



```
{
  "timestamp": 1585830391982,
  "status": 401,
  "error": "Unauthorized",
  "message": "Bad credentials",
  "path": "/oauth/token"
}
```

#### Прочие ошибки

```
{
  "error": "unsupported_grant_type",
  "error_description": "Unsupported grant type: client_credentials1"
}
```

#### Примерт выполнения запроса к API

##### Запрос на получение списка городов

```
GET https://agorafreight.com/external/api/v1/cities
Authorization: bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzY29wZSI6WyJleHRlcm5hbCJdLCJleHAiOiJlODU5MTYwOTAsImF1dGhvcml0aWVzIjpbIkpVYVEVSTkFMIl0sImp0aSI6ImQ2M2JjNDhhLTMzZWmtNGJiMS05NTk2LWExN2Q5MTkZDIyMCI6ImNsaWVudF9pZCI6InRlc3QifQ.OkdGFIKubJtQzBA7B148qRQZQ14Miy4RpsqdvvAoZtvPtJZV1NT7rTT1E9u7PC0ASBUAG_rknWq2qrXWg10a4wV2y0DWuvU56EQMTIq_SW6r_EOd87OClnQCyWajPVicFDoIygZSRatYbb9K6vaQHWtn0nIJL0_UM82XP0UfUz2BQ_yWo2bvXhcHYn-D_M8yUEp4GqgQEmAJlS3-MFtqnxX8xSvqZfyOG100g61o43_T-Iuy0Zlr_FhcrYtSdSBRuNXfmxNupdsjCksBIuuvYZ6QdI7xvhwshrScpaPjboBlcr-MPiGqsnS7H5XoiEvrU--S1W57BuRn0ddf_3dvBg
Cache-Control: no-cache
```

В случае успеха придет ответ HTTP 200

В случае проблем с токеном придет ответ HTTP 401 с примерно таким содержимым

```
{
  "error": "invalid_token",
  "error_description": "Cannot convert access token to JSON"
}
```

#### **Список методов:**

URL <http://agorafreight.com/external/api/v1/> является базовым и далее в документации не указывается.

GET /cities – метод, возвращающий города в системе.

GET /railway-stations – метод, возвращающий железнодорожные станции.

POST /calculations – метод, возвращающий расчеты по заданному маршруту.

### ***Используемый протокол:***

Запрос должен осуществляться по протоколу HTTP.

Формат передачи входящих и исходящих данных - JSON.

Все поля JSON-объекта в запросе являются обязательными, если не указано обратное.

Входящие параметры должны передаваться прямо в теле POST запроса в виде строки в формате JSON.

### ***Описание методов, примеры использования:***

1. Города, в системе /cities

Описание каждого города включает externalId,name;region;country, где

externalId – id города в системе

name – название города в системе на английском языке

region – название региона в системе на английском языке

country – код страны

Пример ответа GET /cities приведен в приложении 1.

2. Железнодорожные станции /railway-stations

Описание каждой железнодорожной станции включает externalId, name, country :

externalId – id железнодорожной станции в системе

name – название города, представленного в виде словаря, где ключ – это код языка, а значение – название железнодорожной станции

country – код страны.

Пример ответа GET /railway-stations приведен в приложении 2.

3. Расчеты POST /calculations

В теле запроса передаются следующие параметры [PM1] [PV2] :

```

{
  "cargo": {
    "containers": [
      {
        "amount": 0,
        "type": "C20", // [C20,C40,C40HC]
        "weight": 0 // [kg] - kilogram
      }
    ],
    "generals": [
      {
        "amount": 0,
        "height": 0, // [m] - meter
        "length": 0, // [m] - meter
        "pallet": "IN_48x40", // [CM_120x80,CM_100x80]
        "type": "BOXES", // [BOXES,PALLETS]
        "weight": 0, // [kg] - kilogram
        "width": 0 // [m] - meter
      }
    ],
    "vehicle": {
      "bodyType": "REFRIGERATOR", // [REFRIGERATOR,TENT,TERM]
      "type": "T1_5" // [T1_5,T3,T5,T10,T20] in tons
    } [3] [PV4]
  },
  "fromLocation": {
    "airportCode": "string", // IATA code
    "cityId": 0, // API request
    "seaPortCode": "string", // Port code(for example, RULED)
    "stationId": 0 // API request
  },
  "shippingDate": "dd.mm.yyyy", [5] [PV6] [7] [PV8]
  "toLocation": {
    "airportCode": "string", // IATA code
    "cityId": 0, // API request
    "seaPortCode": "string", // Port code(for example, RULED)
    "stationId": 0 // API request
  }
}

```

В запросе должен быть указан один из типов груза.

