метр
квадратный

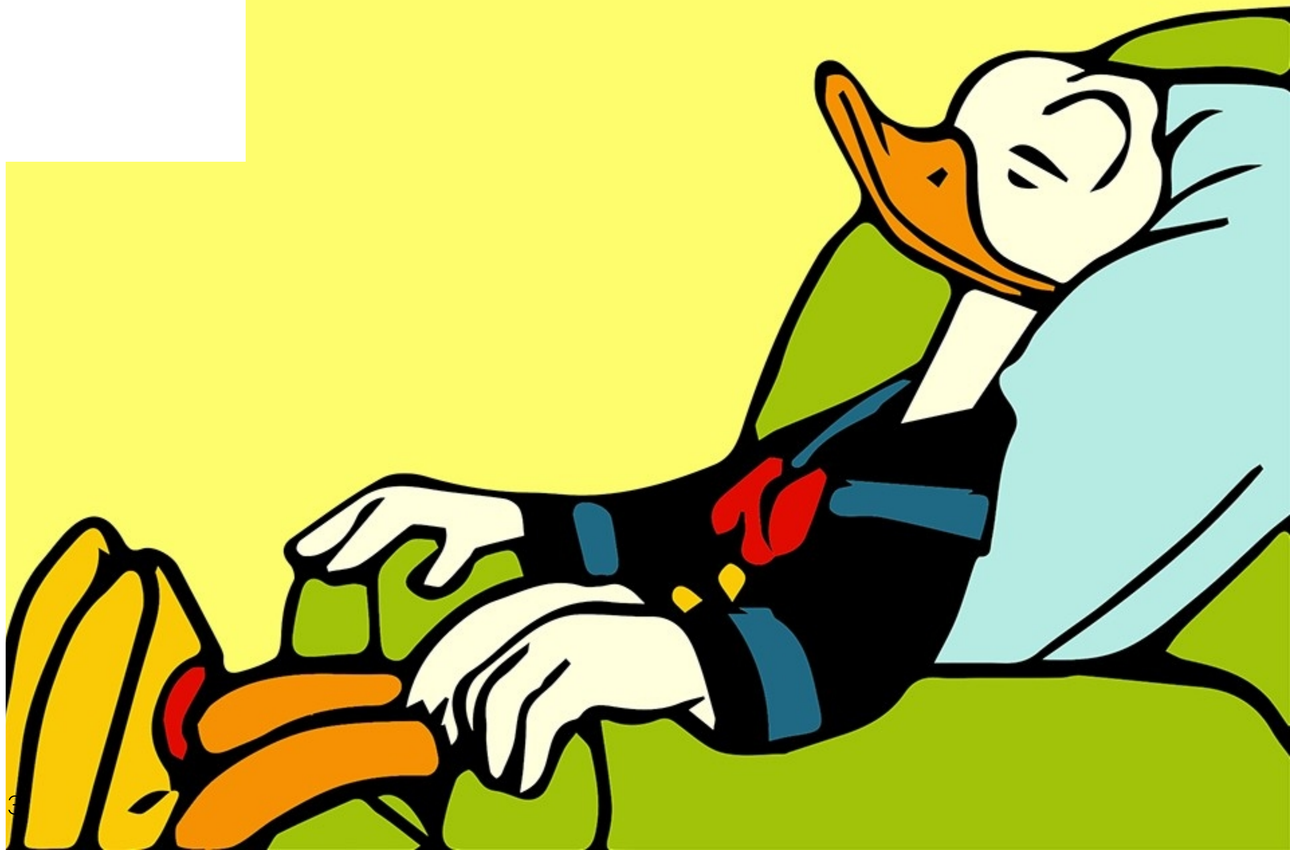
gRPC в iOS приложениях. REST in peace?

Светослав Карасев

Что будет дальше?

- Поговорим про технологии клиент-серверного взаимодействия
- Рассмотрим самые популярные и сравним их
- Поближе познакомимся с gRPC
- Рассмотрим как подключить gRPC в свой iOS проект
- Сделаем выводы

REST

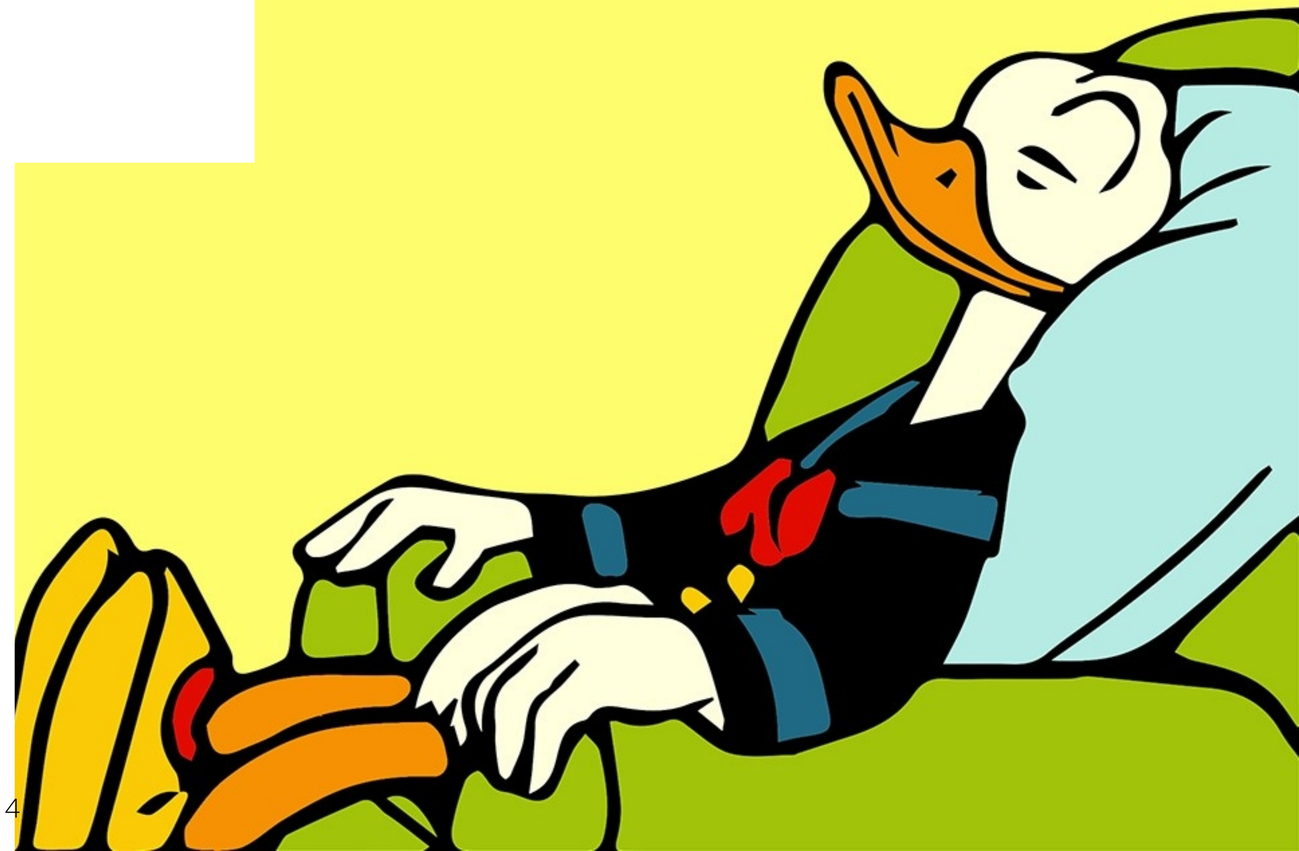


Проблемы

1. До сих пор нет единого понимания что такое RESTful API.

