

INDICATA ViVi WebService REST API v2

Date: 2024-09-26

Revision: #ebe43188

Table of contents

- [Introduction](#)
 - [Rest API, HATEOAS and URI Templates](#)
 - [Support](#)
 - [Single API](#)
 - [Bulk API](#)
 - [Authentication and Authorization](#)
 - [Authentication](#)
 - [Authorization](#)
 - [HTTP Redirects](#)
 - [Timestamps and Timezone](#)
 - [Static IP range and Accelerated access](#)
 - [Proof of Concept Examples](#)
 - [Minimum Requirements Checklist](#)
- [Single API](#)
 - [HTTP Headers](#)
 - [Base Resource Entry Points \(HATEOAS\)](#)
 - [Relation: Whoami](#)
 - [Request](#)
 - [Response](#)
 - [Relation: Sequence](#)
 - [Request](#)
 - [Parameters](#)
 - [Response](#)
 - [Special Cases for Registration Month and Registration Year steps \(UK\)](#)
 - [Examples](#)
 - [Sample Response: Step Facelift](#)
 - [Relation: VIN \(Vehicle Identification Number\) / Reg \(Registration Number\) / Exid \(External ID\)](#)
 - [Request](#)
 - [Parameters](#)
 - [Response](#)
 - [Relation: Text \(Label/Value\)](#)
 - [Request](#)
 - [HTTP Headers & Parameters](#)
 - [Body](#)
 - [Response](#)
 - [Relation: VTI](#)
 - [Request](#)
 - [Parameters](#)
 - [Response](#)
 - [Sequence Response](#)
 - [Valuation](#)
 - [European Target Price](#)
 - [Parameters](#)
 - [Relation: Valuation Entry](#)
 - [Request](#)
 - [Parameters](#)
 - [Response](#)
 - [No similar vehicles](#)
 - [Deprecated identifications](#)
 - [Competitive Set Profile](#)
 - [PDF Profile](#)
 - [List price Profile](#)
 - [Relation: European Target Price Entry](#)
 - [Request](#)
 - [HTTP Headers & Parameters](#)
 - [Body](#)
 - [Response](#)
 - [Relation: Forecast Entry](#)
 - [Request](#)
 - [Parameters](#)
 - [Body](#)

- [Response](#)
- [Request Limiting](#)
 - [Billable Requests](#)
 - [Exceeding the request limit](#)
 - [Request Limit information](#)
- [Bulk API](#)
 - [HTTP Headers](#)
 - [Base Resource](#)
 - [Relation: Identification](#)
 - [Request](#)
 - [HTTP Headers](#)
 - [Parameters](#)
 - [Body](#)
 - [Response](#)
 - [Relation: Vehicle-Register](#)
 - [Request](#)
 - [HTTP Headers](#)
 - [Parameters](#)
 - [Body](#)
 - [Response](#)
 - [Relation: Lookup](#)
 - [Request](#)
 - [HTTP Headers](#)
 - [Parameters](#)
 - [Body](#)
 - [Response](#)
 - [Bulk Identification/Lookup Status Response](#)
 - [Relation: Summary](#)
 - [Relation: Preview](#)
 - [Relation: Report](#)
 - [Hateoas NextStep links](#)
 - [Relation: Valuation-Delegate](#)
 - [Relation: Valuation \(Bulk Direct\)](#)
 - [Relation: Valuation \(Bulk\)](#)
 - [Request](#)
 - [HTTP Headers](#)
 - [Parameters](#)
 - [Body](#)
 - [Response](#)
 - [Relation: Report](#)
- [First Registration Date](#)
 - [Label/Value inputs](#)
- [Odometer Unit & GeoRadius Unit](#)
- [Valuation Profiles](#)
- [Label/Value & CSV Identification Headers In Depth](#)
 - [Identification Country and Language](#)
- [Vehicle Type \(AVT/PVT\)](#)
 - [Precedence rules](#)
- [Identification & Valuation Caveats](#)
 - [Deprecated Identification](#)
 - [Country Specific Components](#)
 - [Valuation without trim](#)
- [Assumptions](#)
 - [Assumptions In depth](#)
 - [Faulty assumptions](#)
 - [Conflicting data](#)
- [Guides](#)
 - [Language \(i18n\)](#)
 - [HATEOAS & Templated Links](#)
 - [Polling Async Status](#)
 - [Navigating Paged Resources](#)
 - [PRG pattern \(Post/Redirect/Get\)](#)
 - [Character Encoding](#)
 - [Response](#)
 - [application/xml](#)

- [application/json](#)
 - [Accept and Accept-Charset HTTP headers](#)
- [Request](#)
 - [Default request charset](#)
 - [Bad charset implications](#)
 - [More References](#)
- [Appendix](#)
 - [Types & Definitions](#)
 - [General](#)
 - [Type: Assumption](#)
 - [Type: Link](#)
 - [Type: BaseResource](#)
 - [Type: Whoami](#)
 - [Type: Description](#)
 - [Type: Image](#)
 - [Type: NextStep](#)
 - [Type: CategoryField](#)
 - [Type: RegDateField](#)
 - [Type: CatField](#)
 - [Type: VTl](#)
 - [Type: ValuationLink](#)
 - [Type: VehicleIdentification](#)
 - [Type: OdometerUnit](#)
 - [Type: Odometer](#)
 - [Type: OdometerType](#)
 - [Type: OdometerWithType](#)
 - [Type: Condition](#)
 - [Type: VehicleAmount](#)
 - [Type: CompactVehicleAmount](#)
 - [Type: Valuation](#)
 - [Type: Amount](#)
 - [Type: RegTaxType](#)
 - [Type: RegTax](#)
 - [Type: VATType](#)
 - [Type: VAT](#)
 - [Type: MDS](#)
 - [Type: PurchasePriceBreakdown](#)
 - [Type: Criteria](#)
 - [Type: VehicleValuation](#)
 - [Type: MaxExportPrice](#)
 - [Type: EuropeanTargetPrice](#)
 - [Type: ComponentWithDiff](#)
 - [Type: OdometerWithDiff](#)
 - [Type: VehicleAmountWithDiff](#)
 - [Type: CompetitiveVehicle](#)
 - [Type: ForecastInput](#)
 - [Type: ForecastResult](#)
 - [Type: ForecastOutcome](#)
 - [Type: ForecastSuccess](#)
 - [Type: ForecastValuation](#)
 - [Type: ForecastFailure](#)
 - [Sequence Specific](#)
 - [Definition: Sequence Identification Steps](#)
 - [Type: GroupType \(enum\)](#)
 - [Type: ProviderEntry](#)
 - [Type: ProviderGroup](#)
 - [Type: ProviderData](#)
 - [Type: VehicleRegisterIdentification](#)
 - [Bulk Specific](#)
 - [Definition: Bulk Identification Steps](#)
 - [Type: JobStatus \(enum\)](#)
 - [Type: AsyncStatus](#)
 - [Type: ComponentStatus \(enum\)](#)
 - [Type: BulkVehicleStatus \(enum\)](#)
 - [Type: BulkResult](#)

- [Type: BulkIdentificationSummary](#)
- [Type: BulkVehiclePreview](#)
- [Type: BulkVehicleIdentification](#)
- [Type: BulkVehicleValuation](#)
- [Type: Page](#)

Introduction

Rest API, HATEOAS and URI Templates

The interaction between client/webservice is rather complex and the workflows for identification and valuation requires a lot of information. To alleviate some of the complexities, ViVi has been built using Best Practices for REST APIs, such as Uri Templates ([RFC 6570](#)) and [HATEOAS](#) style navigation.

Generally speaking a first level knowledge of HATEOAS and URI Templates are necessary as they form the basis of all interaction. Failure to use HATEOAS and URI Templates equals a failed integration, which will eventually break (unfortunately we have seen this multiple times).

Info

Since the API is HATEOAS style, the link structure may change over time, thus you must always follow the HATEOAS principles for navigation.

Do not attempt to modify links provided by ViVi, unless they are templated. Doing so can cause undefined behavior and is not supported.

You should never hardcode any links other than the base resources.

The REST APIs have a Maturity Level 3 on Richardson's Maturity Model and developers inexperienced with RMM, HATEOAS and URI Templates will probably need supervision or time to study those subjects.

There are a lot of library implementations for URI Templates, for a wide variety of languages. You can find a list here (incomplete): <https://code.google.com/archive/p/uri-templates/wikis/Implementations.wiki>

Support

For general support, please contact your contact person or write to our support mail at ws@indicata.com.

Single API

All responses from the Single API contains an *X-TracerId* HTTP header

```
X-TracerId: HFDS543MFDS543FDL5
```

To be able to provide support for specific requests to the Single API, the TracerId must be included in the support request.

Note: the TracerId is coupled to a user account. Trying to replace the TracerId or attempting to access the Single API with an invalid TracerId / user account combination, will cause an error (403 Forbidden).

Bulk API

All responses from the Bulk API contains an *X-Bulk-Job-Id* HTTP header

```
X-Bulk-Job-Id: 491f2245-8591-4115-b307-02365a7a1124
```

To be able to provide support for specific requests to the Bulk API, the bulkJobId must be included in the support request.

Authentication and Authorization

The API uses [HTTP Basic Authentication](#) over HTTPS.

Authentication

A request which is not authenticated will be rejected with a *401 Full authentication* is required to access this resource response

```
HTTP/1.1 401 Full authentication is required to access this resource
WWW-Authenticate: Basic realm="Indicata"
```

Authorization

Some endpoints requires the client to be authorized to be able to access them.

In case the required authorization does not exist, the request will be rejected with a *403 Forbidden* response

```
HTTP/1.1 403 Forbidden
```

HTTP Redirects

The API uses HTTP redirection where appropriate. Clients should assume that any request may result in a redirection.

HTTP Redirection is not an error and clients should follow the redirect.

Example of a redirect

```
HTTP/1.1 303 See Other
Location: /vivi/v2/HFDS543MFD5543FDL5/DK?category=PASSENGER&make=180
```

Timestamps and Timezone

All timestamps are returned in [ISO 8601](#) format:

YYYY-MM-DDTHH:MM:SSZ

If Timezone (Z) is not specified, then [UTC](#) (Coordinated Universal Time) is implied.

Static IP range and Accelerated access

There are 2 *hosts* available for accessing ViVi, *ws.indicata.com* and *gaws.indicata.com*.

Host	Description	IP Range
ws.indicata.com	preferred	N/A (dynamic)
gaws.indicata.com	alternate	75.2.82.63, 99.83.206.218

The standard host *ws.indicata.com* uses a dynamic IP range and in case your organization requires IP whitelisting of outgoing https traffic, you should use the alternate host *gaws.indicata.com*.

The host *gaws.indicata.com* is an *accelerated* endpoint, which simply means that it will use the most direct route to ViVi.

Most connections originating from within Europe will have no problems connecting directly to the standard host, however we have experienced performance issues in the outskirts of Europe and outside of Europe, but also in areas with unstable connections and/or with limited bandwidth.

In such cases, using the host *gaws.indicata.com* could alleviate some of these issues.

Proof of Concept Examples

We have a suite of proof of concept examples, written in 4 different languages.

The proof of concept examples contains both Single API and Bulk API integrations.

- Java 8+
- Python 2 / Python 3
- PHP 7+
- NodeJS (JavaScript)

Please ask support for any of these and we will send you the source code.

Minimum Requirements Checklist

For a successful ViVi integration, these minimum requirements should be fulfilled.

- Always use the HATEOAS links ([Rest API, HATEOAS and URI Templates](#)), never attempt to create urls by hand, because the interaction is not as hardcoded as you may believe
- Always parse and expand the URI Templates from HATEOAS links, using a library ([Rest API, HATEOAS and URI Templates](#)), because we may add more optional parameters, which is not a breaking change
- Always parse the response entities (XML/JSON), don't use simple substring/search ([Character Encoding](#))
- Always parse response headers according to specification, if necessary, instead of string matching ([Character Encoding](#))
- Always know your charset and specify it when posting data ([Character Encoding](#))
- Never re-use Tracer Id (while it is possible, you ultimately pay the same amount, but we can't provide support if we can't distinguish requests from one another) - you are also in violation of HATEOAS (see 1st and 2nd item in this checklist)
- Use the Single API / Bulk API according to your needs - the Single API is typically for interactive identification of a single vehicle, while the Bulk API is for multiple vehicles.

The Proof of Concept examples all use the HATEOAS links for navigating. The only 2 endpoints that are hardcoded are the 2 base resources. The examples also enforce the usage of [Character Encoding](#) to specify the input encoding. Of course, the response entities are also parsed in accordance with the above. Please review the examples for your inspiration.

Single API

The Single API is a Stateless REST Webservice.

The core of the service is centered around the Sequence, which is an interactive series of steps guiding a client through a Vehicle Identification.

When a vehicle has been identified, it can be valued.

The Sequence can be bootstrapped using Label/Value Input or from external sources, such as Vehicle Register Providers, using an identifier.

Please make sure to read the [Introduction](#) before proceeding with the specific documentation presented below.

HTTP Headers

The Single API supports both *application/xml* and *application/json*, using *application/json* as the default.

```
// for JSON
Accept: application/json

// for XML
Accept: application/xml
```

Always specify the Accept HTTP header, to ensure maximum stability and to show intent. It also allows the Service to detect unsupported Media Types and reject requests with a *406 Not Acceptable response* (in accordance with [RFC 7231](#), section 6.5.6).

The Single API supports the following general HTTP headers (other HTTP headers are specified where relevant)

HTTP Header	Details
Accept-Language	This controls in what language the response is generated in. See Language (i18n) .
Accept	<p>Only <i>application/json</i> and <i>application/xml</i> is supported, so one of these has to be one of the options in the HTTP header.</p> <p>If <i>application/json</i> or <i>application/xml</i> is not included as an accepted representation, then the service responds with <i>406 Not Acceptable</i>.</p>
Accept-Encoding	<p>Most http libraries contains automatic decompression capabilities, thus enabling this option will save a lot of bandwidth and download times improves significantly.</p> <p>ViVi only supports the <i>gzip</i> option (and <i>identity</i>).</p> <p>Not all responses are compressed when using this HTTP header, as there is a minimum response size (<i>Content-Length</i>) limit (following current best practices regarding compression).</p>
Authorization	Your authorization token (HTTP Basic Authentication)

Example:

```
GET ...
Accept-Language: en-GB
Accept: application/json
Accept-Encoding: gzip
Authorization: Basic <TOKEN-HERE>
```

Base Resource Entry Points (HATEOAS)

The [BaseResource](#) is the HATEOAS starting point of the Single API and is located at <https://ws.indicata.com/vivi/v2> (see [Static IP range and Accelerated access](#)).

HTTP/1.1 200 OK

Content-Type: application/json; charset=UTF-8

```
{
  "links": [
    {
      "href": "https://ws.indicata.com/vivi/v2/whoami",
      "type": "application/json",
      "rel": "whoami",
      "templated": false
    },
    {
      "href": "https://ws.indicata.com/vivi/v2{/country}",
      "type": "application/json",
      "rel": "sequence",
      "templated": true
    },
    {
      "href": "https://ws.indicata.com/vivi/v2{/country}/vin{/vin}{?assumption,regdate}",
      "type": "application/json",
      "rel": "vin",
      "templated": true
    },
    {
      "href": "https://ws.indicata.com/vivi/v2{/country}/reg{/reg}{?assumption,regdate}",
      "type": "application/json",
      "rel": "reg",
      "templated": true
    },
    {
      "href": "https://ws.indicata.com/vivi/v2{/country}/exid{/exid}{?assumption,regdate}",
      "type": "application/json",
      "rel": "exid",
      "templated": true
    },
    {
      "href": "https://ws.indicata.com/vivi/v2{/country}{?assumption}",
      "type": "application/json",
      "rel": "text",
      "templated": true
    },
    {
      "href": "https://ws.indicata.com/vivi/v2{/country,identifications}/valuation{/profiles}{?regdate,odometer,condition,vatreclaimable,geopostalcode,georadius,vehicletype,equipment}",
      "type": "application/json",
      "rel": "valuation",
      "templated": true
    },
    {
      "href": "https://ws.indicata.com/vivi/v2{/country}/european-target-price{?assumption,condition}",
      "type": "application/json",
      "rel": "european-target-price",
      "templated": true
    },
    {
      "href": "https://ws.indicata.com/vivi/v2{/country}/vti{/vti}{?regdate,equipment,vehicletype}",
      "type": "application/json",
      "rel": "vti",
      "templated": true
    }
  ]
}
```

```
]
}
```

One should always use the base resource and filter by the Relation (rel attribute) to obtain a templated uri, which can then be expanded and used to interact with the Single API. This is at the heart of ALL HATEOAS style APIs and ensures stability, even if the uri's, templates or variables changes.

There are 7 distinct entry points, which are explained in greater detail in the next couple of sections.

Relation: Whoami

The whoami endpoint is a simple endpoint, which can be used to query user information, such as roles. This is typically only relevant for multi-tenant setups and/or highly generic code (such as the vivi proof of concepts).

Request

```
https://ws.indicata.com/vivi/v2/whoami
```

Example:

```
GET /vivi/v2/whoami
Accept-Language: en-GB
Accept: application/json
Accept-Encoding: gzip
Authorization: Basic <TOKEN-HERE>
```

Response

The response is simply a [Whoami](#)

Example:

```
{
  "username": "someuser+ws@example.com",
  "name": "Some User Ws Api",
  "roles": [
    "ROLE_ACCOUNT_112522",
    "ROLE_AVG_DAYS_TO_SELL",
    "ROLE_EQUIPMENT",
    "ROLE_VEHICLE_TYPE",
    "ROLE_VIVI_V2",
    "ROLE_VIVI_V2_AT",
    "ROLE_VIVI_V2_BE",
    "ROLE_VIVI_V2_DE",
    "ROLE_VIVI_V2_MAX_PURCHASE_PRICE_100",
    "ROLE_VIVI_V2_PDF",
    "ROLE_VIVI_V2_RETAIL_100",
    "ROLE_VIVI_V2_SEQ",
    "ROLE_VIVI_V2_SUPPLY_DEMAND",
    "ROLE_VIVI_V2_TEXT",
    "ROLE_VIVI_V2_VALUATION"
  ],
  "accessibleCountries": [
    "AT",
    "BE",
    "DE"
  ],
  "accessibleStrategicCountries": []
}
```

Relation: Sequence

The Sequence is an interactive HATEOAS style API, where the client is presented with options on how to continue to next step (or alternatively back step). The result is a fully identified vehicle, which can be valued.

Request

```
https://ws.indicata.com/vivi/v2{/country}
```

Example:

```
GET /vivi/v2/DK
Accept-Language: en-GB
Accept: application/json
Accept-Encoding: gzip
Authorization: Basic <TOKEN-HERE>
```

Parameters

Parameter	Type	Description
country	string	2 digit Country Code (ISO 3166-1 alpha-2 code)

Response

See [Sequence Response](#)

Special Cases for Registration Month and Registration Year steps (UK)

INDICATA respects the standard UK Registration Date format, where Age Identifier is used as opposed to Registration Month. (https://en.wikipedia.org/wiki/Vehicle_registration_plates_of_the_United_Kingdom).

That means that the Registration Month step is *not* accessible for UK. Instead, the Registration Year and Age Identifier is presented in the Registration Date step.

Note: only valid Age Identifiers are shown.

Examples

```
2015/15
2015/64
2015/65
...
2000/V
2000/W
2000/X
```

Sample Response: Step Facelift

This sample response shows that the Sequence has arrived at the Facelift Step.

Several components have already been identified (Category, Make, Model, et.c).

