G2Rail API Document

G2Rail API Document

Introduction

This article explains the APIs of G2Rail and how to use them to Search, Book, Confirm and download tickets of Railways (DB Deutsche Bahn, China Railway(CR), Trenitalia(Italy), Italo(Italy), Renfe(Spain), iryo (Spain), Korail, JR, Taiwan high speed rail, SBB(Switzerland), ÖBB(Austria), NSB(Norway), Flixbus(Europe, USA, Canada, Turkey, Brazil) and so on.

You only need five APIs to accomplish the very basic ticket search and booking scenario including **Search**, **Book**, **Confirm**, **Download** and **Refund** tickets.

Examples

- Go (https://github.com/G2Rail/g2rail-grpc-go)
- Java (https://github.com/G2Rail/g2rail-grpc-java)
- C# (https://github.com/G2Rail/g2rail-grpc-net)

Search for Journey Solutions

The following is the request and response message for search journey from 11:00 am local time, on Mar. 08, 2017, from Milano Centrale to Roma Termini.

> Make sure provide security params with each request

> Use async query to retrieve the response.

Search Request

```
GET /api/v2/online_solutions
```

This is an async call, an async-key will be returned. Try to use

```
GET /api/v2/async_results/{async_key}
```

to retrieve response data.

G2Rail supports search by city and search by station.

Search Timebox

By default, each search will return journey solutions of next 3 hours.

Search-Book-Confirm

The whole process from Searching to Confirmation should be finished within 20 minutes. Otherwise, you have to search result expire error and you need to do another search. This is because we have live data including timetable, price, remaining seats and it is so dynamic.

Search by City

Just pass city name to From and To, you can retrieve journey solution between departure city and destination city. The following is a search request for one adult traveler from Roma to Milano.

```
{
  "from": "CT_60Y990YWR",
  "to": "CT_LV7D4WNOK",
  "date": "2017-03-08",
  "time": "11:00",
  "duration": 3,
  "adult": 1,
  "child": 0,
  "junior": 0,
  "senior": 0,
  "child": 0,
  "infant": 0
}
```

Note : Many cities (such as Berlin, Frankfurt, Roma, Paris) support search from/to all stations in that city. For these cities the search response will be from/to all possible stations. Some other cities don't support this, for these cities, the search response will be from main station (Central or Hbf) of the city.

Tutte station of City If the station name is shown in Capitial letter, that means the searching result includes all the station in that city. For example: MILANO, BERLIN, PARIS. Such stations are virtual station to solve the problem of not-knowing which station to search from. For instance, if traveler searches for journeys from Berlin to Paris, he needs to specify Paris Est. because Germany Railway (DB) calls at Paris Est. Station. If travel searches for journeys from Milano to Paris, he need to set Paris Lyon station as destination. To make it simple, just pass PARIS or use City, we will take care the station mapping.

So if you search with City name, the logic is like this, if City has Tutte station, it will search via Tutte station, otherwise, it will search fro, main station. For example München doesn't have a Tutte, so the search from München Hbf station.

Search by Station

Certainly if travel are experienced, he can use statin search also. We have 100,000 stations in database. The following is request json for a search request for an adult traveler, from 11:00 am on Mar. 08, 2017, departures from Milano Centrale (station code: 'ST_L6NN3P6K') and arrives in Roma Termini (station code: 'ST_LX225YVP').

```
{
  "from": "ST_L6NN3P6K",
  "to": "ST_LX225YVP",
  "date": "2017-03-08",
  "time": "11:00",
  "duration": 3,
  "adult": 1,
  "child": 0,
  "junior": 0,
  "senior": 0,
  "child": 0,
  "infant": 0
}
```

Parameters

Parameter	Type	Description	Required?
from	Departure station code	string	Yes
to	Destination station code	string	Yes
date	Departure date, format: yyyy-MM-dd	string	Yes
time	Departure time, format: HH:mm	string	No
duration	Search timebox, default 3 hours	int	No
adult	Number of adults	int	Yes
child	Number of children	int	Yes

About the age of children in the booking process

Different railway companies have different regulations and policies for the age of babies and children. Therefore, it is necessary to enter the specific date of birth during the booking process. The suitable ticket would be determined by each railway company system. For example, DB will calculate the ages at the Booking stage. The purpose of the searching stage is more like to get a train timetable and price range of tickets. In this situation, you only need to enter the number of children.