2. Большое количество эндпоинтов

3. OpenAPI(SWAGGER) не панацея



Что может решить эти проблемы?

- Жесткий контракт
- Документируемость
- Как можно меньше эндпоинтов (желательно 1)

Remote procedure call

Класс технологий, позволяющих вызывать функции или процедуры в другом адресном пространстве (на удалённых компьютерах, либо в независимой сторонней системе на том же устройстве).

```
struct ToDoItem {
    var id: String
    var text: String
}

protocol ToDoService {
    func list() -> [ToDoItem]
    func add(task: String) -> ToDoItem
    func remove(id: String) -> ToDoItem
}
```

SOAP

Simple Object Access Protocol

Протокол обмена структурированными сообщениями в распределённой вычислительной среде. Первоначально SOAP предназначался в основном для реализации удалённого вызова процедур (RPC). Сейчас протокол используется для обмена произвольными сообщениями в формате XML, а не только для вызова процедур.

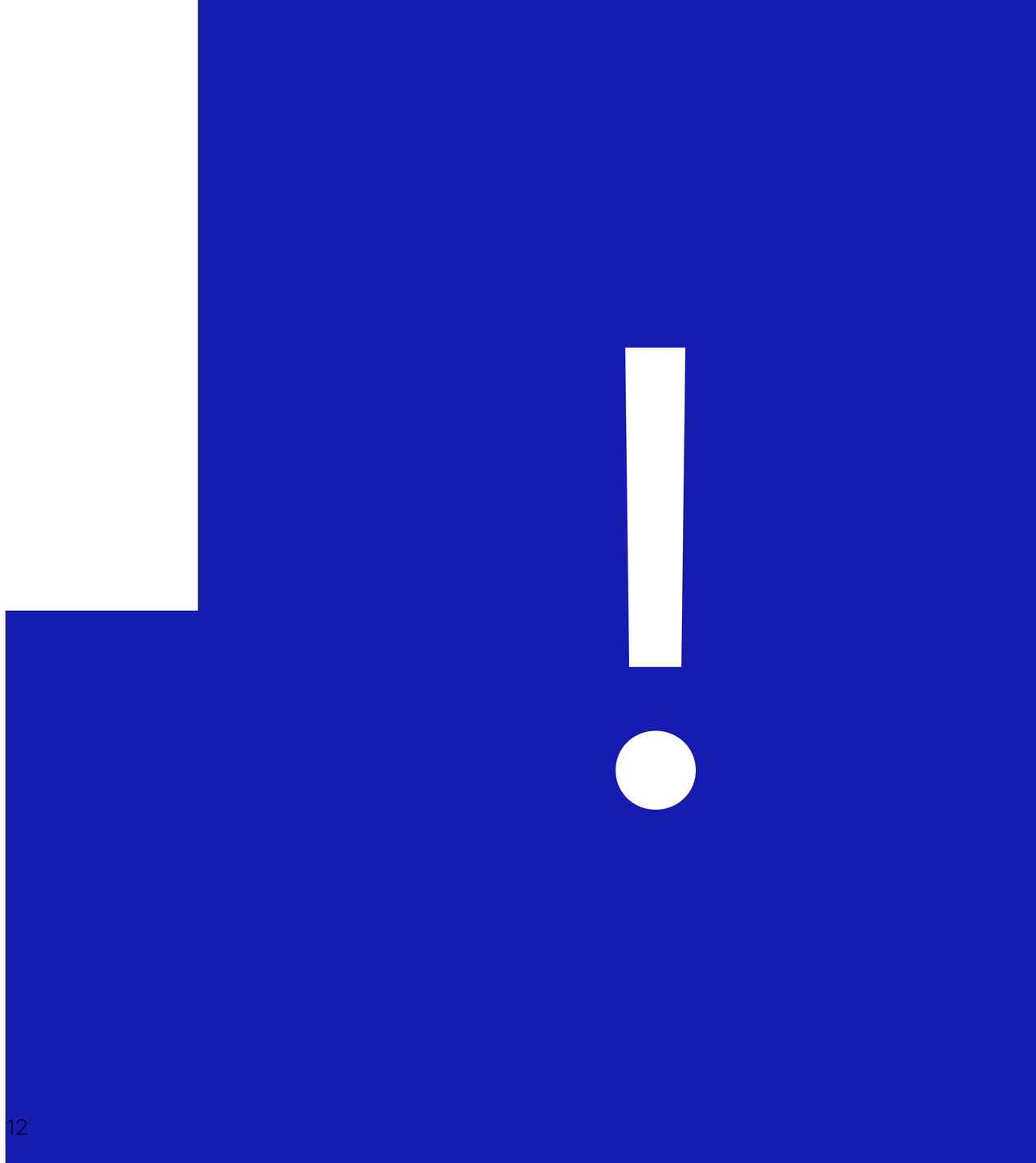
This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://tempuri.org/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:http="http://schemas.microsoft.com/ws/06/2004/policy/http" xmlns:msc="http://schemas.microsoft.com/ws/2005/12/wsdl/contract" xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://tempuri.org/" name="ISoapToDoService">
<wsdl:types>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:ser="http://schemas.microsoft.com/2003/10/Serialization/" elementFormDefault="qualified" targetNamespace="http://tempuri.org/">
<xs:import namespace="http://schemas.datacontract.org/2004/07/todo.Services.SOAP"/>
<xs:element name="SayHello">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="name" nillable="true" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="SayHelloResponse">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="SayHelloResult" nillable="true" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="List">
<xs:complexType>
<xs:sequence/>
</xs:complexType>
</xs:element>
<xs:element name="ListResponse">
<xs:complexType>
<xs:sequence>
<xs:element xmlns:q1="http://schemas.datacontract.org/2004/07/todo.Services.SOAP" minOccurs="1" name="ListResult" nillable="true" type="q1:ArrayOfToDoItem"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Add">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="text" nillable="true" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="AddResponse">
<xs:complexType>
<xs:sequence>
<xs:element xmlns:q2="http://schemas.datacontract.org/2004/07/todo.Services.SOAP" minOccurs="1" name="AddResult" nillable="true" type="q2:ToDoItem"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Remove">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="ite" nillable="true" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="RemoveResponse">
<xs:complexType>
<xs:sequence/>
</xs:complexType>
</xs:element>
</xs:schema>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://schemas.microsoft.com/2003/10/Serialization/" attributeFormDefault="qualified" elementFormDefault="qualified" targetNamespace="http://schemas.microsoft.com/2003/10/Serialization/">
<xs:element name="anyType" nillable="true" type="xs:anyType"/>
<xs:element name="anyURI" nillable="true" type="xs:anyURI"/>
<xs:element name="base64Binary" nillable="true" type="xs:base64Binary"/>
<xs:element name="boolean" nillable="true" type="xs:boolean"/>
<xs:element name="byte" nillable="true" type="xs:byte"/>
<xs:element name="dateTime" nillable="true" type="xs:dateTime"/>
<xs:element name="decimal" nillable="true" type="xs:decimal"/>
<xs:element name="double" nillable="true" type="xs:double"/>
<xs:element name="float" nillable="true" type="xs:float"/>
<xs:element name="int" nillable="true" type="xs:int"/>
<xs:element name="long" nillable="true" type="xs:long"/>
<xs:element name="QName" nillable="true" type="xs:QName"/>
<xs:element name="short" nillable="true" type="xs:short"/>
<xs:element name="string" nillable="true" type="xs:string"/>
<xs:element name="unsignedByte" nillable="true" type="xs:unsignedByte"/>
<xs:element name="unsignedInt" nillable="true" type="xs:unsignedInt"/>
<xs:element name="unsignedLong" nillable="true" type="xs:unsignedLong"/>
<xs:element name="unsignedShort" nillable="true" type="xs:unsignedShort"/>
<xs:element name="char" nillable="true" type="tns:char"/>
<xs:simpleType name="char">
<xs:restriction base="xs:int"/>
</xs:simpleType>
<xs:element name="duration" nillable="true" type="tns:duration"/>
<xs:simpleType name="duration">
<xs:restriction base="xs:duration">
<xs:pattern value="\-?P(\d*D)?(T(\d*H)?(\d*M)?(\d*(\.\d*)?S)?)?"/>
<xs:minInclusive value="-P10675199DT2H48M5.4775808S"/>
<xs:maxInclusive value="P10675199DT2H48M5.4775807S"/>
</xs:restriction>
</xs:simpleType>
<xs:element name="guid" nillable="true" type="tns:guid"/>
<xs:simpleType name="guid">
<xs:restriction base="xs:string">
<xs:pattern value="[\da-fA-F]{8}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{12}"/>
</xs:restriction>
</xs:simpleType>
<xs:attribute name="FactoryType" type="xs:QName"/>
<xs:attribute name="Id" type="xs:ID"/>
<xs:attribute name="Ref" type="xs:IDREF"/>
</xs:schema>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://schemas.datacontract.org/2004/07/todo.Services.SOAP" xmlns:ser="http://schemas.microsoft.com/2003/10/Serialization/" elementFormDefault="qualified" targetNamespace="http://schemas.datacontract.org/2004/07/todo.Services.SOAP">
<xs:import namespace="http://schemas.datacontract.org/2004/07/System"/>
<xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
<xs:complexType xmlns:ser="http://schemas.microsoft.com/2003/10/Serialization/" name="ToDoItem">
<xs:sequence>
<xs:element minOccurs="0" name="Id" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="Text" nillable="true" type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:element name="ToDoItem" nillable="true" type="tns:ToDoItem"/>
<xs:complexType xmlns:ser="http://schemas.microsoft.com/2003/10/Serialization/" name="ArrayOfToDoItem">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="unbounded" name="ToDoItem" nillable="true" type="tns:ToDoItem"/>
</xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfToDoItem" nillable="true" type="tns:ArrayOfToDoItem"/>
</xs:schema>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://schemas.microsoft.com/2003/10/Serialization/Arrays" xmlns:ser="http://schemas.microsoft.com/2003/10/Serialization/" elementFormDefault="qualified" targetNamespace="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
<xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/"/>
</wsdl:types>
<wsdl:message name="ISoapToDoService_SayHello_InputMessage">
<wsdl:part name="parameters" element="tns:SayHello"/>
</wsdl:message>
<wsdl:message name="ISoapToDoService_SayHello_OutputMessage">
<wsdl:part name="parameters" element="tns:SayHelloResponse"/>
</wsdl:message>
<wsdl:message name="ISoapToDoService_List_InputMessage">
<wsdl:part name="parameters" element="tns:List"/>
</wsdl:message>
<wsdl:message name="ISoapToDoService_List_OutputMessage">
<wsdl:part name="parameters" element="tns:ListResponse"/>
</wsdl:message>
<wsdl:message name="ISoapToDoService_Add_InputMessage">
<wsdl:part name="parameters" element="tns:Add"/>
</wsdl:message>
<wsdl:message name="ISoapToDoService_Add_OutputMessage">
<wsdl:part name="parameters" element="tns:AddResponse"/>
</wsdl:message>
<wsdl:message name="ISoapToDoService_Remove_InputMessage">
<wsdl:part name="parameters" element="tns:Remove"/>
</wsdl:message>
<wsdl:message name="ISoapToDoService_Remove_OutputMessage">
<wsdl:part name="parameters" element="tns:RemoveResponse"/>
</wsdl:message>
<wsdl:portType name="ISoapToDoService">
<wsdl:operation name="SayHello">
<wsdl:input wsam:Action="http://tempuri.org/ISoapToDoService/SayHello" message="tns:ISoapToDoService_SayHello_InputMessage"/>
<wsdl:output wsam:Action="http://tempuri.org/ISoapToDoService/SayHelloResponse" message="tns:ISoapToDoService_SayHello_OutputMessage"/>
</wsdl:operation>
<wsdl:operation name="List">
<wsdl:input wsam:Action="http://tempuri.org/ISoapToDoService/List" message="tns:ISoapToDoService_List_InputMessage"/>
<wsdl:output wsam:Action="http://tempuri.org/ISoapToDoService/ListResponse" message="tns:ISoapToDoService_List_OutputMessage"/>
</wsdl:operation>
<wsdl:operation name="Add">
<wsdl:input wsam:Action="http://tempuri.org/ISoapToDoService/Add" message="tns:ISoapToDoService_Add_InputMessage"/>
<wsdl:output wsam:Action="http://tempuri.org/ISoapToDoService/AddResponse" message="tns:ISoapToDoService_Add_OutputMessage"/>
</wsdl:operation>
<wsdl:operation name="Remove">
<wsdl:input wsam:Action="http://tempuri.org/ISoapToDoService/Remove" message="tns:ISoapToDoService_Remove_InputMessage"/>
<wsdl:output wsam:Action="http://tempuri.org/ISoapToDoService/RemoveResponse" message="tns:ISoapToDoService_Remove_OutputMessage"/>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="BasicHttpBinding" type="tns:ISoapToDoService">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
<wsdl:operation name="SayHello">
<soap:operation soapAction="http://tempuri.org/ISoapToDoService/SayHello" style="document"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="List">
<soap:operation soapAction="http://tempuri.org/ISoapToDoService/List" style="document"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="Add">
<soap:operation soapAction="http://tempuri.org/ISoapToDoService/Add" style="document"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="Remove">
<soap:operation soapAction="http://tempuri.org/ISoapToDoService/Remove" style="document"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="ISoapToDoService">
<wsdl:port name="BasicHttpBinding" binding="tns:BasicHttpBinding">
<soap:address location="http://localhost:5050/soap.asmx"/>
</wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

```
    <wsdl:portType name="ISoapToDoService">
        <wsdl:operation name="List">
            <wsdl:input wsam:Action="http://tempuri.org/ISoapToDoService/List"
message="tns:ISoapToDoService_List_InputMessage"/>
            <wsdl:output wsam:Action="http://tempuri.org/ISoapToDoService/ListResponse"
message="tns:ISoapToDoService_List_OutputMessage"/>
        </wsdl:operation>
        <wsdl:operation name="Add">
            <wsdl:input wsam:Action="http://tempuri.org/ISoapToDoService/Add"
message="tns:ISoapToDoService_Add_InputMessage"/>
            <wsdl:output wsam:Action="http://tempuri.org/ISoapToDoService/AddResponse"
message="tns:ISoapToDoService_Add_OutputMessage"/>
        </wsdl:operation>
        <wsdl:operation name="Remove">
            <wsdl:input wsam:Action="http://tempuri.org/ISoapToDoService/Remove"
message="tns:ISoapToDoService_Remove_InputMessage"/>
            <wsdl:output wsam:Action="http://tempuri.org/ISoapToDoService/RemoveResponse"
message="tns:ISoapToDoService_Remove_OutputMessage"/>
        </wsdl:operation>
    </wsdl:portType>
```

Как подключить в мой проект?