HTTP/1.1 200 OK

Content-Type: application/json; charset=UTF-8

X-TracerId: HFDS543MFDS543FDL5

X-RequestLimit: 1000

X-RequestLimit-Remaining: 835

X-RequestLimit-Reset: 1554076800000

```
{
  "nextStep": [
    {
      "summary": "2002 - 2005",
      "name": "2002",
      "description": [
        {
          "label": "from",
          "value": "2002"
        },
        {
          "label": "to",
          "value": "2005"
        }
      ],
      "images": [
        ...
      ],
      "href": "https://ws.indicata.com/vivi/v2/HFDS543MFDS543FDL5/DK?category=PASSENGER&make=180&model=140&regdate=2007-02&body=521&facelift=291",
      "type": "application/json",
      "rel": "facelift",
      "templated": false
    },
    {
      "summary": "2005 - 2008",
      "name": "2005",
      "description": [
        {
          "label": "from",
          "value": "2005"
        },
        {
          "label": "to",
          "value": "2008"
        }
      ],
      "images": [
        ...
      ],
      "href": "https://ws.indicata.com/vivi/v2/HFDS543MFDS543FDL5/DK?category=PASSENGER&make=180&model=140&regdate=2007-02&body=521&facelift=292",
      "type": "application/json",
      "rel": "facelift",
      "templated": false
    }
  ],
  "country": "DK",
  "category": {
    "key": "PASSENGER",
    "summary": "Passenger",
    "name": "Passenger",
    "assumed": false,
    "link": {
```


```
        "href": "https://ws.indicata.com/vivi/v2/HFDS543MFDS543FDL5/DK",
        "type": "application/json",
        "rel": "category",
        "templated": false
    }
},
"make": {
    "key": [
        180
    ],
    "summary": "Ford",
    "name": "Ford",
    "assumed": false,
    "description": [],
    "link": {
        "href": "https://ws.indicata.com/vivi/v2/HFDS543MFDS543FDL5/DK?category=PASSENGER",
        "type": "application/json",
        "rel": "make",
        "templated": false
    }
},
"model": {
    "key": [
        140
    ],
    "summary": "Fiesta",
    "name": "Fiesta",
    "assumed": false,
    "description": [],
    "link": {
        "href": "https://ws.indicata.com/vivi/v2/HFDS543MFDS543FDL5/DK?category=PASSENGER&make=180",
        "type": "application/json",
        "rel": "model",
        "templated": false
    }
},
"regDate": {
    "key": "2007-02",
    "summary": "2007-02",
    "name": "2007-02",
    "link": {
        "href": "https://ws.indicata.com/vivi/v2/HFDS543MFDS543FDL5/DK?category=PASSENGER&make=180&model=1
40",
        "type": "application/json",
        "rel": "regmonth",
        "templated": false
    }
},
"body": {
    "key": [
        521
    ],
    "summary": "Hatchback, 5 Doors, 5 Seats",
    "assumed": false,
    "description": [
        {
            "label": "type",
            "value": "Hatchback"
        },
        {
            "label": "doors",
            "value": "5"
        }
    ]
}
```



```
    },
    {
      "label": "seats",
      "value": "5"
    }
  ]
}
```

Relation: VIN (Vehicle Identification Number) / Reg (Registration Number) / Exid (External ID)

The VIN/Reg/Exid lookup can bootstrap the Sequence with data from a Vehicle Register, thus skipping many (if not all) of the sequence steps, often making it possible to jump directly to a valuation/ETP.

 **Info**

Supported lookup types are different on a country by country basis, please ask support which lookups are supported in the different countries.

Request

```
https://ws.indicata.com/vivi/v2{/country}/{lookup-type}/{identifier}{?assumption,regdate,vrs-raw-link}
```

VIN Example:

```
GET /vivi/v2/DE/vin/WF0HXXWPJH6M70359?assumption=FULL&vrs-raw-link=true
Accept-Language: en-GB
Accept: application/json
Accept-Encoding: gzip
Authorization: Basic <TOKEN-HERE>
```

Reg Example:

```
GET /vivi/v2/NL/reg/KP-429-T?assumption=FULL&vrs-raw-link=true
Accept-Language: en-GB
Accept: application/json
Accept-Encoding: gzip
Authorization: Basic <TOKEN-HERE>
```

Exid Example:

```
GET /vivi/v2/TR/exid/2117852023?assumption=FULL&vrs-raw-link=true
Accept-Language: en-GB
Accept: application/json
Accept-Encoding: gzip
Authorization: Basic <TOKEN-HERE>
```

Parameters

Parameter	Type	Description
country	string	2 digit Country Code (ISO 3166-1 alpha-2 code)

Parameter	Type	Description
lookup-type	string	<ul style="list-style-type: none"> - vin - reg - exid
identifier	string	<ul style="list-style-type: none"> - A 17 character VIN number - Registration Number - Exid
assumption (Optional)	Assumption	<p>The assumption level</p> <p>Will default to NONE</p>
regdate (Optional)	string	If included, it will be interpreted as the first registration date, overriding the one obtained from the vehicle register provider.
vrs-raw-link (Optional)	boolean	<p>Whether to output links to the raw data obtained from the Vehicle Register Service providers. Default is false</p> <p>Not all countries are supported and the format (xml, json, proprietary, etc.) will vary according to the provider being used.</p> <p>The generated links are temporary and you are expected to fetch the data within 24 hours</p> <p>INDICATA reserves the right to change providers without warning.</p>

Response

Since the lookup is bootstrapping the Sequence, the response *is* the Sequence.

The only difference is that the Response is of type [VehicleRegisterIdentification](#).

VIN:

HTTP/1.1 200 OK

Content-Type: application/json; charset=UTF-8

X-TracerId: HFDS543MFDS543FDL5

X-RequestLimit: 1000

X-RequestLimit-Remaining: 835

X-RequestLimit-Reset: 1554076800000

```
{
  "identifier": "WF0HXXWPJH6M70359",
  "providerData": {
    "providerName": "DAT",
    "providerGroups": [
      ...
    ]
  },
  "nextStep": ...,
  "valuation": ...,
  "country": "DE",
  "category": ...,
  "make": ...,
  "model": ...,
  ...,
  "links": [
    {
      "href": "/vivi/v2/HFDS543MFDS543FDL5/DE/raw/v84LssZ0Byd5UVFE2zpRdQDYkM1ps4",
      "type": "application/xml",
      "rel": "vrs-raw-data-GetVehicleDataResponse",
      "templated": false
    },
    {
      "href": "/vivi/v2/HFDS543MFDS543FDL5/DE/raw/FvQ0KDE3dWi2DmuCGRlTZ9YRtGnwKGucaA13K",
      "type": "application/xml",
      "rel": "vrs-raw-data-GetVehicleIdentificationByVinResponse",
      "templated": false
    }
  ]
}
```

Reg:

HTTP/1.1 200 OK

Content-Type: application/json; charset=UTF-8

X-TracerId: HFDS543MFDS543FDL5

X-RequestLimit: 1000

X-RequestLimit-Remaining: 835

X-RequestLimit-Reset: 1554076800000

```
{
  "identifier": "YJ36093",
  "providerData": {
    "providerName": "A2SP",
    "providerGroups": [
      ...
    ]
  },
  "nextStep": ...,
  "valuation": ...,
  "country": "NL",
  "category": ...,
  "make": ...,
  "model": ...,
  ...,
  "links": [
    {
      "href": "/vivi/v2/HFDS543MFDS543FDL5/NL/raw/A2kT6H4SCJK0ILVKrgk2fm01X12z6MhrCT27A8Mlpgo",
      "type": "application/xml",
      "rel": "vrs-raw-data",
      "templated": false
    }
  ]
}
```

Exid:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
X-TracerId: HFDS543MFDS543FDL5
X-RequestLimit: 1000
X-RequestLimit-Remaining: 835
X-RequestLimit-Reset: 1554076800000

{
  "identifier": "2117852023",
  "providerData": {
    "providerName": "Otomotiv",
    "providerGroups": [
      ...
    ]
  },
  "nextStep": ...,
  "valuation": ...,
  "country": "TR",
  "category": ...,
  "make": ...,
  "model": ...,
  ...,
  "links": [
    {
      "href": "/vivi/v2/HFDS543MFDS543FDL5/TR/raw/A2kT6H4SCJK0ILVKrgk2fm01X12z6MhrCT27A8Mlpg0",
      "type": "application/xml",
      "rel": "vrs-raw-data",
      "templated": false
    }
  ]
}
```

See [Sequence Response](#) for more information on how to proceed.

Relation: Text (Label/Value)

The Label/Value entry point can bootstrap the Sequence from direct user input, thus skipping many (if not all) of the sequence steps, often making it possible to jump directly to a valuation/ETP.

Request

```
https://ws.indicata.com/vivi/v2{/country}{?assumption}
```

Example:

```
POST /vivi/v2/DK?assumption=NONE
Content-Type: text/plain; charset=UTF-8
Accept-Language: en-GB
Accept: application/json
Accept-Encoding: gzip
Authorization: Basic <TOKEN-HERE>

CATEGORY=passenger
MAKE=ford
MODEL=fiesta
VARIANT=1.3, Hatchback, Ambiente
FIRSTREG=2007-04
BODY=Hatchback
WEIGHT=900kg
DOORS=5
ENGINE=1.3
ENGINELITERS=1299ccm
FUEL=Petrol
TRANSMISSION=manual
WHEELDRIVE=front (FWD)
SEATS=5
VEHICLETYPE=demo
EQUIPMENT=Cruise Control, Lane Assist, Parking Camera, Leather Seats, SatNav
```

Info

If a future registration date, i.e.: *FIRSTREG*, is wanted, the “Allow future model forecast” Web Service role is needed.

HTTP Headers & Parameters

The only accepted Content-Type is text/plain along with charset information.

```
Content-Type: text/plain; charset=UTF-8
```

Warning

Always specify the charset!

Please refer to [Character Encoding](#) for more information.

Parameter	Type	Description
country	string	2 digit Country Code (ISO 3166-1 alpha-2 code)
assumption (Optional)	Assumption	The assumption level Will default to NONE

Body

The body of the request consists of a structured input in the form of Label/value pairs.
For an in depth discussion, please refer to [Label/Value & CSV Identification Headers In Depth](#).

	Label	Description
Mandatory	MAKE, MODEL	These Labels are mandatory and must be present. Failure to provide these will result in a <i>400 Bad Request</i> error response.
Optional	*	Optional headers further specify the vehicle, refer to Label/Value & CSV Identification Headers In Depth

Response

The Label / Value input is bootstrapping the Sequence, using the [PRG pattern \(Post/Redirect/Get\)](#). The response is therefore a redirect into the Sequence.

```
HTTP/1.1 303 See Other
X-TracerId: HFDS543MFDS543FDL5
X-RequestLimit: 1000
X-RequestLimit-Remaining: 835
X-RequestLimit-Reset: 1554076800000
Location: https://ws.indicata.com/vivi/v2/HFDS543MFDS543FDL5/DK?c-body=521&c-facelift=291,292&regdate=2007-04&c-engine=1598,1599,1606,1607&model=140&c-transmission=464&category=PASSENGER&c-trim=1191,1192&c-wheeldrive=222&make=180&seats=5&regmonth=4
```

The output of the Redirected HTTP GET Request is simply a [VehicleIdentification](#).

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
X-TracerId: HFDS543MFDS543FDL5
X-RequestLimit: 1000
X-RequestLimit-Remaining: 835
X-RequestLimit-Reset: 1554076800000

{
  "nextStep": ...,
  "country": "DK",
  "category": ...,
  "make": ...,
  "model": ...,
  "regDate": ...,
  "body": ...,
  "seats": ...
}
```

See [Sequence Response](#) for more information on how to proceed.

Relation: VTI

The VTI lookup can bootstrap the Sequence with a previously identified vehicle in the form of variantId:trimId:seats, making it possible to go backwards in the sequence and alter the identification or jump directly to valuation/ETP.

Request

```
https://ws.indicata.com/vivi/v2{/country}/vti{/vti}{?regdate,equipment,vehicletype,disableValidation}
```

Example:

```
GET /vivi/v2/DK/vti/8847:1192:5?regdate=2007-04&vehicletype=1&equipment=34,41,43
Accept-Language: en-GB
Accept: application/json
Accept-Encoding: gzip****
Authorization: Basic <TOKEN-HERE>
```

Parameters

Parameter	Type	Description
country	string	The identification country
vti	string	<p>The VTI is a string combining multiple values, joined with a colon: “variant:trim:seats”.</p> <p>Example: variant: 8847, trim: 1192, seats: 5 8847:1192:5</p> <p>Do note, that the only mandatory value is variant, so it is perfectly legal to omit trim and/or seats, however the colons must remain, such that: 8847:: or 8847:1192: or 8847::5</p>
regdate	string	The first Registration Date of the vehicle. First Registration Date
vehicletype (Optional)	integer	<p>Vehicle type key.</p> <p>If you want to disable AVT (Automatic Vehicle Type) filtering and force the valuation to treat the current vehicle as a <i>Used</i> vehicle, then expand <i>vehicletype</i> with a -1. Refer to Vehicle Type (AVT/PVT) for more information</p>
equipment (Optional)	array(integer)	Comma separated array of Equipment Key(s).
disableValidation (Optional)	boolean	This disables semantic validation checks with regard to the validity of the individual vti components (VariantId, TrimId, Seats) vs the First Registration Date and Country. It is not advised to disable this.

Response

See [Sequence Response](#) for more information on how to proceed.

Sequence Response


```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
X-TracerId: HFDS543MFDS543FDL5
X-RequestLimit: 1000
X-RequestLimit-Remaining: 835
X-RequestLimit-Reset: 1554076800000

{
  "nextStep": [
    {
      "summary": "Commercial",
      "name": "Commercial",
      "description": [],
      "href": "https://ws.indicata.com/vivi/v2/HFDS543MFDS543FDL5/DK?category=COMMERCIAL",
      "type": "application/json",
      "rel": "category",
      "templated": false
    },
    {
      "summary": "Passenger",
      "name": "Passenger",
      "description": [],
      "href": "https://ws.indicata.com/vivi/v2/HFDS543MFDS543FDL5/DK?category=PASSENGER",
      "type": "application/json",
      "rel": "category",
      "templated": false
    }
  ],
  "country": "DK"
}
```

The Response Entity is a [VehicleIdentification](#). (Please refer to [Request Limiting](#) to read more about the X-RequestLimit-* HTTP headers).

The response contains HATEOAS links in the nextStep property and these can be used to progress in the Sequence. See [Definition: Sequence Identification Steps](#).

Warning

Do not attempt to build these URLs yourself, as ViVi uses multiple augmented parameters under certain circumstances (c-* parameters, a-* parameters et al.).

Attempting to build these URLs yourself will eventually lead to a breakdown of your code or at least a non-working integration.

Don't violate HATEOAS principles, use the links that is presented in the response - if you are in doubt on how to handle this, please refer to the proof of concept example code.

On each invocation (step selection), more identified components (properties) will appear in the [VehicleIdentification](#) (category make, model, etc) and new steps will be presented.

When there is only 1 option to choose from, the API will automatically skip that step and add the identified component to the description. This has the implication that the number of steps required to identify 2 different vehicles might not be the same, as some steps are skipped.

Info

Do note that ViVi might determine that a step has only 1 option, even though the Sequence has not yet arrived at that step, e.g. it is a future step. In such cases, the component identification may be presented ahead of time and when reaching that particular step, it is skipped.

When there are no more next steps, the vehicle is fully identified.

Because of the caveats described in [Identification & Valuation Caveats](#), there may be times when there is no next step and the vehicle identification cannot proceed. In such cases a next step to restart will be presented.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
X-TracerId: HFDS543MFDS543FDL5
X-RequestLimit: 1000
X-RequestLimit-Remaining: 835
X-RequestLimit-Reset: 1554076800000

{
  "nextStep": [
    {
      "summary": "Unable to proceed with current identification. Return to start.",
      "name": "Unable to proceed with current identification. Return to start.",
      "description": [],
      "href": "/vivi/v2/HFDS543MFDS543FDL5/DK",
      "type": "application/json",
      "rel": "category",
      "templated": false
    }
  ],
  ...
}
```

Such cases can be reported to/discussed with support.

Valuation

When the vehicle identification is near-complete, such that INDICATA is able to value the vehicle, a Valuation Link will appear in the response (typically when step Trim is presented).

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
X-TracerId: HFDS543MFDS543FDL5
X-RequestLimit: 1000
X-RequestLimit-Remaining: 835
X-RequestLimit-Reset: 1554076800000

{
  ...
  "valuation": [
    {
      "variants": [
        8847
      ],
      "trim": 1192,
      "seats": 5,
      "href": "/vivi/v2/HFDS543MFDS543FDL5/DK/8847:1192:5/valuation{/profiles}?regdate=2006-09{&odometer,condition,vatreclaimable,geopostalcode,georadius,vehicletype,equipment}",
      "type": "application/json",
      "rel": "valuation",
      "templated": true
    }
  ],
  ...
}
```

Only the `/profiles` path and `odometer` parameter needs to be expanded, as all other parameters are optional. Please refer to [Relation: Valuation Entry](#) for a discussion of the profiles and parameters.

The main differences to the Valuation Entry is that

- Variables `country`, `identifications`, and `regdate` have been expanded
- Possibly also `vehicletype` and `equipment` if these have been identified. Refer to [Vehicle Type \(AVT/PVT\)](#) for more information regarding Vehicle Type.
- `TracerId` has been retained (to allow for better support, as this can be traced back to the identification)

Warning

Though it is possible to get a valuation before the vehicle is fully identified (in some cases), it is recommended to identify the vehicle fully (with trim). A full identification provides INDICATA with more information to value the vehicle. See more in [Identification & Valuation Caveats](#).

European Target Price

When the vehicle identification is near-complete, such that INDICATA is able to value the vehicle, a *European Target Price* Link will appear in the response (typically when step Trim is presented).

