

Schema

SUTI MessageSchema

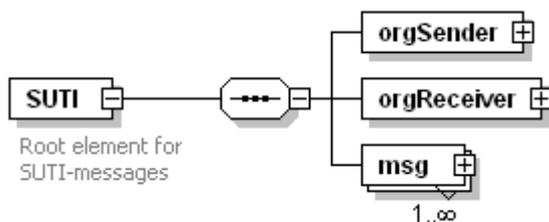
Schema **SUTI_MessageXSD_1_1_0.xsd**

schema location: L:\Admin\SUT\Standard\SUTI V 1.1.0\Pdf\SUTI_MessageXSD_1_1_0.xsd
attribute form default: **unqualified**
element form default: **qualified**

Elements Complex types
SUTI [addressType](#)
 [agreement](#)
 [associatedReservation](#)
 [attribute](#)
 [attributesType](#)
 [cancellationConsequence](#)
 [capacity](#)
 [connection](#)
 [contactInfo](#)
 [contactInfosType](#)
 [content](#)
 [contents](#)
 [driver](#)
 [economyType](#)
 [exchangeRates](#)
 [exchangeRate](#)
 [formOfPayment](#)
 [geographicLocation](#)
 [idType](#)
 [manualDescriptionType](#)
 [msg](#)
 [multiDispatch](#)
 [node](#)
 [order](#)
 [orderReject](#)
 [orgType](#)
 [payment](#)
 [pickupConfirmation](#)
 [position](#)
 [price](#)
 [priceCalculation](#)
 [process](#)
 [product](#)
 [referencesTo](#)
 [resourceType](#)
 [route](#)
 [seats](#)
 [subOrderType](#)
 [taxiMeter](#)
 [time](#)
 [timesType](#)
 [vehicle](#)

element SUTI

diagram



properties content complex

children [orgSender](#) [orgReceiver](#) [msg](#)

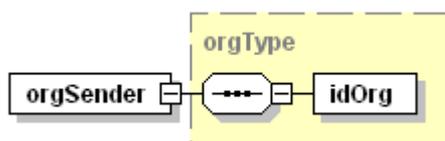
annotation documentation
Root element for SUTI-messages

```

source <xs:element name="SUTI">
  <xs:annotation>
    <xs:documentation>Root element for SUTI-messages</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="orgSender" type="orgType"/>
      <xs:element name="orgReceiver" type="orgType"/>
      <xs:element name="msg" maxOccurs="unbounded">
        <xs:complexType>
          <xs:complexContent>
            <xs:extension base="msg"/>
          </xs:complexContent>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
    
```

element SUTI/orgSender

diagram



type [orgType](#)

properties isRef 0
content complex

children [idOrg](#)

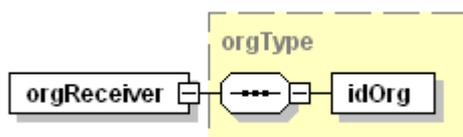
attributes	Name	Type	Use	Default	Fixed	Annotation
	name	xs:string	required			

```

source <xs:element name="orgSender" type="orgType"/>
    
```