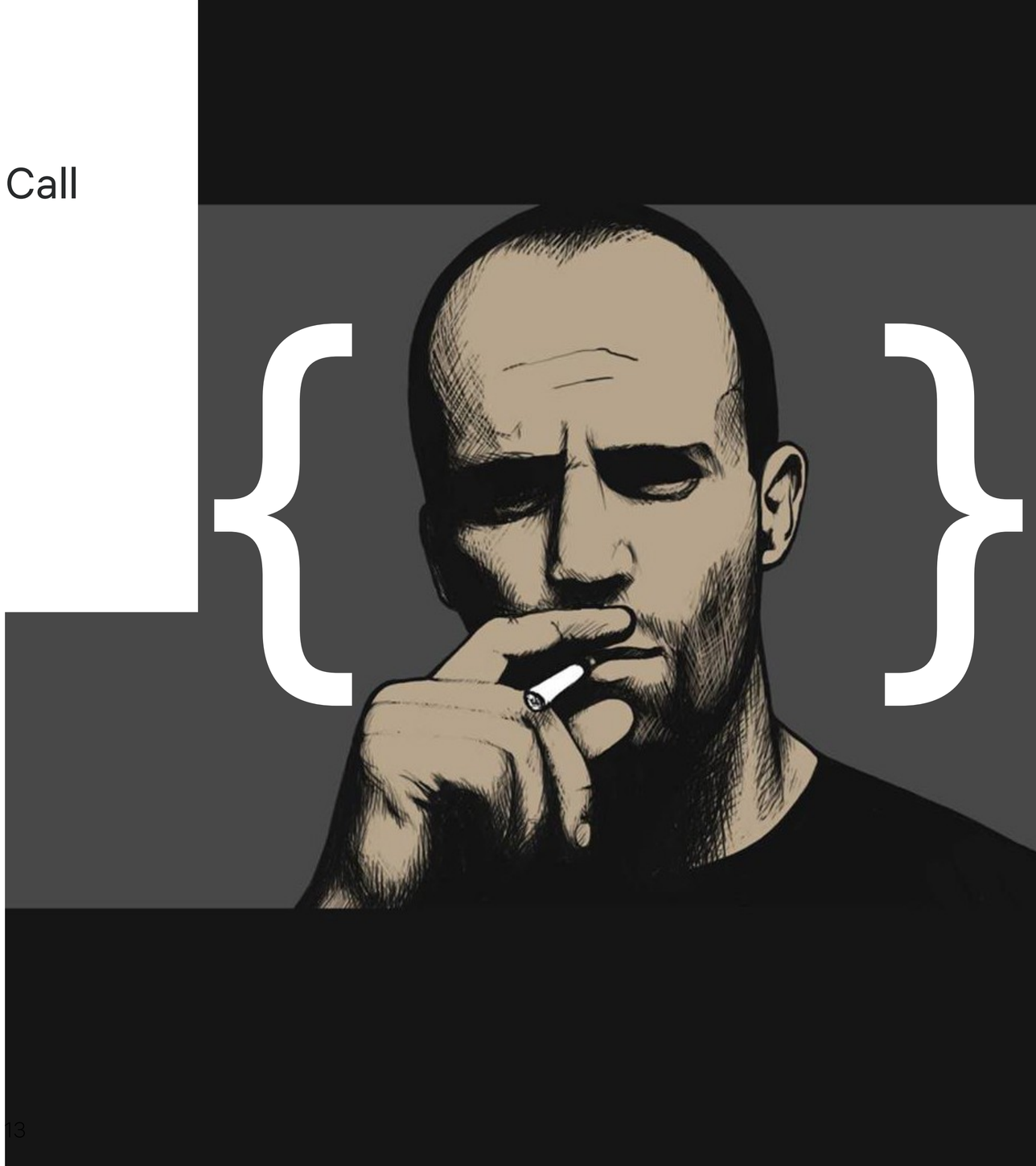
Не надо подключать в ваш проект



JSON-RPC

JavaScript Object Notation Remote Procedure Call

Протокол удалённого вызова процедур, использующий JSON для кодирования сообщений. Это очень простой протокол, определяющий только несколько типов данных и команд. JSON-RPC поддерживает уведомления (информация, отправляемая на сервер, не требует ответа) и множественные вызовы.



```
--> {"method": "add", "params": [{ "text": "Say Hello" }], "id": "1"}
<-- {"result": {"id": "2a74a01a-e61a-4c67-8660-3ff07d2d2ed1", "text": "Say Hello"}, "error": null, "id": "1"}
```

Плюсы:

- Очень простой
- Решает проблему множества эндпоинтов

Минусы:

- Все остальные проблемы в общем-то сохраняются

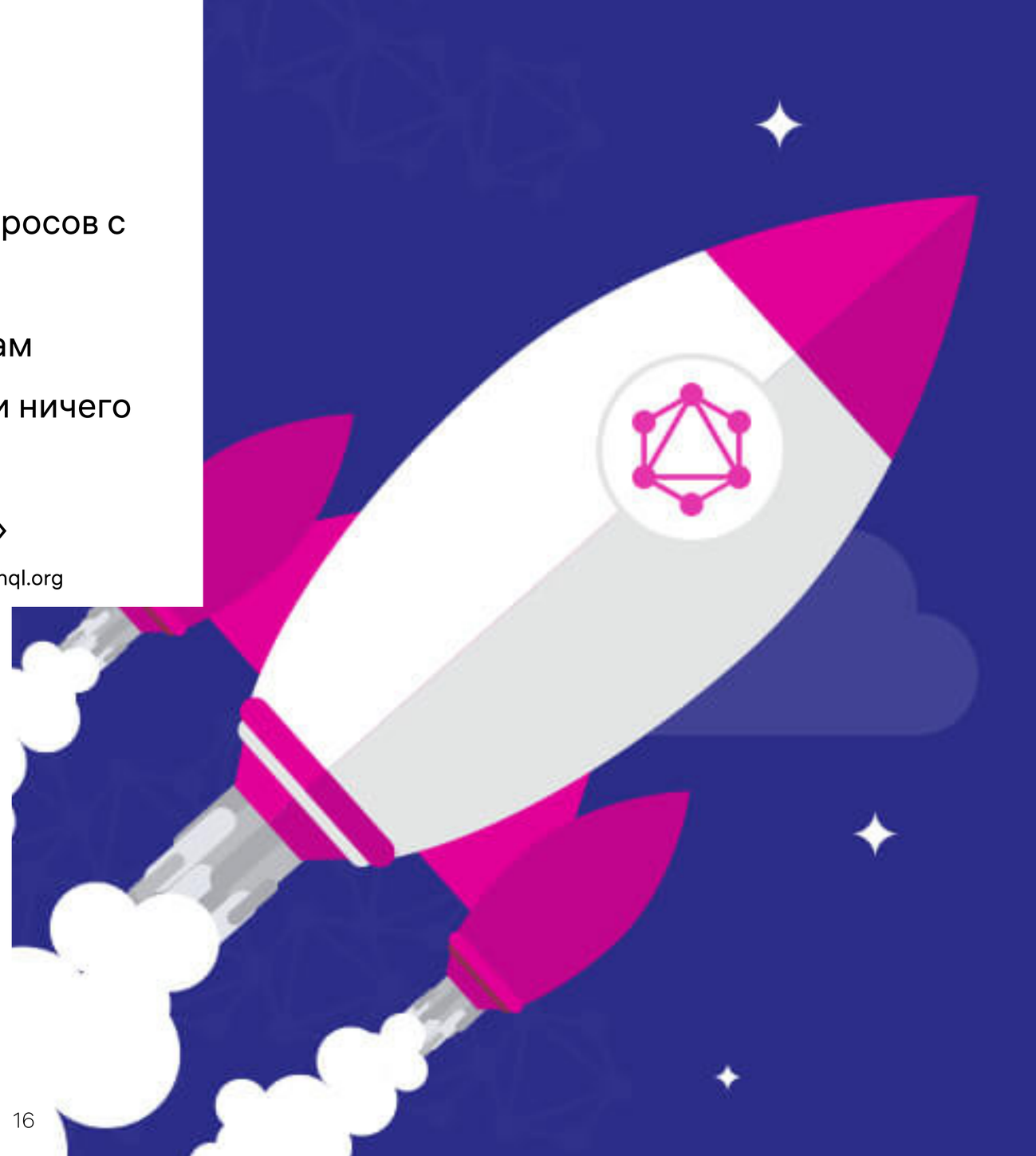
Как подключить в проект?