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
X-TracerId: HFDS543MFDS543FDL5
X-RequestLimit: 1000
X-RequestLimit-Remaining: 835
X-RequestLimit-Reset: 1554076800000

{
  ...
  "valuation": [
    {
      "variants": [
        8847
      ],
      "trim": 1192,
      "seats": 5,
      "href": "/vivi/v2/HFDS543MFDS543FDL5/DK/8847:1192:5/european-target-price?regdate=2006-09{&odometer,condition,reference,ownerid,vehicletype,equipment}",
      "type": "application/json",
      "rel": "valuation",
      "templated": true
    }
  ],
  ...
}
```

Parameters

All parameters are optional.

Parameter	Type	Description
odometer (Optional)	string or integer	<i>Integer</i> The valuation will be based on the exact odometer given by the client, interpreted in the unit as specified in Odometer Unit & GeoRadius Unit . <i>String (Odometer Estimation)</i> The only available option (for valuation) is STANDARD. Please refer to Type: OdometerType for more information.

Parameter	Type	Description
		<i>Empty or not present</i> Leaving odometer empty (or completely removing the parameter) will cause ViVi to use STANDARD Odometer Estimation
condition (Optional)	integer	Reserved for future use If present, it must be in the range 0-5
reference (Optional)	string	A simple metadata parameter, which is simply relayed back in the response, for use as a unique reference.
ownerid (Optional)	string	A reserved metadata parameter, which is ignored, unless special configuration has been done. Very few client integrations will ever use this.
vehicletype (Optional)	integer	Vehicle type key. If you want to disable AVT (Automatic Vehicle Type) filtering and force the valuation to treat the current vehicle as a <i>Used</i> vehicle, then expand <i>vehicletype</i> with a -1. Refer to Vehicle Type (AVT/PVT) for more information
equipment (Optional)	array(integer)	Comma separated array of Equipment Key(s).

As a site note, if one *only* wants to perform Text (Label/Value) Identification, immediately followed by European Target Price, ViVi does provide a specialized endpoint for that use case.

Please refer to [Relation: European Target Price Entry](#).

The main difference to the European Target Price Entry is that *odometer*, *reference*, *ownerid*, *vehicletype* and *equipment* parameters are specified as query parameters instead of in the Text (Label/Value).

Warning

Though it is possible to get a European Target Price before the vehicle is fully identified (in some cases), it is recommended to identify the vehicle fully (with trim). A full identification provides INDICATA with more information to value the vehicle. See more in [Identification & Valuation Caveats](#).

Relation: Valuation Entry

Most clients will not use this endpoint directly.

The relation valuation can be used for direct valuations, without having to go through the identification steps yet again.

This does require having stored some information from a previous identification, because some internal INDICATA identification representations are necessary.

Request

```
https://ws.indicata.com/vivi/v2{/country,identifications}/valuation{/profiles}{?regdate,odometer,condition,vat
reclaimable,geopostalcode,georadius,vehicletype,equipment}
```

Example:

```
GET /vivi/v2/DK/8847:1192:5/valuation/RETAIL_100,SUPPLY_DEMAND,MAX_PURCHASE_PRICE_100?regdate=2007-04&odometer=145546&geopostalcode=5000&georadius=100&condition=3&vatreclaimable=NO&vehicletype=1&equipment=34,41,43
Accept-Language: en-GB
Accept: application/json
Accept-Encoding: gzip
Authorization: Basic <TOKEN-HERE>
```

Parameters

Parameter	Type	Description
country	string	The identification country
identifications	string	<p>The identifications is a string combining multiple values, joined with a colon: "variant:trim:seats".</p> <p>Example: variant: 8847, trim: 1192, seats: 5 8847:1192:5</p> <p>Do note, that the only mandatory value is variant, so it is perfectly legal to omit trim and/or seats, however the colons must remain, such that: 8847:: or 8847:1192: or 8847::5</p>
profiles	string	<p>Comma separated string of the valuation profiles.</p> <p>Read more in Valuation Profiles</p>
regdate	string	The first Registration Date of the vehicle. First Registration Date
odometer (Optional)	string or integer	<p><i>Integer</i> The valuation will be based on the exact odometer given by the client, interpreted in the unit as specified in Odometer Unit & GeoRadius Unit.</p> <p><i>String (Odometer Estimation)</i> The only available option (for valuation) is STANDARD. Please refer to Type: OdometerType for more information.</p> <p><i>Empty or not present</i> Leaving odometer empty (or completely removing the parameter) will cause ViVi to use STANDARD Odometer Estimation</p>
condition (Optional)	integer	<p>Reserved for future use</p> <p>If present, it must be in the range 0-5</p>
vatreclaimable (Optional)	boolean (YES/NO)	<p>Boolean When set, ViVi will make the valuation based on the reclaimability on the VAT. Not all countries support both VAT models, please ask support if you wish to know more about this subject.</p> <p><i>Empty or not present</i> ViVi will use the defaults as defined in INDICATA. In the valuation response, the reclaimability of VAT is present. Please contact support for more information</p>

Parameter	Type	Description
geopostalcodes (Optional)	string	A postal code, to use as geo center for competitive vehicle search. The Postal Code must be a valid Postal Code in the identification country
georadius (Optional)	integer	The radius from which to include competitive vehicles. The unit is interpreted as specified in Odometer Unit & GeoRadius Unit
vehicletype (Optional)	integer	Vehicle type key. If you want to disable AVT (Automatic Vehicle Type) filtering and force the valuation to treat the current vehicle as a <i>Used</i> vehicle, then expand <i>vehicletype</i> with a -1. Refer to Vehicle Type (AVT/PVT) for more information
equipment (Optional)	array(integer)	Comma separated array of Equipment Key(s).

Response

The valuation response is a [VehicleValuation](#).

HTTP/1.1 200 OK

Content-Type: application/json; charset=UTF-8

X-TracerId: HFDS543MFDS543FDL5

X-RequestLimit: 1000

X-RequestLimit-Remaining: 835

X-RequestLimit-Reset: 1554076800000

```
{
  "odometer": {
    "odometerType": "EXPLICIT",
    "distance": 145546,
    "unit": "KILOMETER"
  },
  "overallCondition": {
    "grade": 3,
    "scale": "NAAA"
  },
  "vatReclaimable": "NO",
  "priceDate": "2019-03-29",
  "valuations": [
    {
      "amount": {
        "value": 32934,
        "currency": "DKK"
      },
      "regTax": {
        "type": "INCL_REG_TAX",
        "country": "DK"
      },
      "vat": {
        "vatType": "INCL_VAT",
        "vatReclaimable": "NO",
        "vatCountryCode": "DK"
      },
      "type": "RETAIL_100"
    },
    {
      "amount": {
        "value": 18561,
        "currency": "DKK"
      },
      "regTax": {
        "type": "INCL_REG_TAX",
        "country": "DK"
      },
      "vat": {
        "vatType": "INCL_VAT",
        "vatReclaimable": "NO",
        "vatCountryCode": "DK"
      },
      "type": "MAX_PURCHASE_PRICE_100"
    }
  ],
  "mds": {
    "marketDaysSupplyOverall": 45,
    "marketDaysSupplySimilar": 79,
    "marketDaysSupplyMarket": 71,
    "marketDaysSupplyMake": 59,
    "competitiveVehiclesSold": 4
  },
  "maxPurchasePriceBreakdown": {
```

```
"daysToSell": 90,
"budgetedPriceReduction": 1977,
"seasonalDepreciation": 596,
"costReconSum": 1500,
"costWarranty": 2000,
"costMarketing": 900,
"costInterest": 383,
"costVAT": 1195,
"profit": 4658
},
"competitiveVehiclesForSale": 7,
"competitiveVehiclesAverageOdometer": {
  "distance": 173000,
  "unit": "KILOMETER"
},
"competitiveVehiclesCriteria": [
  {
    "name": "Body",
    "type": "MATCH_EXACT",
    "value": "5 Doors Hatchback"
  },
  {
    "name": "FirstRegYear",
    "type": "MATCH_ALL",
    "value": "Any Year"
  },
  {
    "name": "Facelift",
    "type": "MATCH_ALL",
    "value": "Any Facelift"
  },
  {
    "name": "WheelDrive",
    "type": "MATCH_ALL",
    "value": "Any Wheel Drive"
  },
  {
    "name": "Transmission",
    "type": "MATCH_ALL",
    "value": "Any Transmission"
  },
  {
    "name": "Engine",
    "type": "MATCH_FUEL",
    "value": "Any Petrol"
  },
  {
    "name": "Trim",
    "type": "MATCH_ALL",
    "value": "Any Trim"
  },
  {
    "name": "Seat",
    "type": "MATCH_NEAR",
    "value": "5"
  }
],
"country": "DK",
"category": ...,
"make": ...,
"model": ...,
"regDate": ...,
```



```
"body": ...,
"facelift": ...,
"seats": ...,
"engine": ...,
"transmission": ...,
"wheelDrive": ...,
"trim": ...
}
```

No similar vehicles

When there are no similar vehicles, then a response of 200 is returned and the valuations are empty. Also competitiveVehiclesForSale will equal 0.

This may happen if the market is small and/or the vehicle is rare - then it simply is not possible to find competing vehicles.

Deprecated identifications

It may happen that an identification becomes deprecated ([Deprecated Identification](#)). In such cases, a response of *410 Gone* is returned and one must re-identify the vehicle.

Competitive Set Profile

When using the COMPETITIVE_SET profile, all the competitive vehicles are embedded in the JSON output.

Example Request:

```
GET /vivi/v2/DK/301181:66003:5/valuation/COMPETITIVE_SET,RETAIL_100?regdate=2017-01&odometer=35758
Accept-Language: en-GB
Accept: application/json
Accept-Encoding: gzip
Authorization: Basic <TOKEN-HERE>
```

Example Response:

HTTP/1.1 200 OK

Content-Type: application/json; charset=UTF-8

X-TracerId: HFDS543MFDS543FDL5

X-RequestLimit: 1000

X-RequestLimit-Remaining: 835

X-RequestLimit-Reset: 1554076800000

```
{
  ...
  "competitiveSet": [
    {
      "country": "DK",
      "category": {
        "name": "Passenger"
      },
      "regDate": {
        "name": "2016-07",
        "difference": -8593.0
      },
      "regDateMonthUnknown": false,
      "body": {
        "name": "Hatchback, 5 Doors, 5 Seats"
      },
      "facelift": {
        "name": "2014 - 2018"
      },
      "engine": {
        "name": "1.0 EcoBoost 74kW 101PS 998ccm",
        "difference": -33523.0
      },
      "transmission": {
        "name": "Manual 5"
      },
      "wheelDrive": {
        "name": "FWD"
      },
      "trim": {
        "name": "Trend",
        "difference": -12890.0
      },
      "odometer": {
        "difference": -6389.0,
        "distance": 48000,
        "unit": "KILOMETER"
      },
      "priceToMarket": 95.72714180519624,
      "price": {
        "amount": {
          "value": 146900.0,
          "currency": "DKK"
        },
        "regTax": {
          "type": "INCL_REG_TAX",
          "country": "DK"
        },
        "vat": {
          "vatType": "INCL_VAT",
          "vatReclaimable": "NO",
          "vatCountryCode": "DK"
        }
      }
    },
  ],
}
```

```

        "correctedPrice": {
          "difference": -61395.0,
          "amount": {
            "value": 153457.0,
            "currency": "DKK"
          },
          "regTax": {
            "type": "INCL_REG_TAX",
            "country": "DK"
          },
          "vat": {
            "vatType": "INCL_VAT",
            "vatReclaimable": "NO",
            "vatCountryCode": "DK"
          }
        },
        "seller": "Example AutoService",
        "excluded": false
      },
      ...
    ]
    ...
  }

```

PDF Profile

Enabling the PDF profile will spawn an async job which will generate a PDF report of the JSON output.

Example Request:

```

GET /vivi/v2/DK/8847:1192:5/valuation/PDF,RETAIL_100,SUPPLY_DEMAND,MAX_PURCHASE_PRICE_100?regdate=2007-04&odometer=145546&geopostalcode=5000&georadius=100&condition=3&vatreclaimable=NO
Accept-Language: en-GB
Accept: application/json
Accept-Encoding: gzip
Authorization: Basic <TOKEN-HERE>

```

Example Response:

```

HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
X-TracerId: HFDS543MFDS543FDL5
X-RequestLimit: 1000
X-RequestLimit-Remaining: 835
X-RequestLimit-Reset: 1554076800000

{
  ...
  "pdf": {
    "href": "https://ws.indicata.com/vivi/v2/HFDS543MFDS543FDL5/DK/pdf/86fc685a-0fc5-47e9-833d-b3b82dbb239d/status",
    "type": "application/json",
    "rel": "status",
    "templated": false
  },
  ...
}

```

Since the PDF report generation is an asynchronous operation, the link provided is a *status* link (rel="status") where the status of the report can be queried. While the link relation is *poll*, the report is still being generated, which means that the status must be *polled* until it changes.

The link is not templated and you do not need to expand a UriTemplate, you can use the link *href* property directly.

```
GET /vivi/v2/HFDS543MFDS543FDL5/DK/pdf/86fc685a-0fc5-47e9-833d-b3b82dbb239d/status
Accept-Language: en-GB
Accept: application/json
Accept-Encoding: gzip
Authorization: Basic <TOKEN-HERE>
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
X-TracerId: HFDS543MFDS543FDL5
X-RequestLimit: 1000
X-RequestLimit-Remaining: 835
X-RequestLimit-Reset: 1554076800000

{
  "links": [
    {
      "href": "https://ws.indicata.com/vivi/v2/HFDS543MFDS543FDL5/DK/pdf/86fc685a-0fc5-47e9-833d-b3b82dbb239d/status",
      "type": "application/json",
      "rel": "poll"
    }
  ]
}
```

Warning

Do not attempt to poll more often than every 5 seconds or you will receive a *429 Too Many Requests*

Under rare circumstances the relation *error* might be returned (if INDICATA is experiencing global problems). Please contact support if this is a recurring issue.

When the report has finished, the relation will be *result* and the type will change to the appropriate *application/pdf*.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
X-TracerId: HFDS543MFDS543FDL5
X-RequestLimit: 1000
X-RequestLimit-Remaining: 835
X-RequestLimit-Reset: 1554076800000

{
  "links": [
    {
      "href": "https://ws.indicata.com/vivi/v2/HFDS543MFDS543FDL5/DK/pdf/86fc685a-0fc5-47e9-833d-b3b82dbb239d",
      "type": "application/pdf",
      "rel": "result"
    }
  ]
}
```

The report can then be fetched and stored within your own storage system.

Warning

The report will be available for retrieval for **24 hours only**, so you must store the report in your own storage system. It may be kept longer, but that is not guaranteed.

Attempting to fetch "old" reports will result in a *410 Gone* response.

List price Profile

When using the LIST_PRICE profile, a list price is embedded in the JSON output if the vehicle has one.

Example Request:

```
GET /vivi/v2/DK/301181:66003:5/valuation/LIST_PRICE?regdate=2017-01
Accept-Language: en-GB
Accept: application/json
Accept-Encoding: gzip
Authorization: Basic <TOKEN-HERE>
```

Example Response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
X-TracerId: HFDSS543MFDS543FDL5
X-RequestLimit: 1000
X-RequestLimit-Remaining: 835
X-RequestLimit-Reset: 1554076800000

{
  ...
  "listPrice": {
    "amount": {
      "value": 5984,
      "currency": "DKK"
    },
    "regTaxType": "INCL_REG_TAX",
    "vatType": "INCL_VAT"
  },
  ...
}
```

Relation: European Target Price Entry

Most clients will not use this endpoint directly, as it is more natural to use the *sequence* to obtain an identification and then proceed from there. Using the sequence also allows identification from multiple sources (VIN/Registration Number/Exid - Sequence Selection) whereas *this endpoint only supports Label/Value input*.

Request

```
https://ws.indicata.com/vivi/v2{/country}/european-target-price{?assumption,condition}
```

Example:

```
POST /vivi/v2/DK/european-target-price?assumption=FULL&condition=3
Content-Type: text/plain; charset=UTF-8
Accept-Language: en-GB
Accept: application/json
Accept-Encoding: gzip
Authorization: Basic <TOKEN-HERE>

CATEGORY=passenger
MAKE=ford
MODEL=fiesta
VARIANT=1.3, Hatchback, Ambiente
FIRSTREG=2007-04
BODY=Hatchback
WEIGHT=900kg
DOORS=5
ENGINE=1.3
ENGINELITERS=1299ccm
FUEL=Petrol
TRANSMISSION=manual
WHEELDRIVE=front (FWD)
SEATS=5
VEHICLETYPE=demo
EQUIPMENT=Cruise Control, Lane Assist, Parking Camera, Leather Seats, SatNav
REFERENCE=0033cd85-0465-41fd-8821-ef02ee5f4b91
OWNERID=5JHskF2ZFdi5sk_iCkepK
ODOMETER=27890
```

HTTP Headers & Parameters

The only accepted Content-Type is text/plain along with charset information.

```
Content-Type: text/plain; charset=UTF-8
```

Warning

Always specify the charset!

Please refer to [Character Encoding](#) for more information.

Parameter	Type	Description
country	string	The <i>origin</i> country of the vehicle specified in a 2 digit Country Code (ISO 3166-1 alpha-2 code)
assumption (Optional)	Assumption	The assumption level Will default to NONE
condition (Optional)	integer	Reserved for future use If present, it must be in the range 0-5

Body

The body of the request consists of a structured input in the form of Label/value pairs.
For an in depth discussion, please refer to [Label/Value & CSV Identification Headers In Depth](#).

	Label	Description
Mandatory	MAKE, MODEL	These Labels are mandatory and must be present. Failure to provide these will result in a <i>400 Bad Request</i> error response.
Optional	*	Optional headers further specify the vehicle, refer to Label/Value & CSV Identification Headers In Depth
Additional	ODOMETER	If the odometer is known, then provide it here. The odometer unit will be interpreted according to the origin country, see more in Odometer Unit & GeoRadius Unit . If omitted then INDICATA will estimate the odometer, using INDICATA STANDARD Odometer Estimation. For more information, please contact support.
Additional	REFERENCE	A simple metadata parameter, which is simply relayed back in the response, for use as a unique reference.
Additional	OWNERID	A reserved metadata parameter, which is ignored, unless special configuration has been done. Very few client integrations will ever use this.

Response

When the Label/Value input does not lead to an identification, a *303 See Other* response is returned, which is a *bootstrapping* of the sequence ([PRG pattern \(Post/Redirect/Get\)](#)). This can be used to fully identify the vehicle, which will subsequently provide a (HATEOAS) link to the 'standard' [European Target Price](#), please see [Sequence Response](#).

```
HTTP/1.1 303 See Other
X-TracerId: HFDS543MFDS543FDL5
X-RequestLimit: 1000
X-RequestLimit-Remaining: 835
X-RequestLimit-Reset: 1554076800000
Location: https://ws.indicata.com/vivi/v2/HFDS543MFDS543FDL5/DK?c-body=5213&c-facelift=1807&c-engine=9285&a-seats=true&model=18&c-transmission=2455&category=PASSENGER&c-wheeldrive=1398&make=39&seats=4
```

On the other hand, if the Text (Label/Value) leads to a successful *identification* of the vehicle, then the status code is *200 OK* and the response is a [EuropeanTargetPrice](#).

An identification is necessary for ViVi to be able to perform the European Target Price calculations.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
X-TracerId: HFDS543MFDS543FDL5
X-RequestLimit: 1000
X-RequestLimit-Remaining: 835
X-RequestLimit-Reset: 1554076800000

{
  "maxExportPrice": {
    "vehicleAmount": {
      "amount": {
        "value": 46356.0,
        "currency": "EUR"
      },
      "regTaxType": "EXCL_REG_TAX",
      "vatType": "EXCL_VAT"
    },
    "localVehicleAmount": {
      "amount": {
        "value": 344218.0,
        "currency": "DKK"
      },
      "regTaxType": "EXCL_REG_TAX",
      "vatType": "EXCL_VAT"
    }
  },
  "exportIndex": 122,
  "reference": "0033cd85-0465-41fd-8821-ef02ee5f4b91",
  "odometer": {
    "odometerType": "EXPLICIT",
    "distance": 27890,
    "unit": "KILOMETER"
  },
  "priceDate": "2022-05-17",
  "country": "DK",

  "category": ...,
  "make": ...,
  "model": ...,
  "regDate": ...,
  "body": ...,
  "bodyLength": ...,
  "facelift": ...,
  "seats": ...,
  "engine": ...,
  "transmission": ...,
  "wheelDrive": ...,
  "equipment": ...,
  "links": []
}
```

Relation: Forecast Entry

Most clients will not use this endpoint directly, as it is more natural to use the sequence to obtain an identification and then proceed from there. Using the sequence also allows identification from multiple sources (VIN/Registration Number/Exid - Sequence Selection) whereas this endpoint only supports VTI inputs. See [Relation VTI](#Relation VTI).



Warning

Not all countries are supported for forecast, please contact support for more information about your specific country.

Info

Forecast is only available for vehicles with a future registration date (*regdate*) max. 12 months into the future. Furthermore, the “Allow future model forecast” Web Service role must be enabled for the user as well.

