
Яндекс

Яндекс

Сириус

Работа с сетью

Алексей Шведчиков , Разработчик

Без сети - никуда

6.8.1

6.8.2

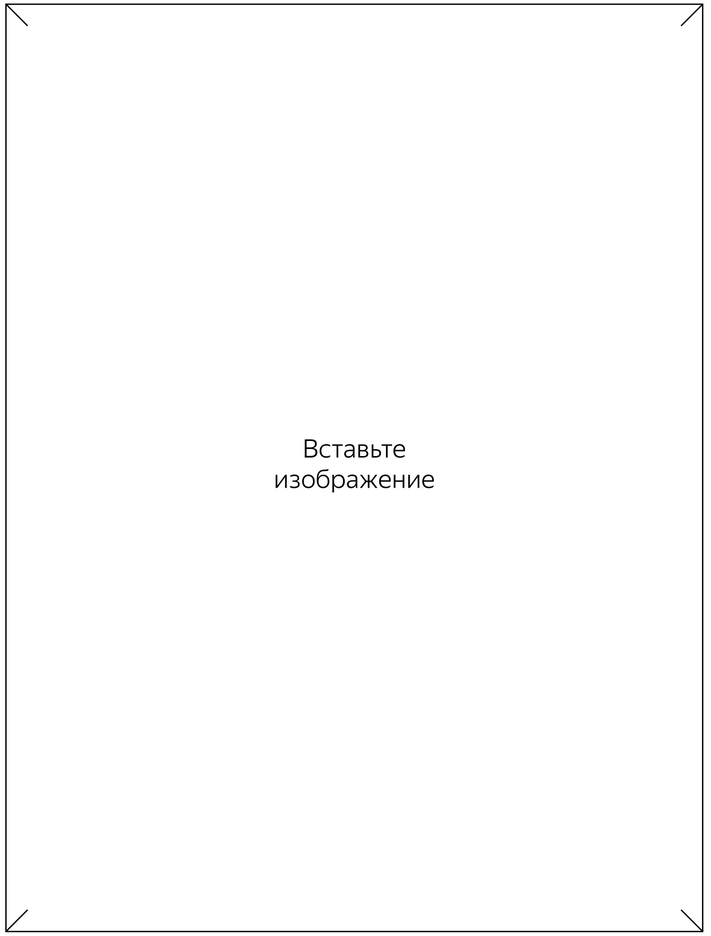
Тупик

6.8.3

А как работать с сетью?