Для начальной и конечной точек должен быть выбран один из вариантов: аэропорт, город, морской порт или железнодорожная станция.

Описание ответа [\[PM9\]](#) [\[PV10\]](#) [\[PM11\]](#) [\[PV12\]](#) :

```

{
  "code": 0,
  "data": [
    {
      "cost": {
        //Total cost
        "currency": "string",
        "value": 0
      },
      "nds": {
        //VAT
        "currency": "string",
        "value": 0
      },
      "pathParts": [
        //Transportation legs
        {
          "from": {
            "airportCode": "string", //IATA code
            "cityId": 0, //API request
            "seaPortCode": "string", //Port code(for example,
RULED)
            "stationId": 0 //API request
          },
          "serviceCategory": "string", //Type of service
          "services": [
            {
              "aviaCompany": "string", //Air Company name
              "seaLine": "string", //SeaLine name
              "service": "string" //Service name
            }
          ],
          "to": {
            "airportCode": "string", //IATA code
            "cityId": 0, //API request
            "seaPortCode": "string", //Port code(for example,
RULED)
            "stationId": 0 //API request
          },
          "transitTime": 0 //Leg transit time
        }
      ],
      "transitTime": 0 //All transit time
    }
  ],
  "message": "string",
  "status": "success"
}

```

Примеры запросов и ответов приведены в приложении 3.

## Виджет

Виджет Agorafreight реализует удобный и простой механизм расчета перевозок и оформления заказа на перевозку на стороннем сайте. Для авторизации используется пользователь с ролью CLIENT от организации, для которой создается виджет. Пример виджета со стилями, отличными от стиля Agorafreight:

### Установка:

Создать айфрейм который будет содержать страницу расчёта со следующими параметрами:

```
var origin = 'https://agorafreight.com',

iframe = document.createElement('iframe'),
    src = origin + '/book/wizard'
    + '&fromAddress=' + ''
    + '&toAddress=' + ''
    + '&fromCode=' + ''
    + '&toCode=' + ''
    + '&language=' + 'ru'
    + '&isWidget=' + 'true';

iframe.src = src;
iframe.title = 'Wizard';
iframe.width = '100%';
iframe.style = 'border:0;';
```

На событие load добавить обработчик, который будет отправлять название компании и цветовую схему приложению внутри айфрейма, а также сбрасывать его состояние в родительском интерфейсе:

```

var resetWidget = function() {
    button.setAttribute('disabled', 'disabled');
    container.style.display = 'none';
};
iframe.onload = function() {
    resetWidget();
    iframe.contentWindow.postMessage({
        organization: 'your_organization_name',
        colorScheme: {
            primaryColor: '#d6ae1e',
            primaryBgColor: '#ffcd00',
            primaryHoverColor: '#f4db83',
            primaryActiveColor: '#b09122',
            primaryActiveDarkenColor: '#836d1d',
            primaryDarkenColor: '#c19e21',
            iconsColorName: 'yellow',
            companyLogo: 'https://yourwebsite.com/img/logo.png',
        },
    }, origin);
};

```

Вызвать функцию `iFrameResize` из библиотеки `iframe-resizer` [\[PM13\]](#) [\[PV14\]](#) чтобы отслеживать изменения размеров окна в айфрейме:

```

window.iFrameResize({ heightCalculationMethod: 'lowestElement' },
    iframe);

```

Добавить на событие `'message'` внешнего окна обработчик, который будет манипулировать айфреймом в зависимости от сообщений которые приходят с сайта `agorafreight`:

```

window.addEventListener('message', function(e) {
    if (e.data === '[wizard] login') {
        button.removeAttribute('disabled');
    }
    if (e.data === '[wizard] exit') {
        container.style.display = 'none';
    }
    if (e.data === '[wizard] error') {
        resetWidget();
    }
});

```

Пример реализации приведен в приложении 4.