Search Response

The following is response json for a search request from Berlin Hbf to München Hbf. Because there are two carriers so far, we will receive two array of journey solutions, one for DB and the other for Flixbus.

```
[
    {
      "railway": {
        "code": "TI"
      },
      "solutions": [
        {
          "from": {
            "code": "ST_L6NN3P6K",
            "name": "Milano Centrale",
            "local_name": "Milano Central Station",
            "help_url": "http://help.g2rail.com/cn/station/milano-centrale"
          },
          "to": {
            "code": "ST_LX225YVP",
            "name": "Roma Termini",
            "local_name":  "Roma Termini",
            "help_url": "http://help.g2rail.com/cn/station/roma-termini"
          },
          "departure": "2017-03-08T11:00:00+01:00",
          "duration": {
            "hour": 2,
            "minutes": 59
          },
          "transfer_times": 0,
          "sections": [
            {
              "offers": [
                {
                  "code": "TI_T01",
                  "description": "Base",
                  "detail": "Change the train connection or cancel the ticket, refund fee 3 euro before
departure. Additionally 20% of the ticket fare will be charged as cancellation fee",
                  "ticket_type": "E-ticket",
                  "seat_type": "non-reserved seat",
                  "refund_type": "Refundable",
                  "change_type": "Exchangeable",
                  "confirm_again": "no_need",
                  "restriction": {
                      "code": "ATOC_00000",
                      "description": "ANY PERMITTED",
                      "detail": "You may take any route permitted by the NRCoT or Routeing Guide"
                    },
                  "services": [
                    {
                      "code": "TI_C06",
                      "description": "Standard",
                      "detail": "Standard",
                      "available": {
                        "seats": 298
```

```
    },
    "average_unit_price": { "currency": "EUR", "cents": 9100 },
    "price": { "currency": "EUR", "cents": 9100 },
    "booking_code": "bc_01"
  },
  {
    "code": "TI_C05",
    "description": "Premium",
    "detail": "Premium",
    "available": {
      "seats": 76
    },
    "average_unit_price": { "currency": "EUR", "cents": 10700 },
    "price": { "currency": "EUR", "cents": 10700 },
    "booking_code": "bc_02"
  },
  {
    "code": "TI_C04",
    "description": "Business Area Silenzio",
    "detail": "Business Area Silenzio",
    "available": {
      "seats": 22
    },
    "average_unit_price": { "currency": "EUR", "cents": 12200 },
    "price": { "currency": "EUR", "cents": 12200 },
    "booking_code": "bc_03"
  },
  {
    "code": "TI_C03",
    "description": "Business",
    "detail": "Business",
    "available": {
      "seats": 43
    },
    "average_unit_price": { "currency": "EUR", "cents": 122000 },
    "price": { "currency": "EUR", "cents": 122000 },
    "booking_code": "bc_04"
  },
  {
    "code": "TI_C01",
    "description": "Executive",
    "detail": "Executive",
    "available": {
      "seats": 10
    },
    "average_unit_price": { "currency": "EUR", "cents": 22000 },
    "price": { "currency": "EUR", "cents": 22000 },
    "booking_code": "bc_05"
  }
]
```

```
    },
    {
      "code": "TI_T03",
      "description": "Super Economy",
      "detail": "non refundable, non changeable",
      "ticket_type": "Mobile voucher",
      "seat_type": "reserved seat",
      "refund_type": "Non-refundable",
      "confirm_again": "no_need",
      "change_type": "Not exchangeable",
      "restriction": {
          "code": "ATOC_00000",
          "description": "ANY PERMITTED",
          "detail": "You may take any route permitted by the NRCoT or Routeing Guide"
        },
      "services": [
        {
          "code": "TI_C06",
          "description": "Standard",
          "detail": "Standard",
          "available": {
            "seats": 33
          },
          "average_unit_price": { "currency": "EUR", "cents": 1900 },
          "price": { "currency": "EUR", "cents": 1900 },
          "booking_code": "bc_01"
        },
        ......
      ]
    },
    ......
  ],
  "trains": [
    {
      "number": "FR9623",
      "type": "FR",
      "from": {
       "code": "ST_L6NN3P6K",
      "name": "Milano Centrale",
      "local_name": "Milano Central Station",
      "help_url": "http://help.g2rail.com/cn/station/milano-centrale"
      },
      "to": {
        "code": "ST_LX225YVP",
        "name": "Roma Termini",
        "local_name":  "Roma Termini",
        "help_url": "http://help.g2rail.com/cn/station/roma-termini"
      },
      "departure": "2017-03-08T11:00:00+01:00",
      "arrival": "2017-03-08T13:59:00+01:00",
```

```
                    "stops": [
                      {
                        "code": "ST_D122M0N0",
                        "name": "Chiusi-Chianciano-Terme",
                        "local_name":  "Roma Termini",
                        "help_url": "http://localhost:9093/zh-cn/stations/chiusi-chianciano-terme"
                      },
                      {
                        "code": "ST_LO881QPR",
                        "name": "Firenze Santa Maria Novella",
                        "local_name":  "Firenze Santa Maria Novella",
                        "help_url": "http://localhost:9093/zh-cn/stations/firenze-santa-maria-novella-sm
n"
                      },
                      {
                        "code": "ST_E7GG95NJ",
                        "name": "Monzuno Vado",
                        "local_name":  "Monzuno Vado",
                        "help_url": "http://localhost:9093/zh-cn/stations/monzuno-vado"
                      },
                      {
                        "code": "ST_L38834KK",
                        "name": "Bologna Centrale",
                        "local_name":  "Bologna Central",
                        "help_url": "http://localhost:9093/zh-cn/stations/bologna-centrale"
                      },
                      {
                        "code": "ST_L4GG9J7Z",
                        "name": "Reggio Emilia AV",
                        "local_name": "Reggio Emilia AV",
                        "help_url": "http://localhost:9093/zh-cn/stations/reggio-emilia-av"
                      },
                      {
                        "code": "ST_L4GG9W1Y",
                        "name": "Piacenza",
                        "local_name": "Piacenza",
                        "help_url": "http://localhost:9093/zh-cn/stations/piacenza"
                      }
                    ]
                  }
                ]
              }
            ]
          }
        ]
      },
      {
        "railway": {
          "code": "NTV"
        },
```

```
    "solutions": [
      {
        "from": {
          "code": "ST_L6NN3P6K",
          "name": "Milano Centrale",
          "local_name": "Milano Central Station",
          "help_url": "http://help.g2rail.com/cn/station/milano-centrale"
        },
        "to": {
          "code": "ST_LX225YVP",
          "name": "Roma Termini",
          "local_name":  "Roma Termini",
          "help_url": "http://help.g2rail.com/cn/station/roma-termini"
        },
        ......
      }
    ]
  },
  {
    "railway": {
      "code": "FB"
    },
    "solutions": [
      {
        "from": {
          "code": "ST_L6NN3P6K",
          "name": "Milano Centrale",
          "local_name": "Milano Central Station",
          "help_url": "http://help.g2rail.com/cn/station/milano-centrale"
        },
        "to": {
          "code": "ST_LX225YVP",
          "name": "Roma Termini",
          "local_name":  "Roma Termini",
          "help_url": "http://help.g2rail.com/cn/station/roma-termini"
        },
        ......
      }
    ]
  }

]
```


Search response parameters

Parameter	Description	Type
railway	Railway code, detail in **Railway Code** table. Since there may be multiple railway companies involved during train transfer, the Railway information is moved to the Carrier (code/description) under Section.	railway

Parameter	Description	Type
loading	bool	Whether still need to wait for journey data from this railway, “loading = false” means data is ready, “loading = true” means still need to wait for response from the railway
solutions	Array of journey solutions from different carries, details in **Solution** table	array

Railway Code

Railway	Name	Value
Italy	Trenitalia	TI
Germany	Deutsche Bahn	DB
Italy	Italo	NTV
Flixbus	Flixbus	FB
China Rail	China Rail	CR
Spain	Renfe	Renfe
France	SNCF	SNCF
South Korea	Korail	KR
Japan	Japan Railway	JR
Finland	VR	VR
Sweden	SJ	SJ
Norway	NSB	NSB
Portugal	CP	CP
Poland	PKP	PKP
Thailand	SRT	SRT

Solutions

Parameter	Description	Type
from	Departure station, Detail in **Station** table	station
to	Destination station，Detail in **Station** table	station
departure	Departure time，UTC format: “2017-03-08T13:30:00+01:00”	string
duration	Duration，Detail in **Duration** table	duration
transfer_times	Transfer Times	integer
sections	Sections, Detail in **Section** table	array

Station

Parameter	Description	Type
code	station code	string
name	station name	string
local_name	station name in local language	string
help_url	station info	string, url