element SUTI/orgReceiver

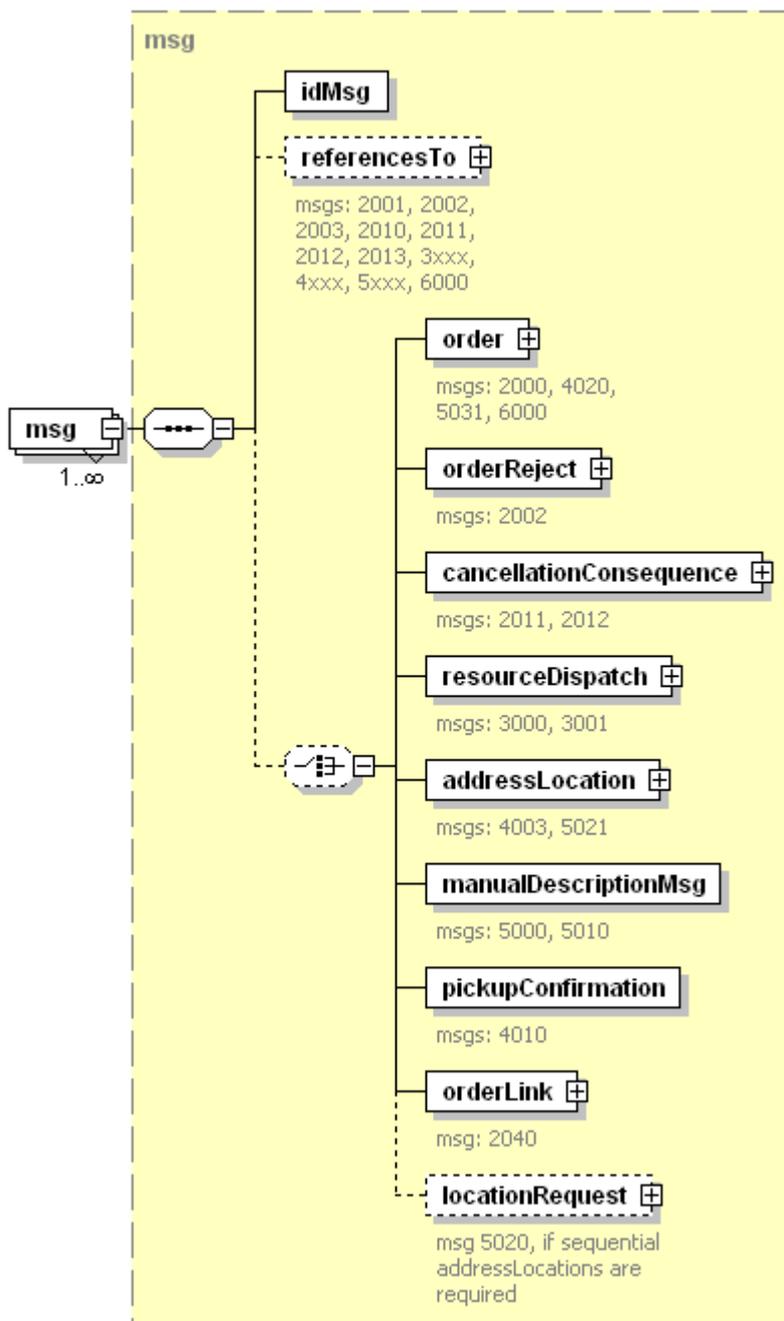
diagram



type [orgType](#)
 properties isRef 0
 content complex
 children [idOrg](#)
 attributes Name Type Use Default Fixed Annotation
 name **xs:string** required
 source `<xs:element name="orgReceiver" type="orgType"/>`

element **SUTI/msg**

diagram

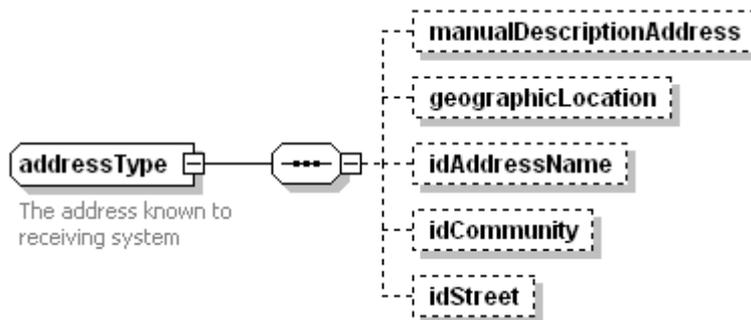


type extension of [msg](#)
 properties isRef 0
 minOcc 1
 maxOcc unbounded

		content	complex			
children		idMsg referencesTo order orderReject cancellationConsequence resourceDispatch addressLocation manualDescriptionMsg pickupConfirmation orderLink locationRequest				
attributes	Name	Type	Use	Default	Fixed	Annotation
	msgType	xs:string	required			
	msgName	xs:string	optional			
source	<pre><xs:element name="msg" maxOccurs="unbounded"> <xs:complexType> <xs:complexContent> <xs:extension base="msg"/> </xs:complexContent> </xs:complexType> </xs:element></pre>					