Обзор

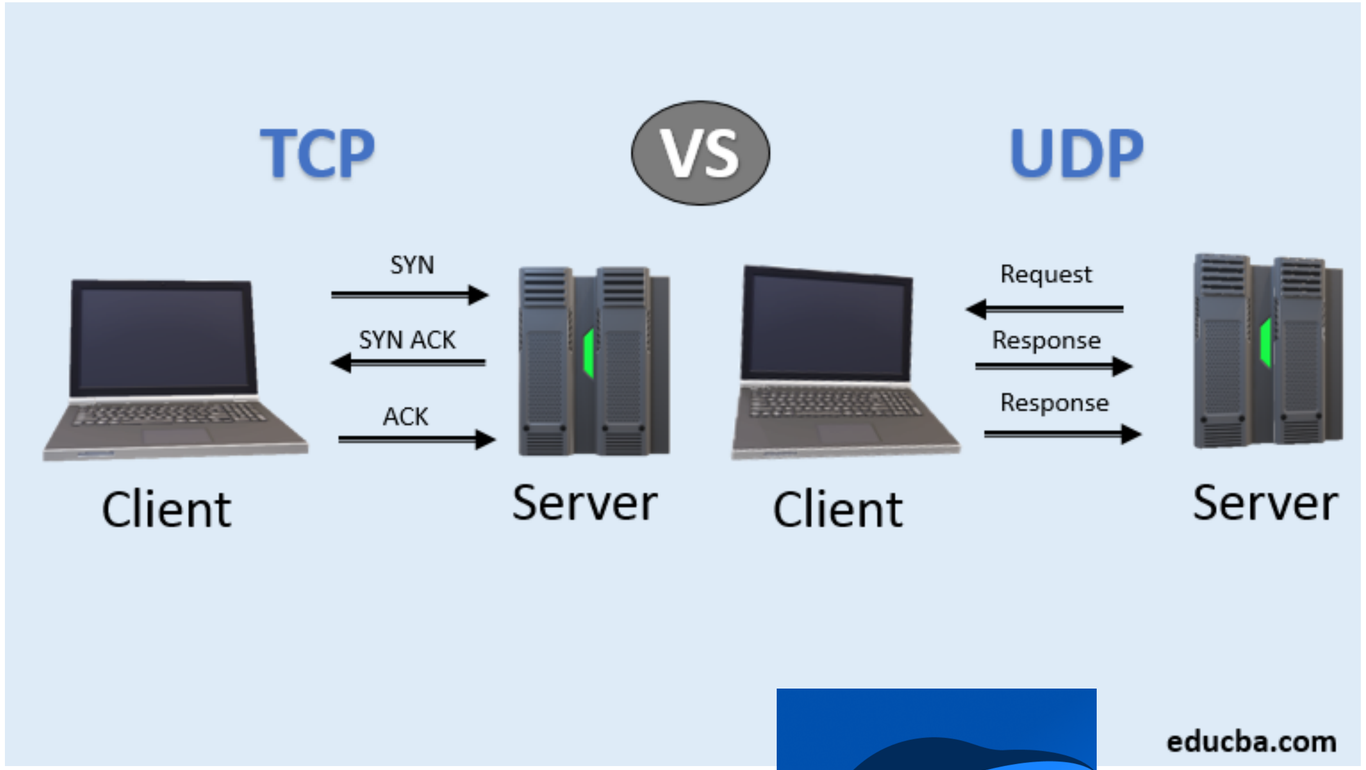
1. Сокеты
2. Пакеты
3. Сериализация



Сокеты

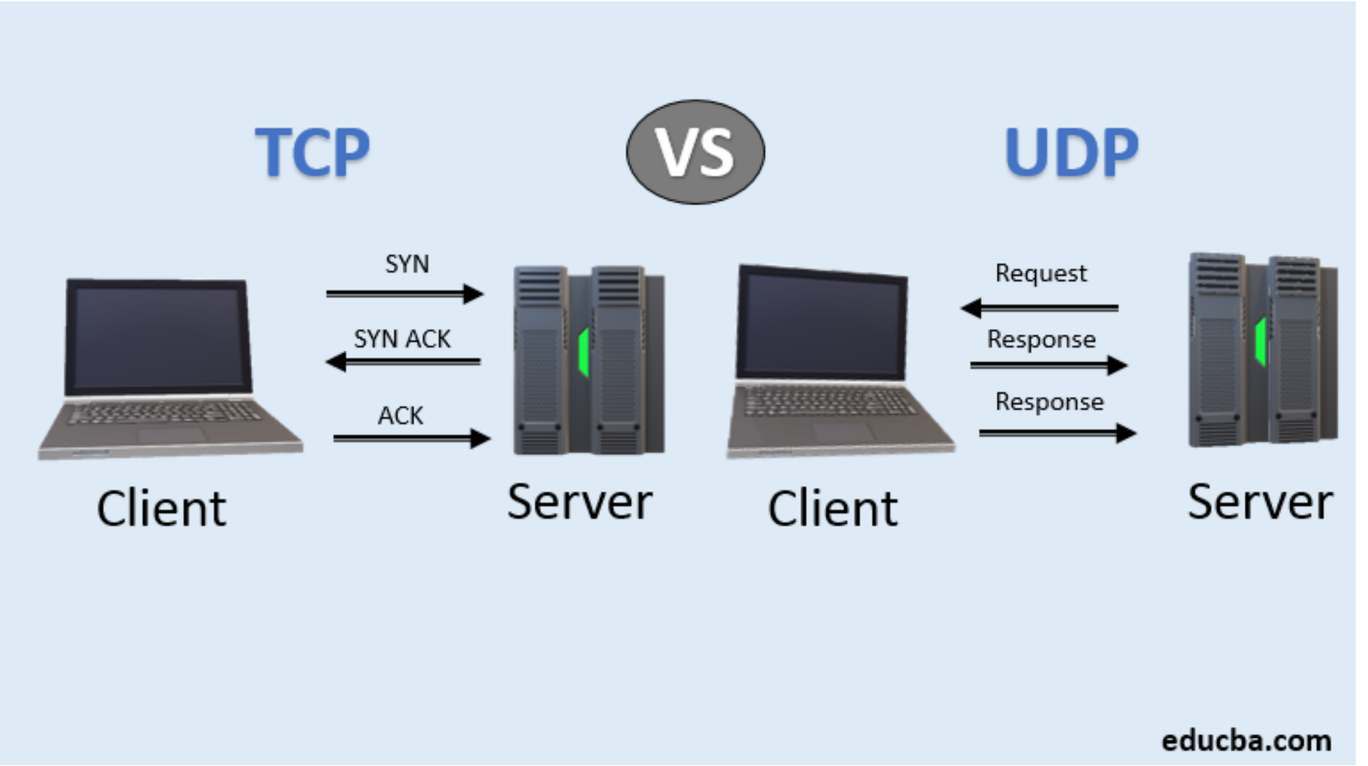
HTTP

gRPC



SSH

FTP


TCP
VS
UDP
SYN
SYN ACK
ACK
Client
Server
Request
Response
Response
Client
Server
educba.com

Сокеты

Dart из коробки

```
//открыть соединение
final socket = await  Socket.connect('localhost', 4567);

//получать данные
socket.listen(…)

//отправлять данные
socket.write(…)
```

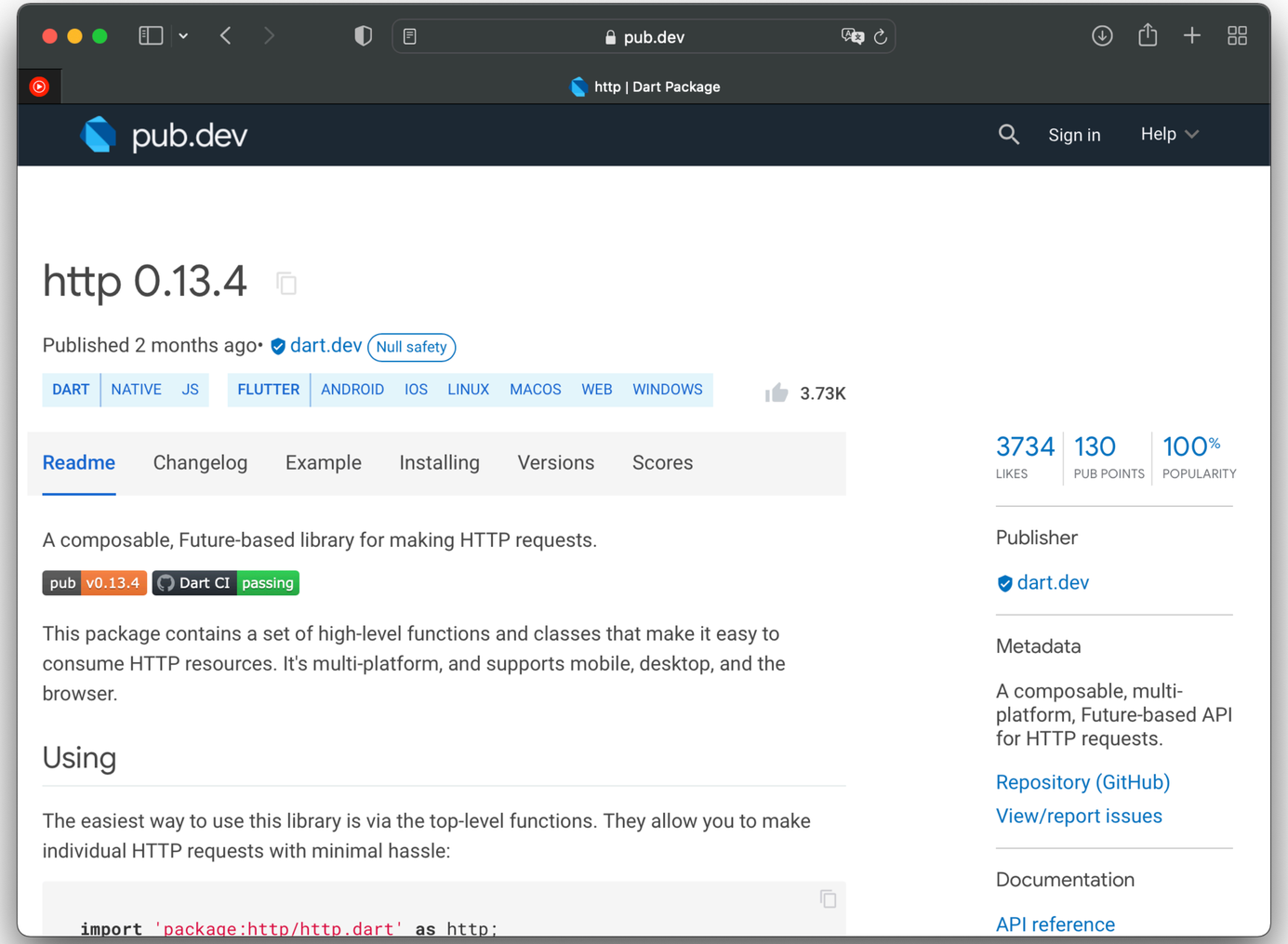
Статья

HTTP

Велосипед.