Request

```
https://ws.indicata.com/vivi/v2{/country,identifications}/forecast{/profiles}{?regdate,odometer,condition,vatr  
eclaimable,geopostalcode,georadius,vehicletype,equipment}
```

To generate forecasts, one must provide pairs of Odometer and Date (see [ForecastInput](#)), which can be in either *application/json* or *text/csv*:

Example (*application/json*):

```
POST /vivi/v2/DK/504997:84906:5/forecast/RETAIL_100,MAX_PURCHASE_PRICE_100?regdate=2021-01&odometer=&condition  
=&equipment=19,21,25,37,40,45,54,59  
Accept-Language: en-GB  
Accept: application/json  
Accept-Encoding: gzip  
Authorization: Basic <TOKEN-HERE>  
Content-Type: application/json  
  
[  
  {  
    "date": "2024-08-02",  
    "odometer": 100000  
  },  
  {  
    "date": "2025-08-02",  
    "odometer": 125000  
  },  
  {  
    "date": "2026-08-02",  
    "odometer": 150000  
  }  
]
```

Example (*text/csv*):

```
POST /vivi/v2/DK/504997:84906:5/forecast/RETAIL_100,MAX_PURCHASE_PRICE_100?regdate=2021-01&odometer=&condition  
=&equipment=19,21,25,37,40,45,54,59  
Accept-Language: en-GB  
Accept: application/json  
Accept-Encoding: gzip  
Authorization: Basic <TOKEN-HERE>  
Content-Type: text/csv; charset=UTF-8  
  
date,odometer  
2024-08-02,100000  
2025-08-02,125000  
2026-08-02,150000
```


Parameters

Parameter	Type	Description
country	string	The identification country
identifications	string	<p>The identifications is a string combining multiple values, joined with a colon: "variant:trim:seats".</p> <p>Example: variant: 8847, trim: 1192, seats: 5 8847:1192:5</p> <p>Do note, that the only mandatory value is variant, so it is perfectly legal to omit trim and/or seats, however the colons must remain, such that: 8847:: or 8847:1192: or 8847::5</p>
profiles	string	<p>Comma separated string of the valuation profiles.</p> <p>Read more in Valuation Profiles</p>
regdate	string	The first Registration Date of the vehicle. First Registration Date
odometer (Optional)	string or integer	<p><i>Integer</i> The forecast (valuation) will be based on the exact odometer given by the client, interpreted in the unit as specified in Odometer Unit & GeoRadius Unit.</p> <p><i>String (Odometer Estimation)</i> There are 2 options available (for forecast valuations), STANDARD or FORECAST_RELATIVE. Please refer to Type: OdometerType for more information.</p> <p><i>Empty or not present</i> Leaving odometer empty (or completely removing the parameter) will cause ViVi to use FORECAST_RELATIVE Odometer Estimation</p>
condition (Optional)	integer	<p>Reserved for future use</p> <p>If present, it must be in the range 0-5</p>
vatreclaimable (Optional)	boolean (YES/NO)	<p>Boolean When set, ViVi will make the valuation based on the reclaimability on the VAT. Not all countries support both VAT models, please ask support if you wish to know more about this subject.</p> <p><i>Empty or not present</i> ViVi will use the defaults as defined in INDICATA. In the valuation response, the reclaimability of VAT is present. Please contact support for more information</p>
geopostcode (Optional)	string	A postal code, to use as geo center for competitive vehicle search. The Postal Code must be a valid Postal Code in the identification country
georadius (Optional)	integer	The radius from which to include competitive vehicles. The unit is interpreted as specified in Odometer Unit & GeoRadius Unit
vehicletype (Optional)	integer	<p>Vehicle type key.</p> <p>If you want to disable AVT (Automatic Vehicle Type) filtering and force the valuation to treat the current vehicle as a <i>Used</i> vehicle, then</p>

Parameter	Type	Description
		expand <i>vehicletype</i> with a -1. Refer to Vehicle Type (AVT/PVT) for more information
equipment (Optional)	array(integer)	Comma separated array of Equipment Key(s).

Body

The body consist of pairs of [Type: ForecastInput](#), in either *application/json* or *text/csv* format.

 **Note**

The id properties are optional and are NOT required to be unique, thus the id's can be used for identification-purposes or grouping purposes.

```
...
Content-Type: application/json


[
  {
    "date": "2024-08-02",
    "odometer": 100000,
    "id": "green"
  },
  {
    "date": "2025-08-02",
    "odometer": 125000,
    id: "blue"
  },
  {
    "date": "2026-08-02",
    "odometer": 150000,
    "id": "red"
  },
  ...
]
```

Example (text/csv):

```
...
Content-Type: text/csv; charset=UTF-8

id,date,odometer
green,2024-08-02,100000
blue,2025-08-02,125000
red,2026-08-02,150000
...
```

Note that the odometer unit is inferred from the context (in other words, according to the country), see more in [Odometer Unit & GeoRadius Unit](#).

 **Warning**

Always specify the [Character Encoding](#) when using *text/csv*, as ViVi cannot infer *your* specific encoding.

Response

The forecast endpoint is an asynchronous operation and the link provided in the response is a *status* link (rel="status") where the status of the forecast can be queried.

While the link relation is *poll*, the forecast is still being processed, which means that the status must be *polled* until it changes.

The link is not templated and you do not need to expand a UriTemplate, you can use the link *href* property directly.

```
HTTP/1.1 202 Accepted
Content-Type: application/json; charset=UTF-8
X-TracerId: HFDS543MFDS543FDL5
X-RequestLimit: 1000
X-RequestLimit-Remaining: 835
X-RequestLimit-Reset: 1554076800000

{
  "links": [
    {
      "href": "https://ws.indicata.com/vivi/v2/HFDS543MFDS543FDL5/forecast/NLLxxydcTG_n/status",
      "type": "application/json",
      "rel": "poll"
    }
  ]
}
```

Subsequent polling requests responds with *200 OK*.

Warning

Do not attempt to poll more often than every 5 seconds or you will receive a *429 Too Many Requests*

Under rare circumstances the relation *error* might be returned (if INDICATA is experiencing global problems). Please contact support if this is a recurring issue.

When the forecast has finished, the relation will be *result*.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
X-TracerId: HFDS543MFDS543FDL5
X-RequestLimit: 1000
X-RequestLimit-Remaining: 835
X-RequestLimit-Reset: 1554076800000

{
  "links": [
    {
      "href": "https://ws.indicata.com/vivi/v2/HFDS543MFDS543FDL5/forecast/NLLxxydcTG_n/result",
      "type": "application/json",
      "rel": "result"
    }
  ]
}
```

The response (of the result link) contains a [Type: ForecastValuation](#)

```
HTTP/1.1 200 OK
Content-Type: application/json
X-TracerId: HFDS543MFDS543FDL5
X-RequestLimit: 1000
X-RequestLimit-Remaining: 835
X-RequestLimit-Reset: 1554076800000

{
  "forecast": {
    "forecastDate": "2021-08-25",
    "country": "DK",
    "vti": "366806:68387:5",
    "regDate": "2018-05",
    "outcomes": [
      {
        "successful": true,
        "id": "green",
        "success": {
          "date": "2023-06-01",
          "age": 39,
          "odometer": 100040,
          "retailPrice": 362561,
          "maxPurchasePrice": 305732
        }
      },
      ...
    ]
  },
  "valuation": {
    ...
  },
  "links": [
    {
      "href": "https://ws.indicata.com/vivi/v2/HFDS543MFDS543FDL5/forecast/NLLxxydcTG_n",
      "type": "application/json",
      "rel": "forecast"
    },
    ...
  ]
}
```

The forecast should then be stored within your own storage system.

Warning

The forecast will be available for retrieval for at most **24 hours**, so you must store the report in your own storage system. It may be kept longer, but that is not guaranteed. Attempting to fetch “old” forecasts will result in a *410 Gone* response.

The links property contains a forecast link, which can be used to generate more forecasts for the same valuation, without doing the valuation over and over.

This is both cost-saving and time-saving.

Request Limiting

Each user has limits in the amount of (billable) requests. For more information about billable requests, the limits or how to adjust them, please contact support.

Billable Requests

Generally speaking billable requests are successful identification and valuation requests.

Identification requests sometimes requires multiple steps but will only count as 1 billable request.

Again, for more information, contact support.

Exceeding the request limit

If the request limit is exceeded, all requests will be rejected with *429 Too Many Requests* ([RFC 6585](#)) responses:

```
HTTP/1.1 429 Too Many Requests
```

Request Limit information

All responses contain information regarding the current limit in the X-RequestLimit HTTP headers

```
X-RequestLimit: 4000
X-RequestLimit-Remaining: 3268
X-RequestLimit-Reset: 2980
```

This enables clients to react to limits before they are reached.

A description of the X-RequestLimit HTTP headers

HTTP Header	Description
X-RequestLimit	The maximum amount of requests (billable) the client is permitted to make
X-RequestLimit-Remaining	The number of remaining requests in the current request limit window
X-RequestLimit-Reset	The time at which the current request limit window resets (UTC Epoch Seconds)

Note that there are multiple request limit strategies, please contact support for more information.

Bulk API

The Bulk API is a Stateless Asynchronous REST WebService. This means that once a Bulk Job is Accepted, a temporary endpoint is created, which can then be used to query the status of the Bulk Job. Once the Job completes, the output can be retrieved.

The temporary endpoints will be available at least 24 hours after the Bulk Job completes, after which the temporary endpoint is removed.

That means, that clients must retrieve the results and store them locally, once a Bulk Job has completed.

Please make sure to read the [Introduction](#) before proceeding with the specific documentation presented below.

HTTP Headers

The Bulk API only supports *application/json*.

Always specify Accept HTTP header, to ensure maximum stability and to show intent.

It also allows the Service to detect unsupported Media Types and reject requests with a *406 Not Acceptable response* (in accordance with [RFC 7231](#), section 6.5.6).

The Bulk API supports the following general HTTP headers (other HTTP headers are specified where relevant)

HTTP Header	Details
Accept-Language	This controls in what language the response is generated in. See Language (i18n) .
Accept	Only <i>application/json</i> is supported If Accept HTTP header is present and <i>application/json</i> is not included as an accepted representation, then the service responds with <i>406 Not Acceptable</i> .
Accept-Encoding	Most http libraries contains automatic decompression capabilities, thus enabling this option will save a lot of bandwidth and download times improves significantly. ViVi only supports the <i>gzip</i> option (and <i>identity</i>). Not all responses are compressed when using this HTTP header, as there is a minimum response size (<i>Content-Length</i>) limit (following current best practices regarding compression).
Authorization	Your authorization token (HTTP Basic Authentication)

Base Resource

The BaseResource is the HATEOAS starting point of the Bulk API and is located at <https://ws.indicata.com/vivi/v2/bulk> (see [Static IP range and Accelerated access](#)).

HTTP/1.1 200 OK

Content-Type: application/json; charset=UTF-8

```
{
  "links": [
    {
      "href": "https://ws.indicata.com/vivi/v2/bulk/identification{?assumption}",
      "type": "application/json",
      "rel": "identification",
      "templated": true
    },
    {
      "href": "https://ws.indicata.com/vivi/v2/bulk/vehicle-register",
      "type": "application/json",
      "rel": "vehicle-register",
      "templated": false
    },
    {
      "href": "https://ws.indicata.com/vivi/v2/bulk/lookup",
      "type": "application/json",
      "rel": "lookup",
      "templated": false
    },
    {
      "href": "https://ws.indicata.com/vivi/v2/bulk/valuation{/profiles}",
      "type": "application/json",
      "rel": "valuation",
      "templated": true
    }
  ]
}
```

One should always use the base resource and filter by the Relation (rel attribute) to obtain a templated link, which can then be expanded and used to interact with the Bulk API.

See [Base Resource Entry Points \(HATEOAS\)](#).

There are 3 distinct entry points, which are explained in greater detail in the next couple of sections.

Relation: Identification

This endpoint is used to identify vehicles from Label/Values.

Request

```
https://ws.indicata.com/vivi/v2/bulk/identification{?assumption}
```

Example:


```
POST /vivi/v2/bulk/identification?assumption=FULL
```

Content-Type: text/csv; charset=UTF-8

Accept-Language: en-GB

Accept: application/json

Accept-Encoding: gzip

Authorization: Basic <TOKEN-HERE>

```
country,category,make,model,variant,firstreg,body,weight,doors,engine,engine liters,fuel,transmission,wheel drive,seats,trim
DE,Personenkraftwagen,Audi,A4,,2008-03,Avant Kombi ,,5,2.0 TFSI 211PS,1984ccm,Benzin,S tronic 7,Quattro,5,
DE,Personenkraftwagen,BMW,5,,2011-08,Limousine,,4,525d 197PS,2993ccm,Diesel,Schaltgetriebe,xDrive,5,
...
```

HTTP Headers

HTTP Header	Details
Content-Type	The only accepted content-type is text/csv. You should also specify the Character Encoding , such that ViVi can interpret the bytes correctly

Parameters

Parameter	Type	Description
assumption	Assumption	This will default to FULL

Body

The only accepted content type is *text/csv*. Always specify the [Character Encoding](#), as ViVi cannot infer *your* specific encoding.

For an in depth discussion, please refer to [Label/Value & CSV Identification Headers In Depth](#).

	CSV Header	Type	Default value	Description
Mandatory	Country	string		The Identification Country of the Vehicle 2 digit Country Code (ISO 3166-1 alpha-2 code)
Mandatory	Make	string		
Mandatory	Model	string		
Mandatory	FirstReg	string		The First Registration Date of the vehicle. See First Registration Date
Optional	*			See more headers in Label/Value & CSV Identification Headers In Depth
Valuation Delegation	Odometer, ValuationCountry, GeoPostalCode, GeoRadius, VATReclaimable, PriceDate, Forecast, VehicleType, Equipment		See CSV Headers section in Relation: Valuation (Bulk)	

	CSV Header	Type	Default value	Description
Additional	*	string		Any additional (user-input) CSV headers, will be propagated back to the client. This could be a VehicleId, rownum, metadata, etc.

Response

```

HTTP/1.1 202 Accepted
Content-Type: application/json; charset=UTF-8
X-Bulk-Job-ID: 249aa5b5-77f5-44c9-bb72-e71fe23de655

{
  "status": "QUEUED",
  "links": [
    {
      "rel": "status",
      "href": "https://ws.indicata.com/vivi/v2/bulk/identification/249aa5b5-77f5-44c9-bb72-e71fe23de655/status",
      "type": "application/json"
    }
  ]
}
```

The response is accepted, pending processing. This is signified by the Http Status Code 202 (remember this is an asynchronous API).

The Response Entity is an [AsyncStatus](#). Please refer to [Polling Async Status](#) on how to progress with polling.

See [Bulk Identification/Lookup Status Response](#)

Relation: Vehicle-Register

This endpoint is used to identify vehicles from Registration Number, Vehicle Identification Number (VIN) or Exid.

Request

```
https://ws.indicata.com/vivi/v2/bulk/identification{?assumption,vrs-raw-link}
```

Example:

```

POST /vivi/v2/bulk/identification?assumption=FULL&vrs-raw-link=true
Content-Type: text/csv; charset=UTF-8
Accept-Language: en-GB
Accept: application/json
Accept-Encoding: gzip
Authorization: Basic <TOKEN-HERE>

#,country,identifier
1,DE,U5YHM81BAGL202170
2,GB,W0VBF6EH6KG310520
3,DK,YJ36093
...
```

HTTP Headers

HTTP Header	Details
Content-Type	The only accepted content-type is text/csv. You should also specify the Character Encoding , such that ViVi can interpret the bytes correctly

Parameters

Parameter	Type	Description
assumption	Assumption	This will default to FULL
vrs-raw-link (Optional)	boolean	<p>Whether to output links to the raw data obtained from the Vehicle Register Service providers. Default is false</p> <p>Not all countries are supported and the format (xml, json, proprietary, etc.) will vary according to the provider being used.</p> <p>The generated links are temporary and you are expected to fetch the data within 24 hours</p> <p>INDICATA reserves the right to change providers without warning.</p>
firstreg (Optional)	boolean	If included, it will be interpreted as the first registration date, overriding the one obtained from the vehicle register provider.

Body

The only accepted content type is *text/csv*. Always specify the [Character Encoding](#), as ViVi cannot infer *your* specific encoding.

For an in depth discussion, please refer to [Label/Value & CSV Identification Headers In Depth](#).

	CSV Header	Type	Default value	Description
Mandatory	Country	string		<p>The Identification Country of the Vehicle</p> <p>2 digit Country Code (ISO 3166-1 alpha-2 code)</p>
Mandatory	Identifier	string		The Registration Number, a 17 character Vehicle Identification Number (VIN) to identify or Exid
Valuation Delegation	Odometer, ValuationCountry, GeoPostalCode, GeoRadius, VATReclaimable, PriceDate, Forecast		See CSV Headers section in Relation: Valuation (Bulk)	

Info

Valuation delegation with equipment and/or vehicle type could be achieved by making a double identification:

1. Do a vehicle register identification
2. Extract equipment and/or vehicleType key(s) from step 1 to a CSV with corresponding headers in lowercase
3. Use CSV from step 2 and perform a text identification

Response

```
HTTP/1.1 202 Accepted
Content-Type: application/json; charset=UTF-8
X-Bulk-Job-ID: 249aa5b5-77f5-44c9-bb72-e71fe23de655

{
  "status": "QUEUED",
  "links": [
    {
      "rel": "status",
      "href": "https://ws.indicata.com/vivi/v2/bulk/identification/249aa5b5-77f5-44c9-bb72-e71fe23de655/status",
      "type": "application/json"
    }
  ]
}
```

The response is accepted, pending processing. This is signified by the Http Status Code 202 (remember this is an asynchronous API).

The Response Entity is an [AsyncStatus](#). Please refer to [Polling Async Status](#) on how to progress with polling.

See [Bulk Identification/Lookup Status Response](#)

Relation: Lookup

Most clients will probably not be using this endpoint.

If a client already possesses “identifications”, e.g. variant, trim & seats, then this endpoint can be used to get textual descriptions of the vehicles.

Request

```
https://ws.indicata.com/vivi/v2/bulk/lookup{?disableValidation}
```

Example:

```
POST /vivi/v2/bulk/lookup
Content-Type: text/csv; UTF-8
Accept-Language: en-GB
Accept: application/json
Accept-Encoding: gzip
Authorization: Basic <TOKEN-HERE>

country,variant,firstreg
DE,314182,2016
DE,274640,2013
```

HTTP Headers

HTTP Header	Description
Content-Type	The only accepted content-type is <i>text/csv</i> . You should also specify the Character Encoding , such that ViVi can interpret the bytes correctly

Parameters

Parameter	Type	Description
disableValidation (Optional)	boolean	This disables semantic validation checks with regard to the validity of the individual vti components (VariantId, TrimId, Seats) vs the First Registration Date and Country. It is not advised to disable this.

Body

The only accepted content type is *text/csv*. Always specify the [Character Encoding](#), as ViVi cannot infer *your* specific encoding.

Optional CSV headers are allowed to be present with empty values.

	CSV Header	Type	Default value	Description
Mandatory	Country	string		Identification Country of Vehicle 2 digit Country Code (ISO 3166-1 alpha-2 code)
Mandatory	Variant	integer (variant) / string (VTI)		The INDICATA variant (output from Single API or Bulk Identification API) or VTI (see VTI in variant CSV header)
Mandatory	FirstReg*	string		The First Registration Date of the vehicle. See First Registration Date
Optional	Trim	integer		The INDICATA trim level of the vehicle (output from Single API or Bulk Identification API)
Optional	Seats	integer		The number of seats in the vehicle (output from Single API or Bulk Identification API)
Valuation Delegation	Odometer, ValuationCountry, GeoPostalCode, GeoRadius, VATReclaimable, PriceDate, Forecast, VehicleType, Equipment			See CSV Headers section in Relation: Valuation (Bulk)
Additional	*	string		Any additional (user-input) CSV headers, will be propagated back to the client. This could be a VehicleId, rownum, metadata, etc.

VTI in variant CSV header

Variant CSV header can also be a VTI (variant:trim:seats), in which case it takes precedence over any values provided in the trim and seats columns.

Precedence rules:

Variant CSV header	Trim CSV header	Seats CSV header	Result
123::			123::

Variant CSV header	Trim CSV header	Seats CSV header	Result
123::	987	5	123::
123:456:2	987	5	123:456:2
123	987	5	123:987:5

In the examples below, the vehicle is identified by 366806:68387:5 whereby the trim column (589624) and the seats column (4) have been ignored.

Request example:

```

POST /vivi/v2/bulk/lookup
Content-Type: text/csv; UTF-8
Accept-Language: en-GB
Accept: application/json
Accept-Encoding: gzip
Authorization: Basic <TOKEN-HERE>

country,variant,firstreg,trim,seats
DE,366806:68387:5,2016,589624,4

```

Response example:

```

{
  ...
  "content": [
    {
      ....
      "identification": {
        "valuation": [
          {
            "variants": [366806],
            ...
          }
          ...
        ],
        "trim": {
          "key": [68387],
          ...
        },
        "seats": {
          "key": [5],
          ...
        },
        "vti": {
          "id": "366806:68387:5",
          "variantId": 366806,
          "trimId": 68387,
          "seats": 5
        },
        ...
      }
    }
  ]
}

```

Response

Please refer to [Bulk Identification/Lookup Status Response](#), as these are exactly the same.

Bulk Identification/Lookup Status Response

When the Bulk Identification/Lookup Job has completed, several new relation links becomes available.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
X-Bulk-Job-ID: 249aa5b5-77f5-44c9-bb72-e71fe23de655

{
  "status": "COMPLETE",
  "total": 25000,
  "remaining": 0,
  "links": [
    {
      "templated": false,
      "rel": "summary",
      "href": "https://ws.indicata.com/vivi/v2/bulk/identification/249aa5b5-77f5-44c9-bb72-e71fe23de655/summary",
      "type": "application/json"
    },
    {
      "templated": true,
      "rel": "preview",
      "href": "https://ws.indicata.com/vivi/v2/bulk/identification/249aa5b5-77f5-44c9-bb72-e71fe23de655/preview{?page,size}",
      "type": "application/json"
    },
    {
      "templated": true,
      "rel": "report",
      "href": "https://ws.indicata.com/vivi/v2/bulk/identification/249aa5b5-77f5-44c9-bb72-e71fe23de655/report{?page,size}",
      "type": "application/json"
    },
    {
      "templated": true,
      "rel": "valuation",
      "href": "https://ws.indicata.com/vivi/v2/bulk/valuation{/profiles}",
      "type": "application/json"
    },
    {
      "templated": true,
      "rel": "valuation-delegate",
      "href": "https://ws.indicata.com/vivi/v2/bulk/identification/249aa5b5-77f5-44c9-bb72-e71fe23de655/valuation{/profiles}",
      "type": "application/json"
    }
  ]
}
```

These are described in the following sections.