complexType **addressType**

diagram



children	manualDescriptionAddress geographicLocation idAddressName idCommunity idStreet					
used by	elements	msg/addressLocation node/addressNode associatedReservation/addressReservation				
attributes	Name	Type	Use	Default	Fixed	Annotation
	addressName	xs:string	optional			
	street	xs:string	optional			
	streetNo	xs:positiveInteger	optional			
	streetNoLetter	xs:string	optional			
	location	xs:string	optional			
	community	xs:string	optional			
	postalNo	xs:string	optional			
	country	xs:string	optional			
annotation	documentation	The address known to receiving system				
source	<pre><xs:complexType name="addressType"> <xs:annotation> <xs:documentation>The address known to receiving system</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="manualDescriptionAddress" type="manualDescriptionType" minOccurs="0"/> <xs:element name="geographicLocation" type="geographicLocation" minOccurs="0"/> <xs:element name="idAddressName" type="idType" minOccurs="0"/> <xs:element name="idCommunity" type="idType" minOccurs="0"/> <xs:element name="idStreet" type="idType" minOccurs="0"/> </xs:sequence> <xs:attribute name="addressName" type="xs:string" use="optional"/> <xs:attribute name="street" type="xs:string" use="optional"/> <xs:attribute name="streetNo" type="xs:positiveInteger" use="optional"/> <xs:attribute name="streetNoLetter" type="xs:string" use="optional"/> <xs:attribute name="location" type="xs:string" use="optional"/> <xs:attribute name="community" type="xs:string" use="optional"/> <xs:attribute name="postalNo" type="xs:string" use="optional"/> <xs:attribute name="country" type="xs:string" use="optional"/> </xs:complexType></pre>					

element addressType/manualDescriptionAddress

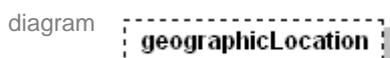


type [manualDescriptionType](#)

properties	isRef	0				
	minOcc	0				
	maxOcc	1				
	content	complex				
attributes	Name	Type	Use	Default	Fixed	Annotation
	manualText	xs:string	required			

source `<xs:element name="manualDescriptionAddress" type="manualDescriptionType" minOccurs="0"/>`

element addressType/geographicLocation

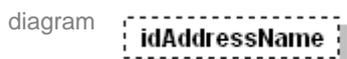


type [geographicLocation](#)

properties	isRef	0				
	minOcc	0				
	maxOcc	1				
	content	complex				
attributes	Name	Type	Use	Default	Fixed	Annotation
	zone	xs:string	optional			
	typeOfCoordinate	xs:string	required			
	lat	xs:float	required			
	long	xs:float	required			
	precision	xs:integer	required			

source `<xs:element name="geographicLocation" type="geographicLocation" minOccurs="0"/>`

element addressType/idAddressName

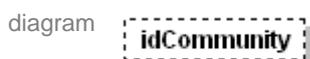


type [idType](#)

properties	isRef	0				
	minOcc	0				
	maxOcc	1				
	content	complex				
attributes	Name	Type	Use	Default	Fixed	Annotation
	src	xs:string	required			
	id	xs:string	required			
	unique	xs:boolean	optional	false		

source `<xs:element name="idAddressName" type="idType" minOccurs="0"/>`

element addressType/idCommunity

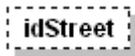


type [idType](#)

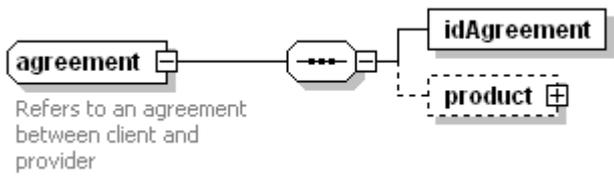
properties	isRef	0
	minOcc	0

	maxOcc	1				
	content	complex				
attributes	Name	Type	Use	Default	Fixed	Annotation
	src	xs:string	required			
	id	xs:string	required			
	unique	xs:boolean	optional	false		
source	<code><xs:element name="idCommunity" type="idType" minOccurs="0"/></code>					

element addressType/idStreet

diagram						
type	idType					
properties	isRef	0				
	minOcc	0				
	maxOcc	1				
	content	complex				
attributes	Name	Type	Use	Default	Fixed	Annotation
	src	xs:string	required			
	id	xs:string	required			
	unique	xs:boolean	optional	false		
source	<code><xs:element name="idStreet" type="idType" minOccurs="0"/></code>					

complexType agreement

diagram						
children	idAgreement product					
used by	element	order/agreement				
attributes	Name	Type	Use	Default	Fixed	Annotation
	name	xs:string	optional			
annotation	documentation	Refers to an agreement between client and provider				
source	<pre> <xs:complexType name="agreement"> <xs:annotation> <xs:documentation>Refers to an agreement between client and provider</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="idAgreement" type="idType"/> <xs:element name="product" type="product" minOccurs="0"/> </xs:sequence> <xs:attribute name="name" type="xs:string" use="optional"/> </xs:complexType> </pre>					

