

**TNO report**

**DST Standard Product File version 2.1**

**Technical Sciences**  
Kampweg 5  
3769 DE Soesterberg

Postbus 23  
3769 ZG Soesterberg

[www.tno.nl](http://www.tno.nl)

T +31 88 866 15 00

F +31 34 635 39 77

Date	18 April 2016
Author(s)	Jasper Roes, MSc.
Number of pages	124 (including attachments)
Commissioned by	Stichting DST

All rights reserved.

Nothing in this publication can be duplicated and/or published by print, copy, microfilm or in any other way, without explicit approval of TNO.

If this publication was commissioned, we refer to TNO's Terms and Conditions that apply for the commissioner and the acceptor of the project, and to the agreement all parties involved agreed upon.

Showing the TNO-report to immediate stakeholders is allowed.

© 2016 TNO

## Version Management

Version	Date	Remarks, changes
1.0	2011-12-27	<ul style="list-style-type: none"> <li>• 1.0 version standard</li> </ul>
1.1	2012-01-25	<ul style="list-style-type: none"> <li>• 1.1 version standard</li> </ul>
2.0	2016-02-14	<ul style="list-style-type: none"> <li>• 2.0 version standard. Added, among other things, are: <ul style="list-style-type: none"> <li>◦ Additional containers for electrical bikes</li> <li>◦ More details possible for parts</li> <li>◦ Small improvements in definitions</li> </ul> </li> </ul>
2.1	2016-04-18	<ul style="list-style-type: none"> <li>• 2.1 version standard. These improvements and changes are made: <ul style="list-style-type: none"> <li>◦ Profiles: deleted container, profiles are placed directly in article information.</li> <li>◦ Special gross cost price: changed datatype and added multiplicity.</li> <li>◦ Battery capacity: datatype added.</li> <li>◦ Pos-image: csv fieldname corrected.</li> <li>◦ Brand, model and type primary rear brake: csv fieldnames corrected.</li> <li>◦ Brand, model and type secondary rear brake: csv fieldnames corrected.</li> <li>◦ Brand, model and type front brake: csv fieldnames corrected.</li> <li>◦ Container general product information and sub elements: level headers adjusted.</li> <li>◦ Container related products and sub elements: level headers adjusted.</li> <li>◦ User documentation: corrected definition type, url and datatype changed, separator in CSV implementation rules changed.</li> <li>◦ Technical documentation: corrected definition type, datatype changed, separator in CSV implementation rules changed.</li> <li>◦ Frame material: datatype added.</li> <li>◦ Drivetrain brand: corrected definition.</li> <li>◦ Drivetrain type: added example.</li> <li>◦ Battery capacity: datatype changed.</li> <li>◦ Battery voltage: example corrected.</li> <li>◦ Drivetrain type: added example.</li> <li>◦ Battery ampere-hour: example corrected.</li> <li>◦ Recharge time normal charger: example corrected.</li> <li>◦ Recharge time fast charger: example corrected.</li> <li>◦ Battery/range: the implementation rule for cvs is adjusted for this container, with this it is no longer possible to specify multiple batteries for one electric bike. Besides that it is no longer possible to specify multiple ranges. For these elements a cvs filename has been added: setting range, min range, max range, average range, battery included in base price, extra cost battery, position battery, battery removable, battery chargeable on bike.</li> </ul> </li> </ul>



# Index

<b>Index</b>	<b>4</b>
<b>1 Introduction</b>	<b>6</b>
1.1 Goals	6
1.2 Overview of exchanges in the sector	6
1.3 Design perspective	9
1.4 Reading guide	10
<b>2 Information model</b>	<b>11</b>
2.1 Language	13
2.2 Country	13
2.3 Currency	14
2.4 Identifiers	14
2.5 Orderinformation	17
2.6 Product classification	21
2.7 Product information	25
2.8 Bike specifics	50
2.9 Electric bike specifics	95
2.10 Battery specifics	108
2.11 Display specifics	113
2.12 Engine specifics	116
<b>3 Code/value lists</b>	<b>119</b>
3.1 Introduction	119
3.2 Use of code/value lists	119
3.3 Codelist Parties	119
3.4 Codelist Statuses	119
3.5 Codelist Product groups	120
3.6 Codelist Customer groups	120
3.7 Codelist Keywords	120
3.8 Codelist Position	120
3.9 Codelist Brands	120
3.10 Codelist Base colours	120
3.11 Codelist VAT	120
3.12 Codelist ERTRO	120
3.13 Codelist Frame types	120
3.14 Codelist Frame materials	120
3.15 Codelist Gear types	120
3.16 Codelist Brake types	121
3.17 Codelist Document types	121
3.18 Codelist Ebike systems	121

3.19	Codelist Electric bike types	121
3.20	Codelist Sensor types	121
3.21	Codelist Battery position	121
3.22	Codelist Engine position	121
3.23	Codelist Battery types	121
3.24	Codelist Display types	121
3.25	Codelist Display operation types	121
<b>4</b>	<b>CSV syntax</b>	<b>122</b>
4.1	Introduction	122
4.2	Layout of CSV	122
4.2	Implementation of CSV elements	123
<b>5</b>	<b>JSON syntax</b>	<b>124</b>
5.1	Introduction	124
5.2	Layout of JSON	124

# 1 Introduction

## 1.1 Goals

The goal for the Dutch foundation for digital cooperation in the cycling sector (DST) is to create a standard for product files. These product files are exchanged by suppliers/ manufacturer, wholesaler and retailers.

Product files are the basis for a lot of processes, such as:

- Product catalogs
- Stock management
- Invoicing
- Ordering
- Etc.

At this moment every party involved has different product files:

- More or less data elements
- Differences in the used syntax
- Differences in the interpretation of the data elements
- Differences in the quality of the supplied data

The standard that is described in this document is supposedly going to be the standard for streamlining the processes in the sector.

## 1.2 Overview of exchanges in the sector

This standard doesn't contain a back-breaking description of actual data exchange processes. But it is desirable to get an idea of the opportunities the standard should support.

In the diagram below you can see an overview of all parties who should be able to work with the standard.



Image: overview of parties involved

- **Suppliers** (or: manufacturers, wholesalers) - produce products. This could be a composed product (like a bicycle) or a (more or less) base product (like a tyre or gear).
- **Importers** bring a foreign product to the Dutch market. This could mean that they only provide the logistics to bring the product to the Netherlands, but it could also mean that they do more (like repacking or relabelling, etc.).
- **Wholesalers** buy products from importers and suppliers and sell these to retailers. Sometimes a wholesaler is also the importer (or even supplier); these roles can overflow depending on the chosen definition.

In general suppliers, importers and (to a lesser extent) wholesalers will be the source of product data, as they are bringing the product on the market and should be familiar with the possibilities and specifications.

Then there are a few parties that interpret and process the data:

- **Retail Service Organisations** (Biretco and Bike Totaal) merge data for retailers. During this merge they can enrich the data in case the supplied data isn't sufficient.
- To supply data to POS-systems of retailers, multiple suppliers of these POS-systems are involved. These suppliers often offer a value added service, in which product files can be processed through a central server. Retailers can collect product files from multiple affiliated suppliers/importers/wholesalers from that central server.
- At this moment the implementation of a sector platform is being discussed. This platform should combine product files from multiple suppliers in to one file for the retailer. The exact scope and functionality of such a branch platform is not known at this moment. Besides, a few suppliers have mentioned they do (also) want to exchange product data outside of the platform.

In the end there are the retailers who receive the product data. At this moment most retailers have multiple product files. These files are periodically collected by their POS-system. In the POS-systems they are used for their administration. They could also be used for stock management and ordering products.

A few exchange formats have been discussed in the working group:

#### *Supplying product file*



Image: supplying product file

To supply a product file to a retailer there are roughly two possibilities:

- Supplying the product file directly to the retailer, who can import the file in to his/her POS-system.
- Supplying a product file to a branch platform, who enriches the file. In an ideal situation the retailer then receives one product file, with all products he can order from all affiliated suppliers.

An important technical discussion is the way the POS-systems are supplied with data. Right now mainly cvs-files are used, and they are sent to the POS-systems via FTP. Perhaps in the future the use of XML based exchange systems (like web services) will be possible/ desirable.

#### *Supplying stock and prices*

Several parties have expressed themselves for supplying stock and prices to the retailer. For this they foresee a web service which will supply this data directly to the retailer.



Image: supplying stock and prices to the retailer

As the price information is potentially sensitive, parties do not think its desirable to exchange this data via the platform (other than for example the RRP).

#### *Orders*

The other way round orders can be sent from retailers to their supplier. Here too a direct connection is anticipated, for example based on a web service.



Image: an order being sent



The above scenario's are (at this moment) not a part of the standard. This means that for these scenario's either decision will have to be made on sector level or that an exchange method will have to be chosen for all implementations.

### 1.3 Design perspective

The work group has named five design perspectives that are used to draw this standard:

- One model for product data

Within the sector there has to be one model for product data. For some kinds of products specific features will be defined. The basic information for the different kinds of products will have to be the same as far as possible.

- One product will be uniquely identified by the combination of 'supplier' and 'item number'

There are more than one ways to identify a product. A much used option is using the EAN-code, but there are other options. Besides it is possible that multiple suppliers carry that product (with the same EAN-code), but these suppliers can provide different specifications for some aspects (for example a difference in order amounts). This is why, on technical level, a product always gets uniquely identified by the combination of 'supplier' and 'item number'. Based on other information (like the EAN-code) products can thereafter be linked.

- There is a codelist for much used containers

For a number of much used containers, that are often identifying for a certain product, there is a code- or value-list. These lists are part of the standard.

- Independent implementation

The standard must support multiple ways of implementation of data exchange.

- Clarity about optional and mandatory containers

It must be clear which containers always have to be filled, which containers are optional and which containers have to be filled in certain situations.

## 1.4 Reading guide

The standard is structured as follows:

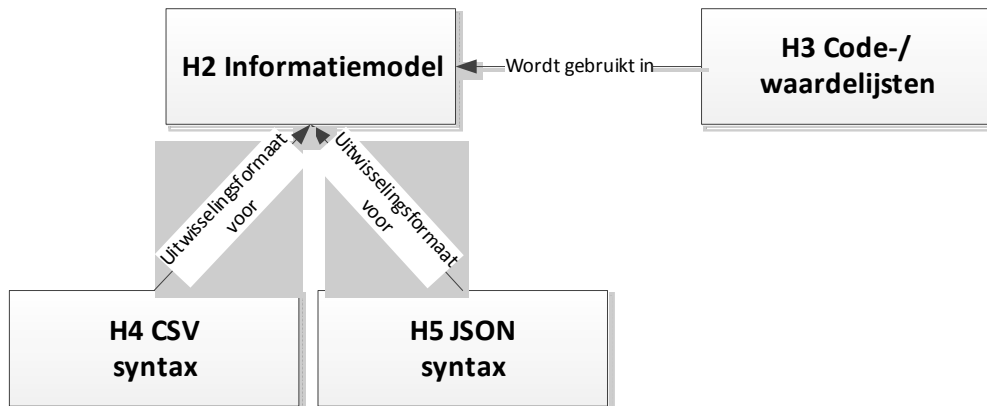


Image: structure of the standard

The basis of the standard is formed by the information model. This describes the various specifications of a product. The information model is described in chapter 2.

For a few data elements a code-/value-list is set. These lists are described in chapter 3.

Then two syntaxes for the actual exchange of productdata are described:

- An implementation using CSV - this format is often used for importing and exporting data from databases and for example POS-systems. This is described in chapter 4.
- An implementation using JSON - this is a much used standard for automatically exchanging information between systems. This is described in chapter 5.

