Msisdn ?>/res:Msisdn
      </res:request>
    </res:GetCounters>
  </soapenv:Body>
</soapenv:Envelope>
```



Пример WSDL

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="http://tempuri.org/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:wsx="http://schemas.xmlsoap.org/ws/2004/09
  <wsdl:import namespace="http://schemas.sitels.ru/FORIS/IL/ResourceInformation" location="ResourceInformation_1.wsdl"/>
  <wsdl:types/>
  <wsdl:binding name="FIGA.BasicHttpBinding_IResourceInformation" type="i0:IResourceInformation">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="GetCurrentCreditLimit">
      <soap:operation soapAction="http://schemas.sitels.ru/FORIS/IL/ResourceInformation/IResourceInformation/GetCurrentCreditLimit" style="document"/>
      <wsdl:input>
        <soap:header message="i0:IResourceInformation_GetCurrentCreditLimit_InputMessage_Headers" part="SubscriberInformation" use="literal"/>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetTDForClosureByDate">
      <soap:operation soapAction="http://schemas.sitels.ru/FORIS/IL/ResourceInformation/IResourceInformation/GetTDForClosureByDate" style="document"/>
      <wsdl:input>
        <soap:header message="i0:IResourceInformation_GetTDForClosureByDate_InputMessage_Headers" part="SubscriberInformation" use="literal"/>
        <soap:body use="literal"/>
      </wsdl:input>
```

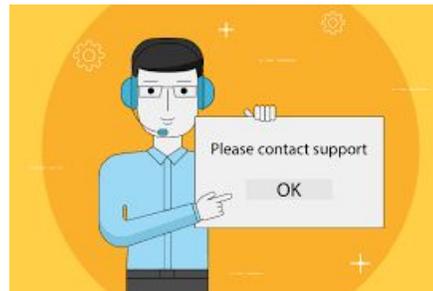
REST

RESTful (REpresentational State Transfer) – архитектурный стиль взаимодействия компонентов распределённого приложения в сети.

WADL (Web application description language) - машинно-читаемое XML-описание для web-приложений HTTP (REST).

Swagger/openAPI - формализованная спецификация в виде API

Пример REST API: EPI, WebSSO

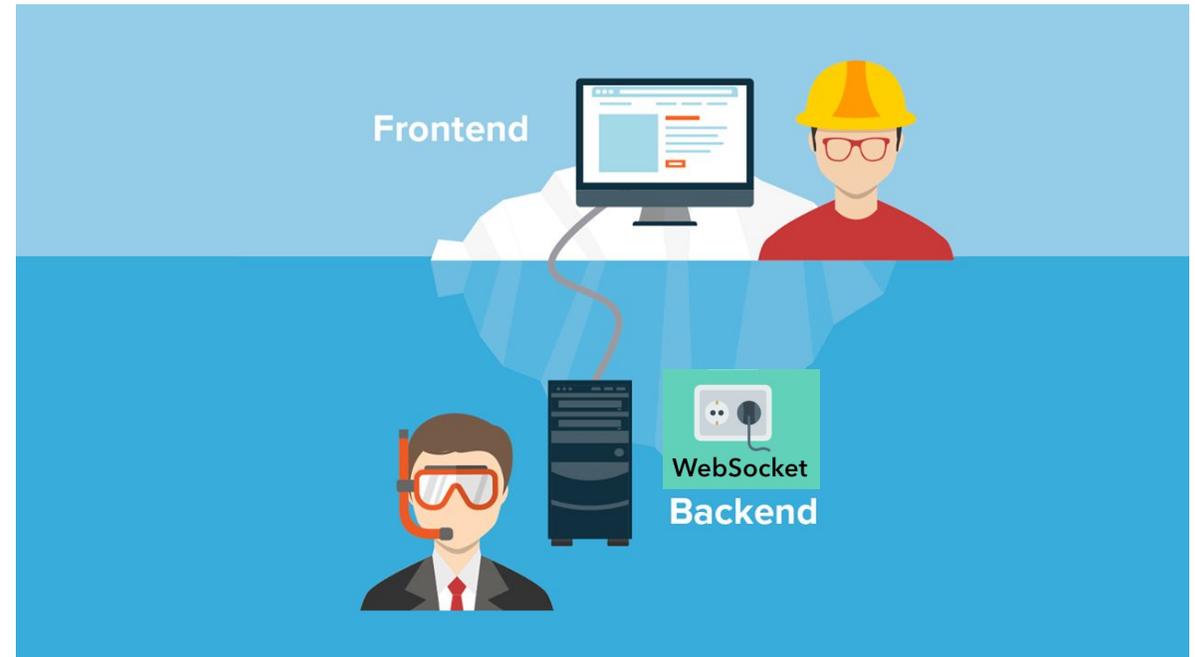


```
openapi: "3.0.0"
info:
  title: EPI REST API
  description: "Единый рекламный интерфейс (EPI)"
  version: "1.0.0"
tags:
- name: "banner"
  description: "Работа с баннером"
- name: "recommendations"
  description: "Рекомендации от RTDM"
- name: "event"
  description: "Работа с уведомлениями"
paths:
  /banner:
    get:
      security:
        - bearerAuth: []
      tags:
        - "banner"
      summary: "Получение баннера"
      description: "Пример: GET /banner?msisdn=79161234567&queryId=00d686a2"
      operationId: "getBanner"
      parameters:
        - name: "msisdn"
          in: "query"
          required: true
          description: "Номер телефона, состоит из 11 цифр"
          schema:
            type: string
        - name: "queryId"
          in: "query"
          required: true
          description: "Идентификатор запроса"
```

WebSocket

WebSocket - протокол связи поверх TCP-соединения, предназначенный для обмена сообщениями (клиент-сервер и др.) в режиме реального времени.