element agreement/idAgreement

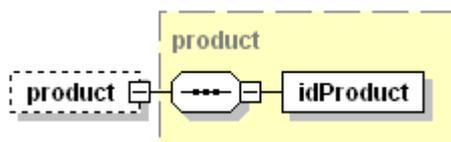
diagram						
type	idType					
properties	isRef	0				
	content	complex				

attributes	Name	Type	Use	Default	Fixed	Annotation
	src	xs:string	required			
	id	xs:string	required			
	unique	xs:boolean	optional	false		

source `<xs:element name="idAgreement" type="idType"/>`

element agreement/product

diagram



type [product](#)

properties	isRef	minOcc	maxOcc	content
	0	0	1	complex

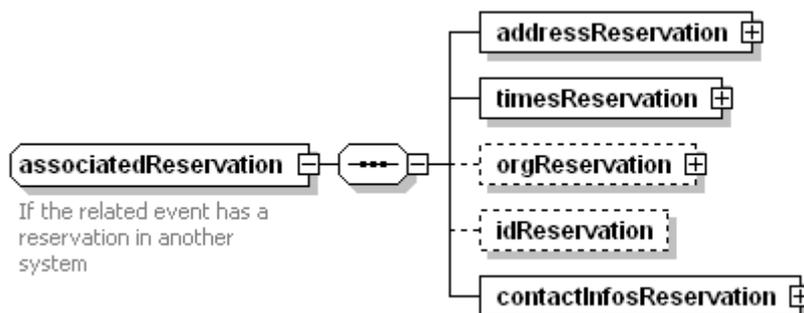
children [idProduct](#)

attributes	Name	Type	Use	Default	Fixed	Annotation
	name	xs:string	optional			

source `<xs:element name="product" type="product" minOccurs="0"/>`

complexType associatedReservation

diagram



children [addressReservation](#) [timesReservation](#) [orgReservation](#) [idReservation](#) [contactInfosReservation](#)

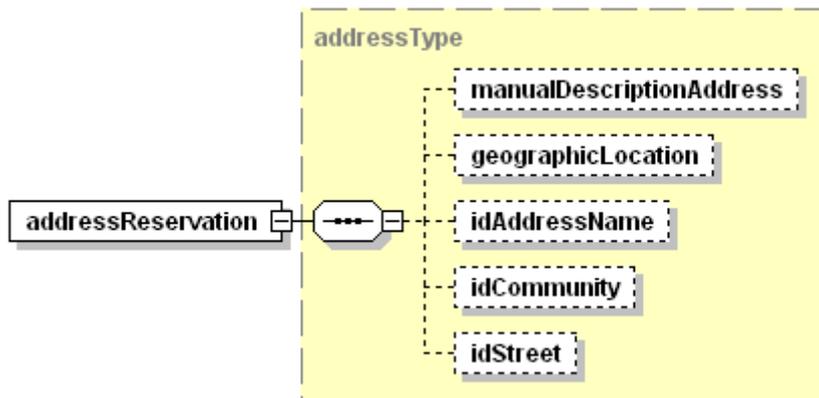
used by element [connection/associatedReservation](#)

annotation documentation
If the related event has a reservation in another system

source `<xs:complexType name="associatedReservation">
 <xs:annotation>
 <xs:documentation>If the related event has a reservation in another system</xs:documentation>
 </xs:annotation>
 <xs:sequence>
 <xs:element name="addressReservation" type="addressType"/>
 <xs:element name="timesReservation" type="timesType"/>
 <xs:element name="orgReservation" type="orgType" minOccurs="0"/>
 <xs:element name="idReservation" minOccurs="0"/>
 <xs:element name="contactInfosReservation" type="contactInfosType"/>
 </xs:sequence>
 </xs:complexType>`

element associatedReservation/addressReservation

diagram



type [addressType](#)

properties isRef 0
content complex

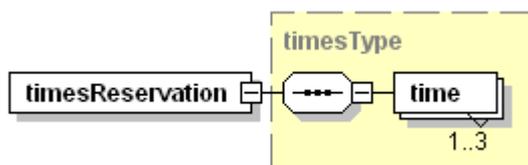
children [manualDescriptionAddress](#) [geographicLocation](#) [idAddressName](#) [idCommunity](#) [idStreet](#)

attributes	Name	Type	Use	Default	Fixed	Annotation
	addressName	xs:string	optional			
	street	xs:string	optional			
	streetNo	xs:positiveInteger	optional			
	streetNoLetter	xs:string	optional			
	location	xs:string	optional			
	community	xs:string	optional			
	postalNo	xs:string	optional			
	country	xs:string	optional			

source `<xs:element name="addressReservation" type="addressType"/>`

element associatedReservation/timesReservation

diagram



type [timesType](#)