## 2 Information model

This chapter describes the information model with all data elements. In this chapter the following information will be specified for each element:

Name of data element	
Description	The semantic description within the context of the cycling sector.
Multiplicity	<p>Whether the element is mandatory and how often it can occur, noted as [min...max]. Possible options are:</p> <ul style="list-style-type: none"> <li>• 0..1: The element is optional and can occur maximum once.</li> <li>• 0..n: The element is optional and can occur more than once (no maximum).</li> <li>• 0..x: The element is optional and can occur an x number of times.</li> <li>• 1..1: The element is mandatory and can occur once.</li> <li>• 1..n: The element is mandatory and can occur more than once (no maximum).</li> <li>• 1..x: The element is mandatory and can occur an x number of times.</li> </ul>
Datatype	<p>The datatype of the element:</p> <ul style="list-style-type: none"> <li>• Boolean: true/false (i.e. yes/no)</li> <li>• Date: date in accordance with ISO 8601 (yyyy-mm-dd).</li> <li>• Decimal: decimal number.</li> <li>• Integer: integer number.</li> <li>• String: combination of alphanumeric characters (numbers, letters, punctuation).</li> <li>• Container: a logical collection of (sub)elements. Does not contain a value it selves.</li> </ul>
Allowed values	<p>Describes the allowed values for this element, this could be:</p> <ul style="list-style-type: none"> <li>• a list with allowed values.</li> <li>• a code-/value-list</li> <li>• a reference to an external list (like an ISO standard).</li> </ul>
Example value	An example value
<i>Business rules</i>	
<ul style="list-style-type: none"> <li>• Other business rules that apply to the use of the element.</li> </ul>	
<i>CSV implementation</i>	
Field name	Field name for use in CSV implementation.
Implementation rule	If applicable for correctly filling the element, additional rules will be specified here.
<i>JSON implementation</i>	
Field name	Field name for use JSON implementation.
Implementation rule	If applicable for correctly filling the element, additional rules will be specified here.

The image below gives an overview of the elements that are defined in this standard.

<b>DST Informationmodel</b> <ul style="list-style-type: none"> <li>- Taal (Language)</li> <li>- Land (Country)</li> <li>- Valuta (Currency)</li> </ul>			
<b>Identifiers</b> <ul style="list-style-type: none"> <li>- Leverancier (supplier)</li> <li>- GLN</li> <li>- Artikelnummer (item number)</li> <li>- EAN/UPC</li> </ul>	<b>Orderinformation</b> <ul style="list-style-type: none"> <li>- Bestelcode (order code)</li> <li>- Besteleenheid (order amount)</li> <li>- Minimale bestelhoeveelheid (minimum order amount)</li> <li>- Status</li> <li>- Leverbaar vanaf (available from)</li> </ul>	<b>Bike specifics</b> <ul style="list-style-type: none"> <li>- Modeljaar (model year)</li> <li>- Wielmaat (wheel size)</li> </ul>	
<b>Product classification</b> <ul style="list-style-type: none"> <li>- Artikelgroep (product group)</li> <li>- Klantgroep (customer group)</li> <li>- Kernwoord (key word)</li> <li>- Plaatsbepaling (position)</li> <li>- Uniek Nummer Serie (unique number series)</li> <li>- Profielen (Profiles)</li> </ul>	<b>Contents product bundle</b> <ul style="list-style-type: none"> <li>- Omverpakkinginhouid (product bundle content)</li> <li>- Omverpakkingartikel (product bundle item)</li> </ul>	<b>Frame</b> <ul style="list-style-type: none"> <li>- Frametype (type of frame)</li> <li>- Framemaat (cm) (frame size (cm))</li> <li>- Framemaat leverancier (frame size supplier)</li> <li>- Framemateriaal (frame material)</li> </ul>	<b>Gears</b> <ul style="list-style-type: none"> <li>- Merk versnellingsysteem (gear brand)</li> <li>- Model versnellingsysteem (gear model)</li> <li>- Type versnellingsysteem (gear type)</li> <li>- Aantal versnellingen (number of gears)</li> </ul>
<b>Product information</b>		<b>Primary rear brake</b> <ul style="list-style-type: none"> <li>- Merk remsysteem (brake brand)</li> <li>- Model remsysteem (brake model)</li> <li>- Type remsysteem (brake type)</li> </ul>	<b>Secondary rear brake</b> <ul style="list-style-type: none"> <li>- Merk remsysteem (brake brand)</li> <li>- Model remsysteem (brake model)</li> <li>- Type remsysteem (brake type)</li> </ul>
<b>General product information</b> <ul style="list-style-type: none"> <li>- Merk (brand)</li> <li>- Model (model)</li> <li>- Modeljaar (model year)</li> <li>- Artikelnummer producent (item number producer)</li> <li>- Artikelpagina url (product page url)</li> <li>- POS-afbeelding (POS-image)</li> <li>- POS-afbeelding bestandsnaam (POS-image filename)</li> <li>- Video URL (video url)</li> <li>- High resolution afbeelding bestandsnaam (high res image file name)</li> <li>- High resolution afbeelding URL (high res image url)</li> <li>- Gebruikersdocumentatie (user documentation)</li> <li>- Technischdocumentatie (technical documentation)</li> </ul>	<b>Physical specifications</b> <ul style="list-style-type: none"> <li>- Kleur (colour)</li> <li>- Primaire basiskleur (primary base colour)</li> <li>- Secundaire basiskleur (secondary base colour)</li> <li>- Bruto gewicht (gross weight)</li> <li>- Netto gewicht (net weight)</li> <li>- Maat (size)</li> <li>- Verpakkingsafmeting (packaging size)</li> </ul>	<b>Front brake</b> <ul style="list-style-type: none"> <li>- Merk remsysteem (brake brand)</li> <li>- Model remsysteem (brake model)</li> <li>- Type remsysteem (brake type)</li> </ul>	<b>Seat post</b> <ul style="list-style-type: none"> <li>- Merk zadelpen (seat post brand)</li> <li>- Model zadelpen (seat post model)</li> </ul>
<b>Related products</b> <ul style="list-style-type: none"> <li>- Identiek artikel (identical product)</li> <li>- Vervangend artikel (replacing product)</li> <li>- Alternatief artikel (alternative product)</li> <li>- Benodigd artikel (needed product)</li> <li>- Accessoires (accessories)</li> </ul>	<b>Price information</b> <ul style="list-style-type: none"> <li>- Basisprijs dealer (dealer base price)</li> <li>- BTW (VAT)</li> <li>- Consumentenadviesprijs (recommended retail price)</li> <li>- Actie (special)</li> </ul>	<b>Tyres</b> <ul style="list-style-type: none"> <li>- Merk band (tyre brand)</li> <li>- Model band (tyre model)</li> </ul>	<b>Seat post clamp</b> <ul style="list-style-type: none"> <li>- Merk zadelpenklem (seat post clamp brand)</li> <li>- Model zadelpenklem (seat post clamp model)</li> </ul>
<b>Descriptions</b> <ul style="list-style-type: none"> <li>- Korte omschrijving (short description)</li> <li>- Lange omschrijving B2C (Long description B2C)</li> <li>- Kassabontekst (receipt text)</li> </ul>		<b>Fork</b> <ul style="list-style-type: none"> <li>- Merk voorvork (fork brand)</li> <li>- Model voorvork (fork model)</li> </ul>	<b>Stem</b> <ul style="list-style-type: none"> <li>- Merk stuurpen (stem brand)</li> <li>- Model stuurpen (stem model)</li> </ul>
		<b>Derailleur front</b> <ul style="list-style-type: none"> <li>- Merk derailleur voor (derailleur front brand)</li> <li>- Model derailleur voor (derailleur front model)</li> </ul>	<b>Bar tape or grips</b> <ul style="list-style-type: none"> <li>- Merk stuurlint of handvatten (bar tape or grips brand)</li> <li>- Model stuurlint of handvatten (bar tape or grips model)</li> </ul>
		<b>Derailleur rear</b> <ul style="list-style-type: none"> <li>- Merk derailleur achter (derailleur rear brand)</li> <li>- Model derailleur achter (derailleur rear model)</li> </ul>	<b>Saddle</b> <ul style="list-style-type: none"> <li>- Merk zadel (saddle brand)</li> <li>- Model zadel (saddle model)</li> </ul>
<b>Electric bikes specifics</b> <ul style="list-style-type: none"> <li>- Merk aandrijflijn (drivetrain brand)</li> <li>- Type aandrijflijn (drivetrain type)</li> <li>- Type sensor (sensor type)</li> <li>- Handvatbediening (handlebar throttle)</li> <li>- Walk-assist (walk assist)</li> <li>- Type elektrische fiets (electric bike type)</li> </ul>		<b>Crank set</b> <ul style="list-style-type: none"> <li>- Merk crankset (crank set brand)</li> <li>- Model crankset (crank set model)</li> </ul>	<b>Wheels</b> <ul style="list-style-type: none"> <li>- Merk wielen (wheels brand)</li> <li>- Model wielen (wheels model)</li> </ul>
<b>Battery/range</b> <ul style="list-style-type: none"> <li>- Accu specifieke velden (battery specifics)</li> <li>- Actieradius (range)</li> <li>- Accu opgenomen in basisprijs fiets (battery included in base price bike)</li> <li>- Meerprijs accu (extra cost battery)</li> <li>- Plaats accu (position battery)</li> <li>- Accu uitneembaar (battery removable)</li> <li>- Accu oplaadbaar in fiets (battery chargeable on bike)</li> </ul>	<b>Engine</b> <ul style="list-style-type: none"> <li>- Motor specifieke velden (engine specifics)</li> <li>- Plaats motor (position engine)</li> </ul>	<b>Shifters</b> <ul style="list-style-type: none"> <li>- Merk shifters (shifters brand)</li> <li>- Model shifters (shifters model)</li> </ul>	<b>Carrier</b> <ul style="list-style-type: none"> <li>- Merk dragers (carrier brand)</li> <li>- Model dragers (carrier model)</li> </ul>
<b>Battery specifics</b> <ul style="list-style-type: none"> <li>- Accuvermogen (battery capacity)</li> <li>- Accuvoltage (battery voltage)</li> <li>- Merk accu (battery brand)</li> <li>- Accu-ampere-uur (battery ampere-hour)</li> <li>- Accumodel (battery model)</li> <li>- Type accu (battery type)</li> <li>- Gewicht accu (battery weight)</li> <li>- Oplaadtijd normale lader (recharge time normal charger)</li> <li>- Oplaadtijd snellader (recharge time fast charger)</li> </ul>	<b>Display</b> <ul style="list-style-type: none"> <li>- Display specifieke velden (display specifics)</li> <li>- In basisprijs fiets (included in base price bike)</li> <li>- Meerprijs display (extra cost display)</li> </ul>	<b>Chain</b> <ul style="list-style-type: none"> <li>- Merk ketting (chain brand)</li> <li>- Model ketting (chain model)</li> </ul>	<b>Mudguards</b> <ul style="list-style-type: none"> <li>- Merk spatborden (mudguards brand)</li> <li>- Model spatborden (mudguards model)</li> </ul>
		<b>Cassette</b> <ul style="list-style-type: none"> <li>- Merk cassette (cassette brand)</li> <li>- Model cassette (cassette model)</li> </ul>	<b>Front light</b> <ul style="list-style-type: none"> <li>- Merk verlichting voor (front light brand)</li> <li>- Model verlichting voor (front light model)</li> </ul>
		<b>Shift cables</b> <ul style="list-style-type: none"> <li>- Merk schakelkabels (shift cables brand)</li> <li>- Model schakelkabels (shift cables model)</li> </ul>	<b>Rear light</b> <ul style="list-style-type: none"> <li>- Merk verlichting achter (rear light brand)</li> <li>- Model verlichting achter (rear light model)</li> </ul>
		<b>Headset</b> <ul style="list-style-type: none"> <li>- Merk balhoofd (headset brand)</li> <li>- Model balhoofd (headset model)</li> </ul>	<b>Kickstand</b> <ul style="list-style-type: none"> <li>- Merk standaard (kickstand brand)</li> <li>- Model standaard (kickstand model)</li> </ul>
		<b>Handlebar</b> <ul style="list-style-type: none"> <li>- Merk stuur (handlebar brand)</li> <li>- Model stuur (handlebar model)</li> </ul>	
<b>Display specifics</b> <ul style="list-style-type: none"> <li>- Merk display (display brand)</li> <li>- Type display (display type)</li> <li>- Type bediening (operation type)</li> <li>- Afneembaar display (display removable)</li> </ul>		<b>Engine specifics</b> <ul style="list-style-type: none"> <li>- Modelnaam motor (engine model name)</li> <li>- Type moter (engine type)</li> <li>- Merk motor (engine brand)</li> <li>- Wattage motor (engine wattage)</li> </ul>	