## Приложение 1. GET /cities

```
externalId;name;region;country
298;Abakan;"Republic of Khakassia";RU
1644;Abilene;Texas;US
1127;Achinsk;"Krasnoyarsk Krai";RU
299;Adler;"Krasnodar Krai";RU
1351;Afipsky;"Krasnodar Krai";RU
1229;Aksarka;"Yamalo-Nenets Autonomous Okrug";RU
1641;Aksu;"Xinjiang Uygur Autonomous Region";CN
1540;Alagir;"Republic of North Ossetia-Alania";RU
1340;Aleksin;"Tula Oblast";RU
1336;Alexandroupoli;;GR
908;Alexeyevka;"Belgorod Oblast";RU
1077;Aleysk;"Altai Krai";RU
1658;Almaty;"Alma-Ata's Oblast";KZ
1225;Alupka;Crimea;RU
300;Alushta;Crimea;RU
406;Amderma;"Nenets Autonomous Okrug";RU
1182;Amsterdam;"North holland";NL
```

[\[PM15\]](#) [\[PV16\]](#) [\[PM17\]](#) [\[PV18\]](#)

## Приложение 2. GET /railway-stations



```
{
  "data": [
    {
      "externalId": 1468,
      "name": {
        "ru": "",
        "en": "Baziha"
      },
      "country": "RU"
    },
    {
      "externalId": 1490,
      "name": {
        "ru": "",
        "en": "Batareynaya"
      },
      "country": "RU"
    },
    ...
  ],
  "status": "success",
  "code": 200
}
```

[PM19] [PV20]

### Приложение 3. POST /calculations

В ответе перечислены следующие параметры:

"cost" – общая стоимость

"nds" – НДС

"transitTime" – время

"pathParts" – элементы расчета

"from" – откуда

"to" – куда

"serviceCategory" – категория услуг

"services" – услуги

One container, 40, 15000 kg, Saint-Petersburg(city) - Moscow(city)

Перевозка 40 контейнера весом 15000 кг из Санкт-Петербурга(город) в Москву(город)

***REQUEST***

POST /api/v1/external/calculations

```
{
  "cargo": {
    "containers": [
      {
        "amount": 1,
        "type": "C40",
        "weight": 15000
      }
    ]
  },
  "fromLocation": {
    "cityId": 373
  },
  "shippingDate": "24.05.2019",
  "toLocation": {
    "cityId": 253
  }
}
```

## RESPONSE

```
{
  "data": [
    {
      "cost": {
        "value": 774,
        "currency": "USD"
      },
      "nds": {
        "value": 42,
        "currency": "USD"
      },
      "transitTime": 16,
      "pathParts": [
        {
          "from": {
            "cityId": 373
          },
          "to": {
            "stationId": 1900
          },
          "transitTime": 0,
          "serviceCategory": "Transportation",
          "services": [
            {

```

```

        "service": "Cargo transportation"
      }
    ]
  },
  {
    "from": {
      "stationId": 1900
    },
    "to": {
      "stationId": 1338
    },
    "transitTime": 6,
    "serviceCategory": "Railway transportation",
    "services": [
      {
        "service": "Railway transportation"
      }
    ]
  },
  {
    "from": {
      "stationId": 1338
    },
    "to": {
      "cityId": 253
    },
    "transitTime": 10,
    "serviceCategory": "Transportation",
    "services": [
      {
        "service": "Cargo transportation"
      }
    ]
  }
]
},
"status": "success",
"code": 200
}

```

One box, 1x1x1 m, 75 kg, Saint-Petersburg(city) - Moscow(city)

Перевозка 1 коробки сборного груза размером 1x1x1 м, весом 75 кг из Санкт-Петербурга(город) в Москву(город)

**REQUEST**

```
{
  "cargo": {
    "generals": [
      {
        "amount": 1,
        "height": 1,
        "length": 1,
        "type": "BOXES",
        "weight": 75,
        "width": 1
      }
    ]
  },
  "fromLocation": {
    "cityId": 373
  },
  "shippingDate": "24.05.2019",
  "toLocation": {
    "cityId": 253
  }
}
```

## RESPONSE

```
{
  "data": [
    {
      "cost": {
        "value": 739,
        "currency": "USD"
      },
      "nds": {
        "value": 107,
        "currency": "USD"
      },
      "transitTime": 4,
      "pathParts": [
        {
          "from": {
            "cityId": 373
          },
          "to": {
            "airportCode": "LED"
          },
          "transitTime": 0,
          "serviceCategory": "Transportation",