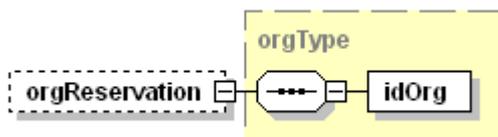
properties isRef 0
content complex

children [time](#)

source `<xs:element name="timesReservation" type="timesType"/>`

element associatedReservation/orgReservation

diagram



type [orgType](#)

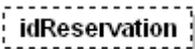
properties isRef 0
 minOcc 0
 maxOcc 1
 content complex

children [idOrg](#)

attributes	Name	Type	Use	Default	Fixed	Annotation
	name	xs:string	required			

source `<xs:element name="orgReservation" type="orgType" minOccurs="0"/>`

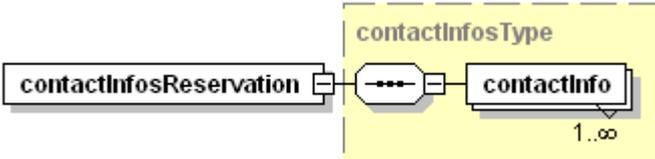
element associatedReservation/idReservation

diagram 

properties isRef 0
 minOcc 0
 maxOcc 1

source `<xs:element name="idReservation" minOccurs="0"/>`

element associatedReservation/contactInfosReservation

diagram 

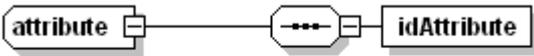
type [contactInfosType](#)

properties isRef 0
 content complex

children [contactInfo](#)

source `<xs:element name="contactInfosReservation" type="contactInfosType"/>`

complexType attribute

diagram 
 There is or should be a presence of the attribute

children [idAttribute](#)

used by element [attributesType/attribute](#)

annotation documentation
 There is or should be a presence of the attribute

source `<xs:complexType name="attribute">
 <xs:annotation>
 <xs:documentation>There is or should be a presence of the attribute</xs:documentation>
 </xs:annotation>
 <xs:sequence>
 <xs:element name="idAttribute" type="idType"/>
 </xs:sequence>
 </xs:complexType>`

element attribute/idAttribute

diagram



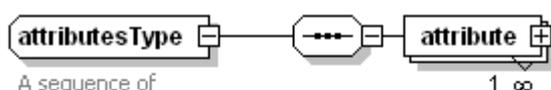
type [idType](#)

properties	isRef	0				
	content	complex				
attributes	Name	Type	Use	Default	Fixed	Annotation
	src	xs:string	required			
	id	xs:string	required			
	unique	xs:boolean	optional	false		

source `<xs:element name="idAttribute" type="idType"/>`

complexType attributesType

diagram



A sequence of attributes, where the presence of an attribute means that the existence of the attribute is true

children [attribute](#)

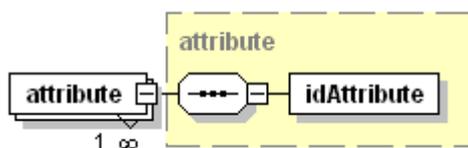
used by elements [driver/attributesDriver](#) [orderReject/attributesReject](#) [vehicle/attributesVehicle](#)

annotation documentation
A sequence of attributes, where the presence of an attribute means that the existence of the attribute is true

source `<xs:complexType name="attributesType">`
`<xs:annotation>`
`<xs:documentation>A sequence of attributes, where the presence of an attribute means that the`
`existence of the attribute is true</xs:documentation>`
`</xs:annotation>`
`<xs:sequence>`
`<xs:element name="attribute" type="attribute" maxOccurs="unbounded"/>`
`</xs:sequence>`
`</xs:complexType>`

element attributesType/attribute

diagram



type [attribute](#)

properties	isRef	0
	minOcc	1
	maxOcc	unbounded
	content	complex

children [idAttribute](#)

source `<xs:element name="attribute" type="attribute" maxOccurs="unbounded"/>`

complexType **cancellationConsequence**



children [economyCancellation](#)