## 2.1 Language

Language	
Description	Specification of the language of the text elements of the product file.
Multiplicity	1..1
Datatype	String
Allowed values	Valid alpha-2-code ISO 639-1
Example value	nl
<i>Business rules</i>	
<b>Alpha-2-code ISO 639-1 is mandatory</b>	
<i>CSV implementation</i>	
Fieldname	taal
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	taal
Implementation rule	-

## 2.2 Country

Country	
Description	Specification of the country for which the product file is made.
Multiplicity	1..1
Datatype	String
Allowed values	Valid alpha-2-code ISO 3166-1
Example value	NL
<i>Business rules</i>	
<b>Alpha-2-code ISO 3166-1 is mandatory</b>	
<i>CSV implementation</i>	
Fieldname	land
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	land
Implementation rule	-

## 2.3 Currency

Currency	
Description	Specification of the currency the prices in the product file have been specified in.
Multiplicity	1..1
Datatype	String
Allowed values	Valid three letter code ISO 4217
Example value	EUR
<i>Business rules</i>	
<b>Three letter code ISO 4217 is mandatory</b>	
<i>CSV implementation</i>	
Fieldname	valuta
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	valuta
Implementation rule	-

## 2.4 Identifiers

Identifiers	
Description	Container for identifiers
Multiplicity	1..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	identifiers
Implementation rule	-

## 2.4.1 Supplier

Supplier	
Description	Unique identifier for the supplier of the product within the context of DST.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Parties
Example value	BATAVUS
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	leverancier
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	leverancier
Implementation rule	-

## 2.4.2 GLN

GLN	
Description	GLN (Global Location Number) code of the supplier of the product.
Multiplicity	0..1
Datatype	String
Allowed values	Valid GLN code
Example value	5790001398644
<i>Business rules</i>	
<b>Use GLN</b>	
<i>CSV implementation</i>	
Fieldname	gln
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	gln
Implementation rule	-

2.4.3 *Item number*

Item number	
Description	Unique identifier for the product within the context of the supplier.
Multiplicity	1..1
Datatype	String
Allowed values	
Example value	DH080050
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	artikelnummer
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	artikelnummer
Implementation rule	-

2.4.4 *EAN/UPC*

EAN/UPC	
Description	Officially registered barcode for the product.
Multiplicity	0..1
Datatype	String
Allowed values	Valid EAN/UPC
Example value	8713568158893
<i>Business rules</i>	
<b>If the product has an EAN/UPC this field is mandatory.</b>	
<i>CSV implementation</i>	
Fieldname	ean_upc
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	ean_upc
Implementation rule	-



## 2.5 Orderinformation

Orderinformation	
Description	Container for orderinformation
Multiplicity	1..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	bestelinformatie
Implementation rule	-

### 2.5.1 Order code

Order code	
Description	Code that the retailer needs to use to order from the supplier.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	8442
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	bestelcode
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	bestelcode
Implementation rule	-

## 2.5.2 Order amount

Order amount	
Description	The number the product has to be ordered by gradually. This is the number of consumer units that are in one distribution pack. The distribution pack is the box or packaging in which the products are supplied to the retailer.
Multiplicity	1..1
Datatype	Integer
Allowed values	
Example value	1
<i>Business rules</i>	
<b>The element order amount needs to contain a positive integer.</b>	
<i>CSV implementation</i>	
Fieldname	besteleenheid
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	besteleenheid
Implementation rule	-

## 2.5.3 Minimum order amount

Minimum order amount	
Description	Minimum number of consumer units that needs to be bought per order.
Multiplicity	1..1
Datatype	Integer
Allowed values	
Example value	5
<i>Business rules</i>	
<b>The element minimum order amount needs to contain a positive integer.</b>	
<i>CSV implementation</i>	
Fieldname	minimale_bestelhoeveelheid
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	minimale_bestelhoeveelheid
Implementation rule	-

## 2.5.4 Status

Status	
Description	Specification of the status of the product when the product file was supplied.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Statuses
Example value	COURANT
<i>Business rules</i>	
<b>The value in this element needs to be in the DST codelist.</b>	
<i>CSV implementation</i>	
Fieldname	status
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	status
Implementation rule	-

## 2.5.5 Available from

Available from	
Description	The date from which the product will be available (again).
Multiplicity	0..1
Datatype	Date
Allowed values	In accordance with ISO 8601
Example value	2014-05-20
<i>Business rules</i>	
<b>Use format YYYY-MM-DD</b>	
<i>CSV implementation</i>	
Fieldname	leverbaar_vanaf
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	leverbaar_vanaf
Implementation rule	-

## 2.5.6 Contents product bundle

Contents product bundle	
Description	If the product is a product bundle, this element can be used multiple times to specify which products and how many products are in the bundle.
Multiplicity	0..n
Datatype	Container
Allowed values	
Example value	n/a
Business rules	
<b>A bundle contains consumer units that are packed together for delivery to the retailer. This is not an extra packaging because of hygiene, shelf life or protection of products. A bundle can contain multiple and different kinds of products, for example a box that contains multiple consumer units, an assorted package or a display. The bundle has it's own item number: it can be identified as a unique product. The item number of the bundle is different from the item number(s) of the products in the bundle.</b>	
CSV implementation	
Fieldname	inhoud_overpakking
Implementation rule	Multiple number-of-product combinations need to be separated by a comma. To specify the content of the bundle, use this construction: "[number]*[item number],[number]*[item number],etc."
JSON implementation	
Fieldname	inhoud_omverpakking
Implementation rule	-

## 2.5.6.1 Product bundle content

Product bundle content	
Description	Number of products that have the same item number that are in the product bundle.
Multiplicity	0..1
Datatype	Integer
Allowed values	
Example value	30
Business rules	
<b>The element product bundle content needs to contain a positive integer.</b>	
CSV implementation	
Fieldname	n/a
Implementation rule	See the parent element 'Contents product bundle'.
JSON implementation	
Fieldname	omverpakkinginhoud
Implementation rule	-

## 2.5.6.2 Product bundle item

Product bundle item	
Description	Item number (within the context of the supplier) of the products that are in the product bundle.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	DH080172
<i>Business rules</i>	
<b>The item number in this element needs be in the product file as a single product. A product bundle can contain multiple products, but has it's own item number. The item number of the products in the bundle isn't the same as the item number(s) of the products that are in the bundle.</b>	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	See the parent element 'Contents product bundle'.
<i>JSON implementation</i>	
Fieldname	omverpakkingartikel
Implementation rule	-

## 2.6 Product classification

Product classification	
Description	Container for product classification
Multiplicity	1..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	artikelindeling
Implementation rule	-

## 2.6.1 Product group

Product group	
Description	Specification of the product group the product belongs to.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Product Groups
Example value	1B00
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	artikelgroep
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	artikelgroep
Implementation rule	-

## 2.6.2 Customer group

Customer group	
Description	Specification of the customer / target group for which the product is primarily intended.
Multiplicity	0..1
Datatype	String
Allowed values	DST Codelist Customer Groups
Example value	DAMES
<i>Business rules</i>	
<ul style="list-style-type: none"> <li>- Including a customer group is mandatory for bikes (product group 1XXX).</li> <li>- The value in this element needs to be in the codelist.</li> </ul>	
<i>CSV implementation</i>	
Fieldname	klantgroep
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	klantgroep
Implementation rule	-

## 2.6.3 Key word

Key word	
Description	Consistent description of the product as it is defined within the sector.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Key Words
Example value	Binnenband
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	kernwoord
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	kernwoord
Implementation rule	-

## 2.6.4 Position

Position	
Description	Position for consistent description of the product.
Multiplicity	0..1
Datatype	String
Allowed values	DST Codelist Position
Example value	R
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	plaatsbepaling
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	plaatsbepaling
Implementation rule	-

## 2.6.5 Unique number series

Unique number series	
Description	Unique number to identify a series of products.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	734645
Business rules	
<b>The unique number of the series as determined by the supplier has to be unique in the context of the supplier. In combination with the element 'supplier' it makes an unique number for the series.</b>	
CSV implementation	
Fieldname	unieknummerserie
Implementation rule	-
JSON implementation	
Fieldname	unieknummerserie
Implementation rule	-

## 2.6.6 Profiles

Profiles	
Description	Specification of the profile the product needs to be put in.
Multiplicity	0..n
Datatype	String
Allowed values	
Example value	1234
Business rules	
<b>This is an identifier that is given by the DST platform. The profile is linked to/owned by the supplier.</b>	
CSV implementation	
Fieldname	profielen
Implementation rule	This element has to be enclosed by quotation marks. Multiple profiles need to be separated by a comma.
JSON implementation	
Fieldname	kernwoord
Implementation rule	-



## 2.7 Product information

Product information	
Description	Container for product information.
Multiplicity	1..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	artikelinformatie
Implementation rule	-

### 2.7.1 General product information

General product information	
Description	Container for general product information
Multiplicity	1..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	algemene_artikelinformatie
Implementation rule	-

## 2.7.1.1 Brand

Brand	
Description	Specification of the products brand.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Brands
Example value	Batavus
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	merk
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	merk
Implementation rule	-

## 2.7.1.2 Model

Model	
Description	Specification of the products model. In this element only the modelname has to be filled out, no additional information.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	Fuente
<i>Business rules</i>	
<b>By example: for a Gazelle Fuente D57, only 'Fuente' needs to be filled out in this element.</b>	
<i>CSV implementation</i>	
Fieldname	model
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	model
Implementation rule	-

## 2.7.1.3 Model year

Model year	
Description	Specification of the year in which the product was added to the collection.
Multiplicity	0..1
Datatype	Integer
Allowed values	
Example value	2011
Business rules	
<ul style="list-style-type: none"> <li>- <b>If the product is still available in the next model year, this element should be adjusted.</b></li> <li>- <b>The element model year should contain 4 numeric characters that together specify a year.</b></li> <li>- <b>Use format YYYY.</b></li> </ul>	
CSV implementation	
Fieldname	modeljaar_algemeen
Implementation rule	-
JSON implementation	
Fieldname	modeljaar_algemeen
Implementation rule	-

## 2.7.1.4 Item number producer

Item number producer	
Description	Item number / type (within the context of the producer). This may be specified on the products packaging.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	XYZ123
Business rules	
CSV implementation	
Fieldname	artikelnummer_producent
Implementation rule	-
JSON implementation	
Fieldname	artikelnummer_producent
Implementation rule	-

## 2.7.1.5 Product page url

Product page url	
Description	A public url where more information about the product can be found.
Multiplicity	0..1
Datatype	url
Allowed values	
Example value	<a href="http://www.domein.nl/productpagina/XYZ123">http://www.domein.nl/productpagina/XYZ123</a>
<i>Business rules</i>	
<b>URL format as specified by IETF in RFC-3986</b>	
<i>CSV implementation</i>	
Fieldname	artikelpagina_url
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	artikelpagina_url
Implementation rule	-

## 2.7.1.6 POS-image

POS-image	
Description	Specification of a public url where a POS-image of the product can be found.
Multiplicity	0..n
Datatype	url
Allowed values	
Example value	<a href="http://www.domein.nl/afbeeldingen/IMG00062.jpg">http://www.domein.nl/afbeeldingen/IMG00062.jpg</a>
<i>Business rules</i>	
<b>URL format as specified by IETF in RFC-3986. If more than one url is applicable, the first url should be the primary image and the other url's should be the secondary images.</b>	
<i>CSV implementation</i>	
Fieldname	pos_afbeelding
Implementation rule	Multiple url's should be separated by a comma.
<i>JSON implementation</i>	
Fieldname	pos_afbeelding
Implementation rule	-