```

```
    "services": [
      {
        "service": "Cargo transportation"
      }
    ]
  },
  {
    "from": {
      "airportCode": "LED"
    },
    "to": {
      "airportCode": "LED"
    },
    "transitTime": 0,
    "serviceCategory": "Airport services",
    "services": [
      {
        "service": "Documentation fee"
      },
      {
        "service": "Terminal handling"
      }
    ]
  },
  {
    "from": {
      "airportCode": "LED"
    },
    "to": {
      "airportCode": "SVO-II"
    },
    "transitTime": 4,
    "serviceCategory": "Air transportation",
    "services": [
      {
        "service": "Air transportation"
      }
    ]
  },
  {
    "from": {
      "airportCode": "SVO-II"
    },
    "to": {
      "airportCode": "SVO-II"
    },
    "transitTime": 0,
    "serviceCategory": "Airport services",
    "services": [
      {
```

```

        "service": "Documentation fee"
      },
      {
        "service": "Terminal handling"
      }
    ]
  },
  {
    "from": {
      "airportCode": "SVO-II"
    },
    "to": {
      "cityId": 253
    },
    "transitTime": 0,
    "serviceCategory": "Transportation",
    "services": [
      {
        "service": "Cargo transportation"
      }
    ]
  }
]
},
...
],
"status": "success",
"code": 200
}

```

One container, 40, 15000 kg, Dalian(city) - Moscow(city)

Перевозка 40 контейнера весом 15000 кг из Даляня(город) в Москву(город)

## REQUEST

POST /api/v1/external/calculations

```
{
  "cargo": {
    "containers": [
      {
        "amount": 1,
        "type": "C40",
        "weight": 15000
      }
    ]
  },
  "fromLocation": {
    "cityId": 272
  },
  "shippingDate": "24.05.2019",
  "toLocation": {
    "cityId": 253
  }
}
```

## RESPONSE

```
{
  "data": [
    {
      "cost": {
        "value": 1660,
        "currency": "USD"
      },
      "nds": {
        "value": 0,
        "currency": "USD"
      },
      "transitTime": 56,
      "pathParts": [
        {
          "from": {
            "cityId": 272
          },
          "to": {
            "seaPortCode": "CNDLC"
          },
          "transitTime": 0,
          "serviceCategory": "Transportation",
          "services": [
            {
              "service": "Cargo transportation"
            }
          ]
        }
      ]
    }
  ]
}
```

```

    }
  ]
},
{
  "from": {
    "seaPortCode": "CNDLC"
  },
  "to": {
    "seaPortCode": "CNDLC"
  },
  "transitTime": 0,
  "serviceCategory": "Services in origin port",
  "services": [
    {
      "service": "OTHC services"
    },
    {
      "service": "DOC"
    },
    {
      "service": "Telex release"
    },
    {
      "service": "Seal fee"
    },
    {
      "service": "ENS"
    },
    {
      "service": "Export service"
    },
    {
      "service": "Port forwarding"
    }
  ]
},
{
  "from": {
    "seaPortCode": "CNDLC"
  },
  "to": {
    "seaPortCode": "FIKTK"
  },
  "transitTime": 54,
  "serviceCategory": "Freight",
  "services": [
    {
      "service": "Freight",
      "seaLine": "MAERSK"
    }
  ]
}

```



```

    ]
  },
  {
    "from": {
      "seaPortCode": "FIKTK"
    },
    "to": {
      "seaPortCode": "FIKTK"
    },
    "transitTime": 0,
    "serviceCategory": "Services in destination port",
    "services": [
      {
        "service": "DTHC"
      },
      {
        "service": "Port forwarding"
      }
    ]
  },
  {
    "from": {
      "seaPortCode": "FIKTK"
    },
    "to": {
      "cityId": 253
    },
    "transitTime": 2,
    "serviceCategory": "Transportation",
    "services": [
      {
        "service": "Cargo transportation"
      }
    ]
  }
]
},
],
"status": "success",
"code": 200
}

```

One box, 1x1x1 m, 75 kg, Dalian(city) - Moscow(city)

Перевозка 1 коробки сборного груза размером 1x1x1 м, весом 75 кг из Даляня(город) в Москву(город)

**REQUEST**

```
{
  "cargo": {
    "generals": [
      {
        "amount": 1,
        "height": 1,
        "length": 1,
        "type": "BOXES",
        "weight": 75,
        "width": 1
      }
    ]
  },
  "fromLocation": {
    "cityId": 272
  },
  "shippingDate": "24.05.2019",
  "toLocation": {
    "cityId": 253
  }
}
```

## RESPONSE

```
{
  "data": [
    {
      "cost": {
        "value": 974,
        "currency": "USD"
      },
      "nds": {
        "value": 10,
        "currency": "USD"
      },
      "transitTime": 56,
      "pathParts": [
        {
          "from": {
            "cityId": 272
          },
          "to": {
            "seaPortCode": "CNDLC"
          },
          "transitTime": 0,