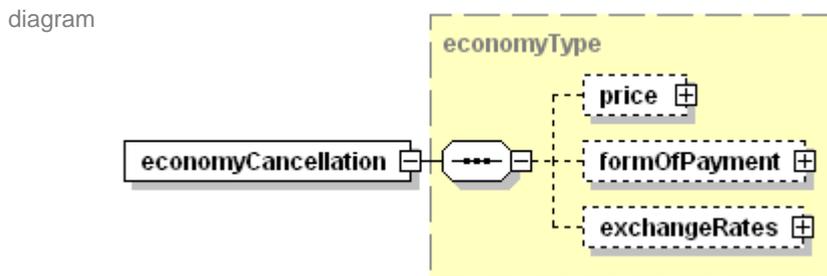
used by element [msg/cancellationConsequence](#)

attributes	Name	Type	Use	Default	Fixed	Annotation
	cancellationAc	xs:boolean	required			
	cancellationConsequence	xs:boolean	required			
annotation	documentation	Whether a cancellation is ok, not ok or their are consequences				

source

```
<xs:complexType name="cancellationConsequence">
  <xs:annotation>
    <xs:documentation>Whether a cancellation is ok, not ok or their are consequences</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="economyCancellation" type="economyType"/>
  </xs:sequence>
  <xs:attribute name="cancellationAcceptance" type="xs:boolean" use="required"/>
  <xs:attribute name="cancellationConsequence" type="xs:boolean" use="required"/>
</xs:complexType>
```

element **cancellationConsequence/economyCancellation**



type [economyType](#)

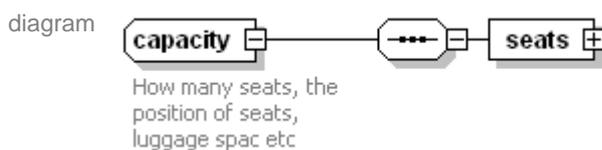
properties isRef 0
content complex

children [price](#) [formOfPayment](#) [exchangeRates](#)

source

```
<xs:element name="economyCancellation" type="economyType"/>
```

complexType **capacity**



children [seats](#)

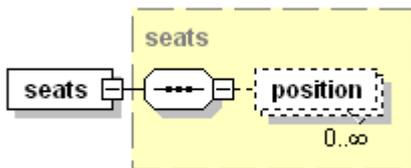
used by element [vehicle/capacity](#)

attributes	Name	Type	Use	Default	Fixed	Annotation
	luggageArea	xs:float	optional			

fullArea **xs:float** optional
 stretcherArea **xs:float** optional
 annotation documentation
 How many seats, the position of seats, luggage spac etc
 source `<xs:complexType name="capacity">`
 `<xs:annotation>`
 `<xs:documentation>How many seats, the position of seats, luggage spac etc</xs:documentation>`
 `</xs:annotation>`
 `<xs:sequence>`
 `<xs:element name="seats" type="seats"/>`
 `</xs:sequence>`
 `<xs:attribute name="luggageArea" type="xs:float" use="optional"/>`
 `<xs:attribute name="fullArea" type="xs:float" use="optional"/>`
 `<xs:attribute name="stretcherArea" type="xs:float" use="optional"/>`
 `</xs:complexType>`

element capacity/seats

diagram



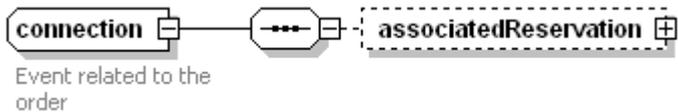
type [seats](#)
 properties isRef 0
 content complex
 children [position](#)
 attributes

Name	Type	Use	Default	Fixed	Annotation
noOfSeats	xs:nonNegati veInteger	required			

 source `<xs:element name="seats" type="seats"/>`

complexType connection

diagram



children [associatedReservation](#)
 used by element [content/connection](#)
 attributes

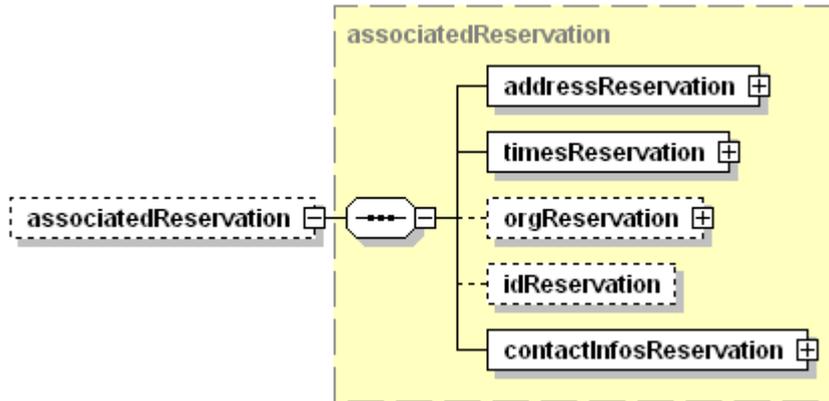
Name	Type	Use	Default	Fixed	Annotation
connectionType	xs:string	required			
connectionName	xs:string	optional			
connectionArr	xs:string	required			