Relation: Summary

The [BulkIdentificationSummary](#) contains information about the component identification status of all vehicles.

This can be used to find common problems with the input, e.g. in the following a lot of trim identifications have failed. Perhaps some trims have been misspelled in the input.

HTTP/1.1 200 OK

Content-Type: application/json; charset=UTF-8

X-Bulk-Job-ID: 249aa5b5-77f5-44c9-bb72-e71fe23de655

```
{
  "taskUuid": "249aa5b5-77f5-44c9-bb72-e71fe23de655",
  "creator": "<username-goes-here>",
  "noEntries": 25000,
  "errors": 0,
  "start": "2019-03-26T07:49:02.497Z",
  "finish": "2019-03-26T07:49:02.578Z",
  "tally": {
    "category": {
      "IDENTIFIED": 25000,
      "ASSUMED": 0,
      "FAILED": 0
    },
    "make": {
      "IDENTIFIED": 25000,
      "ASSUMED": 0,
      "FAILED": 0
    },
    "model": {
      "IDENTIFIED": 25000,
      "ASSUMED": 0,
      "FAILED": 0
    },
    "regdate": {
      "IDENTIFIED": 25000,
      "ASSUMED": 0,
      "FAILED": 0
    },
    "body": {
      "IDENTIFIED": 25000,
      "ASSUMED": 0,
      "FAILED": 0
    },
    "facelift": {
      "IDENTIFIED": 25000,
      "ASSUMED": 0,
      "FAILED": 0
    },
    "seats": {
      "IDENTIFIED": 0,
      "ASSUMED": 0,
      "FAILED": 25000
    },
    "bodyHeight": {
      "IDENTIFIED": 25000,
      "ASSUMED": 0,
      "FAILED": 0
    },
    "bodyLength": {
      "IDENTIFIED": 25000,
      "ASSUMED": 0,
      "FAILED": 0
    },
    "weight": {
      "IDENTIFIED": 25000,
      "ASSUMED": 0,
      "FAILED": 0
    }
  }
}
```



```

    },
    "engine": {
      "IDENTIFIED": 25000,
      "ASSUMED": 0,
      "FAILED": 0
    },
    "wheeldrive": {
      "IDENTIFIED": 25000,
      "ASSUMED": 0,
      "FAILED": 0
    },
    "transmission": {
      "IDENTIFIED": 25000,
      "ASSUMED": 0,
      "FAILED": 0
    },
    "trim": {
      "IDENTIFIED": 12423,
      "ASSUMED": 0,
      "FAILED": 12577
    }
  }
}

```

Relation: Preview

This resource shows a preview of the input vehicles, with status information. This can be used similarly to the summary, to locate problematic inputs, however the preview shows status per vehicle.

The preview resource is a paged resource with content type [BulkVehiclePreview](#).

See [Navigating Paged Resources](#).

HTTP/1.1 200 OK

Content-Type: application/json; charset=UTF-8

X-Bulk-Job-ID: 249aa5b5-77f5-44c9-bb72-e71fe23de655

```
{
  "links": [
    {
      "rel": "next",
      "href": "https://ws.indicata.com/vivi/v2/bulk/identification/249aa5b5-77f5-44c9-bb72-e71fe23de655/preview?page=2&size=1000",
      "type": "application/json"
    },
    {
      "rel": "previous",
      "href": "https://ws.indicata.com/vivi/v2/bulk/identification/249aa5b5-77f5-44c9-bb72-e71fe23de655/preview?page=0&size=1000",
      "type": "application/json"
    }
  ],
  "last": false,
  "first": false,
  "totalElements": 25000,
  "totalPages": 25,
  "numberOfElements": 1000,
  "size": 1000,
  "number": 1,
  "content": [
    ...,
    {
      "input": {
        "indicata-bulk-record-id": "001002",
        "country": "DE",
        "category": "Personenkraftwagen",
        "make": "Audi",
        "model": "A4",
        "variant": "",
        "firstreg": "2008-03",
        "body": "Avant Kombi ",
        "doors": "5",
        "engine": "2.0 TFSI 211PS",
        "engineLiters": "1984ccm",
        "fuel": "Benzin",
        "transmission": "S tronic 7",
        "wheelDrive": "Quattro",
        "seats": "5",
        "trim": ""
      },
      "result": {
        "status": "SUCCESS"
      },
      "nextStep": "trim",
      "statusMap": {
        "category": "IDENTIFIED",
        "make": "IDENTIFIED",
        "model": "IDENTIFIED",
        "regdate": "IDENTIFIED",
        "body": "IDENTIFIED",
        "bodyHeight": "IDENTIFIED",
        "bodyLength": "IDENTIFIED",
        "weight": "IDENTIFIED",
        "facelift": "IDENTIFIED",
```

```
    "seats": "IDENTIFIED",
    "engine": "IDENTIFIED",
    "transmission": "IDENTIFIED",
    "wheeldrive": "IDENTIFIED",
    "trim": "FAILED",
    "odometer": "ASSUMED"
  }
}
...
]
```

Relation: Report

Info

Retrieving the report incurs cost

This resource is the full identification report, which contains all vehicles, regardless of whether they have been identified.

The report resource is a paged resource, with content type [BulkVehicleIdentification](#).

HTTP/1.1 200 OK

Content-Type: application/json; charset=UTF-8

X-Bulk-Job-ID: 249aa5b5-77f5-44c9-bb72-e71fe23de655

```
{
  "links": [
    {
      "rel": "next",
      "href": "https://ws.indicata.com/vivi/v2/bulk/identification/249aa5b5-77f5-44c9-bb72-e71fe23de655/preview?page=1&size=1000",
      "type": "application/json"
    }
  ],
  "last": false,
  "first": true,
  "totalElements": 25000,
  "totalPages": 250,
  "numberOfElements": 1000,
  "size": 1000,
  "number": 0,
  "content": [
    ...
    {
      "input": {
        "indicata-bulk-record-id": "000000",
        "country": "DE",
        "category": "Personenkraftwagen",
        "make": "Audi",
        "model": "A4",
        "variant": "",
        "firstreg": "2008-03",
        "body": "Avant Kombi ",
        "doors": "5",
        "engine": "2.0 TFSI 211PS",
        "engineLiters": "1984ccm",
        "fuel": "Benzin",
        "transmission": "S tronic 7",
        "wheelDrive": "Quattro",
        "seats": "5",
        "trim": ""
      },
      "result": {
        "status": "SUCCESS"
      },
      "identification": {
        "nextStep": [
          {
            "summary": "bootstrap-trim",
            "name": "bootstrap-trim",
            "description": [],
            "href": "https://ws.indicata.com/vivi/v2/DE/bootstrap?c-body=660&c-facelift=356&regdate=2008-03&c-engine=2116&model=169&c-transmission=573&category=PASSENGER&c-wheelDrive=272&make=30&seats=5&regmonth=3",
            "type": "application/json",
            "rel": "bootstrap-trim",
            "templated": false
          },
          ...
        ],
        "valuation": [
          {
```

```
        "variants": [
            12383
        ],
        "seats": 5,
        "href": "https://ws.indicata.com/vivi/v2/HFDS543MFDS543FDL5/DE/12383::5/valuation{/pro
files}?regdate=2008-03{&odometer,condition,vatreclaimable,geopostalcode,georadius}",
        "type": "application/json",
        "rel": "valuation",
        "templated": true
    }
],
"country": "DE",
"category": {
    "key": "PASSENGER",
    "summary": "Passenger",
    "name": "Passenger",
    "assumed": false
},
"make": {
    "key": [
        30
    ],
    "summary": "Audi",
    "name": "Audi",
    "assumed": false,
    "description": []
},
"model": {
    "key": [
        169
    ],
    "summary": "A4",
    "name": "A4",
    "assumed": false,
    "description": []
},
"regDate": {
    "key": "2008-03",
    "summary": "2008-03",
    "name": "2008-03"
},
"body": {
    "key": [
        660
    ],
    "summary": "Avant, Estate, 5 Doors, 5 Seats",
    "name": "Avant",
    "assumed": false,
    "description": [
        {
            "label": "type",
            "value": "Estate"
        },
        {
            "label": "doors",
            "value": "5"
        },
        {
            "label": "seats",
            "value": "5"
        }
    ]
}
```

```
},
"facelift": {
  "key": [
    356
  ],
  "summary": "2007 - 2012",
  "name": "2007",
  "assumed": false,
  "description": [
    {
      "label": "from",
      "value": "2007"
    },
    {
      "label": "to",
      "value": "2012"
    }
  ]
},
"seats": {
  "key": [
    5
  ],
  "summary": "5 seats",
  "name": "5",
  "assumed": false,
  "description": []
},
"engine": {
  "key": [
    2116
  ],
  "summary": "2.0 TFSI 155kW 211PS 1984ccm",
  "name": "2.0 TFSI",
  "assumed": false,
  "description": [
    {
      "label": "fuel_type",
      "value": "Petrol"
    },
    {
      "label": "displacement[cc]",
      "value": "1984"
    },
    {
      "label": "power[hp]",
      "value": "208"
    },
    {
      "label": "power[PS]",
      "value": "211"
    },
    {
      "label": "power[kW]",
      "value": "155"
    }
  ]
},
"transmission": {
  "key": [
    573
  ],
```

```

        "summary": "S tronic 7",
        "name": "S tronic",
        "assumed": false,
        "description": [
            {
                "label": "type",
                "value": "Automatic"
            },
            {
                "label": "steps",
                "value": "7"
            }
        ]
    },
    "wheelDrive": {
        "key": [
            272
        ],
        "summary": "Quattro",
        "name": "Quattro",
        "assumed": false,
        "description": [
            {
                "label": "type",
                "value": "4WD"
            }
        ]
    },
    "links": [
        {
            "href": "/vivi/v2/HFDS543MFDS543FDL5/DE/raw/v84LssZ0Byd5UVFE2zpRdQDYkM1ps4",
            "type": "application/xml",
            "rel": "vrs-raw-data-GetVehicleDataResponse",
            "templated": false
        },
        {
            "href": "/vivi/v2/HFDS543MFDS543FDL5/DE/raw/FvQ0KDE3dWi2DmuCGRlTZ9YRtGnwKGucaA13K",
            "type": "application/xml",
            "rel": "vrs-raw-data-GetVehicleIdentificationByVinResponse",
            "templated": false
        }
    ]
}
    },
    ...
]
}

```

Hateoas NextStep links

Each [BulkVehicleIdentification](#) contains a [VehicleIdentification](#), which in turn contains the nextStep property which is an array of [NextStep](#), unless the vehicle has been fully identified (including trim).

The next step links are specialized sequence bootstrap links which can be used to jump directly into the Single API sequence identification.

The relation always takes the form *bootstrap-[sequence-step](#)*.

```

...
"nextStep": [
  {
    "summary": "bootstrap-body",
    "name": "bootstrap-body",
    "description": [],
    "href": "https://ws.indicata.com/vivi/v2/DE/bootstrap?regdate=2008-03&model=169&category=PASSENGER&make=30&seats=5&regmonth=3",
    "type": "application/json",
    "rel": "bootstrap-body",
    "templated": false
  }
],
...

```

This is a feature with which partially identified vehicles from Bulk can be fully identified manually, using the sequence.

Because a vehicle can be valuated without a trim, there is yet another feature especially for the trim step.

If *trim* is the only non-identified component, there is a bootstrap link to the sequence + all available trims. The available trim links do not have the bootstrap- prefix.

```

...
"nextStep": [
  {
    "summary": "bootstrap-trim",
    "name": "bootstrap-trim",
    "description": [],
    "href": "https://ws.indicata.com/vivi/v2/DE/bootstrap?c-body=660&c-facelift=356&regdate=2008-03&c-engine=2116&model=169&c-transmission=573&category=PASSENGER&c-wheeldrive=272&make=30&seats=5&regmonth=3",
    "type": "application/json",
    "rel": "bootstrap-trim",
    "templated": false
  },
  {
    "summary": "Active",
    "name": "Active",
    "description": [
      {
        "label": "key",
        "value": "80566"
      }
    ],
    "href": "https://ws.indicata.com/vivi/v2/DE/bootstrap?category=PASSENGER&make=368&model=150&regdate=2016&body=5235&facelift=1825&engine=9170&wheeldrive=1413&transmission=2479&trim=80566",
    "type": "application/json",
    "rel": "trim",
    "templated": false
  },
  ...
],
...

```

Relation: Valuation-Delegate

Since it is extremely common, to delegate identified vehicles directly to a valuation, ViVi has a specialized endpoint for doing so. This is the valuation-delegate.

Simply put, ViVi will

- Split the identified vehicles (any vehicle which can be valued, meaning with/without trim) from the *UNidentified* vehicles
- Value the identified vehicles
- Concatenate the valued vehicles with the *UNidentified* vehicles (in-order, according to the original input)
- Provide the combined report for download

Info

You should make sure to add any necessary CSV Valuation Specific Headers in the initial HTTP POST Request. See CSV Headers section in [Relation: Valuation \(Bulk\)](#).

The only necessary parameter, is the [Valuation Profiles](#).

```
POST https://ws.indicata.com/vivi/v2/bulk/identification/249aa5b5-77f5-44c9-bb72-e71fe23de655/valuation/RETAIL_100,SUPPLY_DEMAND,MAX_PURCHASE_PRICE_100
Accept-Language: en-GB
Accept: application/json
Accept-Encoding: gzip
Authorization: Basic <TOKEN-HERE>
```

Do note that the *X-Bulk-Job-Id* HTTP header now changes, as you will receive a composite id in the response HTTP header.

The composite id is the identification bulk job id and the new valuation bulk job id, combined with a slash. The status relation now only contains a link to the new valuation bulk job id.

```
HTTP/1.1 202 Accepted
Content-Type: application/json; charset=UTF-8
X-Bulk-Job-Id: 249aa5b5-77f5-44c9-bb72-e71fe23de655/a42a92a5-3980-4bb3-b4bd-8bea1b217def

{
  "status": "QUEUED",
  "links": [
    {
      "rel": "status",
      "href": "https://ws.indicata.com/vivi/v2/bulk/valuation/a42a92a5-3980-4bb3-b4bd-8bea1b217def/status",
      "type": "application/json"
    }
  ]
}
```

See [Relation: Valuation \(Bulk\)](#) for a discussion about the response, as the response is an [AsyncStatus](#).

Relation: Valuation (Bulk Direct)

This is used for direct inputs of already identified vehicles and is rarely used from this context, although it is quite possible.

See [Relation: Valuation \(Bulk\)](#).

Relation: Valuation (Bulk)

This is used for direct inputs of already identified vehicles.

The typical use cases includes (but are not limited to):

- Repeated Portfolio Valuations
- Geo Valuations

- International Valuations

Repeated Portfolio Valuations

Assuming one has a portfolio of 4000 vehicles, that one wants to value on a regular basis, say monthly, then the portfolio-identifications can be stored and reused.

Delta updates may occur and can be handled by simply:

- Removing (sold) vehicles from the stored portfolio-identifications
- Re-identifying new/modified vehicles and adding them to the stored portfolio-identifications

Each month, a new valuation is performed of the portfolio-identifications.

Instead of re-identifying all vehicles every month, the identification step can be skipped, as the identifications are stored and delta-updated.

Thus, one can save on the identification costs.

International Valuations

Assuming one has a portfolio of 100 vehicles, which one wishes to value in multiple countries.

Then one can identify the 100 vehicles and use the 100 identifications to create 400 valuations (where each identification is duplicated with 4 different valuation countries).

Thus, one can save on the identification costs.

Geo Valuation

Quite similar to International Valuations, but using different GeoPostalCode(s) and/or Radii, per vehicle, instead of varying the countries.

Request

```
https://ws.indicata.com/vivi/v2/bulk/valuation{/profiles}
```

Example:

```
POST /vivi/v2/bulk/valuation/RETAIL_100,SUPPLY_DEMAND,MAX_PURCHASE_PRICE_100
```

```
Content-Type: text/csv; UTF-8
```

```
Accept-Language: en-GB
```

```
Accept: application/json
```

```
Accept-Encoding: gzip
```

```
Authorization: Basic <TOKEN-HERE>
```

```
country,variant,firstreg
```

```
DE,314182,2016
```

```
DE,274640,2013
```

```
...
```

HTTP Headers

HTTP Header	Description
Content-Type	The only accepted content-type is <i>text/csv</i> . You should also specify the Character Encoding , such that ViVi can interpret the bytes correctly

Parameters

Parameter	Type	Description
profiles	string	The comma separated string of the valuation profiles. Read more in Valuation Profiles .

Body

The only accepted content type is *text/csv*. Always specify the [Character Encoding](#), as ViVi cannot infer *your* specific encoding.