```

```
"serviceCategory": "Transportation",
"services": [
  {
    "service": "Cargo transportation"
  }
]
},
{
  "from": {
    "seaPortCode": "CNDLC"
  },
  "to": {
    "seaPortCode": "CNDLC"
  },
  "transitTime": 0,
  "serviceCategory": "Services in origin port",
  "services": [
    {
      "service": "VGM"
    },
    {
      "service": "OTHC services"
    },
    {
      "service": "DOC"
    },
    {
      "service": "ustoms clearance charge"
    },
    {
      "service": "ENS"
    },
    {
      "service": "Telex release"
    },
    {
      "service": "Solas admin"
    },
    {
      "service": "Lift On Lift Off / HDL"
    },
    {
      "service": "Port forwarding"
    }
  ]
},
{
  "from": {
    "seaPortCode": "CNDLC"
  },
  }
```

```
"to": {
  "seaPortCode": "RULED"
},
"transitTime": 54,
"serviceCategory": "Freight",
"services": [
  {
    "service": "Freight",
    "seaLine": "CMA CGM Log"
  }
]
},
{
  "from": {
    "seaPortCode": "RULED"
  },
  "to": {
    "seaPortCode": "RULED"
  },
  "transitTime": 0,
  "serviceCategory": "Services in destination port",
  "services": [
    {
      "service": "Port forwarding"
    },
    {
      "service": "VGM"
    },
    {
      "service": "DTHC"
    },
    {
      "service": "DOC"
    },
    {
      "service": "ISPS local"
    }
  ]
},
{
  "from": {
    "seaPortCode": "RULED"
  },
  "to": {
    "cityId": 373
  },
  "transitTime": 0,
  "serviceCategory": "Transportation",
  "services": [
    {
```

```

        "service": "Cargo transportation"
      }
    ]
  },
  {
    "from": {
      "cityId": 373
    },
    "to": {
      "cityId": 253
    },
    "transitTime": 2,
    "serviceCategory": "Transportation",
    "services": [
      {
        "service": "Cargo transportation"
      }
    ]
  },
  {
    "from": {
      "cityId": 253
    },
    "to": {
      "cityId": 253
    },
    "transitTime": 0,
    "serviceCategory": "Transportation",
    "services": [
      {
        "service": "Cargo transportation"
      }
    ]
  }
],
],
"status": "success",
"code": 200
}

```

One container, 40, 15000 kg, New-York(city) - Moscow(city)

[PM21] [PV22]

Перевозка 40 контейнера весом 15000 кг из Нью-Йорка(город) в Москву(город)

**REQUEST**

POST /api/v1/external/calculations

```
{
  "cargo": {
    "containers": [
      {
        "amount": 1,
        "type": "C40",
        "weight": 15000
      }
    ]
  },
  "fromLocation": {
    "cityId": 1571
  },
  "shippingDate": "24.05.2019",
  "toLocation": {
    "cityId": 253
  }
}
```

## RESPONSE

```
{
  "data": [
    {
      "cost": {
        "value": 4213,
        "currency": "USD"
      },
      "nds": {
        "value": 179,
        "currency": "USD"
      },
      "transitTime": 36,
      "pathParts": [
        {
          "from": {
            "cityId": 1571
          },
          "to": {
            "cityId": 1571
          },
          "transitTime": 2,
          "serviceCategory": "Origin charges",
          "services": [
            {

```

```
        "service": "Chassis fee"
      },
      {
        "service": "VGM"
      },
      {
        "service": "Document fee"
      }
    ]
  },
  {
    "from": {
      "cityId": 1571
    },
    "to": {
      "seaPortCode": "RULED"
    },
    "transitTime": 30,
    "serviceCategory": "Truck + Sea freight",
    "services": [
      {
        "service": "Truck + Sea freight"
      }
    ]
  },
  {
    "from": {
      "seaPortCode": "RULED"
    },
    "to": {
      "seaPortCode": "RULED"
    },
    "transitTime": 2,
    "serviceCategory": "Services in destination port",
    "services": [
      {
        "service": "DTHC"
      },
      {
        "service": "Import service"
      }
    ]
  },
  {
    "from": {
      "cityId": 373
    },
    "to": {
      "cityId": 253
    },
  },
```

```

        "transitTime": 2,
        "serviceCategory": "Transportation",
        "services": [
            {
                "service": "Cargo transportation"
            }
        ]
    },
    ],
    "status": "success",
    "code": 200
}