Duration

Parameter	Description	Type
hour	# of hours	integer
minutes	# of minutes	integer

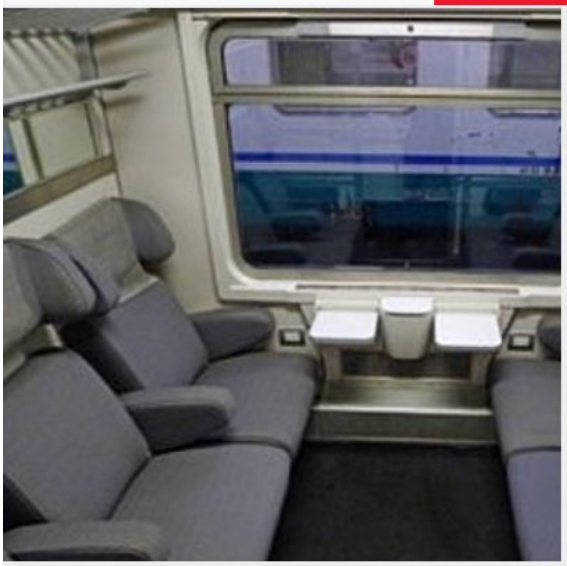
Section

Because different railway companies have different kinds of trains, and the offer and service (coach class), many railway companies split the entire journey solutions into multiple sections and for each section, the offers and services are the same. For example, one of Trenitalia's high speed train is called "Frecciargento", it has 5 different services including Executive, Business, Business Area Silenzio, Premium, Standard. It also has different kinds of offers such as Base,Economy,Super Economy.

When you are searching for the route, the concept of Section is more like the concept of ticketing. All the trains in the same section use the same ticket and choose the same class. Since there are many railways in Europe that can be used for arriving in different cities, They can be directly searched by us but there are several kinds of tickets involved in the whole process. For example, from Shanghai to Beijing. One is the subway ticket from Shanghai West Nanjing Road Metro Station to Shanghai Hongqiao Railway Station, one is the high-speed railway ticket from Shanghai Hongqiao to Beijing South Railway Station, and the last one is from Beijing South Railway Station to Xizhimen Metro station. There will be three sections in this case.

INTERCITY / REGIONAL	departure	Rome Termini (Main) 09:57	09:35 1 transfers	arrival	Milano Rho Fiera 19:32	**€ 73.85**

IC 35510	2nd Class	1st Class
Full Price	**€ 63.50** Sufficient	**€ 87.00** 84 Seats Left
Saver Price		**€ 62.90** 4 Seats Left
RV 2027	**2nd Class**	**1st Class**
Ordinary	**€ 10.95** Sufficient	**€ 16.35** Sufficient



Train Details

IC 35510		Rome Termini		Turin Porta
1st Class	departure	(Main)	transfer	Nuova (Main)
Saver Price		09:57		17:40
		Transfer Time: 14minutes		
RV 2027		Turin Porta	arrival	Milano Rho Fiera
2nd Class	transfer	Nuova (Main)		19:32
Ordinary		17:54		

Details

- Please print the ticket on A4 paper. After successful purchase of the ticket, you can download the ticket in the following places: 1. Email 2. Top right of the webpage (My Order), 3. Booking success page 4. APP (My Order)
- Change is possible
- Refundable
- Before the train departure, exchange fee for the other train is 20% of the ticket value
- Please follow the seat number on the ticket

1st Class:

- Fabric armchair with tilt function up to 117°
- The distance between the armchairs can reach 90 cm
- The spacious space of the suitcase
- Bar Bistro Service

Total Amount: **€ 73.85** Quote **Book**

Refer to the picture above, from the Rome Termini to the Milan Rho Fiera station. The first section is Trenitalia Intercity 23310 from Roma Termini to Turin Port Nuova, and the second section is Trenitalia Regionale Veloce 2027 from Turin Port Nuova to Milano Rho Fiera. Traveler can choose services individually in each section and possibly there are two tickets issued, one from IC 35501 and one for RV 2027.

Parameter	Description	Type
carrier_code	DB, EUR, TI	string
carrier_description	Deutsche Bahn, Euro Star, Trenitalia	string
offers	Combinations from different offers，details in **Offer** below	array
trains	Details in **Train**	array

Offer

Railway companies have different discount for their tickets, it is called Offer.

Common offer code, details inCommon Offer list (https://docs.g2rail.com/zh/pages/09_offers)

Parameter	Description	Type
code	Offer code	string
description	Offer description	string
detail	Offer detail	string
ticket_type	Ticket type, value could be one of **E-ticket/Mobile voucher/Paper voucher**	string
seat_type	Whether seat included, value could be one of **reserved seat/non-reserved seat**	string
refund_type	Whether this offer can be refunded, value could be one of **Non-refundable/Refundable**	string
change_type	Whether this offer can be changed, value could be one of **Not exchangeable/Exchangeable**	string
confirm_again	Whether the tickets need to be issue right away, possible values includes: **need** and **no_need**	string
restrictions	Route restrictions, details see **Route Restriction**	string
services	List of available services, detail in **Service** table	array

Confirm Again

Some offers may reqire an offline confirm such as group ticket, rail pass etc. Some railway companies also require this. It this is the case, the **confirm_again** is **need**. Otherwise it is **no_need**.

If **no_need**, this ticket will be issued within seconds. Otherwise (**need**), normally ticket will be issued within 30 mins.

There is also possibility that confirm_again is nil. This field in Search response is only for reference purpose. There is another Confirm_Again in Confirm response, that field is more accurate.

Note It is possible that the confirm_again in search result is **no_need** and when confirming, confirm_again becomes **need**. This is rare case, maybe caused by API error from railway company.

Route Restriction

For example in UK, there are multiple “Anytime”, “Off-Peak” offers for one journey, the difference is restriction such as “Any Permitted”, “WMTRAINS. ONLY”, “VIA STEVENAGE” etc.

Parameter	Description	Type
description	description	string
detail	detail information	string

Service

Many railway carriers have different service classes, we call it Service. For example, one of Trenitalia's high speed train is called "Frecciargento", it has 5 different services including Executive, Business, Business Area Silenzio, Premium, Standard. DB's high speed train is called "ICE", it has first class and second class.

Parameter	Description	Type
code	service code	string
description	description	string
detail	detail information	string
available	seat remaining, detail in **Available** table	available
price	price, detail in **Price** table	price
average_unit_price	average price per passenger, detail in **Price** table	price
booking_code	booking code to be used to in Book Request	string
booking_type	Differnt booking_type with different booking process，details in booking_type list below	string

booking_type

booking_type	Description
Online_Booking	directly using the BOOK and CONFIRM API below to finish the booking
Group_Booking	DB: If ticket no. >=6，Other Railway Company: ticket no. >=10,automatically change to Group Booking，"Group Inquiry" is needed to connect the process，details in Group Ticket (Hold and Book) Flow API Document (https://docs.g2rail.com/en/pages/01_group_booking)
Pass_Booking	Finish Booking through "Book Fixed Price Products"，details in Fixed price API file (https://docs.g2rail.com/zh/pages/02_products)

Available

Parameter	Description	Type
seats	# of seats remaining	integer

Price

Parameter	Description	Type
currency	Currency such as EUR, CNY, USD, HKD	string
cents	amount of price in cents (muitiplied by 100), for example the value of 39 EURO is **3900**	integer