Пример: BE Мой МТС



Преимущества и недостатки

-  SOAP
- Преимущества:
- 1) Наличие строгой спецификации
 - 2) Однозначность.
 - 3) Отраслевой стандарт по версии W3C
 - 4) Можно использовать с другими протоколами, помимо HTTP

-  REST
- Преимущества:
- 1) Надёжность (за счёт отсутствия необходимости сохранять информацию о состоянии клиента, которая может быть утеряна).
 - 2) Масштабируемость
 - 3) Лёгкость внесения изменений
 - 4) Производительность (за счет кеша)
 - 5) Сообщения в формате json/xml

-  WebSocket
- Преимущества:
- 1) Нет ограничений с кросс-доменными запросами
 - 2) Можно обмениваться в том числе бинарными данными
 - 3) Скорость и эффективность (соединение открыто)
 - 4) Нет ограничений на время жизни канала
 - 5) Масштабируемость (большое количество пользователей)
 - 6) Работают асинхронно

-  Недостатки:
- 1) сложность реализации.
 - 2) XML имеет большой размер
 - 3) сложность/ресурсоемкость парсинга XML-данных.

-  Недостатки:
- 1) Ограничения на кросс-доменные запросы
 - 2) отсутствие спецификации;

-  Недостатки:
- 1) Сложно поддерживать и требуется хорошее железо (KTC)

Инструменты для тестирования

SOAP

- 1) SOAP UI
- 2) Язык программирования, например Python

REST

- 1) Postman
- 2) Curl
- 3) SOAP UI
- 4) Язык программирования, например, Python

WebSocket

- 1) Язык программирования, например, Python



Глава 3

Валидация данных

Валидация

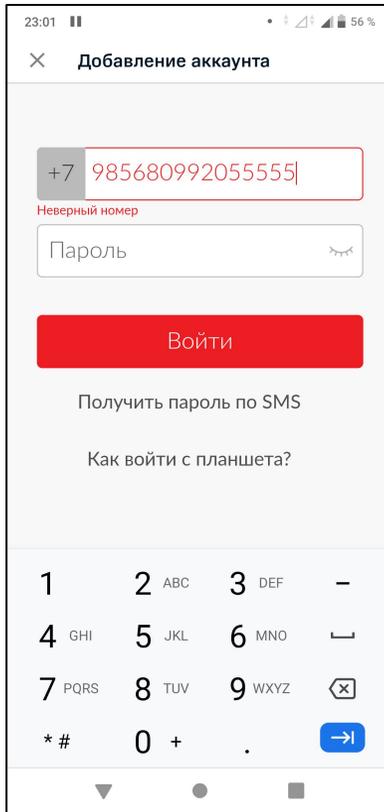
Валидация – это проверка на соответствие требованиям.

Для REST и WebSocket: проверка json.

Для SOAP: проверка xml.

Пример: валидация json файла на соответствие параметров определенным типам данных (строка, число, булево значение)

Валидация (Примеры)



23:01 56%

Добавление аккаунта

+7 98568099205555

Неверный номер

Пароль

Войти

Получить пароль по SMS

Как войти с планшета?

1 2 ABC 3 DEF -

4 GHI 5 JKL 6 MNO ↵

7 PQRS 8 TUV 9 WXYZ ✕

*# 0 + . →

FE

```
Ответ
{
  "request_id": "de7b956e-ba89-4327-a3b7-593a68979677",
  "status": "OK",
  "result": {
    "query_id": "djfveir03",
    "recommendations": [
      {
        "order": 1,
        "account_id": "79858219440",
        "title": "Мобильная связь",
        "type": "mobile"
      },
      {
        "order": 4,
        "account_id": "79858218090",
        "type": "mobile"
      }
    ]
  }
}
```

BE

На что мы потратили время?

- 1) Узнали чем отличаются потоки и процессы.
- 2) Узнали про способы реализации асинхронности.
- 3) Узнали разницу между синхронностью и асинхронностью на примеры проекта Мой МТС.
- 4) Узнали про программные интерфейсы взаимодействия.
- 5) Узнали чем отличаются REST, WebSocket, SOAP.
- 6) Узнали что такое WADL, WSDL.
- 7) Узнали про валидацию данных.



Что если Вы ничего не узнали нового?

1) Для того, чтобы открыть свой собственный банк достаточно всего 300 млн. рублей.

2) На основании интеграла ИТО самым лучшим прогнозом цены актива на завтра является цена сегодняшняя.

$$X_t = X_0 + \int_0^t \sigma_s dB_s + \int_0^t \mu_s ds.$$

3) Основатель компании «Toyota» Сакити Тойота использовал правило пять почему для решения проблем. Данное правило позволяет выявить истинные причинно-следственные связи.



Спасибо за внимание!

Можете задавать вопросы

Контакты:

E-mail: dkravchuk@stream.ru

Тел.: 7 (985) 821-94-40