## 2.7.1.7 POS-image filename

POS-image filename	
Description	The filename of a POS-image of the product, including the format. With this a link can be made between the product file and a separate image.
Multiplicity	0..n
Datatype	String
Allowed values	
Example value	871050000365T-klein.jpg
<i>Business rules</i>	
<b>If more than one filename is filled out, the first filename should be the primary image and the other filenames should be the secondary images.</b>	
<i>CSV implementation</i>	
Fieldname	pos_afbeelding_bestandsnaam
Implementation rule	Multiple filenames should be separated by a comma.
<i>JSON implementation</i>	
Fieldname	pos_afbeelding_bestandsnaam
Implementation rule	-

## 2.7.1.8 Video url

Video url	
Description	A public url of a video related to the product (for example for product promotion).
Multiplicity	0..n
Datatype	url
Allowed values	
Example value	<a href="https://www.youtube.com/watch?v=Ucz5BMfZSu0">https://www.youtube.com/watch?v=Ucz5BMfZSu0</a>
<i>Business rules</i>	
<b>URL format as specified by IETF in RFC-3986.</b>	
<i>CSV implementation</i>	
Fieldname	video_url
Implementation rule	Multiple url's should be separated by a comma.
<i>JSON implementation</i>	
Fieldname	video_url
Implementation rule	-

## 2.7.1.9 High res image filename

High res image filename	
Description	The filename of an image of the product of high res quality. Intended for print.
Multiplicity	0..n
Datatype	String
Allowed values	
Example value	8710500000365T-hires.jpg
<i>Business rules</i>	
<b>If more than one filename is filled out, the first filename should be the primary image and the other filenames should be the secondary images.</b>	
<i>CSV implementation</i>	
Fieldname	high_resolution_afbeelding_bestandsnaam
Implementation rule	Multiple url's should be separated by a comma.
<i>JSON implementation</i>	
Fieldname	high_resolution_afbeelding_bestandsnaam
Implementation rule	-

## 2.7.1.10 High res image url

High res image url	
Description	A public url of a high res image of the product.
Multiplicity	0..n
Datatype	url
Allowed values	
Example value	<a href="http://www.domein.nl/afbeeldingen/IMG00063.jpg">http://www.domein.nl/afbeeldingen/IMG00063.jpg</a>
<i>Business rules</i>	
<b>URL format as specified by IETF in RFC-3986. If more than one filename is filled out, the first filename should be the primary image and the other filenames should be the secondary images.</b>	
<i>CSV implementation</i>	
Fieldname	high_resolution_afbeelding_url
Implementation rule	Multiple url's should be separated by a comma.
<i>JSON implementation</i>	
Fieldname	high_resolution_afbeelding_url
Implementation rule	-

## 2.7.1.11 User documentation

User documentation	
Description	Container for the specification of the documentation of the product, intended for the user (for example user manual, warranty. etc.).
Multiplicity	0..n
Datatype	Container
Allowed values	
Example value	n/a
Business rules	
<b>PDF files are preferred.</b>	
CSV implementation	
Fieldname	gebruikersdocumentatie
Implementation rule	Multiple user documentations should be separated by a comma. To specify the contents of this element the following construction should be used: "[type]*[url],[type]*[url],etc". The value for [type] needs to be in the DST Codelist Document Types.
JSON implementation	
Fieldname	gebruikersdocumentatie
Implementation rule	-

## 2.7.1.11.1 Type

Type	
Description	The type of documentation.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Document Types
Example value	Gebruikers handleiding
Business rules	
<b>The value in this element needs to be in the Codelist.</b>	
CSV implementation	
Fieldname	n/a
Implementation rule	See the parent element 'user documentation'.
JSON implementation	
Fieldname	type
Implementation rule	-

## 2.7.1.11.2 Url

Url	
Description	A public url where the documentation can be found.
Multiplicity	1..1
Datatype	url
Allowed values	
Example value	<a href="http://www.handleiding.nl/upload/usermanual2013.pdf">http://www.handleiding.nl/upload/usermanual2013.pdf</a>
<i>Business rules</i>	
<b>URL format as specified by IETF in RFC-3986.</b>	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	See the parent element 'user documentation'.
<i>JSON implementation</i>	
Fieldname	url
Implementation rule	-

## 2.7.1.12 Technical documentation

High res image url	
Description	Container for the specification of the documentation of the product, intended for the retailer (for example installation guide, service documents, assembly manual, warranty documents).
Multiplicity	0..n
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
<b>PDF files are preferred.</b>	
<i>CSV implementation</i>	
Fieldname	technischedocumentatie
Implementation rule	Multiple technical documentations should be separated by a comma. To specify the contents of this element the following construction should be used: "[type]*[url],[type]*[url],etc". The value for [type] needs to be in the DST Codelist Document Types.
<i>JSON implementation</i>	
Fieldname	technischedocumentatie
Implementation rule	-



## 2.7.1.12.1 Type

Type	
Description	The type of documentation.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Document Types
Example value	Technische tekening
<i>Business rules</i>	
<b>The value in this element needs to be in the Codelist.</b>	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	See the parent element 'technical documentation'.
<i>JSON implementation</i>	
Fieldname	type
Implementation rule	-

## 2.7.1.12.2 Url

Url	
Description	A public url where the documentation can be found.
Multiplicity	1..1
Datatype	url
Allowed values	
Example value	<a href="http://www.handleiding.nl/upload/servicemanual7TR56.pdf">http://www.handleiding.nl/upload/servicemanual7TR56.pdf</a>
<i>Business rules</i>	
<b>URL format as specified by IETF in RFC-3986.</b>	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	See the parent element 'technical documentation'.
<i>JSON implementation</i>	
Fieldname	url
Implementation rule	-

## 2.7.2 Physical specifications

Physical specification	
Description	Container for physical specifications.
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	fysieke_kenmerk
Implementation rule	-

## 2.7.2.1 Colour

Colour	
Description	Textual description of the colour of the product.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	Inktblauw met witte strepen
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	kleur
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	kleur
Implementation rule	-

## 2.7.2.2 Primary base colour

Primary base colour	
Description	Specification of the base colour of the product.
Multiplicity	0..1
Datatype	String
Allowed values	DST Codelist Base Colours
Example value	BLAUW
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	primaire_basiskleur
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	primaire_basiskleur
Implementation rule	-

## 2.7.2.3 Secondary base colour

Secondary base colour	
Description	Specification of the secondary base colour of the product.
Multiplicity	0..1
Datatype	String
Allowed values	DST Codelist Base Colours
Example value	WIT
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	secundaire_basiskleur
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	secundaire_basiskleur
Implementation rule	-

## 2.7.2.4 Gross weight

Gross weight	
Description	Weight of the product in kilogram, including retail packaging, excluding transport packaging.
Multiplicity	0..1
Datatype	Decimal
Allowed values	
Example value	12.00
<i>Business rules</i>	
<b>The element gross weight needs to contain a positive decimal with a point (.) as separator.</b>	
<i>CSV implementation</i>	
Fieldname	bruto_gewicht
Implementation rule	Always enclosed by quotation marks.
<i>JSON implementation</i>	
Fieldname	bruto_gewicht
Implementation rule	-

## 2.7.2.5 Net weight

Net weight	
Description	Weight of the product in kilograms, excluding any form of packaging.
Multiplicity	0..1
Datatype	Decimal
Allowed values	
Example value	10.05
<i>Business rules</i>	
<b>The element net weight needs to contain a positive decimal with a point (.) as separator.</b>	
<i>CSV implementation</i>	
Fieldname	netto_gewicht
Implementation rule	Always enclosed by quotation marks.
<i>JSON implementation</i>	
Fieldname	netto_gewicht
Implementation rule	-

## 2.7.2.6 Size

Size	
Description	Size of the product.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	XL
<i>Business rules</i>	
<ul style="list-style-type: none"> <li>- <b>Mandatory when the product is clothing (clothing size, shoe size, tyre size etc.).</b></li> <li>- <b>Pay attention! This isn't for frame or wheel size, they have a different element.</b></li> <li>- <b>In case of a tyre size, use the Codelist ETRTO.</b></li> <li>- <b>For now there's no uniform unit for this element. In case the unit isn't clear (for example for contents) put the unit after the value (for example 300 ml, 20 cm or 0.2 m).</b></li> </ul>	
<i>CSV implementation</i>	
Fieldname	maat
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	maat
Implementation rule	-

## 2.7.2.7 Packaging size

Packaging size	
Description	Container for the specification of the size of the retail packaging, excluding transport packaging.
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	verpakkingsafmeting
Implementation rule	-

## 2.7.2.7.1 Length packaging

Length packaging	
Description	Length in centimer (cm).
Multiplicity	0..1
Datatype	Decimal
Allowed values	
Example value	220
<i>Business rules</i>	
<b>Positive value, point as separator.</b>	
<i>CSV implementation</i>	
Fieldname	lengte_verpakking
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	lengte_verpakking
Implementation rule	-

## 2.7.2.7.2 Width packaging

Width packaging	
Description	Width in centimeter (cm).
Multiplicity	0..1
Datatype	Decimal
Allowed values	
Example value	110
<i>Business rules</i>	
<b>Positive value, point as separator.</b>	
<i>CSV implementation</i>	
Fieldname	breedte_verpakking
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	breedte_verpakking
Implementation rule	-

## 2.7.2.7.3 Hight packaging

Hight packaging	
Description	Hight in centimeter (cm)
Multiplicity	0..1
Datatype	Decimal
Allowed values	
Example value	20
<i>Business rules</i>	
<b>Positive value, point as separator.</b>	
<i>CSV implementation</i>	
Fieldname	hoogte_verpakking
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	hoogte_verpakking
Implementation rule	-

## 2.7.2.7.4 Diameter packaging

Diameter packaging	
Description	Diameter in centimeter (cm).
Multiplicity	0..1
Datatype	Decimal
Allowed values	
Example value	20
<i>Business rules</i>	
<b>Positive value, point as separator.</b>	
<i>CSV implementation</i>	
Fieldname	diameter_verpakking
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	diameter_verpakking
Implementation rule	-

## 2.7.3 Price information

Price information	
Description	Container with price information
Multiplicity	1..1
Datatype	Container
Allowed values	
Example value	n/a
Business rules	
-	
CSV implementation	
Fieldname	n/a
Implementation rule	-
JSON implementation	
Fieldname	prijsinformatie
Implementation rule	-

## 2.7.3.1 Dealer base price

Dealer base price	
Description	Standard cost price per product from the supplier for the retailer in the specified currency (excluding VAT).
Multiplicity	1..1
Datatype	Decimal
Allowed values	
Example value	641.00
Business rules	
<ul style="list-style-type: none"> <li>- <b>When the product is a bundle: specify the price of the bundle!</b></li> <li>- <b>The element needs to contain a positive decimal with a point (.) as separator and 2 decimals after the point.</b></li> <li>- <b>The element shouldn't contain a currency specification.</b></li> </ul>	
CSV implementation	
Fieldname	basisprijs_dealer
Implementation rule	Always enclosed by quotation marks. No currency specification.
JSON implementation	
Fieldname	basisprijs_dealer
Implementation rule	-



## 2.7.3.2 VAT

VAT	
Description	Specification of the VAT rate on the product.
Multiplicity	0..1
Datatype	String
Allowed values	DST Codelist VAT
Example value	H
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	btw
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	btw
Implementation rule	-

## 2.7.3.3 Recommended retail price

Recommended retail price	
Description	Standard recommended retail price per product in the specified currency (including VAT).
Multiplicity	0..1
Datatype	Decimal
Allowed values	
Example value	699.95
<i>Business rules</i>	
<ul style="list-style-type: none"> <li>- <b>When the product is a bundle: specify the price of the bundle!</b></li> <li>- <b>In case the bundle isn't sold to consumers, we advise not to add a recommended retail price.</b></li> <li>- <b>The element needs to contain a positive decimal with a point (.) as separator and 2 decimals after the point.</b></li> <li>- <b>The element shouldn't contain a currency specification.</b></li> </ul>	
<i>CSV implementation</i>	
Fieldname	consumentenadviesprijs
Implementation rule	Always enclosed by quotation marks. No currency specification.
<i>JSON implementation</i>	
Fieldname	consumentenadviesprijs
Implementation rule	-