- Так же как и любой другой REST API
- URLSession + Codable, например

GraphQL

«Язык запросов для API и среда выполнения этих запросов с вашими данными. GraphQL предоставляет полное и понятное описание данных в вашем API, дает клиентам возможность запрашивать именно то, что им нужно, и ничего более, упрощает развитие API с течением времени и предоставляет мощные инструменты разработчика.»

© graphql.org



```
schema {
    query: Query
    mutation: Mutation
}

type Query {
    list(): [ToDoItem!]!
}

type Mutation {
    add(text: String!): ToDoItem!
    remove(id: String!): ToDoItem!
}

type ToDoItem {
    id: String!
    text: String!
}
```

```
query {
  toDoList {
    id
    name
  }
}
```

```
POST /graphql HTTP/1.1
Content-Type: application/json; charset=utf-8
Host: localhost:5000
Connection: close
User-Agent: Paw/3.2 (Macintosh; OS X/10.15.7)
GCDHTTPRequest
Content-Length: 58

{
    "query": "query{toDoList{id name}}",
    "variables": {}
}
```


HTTP/1.1 200 OK
Connection: close
Date: Mon, 09 Nov 2020 11:56:08 GMT
Content-Type: application/json
Server: Kestrel
Transfer-Encoding: chunked
{"data":{"toDoList":[{"id":0,"name":"First"}]}}

Плюсы:

+ Жесткий контракт

+ Строгая типизация

+ Выбор данных, содержащихся в ответе, задает клиент

+ Возможность объединения запросов

+ Есть рефлексия

Минусы:

- Из-за того, что модель ответа в одном и том же методе может быть разной все варианты кодгена клиентов оставляют желать лучшего

- Довольно большой объем данных передается по сети

- Проблемы с http кэшированием

Как подключить в проект?

- Руками. Все Query и Mutations можно собрать скриптом в файл констант. И просто отсылать и принимать JSON через любимый сетевой клиент.
- Appolo iOS. Кодген из схемы. Добавит NPM вам в билдфазу. Будет создавать новые отдельные модели для кажой Query.

Apollo vs Hands

https://github.com/svedm/graphql-sample

https://youtu.be/FHScWP8M844?t=2519

Thrift

произносится как [θrift]

Язык описания интерфейсов, который используется для определения и создания служб под разные языки программирования.

```
struct ToDoTask {
    1: string id
    2: string value
}

exception NotFound {
    1: string message
}

service ToDoService {
    list<ToDoTask> itemsList(),
    ToDoTask add(1:string text),
    ToDoTask remove(1:string id) throws (1:NotFound error),
}
```

Плюсы

+ Жесткий строготипизированый контракт

+ Поддерживаются исключения

+ Формат связи уровня приложений и формат связи уровня сериализации строго разделены. Они могут быть изменены независимо друг от друга.

+ Встроенные типы сериализации включают в себя: двоичный, текстовый(JSON) и компактный двоичный.

Минусы

- Мало распространен

- Очень слабая документация

Как подключить в проект?

- https://github.com/apache/thrift/tree/master/lib/swift
- Клиент не обновлялся 2 года и написан под Swift3
- Кодген не умеет в дефолтные значения

gRPC

Высокопроизводительный фреймворк разработанный компанией Google для вызов удаленных процедур (RPC), работает поверх HTTP/2.

```
syntax = "proto3";

package todo;

message Empty {
}

message ToDoItem {
  string id = 1;
  string text = 2;
}

message ListResponse {
  repeated ToDoItem items = 1;
}

message AddRequest {
  string text = 1;
}

message RemoveRequest {
  string id = 1;
}

service ToDoService {
  rpc List(Empty) returns (ListResponse);
  rpc Add(AddRequest) returns (ToDoItem);
  rpc Remove(RemoveRequest) returns (ToDoItem);
}
```

+ Protobuf в качестве инструмента описания типов данных и сериализации.

+ HTTP/2 в качестве транспорта.

+ Статические пути — никаких больше «сервис/коллекция/ресурс/запрос?параметр=значение». Теперь только «сервис», а что внутри — описывайте в терминах вашей модели и её событий.

+ SSL/TLS, OAuth 2.0, аутентификация через сервисы Google, плюс можно прикрутить свою (например, двухфакторную)