 annotation documentation
 Event related to the order
 source `<xs:complexType name="connection">`
 `<xs:annotation>`
 `<xs:documentation>Event related to the order</xs:documentation>`
 `</xs:annotation>`
 `<xs:sequence>`
 `<xs:element name="associatedReservation" type="associatedReservation" minOccurs="0"/>`
 `</xs:sequence>`
 `<xs:attribute name="connectionType" type="xs:string" use="required"/>`
 `<xs:attribute name="connectionName" type="xs:string" use="optional"/>`

```
<xs:attribute name="connectionArrDep" type="xs:string" use="required"/>
</xs:complexType>
```

element connection/associatedReservation

diagram



type [associatedReservation](#)

properties
 isRef 0
 minOcc 0
 maxOcc 1
 content complex

children [addressReservation](#) [timesReservation](#) [orgReservation](#) [idReservation](#) [contactInfosReservation](#)

source `<xs:element name="associatedReservation" type="associatedReservation" minOccurs="0"/>`

complexType contactInfo

diagram



used by element [contactInfosType/contactInfo](#)

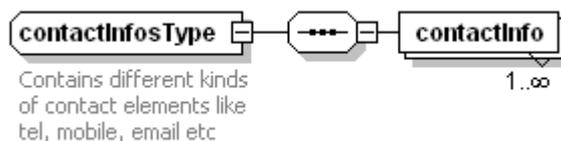
attributes	Name	Type	Use	Default	Fixed	Annotation
	contactType	xs:string	required			
	contactInfo	xs:string	required			

documentation
 The telno, mailaddress etc

```
source <xs:complexType name="contactInfo">
  <xs:annotation>
    <xs:documentation>The telno, mailaddress etc</xs:documentation>
  </xs:annotation>
  <xs:attribute name="contactType" type="xs:string" use="required"/>
  <xs:attribute name="contactInfo" type="xs:string" use="required"/>
</xs:complexType>
```

complexType contactInfosType

diagram



children [contactInfo](#)

used by elements [content/contactInfosContent](#) [associatedReservation/contactInfosReservation](#)

annotation documentation
Contains different kinds of contact elements like tel, mobile, email etc

source `<xs:complexType name="contactInfosType">
<xs:annotation>
<xs:documentation>Contains different kinds of contact elements like tel, mobile, email
etc</xs:documentation>
</xs:annotation>
<xs:sequence>
<xs:element name="contactInfo" type="contactInfo" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>`

element **contactInfosType/contactInfo**



type [contactInfo](#)

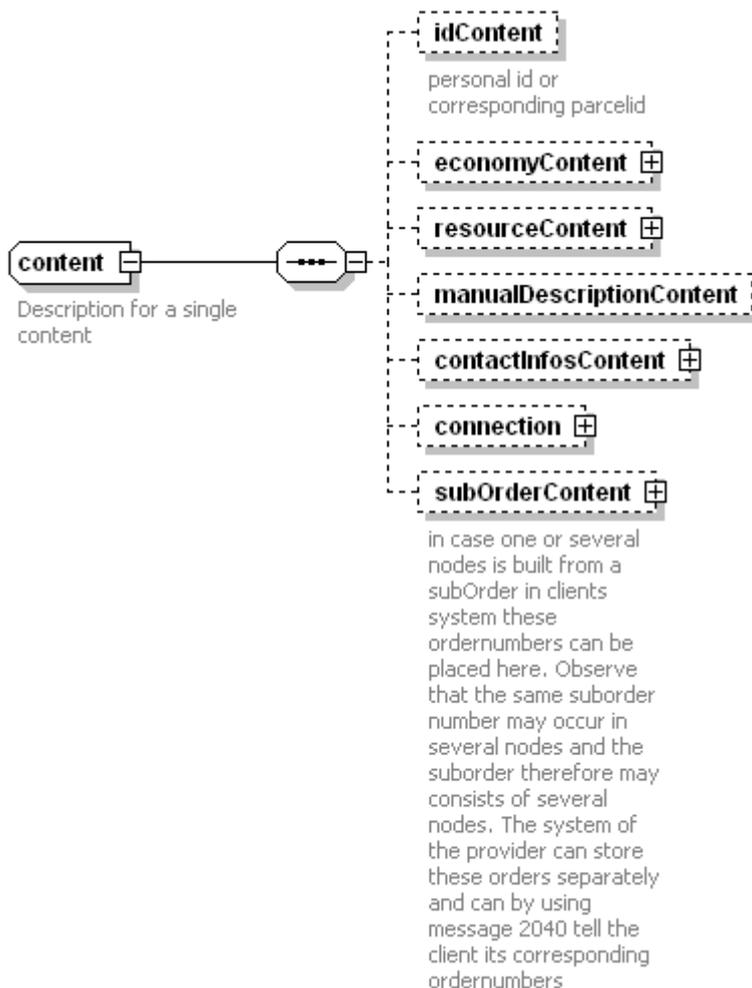
properties isRef 0
minOcc 1
maxOcc unbounded
content complex