Train

Parameter	Description	Type
number	train number such as "ICE 1609	string
type	such as "ICE" at DB carrier List (https://docs.g2rail.com/zh/pages/08_carriers)	string
from	departure station, detail in **Station** table	station
to	destination station, detail in **Station** table	station
departure	departure time in UTC time format such as :"2017-03-08T13:30:00+01:00"	string
arrival	arrival time in UTC time format such as: "2017-03-08T18:17:00+01:00"	string
stops	list of intermediate stops, detail in **Station** table	station

Error Code

Common Errors include:

System Error

Parameter	Description
internal_error	System busy
station_not_found	The city you specified could not be connected to the station. Please specify the specific station (such as Berlin Hbf) and then search again

The following is sample code to search for journey solutions from both Trenitalia and Italo for an adult traveler, on Mar 8th, 2017, from Berlin Hbf to München Hbf

```
Ruby
```

```
ruby#!/usr/bin/env ruby

require "digest/md5"
require 'time'
require 'net/http'
require "cgi"

require 'active_support/time'
require 'active_support/json'

class Object
  # Alias of <tt>to_s</tt>.
  def to_param
    to_s
  end

  # Converts an object into a string suitable for use as a URL query string,
  # using the given <tt>key</tt> as the param name.
  def to_query(key)
    "#{CGI.escape(key.to_param)}=#{CGI.escape(to_param.to_s)}"
  end
end

class NilClass
  # Returns +self+.
  def to_param
    self
  end
end

class TrueClass
  # Returns +self+.
  def to_param
    self
  end
end

class FalseClass
  # Returns +self+.
  def to_param
    self
  end
end

class Array
  # Calls <tt>to_param</tt> on all its elements and joins the result with
  # slashes. This is used by <tt>url_for</tt> in Action Pack.
  def to_param
    collect(&:to_param).join "/"
  end
```

```
  # Converts an array into a string suitable for use as a URL query string,
  # using the given +key+ as the param name.
  #
  #   ['Rails', 'coding'].to_query('hobbies') # => "hobbies%5B%5D=Rails&hobbies%5B%5D=coding"
  def to_query(key)
    prefix = "#{key}[]"

    if empty?
      nil.to_query(prefix)
    else
      collect { |value| value.to_query(prefix) }.join '&'
    end
  end
end

class Hash

  def to_query(namespace = nil)
    collect do |key, value|
      unless (value.is_a?(Hash) || value.is_a?(Array)) && value.empty?
        value.to_query(namespace ? "#{namespace}[#{key}]" : key)
      end
    end.compact.sort! * "&"
  end

  alias_method :to_param, :to_query
end



search_criteria = {from:"ST_L6NN3P6K",to:"ST_LX225YVP",date: "2017-03-08",adult:1,child:0}

def signature_of api_key, secret, params = {}
  time = Time.new.to_i
  hashdata = {api_key: api_key, t: time}.merge(params.reject {|k, v| v.is_a? Hash}.reject {|k, v| v.is_a?
Array}.reject {|k, v| v.nil?})
  sign = Digest::MD5.hexdigest(hashdata.sort.map{|k,v| "#{k}=#{v}"}.join + secret)
  result = {
    From: api_key,
    Date: Time.at(time).httpdate,
    Authorization: sign,
    "Api-Locale"=> 'zh-CN'
  }
  p result
  result
end

#Alpha
api_key = "1fdeae6e7fd44c9e991d21066a828f0c"
secret = "4dae4d6a-4874-4d60-8eac-67701520671d"
```

```
def send_http_get uri, api_key, secret, params
  Net::HTTP.start(uri.host, uri.port,
    :use_ssl => uri.scheme == 'https') { |http|
    request = Net::HTTP::Get.new uri
    signature = signature_of(api_key, secret, params)
    request["From"]=signature[:From]
    request["Date"]=signature[:Date]
    request["Authorization"]=signature[:Authorization]
    response = http.request request # Net::HTTPResponse object
    async_resp = JSON(response.body)
  }
end

begin
  uri = URI("https://alpha.api.detie.cn/api/v2/online_solutions?#{search_criteria.to_query}")
  async_resp = send_http_get uri, api_key, secret, search_criteria
  p async_resp
  sleep(3)

  get_result_uri = URI("https://alpha.api.detie.cn/api/v2/async_results/#{async_resp['async']}")
  50.times do
    sleep(3)
    solutions = send_http_get get_result_uri, api_key, secret, {async_key: async_resp['async']}
    p solutions
  end
rescue =>e
  p e
end
```

Book

> Make sure provide security params with each request

> Use async query to retrieve the response.

Book Request

```
POST /api/v2/online_orders
```

This is an async call, an async-key will be returned. Try to use

```
GET /api/v2/async_results/{async_key}
```

to retrieve response data.

The following is book request json for a ticket for train FR 9626 from Milano Centrale to Roma Termini, Executive service, base offer

```
  {
    "passengers": [
    {
      "last_name": "zhang",
      "first_name": "san",
      "birthdate": "1986-09-01",
      "passport": "A123456",
      "email": "x@a.cn",
      "phone": "+8615000367081",
      "gender": "male"
    }
  ],
  "sections": [
    "bc_05"
  ],
  "seat_reserved": true,
  "memo": "partner_order_id"
}
```

Book Parameters

In order to book ticket, we need two kinds of information including travelers information and order information such as section data such as book code. Also, only DB (2nd class, domestic trains) needs to specify whether to reserve a seat and the servation fee is 4.5 euro per seat.

Important notice: please make sure the email and cell phone number of the first passenger because the first passenger will be the contact person of the booking. The pdf ticket will be sent to his email address and a mobile ticket will be sent to his cell number.