+ Поддержка gRPC в публичных API от Google и других сервисов

- Довольно тяжело дебажить сеть

- Нужно постоянно думать об обратной совместимости

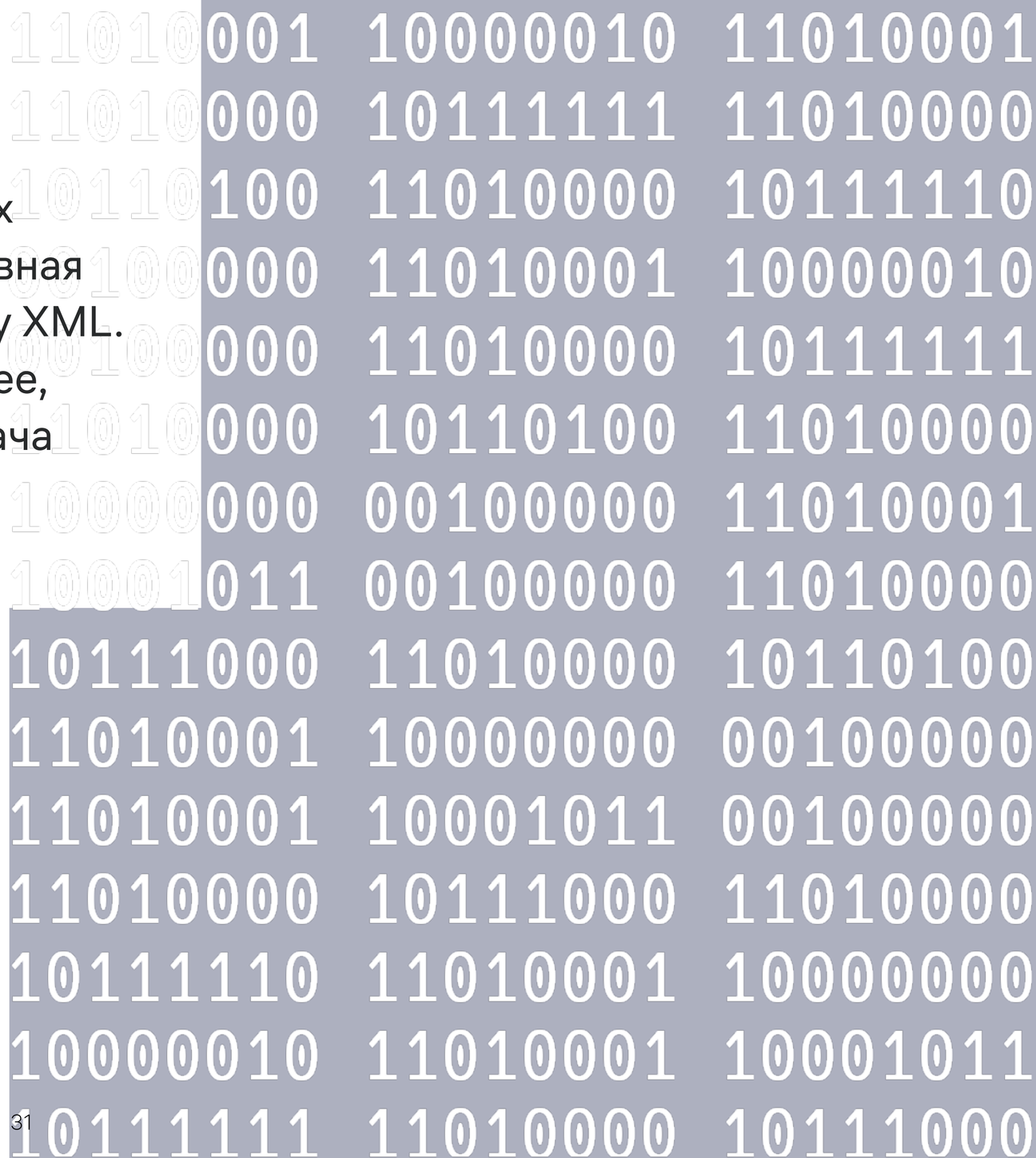
- Проблемы с использованием из браузера
(Современные браузеры не могут предоставить уровень управления HTTP/2, необходимый для gRPC клиента)

Технология	REST	SOAP	JSON-RPC	GraphQL	Thrift	gRPC
Контракт	-	WSDL	-	graphql schema	Thrift	Proto
Протокол	HTTP1/2	Любой прикладной	Любой прикладной. Чаще http	Любой прикладной. Чаще http	Несколько прикладных, легко переключаться	HTTP/2
Payload	Любой, чаще json	xml	json	json	json/binary	binary(protobuf)
Streaming	-	-	-	+	+	+
Поддержка вызовов из браузера(JS)	+	+	+	+	+	-
Кодогенерация	-/Сторонняя OpenAPI	+	-	+	+	+
API reflection	-	-	-	+	-	+
Версионность контракта	-	+	-	-	-	-

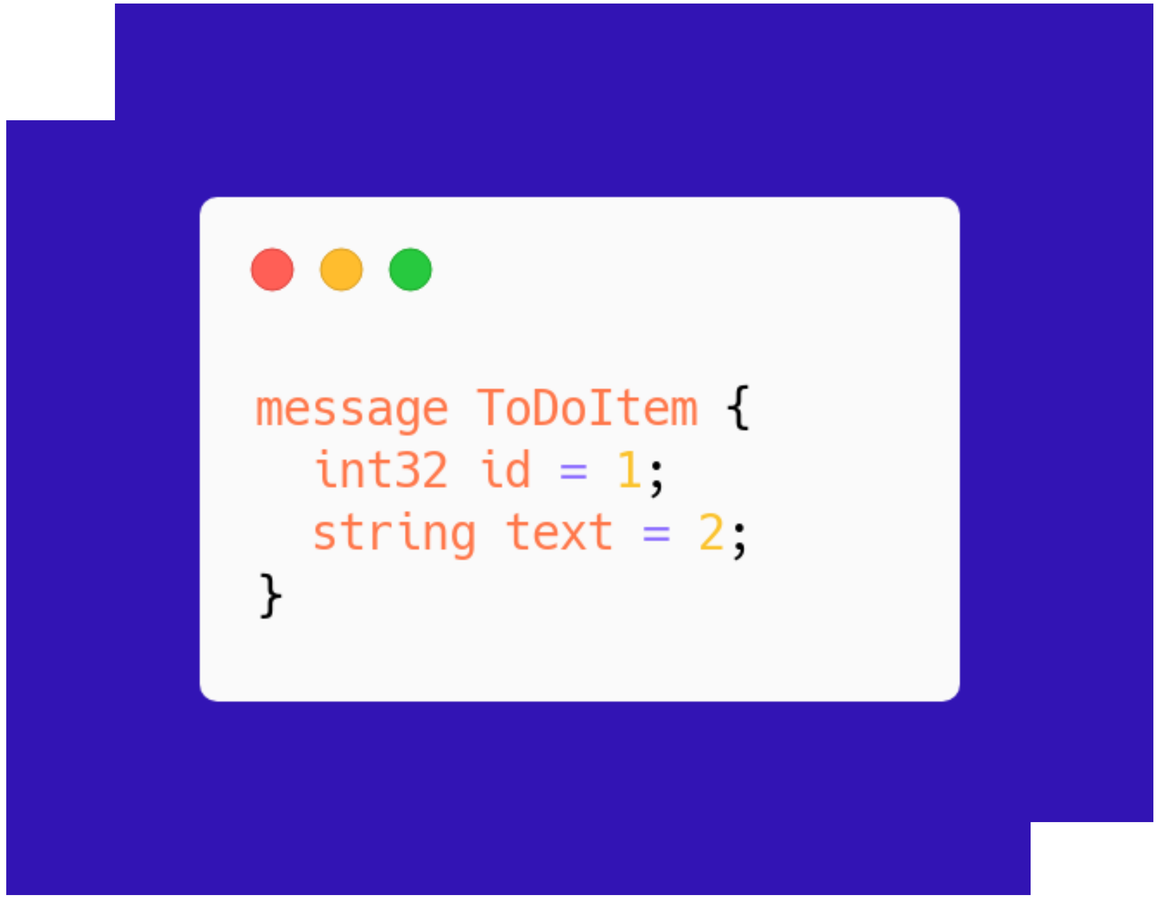

gRPC

Protocol Buffers

Протокол сериализации структурированных данных, предложенный Google как эффективная бинарная альтернатива текстовому формату XML. Protocol Buffers проще, компактнее и быстрее, чем XML, поскольку осуществляется передача бинарных данных, оптимизированных под минимальный размер сообщения.



Как это работает?