```

## Приложение 4. Пример реализации виджета

[\[PM23\]](#) [\[PV24\]](#)

```

!function() {
    var button = document.getElementById('button'),
        origin = 'https://agorafreight.com',
        container = document.getElementById('container'),
        iframe = document.createElement('iframe'),
        src = origin + '/book/wizard'
            + '&fromAddress=' + ''
            + '&toAddress=' + ''
            + '&fromCode=' + ''
            + '&toCode=' + ''
            + '&language=' + 'ru'
            + '&isWidget=' + 'true';

    var resetWidget = function() {
        button.setAttribute('disabled', 'disabled');
        container.style.display = 'none';
    };

    iframe.onload = function() {
        resetWidget();
        iframe.contentWindow.postMessage({
            organization: 'your_organization_name',
            colorScheme: {
                primaryColor: '#d6aele',
                primaryBgColor: '#ffcd00',
                primaryHoverColor: '#f4db83',
                primaryActiveColor: '#b09122',
                primaryActiveDarkenColor: '#836d1d',
                primaryDarkenColor: '#c19e21',
            }
        });
    };
}

```



```

        iconsColorName: 'yellow',
        companyLogo: 'https://yourwebsite.com/img/logo.png',
    }
    }, origin);
};

iframe.src = src;
iframe.title = 'Wizard';
iframe.width = '100%';
iframe.style = 'border:0;';

window.iFrameResize({ heightCalculationMethod: 'lowestElement' },
iframe);
container.appendChild(iframe);

window.addEventListener('message', function(e) {
    if (e.data === '[wizard] login') {
        button.removeAttribute('disabled');
    }

    if (e.data === '[wizard] exit') {
        container.style.display = 'none';
    }

    if (e.data === '[wizard] error') {
        console.log(e.data);
        resetWidget();
    }
});

button.addEventListener('click', function() {
    var src = '/book/wizard'
        + '?fromWhere=' + 'landing'
        + '&fromAddress=' + ''
        + '&toAddress=' + ''
        + '&fromCode=' + 'DE'
        + '&toCode=' + 'RU'
        + '&isWidget=' + 'true';

    iframe.contentWindow.postMessage({ pathname: src }, origin);
    container.style.display = 'block';
});

resetWidget();
}());

```

**[PM1]** Можно разобрать подробнее пример? Что мы хотим сделать этим запросом в двух словах?

**[PV2]** Это каракас запроса, который фактически инициирует расчет

[PM3] Не помню чтоб это можно было указывать в интерфейсе

[PV4] Это магистральная перевозка

[PM5] Не помню где указывается в интерфейсе

[PV6] Был убрана впоследствии

[PM7] Почему тогда оно попало в документацию?

[PV8] Без нее не будет работать. Можно убрать, но придется на беке проставлять текущую дату.

[PM9] Интерпретация ответа добавила бы информативности данному куску документации

Типа:

Из ответа понятно, что общая стоимость такая-то груз проедет там-то и там-то... и всякая подобная вещь

[PM10] Это фактически представление текущего расчета. Для каждого элемента дано пояснение в приложении 3.

[PM11] Увидел интерпретацию для запросов, а для ответов?

[PV12] В начале Приложения 3 указана расшифровка

[PM13] Ставится отдельно владельцем сайта?

[PV14] Думаю, что да. Установкой не занимался, пример есть на сайте Аистлога

[PM15] Это действительно такой ответ? Невалидный жсон?

[PV16] Фактически это csv

[PM17] Я это понимаю. Вопрос был: это отображается как json, с csv внутри тела объекта?

Я вижу {}

и я вижу string:string:string внутри {}

Это json внутри тела которого записаны строки в формате csv?

Если да, то зачем так сделано?

[PM18] По аналогии с другими организациями, предоставляющими внешнее API. По-моему, у ПЭК такой же формат. Фигурные скобки убрал.

[PM19] т.е. GET /railway-stations вернет все станции из системы? Параметров пагинации нет?

[PV20] Вернет все. Это отдельный эндпойнт, не связанный с эндпойнтом основного приложения

[PM21] Интерпретация ответов добавит информативности и смысла данным строкам в документации.

Здесь и примерам выше.

Еще было бы понятнее если бы были такие же комментарии как выше

Сейчас эти строки ничего в целом не говорят.

[PV22] Добавил пояснения в начале

[PM23] Это типовая настройка?

[PV24] Это в качестве примера оставил фронтенд-разработчик