Parameter	Description	Type
passengers	list of passengers, detail in **Passenger** table	array
sections	List of sections, detail in **Section** table	array
seat_reserved	reserve seat or not, only used for 2nd class of DB domestic. 4.5 Euro is required for reservation fee	boolean
memo	return as Order.memo or tp_order_id if offline booking	string

Passenger

Parameter	Description	Type
last_name	last name as on passport	string
first_name	first name as on passport	string
birthdate	birthday, format: yyyy-MM-dd	string
passport	passport number	string
email	email	string
phone	telephone, please add country code such as +86 for China, +852 for Hong Kong, or +886 for Taiwan	string
gender	male or female	enum

Sections

Parameter	Description	Type
book_code	From search response	array

Book Response

The following is response json for the book request above.

```
{
      "id": "OD_02NY86GJP",
      "railway": {
        "code": "TI"
      },
      "from": {
        "code": "ST_L6NN3P6K",
        "name": "Milano Centrale",
        "local_name": "Milano Central Station",
        "help_url": "http://help.g2rail.com/cn/station/milano-centrale"
      },
      "to": {
        "code": "ST_LX225YVP",
        "name": "Roma Termini",
        "local_name":  "Roma Termini",
        "help_url": "http://help.g2rail.com/cn/station/roma-termini"
      },
      "departure": "2017-03-08T11:00:00+01:00",
      "arrival": "2017-03-08T14:00:02+01:00",
      "payment_price": { "currency": "EUR", "cents": 0 },
      "charging_price": { "currency": "EUR", "cents": 8000 },
      "rebate_amount": { "currency": "EUR", "cents": 117 },
      "passengers": [
        {
          "id": "PN_69NKJLY13",
          "first_name": "san",
          "last_name": "zhang",
          "birthdate": "1986-09-01",
          "email": "x@a.cn",
          "phone": "+8615000367081",
          "gender": "male"
        }
      ],
      "tickets": [
        {
          "id": "TK_2E6GY7MYZ",
          "from": {
            "code": "ST_L6NN3P6K",
            "name": "Milano Centrale",
            "local_name": "Milano Central Station",
            "help_url": "http://help.g2rail.com/cn/station/milano-centrale"
          },
          "to": {
            "code": "ST_LX225YVP",
            "name": "Roma Termini",
            "local_name":  "Roma Termini",
            "help_url": "http://help.g2rail.com/cn/station/roma-termini"
          }
        }
```

```
    ]
  }
```

Parameters

Parameter	Description	Type
id	Order ID	string
railway	railway code, detail in **Railway** table	railway
from	departure station, detail in **Station** table	station
to	destination, detail in **Station** table	station
departure	departure time in UTC format such as “2017-03-08T13:30:00+01:00”	string
arrival	arrival time in UTC format such as “2017-03-08T13:30:00+01:00”	string
payment_price	total amount to be paid including booking fee. If pay with deposit account, this price is 0. Detail in **Price** table	price
charging_price	The amount to be charged from deposit account, detail in **Price** table	price
rebate_amount	commission fee or ticket refund amount, detail in **Price**	price
passengers	passengers information, detail in **Passenger**列表	array
tickets	list of tickets, detail in **Ticket** table	array

Ticket

Parameter	Description	Type
id	ticket ID	string
from	departure station, detail in **Station** table	station
to	destination station, detail in **Station** table	station

Category

Category	Description
custom	ID
ticket	ticket price
seat_reservation	seat reservation fee
fee	booking fee
commission	commission fee
refunded_ticket	amount refunded
refunded_seat_reservation	refunded seat reservation fee
return_commission	commission fee to be return when the ticket was refunded
coupon	coupon

Common Errors include:

System Error

Parameter	Description
internal_error	System busy
async_key_not_found	Your searching or reservation has timed out, please search again
not_enough_balance	The city you specified cannot be connected to the station. Please specify the specific station (such as Berlin Hbf) and then search again

Business Errors

Error Code	Error Description
11	The number of passengers is in error, please check
12	The departure date is in error, please check
14	The cabin is in error, please check
15	The itinerary is in error, please check
16	Your travel plan is in error, please check
21	The pre-sale period is in error, please check
22	Your travel plan is in error, please check
23	Part of your travel plan is in error, please check
33	Part of your travel plan is in error, please check
35	Enter invalid information, please check
47	Unsuccessful to purchase tickets from this train, please try another trains
49	Train has departed. Or the system has been out of time
55	Undefined information, please check your input
56	Undefined information, please check your input
57	Undefined information, please check your input
58	Undefined information, please check your input
67	This ticket cannot be purchased on the departure day. Please change to another date, class or train.
68	The number of passengers does not match, please check
69	Unable to book this ticket
70	This ticket cannot be purchased on the departure day
72	This ticket cannot be used by this type of account
75	The Date Of Birth format is wrong, please check
76	Unable to purchase the ticket, please choose another type of ticket, class or train
77	This class cannot be issued
80	Departure date must be entered
81	Departure date format is incorrect
83	You cannot reserve a seat in this class
88	Unable to book in advance
90	This discount is not applicable to the passengers in this country
91	The number of passengers in this class is illegal
95	Unable to book this ticket, please try another type of ticket
190	Not enough tickets to be issued, please choose another class or train
465	The selected train has been stopped or not yet on sale, please book another shift
551	The train is about to depart, unable to be reserved
1004	Failed to allocate seats, please choose another class or train
1007	UnableToRetrievePNR
1013	FinalizeBookingFailure
1014	Repeated booking
1015	UnableToRetrieveRFITrainNumber
1018	Booking failed, please choose another class or train
1019	Repeated trip, please select again
13006	The entered phone number is error
13007	Invalid email entered

Error Code	Error Description
13011	Invalid or incomplete email address
14001	The characters are invalid, please enter English character
20041	The number of children is more than one in an unaccompanied circumstances
20042	Can only reserve paper tickets for unaccompanied children
33003	This type of ticket cannot be issued
33004	This type of ticket does not have E-ticket
33021	Reservation is failed, please contact customer service
33022	Reservation is failed, please contact customer service
33023	Reservation is failed, please contact customer service
34050	Unable to issue E-ticket

The following is ruby sample code for above book request.

```
Ruby
```

```
#!/usr/bin/env ruby

require "digest/md5"
require 'time'
require 'net/http'
require "cgi"

require 'active_support/time'
require 'active_support/json'

book_information = {
      passengers: [
      {
        last_name: "zhang",
        first_name: "san",
        birthdate: "1986-09-01",
        passport: "A123456",
        email: "x@a.cn",
        phone: "+8615000367081",
        gender: "male"
      }
      ],
      sections: [
        "bc_05"
      ],
      seat_reserved: true
    }

def signature_of api_key, secret, params = {}
  time = Time.new.to_i
  hashdata = {api_key: api_key, t: time}.merge(params.reject {|k, v| v.is_a? Hash}.reject {|k, v| v.is_a?
Array}.reject {|k, v| v.nil?})
  sign = Digest::MD5.hexdigest(hashdata.sort.map{|k,v| "#{k}=#{v}"}.join + secret)
  result = {
    From: api_key,
    Date: Time.at(time).httpdate,
    Authorization: sign,
    "Api-Locale"=> 'zh-CN'
  }
end

#alpha
api_key = "1fdeae6e7fd44c9e991d21066a828f0c"
secret = "4dae4d6a-4874-4d60-8eac-67701520671d"

def send_http_post uri, api_key, secret, params
  res = Net::HTTP.start(uri.host, uri.port,
    :use_ssl => uri.scheme == 'https') { |http|
    request = Net::HTTP::Post.new uri
    signature = signature_of(api_key, secret, params.slice(:seat_reserved)) # Only need the simple hash k
```

```
ey/value pair for calculating singature
    request["From"]=signature[:From]
    request["Date"]=signature[:Date]
    request["Authorization"]=signature[:Authorization]
    request["Content-Type"] = 'application/json'
    request.body = params.to_json
    response = http.request request # Net::HTTPResponse object
    async_resp = JSON(response.body)
  }

end

def send_http_get uri, api_key, secret, params
  Net::HTTP.start(uri.host, uri.port,
    :use_ssl => uri.scheme == 'https') { |http|
    request = Net::HTTP::Get.new uri
    signature = signature_of(api_key, secret, params)
    request["From"]=signature[:From]
    request["Date"]=signature[:Date]
    request["Authorization"]=signature[:Authorization]
    response = http.request request # Net::HTTPResponse object
    async_resp = JSON(response.body)
  }
end

begin
  uri = URI("https://alpha.api.detie.cn/api/v2/online_orders")
  async_resp = send_http_post uri, api_key, secret, book_information
  p async_resp
  sleep(3)

  get_result_uri = URI("https://alpha.api.detie.cn/api/v2/async_results/#{async_resp['async']}")
  50.times do
    sleep(3)
    book_result = send_http_get get_result_uri, api_key, secret, {async_key: async_resp['async']}
    p book_result
  end
rescue =>e
  p e
end
```