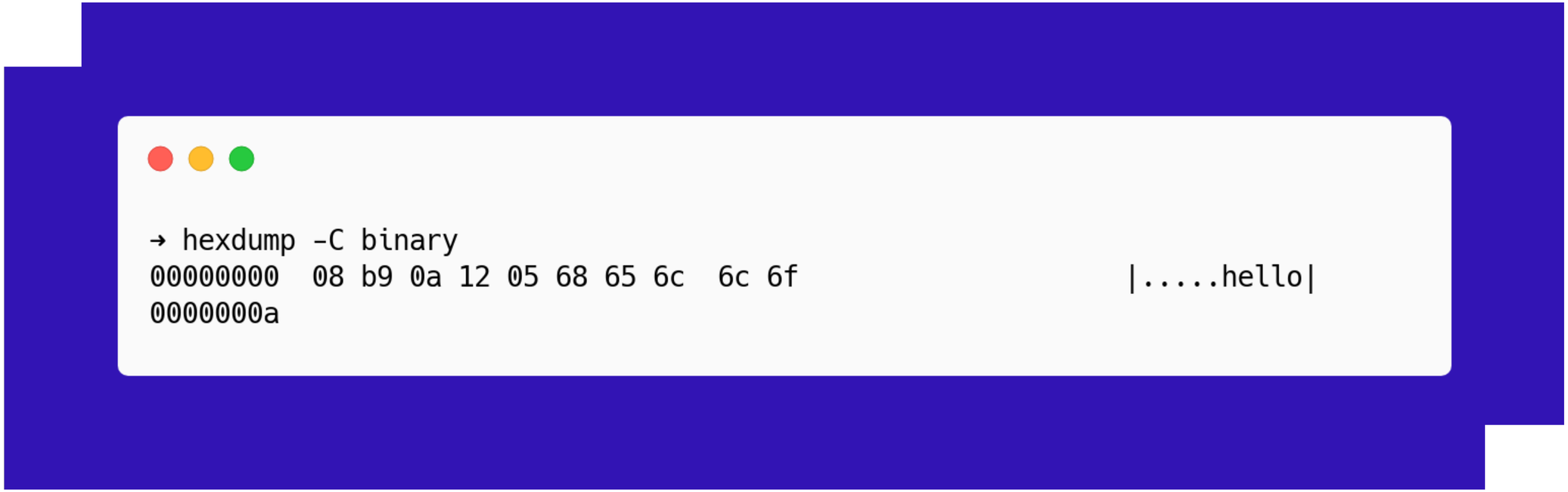
Преобразуем в бинарный вид

Файл «data»

id: 1337

message: "hello"

$ protoc --encode=ToDo todo.proto < data > binary


→ hexdump -C binary
00000000 08 b9 0a 12 05 68 65 6c 6c 6f |.....hello|
0000000a

[meta][payload][meta][payload]...

Мета формируется следующим образом:

(field_number << 3) | wire_type

Типы данных

Type	Meaning	Used For
0	Varint	int32, int64, uint32, uint64, sint32, sint64, bool, enum
1	64-bit	fixed64, sfixed64, double
2	Length-delimited	string, bytes, embedded messages, packed repeated fields
3	Start group	groups (deprecated)
4	End group	groups (deprecated)
5	32-bit	fixed32, sfixed32, float

Объявление сервиса

```
message ListResponse {
  repeated ToDoItem items = 1;
}

message AddRequest {
  string text = 1;
}

message RemoveRequest {
  string id = 1;
}

service ToDoService {
  rpc List(Empty) returns (ListResponse);
  rpc Add(AddRequest) returns (ToDoItem);
  rpc Remove(RemoveRequest) returns (ToDoItem);
}
```