	CSV Header	Type	Default value	Description
Mandatory	Country	string		The Identification Country of the Vehicle 2 digit Country Code (ISO 3166-1 alpha-2 code)
Mandatory	Variant	integer		The identified variant
Mandatory	FirstReg	string		The First Registration Date of the vehicle. See First Registration Date
Optional	Seats	integer		The number of seats
Optional	Trim	integer		The identified trim
Optional	Odometer	integer / string	STANDARD / FORECAST_RELATIVE	<p><i>Integer</i> The request will be based on the exact odometer given by the client, interpreted in the unit as specified in Odometer Unit & GeoRadius Unit.</p> <p><i>String (Odometer Estimation)</i> STANDARD is always available, while the additional FORECAST_RELATIVE is available for forecasts. Please refer to Type: OdometerType for more information.</p> <p>Empty or not present Leaving odometer empty (or completely removing the parameter) will cause ViVi to use the STANDARD Odometer Estimation for pure valuations and FORECAST_RELATIVE for forecasts (The request is considered a forecast if the profile FORECAST is used)</p>
Optional	ValuationCountry	string	Country CSV Header	The country in which to perform a valuation 2 digit Country Code (ISO 3166-1 alpha-2 code)
Optional	GeoPostalCode	string	ALL	A postal code, to use as geo center for competitive vehicle search. The Postal Code must be a valid Postal Code in the ValuationCountry CSV header (which defaults to <i>Country</i>)

	CSV Header	Type	Default value	Description
Optional	GeoRadius	integer	ALL	The radius from which to include competitive vehicles. The unit is interpreted as specified in Odometer Unit & GeoRadius Unit
Optional	VATReclaimable	boolean (YES/NO)	Depending on the vehicle category and the INDICATA country defaults	Ask support for more information regarding this
Optional	PriceDate	string	Current Date	An ISO 8601 Date (yyyy-MM-dd). Must not be in the future, but this can be used to create historical valuations (for sold vehicles, price-chart of a vehicle market value, etc.)
Optional	VehicleType	integer		Vehicle type key. If empty, AVT filtering will occur, provided that AVT filters have been enabled. If you want to disable AVT (Automatic Vehicle Type) filtering and force the valuation to treat the current vehicle as a <i>Used</i> vehicle, then set the <i>VehicleType</i> column to -1. Refer to Vehicle Type (AVT/PVT) for more information
Optional	Equipment	array(integer)		Comma separated list of the identified Equipment Key(s)
Optional	F-date-{n}, F-odom-{n}	string, integer		Forecast parameters are specified in pairs of <i>date</i> and <i>odometer</i> . The date is an ISO 8601 Date: yyyy-MM-dd, while the odometer is a positive integer, whose unit is inferred by the context, i.e. the country (Odometer Unit & GeoRadius Unit). The pairing is done using numerical suffixes, such that pair 1 is given by the two columns <i>F-date-1</i> and <i>F-odom-1</i> , pair 2 by <i>F-date-2</i> and <i>F-odom-2</i> and so forth. Not all countries are supported, please contact support for more information about your specific country.

	CSV Header	Type	Default value	Description
Additional	*	string		Any additional (user-input) CSV headers, will be propagated back to the client. This could be a VehicleId, rownum, metadata, etc.

Response

The response is accepted, pending processing. This is signified by the Http Status Code *202 Accepted*, as this is an asynchronous API.

The Response Entity is an [AsyncStatus](#). Refer to [Polling Async Status](#) on how to progress with polling.

Once completed, only 1 link is present, the report.

Relation: Report

This resource is the full valuation report, which contains all valuation results.

The report resource is a paged resource, with content type [BulkVehicleValuation](#). See [Navigating Paged Resources](#).

200 OK

Content-Type: application/json; charset=UTF-8

X-Bulk-Job-ID: 249aa5b5-77f5-44c9-bb72-e71fe23de655

```
{
  "links": [
    {
      "rel": "next",
      "href": "https://ws.indicata.com/vivi/v2/bulk/identification/249aa5b5-77f5-44c9-bb72-e71fe23de655/preview?page=1&size=1000",
      "type": "application/json"
    }
  ],
  "last": false,
  "first": true,
  "totalElements": 25000,
  "totalPages": 250,
  "numberOfElements": 1000,
  "size": 1000,
  "number": 0,
  "content": [
    ...
    {
      "input": {
        "indicata-bulk-record-id": "000000",
        "country": "DE",
        "variant": "314182",
        "firstreg": "2016",
      },
      "result": {
        "status": "SUCCESS"
      },
      "valuationStatus": {
        "SUPPLY_DEMAND": "SUCCESS",
        "RETAIL_100": "SUCCESS"
      },
      "valuation": {
        "odometer": {
          "odometerType": "STANDARD",
          "distance": 55191,
          "unit": "KILOMETER"
        },
        "overallCondition": {
          "grade": 3,
          "scale": "NAAA"
        },
        "vatReclaimable": "YES",
        "priceDate": "2019-04-01",
        "valuations": [
          {
            "amount": {
              "value": 12207,
              "currency": "EUR"
            },
            "regTax": {
              "type": "EXCL_REG_TAX",
              "country": "DE"
            },
            "vat": {
              "vatType": "INCL_VAT",
              "vatReclaimable": "YES",
            }
          }
        ]
      }
    }
  ]
}
```

```
        "vatCountryCode": "DE"
      },
      "type": "RETAIL_100"
    }
  ],
  "mds": {
    "marketDaysSupplyOverall": 82,
    "marketDaysSupplySimilar": 84,
    "marketDaysSupplyMarket": 59,
    "marketDaysSupplyMake": 59,
    "competitiveVehiclesSold": 7
  },
  "competitiveVehiclesForSale": 13,
  "competitiveVehiclesAverageOdometer": {
    "distance": 50647,
    "unit": "KILOMETER"
  },
  "competitiveVehiclesCriteria": [
    {
      "name": "Body",
      "type": "MATCH_EXACT",
      "value": "5 Doors Hatchback"
    },
    {
      "name": "FirstRegYear",
      "type": "MATCH_NEAR",
      "value": "2015 - 2017"
    },
    {
      "name": "Facelift",
      "type": "MATCH_EXACT",
      "value": "2015 -"
    },
    {
      "name": "WheelDrive",
      "type": "MATCH_EXACT",
      "value": "FWD"
    },
    {
      "name": "Transmission",
      "type": "MATCH_EXACT",
      "value": "Manual 6"
    },
    {
      "name": "Engine",
      "type": "MATCH_EXACT",
      "value": "1.6 CDTI 95PS 1598ccm"
    },
    {
      "name": "Trim",
      "type": "MATCH_ALL",
      "value": "Any Trim"
    },
    {
      "name": "Seat",
      "type": "MATCH_NEAR",
      "value": "5"
    },
    {
      "name": "Odometer",
      "type": "MATCH_NEAR",
      "value": "25.000 - 125.000Km"
    }
  ]
}
```

```

    }
  ],
  "country": "DE",
  "category": ...,
  "make": ...,
  "model": ...,
  "regDate": ...,
  "body": ...,
  "facelift": ...,
  "engine": ...,
  "transmission": ...,
  "wheelDrive": ...
},
"forecast": {
  "forecastDate": "2019-04-01",
  "country": "DK",
  "vti": "314182::",
  "regDate": "2016-05",
  "outcomes": [
    {
      "successful": true,
      "id": 1,
      "success": {
        "date": "2021-04-01",
        "age": 59,
        "odometer": 139400,
        "retailPrice": 36256,
        "maxPurchasePrice": 30573
      }
    },
    ...
    {
      "successful": false,
      "id": 13,
      "failure": {
        "errorCode": 44,
        "errorDescription": "Future date not in the future: 2013-01-01"
      }
    },
    ...
  ]
},
...
]
}

```

Note

The ids in the forecast output correspond to the number suffixes in the `_F-date-{n}`, `F-odom-{n}` pairs.

First Registration Date

INDICATA uses the First Registration Date to help identify the vehicle.
Generally speaking, the following formats are accepted (directly) in any country:

Pattern	Comment
yyyy	Only the year is present
yyyy-mm	Year and Month is present
yyyy-mm-dd	Year, Month and Day is present (ISO-8601)

Having the month present in the Registration Date allows for higher accuracy in the valuation.

There are special cases for UK, since UK uses *Age Identifier* instead of Registration Month (further reading at https://en.wikipedia.org/wiki/Vehicle_registration_plates_of_the_United_Kingdom).
Note: The UK patterns are additional to the common patterns above, which means vivi will also understand the above patterns in UK.

Pattern	Comment
yyyyGB-AI	This is a special case, which INDICATA can interpret as well
yyyy/AI	This is the INDICATA representation and how dates are presented in ViVi and in the INDICATA Web Application

where AI = Age Identifier
such as X, W, 11, 65, etc.

Label/Value inputs

For Label/Value inputs ViVi will attempt to identify “any” format given, not only the aforementioned patterns. This includes localized formats, such as month names and abbreviations of those + alternate formats.

Examples for Denmark (DK) with month name October (Oktober in Danish) and different representations:

- * Okt, 2012
- * 1. okt 2012
- * Oktober 2012

Examples for United Kingdom (UK) with alternate syntax

- * 2013 (62)
- * 2017 (64 reg)

More specifically, these alternate representations can be used in:

- Single API [Relation: Text \(Label/Value\)](#)
- Bulk API [Relation: Identification](#)

We encourage you to do trial-and-error if you have specialized patterns and/or ask support with regard to these.

Odometer Unit & GeoRadius Unit

Odometer is interpreted in the unit corresponding to the *Vehicle Identification* country, whereas GeoRadius is interpreted in the unit corresponding to the *Valuation* country, as summarized in the following table:

ISO Country	Unit
GB	English Mile
SE	Swedish Mile
all others	Kilometer

When no Valuation Country is provided, it is assumed that the valuation country is the same as the identification country.

Valuation Profiles

When performing a valuation, the client can choose from the available profiles. The profiles are enabled per client and not all clients can expect access to all profiles - that depends on how the user access is configured.

Please ask support for details with regard to what the different profiles mean and which profiles the client should be configured with.

It is possible to request multiple profiles simultaneously, by separating them with a comma.

Profile Name	Profile Value	Single API	Bulk API
Supply Demand	SUPPLY_DEMAND	✓	✓
Retail 100	RETAIL_100	✓	✓
Max Purchase Price 100	MAX_PURCHASE_PRICE_100	✓	✓
PDF Report	PDF	✓	
Competitive Set	COMPETITIVE_SET	✓	✓
Forecast	FORECAST		✓
List price	LIST_PRICE	✓	✓
Link to INDICATA	INDICATA_LINK	✓	✓

The PDF profile is only available in the Single API and *not* in the Bulk API. Please see [PDF Profile](#) for usage examples.

The Forecast profile is only available in the Bulk API and *not* in the Single API. Please see [Relation: Forecast Entry](#) for example usages.

Label/Value & CSV Identification Headers In Depth

Label	Example	Description
MAKE	Ford	The make of the vehicle
MODEL	Fiesta	The model of the vehicle
VARIANT	Fiesta 1.25 Ambiente	<p>The variant field can be used for text that does not quite fit into any other key. Typically this contains the features or highlights of the vehicle in short form. This is not for description or equipment, but identification relevant details, such as Trim, Engine, Body, etc.</p> <p>This is particularly useful if available information about a vehicle is in unstructured format.</p> <p>Lets say the client only have information about the vehicle in an unstructured String, for instance: <i>"5d Hatchback, 1.25 Duratec, Petrol, Manual Transmission, Ambiente"</i> Then the client could have a hard time splitting the String into the more precise keys, such as BODY, ENGINE, TRANSMISSION. It would be prudent to use VARIANT for such unstructured data.</p> <p>The best results are achieved by using the predefined keys for the correct data-values</p>
VEHICLENAME	Ford Fiesta 1.25 Duratec	The Vehicle Name has the same functionality as the Variant field. This is simply a secondary field.
DESCRIPTION	12Mot, Service History, Warranted Mileage	The description typically holds free-text about the vehicle, a resume.
EQUIPMENT	Sunroof Electric, Climate Control, Cruise Control, Seats Heated (Driver/Passenger), Alarm, Alloy Wheels (17in), Electric Windows (Front), In Car Entertainment (Radio/CD), Tinted Glass, Upholstery Leather	The Vehicle Equipment. To ease the identification we suggest that you separate the equipment with comma.
VEHICLETYPE	Demo	The Vehicle Type. Refer to Vehicle Type (AVT/PVT) for more information
CATEGORY	Passenger	Which category the vehicle belongs to, passenger or commercial
FIRSTREG	2012	The first registration of the vehicle (alternatively the year of manufacturing). See First Registration Date
MODELYEAR	2013	When the model was produced (typically used when the First Registration is not

Label	Example	Description
		known)
MODEL_PERIOD	2012-2017	<p>The period when the model was active. This is a range between 2 years separated by DASH, but we also support the shorthand single-year for very short-lived models.</p> <p>Range: "2012-2017" Shorthand: "2017"</p>
MODELYEAR_FROM	2012	When the model began production (left-hand side of model_period)
MODELYEAR_TO	2017	When the model ended production (right-hand side of model_period)
BODY	Panel Van, H2, L3, 3200kg	<p>This can contain multiple properties of the vehicle body, although typically it is simply a body type such as ESTATE. ViVi will also identify many trade names such as 'Avant' & 'Touring'.</p> <p>Examples of additional body properties</p> <p>Weight: 3200 kg</p> <p>Length: L2</p> <p>Height: H3</p> <p>Body Type and/or shape: SUV, Panel Van, Estate, Hatchback, MPV, Cabriolet, Saloon, Coupe, etc.</p>
WEIGHT	1100 kg	The weight of the vehicle
DOORS	5	The number of doors
ENGINE	1.25 Duratec 80HP 1242cc 82PS 60kW	The engine name and specification (including units)
ENGINELITERS	1299ccm	<p>The engine CC/CCM/LITERS, <i>including unit</i></p> <p>Examples 1.25 LITER 1299 CCM 1299cc</p>
FUEL	Petrol	The fuel type, Petrol, Diesel, Electricity
TRANSMISSION	Manual 5g	<p>The transmission type and number of gears.</p> <p>Many types are identified including 'Direct Drive', 'Manual', 'Semi-automatic', Automatic and trade names such as 'Steptronic'</p>
GEARS	5	The number of gears in the transmission (excluding reverse)

Label	Example	Description
WHEELDRIVE	FWD	The wheeldrive. Many types are identified including 'FWD', 'RWD', '4WD' and trade names such as 'Quattro'
SEATS	5	The number of seats
TRIM	Ambiente	The trim level name of the vehicle

The VARIANT label can be used for pretty much anything - duplicate values in multiple fields are completely OK as long as they do not contradict (in which case the results can lead to a conflict, see [Conflicting data](#)). For the best results we advise you to fill out as many fields as possible. In some cases it may even be prudent to concatenate others values and add them to VARIANT.

To obtain the best result for *your* particular data, try different strategies of mapping/duplicating information to that of ViVis Labels.

```
DOORS=5
VARIANT=5d ...
...
```

Refer to [Vehicle Type \(AVT/PVT\)](#) for more information about the VEHICLETYPE label.

Identification Country and Language

The Value part of the Label/Value input, is interpreted according to the *identification country*. In other words the interpretation of the Values are context aware. From the identification country, INDICATA knows how to interpret values in different languages.

Warning

Please note that the Accept-Language HTTP header does not have any influence on the interpretation of the Label/Value input, the Accept-Language HTTP header only has an effect on the presentation language in the response output.

To summarize with a few examples:

- In DK, Saloon body type is called "Sedan"; in IT, Saloon body type is called "Berlina"
- In UK, Gasoline is called "Petrol"; in FR, Gasoline is called "Essence"
- In DE, Sliding Door is called "Schiebetür"; in ES, Sliding Door is called "Puerta corrediza"
- etc.

Because the Value part of the Label/Value Input is context aware, the language used in the Value parts are important. E.g. body type "Berlina" (Italian language) is not recognized in SE (Sweden), because Italian is not a formal language of Sweden.

If your data is in Swedish, then use Sweden (SE) as your identification country, if your data is in Italian, then use Italy (IT) as your identification country, and so on.

Info

English terms are typically recognized in any identification country, as it is the base language used in the INDICATA catalog.

Many countries have a single official language, like Denmark (DK), which only has Danish (da) as official language. Some countries have multiple (more or less) official languages, such as Belgium (BE), where languages includes Dutch (nl), French (fr) and others. INDICATA typically supports the most common languages

used in the Automotive Industry, however should some language not be supported for a particular country, and it is commonly used in the Automotive Industry, please contact support.

Vehicle Type (AVT/PVT)

Vehicles can be identified as having a certain *type* and thus be valued according to that type.

This means that INDICATA will be able to

- * Adjust rules w.r.t which vehicles are eligible to participate in statistical basis for the valuation (e.g. including otherwise *quarantined* vehicles, which exhibit non-used vehicle-like properties).
- * Sometimes narrow the Competitive Set of vehicles to only include vehicles of the same type (depending on the prevalence of such vehicles in the current market).

Generally speaking, a Vehicle Type consists of 4 independent values:

- DEMO
- IMPORT
- PREREGISTRATION
- NEW

The configuration and meaning of a particular type is country-dependent and subject to change. As such it is out of scope of this document, please contact support for more information.

There are 2 categories of Vehicle Types

* **PVT**: Proclaimed Vehicle Type is identified as part of the description of a vehicle - in this context, only through text input (Label/Value or CSV Text)

* **AVT**: Automatic Vehicle Type is determined using account specific filters which can be setup in INDICATA

VehicleType is a dedicated Label/column in which keywords can be added (VEHICLETYPE=demo), but INDICATA will also recognize such keywords in other fields such as VARIANT, VEHICLE and DESCRIPTION. On the other hand, AVT is only applied during the valuation phase, because only then will INDICATA have enough information to actually run the AVT filter(s).

In particular if a vehicle is *not* identified to be of a particular type during the identification phase, AVT filtering will occur and might assign a type to the vehicle in the valuation phase.

To circumvent the AVT filtering, one can simply expand the *{vehicletype}* parameter in the Single API using a *-1* which disables AVT filtering.

Similarly to disable this in the Bulk API, set the column VehicleType to *-1*.

Precedence rules

PVT always supersedes AVT, thus if a vehicle is identified as IMPORT by identification (PVT), then any AVT filter(s) will have no effect.

A vehicle can only have 1 *type*, thus if it is identified as both DEMO and PREREGISTRATION, one of them will take precedence (regardless of it being PVT or AVT).

The rules of precedence are out of scope for this document, as they are country-dependent and subject to change. Please contact support for more information.

Identification & Valuation Caveats

The identification of a vehicle relies on INDICATA's vehicle catalog, which has some limitations.

One such limitation is the vehicle age, because INDICATA does not contain information about very old vehicles.

Another limitation is rare vehicles, which may not be present in the catalog.

These limitations can make an Identification impossible.

The exact limitations are beyond the scope of this document. If you have a question regarding a specific model, ask support for more information.

Deprecated Identification

On rare occasions it is necessary to retroactively update the INDICATA vehicle catalog. This typically happens when a catalog error is discovered.

When that happens, vehicle information *can* be obsoleted in favor of new catalog components. This requires a re-identification of a vehicle, because a backwards-incompatible change has been made in the catalog (to fix errors).

If this occurs, it is reported when valuating a vehicle which has previously been identified.

Country Specific Components

Some components of a vehicle identification may be country specific and the vehicle identifications are *not* cross-country comparable.

Examples of components include Body, Facelift, Trim.

Although these components are not directly comparable, it is possible to get a valuation in another country, using the *valuationcountry* parameter.

In practice, INDICATA will find competitive vehicles which may differ significantly (with regard to the country specific components). This may affect the valuation as the basis for the computation may be a broader set of competing vehicles.

Valuation without trim

Although a vehicle can be valued without selecting the appropriate trim, the valuation will be conservative (it can be low).

This is because a trim level can have a significant impact on the vehicle market price; some trims include a lot of equipment while others include virtually nothing and can thus represent significant differences in values.

When the trim is not specified, INDICATA will assume a 0-trim value and normalize the market prices of the competitive vehicles, in order to compensate for trim levels. In effect, this will cause the valuation to be conservative, especially if it just so happens that the valued vehicle actually has a premium trim level.

This can be somewhat mitigated by specifying the equipment. This is currently only possible when using Label/Value inputs (Single or Bulk). However, to the extent Vehicle Register Providers supply equipment specifications, ViVi will use this. Note that ViVi does not support manually selecting equipment in the sequence.

Assumptions

When not using the sequence to identify a vehicle, the data is either given to us directly from the client or an identifier is used to lookup information from an external source (Vehicle Register). The data can vary from virtually non-existing to being a full description.