Confirm Booking

Please make sure confirm booking within **20 mins** after booking, then the ticket will be issued.

> Make sure provide security params with each request

> Use async query to retrieve the response.

Confirm Request

```
POST /api/v2/online_orders/{online_order_id}/online_confirmations
```

This is an async call, an async-key will be returned. Try to use

```
GET /api/v2/async_results/{async_key}
```

to retrieve response data.

The following is Request json for the booking above

```
{
}
```

The most important parameter is online_order_id.

Parameters

Parameter	Description	Type
online_order_id	id in Book Response, make it in query string	path

Confirm Response


```
{
      "id": "OC_60OYGMYKX",
      "order": {
        "id": "OD_02NY86GJP",
        "PNR": "A4DL5N",
        "railway": {
          "code": "TI"
        },
        "from": {
          "code": "ST_L6NN3P6K",
          "name": "Milano Centrale",
          "local_name": "Milano Central Station",
          "help_url": "http://help.g2rail.com/cn/station/milano-centrale"
        },
        "to": {
          "code": "ST_LX225YVP",
          "name": "Roma Termini",
          "local_name":  "Roma Termini",
          "help_url": "http://help.g2rail.com/cn/station/roma-termini"
        },
        "departure": "2017-03-08T11:00:00+01:00",
        "passengers": [
          {
            "id": "PN_69NKJLY13",
            "first_name": "san",
            "last_name": "zhang",
            "birthdate": "1986-09-01",
            "email": "x@a.cn",
            "phone": "+8615000367081",
            "gender": "male"
          }
        ],
        "tickets": [
          {
            "id": "TK_2E6GY7MYZ",
            "from": {
              "code": "ST_L6NN3P6K",
              "name": "Milano Centrale",
              "local_name": "Milano Central Station",
              "help_url": "http://help.g2rail.com/cn/station/milano-centrale"
            },
            "to": {
              "code": "ST_LX225YVP",
              "name": "Roma Termini",
              "local_name":  "Roma Termini",
              "help_url": "http://help.g2rail.com/cn/station/roma-termini"
            }
```

```
          }
        ],
        "reservations": [
          {
            "train_name": "ICE 001",
            "car": "1",
            "seat": "1"
          }
        ]
      },
      "payment_price": { "currency": "EUR", "cents": 0 },
      "charging_price": { "currency": "EUR", "cents": 800 },
      "rebate_amount": { "currency": "EUR", "cents": 117 },
      "confirm_again": "no_need",
      "ticket_check_ins": [{
        "check_in_url": "https://staging.eurostar.com/customer-dashboard/en/booking?pnr=TVYG7F&surname=Ko
men",
        "earliest_check_in_timestamp": "2024-05-03 00:00",
        "latest_check_in_timestamp": "2024-06-01 14:52"
      }]
    }
```


Parameters

Chargine_price is the actual amount to be deducted from the deposit account.

Parameter	Description	Type
id	Confirm Order ID	string
order	order information, detail in **Order** table	order
ticket_price	total amount of ticket price, detail in **Price**	price
payment_price	total amount paid. This does include the amount directly charged from deposit account, detail in **Price** table	price
rtp_price	total amount directly charged from credit card, only used for DB ticket booking, detail in **Price** tabel	price
charging_price	total amount charged from deposit account, detail in **Price** table	price
rebate_amount	commission fee or refunded amound, detail in **Price** table	price
passengers	passenger information, detail in **Passenger** table	array
tickets	ticket information, detail in **Ticket** table	array
records	financial records, detail in **Record** table	array
confirm_again	Whether the tickets need to be issue right away, possible values includes: **need** and **no_need**	string
ticket_check_ins	check in url of Trenitalia regional train&EuroStar, traveler need to do check before boarding	string

Order

Parameter	Description	Type
id	Order ID	string
PNR	ticket PNR	string
railway	railway code, detail in **Railway** table	railway

Parameter	Description	Type
from	departure station, detail in **Station** table	station
to	destination station, detail in **Station** table	station
departure	departure time in UTC format, like “2017-03-08T13:30:00+01:00”	string
passengers	passenger information, detail in **Passenger** table	array
tickets	ticket information, detail in **Ticket** table	array
reservations	seat reservation information, detail in **Reservation** table	array
memo	order number of partner system	string

Reservation

Parameter	Description	Type
train_name	train name	string
car	carriage number	string
seat	seat number	string

TicketCheckIn

Parameter	Description	Type
check_in_url	url for online check in	string
earliest_check_in_timestamp	earliest_check_in_timestamp	string
latest_check_in_timestamp	latest_check_in_timestamp	string

Confirm_again

Based on types of offer, railway opertors, some tickets need to be confirmed again such as group ticket, pass. In addition, when there is error while booking, the confirm_again flag of book response or confirm response will be set to need. Partners need to provide a call back function when ticket ready.