Жак Фреско

http


pub.dev
http | Dart Package
pub.dev
Sign in
Help
http 0.13.4
Published 2 months ago• dart.dev Null safety
DART NATIVE JS FLUTTER ANDROID IOS LINUX MACOS WEB WINDOWS
3.73K
Readme Changelog Example Installing Versions Scores
A composable, Future-based library for making HTTP requests.
pub v0.13.4 Dart CI passing
This package contains a set of high-level functions and classes that make it easy to consume HTTP resources. It's multi-platform, and supports mobile, desktop, and the browser.
Using
The easiest way to use this library is via the top-level functions. They allow you to make individual HTTP requests with minimal hassle:
import 'package:http/http.dart' as http;
3734 LIKES
130 PUB POINTS
100% POPULARITY
Publisher
dart.dev
Metadata
A composable, multi-platform, Future-based API for HTTP requests.
Repository (GitHub)
View/report issues
Documentation
API reference

http

Как сделать запрос?

1. Подключи http

```
http: ^0.13.4
```

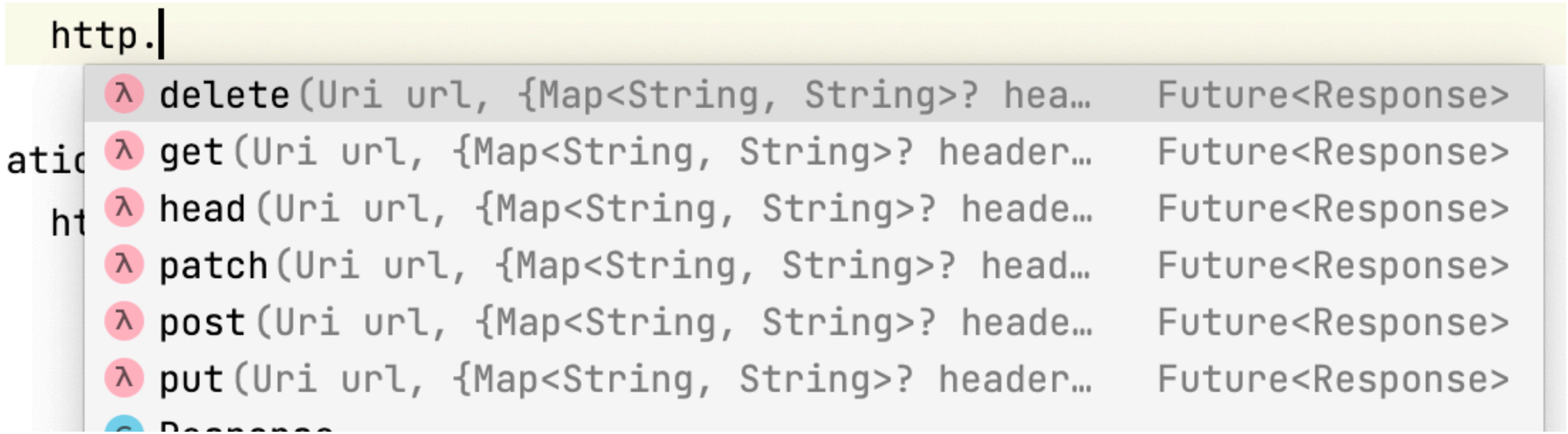
2. Импортируй

```
import 'package:http/http.dart' as http;
```

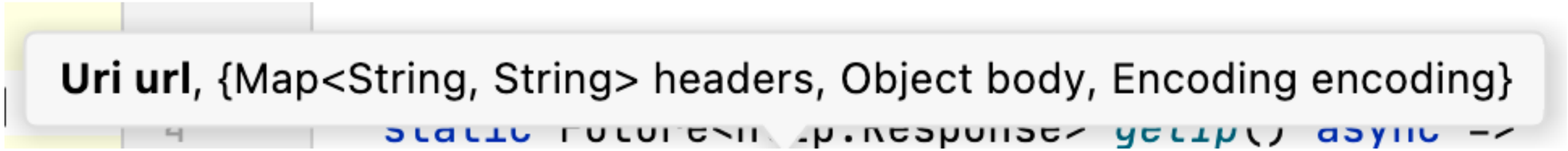
3. Используй

```
http.get(Uri.parse('https://api.ipify.org'));
```