## 2.7.3.4 Special

Special	
Description	Container to specify a (national) special during a specified period.
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	actie
Implementation rule	-

## 2.7.3.4.1 Special recommended retail price

Special recommended retail price	
Description	The recommended retail price during the period the special is on in the specified currency (including VAT).
Multiplicity	0..1
Datatype	Decimal
Allowed values	
Example value	99.95
<i>Business rules</i>	
<ul style="list-style-type: none"> <li>- <b>When the product is a bundle: specify the price of the bundle!</b></li> <li>- <b>Positive decimal, 2 decimals, point as separator, no currency specified.</b></li> </ul>	
<i>CSV implementation</i>	
Fieldname	actie_consumentenadviesprijs
Implementation rule	Always enclosed by quotation marks. No currency specified.
<i>JSON implementation</i>	
Fieldname	actie_consumentenadviesprijs
Implementation rule	-

## 2.7.3.4.2 Special gross cost price

Special	
Description	The gross cost price during the period the special is on in the specified currency ((including VAT).
Multiplicity	0..1
Datatype	Decimal
Allowed values	
Example value	89.95
<i>Business rules</i>	
<b>Positive decimal, 2 decimals, point as separator, no currency specified.</b>	
<i>CSV implementation</i>	
Fieldname	actie_bruto_inkoopprijs
Implementation rule	Always enclosed by quotation marks. No currency specified.
<i>JSON implementation</i>	
Fieldname	actie_bruto_inkoopprijs
Implementation rule	-

## 2.7.3.4.3 Special date from

Special date from	
Description	The date the special starts.
Multiplicity	0..1
Datatype	Date
Allowed values	Conform ISO 8601
Example value	2014-05-26
<i>Business rules</i>	
<b>Use format YYYY-MM-DD.</b>	
<i>CSV implementation</i>	
Fieldname	actie_datum_vanaf
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	actie_datum_vanaf
Implementation rule	-

## 2.7.3.4.4 Special date to

Special date from	
Description	The date until which the special is on.
Multiplicity	0..1
Datatype	Date
Allowed values	Conform ISO 8601
Example value	2014-05-31
<i>Business rules</i>	
<b>Use format YYYY-MM-DD.</b>	
<i>CSV implementation</i>	
Fieldname	actie_datum_tot_en_met
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	actie_datum_tot_en_met
Implementation rule	-

## 2.7.4 Descriptions

Descriptions	
Description	Container for product descriptions.
Multiplicity	1..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	omschrijvingen
Implementation rule	-

## 2.7.4.1 Short description

Short description	
Description	Short description of the product.
Multiplicity	1..1
Datatype	String
Allowed values	
Example value	Batavus Alamo dames 57 V-brake LX24
Business rules	
<b>This element can contain a maximum of 50 characters.</b>	
CSV implementation	
Fieldname	korte_omschrijving
Implementation rule	-
JSON implementation	
Fieldname	korte_omschrijving
Implementation rule	-

## 2.7.4.2 Long description

Long description	
Description	Commercial description of the product intended for consumers. For example for use in an online shop.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	De Gazelle Orange is de nieuwste stadshybride uit de....
Business rules	
<b>There's no maximum number of characters for this element.</b>	
CSV implementation	
Fieldname	lange_omschrijving
Implementation rule	-
JSON implementation	
Fieldname	lange_omschrijving
Implementation rule	-

## 2.7.4.3 Receipt text

Receipt text	
Description	Composition of specifications of the product (abbreviated or not), that describe the product in such a way it can be recognised by the consumer when it's on the receipt.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	Orange DLI53 T8 RB
<i>Business rules</i>	
<b>This element can contain a maximum of 26 character (2 lines on a standard receipt).</b>	
<i>CSV implementation</i>	
Fieldname	kassabontekst
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	kassabontekst
Implementation rule	-

## 2.7.5 Related products

Related products	
Description	Container for referral to related products.
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	gerelateerde_artikelen
Implementation rule	-

## 2.7.5.1 Identical product

Identical product	
Description	Referral to a product that is an identical type of product with a different identification.
Multiplicity	0..n
Datatype	String
Allowed values	
Example value	8713568158894
<i>Business rules</i>	
<b>EAN or suppliers item number (EAN is preferred).</b>	
<i>CSV implementation</i>	
Fieldname	identiek_artikel
Implementation rule	Separate multiple values by a comma.
<i>JSON implementation</i>	
Fieldname	identiek artikel
Implementation rule	-

## 2.7.5.2 Replacing product

Replacing product	
Description	Referral to a product that can replace the product in case it's no longer available.
Multiplicity	0..n
Datatype	String
Allowed values	
Example value	8713568158895
<i>Business rules</i>	
<b>EAN or suppliers item number (EAN is preferred).</b>	
<i>CSV implementation</i>	
Fieldname	vervangend_artikel
Implementation rule	Separate multiple values by a comma.
<i>JSON implementation</i>	
Fieldname	vervangend_artikel
Implementation rule	-

2.7.5.3 *Alternative product*

Alternative product	
Description	Referral to an alternative product, in case the product is no longer available, or as a cheaper/more expensive alternative.
Multiplicity	0..n
Datatype	String
Allowed values	
Example value	8713568158895
<i>Business rules</i>	
<b>EAN or suppliers item number (EAN is preferred).</b>	
<i>CSV implementation</i>	
Fieldname	alternatief_artikel
Implementation rule	Separate multiple values by a comma.
<i>JSON implementation</i>	
Fieldname	alternatief_artikel
Implementation rule	-

2.7.5.4 *Needed product*

Needed product	
Description	Referral to another product supplementary to the product, in case it is needed/mandatory for the product to operate properly.
Multiplicity	0..n
Datatype	String
Allowed values	
Example value	8713568158897
<i>Business rules</i>	
<b>EAN or suppliers item number (EAN is preferred).</b>	
<i>CSV implementation</i>	
Fieldname	benodigd_artikel
Implementation rule	Separate multiple values by a comma.
<i>JSON implementation</i>	
Fieldname	benodigd_artikel
Implementation rule	-



## 2.7.5.5 Accessories

<b>Accessories</b>	
Description	Referral to another product to complement, in case this is extends the functions of the product.
Multiplicity	0..n
Datatype	String
Allowed values	
Example value	8713568158898
<i>Business rules</i>	
<b>EAN or suppliers item number (EAN is preferred).</b>	
<i>CSV implementation</i>	
Fieldname	accessoires
Implementation rule	Separate multiple values by a comma.
<i>JSON implementation</i>	
Fieldname	accessoires
Implementation rule	-

## 2.8 Bike specifics

Bike specifics	
Description	Container with specifics for a bike or bike part.
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
Business rules	
-	
CSV implementation	
Fieldname	n/a
Implementation rule	-
JSON implementation	
Fieldname	fiets_specifieke_velden
Implementation rule	-

### 2.8.1 Model year

Model year	
Description	Specification of the year the product was added to the collection.
Multiplicity	0..1
Datatype	Integer
Allowed values	
Example value	2011
Business rules	
<ul style="list-style-type: none"> <li>- <b>If the product is still available in the next model year, this element should be adjusted.</b></li> <li>- <b>The element model year should contain 4 numeric characters that together specify a year.</b></li> <li>- <b>Use format YYYY.</b></li> </ul>	
CSV implementation	
Fieldname	modeljaar
Implementation rule	-
JSON implementation	
Fieldname	modeljaar
Implementation rule	-

2.8.2 *Wheel size*

Wheel size	
Description	Size of the drive wheel in inches.
Multiplicity	0..1
Datatype	Decimal
Allowed values	
Example value	22.50
<i>Business rules</i>	
<b>This element needs to contain a positive decimal with a point (.) as separator and 2 decimals after the point.</b>	
<i>CSV implementation</i>	
Fieldname	wielmaat
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	wielmaat
Implementation rule	-

2.8.3 *Frame*

Frame	
Description	Container with information about the frame.
Multiplicity	1..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	frame
Implementation rule	-

2.8.3.1 *Frame type*

Frame type	
Description	Description of the type of frame. This name needs to be recognisable within the sector.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Frame Types
Example value	DAMESMONO
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	frametype
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	frametype
Implementation rule	-

2.8.3.2 *Frame size*

Frame size	
Description	Size of the frame in centimeters. The frame size is the distance between the middel of the bottom bracket and the top tube.
Multiplicity	0..1
Datatype	Integer
Allowed values	
Example value	57
<i>Business rules</i>	
<ul style="list-style-type: none"> <li>- <b>Mandatory if frame size supplier isn't given.</b></li> <li>- <b>This element needs to contain a positive integer.</b></li> </ul>	
<i>CSV implementation</i>	
Fieldname	framemaat
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	framemaat
Implementation rule	-

2.8.3.3 *Frame size supplier*

Frame size supplier	
Description	Size of the frame according to the suppliers semantics.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	26 inch
<i>Business rules</i>	
<b>Mandatory if frame size isn't given.</b>	
<i>CSV implementation</i>	
Fieldname	framemaat_leverancier
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	framemaat_leverancier
Implementation rule	-

2.8.3.4 *Frame material*

Frame material	
Description	Specification of the material the frame is made of.
Multiplicity	0..1
Datatype	String
Allowed values	DST Codelist Frame Materials
Example value	ALUMINIUM
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	framemateriaal
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	framemateriaal
Implementation rule	-

## 2.8.4 Gears

Gears	
Description	Container with information about the gears.
Multiplicity	1..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	versnellingsysteem
Implementation rule	-

## 2.8.4.1 Gear brand

Gear brand	
Description	Specification of the brand of the gears. Usually the name of the producer.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Brands
Example value	SHIMANO
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	merk_versnellingsysteem
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	merk_versnellingsysteem
Implementation rule	-

## 2.8.4.2 Gear model

Gear model	
Description	Specification of the model/version/type of the gears.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	Nexus 7 speed
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	model_versnellingsysteem
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	model_versnellingsysteem
Implementation rule	-

## 2.8.4.3 Gear type

Frame material	
Description	Specification of the gears mechanics.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Gear Types
Example value	NAAF
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	type_versnellingsysteem
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	type_versnellingsysteem
Implementation rule	-

## 2.8.4.4 Number of gears

Number of gears	
Description	Specification of the number of gears.
Multiplicity	1..1
Datatype	Integer
Allowed values	
Example value	24
<i>Business rules</i>	
<ul style="list-style-type: none"> <li>- In case of a continuous variable transmission: use value 999.</li> <li>- This element needs to contain a positive integer.</li> </ul>	
<i>CSV implementation</i>	
Fieldname	aantal_versnellingen
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	aantal_versnellingen
Implementation rule	-

## 2.8.5 Primary rear brake

Primary rear brake	
Description	Container with information about the primary rear brake system.
Multiplicity	1..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	primair_remsysteem_achter
Implementation rule	-



## 2.8.5.1 Brake brand

Brake brand	
Description	Specification of the brand of the brake system. Usually the name of the producer.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Brands
Example value	SHIMANO
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	merk_primaire_remsysteem_achter
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	merk_primaire_remsysteem_achter
Implementation rule	-

## 2.8.5.2 Brake model

Brake model	
Description	Specification of the model/version/type of the brake system.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	Hydr. Schijfremmen 180/160
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	model_primaire_remsysteem_achter
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	model_primaire_remsysteem_achter
Implementation rule	-

## 2.8.5.3 Brake type

Brake type	
Description	Specification of the brake mechanics that are used for this brake system.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Brake Types
Example value	SCHIJFHYDR
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	model_primaire_remsysteem_achter
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	type_primaire_remsysteem_achter
Implementation rule	-

## 2.8.6 Secondary rear brake

Secondary rear brake	
Description	Container with information about the secondary rear brake system.
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	secondaire_remsysteem_achter
Implementation rule	-

## 2.8.6.1 Brake brand

Brake brand	
Description	Specification of the brand of the brake system. Usually the name of the producer.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Brands
Example value	SHIMANO
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	merk_secundaire_remsysteem_achter
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	merk_secundaire_remsysteem_achter
Implementation rule	-