Confirm Again Process

If **confirm_again** of confirm response is “need”, G2rail will call back client API to inform the readiness of ticket. Then partners can call download ticket API do retrieve ticket.

```
{
  "method": "issue_ticket",
  "order_id": "OD_0Y5WZLOG4",
  "client_order_id": "client_order_id",
  "ticket_no": "A8MFAN"
}
```

Call back parameters

Parameter	Description	Type
method	issue_ticket	string
order_id	G2Rail Order Id	string
client_order_id	Passed as memo when performing boooking, usually clients set it to order id of client system, optional	string
ticket_no	Ticket PNR	string

Error Code

Common Errors include:

System Error

Parameter	Description
internal_error	System busy
async_key_not_found	Your searching or reservation has timed out, please search again
solution_expired	searching has timed out, please search again
not_enough_balance	The city you specified cannot be connected to the station, please search again after specifying the specific station (such as Berlin Hbf)

Business Errors

Error Code	Error Description
47	Unsuccessful to purchase tickets from this train, please try another trains
49	Train has departed. Or the system has been out of time
69	Unable to book this ticket
77	This class cannot be issued
83	You cannot reserve a seat in this class
190	Not enough tickets to be issued, please choose another class or train
465	The selected train has been stopped or not yet on sale, please book another shift
551	The train is about to depart, unable to be reserved
1004	Failed to allocate seats, please choose another class or train
1007	UnableToRetrievePNR
1013	FinalizeBookingFailure
1014	Repeated booking
1015	UnableToRetrieveRFITrainNumber
1018	Booking failed, please choose another class or train
33003	This type of ticket cannot be issued
33004	This type of ticket does not have E-ticket
33021	Reservation is failed, please contact customer service
33022	Reservation is failed, please contact customer service
33023	Reservation is failed, please contact customer service
34050	Unable to issue E-ticket

Ruby

```ruby
#!/usr/bin/env ruby

require "digest/md5"
require 'time'
require 'net/http'
require "cgi"

require 'active_support/time'
require 'active_support/json'

confirm_information = {
  }

def signature_of api_key, secret, params = {}
  time = Time.new.to_i
  hashdata = {api_key: api_key, t: time}.merge(params.reject {|k, v| v.is_a? Hash}.reject {|k, v| v.is_a?
Array}.reject {|k, v| v.nil?})
  sign = Digest::MD5.hexdigest(hashdata.sort.map{|k,v| "#{k}=#{v}"}.join + secret)
  result = {
    From: api_key,
    Date: Time.at(time).httpdate,
    Authorization: sign,
    "Api-Locale"=> 'zh-CN'
  }
end

#alpha
api_key = "1fdeae6e7fd44c9e991d21066a828f0c"
secret = "4dae4d6a-4874-4d60-8eac-67701520671d"

def send_http_post uri, api_key, secret, params
  res = Net::HTTP.start(uri.host, uri.port,
    :use_ssl => uri.scheme == 'https') { |http|
    request = Net::HTTP::Post.new uri
    signature = signature_of(api_key, secret, { online_order_id: 'OD_V3G44VG85' })
    request["From"]=signature[:From]
    request["Date"]=signature[:Date]
    request["Authorization"]=signature[:Authorization]
    request["content_type"] = 'application/json'
    request.body = params.to_json
    response = http.request request # Net::HTTPResponse object
    async_resp = JSON(response.body)
  }

end

def send_http_get uri, api_key, secret, params
  Net::HTTP.start(uri.host, uri.port,
    :use_ssl => uri.scheme == 'https') { |http|
    request = Net::HTTP::Get.new uri
```

```
    signature = signature_of(api_key, secret, params)
    request["From"]=signature[:From]
    request["Date"]=signature[:Date]
    request["Authorization"]=signature[:Authorization]
    response = http.request request # Net::HTTPResponse object
    async_resp = JSON(response.body)
  }
end

begin
  uri = URI("http://alpha.api.g2rail.com/api/v2/online_orders/OD_V3G44VG85/online_confirmations")
  async_resp = send_http_post uri, api_key, secret, confirm_information
  p async_resp

  get_result_uri = URI("https://alpha.api.detie.cn/api/v2/async_results/#{async_resp['async']}")
  50.times do
    sleep(3)
    book_result = send_http_get get_result_uri, api_key, secret, {async_key: async_resp['async']}
    p book_result
  end
rescue =>e
  p e
end
```

Download Ticket

> Make sure provide security params with each request

> Use async query to retrieve the response.

Download Request

After successful confirmation, you may download electronic tickets. The ticket issuing processes of different carriers are different, some issue sync, some insync, you may need to wait a while to be able to download ticket.

```
Get /api/v2/online_orders/{online_order_id}/online_tickets
```

This is a sync call, it will return urls of tickets.

The following example is a download request json

```
  {
  }
```

The only parameter is online_order_id.

Parameters

Parameter	Description	Type
online_order_id	Confirm Order ID	path

Download Ticket Response

```
[
 {
    file: "https://alpha.api.detie.cn/tickets/8a37b66816e543d433ffd76e1eb98f65de6eabd04ba2a829a123304443c
6572a",
    kind: "pdf_ticket"
  },
  {
    file: "https://alpha.api.detie.cn/tickets/a9529bf6c561f38bd1745458d78fa9f25f8902d00ce16f86345562058c9
674c5",
    kind: "mobile_ticket"
  }
]
```

Refund Application

Many tickets support refund. Depending on the railway operators, the ways of discount and the seats you chose, the specific refund rules will be different. Some tickets support free refunds, some require a certain fee, and some tickets cannot be refunded. After the refund, the fee will be transferred to the G2Rail account.

> Note: You can refer to offer.refund_type (Refundable) to determine whether the the ticket is refundable. You can refer to offer.detail to see refund fee.

> Note: Deutsche Bahn’s reservation fee of 4.5 Euro cannot be refunded.

> Make sure provide security params with each request

> G2Rail will perform Refund Application call back to notify the outcome of refund.

Refund Application Request

```
POST /api/v2/online_orders/{online_order_id}/online_refund_applications
```

This is a sync call

The following is Request json for the Refund”OD_02NY86GJP”

```
  {
  }
```

The most important parameter for refund is online_order_id.

Parameters

Parameter	Description	Type
online_order_id	id in Book Response, make it in query string	path

Refund Application Response

```
{
  "result": "ok"
}
```

Refund Application Call Back

After receiving application, G2Rail will apply for refund from train operator. Once received result of refund from train operator, G2Rail will do Refund Application call back API to notify the result. In case of successful refund, refund amount will be transfered to your account. The followings are json response of refund application call back:

Call Back - Successful

```
  {
    "method": "refund",
    "timestamp": "2022-12-21 14:28:45",
    "order_id": "OD_02NY86GJP",
    "client_order_id": "client_order_id",
    "charge_fee": { "currency": "EUR", "cents": 60000 },
    "refund_fee": { "currency": "EUR", "cents": 80000 },
    "status": true,
  }
```

Parameters

Parameter	Description	Type
method	refund	string
timestamp	date time	string
order_id	G2Rail order ID	string
client_order_id	Passed as memo when performing boooking, usually clients set it to order id of client system, optional	string
charge_fee	refund fee	int
refund_fee	total amount to be refunded to account	int
status	refund result(true)	bool

Call Back - Failure

```
{
  "method": "refund",
  "timestamp": "2022-12-21 14:28:45",
  "order_id": "OD_02NY86GJP",
  "client_order_id": "client_order_id",
  "reason": {
    "memo": "Train departed already"
  },
  "status": false
}
```