Виды запросов

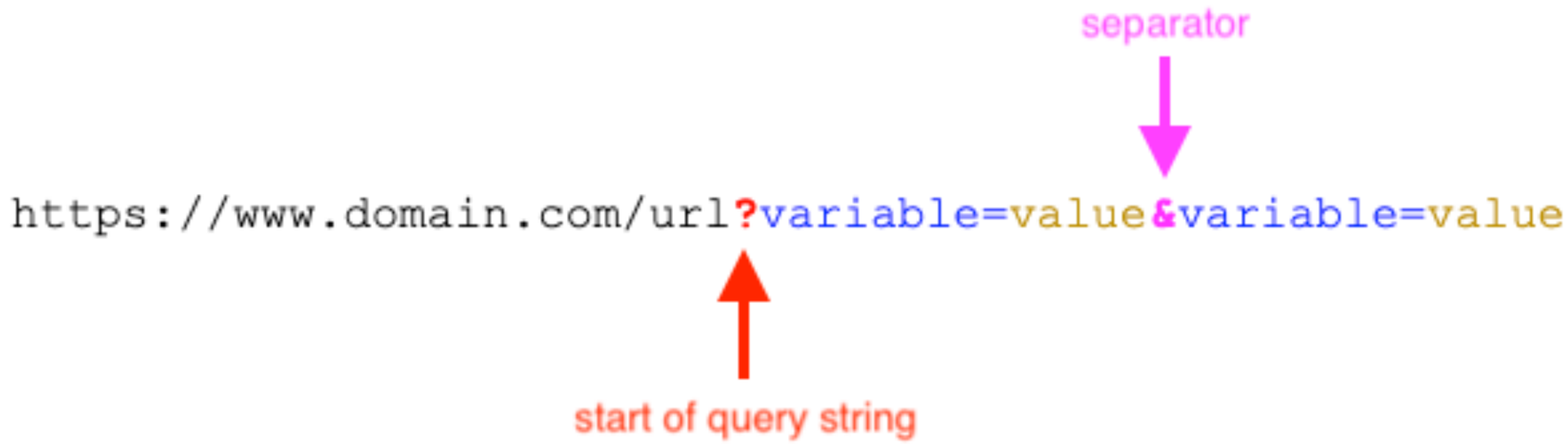


Параметры запросов



Про параметры

Query




```
Uri(
  …
  queryParameters: {'search': ‘Sochi'}
  …
);
```

Headers

```
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Host: vk.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15
    (KHTML, like Gecko) Version/15.1 Safari/605.1.15
Accept-Language: en-us
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
```

```
http.get(
        …
       headers: {'Auth': ‘alexs’},
        …
);
```

Тело запроса

```
http.get(
        …
        body: ‘some string’,
        …
);
```

Клиент

```dart
final client = http.Client();
```

```dart
client.
```

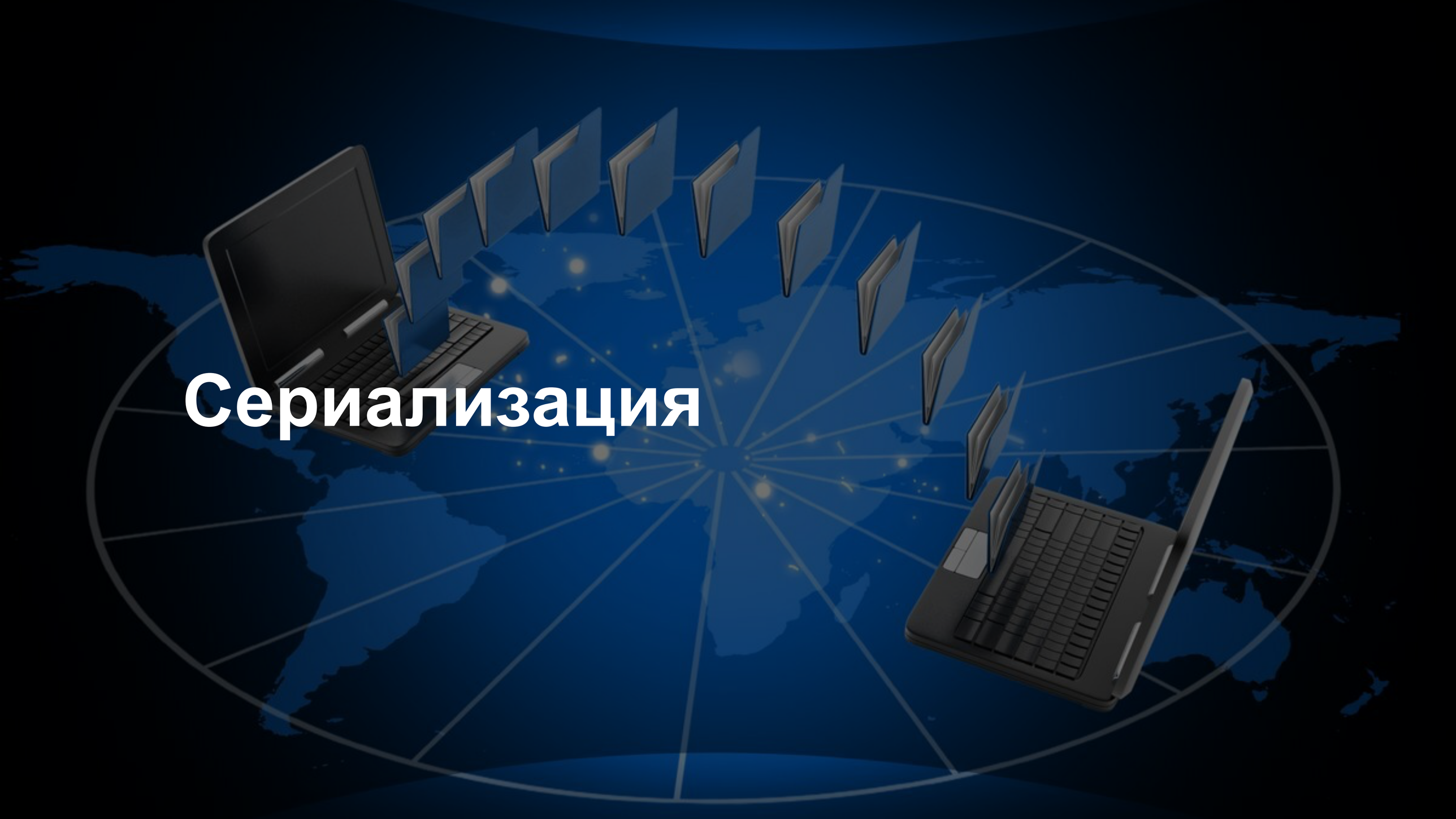
```
send(BaseRequest request)                    Future<StreamedResponse>
delete(Uri url, {Map<String, String>? hea…   Future<Response>
close()                                                        void
get(Uri url, {Map<String, String>? header…   Future<Response>
head(Uri url, {Map<String, String>? heade…   Future<Response>
patch(Uri url, {Map<String, String>? head…   Future<Response>
post(Uri url, {Map<String, String>? heade…   Future<Response>
put(Uri url, {Map<String, String>? header…   Future<Response>
read(Uri url, {Map<String, String>? headers…   Future<String>
readBytes(Uri url, {Map<String, String>?…   Future<Uint8List>
```

```dart
client.send(request);
```