## 2.8.6.2 Brake model

Brake model	
Description	Specification of the model/version/type of the brake system.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	Hydr. Schijfremmen 180/160
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	merk_secundaire_remsysteem_achter
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	model_secundaire_remsysteem_achter
Implementation rule	-

## 2.8.6.3 Brake type

Brake type	
Description	Specification of the brake mechanics that are used for this brake system.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Brake Types
Example value	SCHIJFHYDR
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	type_secondaire_remsysteem_achter
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	type_secondaire_remsysteem_achter
Implementation rule	-

## 2.8.7 Front brake

Front brake	
Description	Container with information about the front brake system.
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	remsysteem_voor
Implementation rule	-

## 2.8.7.1 Brake brand

Brake brand	
Description	Specification of the brand of the brake system. Usually the name of the producer.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Brands
Example value	SHIMANO
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	merk_remsysteem_voor
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	merk_remsysteem_voor
Implementation rule	-

## 2.8.7.2 Brake model

Brake model	
Description	Specification of the model/version/type of the brake system.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	Hydr. Schijfremmen 180/160
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	model_remsysteem_voor
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	model_remsysteem_voor
Implementation rule	-

## 2.8.7.3 Brake type

Brake type	
Description	Specification of the brake mechanics that are used for this brake system.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Brake Types
Example value	SCHIJFHYDR
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	type_remsysteem_voor
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	type_remsysteem_voor
Implementation rule	-

## 2.8.8 Tyres

Tyres	
Description	Container with information about the tyres.
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	banden
Implementation rule	-

## 2.8.8.1 Tyre brand

Tyre brand	
Description	Specification of the brand of the tyre. Usually the name of the producer.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Brands
Example value	SCHWALBE
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	merk_band
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	merk_band
Implementation rule	-

## 2.8.8.2 Tyre model

Tyre model	
Description	Specification of the model/version/type of the tyre.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	One, Kevlar, 25x622
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	model_band
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	model_band
Implementation rule	-

## 2.8.9 Fork

Fork	
Description	Container with information about the fork.
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	voork
Implementation rule	-

## 2.8.9.1 Fork brand

Fork brand	
Description	Specification of the brand of the fork. Usually the name of the producer.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Brands
Example value	CUBE
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	merk_voork
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	merk_voork
Implementation rule	-



## 2.8.9.2 Fork model

Fork model	
Description	Specification of the model/version/type of the fork.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	CUBE CSL Evo C:62 Full Carbon Technology, 1 1/8 - 1 1/4" Tapered
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	model_voorvork
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	model_voorvork
Implementation rule	-

## 2.8.10 Derailleur front

Derailleur front	
Description	Container with information about the front derailleur.
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	derailleur_voor
Implementation rule	-

2.8.10.1 *Derailleur front brand*

Derailleur front brand	
Description	Specification of the brand of the front derailleur system. Usually the name of the producer.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Brands
Example value	SHIMANO
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	merk_derailleur_voor
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	merk_derailleur_voor
Implementation rule	-

2.8.10.2 *Derailleur front model*

Derailleur front model	
Description	Specification of the model/version/type of the front derailleur.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	Ultegra Di2 FD-6870, Braze-On
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	model_derailleur_voor
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	model_derailleur_voor
Implementation rule	-

2.8.11 *Derailleur rear*

Derailleur rear	
Description	Container with information about the rear derailleur.
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	derailleur_achter
Implementation rule	-

2.8.11.1 *Derailleur rear brand*

Derailleur rear brand	
Description	Specification of the brand of the rear derailleur. Usually the name of the producer.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Brands
Example value	SHIMANO
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	merk_derailleur_achter
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	merk_derailleur_achter
Implementation rule	-

2.8.11.2 *Derailleur rear model*

<b>Derailleur rear model</b>	
Description	Specification of the model/version/type of the rear derailleur.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	Ultegra Di2 RD-6870, 11-Speed
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	model_derailleur_achter
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	model_derailleur_achter
Implementation rule	-

2.8.12 *Crank set*

<b>Crank set</b>	
Description	Container with information about the crank set.
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	crankset
Implementation rule	-

## 2.8.12.1 Crank set brand

Crank set brand	
Description	Specification of the brand of the crank set. Usually the name of the producer.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Brands
Example value	SHIMANO
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	merk_crankset
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	merk crankset
Implementation rule	-

## 2.8.12.2 Crank set model

Crank set model	
Description	Specification of the model/version/type of the crank set.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	Ultegra FC-6800, Hollowtech II, 50x34T
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	model_crankset
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	model_crankset
Implementation rule	-

## 2.8.13 Shifters

Crank set	
Description	Container with information about the shifters
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
Business rules	
-	
CSV implementation	
Fieldname	n/a
Implementation rule	-
JSON implementation	
Fieldname	shifters
Implementation rule	-

## 2.8.13.1 Shifters brand

Crank set brand	
Description	Specification of the brand of the shifters. Usually the name of the producer.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Brands
Example value	SHIMANO
Business rules	
<b>The value in this element needs to be in the codelist.</b>	
CSV implementation	
Fieldname	merk_shifters
Implementation rule	-
JSON implementation	
Fieldname	merk_shifters
Implementation rule	-

## 2.8.13.2 Shifters model

Shifters model	
Description	Specification of the model/version/type of the shifters.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	Ultegra Di2 ST-6870
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	model_shifters
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	model_shifters
Implementation rule	-

## 2.8.14 Chain

Chain	
Description	Container with information about the chain.
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	ketting
Implementation rule	-

## 2.8.14.1 Chain brand

Crank set brand	
Description	Specification of the brand of the chain. Usually the name of the producer.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Brands
Example value	SHIMANO
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	merk_ketting
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	merk_ketting
Implementation rule	-

## 2.8.14.2 Chain model

Chain model	
Description	Specification of the model/version/type of the chain
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	CN-HG700-11
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	model_ketting
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	model_ketting
Implementation rule	-



## 2.8.15 Cassette

Cassette	
Description	Container with information about the cassette.
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	cassette
Implementation rule	-

## 2.8.15.1 Cassette brand

Cassette brand	
Description	Specification of the brand of the cassette. Usually the name of the producer.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Brands
Example value	SHIMANO
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	merk_cassette
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	mark_cassette
Implementation rule	-

2.8.15.2 *Cassette model*

<b>Cassette model</b>	
Description	Specification of the model/version/type of the cassette.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	Ultegra CS-6800, 11-28
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	model_cassette
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	model_cassette
Implementation rule	-

2.8.16 *Shift cables*

<b>Shift cables</b>	
Description	Container with information about the shift cables
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	schakelkabels
Implementation rule	-

## 2.8.16.1 Shift cables brand

Shift cables brand	
Description	Specification of the brand of the shift cables. Usually the name of the producer.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Brands
Example value	SHIMANO
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	merk_schakelkabels
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	merk_schakelkabels
Implementation rule	-

## 2.8.16.2 Shift cables model

Shift cables model	
Description	Specification of the model/version/type of the shift cables.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	Regulier
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	model_schakelkabels
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	model_schakelkabels
Implementation rule	-

## 2.8.17 Headset

Headset	
Description	Container with information about the headset.
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	balhoofd
Implementation rule	-

## 2.8.17.1 Headset brand

Crank set brand	
Description	Specification of the brand of the headset. Usually the name of the producer.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Brands
Example value	FSA
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	merk_balhoofd
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	merk_balhoofd
Implementation rule	-

## 2.8.17.2 Headset model

Headset model	
Description	Specification of the model/version/type of the headset.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	Orbit C33 ACB I-t Integrated, Top 1 1/8", Bottom 1 1/4"
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	model_balhoofd
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	model_balhoofd
Implementation rule	-

## 2.8.18 Handlebar

Handlebar	
Description	Container with information about the handlebar.
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	stuur
Implementation rule	-

## 2.8.18.1 Handle bar brand

Handle bar brand	
Description	Specification of the brand of the handle bar. Usually the name of the producer.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Brands
Example value	CUBE
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	merk_stuur
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	merk_stuur
Implementation rule	-

## 2.8.18.2 Handle bar model

Handle bar model	
Description	Specification of the model/version/type of the handle bar.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	Wing Race Bar Carbon
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	model_stuur
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	model stuur
Implementation rule	-

## 2.8.19 Stem

Stem	
Description	Container with information about the stem.
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	stuurpen
Implementation rule	-

## 2.8.19.1 Stem brand

Crank set brand	
Description	Specification of the brand of the stem. Usually the name of the producer.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Brands
Example value	SYNTACE
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	merk_stuurpen
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	merk_stuurpen
Implementation rule	-

## 2.8.19.2 Stem model

Crank set model	
Description	Specification of the model/version/type of the stem.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	F149, 31.8mm
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	model_stem
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	model_stem
Implementation rule	-

## 2.8.20 Bar tape or grips

Bar tape or grips	
Description	Container with information about the bar tape or grips.
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	sturlint_of_handvatten
Implementation rule	-



## 2.8.20.1 Bar tape or grips brand

Bar tape or grips brand	
Description	Specification of the brand of the bar tape or grips. Usually the name of the producer.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Brands
Example value	CUBE
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	merk_stuurlint_of_handvatten
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	merk_stuurlint_of_handvatten
Implementation rule	-

## 2.8.20.2 Bar tape of grips model

Bar tape or grips model	
Description	Specification of the model/version/type of the bar tape or grips.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	Grip Control
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	model_stuurlint_of_handvatten
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	model_stuurlint_of_handvatten
Implementation rule	-

## 2.8.21 Seat post

Crank set	
Description	Container with information about the seat post.
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
Business rules	
-	
CSV implementation	
Fieldname	n/a
Implementation rule	-
JSON implementation	
Fieldname	zadelpen
Implementation rule	-

## 2.8.21.1 Seat post brand

Seat post brand	
Description	Specification of the brand of the seat post. Usually the name of the producer.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Brands
Example value	CUBE
Business rules	
<b>The value in this element needs to be in the codelist.</b>	
CSV implementation	
Fieldname	merk_zadelpen
Implementation rule	-
JSON implementation	
Fieldname	merk_zadelpen
Implementation rule	-

2.8.21.2 *Seat post model*

<b>Seat post model</b>	
Description	Specification of the model/version/type of the seat post.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	Performance Motion Post, 27.2mm
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	model_zadelpen
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	model_zadelpen
Implementation rule	-

2.8.22 *Seat post clamp*

<b>Crank set</b>	
Description	Container with information about the seat post clamp.
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	zadelpenklemp
Implementation rule	-

## 2.8.22.1 Seat post clamp brand

Seat post clamp brand	
Description	Specification of the brand of the seat post clamp. Usually the name of the producer.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Brands
Example value	CUBE
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	merk_zadelpenklem
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	merk_zadelpenklem
Implementation rule	-

## 2.8.22.2 Seat post clamp model

Crank set model	
Description	Specification of the model/version/type of the seat post clamp.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	Screwlock, 31.8mm
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	model_zadelpenklem
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	model_zadelpenklem
Implementation rule	-

## 2.8.23 Saddle

Saddle	
Description	Container with information about the saddle.
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	zadel
Implementation rule	-

## 2.8.23.1 Saddle brand

Saddle brand	
Description	Specification of the brand of the saddle brand. Usually the name of the producer.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Brands
Example value	SELLE ITALIA
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	merk_zadel
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	merk_zadel
Implementation rule	-

## 2.8.23.2 Saddle model

Saddle model	
Description	Specification of the model/version/type of the saddle.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	X1 Road
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	model_zadel
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	model_zadel
Implementation rule	-

## 2.8.24 Wheels

Wheels	
Description	Container with information about the wheels.
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	wielen
Implementation rule	-

2.8.24.1 *Wheels brand*

Wheels brand	
Description	Specification of the brand of the wheels. Usually the name of the producer.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Brands
Example value	FULCRUM
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	merk_wielen
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	merk_wielen
Implementation rule	-

2.8.24.2 *Wheels model*

Wheels model	
Description	Specification of the model/version/type of the wheels
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	Racing 55
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	model_wielen
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	model_wielen
Implementation rule	-