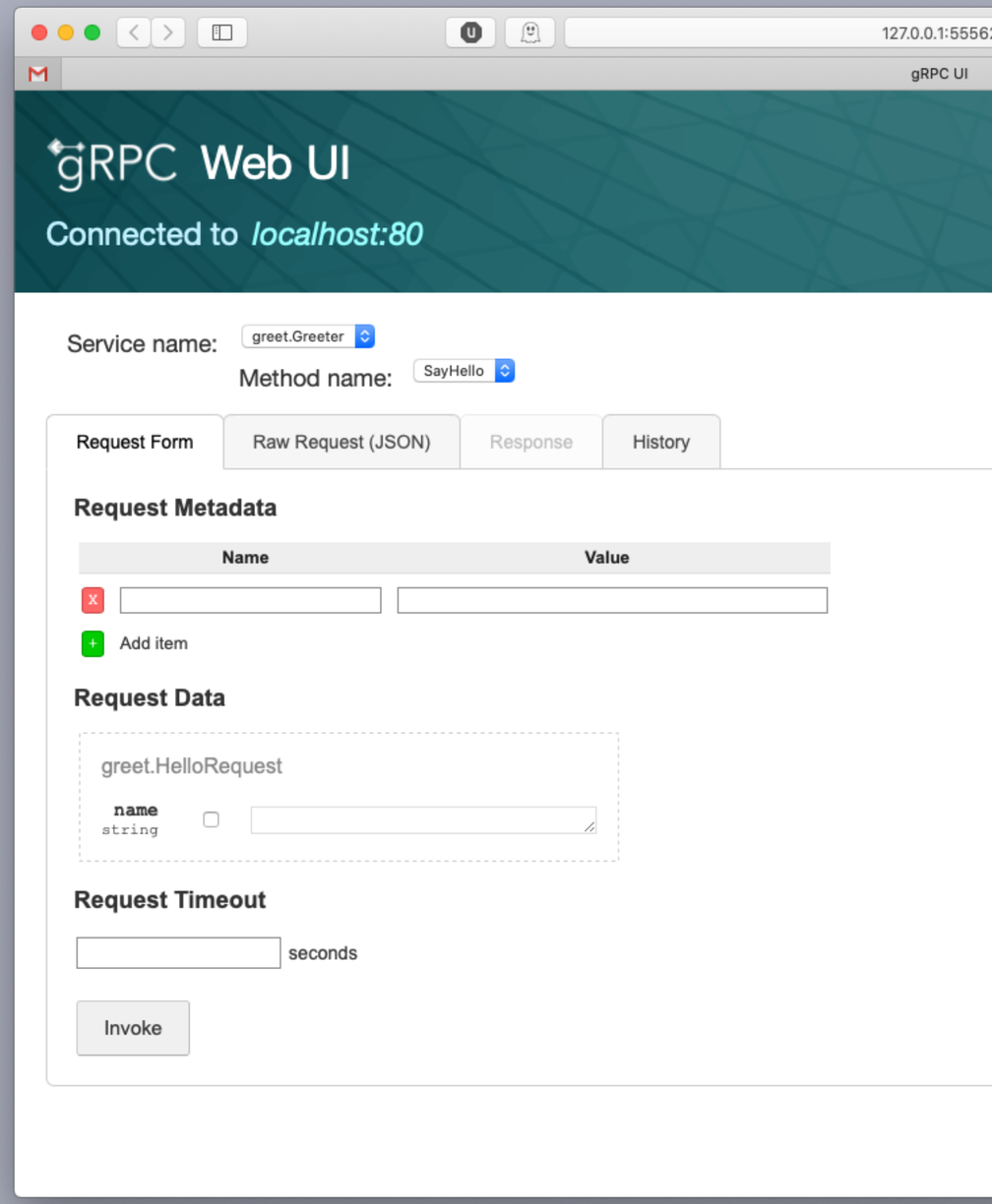
gRPCUI

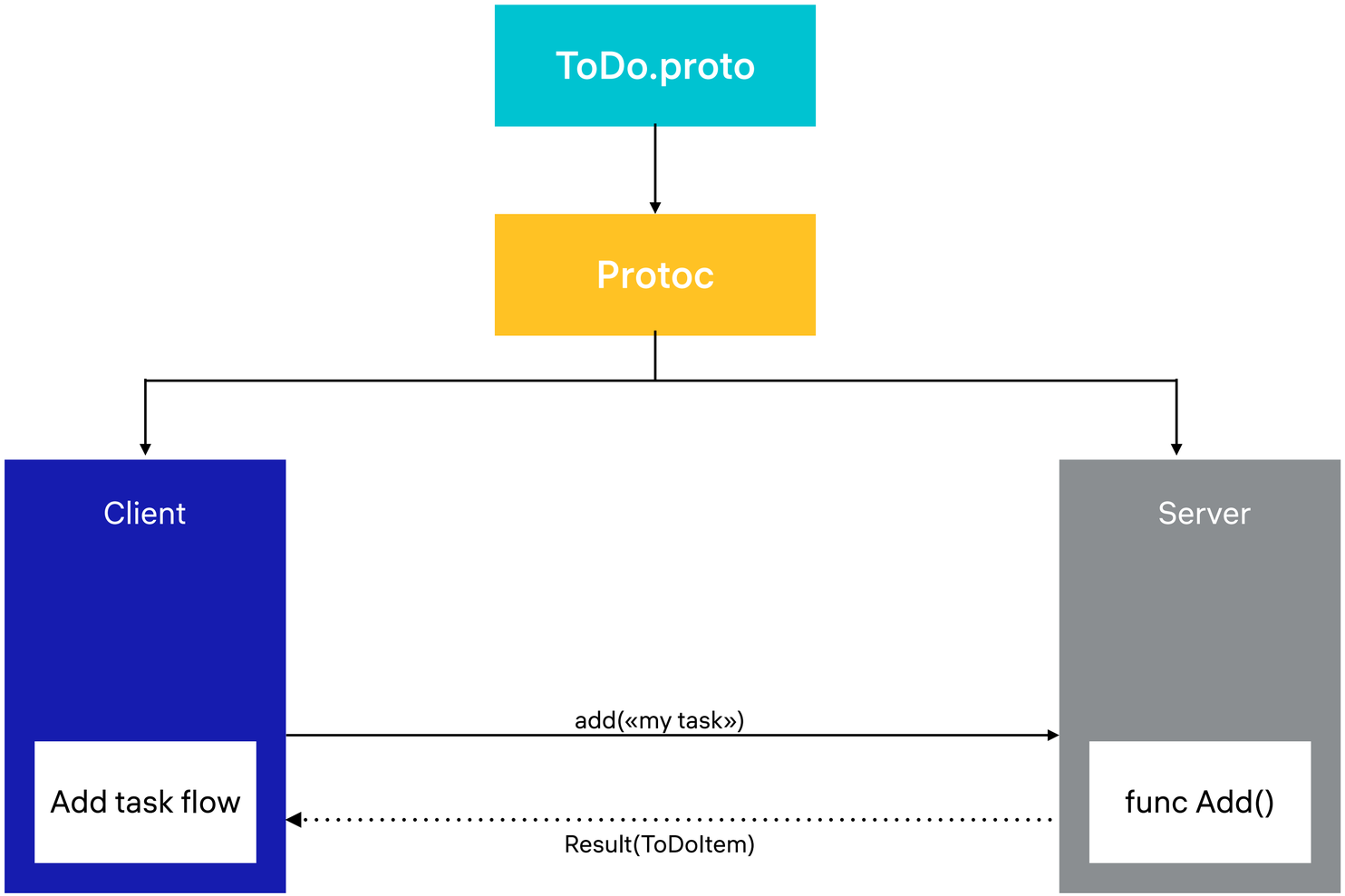
https://github.com/fullstorydev/grpcui

Wireshark

Много полезного

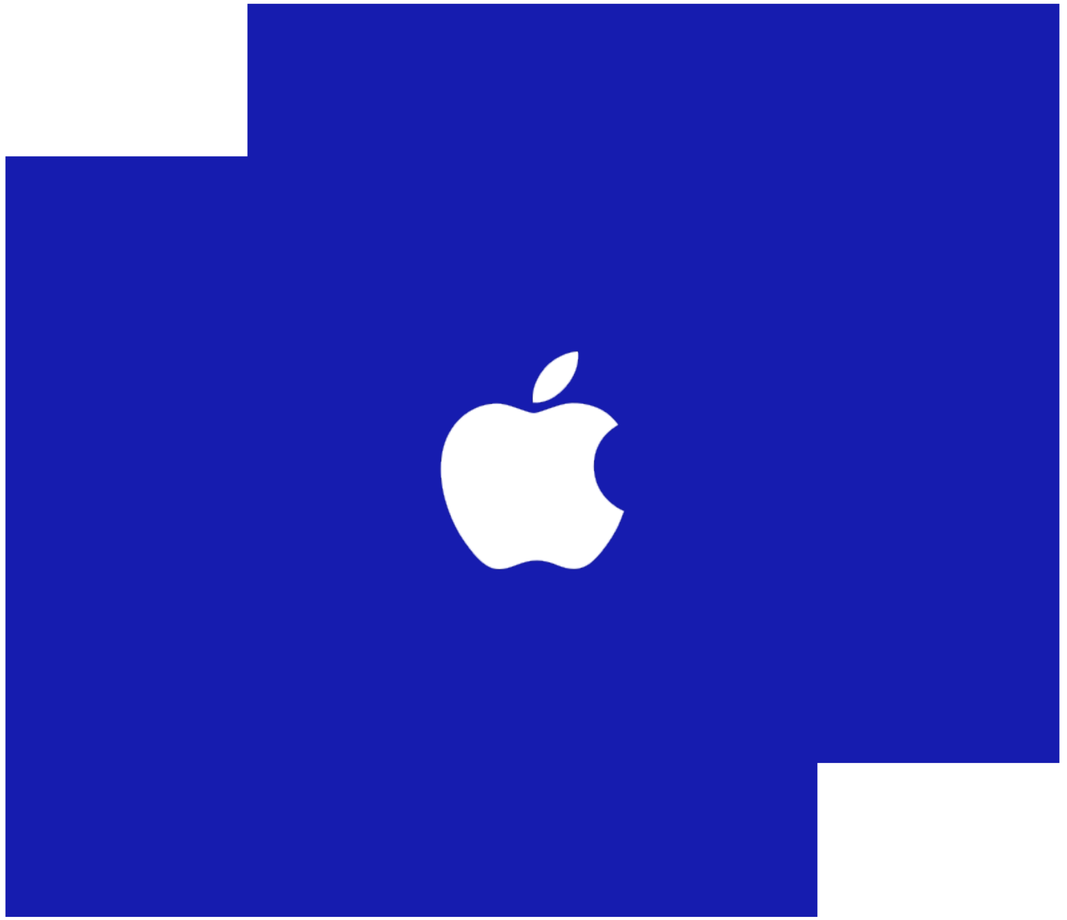
https://github.com/grpc-ecosystem/awesome-grpc




ToDo.proto
Protoc
Client
Server
add(«my task»)
Add task flow
func Add()
Result(ToDoItem)

Подключаем в iOS проект

- Protobuf
- apple/swift-protobuf
- grpc/grpc-swift



Выводы

+ grpc-swift вполне продакшн ready

+ Жесткий типизированный контракт

+ Скорость работы

- Кодген может и не взлететь (проблемы с версионированием и доставкой плагинов)

- Underscore нэйминг сгенерированных методов

- Подключение репозитория с протофайлами как сабмодуля может причинять боль

- Тестировщик самостоятельно не посмотрит трафик в чарлике и не заведет баг на бэк

Подводя итог

- Сравнили различные RPC
- Разобрались с утройством gRPC
- Рассмотрели использование технологии на iOS
- Осознали плюсы и минусы технологии

Использовать ли?

Вопросы?

Исходный код

https://github.com/svedm/grpc-sample

https://github.com/svedm/graphql-sample

Светослав Карасев

tg: @svedm