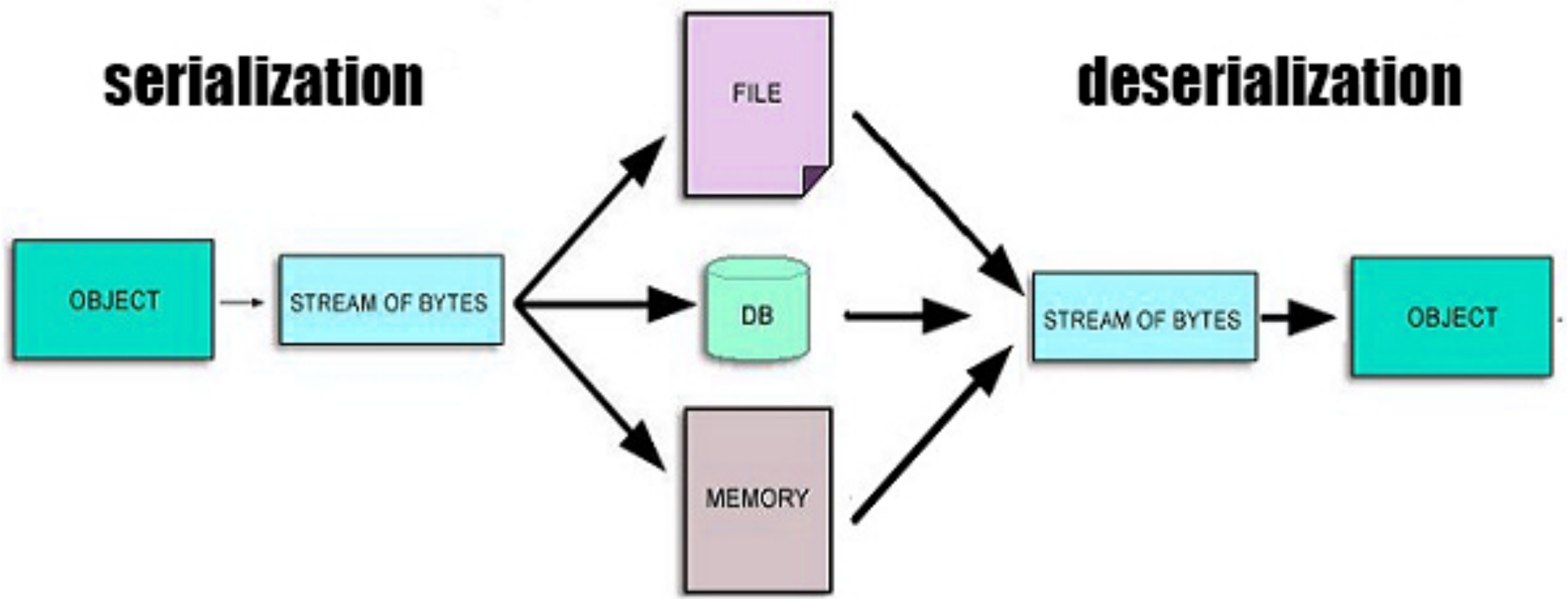
MultipartRequest extends BaseRequest

```
final multipart =  http.MultipartRequest('POST', uri);
multipart.files.add(MultipartFile(...));

client.send(multipart);
```

Сериализация



Сериализация



JSON

JSON generated from JSONC.json

```
{
    "foo": "bar foo",
    "true": false,
    "number": 42,
    "array": [1, 2, 3]
}
```

Dart

Из коробки

1. Импорт

```dart
import 'dart:convert';
```

2. Сериализация

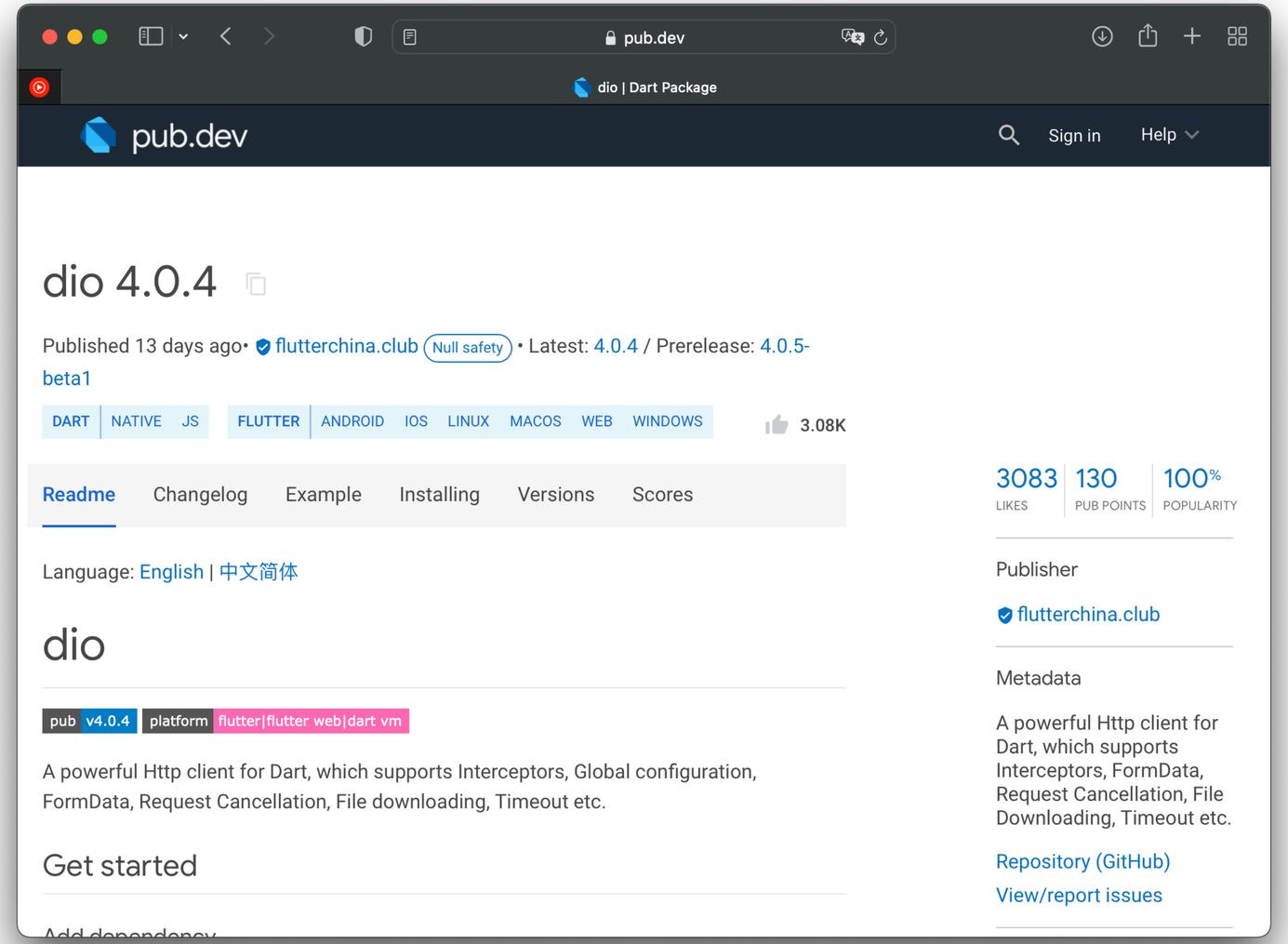
```dart
jsonEncode(…);  // передаем Map, получаем строку
```