## 2.8.25 Carrier

Carrier	
Description	Container with information about the carrier.
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	dragers
Implementation rule	-

## 2.8.25.1 Carrier brand

Carrier brand	
Description	Specification of the brand of the carrier. Usually the name of the producer.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Brands
Example value	STANDWELL
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	merk_dragers
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	merk_dragers
Implementation rule	-



## 2.8.25.2 Carrier model

Carrier model	
Description	Specification of the model/version/type of the carrier.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	Bag Carrier
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	model_dragers
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	model_dragers
Implementation rule	-

## 2.8.26 Mudguards

Mudguards	
Description	Container with information about the mudguards.
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	spatborden
Implementation rule	-

2.8.26.1 *Mudguards brand*

<b>Mudguards brand</b>	
Description	Specification of the brand of the mudguards. Usually the name of the producer.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Brands
Example value	CUBE
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	merk_spatborden
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	merk_spatborden
Implementation rule	-

2.8.25.2 *Mudguards model*

<b>Mudguards model</b>	
Description	Specification of the model/version/type of the mudguards.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	CUBEstand Pro, adjustable
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	model_spatborden
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	model_spatborden
Implementation rule	-

2.8.27 *Front light*

Front lights	
Description	Container with information about the front light.
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
Business rules	
-	
CSV implementation	
Fieldname	n/a
Implementation rule	-
JSON implementation	
Fieldname	verlichting_voor
Implementation rule	-

2.8.27.1 *Front light brand*

Front light brand	
Description	Specification of the brand of the front light. Usually the name of the producer.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Brands
Example value	BUSCH & MÜLLER
Business rules	
<b>The value in this element needs to be in the codelist.</b>	
CSV implementation	
Fieldname	merk_verlichting_voor
Implementation rule	-
JSON implementation	
Fieldname	merk_verlichting_voor
Implementation rule	-

2.8.27.2 *Front light model*

Front light model	
Description	Specification of the model/version/type of the front light.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	Lyt 1781
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	model_verlichting_voor
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	model_verlichting_voor
Implementation rule	-

2.8.28 *Rear light*

Rear lights	
Description	Container with information about the rear light.
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	verlichting_achter
Implementation rule	-

## 2.8.28.1 Rear light brand

Rear light brand	
Description	Specification of the brand of the rear light. Usually the name of the producer.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Brands
Example value	BUSCH & MÜLLER
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	merk_verlichting_achter
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	merk_verlichting_achter
Implementation rule	-

## 2.8.28.2 Rear light model

Rear light model	
Description	Specification of the model/version/type of the rear light.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	Toplight Flat Plus
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	model_verlichting_achter
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	model_verlichting_achter
Implementation rule	-

## 2.8.29 Kickstand

Kickstand	
Description	Container with information about the kickstand.
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	standaard
Implementation rule	-

## 2.8.29.1 Kickstand brand

Kickstand brand	
Description	Specification of the brand of the kickstand. Usually the name of the producer.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Brands
Example value	CUBE
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	merk_standraad
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	merk_standraad
Implementation rule	-

## 2.8.29.2 Kickstand model

Kickstand model	
Description	Specification of the model/version/type of the kickstand.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	CUBEstand Pro, Adjustable
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	model_standaard
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	model_standaard
Implementation rule	-

## 2.9 Electric bike specifics

Electric bike specifics	
Description	Container with elements specific for an electric bike or an electric bike part.
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	elektrische_fiets_specifieke_velden
Implementation rule	-

## 2.9.1 Drivetrain brand

Drivetrain brand	
Description	Specification of the brand of the drivetrain. Usually the name of the producer.
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Ebike Systems
Example value	BOS
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	merk_aandrijflijn
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	merk_aandrijflijn
Implementation rule	-

## 2.9.2 Drivetrain type

Drivetrain type	
Description	Specification of the type of drivetrain that is used in the bike.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	Ketting
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	type_aandrijflijn
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	type_aandrijflijn
Implementation rule	-



## 2.9.3 Sensor type

Sensor type	
Description	Type of sensor that is used to drive the engine.
Multiplicity	0..n
Datatype	String
Allowed values	DST Codelist Type Sensor
Example value	ROTATIE
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	type_sensor
Implementation rule	Separate multiple values by a comma.
<i>JSON implementation</i>	
Fieldname	type_sensor
Implementation rule	-

## 2.9.4 Handlebar throttle

Handlebar throttle	
Description	Indicaton wether the bike has a handlebar throttle.
Multiplicity	0..1
Datatype	Boolean
Allowed values	TRUE FALSE
Example value	TRUE
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	handvatbediening
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	handvatbediening
Implementation rule	-

## 2.9.5 Walk assist

Walk assist	
Description	Indication wether the bike has walk assist.
Multiplicity	0..1
Datatype	Boolean
Allowed values	TRUE FALSE
Example value	FALSE
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	walk_assist
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	walk_assist
Implementation rule	-

## 2.9.6 Electric bike type

Electric bike type	
Description	Type of electric bike
Multiplicity	1..1
Datatype	String
Allowed values	DST Codelist Electric Bike Types
Example value	NORMAAL
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	type_elektrische_fiets
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	type_elektrische_fiets
Implementation rule	-

## 2.9.7 Battery/Range

Battery/Range	
Description	Container for the specification of the possible batteries in the bike and their range.
Multiplicity	1..n
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	If a csv file is used to exchange data it is not possible to specify multiple batteries for an electric bike. Besides that it's also not possible to specify multiple ranges. All elements in the container Battery/range, including the battery specifics, need to be used to specify the data for 1 battery.
<i>JSON implementation</i>	
Fieldname	accu_actieradius
Implementation rule	-

## 2.9.7.1 Battery specific elements

Battery specific elements	
Description	All elements in the group 'Battery specifics' can be used here (for an explanation see chapter 2.10).
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	accu_specifieke_elementen
Implementation rule	-

## 2.9.7.2 Range

Range	
Description	Container for specification of the range of the specified battery.
Multiplicity	0..n
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	If a csv file is used for the data exchange, it isn't possible to possible to specify multiple ranges. All fields in the container 'Range' can only be used to specify the information on one range.
<i>JSON implementation</i>	
Fieldname	actieradius
Implementation rule	-

## 2.9.7.2.1 Setting range

Setting range	
Description	Used setting on the bike.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	Eco
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	stand_actieradius
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	stand_actieradius
Implementation rule	-

## 2.9.7.2.2 Min range

Min range	
Description	Minimum kilometers.
Multiplicity	0..1
Datatype	Integer
Allowed values	
Example value	55
Business rules	
-	
CSV implementation	
Fieldname	min_actieradius
Implementation rule	-
JSON implementation	
Fieldname	min_actieradius
Implementation rule	-

## 2.9.7.2.3 Max range

Max range	
Description	Maximum kilometers.
Multiplicity	0..1
Datatype	Integer
Allowed values	
Example value	90
Business rules	
-	
CSV implementation	
Fieldname	max_actieradius
Implementation rule	-
JSON implementation	
Fieldname	max_actieradius
Implementation rule	-

## 2.9.7.2.4 Average range

Average range	
Description	Average kilometers.
Multiplicity	0..1
Datatype	Integer
Allowed values	
Example value	75
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	gemiddeld_actieradius
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	gemiddeld_actieradius
Implementation rule	-

## 2.9.7.3 Battery included in base price

Battery included in base price	
Description	Indication wether the specified battery in included in the recommended retail price.
Multiplicity	1..1
Datatype	Boolean
Allowed values	TRUE FALSE
Example value	TRUE
<i>Business rules</i>	
-	
<i>CSV implementation</i>	
Fieldname	accu_in_basisprijs
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	accu_in_basisprijs
Implementation rule	-

## 2.9.7.4 Extra cost battery

Extra cost battery	
Description	The extra cost for the specified battery in the specified currency relative to the battery that would be delivered with the bike by default.
Multiplicity	0..1
Datatype	Decimal
Allowed values	
Example value	150.00
<i>Business rules</i>	
<ul style="list-style-type: none"> <li>- <b>The element needs to contain a positive decimal with a point (.) as separator and 2 decimals after the point.</b></li> <li>- <b>The element shouldn't contain a currency specification.</b></li> </ul>	
<i>CSV implementation</i>	
Fieldname	meerprijs_accu
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	meerprijs_accu
Implementation rule	-

## 2.9.7.5 Position battery

Position battery	
Description	The position of the battery.
Multiplicity	0..1
Datatype	String
Allowed values	DST Codelist Position battery
Example value	FRAME
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	plaats_accu
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	plaats_accu
Implementation rule	-

## 2.9.7.6 Battery removable

<b>Battery removable</b>	
Description	Indication wether the battery is/isn't removable.
Multiplicity	0..1
Datatype	Boolean
Allowed values	TRUE FALSE
Example value	TRUE
<i>Business rules</i>	
<i>CSV implementation</i>	
Fieldname	accu_uitneembaar
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	accu_uitneembaar
Implementation rule	-

## 2.9.7.7 Battery chargeable on bike

<b>Battery chargeable on bike</b>	
Description	Indication wether the battery is/isn't chargeable while it's on the bike.
Multiplicity	0..1
Datatype	Boolean
Allowed values	TRUE FALSE
Example value	TRUE
<i>Business rules</i>	
<i>CSV implementation</i>	
Fieldname	accu_oplaadbaar_in_fiets
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	accu_oplaadbaar_in_fiets
Implementation rule	-



## 2.9.8 Engine

Engine	
Description	Container for information about the engine in the bike.
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	motor
Implementation rule	-

## 2.9.8.1 Engine specific elements

Engine specifics elements	
Description	All elements in the group 'Engine specifics' can be used here (for an explanation see chapter 2.12).
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	motor_specifieke_elementen
Implementation rule	-

## 2.9.8.2 Position engine

Position engine	
Description	The position of the engine on the bike.
Multiplicity	0..1
Datatype	String
Allowed values	DST Codelist Position engine
Example value	VOORWIEL
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	plaats_motor
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	plaats_motor
Implementation rule	-

## 2.9.9 Display

Display	
Description	Container for information about the display on the bike.
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	display
Implementation rule	-

## 2.9.9.1 Display specific elements

Display specifics elements	
Description	All elements in the group 'Display specifics' can be used here (for an explanation see chapter 2.11).
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	display_specifieke_elementen
Implementation rule	-

## 2.9.9.2 Display included in base price bike

Display included in base price bike	
Description	Indication whether the display is included in the base price of the bike.
Multiplicity	0..1
Datatype	Boolean
Allowed values	TRUE FALSE
Example value	TRUE
<i>Business rules</i>	
<i>CSV implementation</i>	
Fieldname	display_in_basisprijs
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	display_in_basisprijs
Implementation rule	-

## 2.9.9.3 Extra cost display

Extra cost display	
Description	The extra cost for the specified display relative to the display that would be delivered with the bike by default.
Multiplicity	0..1
Datatype	Decimal
Allowed values	
Example value	100.00
Business rules	
<ul style="list-style-type: none"> <li>- <b>The element needs to contain a positive decimal with a point (.) as separator and 2 decimals after the point.</b></li> <li>- <b>The element shouldn't contain a currency specification.</b></li> </ul>	
CSV implementation	
Fieldname	meerprijs_display
Implementation rule	-
JSON implementation	
Fieldname	meerprijs_display
Implementation rule	-

## 2.10 Battery specifics

Battery specifics	
Description	Container for information about the battery.
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
Business rules	
CSV implementation	
Fieldname	n/a
Implementation rule	-
JSON implementation	
Fieldname	accu_specifieke_velden
Implementation rule	-

## 2.10.1 Battery capacity

Battery capacity	
Description	The capacity of the battery in Wh.
Multiplicity	0..1
Datatype	Integer
Allowed values	
Example value	400
<i>Business rules</i>	
<i>CSV implementation</i>	
Fieldname	accuvermogen
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	accuvermogen
Implementation rule	-