A Single API Registration Number example from the Danish Vehicle Register (Motorregistret, DMR) - without assumptions.


```
{
  "identifier": "4J4FJB8S4NL132945",
  "nextStep": ...,
  "providerData": ...,
  "country": "DK",
  "category": ...,
  "make": ...,
  "model": ...,
  "regDate": ...
}
```

From this example, it is clear that INDICATA was not able to obtain much information from the Vehicle Register - only enough to identify Category, Make, Model and Registration Date.

Thus, the bootstrapping of the sequence did not spare the client of many steps. Enter assumptions.

Assumptions In depth

Assumption allows INDICATA to make educated guesses about the unidentified components.

 **Info**

As this is just a heuristic based on the catalog and what little information is available to INDICATA, different results may occur even when invoking the API with the same parameters. This is typically caused by catalog updates and pricing fluctuations on the market.

There are 3 types of assumptions, all 3 explained in the table below.

Assumption	Description
NONE	<p>This forces INDICATA to never make any guesses about components. This will most often result in an interactive session in the sequence before the vehicle is fully identified.</p> <p>NONE is recommended for interactive clients. This is the default for the Single API</p>
ADVANCED	<p>This allows INDICATA to make assumptions about Engine, Transmission, Wheeldrive, Weight. However no assumptions regarding Body, Length, Height or Facelift. This will sometimes be enough for PASSENGER vehicles to be identified fully. However COMMERCIAL vehicles will only bootstrap the sequence halfway through, thus requiring manual intervention through the sequence before the vehicle is fully identified.</p> <p>Typically used for interactive clients, where pictures are used to identity facelift and body.</p>
FULL	<p>INDICATA will assume as much as possible. This will most likely cause a Valuation Link to be present directly in the Response, though this is not guaranteed. INDICATA cannot make assumptions when the data is extremely sparse.</p> <p>FULL is recommended for automated systems, without human interaction. Sometimes this can result in failed or inaccurate identifications, when the data input is insufficient or</p>

Assumption	Description
	inaccurate.
	This is the default for the Bulk API.

Using the previous example with FULL assumption, we see that INDICATA has managed to identify the vehicle to a point where it is possible to value the vehicle (a Valuation Link is present). The only missing identification is the trim level.

```

{
  "valuation": ...,
  "identifier": "4J4FJB8S4NL132945",
  "nextStep": ...,
  "providerData": ...,
  "country": "DK",
  "category": ...,
  "make": ...,
  "model": ...,
  "regDate": ...,
  "body": ...,
  "facelift": ...,
  "seats": ...,
  "engine": ...,
  "transmission": ...,
  "wheelDrive": ...
}

```

The nextStep in the original example (with assumption NONE) is BODY, which is why using ADVANCED is no different from using NONE in this particular case.

Faulty assumptions

Since assumptions make educated guesses about the data presented, faulty assumptions can occur. An example of this could be the following input:

Type	Value
Country	DK
Make	Dacia
Model	Logan
Doors	4
Body	Stationcar (Estate)
Variant	MCV

In DK, an estate always has 5 doors, also MCV is an MPV, not an estate. Last but not least this particular vehicle does not exist in the INDICATA catalog with an Estate body (as of writing) - it only exists as a Saloon (Sedan) or an MPV. This causes confusion in INDICATA and it will be assumed to be a Saloon (Sedan), which is faulty and will have an impact on the valuation of the vehicle.

Unfortunately these contradictions occur in third party source data, such as Vehicle Register data or client Label / Value input data. That is why it is possible to go back in the sequence and select an alternate vehicle.

On rare occasions this can create a conflict in INDICATA, which is unresolvable - these are called conflicts.

Conflicting data

When data is ambiguous, such as is the case in the example in [Faulty assumptions](#), INDICATA may not be able to choose a single identification - these are known as conflicts.

Conflicting data can cause assumptions not to function properly and can short-circuit the identification of a vehicle. This can cause the sequence to stop in what appears to be an unrelated step.

As an example, lets create a conflict in the sequence (similarly, this could be done in the Bulk API with CSV Inputs), by using a conflicting transmission.

```
MAKE=MERCEDES-BENZ  
MODEL=C  
VARIANT=180 KOMP. AUT  
FIRSTREG=2007-06  
CATEGORY=Passenger  
BODY=Sedan  
DOORS=4  
ENGINE=1796 ccm 105kW  
FUEL=PETROL  
TRANSMISSION=7g
```

```
{  
  "nextStep": [  
    ... body step ...  
  ],  
  "country": "DK",  
  "category": ...,  
  "make": ...,  
  "model": ...,  
  "regDate": ...  
}
```

The conflict is created by the Transmission Steps (or Gears). The 180 Komp. engine does not exist with 7g (a later BlueEFFICIENCY version does exist with 7g). Removing Transmission from the Label / Value Input will result in a full Identification (bootstrapping of the sequence), apart from the Trim.

```
MAKE=MERCEDES-BENZ  
MODEL=C  
VARIANT=180 KOMP. AUT  
FIRSTREG=2007-06  
CATEGORY=Passenger  
BODY=Sedan  
DOORS=4  
ENGINE=1796 ccm 105kW  
FUEL=PETROL
```

```

{
  "nextStep": [
    ... trim step ...
  ],
  "valuation": [
    {
      "variants": [
        9813
      ],
      "seats": 5,
      "href": "/vivi/v2/HFDS543MFDS543FDL5/DK/9813::5/valuation{/profiles}?regdate=2007-06{&odometer,condition,vatreclaimable,geopostalcode,georadius}",
      "type": "application/json",
      "rel": "valuation",
      "templated": true
    }
  ],
  "country": "DK",
  "category": ...,
  "make": ...,
  "model": ...,
  "regDate": ...,
  "body": ...,
  "facelift": ...,
  "seats": ...,
  "engine": ...
  "transmission": {
    ...
    "summary": "Manual 6",
    "assumed": true,
    ...
  },
  "wheelDrive": ...
}

```

These conflicts occurs both in third party source data as well as Label / Value Input as shown here.

As such INDICATA cannot alleviate those as-is, since they are dependent on the input data. These require manual intervention.

Guides

Language (i18n)

The API supports internationalization (i18n) through the Accept-Language HTTP header:

```
Accept-Language: sv-SE
```

The *Accept-Language* HTTP header adheres to [RFC 2616](#) (section 14.4), with language (primary) being 2 digit ISO 639 and country (subtag) being ISO 3166 country code.

The default *Accept-Language* is “en-GB”.

- The *Accept-Language* HTTP header is not to be confused with endpoints, which specifies an ISO 3166 Country Code (which is used to identify the market) or CSV files containing countries. The Accept-Language HTTP header is purely for presentational purposes
- Not all languages are supported, ask support for more information

HATEOAS & Templated Links

Using HATEOAS and templated uris ([RFC 6570](#)) is extremely simple. There are many libraries for expanding templated links for many languages (see a list here: <https://code.google.com/archive/p/uri-templates/wikis/Implementations.wiki>)

Example of expansion:

```
// get start endpoints from the base resource
baseResource = client.get("https://ws.indicata.com/vivi/v2/bulk")

// filter the link by relation
link = get_link_by_relation(baseResource, "identification")

// expand the link
url = UriTemplate.parse(link.href).expand({assumption: "FULL"})

// now you can use the url for making HTTP Requests
// Responses are HATEOAS compliant and will contain new templated links for further interaction with the services
```

Polling Async Status

The output is an [AsyncStatus](#) entity.

Initially there is only 1 link in the response body, which is an async status polling endpoint, where the status can be polled at regular intervals.

The status link is not templated and you do not need to expand a UriTemplate, you can use the links *href* property directly.

As this is a Bulk API, 100.000+ of vehicles are being processed at a time, which means the Bulk Jobs are typically running for several hours. Thus it is not necessary to poll very often, perhaps every 10 minutes or so, depending on the size of your input.

Warning

Do not attempt to poll more often than every 5 seconds or you will receive a *429 Too Many Requests*

The status field of the response body determines how to proceed.

```
// QUEUED: the job is queued, pending other bulk jobs
// action: get the status link and schedule a new poll in X minutes
{
  "status": "QUEUED",
  "links": [
    {
      "rel": "status",
      "href": "https://ws.example.com/foo/bar/249aa5b5-77f5-44c9-bb72-e71fe23de655/status",
      "type": "application/json"
    }
  ]
}

// PROCESSING: the job is currently being processed
// action: get the status link and schedule a new poll in X minutes
{
  "status": "PROCESSING",
  "total": 25000,
  "remaining": 14631,
  "links": [
    {
      "rel": "status",
      "href": "https://ws.example.com/foo/bar/249aa5b5-77f5-44c9-bb72-e71fe23de655/status",
      "type": "application/json"
    }
  ]
}

// COMPILING: the final report is being generated
// action: get the status link and schedule a new poll in X minutes
{
  "status": "COMPILING",
  "total": 25000,
  "remaining": 0,
  "links": [
    {
      "rel": "status",
      "href": "https://ws.example.com/foo/bar/249aa5b5-77f5-44c9-bb72-e71fe23de655/status",
      "type": "application/json"
    }
  ]
}

// COMPLETE: the job has completed and the report(s) are ready
// action: see description below
{
  "status": "COMPLETE",
  "total": 25000,
  "remaining": 0,
  "links": [
    ....
  ]
}
```

When the status becomes *COMPLETE*, then the *hateoas* link relation “status” will disappear and new relation links will appear.

These links are described in further detail where appropriate.

Navigating Paged Resources

The maximum page size is 1000 elements per request (1000 elements is also the default page size).

The pages are 0-indexed and range from 0 - total number of pages. The default page is the first page, page 0.

Warning

Attempting to use a page size greater than 1000 or less than 1 will result in an error response *400 Bad Request*

It is rarely necessary to use a page size different from the default (1000), but if your server has limited resources in terms of RAM and you are using a memory-intensive deserialization library, you may consider using a smaller page size, in order to limit the memory impact on your server.

Since paged resources are templated, you will need to expand the links.

```
// assuming the link is "foo/bar{?page,size}"

// if you just want to use the defaults, then you can expand with an empty object
urlDefault = UriTemplate.parse(link.href).expand({}) // foo/bar

// if you wish to have more control, you could expand with the values you want to use
url = UriTemplate.parse(link.href).expand({page: 0, size: 1000}) // foo/bar?page=0&size=1000

// do note that because the default page size is 1000, both these links will contain the same response body
// foo/bar
// foo/bar?page=0&size=1000
```

It is pretty easy to loop over the pages, using either the “last” property and/or the checking for the existence of the `hateoas` link relation “next”. See more in type `Page`.

The relation links “next” and “previous” are not templated and you do not need to expand them. They can be used directly as is.

If you are using a custom page size, the relation links “next” and “previous” will respect your chosen page size.

Example with size = 48

```
url = UriTemplate.parse(link.href).expand({page: 1, size: 48}) // ...?page=1&size=48
res = client.get(url)
```

```
{
  "links": [
    {
      "rel": "next",
      "href": "https://ws.indicata.com/vivi/v2/bulk/identification/249aa5b5-77f5-44c9-bb72-e71fe23de655/preview?page=2&size=48",
      "type": "application/json"
    },
    {
      "rel": "previous",
      "href": "https://ws.indicata.com/vivi/v2/bulk/identification/249aa5b5-77f5-44c9-bb72-e71fe23de655/preview?page=0&size=48",
      "type": "application/json"
    }
  ]
}
```



```
url = UriTemplate.parse(link.href).expand({});
page = client.get(url)
// do stuff with the first page

while (!page.last) {
    page = client.get(page.links.next.href)
    // do stuff with all the subsequent pages
}
```

PRG pattern (Post/Redirect/Get)

The Post/Redirect/Get pattern is a common pattern, used when posting information to WebServices (<https://en.wikipedia.org/wiki/Post/Redirect/Get>).

The client POSTs a request, then the WebService responds with a REDIRECT containing a Location HTTP header. The client then uses a GET to retrieve the response.

```
-- client
POST /foo

-- server
HTTP/1.1 303 See Other
Location: /bar

-- client
GET /bar
```

Most client libraries will automatically follow redirects (or can be configured to do so), otherwise it is quite simple to create a code snippet which use the standard PRG pattern.

```
function prg(client, url, headers, data, contentType) {
    p = client.post(url, data, headers + contentType)
    r = p.getHeader("Location")
    g = client.get(r, headers)
    return g;
}
```

ViVi responds with HTTP Status Code 303, which means that the subsequent Request should use HTTP Request Method GET (as opposed to using the same HTTP Method as the original Request, as is the case for 307 Responses).

Character Encoding

Response

ViVi can respond with application/xml (Single API only) or application/json (Single API & Bulk API).

Both will always be returned in UTF-8 encoding and thus consumers are encouraged to hardcode this in readers and/or parsers, for maximum compatibility.

Always parse the response entities to be able to read/extract information. Use a Json library (or XML library), instead of trying to search with a Regular Expression or do direct substring extraction. Unfortunately, we have seen inexperienced developers try to substring instead of “just” parsing the response entity, and it will inevitably fail.

Do note, that the Content-Type HTTP header may contain charset information. These are all valid response HTTP headers and there are many libraries which can parse HTTP headers properly. Reference [RFC 7231](https://tools.ietf.org/html/rfc7231)

(section 3.1.1.5):

```
Content-Type: application/json
Content-Type: application/json; charset=UTF-8
Content-Type: application/xml
Content-Type: application/xml; charset=UTF-8
```

Do note that it is not necessary to parse the Content-Type response HTTP header, to have a working client. If the client requests a representation (using Accept HTTP header), then that representation will be in the response. If the service is not equipped to respond in the requested representation, a *406 Not Acceptable* response is returned (in accordance with [RFC 7231](#), section 6.5.6).

While it is not necessary to parse the Content-Type HTTP header, some will still choose to do so. If so, then do take care that you actually parse the HTTP header according to [RFC 7231](#) (section 3.1.1.5) and don't just do string matching directly in the HTTP header (as it may or may not contain charset information).

application/xml

ViVi will respond with application/xml in UTF-8 encoding, which is also specified in the XML document.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

When the server responds in XML, special characters must be escaped for the XML to be valid. Unfortunately we have seen multiple integration errors caused by this fact.

Consider an example response in XML from the sequence:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<vehicleIdentification>
  ...
  <nextStep>
    <link href="https://ws.indicata.com/vivi/v2/HFDS543MFDS543FDL5/DK?category=PASSENGER&make=530&model=98&regdate=2016&body=5124&facelift=1771&seats=5&bodyHeight=5124&bodyLength=5124&weight=-1&engine=8848&trim=65897&c-transmission=2441&wheeldrive=1365" type="application/xml" rel="wheeldrive" templated="false">
      <summary>FWD</summary>
      <description label="type">FWD</description>
    </link>
    <link href="https://ws.indicata.com/vivi/v2/HFDS543MFDS543FDL5/DK?category=PASSENGER&make=530&model=98&regdate=2016&body=5124&facelift=1771&seats=5&bodyHeight=5124&bodyLength=5124&weight=-1&engine=8848&trim=65897&c-transmission=2441&wheeldrive=1366" type="application/xml" rel="wheeldrive" templated="false">
      <summary>4motion</summary>
      <name>4motion</name>
      <description label="type">4WD</description>
    </link>
  </nextStep>
  ...
</vehicleIdentification>
```

The *href* attribute of the <link> xml tag, has been *escaped* as well as the text in the <summary> tag. Because the server will respond in *valid* XML only, **all** text are escaped, which means that text must be *unescaped* by the client.

As an example using Java, this can be achieved using `StringEscapeUtils.unescapeXml(...)` from [apache commons-text](#) library.

```
import org.apache.commons.text.StringEscapeUtils;
...
...
final String href = StringEscapeUtils.unescapeXml(link.getHref());

// input
// .../DK?category=PASSENGER&make=530&model=98&regdate=2016&body=5124...
// output
// .../DK?category=PASSENGER&make=530&model=98&regdate=2016&body=5124...

final String summary = StringEscapeUtils.unescapeXml(link.getSummary());
// input
// 4motion&apos;
// output
// 4motion'
```

Especially the *href* attribute is cause for confusion, because the links are not readily usable (hateoas), they *must* be unescaped, which should be very clear from the example above.

More information:

- * <http://xml.silmaril.ie/specials.html>
- * [RFC 7303](#) (especially section 3).

application/json

application/json must always be encoded as UTF-8, as per [RFC 8259](#) (section 8.1). ViVi will always return application/json in UTF-8 encoding.

Accept and Accept-Charset HTTP headers

Accept HTTP headers specifying charset will be ignored as per the sections regarding application/xml and application/json. That means charset specified in any Accept HTTP headers will not be honored.

```
// charset will be ignored
Accept: application/xml; charset=ISO-8859-1

// charset will be ignored
Accept: application/xml
Accept-Charset: ISO-8859-1
```

Request

When data is transferred from a client to ViVi it is imperative that ViVi knows the charset encoding, such that text is treated properly. Failure to communicate the charset can lead to poor identification.

While it is recommended to use unicode encodings, some legacy systems might still be using other encodings. Therefore ViVi can handle different encodings, but they must be explicitly relayed.

If for instance, you have data encoded in ISO-8859-15, you should specify this explicitly in the charset property of the Content-Type HTTP header:

```
Content-Type: text/plain; charset=ISO-8851-15
```

Default request charset

If the charset is not present in the Content-Type HTTP header, then ViVi will be lenient and treat the input as UTF-8.

Note: this is more lenient than [RFC 7231](#) for text/plain, which is specified in [RFC 6657](#) (section 4), to be US-ASCII (US-ASCII will still work as UTF-8 is a superset of that).

Bad charset implications

A very simple test is Škoda, since it uses a [Caron](#) above the S. There are many other special characters which need to be handled properly also; Citroën, etc.

The following table with some of the mostly used charsets should explain why.

Charset	byte (sequence)
UTF-8	197 160
UTF-16	254 255 1 96
ISO-8859-1	- (not available)
ISO-8859-2	169
ISO-8859-15	166
Window 1250 / Windows 1252	138

Do note that in the following examples, if a regular 'S' had been used, everything would have worked, even though specifying a wrong charset. That is because a-z and A-Z rarely switch positions in most standard charsets used (at least in Western Europe).

ISO-8859-1 encoded text

Š is not available in ISO-8859-1, so this case is simply impossible.

ISO-8859-2 encoded text

Encoded characters

Š	k	o	d	a
169	107	111	100	97

Interpretation

Content-Type	Interpretation	Notes
text/plain	◆koda	UTF-8 assumed because charset is unspecified
text/plain; charset=ISO-8859-15	©koda	169 is © in ISO-8859-15
text/plain; charset=ISO-8859-2	Škoda	Correct charset specified

ISO-8859-15 encoded text

Encoded characters

Š	k	o	d	a
166	107	111	100	97

Interpretation

Content-Type	Interpretation	Notes
text/plain	Škoda	UTF-8 assumed because charset is unspecified
text/plain; charset=Windows-1250	koda	166 is in Windows-1250
text/plain; charset=ISO-8859-15	Škoda	Correct charset specified

UTF-8 encoded text

Encoded characters

Š	k	o	d	a
197 160	107	111	100	97

Interpretation

Content-Type	Interpretation	Notes
text/plain	Škoda	UTF-8 assumed because charset is unspecified
text/plain; charset=UTF-8	Škoda	Correct charset specified
text/plain; charset=ISO-8859-1	Å koda	Š cannot be interpreted in ISO-8859-1
text/plain; charset=UTF-16	앨齒擬	UTF-16

UTF-16 encoded text

Encoded characters

Š	k	o	d	a
254 255 1 96 0	0 107	0 111	0 100	0 97

Interpretation

Content-Type	Interpretation	Notes
text/plain	Škoda	UTF-8 assumed because charset is unspecified
text/plain; charset=UTF-16	Škoda	Correct charset specified

More References

- [RFC 2616](#)
- [RFC 7231](#) (especially section 3.1.1.1, section 3.1.1.5 and appendix B)
- [RFC 6657](#) (especially section 4)
- [IANA Character Sets](#)
- https://en.wikipedia.org/wiki/Character_encoding
- [Simple Explanation of Character Sets](#)

Appendix

Types & Definitions

General

These are general types

Type: Assumption

Please refer to [Assumptions](#) for an in-depth discussion about assumptions.

Value	Description
NONE	No assumptions
FULL	Full assumptions on all components
ADVANCED	Assumptions only on Engine, Transmission, Wheeldrive, Weight

Type: Link

Property	Type	Description
rel	string	The relation of the link
href	string	The hypertext reference, possibly templated
type	string	The media type
template	boolean	Whether the link is templated or not

Type: BaseResource

Property	Type	Description
links	array (Link)	Hateoas links for navigation

Type: Whoami

Property	Type	Description
username	string	The username
name	string	The name
roles	array(string)	The roles which this user has access to
accessibleCountries	array(string)	The markets that this user can use wrt INDICATA valuations (and identification), 2 digit Country Code (ISO 3166-1 alpha-2 code)
accessibleStrategicCountries	array(string)	2 digit Country Code (ISO 3166-1 alpha-2 code)

Type: Description

Property	Type	Description
label	string	The label of the description
value	string	The value of the label

Type: Image

Property	Type	Description
key	string	Representation of the image
links	array (Link)	links to images

Type: NextStep

NextStep contains the same properties as a [Link](#) + the following properties

Property	Type	Description
summary	string	Opinionated text summary of the description
name	string	Optional name of the component
description	Description	A list of (unstructured) properties describing the component
images	array (Image)	Some steps have images to help the user to distinguish between which step to choose next - this property is only present when there are images. An example could be a Body or Facelift

Type: CategoryField

Property	Type	Description
key	string	Internal component representation
summary	string	Opinionated text summary of the description
name	string	Optional name of the component
assumed	boolean	If the component was assumed
link	Link	A link to go back to the selection of the element, as to replace the selection

Type: RegDateField

Property	Type	Description
key	string	Internal component representation
summary	string	Opinionated text summary of the description.
name	string	Optional name of the component
link	Link	A link to go back to the selection of the element, as to replace the selection

Type: CatField

Property	Type	Description
key	array (integer)	Internal component representation
summary	string	Opinionated text summary of the description
name	string	Optional name of the component
assumed	boolean	If the component was assumed
description	array (Description)	A list of (unstructured) properties describing the component
link	Link	A link to go back to the selection of the element, as to replace the selection

Type: VTl

Property	Type	Description
id	string	VTI (variant:trim:seats)
variantId	integer	The variant id
trimId (Optional)	integer	The trim id
seats (Optional)	integer	Number of seats

Type: ValuationLink

ValuationLink contains the same properties as [Link](#) + the following properties

Property	Type	Description
variants	array (integer)	Internal component representation
seats	integer	The number of seats
trim	integer	Internal component representation

Type: VehicleIdentification

Property	Type	Description
country	string	2 digit Country Code (ISO 3166-1 alpha-2 code)
nextStep	array (NextStep)	The HATEOAS navigation steps
valuation	array (ValuationLink)	The valuation link(s). In practice there will only be 1 link. We may introduce more valuation links in a future version, but not as of this version
category	CategoryField	Passenger or Commercial
make	CatField	
model	CatField	
regDate	RegDateField	The First Registration Date of the vehicle
body	CatField	

Property	Type	Description
bodyHeight	CatField	Typically only for Commercial vehicles
bodyLength	CatField	Typically only for Commercial vehicles
weight	CatField	Typically only for Commercial vehicles
facelift	CatField	
seats	CatField	
engine	CatField	
wheeldrive	CatField	
transmission	CatField	
trim	CatField	The trim is not required to value the vehicle. However it will affect the valuation, see more in Identification & Valuation Caveats .
equipment	array (CatField)	A list of equipments
vehicleType	CatField	The type of vehicle, see more in Vehicle Type (AVT/PVT)
links	array (Link)	A number of links which can be used for further operations
vti	VTI	

Type: OdometerUnit

Value	Description
MILE	GB Miles
SCANDINAVIAN_MILE	SE Scandinavian Miles
KILOMETER	Kilometer

Type: Odometer

Property	Type	Description
distance	integer	The distance of the odometer
unit	OdometerUnit	the Unit of the odometer

Type: OdometerType

Value	Description
EXPLICIT	The odometer is explicitly set by the client
STANDARD	INDICATA standard odometer estimation, based on the identified vehicle
FORECAST_RELATIVE	Only applicable for forecasting. The odometer is extrapolated (linearly) from the first and last forecast Date/Odometer pairs

Please contact support for questions regarding Odometer Estimation.

Type: OdometerWithType

The OdometerWithType contains the properties from [Odometer](#) + an indication of the type from where the value is derived.

Property	Type	Description
type	OdometerType	The type of odometer

Type: Condition

The Type Condition is reserved for future use

Property	Type	Description
grade	integer	Reserved for future use
scale	string	Reserved for future use

Type: VehicleAmount

Property	Type	Description
amount	Amount	The valuation amount
regTax	RegTax	The registration tax
vat	VAT	The VAT

Type: CompactVehicleAmount

Property	Type	Description
amount	Amount	The valuation amount
regTaxType	RegTaxType	The type of registration tax
vatType	VATType	The type of VAT

Type: Valuation

The Valuation contains the properties from [VehicleAmount](#) + the type of the valuation.

Property	Type	Description
type	string	The type of valuation, which is equal to the aforementioned profiles

Type: Amount

Property	Type	Description
value	float	The valuation
currency	string	3 digit ISO 4127 Currency Code

Type: RegTaxType

Value	Description
EXCL_REG_TAX	Excluding Registration Tax
INCL_REG_TAX	Including Registration Tax

Type: RegTax

Property	Type	Description
type	RegTaxType	The type of Registration Tax
country	string	2 digit ISO 3166 Country Code

Type: VATType

Value	Description
EXCL_VAT	Excluding VAT
INCL_VAT	Including VAT

Type: VAT

Property	Type	Description
vatType	VATType	The type of VAT
vatReclaimable	boolean (YES/NO)	If VAT is reclaimable in the given valuation
vatCountryCode	string	2 digit ISO 3166 Country Code

Type: MDS

Property	Type	Description
marketDaysSupplyOverall	integer	Market Days supply for all vehicles of the same Make, Model, Generation and Body Type
marketDaysSupplySimilar	integer	Market Days supply based on similar vehicles (the same criteria as the competitive set for the valuation)
marketDaysSupplyMarket	integer	Market Days supply for the full market
marketDaysSupplyMake	integer	Market Days supply for all vehicles with the same make
competitiveVehiclesSold	integer	The number of similar vehicles sold in the last 45 days

Type: PurchasePriceBreakdown

Please ask support for more details about this type.

Property	Type	Description
daysToSell	integer	
budgetedPriceReduction	integer	
seasonalDepreciation	integer	

Property	Type	Description
costReconSum	integer	
costWarranty	integer	
costMarketing	integer	
costInterest	integer	
costVAT	integer	
profit	integer	

Type: Criteria

Property	Type	Description
name	string	The name of the criteria
type	string	The type of the criteria
value	string	The value used to select competing vehicles

Type: VehicleValuation

The VehicleValuation type contains the identified components, which includes all properties from [VehicleIdentification](#) except for *nextStep* and *valuation*.

Furthermore, the valuation specific components are available, as described in the following table.

Property	Type	Description
odometer	OdometerWithType	Which odometer has been used
overallCondition	Condition	Reserved for future use
vatReclaimable	boolean (YES/NO)	Reclaimability of VAT
valuations	array (Valuation)	This contains the valuations requested
mds	MDS	This is the specialized valuation SUPPLY_DEMAND
maxPurchasePriceBreakdown	PurchasePriceBreakdown	Trade In Calculation Breakdown, shows “hidden” costs of buying a vehicle (same as INDICATA)
competitiveVehiclesForSale	integer	The number of competing vehicles for sale
competitiveVehiclesAverageOdometer	Odometer	The average odometer of the competing vehicles
competitiveVehiclesCriteria	array (Criteria)	The criteria used for selecting the competing vehicles
priceDate	string	ISO 8601 Date (yyyy-MM-dd) - the date this valuation was performed

Property	Type	Description
pdf	Link	The link to async pdf generation progress. Only present if requesting with profile PDF and only available in the Single API
links	array (Link)	A number of links which can be used for further operations
competitiveSet	array (CompetitiveVehicle)	The competitive vehicles used for calculation. Only present if requesting with profile COMPETITIVE_SET

Type: MaxExportPrice

The max export price contains 2 (compact) vehicle amounts, where one is in EUR and the other is in local currency. This is a simple currency conversion and does not include any TAX or VAT.

Property	Type	Description
vehicleAmount	CompactVehicleAmount	The max export price in EUR
localVehicleAmount	CompactVehicleAmount	The max export price in local currency. Note that this may also be in EUR, if origin country also uses EUR.

Type: EuropeanTargetPrice

The European Target Price type contains the identified components, which includes all properties from [VehicleIdentification](#) except for *nextStep* and *valuation*.

Furthermore, the european target price specific components are available, as described in the following table.

Property	Type	Description
exportIndex	integer	The export index in the range of 0-250
maxExportPrice	MaxExportPrice	The Maximum export price
odometer	OdometerWithType	Which odometer has been used
priceDate	string	ISO 8601 Date (yyyy-MM-dd) - the date this valuation was performed
reference	string	A simple metadata property which is set only if it was provided by the client
links	array (Link)	A number of links which can be used for further operations

Type: ComponentWithDiff

Property	Type	Description
name	string	The name of the component
value	float	The value difference this component contributes with. If no value difference, then the property <i>value</i> does not exist

Type: OdometerWithDiff

The OdometerWithDiff contains the properties from [Odometer](#) + the value difference the odometer contributes with.

Property	Type	Description
value	float	The value difference this component contributes with. If no value difference, then the property <i>value</i> does not exist

Type: VehicleAmountWithDiff

The VehicleAmountWithDiff contains the properties from [VehicleAmount](#) + the value difference the amount contributes with.

Property	Type	Description
type	string	The type of valuation, which is equal to the aforementioned profiles

Type: CompetitiveVehicle

Property	Type	Description
country	string	2 digit ISO 3166 Country Code
category	ComponentWithDiff	The category
regDate	ComponentWithDiff	The first registration
regDateMonthUnknown	boolean	If present and true, then the month of the first registration is unknown (only the year is available)
bodyHeight	string	The body height (typically only for Commercial vehicles)
bodyLength	string	The body length (typically only for Commercial vehicles)
facelift	ComponentWithDiff	The facelift
seats	string	The seats
weight	string	The weight (typically only for Commercial vehicles)
engine	ComponentWithDiff	The engine
transmission	ComponentWithDiff	The transmission
wheelDrive	ComponentWithDiff	The wheelDrive
trim	ComponentWithDiff	The trim
odometer	OdometerWithDiff	The odometer
priceToMarket	float	The Price/Market percentage of the competing vehicle
price	VehicleAmountWithDiff	The price of the competing vehicle
correctedPrice	VehicleAmountWithDiff	The corrected price of the competing vehicle, e.g. the 100% value of the vehicle with the total difference
seller	string	The name of the seller, that has the vehicle for sale
excluded	boolean	True if the vehicle is not used in the calculation

Type: ForecastInput

Property	Type	Description
date	string	ISO 8601 Date (yyyy-MM-dd) - the date with which to do the forecast
odometer	integer	The odometer with which to perform the forecast - note that the odometer unit is implicit, depending on the context (country), see more in Odometer Unit & GeoRadius Unit
id	string	Optional parameter which can be used to identify the specific date-odometer pair

Type: ForecastResult

Property	Type	Description
country	string	2 digit ISO 3166 Country Code
forecastDate	string	ISO 8601 Date (yyyy-MM-dd) - the date this forecast was performed
regDate	string	The first registration date of the vehicle
vti	string	internal representation of the vehicle
outcomes	array (ForecastOutcome)	The results of the forecast inputs

Type: ForecastOutcome

Property	Type	Description
successful	boolean	Whether or not this outcome is a success or failure
success	ForecastSuccess	If 'successful' is true, then this property is present
failure	ForecastFailure	If 'successful' is false, then this property is present

Type: ForecastSuccess

Property	Type	Description
date	string	ISO 8601 Date (yyyy-MM-dd) - the forecast date
age	integer	The age (in months) from now
odometer	integer	The <i>actual</i> used odometer, which may differ slightly from the input odometer
retailPrice	float	The forecasted retail price
maxPurchasePrice	float	The forecasted Max Purchase Price

Type: ForecastValuation

Property	Type	Description
forecast	ForecastResult	The forecast
valuation	VehicleValuation	The accompanying valuation
links	array (Link)	Hateoas links for navigation

Type: ForecastFailure

Property	Type	Description
errorCode	integer	The error code
errorDescription	string	The description of the error

Sequence Specific

These are sequence specific types

Definition: Sequence Identification Steps

Value	Description
Category	Passenger or Commercial
Make	
Model	
Reg Month	The registration month step is not present in all countries, e.g. UK uses Age Identifiers. See First Registration Date
Reg Date	The First Registration Date of the vehicle
Body	
Facelift	
Seats	
Body Height	Typically only for Commercial vehicles
Body Length	Typically only for Commercial vehicles
Weight	Typically only for Commercial vehicles
Engine	
Wheel Drive	
Transmission	
Trim	This step is not required to value the vehicle. However it will affect the valuation, see more in Identification & Valuation Caveats .

Type: GroupType (enum)

Value	Description
Vehicle	
Registration	
Inspection	
Engine	
Body	
Weight	

Value	Description
Environment	
Pricing	
Transmission	

Type: ProviderEntry

Property	Type	Description
key	string	The key
value	string	The value of the key

Type: ProviderGroup

Property	Type	Description
type	GroupType	The type of the group
entries	array (ProviderEntry)	array of entries

Type: ProviderData

Please note that the external vehicle register providers do not expose the same information across providers. Thus the provider data significantly differs when using lookups in different countries.

Property	Type	Description
providerName	string	The name of the external vehicle register provider
providerGroups	array (ProviderGroup)	The groups of the provider data

Type: VehicleRegisterIdentification

The VehicleRegisterIdentification type contains the same properties as [VehicleIdentification](#) + the following properties

Property	Type	Description
identifier	string	Either VIN (Vehicle Identification Number), Registration Number or Exid
providerData	ProviderData	Additional data from the external Vehicle Register Provider

Bulk Specific

These are bulk specific types

Definition: Bulk Identification Steps

These are the steps associated with Bulk Identification.

They differ a bit from the Sequence Steps, in that there is no Registration Month Step. Also there is an Odometer Step.

Value	Description
Category	Passenger or Commercial
Make	
Model	
RegDate	The First Registration Date of the vehicle
Body	
BodyHeight	Typically only for Commercial vehicles
BodyLength	Typically only for Commercial vehicles
Weight	Typically only for Commercial vehicles
Facelift	
Seats	
Engine	
Transmission	
Wheeldrive	
Trim	This step is not required to value the vehicle. However it will affect the valuation, see more in Identification & Valuation Caveats
Odometer	The odometer step is used to signify the identification state of the odometer CSV input field

Type: JobStatus (enum)

Value	Order	Description
QUEUED	1	Pending processing
PROCESSING	2	The Bulk Job is currently being processed
COMPILING	3	The Bulk Job has completed processing and the outputs are being compiled
COMPLETE	4	The Bulk Job has completed both processing and compiling and is ready for consumption
ERROR	N/A	Internal Server Error - please contact support if this should ever happen

Type: AsyncStatus

Property	Type	Description
status	JobStatus	The status of the Bulk Job
links	array (Link)	Hateoas links for navigation
total	integer	The total number of vehicles in the Bulk Job. This property is not available when status is QUEUED
remaining	integer	The remaining number of vehicles (pending completion) in the Bulk Job. This property is not available when status is QUEUED

Type: **ComponentStatus** (enum)

Value	Description
IDENTIFIED	When a component has been identified (recognized)
ASSUMED	The component could not be identified, but was assumed to be a certain value (see Assumptions)
FAILED	When an identification of the component was not possible

Type: **BulkVehicleStatus** (enum)

Value	Description
SUCCESS	When the vehicle has been identified to allow for a valuation. A vehicle does not need to have “trim level” identified, for it to be valued. However, the valuation will be quite conservative if the trim level is unknown.
FAILURE	When the vehicle has only been partially identified and a valuation is not possible
CLIENT_ERROR	When there was an input error in the CSV
SERVER_ERROR	When the server experienced an error, from which it could not recover. Please contact support if this occurs.

Type: **BulkResult**

Property	Type	Description
status	BulkVehicleStatus	The status of the vehicle
reason	string	Contains a textual reason explaining the status This property is not present if the status is SUCCESS

Type: **BulkIdentificationSummary**

Property	Type	Description
taskId	string	The Bulk Job Id
creator	string	Username of the WebService User
noEntries	integer	The number of vehicles in the Bulk Job
errors	integer	The number of errors (counted by vehicle) in the Bulk Job These are the vehicles in the Bulk Job which have a BulkVehicleStatus of <i>CLIENT_ERROR</i> or <i>SERVER_ERROR</i>
start	string	When the job started. ISO 8601 DateTime (yyyy-mm-dd'T'HH:mm:ss.zzz'Z')
finish	string	When the job finished. ISO 8601 DateTime (yyyy-mm-dd'T'HH:mm:ss.zzz'Z')

Property	Type	Description
tally	Map (string, Map (ComponentStatus , integer))	This is a mapping of the Bulk Identification Steps to ComponentStatus and a number, describing the number vehicles of each ComponentStatus per step in a Bulk Job.

Example of tally

```
"tally": {
  "category": {
    "IDENTIFIED": 25000,
    "ASSUMED": 0,
    "FAILED": 0
  },
  "make": {
    "IDENTIFIED": 25000,
    "ASSUMED": 0,
    "FAILED": 0
  },
  ...
}
```

Type: BulkVehiclePreview

Property	Type	Description
input	Map (string, string)	A map containing the csv inputs for this vehicle
result	BulkResult	The result for the current vehicle
nextStep	string	The next component to be identified. For a full valuation, nextStep is empty, while nextStep is non-empty if the vehicle is not fully identified
statusMap	Map (string, ComponentStatus)	A map containing “step” as key and “ComponentStatus” as value. This allows you to see which steps has which ComponentStatus

Type: BulkVehicleIdentification

Property	Type	Description
input	Map (string, string)	A map containing the csv inputs for this vehicle
result	BulkResult	The result for the current vehicle
identification	VehicleIdentification	The vehicle identification

Type: BulkVehicleValuation

Property	Type	Description
input	Map (string, string)	A map containing the csv inputs for this vehicle
result	BulkResult	The result for the current vehicle

Property	Type	Description
valuationStatus	Map (string, BulkVehicleStatus)	The status of the requested profiles. The keys are valuation profiles and the values are the status.
identification	VehicleIdentification	When using valuation delegation, with unsuccessful identifications, then this field is populated with the partial identification.
valuation	VehicleValuation	The result of the valuation (if successfully identified)
forecast	ForecastResult	The result of the forecast (if successfully identified + valuated and forecast profile was enabled)

Type: Page

Property	Type	Description
links	array (Link)	Hateoas links for navigation
last	boolean	true if current page is the last page
first	boolean	true if current page is the first page
totalElements	integer	The total number of elements when counting all pages
totalPages	integer	The total number of pages, given the current page size
numberOfElements	integer	The number of elements present in the current page (the number of elements in the content array) Note: this will always equal the page size (size property) except for the last page, where it may be less than the page size
size	integer	The page size
number	integer	The page number
content	array (<T>)	The content of the page

The content <T> will be specified, in the relevant documentation.