3. Десериализация

```dart
jsonDecode(…);  // передаем строку, получаем Map
```

хочется чего-то большего

dio


pub.dev
dio | Dart Package
pub.dev
Sign in
Help
dio 4.0.4
Published 13 days ago• flutterchina.club Null safety • Latest: 4.0.4 / Prerelease: 4.0.5-beta1
DART NATIVE JS
FLUTTER ANDROID IOS LINUX MACOS WEB WINDOWS
3.08K
Readme Changelog Example Installing Versions Scores
Language: English | 中文简体
dio
pub v4.0.4 platform flutter|flutter web|dart vm
A powerful Http client for Dart, which supports Interceptors, Global configuration, FormData, Request Cancellation, File downloading, Timeout etc.
Get started
Add dependency
3083 LIKES
130 PUB POINTS
100% POPULARITY
Publisher
flutterchina.club
Metadata
A powerful Http client for Dart, which supports Interceptors, FormData, Request Cancellation, File Downloading, Timeout etc.
Repository (GitHub)
View/report issues

dio

Как сделать запрос?

1. Подключи dio

```
dio: ^4.0.4
```

2. Импортируй

```
import 'package:dio/dio.dart';
```

3. Создай клиент

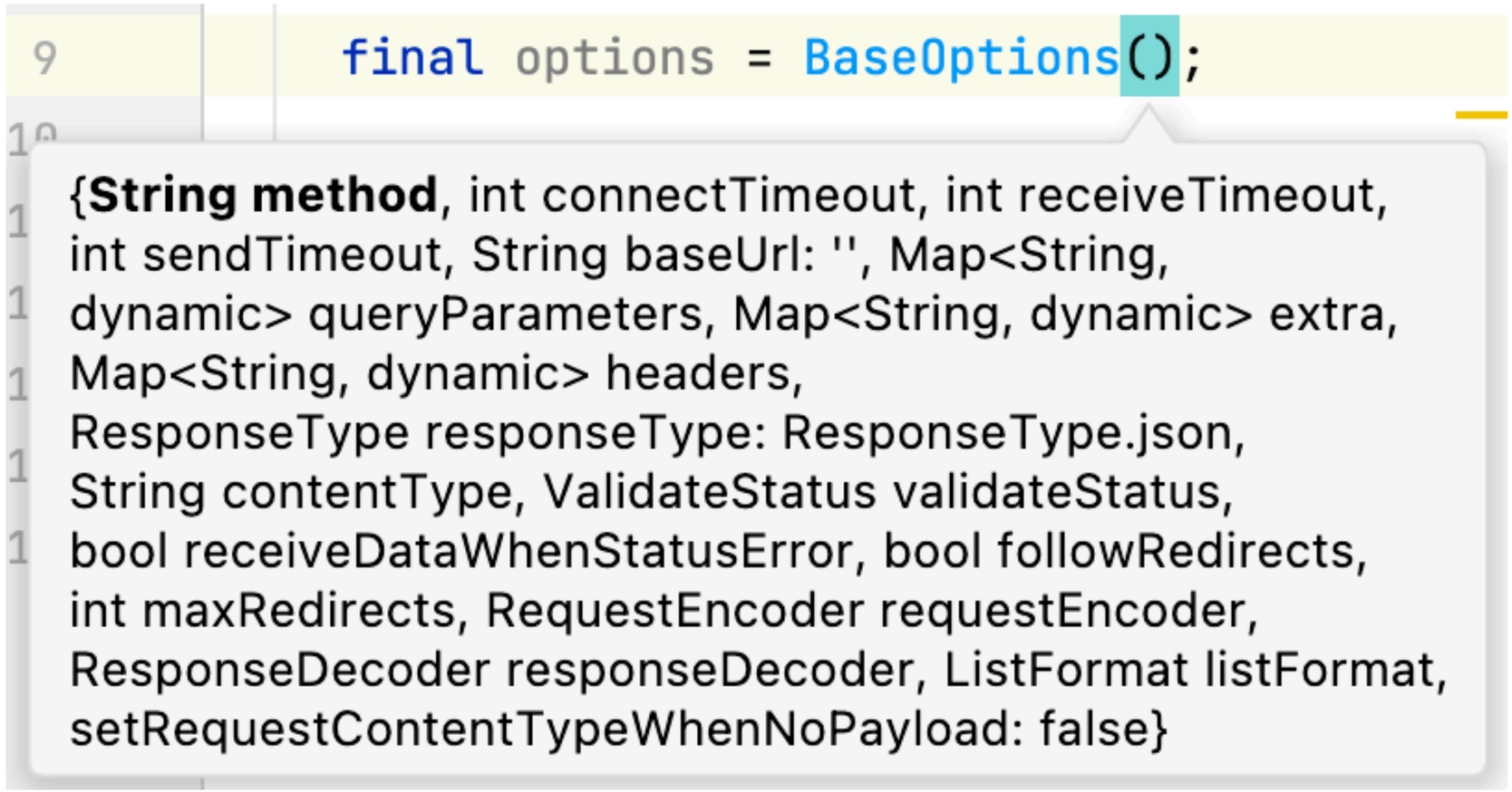
```dart
final Dio _dio = Dio();
```

4. Используй

```dart
_dio.get('/data');
```

Фичи dio

Конфигурация клиента




```
final Dio _dio = Dio(options);
```

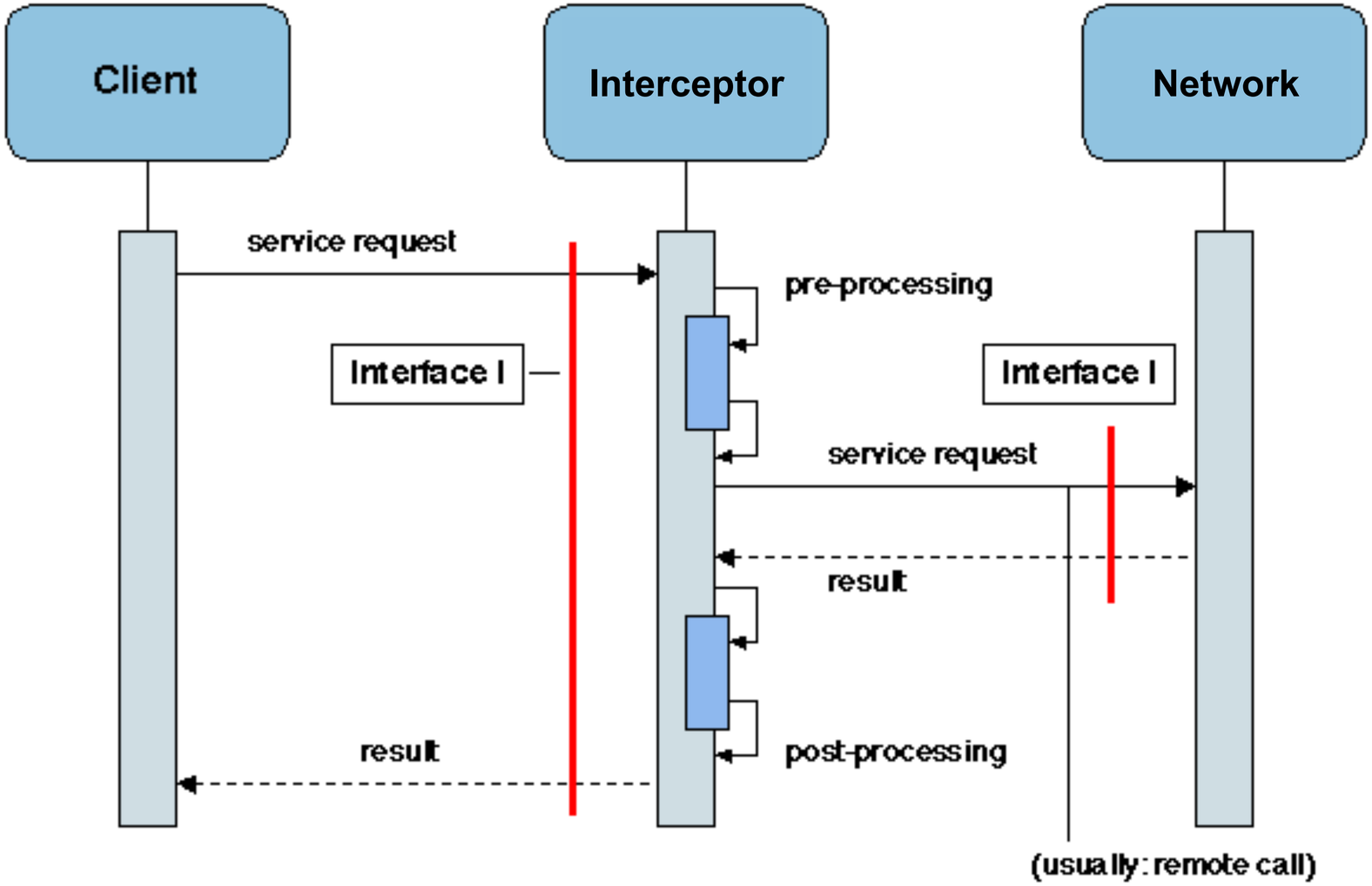
Десериализация // получаем Map

```
_dio.get<Map<String,dynamic>>('/data');
```

Сериализация // отправляем Map

```
_dio.post('/data', data: {
  'field': 'string'
});
```

Интерцепторы



Интерцепторы

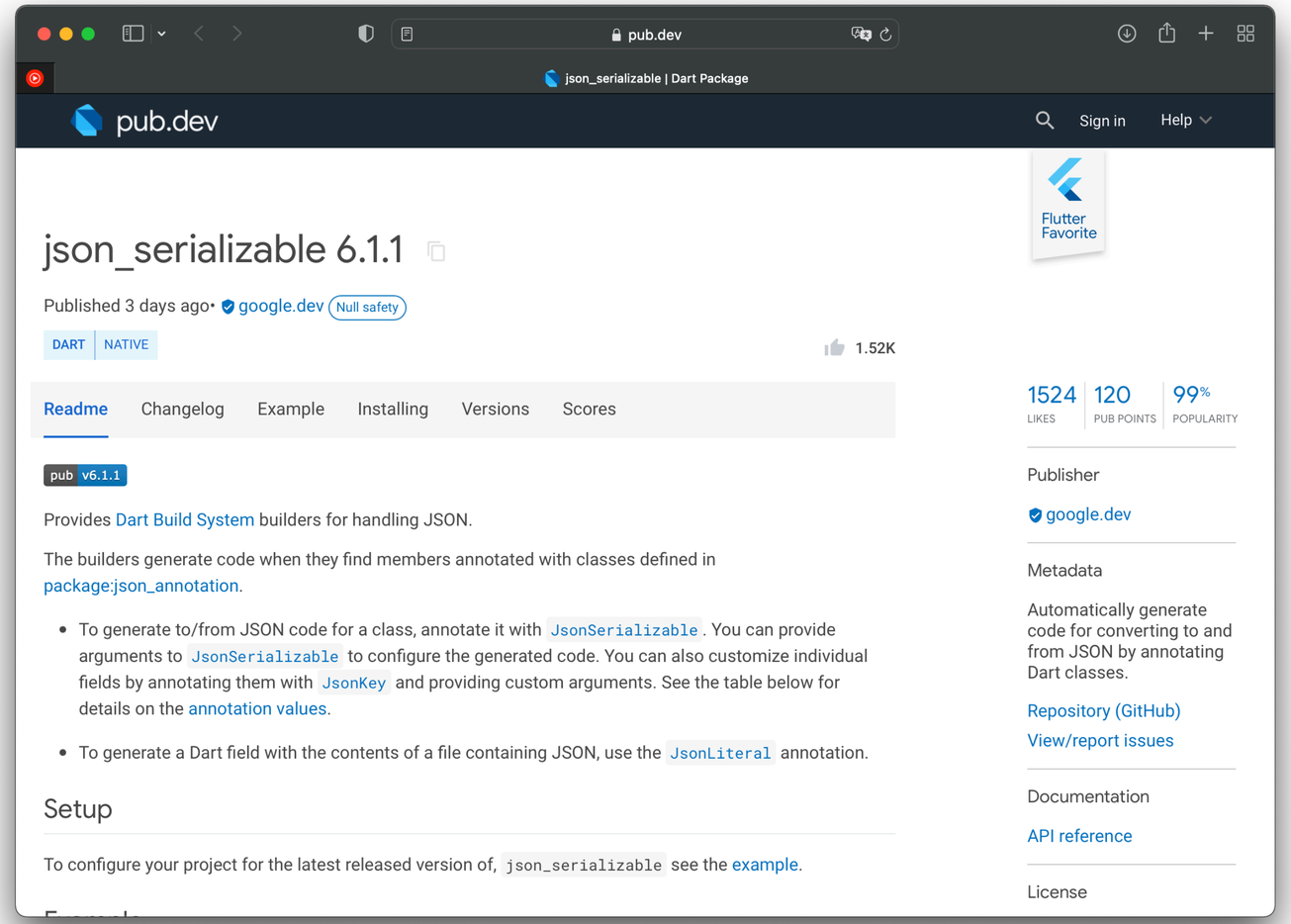
```
class AuthInterceptor extends Interceptor {

  …

  @override
  void onRequest(RequestOptions options, RequestInterceptorHandler handler) {
    final newOptions = options.copyWith(
      headers: {
        ...options.headers,
        'Auth': token
      }
    );
    handler.next(newOptions);
  }

  …

}
```