## 2.10.2 Battery voltage

Battery voltage	
Description	The voltage of the battery in Volt.
Multiplicity	0..1
Datatype	Decimal
Allowed values	
Example value	36.00
<i>Business rules</i>	
<b>The element needs to contain a positive decimal with a point (.) as separator and 2 decimals after the point.</b>	
<i>CSV implementation</i>	
Fieldname	accuvoltage
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	accuvoltage
Implementation rule	-

## 2.10.3 Battery brand

Battery brand	
Description	The brand of the battery.
Multiplicity	0..1
Datatype	String
Allowed values	DST Codelist Brands
Example value	SPARTA
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	merk_accu
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	merk_accu
Implementation rule	-

## 2.10.4 Battery ampere-hour

Battery ampere-hour	
Description	The number of ampere-hour of the battery.
Multiplicity	0..1
Datatype	Decimal
Allowed values	
Example value	11.10
<i>Business rules</i>	
<b>The element needs to contain a positive decimal with a point (.) as separator and 2 decimals after the point.</b>	
<i>CSV implementation</i>	
Fieldname	accu_ampere_uur
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	accu_ampere_uur
Implementation rule	-

## 2.10.5 Battery model

Battery model	
Description	The model name of the battery.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	ION-300
<i>Business rules</i>	
<i>CSV implementation</i>	
Fieldname	accumodel
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	accumodel
Implementation rule	-

## 2.10.6 Battery type

Battery type	
Description	The type of the battery.
Multiplicity	0..1
Datatype	String
Allowed values	DST Codelist Battery type
Example value	LI-ION
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	type_accu
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	type_accu
Implementation rule	-

## 2.10.7 Battery weight

Battery weight	
Description	The weight of the battery in kilogram.
Multiplicity	0..1
Datatype	Decimal
Allowed values	
Example value	2.50
<i>Business rules</i>	
<b>The element needs to contain a positive decimal with a point (.) as separator and 2 decimals after the point.</b>	
<i>CSV implementation</i>	
Fieldname	gewicht_accu
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	gewicht_accu
Implementation rule	-

## 2.10.8 Recharge time normal charger

Recharge time normal charger	
Description	The time needed to charge the battery with a normal charger in hours.
Multiplicity	0..1
Datatype	Decimal
Allowed values	
Example value	2.50
<i>Business rules</i>	
<b>The element needs to contain a positive decimal with a point (.) as separator and 2 decimals after the point.</b>	
<i>CSV implementation</i>	
Fieldname	oplaadtijd_normale_lader
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	oplaadtijd_normale_lader
Implementation rule	-



## 2.10.9 Recharge time fast charger

Recharge time fast charger	
Description	The time needed to charge the battery with a fast charger in hours.
Multiplicity	0..1
Datatype	Decimal
Allowed values	
Example value	0.50
<i>Business rules</i>	
<b>The element needs to contain a positive decimal with a point (.) as separator and 2 decimals after the point.</b>	
<i>CSV implementation</i>	
Fieldname	oplaadtijd_snellader
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	oplaadtijd_snellader
Implementation rule	-

## 2.11 Display specifics

Display Specifics	
Description	Container for information about the display.
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	display_specifieke_velden
Implementation rule	-

## 2.11.1 Display brand

Display brand	
Description	The brand of the display that is provided.
Multiplicity	0..1
Datatype	String
Allowed values	DST Codelist Brands
Example value	BOSCH
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	merk_display
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	merk_display
Implementation rule	-

## 2.11.2 Type display

Type display	
Description	The type of display that is provided.
Multiplicity	0..1
Datatype	String
Allowed values	DST Codelist Display Types
Example value	hd-display
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	type_display
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	type_display
Implementation rule	-

## 2.11.3 Operation type

Operation type	
Description	The type of operation of the display
Multiplicity	0..1
Datatype	String
Allowed values	DST Codelist Operation Type
Example value	STUURBEDIENING
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	type_bediening
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	type_bediening
Implementation rule	-

## 2.11.4 Display removable

Display removable	
Description	Indication whether the display is removable or not.
Multiplicity	0..1
Datatype	Boolean
Allowed values	TRUE FALSE
Example value	TRUE
<i>Business rules</i>	
<i>CSV implementation</i>	
Fieldname	afneembaar_display
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	afneembaar_display
Implementation rule	-

## 2.12 Engine specifics

Engine specifics	
Description	Container for information about an engine.
Multiplicity	0..1
Datatype	Container
Allowed values	
Example value	n/a
<i>Business rules</i>	
<i>CSV implementation</i>	
Fieldname	n/a
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	motor_specifieke_velden
Implementation rule	-

### 2.12.1 Engine model name

Engine model name	
Description	The model name of the engine as used by the supplier.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	Innergy
<i>Business rules</i>	
<i>CSV implementation</i>	
Fieldname	modelnaam_motor
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	modelnaam_motor
Implementation rule	-

## 2.12.2 Engine type

Engine type	
Description	The type of engine the bike has.
Multiplicity	0..1
Datatype	String
Allowed values	
Example value	X2
<i>Business rules</i>	
<i>CSV implementation</i>	
Fieldname	type_motor
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	type_motor
Implementation rule	-

## 2.12.3 Engine brand

Engine brand	
Description	The brand of the engine.
Multiplicity	0..1
Datatype	String
Allowed values	DST Codelist Brands
Example value	BOSCH
<i>Business rules</i>	
<b>The value in this element needs to be in the codelist.</b>	
<i>CSV implementation</i>	
Fieldname	merk_motor
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	merk_motor
Implementation rule	-

## 2.12.4 Engine wattage

Engine wattage	
Description	The time needed to charge the battery with a fast charger in hours.
Multiplicity	0..1
Datatype	Decimal
Allowed values	
Example value	250
<i>Business rules</i>	
<i>CSV implementation</i>	
Fieldname	wattage_motor
Implementation rule	-
<i>JSON implementation</i>	
Fieldname	wattage_motor
Implementation rule	-

## **3 Code/value lists**

### **3.1 Introduction**

The values for a few elements need to be in a list. Such a list is called a code- or value list. When the value for an element needs be in a defined list, we call this a value list. When there's a code linked to this value, that needs to be used in the data exchange, we call this a codelist.

### **3.2 Use of code/value lists**

Value lists can be used for:

- Finding possible values (for example for the pre fill of an optdonlist in a search system).
- The validation of exchanged files/messages. It can be checked if a value is in a value list.

DST has defined a few value lists. That means:

- It is defined which value list there are.
- Per value list the allowed values are defined.
- For value lists with mainly text values, but where unique identifiers are important, unique codes are defined (codelist).

The values are defined per value list in a spreadsheet, which is added to this standard as an attachment. The code/value lists can be edited separately from the standard. Change requests for the code/value lists can be made at the organisation that manages this standard.

In the next paragraphs the different code/value lists will be discussed briefly.

### **3.3 Codelist Parties**

This codelist contains a unique identifier for all parties (suppliers) that (can) provide product information. This identifiers is used to - in combination with an item number - uniquely identify products.

### **3.4 Codelist Statuses**

This valueless contains the possible order statuses of a product.

### **3.5 Codelist Product groups**

This codelist contains the possible product groups for the product. Per product group a main group (collection product groups) and a sub group (subdivision product groups) is defined.

The layout of the codes for the product groups are XYZZ and are mad as follows:

- One number X has been allocated to every main group (1, 2, ...).
- One letter Y has been allocated to every product group within this main group.
- Two numbers have been allocated to every sub group: ZZ (01, 02, ..).

### **3.6 Codelist Customer groups**

This value list contains the possible customer groups of a product.

### **3.7 Codelist Keywords**

This value list contains the keywords for a product.

### **3.8 Codelist Position**

This codelist contains the positions that can be used for a product in combination with a keyword.

### **3.9 Codelist Brands**

This value list describes the possible brands that can occur.

### **3.10 Codelist Base colours**

This value list describes the possible base colour(s) of a product.

### **3.11 Codelist VAT**

This codelist describes the possible VAT percentages.

### **3.12 Codelist ERTRO**

This codelist contains the ERTRO sizes for tyres.

### **3.13 Codelist Frame types**

This value list contains the possible frame types of the bike.

### **3.14 Codelist Frame materials**

This value list contains the possible materials of a frame.

### **3.15 Codelist Gear types**

This valueless describes the possible gear types.



### **3.16 Codelist Brake types**

This value list contains the possible brake types.

### **3.17 Codelist Document types**

This value list describes the possible document types that are attached.

### **3.18 Codelist Ebike systems**

This value list describes the possible electric bike systems.

### **3.19 Codelist Electric bike types**

This value list describes the possible electric bike types.

### **3.20 Codelist Sensor types**

This value list describes the possible sensor types for an electric bike.

### **3.21 Codelist Battery position**

This value list describes the possible positions for a battery on an electric bike.

### **3.22 Codelist Engine position**

This value list describes the possible positions for an engine on an electric bike.

### **3.23 Codelist Battery types**

This value list describes the possible battery types for an electric bike.

### **3.24 Codelist Display types**

This value list describes the possible display types for an electric bike.

### **3.25 Codelist Display operation types**

This value list describes the possible operation types for the display on an electric bike.

## 4 CSV syntax

### 4.1 Introduction

A traditional exchange of data is done based on CSV. This is the way a lot of POS-systems work at the moment: the system reads a 'comma separated values' file and translates this to the internal database of the system.

### 4.2 Layout of CSV

CSV is a collective name for multiple file formats that are all:

- based on text.
- separated by a comma or semicolon.
- ending lines by a semicolon or carriage return.

***To make sure that the same file format is used within the sector, this standard specifies that guideline 4180 of the IETF<sup>1</sup> will be used for CSV files.***

The most important characteristics of this file format are:

1. Every element is ended with a comma.  
jan,adresstraat 12,diemen
2. Every element can be started and ended with double quotes. *Within the sector it is advised not to do this unless necessary (see 3 and the implementation rules per element).*  
"jan","adresstraat 12","diemen"
3. If an element has double quotes, carriage returns or a comma, the element needs to be started and ended with double quotes.  
"jan, klaassen",adresstraat 12,diemen
4. Every record has it's own rule, which is ended with a carriage return.
5. The last record can have a carriage return as well.
6. The first rule contains the column headers. Within the sector, in addition to RFC4180, these column headers are mandatory.  
voornaam,achternaam,adres,plaats
7. For elements that have no value (because the it's optional) the column header does need to be in the file. There just wouldn't be a value for that specific product in this column.
8. The encoding of the CSV file is utf-8 without BOM.

---

<sup>1</sup> <http://tools.ietf.org/html/rfc4180>

## **4.2 Implementation of CSV elements**

For all elements of the information model the fieldnames (in Dutch) are described in the tables of chapter 2. If applicable, extra implementation rules are included.

## 5 JSON syntax

### 5.1 Introduction

JSON (JavaScript Object Notation) is a lightweight format for data exchange. JSON is easy to use for programmers, and easy to process and generate for computers.

### 5.2 Layout of JSON

***To make sure that the same file format is used within the sector, this standard specifies that guideline RFC 7159 of the IETF<sup>2</sup> will be used for JSON files.***

The most important characteristics of this file format are:

1. Within JSON objects and arrays (lists) are distinguished. An object is an unsorted collection of *name-value pairs*. The *name* (fieldname) is a string is described in this standard in chapter 2 at all the different elements. The *value* can be a *string*, *number*, *boolean* (*true*, *false*), *null*, *object* or *array*. An array is a sorted collection of zero or more values.
2. For the use of numbers (*integers* and *decimals*) these are the rules:
  - A. A number is represented in base 10 with decimals.
  - B. Optional a minus ( - ) can be used before the number to indicate a negative.
  - C. For decimals a point ( . ) is used as a separator.
  - D. *Leading zeros* are not allowed.
3. For the representation of some characters in strings escaping has to be used. We use the backslash ( \ ) to do this. The most important characters that need to be escaped are the dubbel quote ( " ) and the backslash it selves.
4. JSON text needs to be encoded according to utf-8, utf-16 or utf-32. However, within this standard utf-8 needs to be used.

For more information on the JSON implementation we refer to the RFC 7159 specification.

---

<sup>2</sup> <http://tools.ietf.org/html/rfc7159>