Parameters

Parameter	Description	Type
method	refund	string
timestamp	date time	date time
order_id	G2Rail order	string
client_order_id	Passed as memo when performing boooking, usually clients set it to order id of client system, optional	string
reason	reason of failure	string
status	refund result(false)	bool

Ruby

```ruby
#!/usr/bin/env ruby

require "digest/md5"
require 'time'
require 'net/http'
require "cgi"

require 'active_support/time'
require 'active_support/json'

refund_information = {
  "client_order_id": "client_order_id"
}

def signature_of api_key, secret, params = {}
  time = Time.new.to_i
  hashdata = {api_key: api_key, t: time}.merge(params.reject {|k, v| v.is_a? Hash}.reject {|k, v| v.is_a?
Array}.reject {|k, v| v.nil?})
  sign = Digest::MD5.hexdigest(hashdata.sort.map{|k,v| "#{k}=#{v}"}.join + secret)
  result = {
    From: api_key,
    Date: Time.at(time).httpdate,
    Authorization: sign,
    "Api-Locale"=> 'zh-CN'
  }
end

#alpha
api_key = "1fdeae6e7fd44c9e991d21066a828f0c"
secret = "4dae4d6a-4874-4d60-8eac-67701520671d"

def send_http_post uri, api_key, secret, params
  res = Net::HTTP.start(uri.host, uri.port,
    :use_ssl => uri.scheme == 'https') { |http|
    request = Net::HTTP::Post.new uri
    signature = signature_of(api_key, secret, { online_order_id: 'OD_V3G44VG85' })
    request["From"]=signature[:From]
    request["Date"]=signature[:Date]
    request["Authorization"]=signature[:Authorization]
    request["content_type"] = 'application/json'
    request.body = params.to_json
    response = http.request request # Net::HTTPResponse object
    async_resp = JSON(response.body)
  }

end

def send_http_get uri, api_key, secret, params
  Net::HTTP.start(uri.host, uri.port,
    :use_ssl => uri.scheme == 'https') { |http|
```

```
    request = Net::HTTP::Get.new uri
    signature = signature_of(api_key, secret, params)
    request["From"]=signature[:From]
    request["Date"]=signature[:Date]
    request["Authorization"]=signature[:Authorization]
    response = http.request request # Net::HTTPResponse object
    async_resp = JSON(response.body)
  }
end

begin
  uri = URI("http://alpha.api.g2rail.com/api/v2/online_orders/OD_V3G44VG85/online_refund_applications")
  response = send_http_post uri, api_key, secret, refund_information
  p JSON(response.body)
rescue =>e
  p e
end
```

Close Order Request

In case of ticket issuing failure, G2Rail will call close_order API call back to notify and client should notify customer of failed ticketing and do refund to customer.

Close order call back

The following is json of close order call back

```
{
  "method": "close_order",
  "timestamp": "2022-12-21 14:28:45",
  "order_id": "OD_02NY86GJP",
  "client_order_id": "client_order_id",
  "reason": "sold out"
}
```

Parameters

Parameter	Description	type
method	close_order	string
timestamp	date time	date time
order_id	G2Rail order ID	string
client_order_id	Passed as memo when performing boooking, usually clients set it to order id of client system, optional	string
reason	failure reason	string

Security Parameters

Security parameters are required for all requests in http header.

```
  {
    "From": "ad53f5806e634e698c0f0f04e628444d",
    "Date":"Mon, 13 Mar 2017 09:29:43 GMT",
    "Authorization": "aafb519dddcb782b9a0e727ffeacf6bc"
  }
```

```
  def signature_of api_key, secret, params = {}
  time = Time.new.to_i
  hashdata = {api_key: api_key, t: time}.merge(params)
  sign = Digest::MD5.hexdigest(hashdata.sort.map{|k,v| "#{k}=#{v}"}.join + secret)
  result = {
    "From": api_key,
    "Date": Time.at(time).httpdate,
    "Authorization": sign
  }
end
```

Three steps to generate t, api_key and p

Step 1

- t is Unix timestamp
- api_key is API Key we sent to you
- p is for other parameters.
- Data is not part of md5 encryption

Step 2

ordered key value + private key concatenation

Concatenate according to ‘key=value+private key’ into string:
“adult=1api_key=040a3c5a7fa94d82b319a41a2e0fa923child=0date=2019-09-20 09:00from=FRANKFURTt=1563983424time=05:00to=BERLINab44d9f0-e70e-45a5-9e3e-7265230eb431”

t format

t is epoch time of Unix, see here (https://www.epochconverter.com/)

Step 3

Encrypt with md5

P.S.To simply the request information, all security paramewters of request example are ignored.

Retrieve response data with Async call

All response data is retrieve asynch. There are two ways to retrieve:

1. Query with HTTP Get

`/api/v1/async_results/218c2825aaa29fdee42de4ca9dcdcde6`

218c2825aaa29fdee42de4ca9dcdcde6 is the async_key returned when posting Search/Book/Confirm

You will receive json format response data once we received response from railway companies.

2. Webhook

Contact admin or send email to oulu@ul-e.com to add your call back url. Please make sure this url accept cross-domain HTTP Post.

The format of request is similar. Once received request, the url will return 200 and system will not send any longer.

```
{
  "key": "a0ec87ee69b8baf72073a5354f48e7d4"
  "result" [
    {
      "rw": "DB",
      "dt": "2017-02-23",
      "dur": "01:28",
      "s": "ST_E020P6M4",
      "d": "ST_DQMOQ7GW",
      "sn": "Berlin Hbf (tief)",
      "dn": "Halle(Saale)Hbf",
      "ni": 0,
      "secs": [
        {
          "id": "SC_14B0J2P",
          "s": "ST_E020P6M4",
          "d": "ST_DQMOQ7GW",
          "sn": "Berlin Hbf (tief)",
          "dn": "Halle(Saale)Hbf",
          "offers": [

            {
              "o": "80003",
              "od": "Flexpreis",
              "svcs": [
                {
                  "sa": 999,
                  "p": 4800,
                  "sc": "",
                  "sd": "2nd Class(2)"
                },
                {
                  "sa": 999,
                  "p": 8000,
                  "sc": "8030001",
                  "sd": "1st Class"
                }
              ]
            }
          ],
          "trzs": [
            {
              "trz": "ICE 1730",
              "s": "ST_E020P6M4",
              "d": "ST_DQMOQ7GW",
              "sn": "Berlin Hbf (tief)",
              "dn": "Halle(Saale)Hbf",
              "dep": "2017-02-23 12:02",
              "arr": "2017-02-23 13:30"
            }
```

```
          ]
        }
      ]
    }
  ]
}
```

(http://www.g2rail.com)　Grail (http://www.g2rail.com)

地面交通 智能预定