Обработка исключений

```
try {
  //make request
} on DioError catch (e){
  //handle error
}
```

json_serializable

```
// 20150625170715
// https://api.github.com/repos/tulios/json-viewer

{
  "id": 12635853,
  "name": "json-viewer",
  "full_name": "tulios/json-viewer",
  "owner": {
    "login": "tulios",
    "id": 33231,
    "avatar_url": "https://avatars.githubusercontent.com/u/33231?v=3",
    "gravatar_id": "",
    "url": "https://api.github.com/users/tulios",
    "html_url": "https://github.com/tulios",
    "followers_url": "https://api.github.com/users/tulios/followers",
    "following_url": "https://api.github.com/users/tulios/following{/other_user}",
    "gists_url": "https://api.github.com/users/tulios/gists{/gist_id}",
    "starred_url": "https://api.github.com/users/tulios/starred{/owner}{/repo}",
    "subscriptions_url": "https://api.github.com/users/tulios/subscriptions",
    "organizations_url": "https://api.github.com/users/tulios/orgs",
    "repos_url": "https://api.github.com/users/tulios/repos",
    "events_url": "https://api.github.com/users/tulios/events{/privacy}",
    "received_events_url": "https://api.github.com/users/tulios/received_events",
    "type": "User",
    "site_admin": false
  },
  "private": false,
  "html_url": "https://github.com/tulios/json-viewer"
```



json_serializable

кодогенерация.

1. Подключи это // dependencies

```
json_annotation: ^4.4.0
```

2. И это // dev_dependencies

```
json_serializable: ^6.1.1
build_runner: ^2.1.5
```

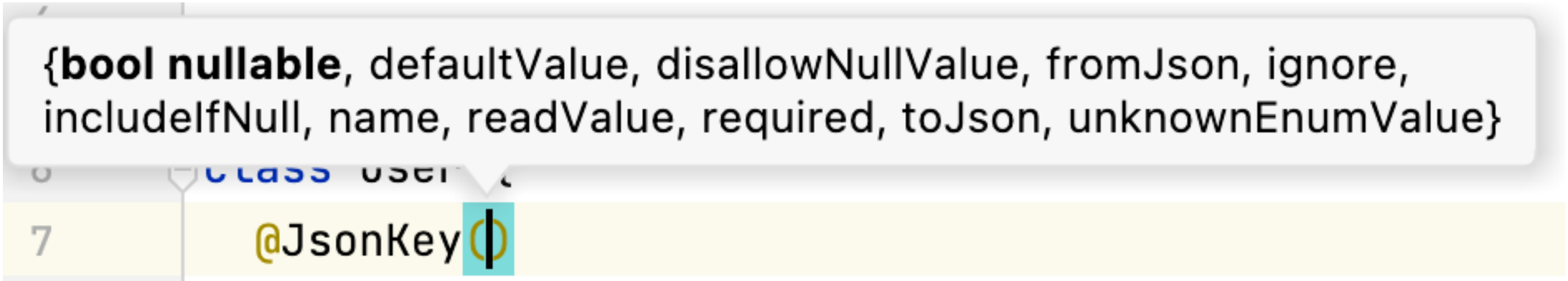
3. Опиши модельку

```dart
import 'package:json_annotation/json_annotation.dart';

part 'user.g.dart';

@JsonSerializable()
class User {
  final String name;
  final String email;

  User(this.name, this.email);

  factory User.fromJson(Map<String, dynamic> json) => _$UserFromJson(json);
  Map<String, dynamic> toJson() => _$UserToJson(this);
}
```

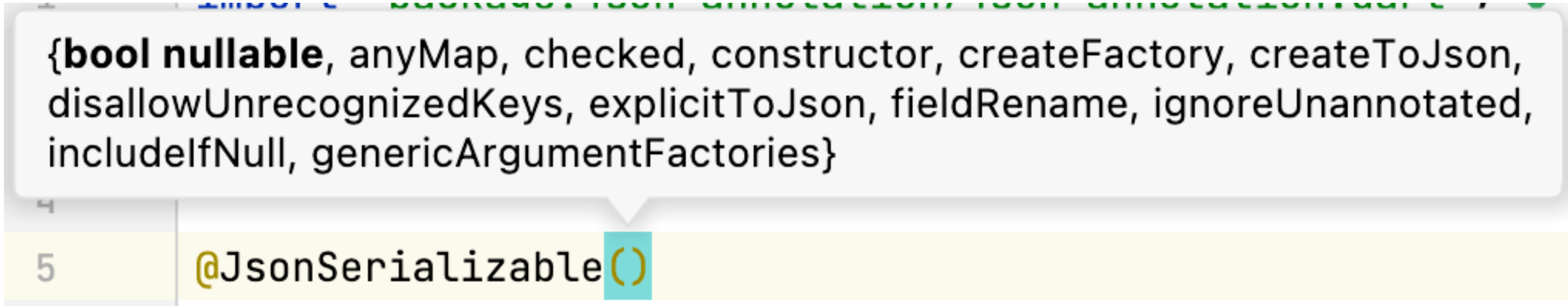
4. Запусти кодогенерацию

```
flutter pub run build_runner build
```

@JsonKey Аннотация поля



@JsonSerializable Аннотация класса



Всякое, всякое

Всякие пакеты

Яндекс Дзен

Chopper

Http Клиенты

Retrofit

Модельки

Freezed

Итог

1. Работа с сетью
2. http
2. Сериализация
4. dio
5. json_serializable
6. Всякое

Яндекс

Спасибо

Алексей Шведчиков

Разработчик

a1exs@yandex-team.ru

@a1ex5