attributes	Name	Type	Use	Default	Fixed	Annotation
	contactType	xs:string	required			
	contactInfo	xs:string	required			

source `<xs:element name="contactInfo" type="contactInfo" maxOccurs="unbounded"/>`

complexType content

diagram



children [idContent](#) [economyContent](#) [resourceContent](#) [manualDescriptionContent](#) [contactInfosContent](#) [connection](#) [subOrderContent](#)

used by element [contents/content](#)

attributes	Name	Type	Use	Default	Fixed	Annotation
	contentType	xs:string	required			
	name	xs:string	required			

annotation
documentation
Description for a single content

```

source <xs:complexType name="content">
  <xs:annotation>
    <xs:documentation>Description for a single content</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="idContent" type="idType" minOccurs="0">
      <xs:annotation>
        <xs:documentation>personal id or corresponding parcelid</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="economyContent" type="economyType" minOccurs="0"/>
    <xs:element name="resourceContent" type="resourceType" minOccurs="0"/>
    <xs:element name="manualDescriptionContent" type="manualDescriptionType" minOccurs="0"/>
    <xs:element name="contactInfosContent" type="contactInfosType" minOccurs="0"/>
    <xs:element name="connection" type="connection" minOccurs="0"/>
    <xs:element name="subOrderContent" type="subOrderType" minOccurs="0">
      <xs:annotation>

```

<xs:documentation>in case one or several nodes is built from a subOrder in clients system these ordernumbers can be placed here. Observe that the same suborder number may occur in several nodes and the suborder therefore may consists of several nodes. The system of the provider can store these orders separately and can by using message 2040 tell the client its corresponding ordernumbers</xs:documentation>

```

</xs:annotation>
</xs:element>
</xs:sequence>
<xs:attribute name="contentType" type="xs:string" use="required"/>
<xs:attribute name="name" type="xs:string" use="required"/>
</xs:complexType>
    
```

element content/idContent



type [idType](#)

properties isRef 0
minOcc 0
maxOcc 1

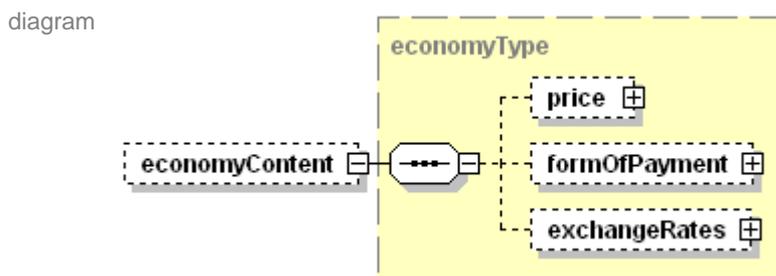
attributes	Name	Type	Use	Default	Fixed	Annotation
	src	xs:string	required			
	id	xs:string	required			
	unique	xs:boolean	optional	false		

annotation documentation
personal id or corresponding parcelid

```

source <xs:element name="idContent" type="idType" minOccurs="0">
  <xs:annotation>
    <xs:documentation>personal id or corresponding parcelid</xs:documentation>
  </xs:annotation>
</xs:element>
    
```

element content/economyContent



type [economyType](#)

properties isRef 0
minOcc 0
maxOcc 1
content complex

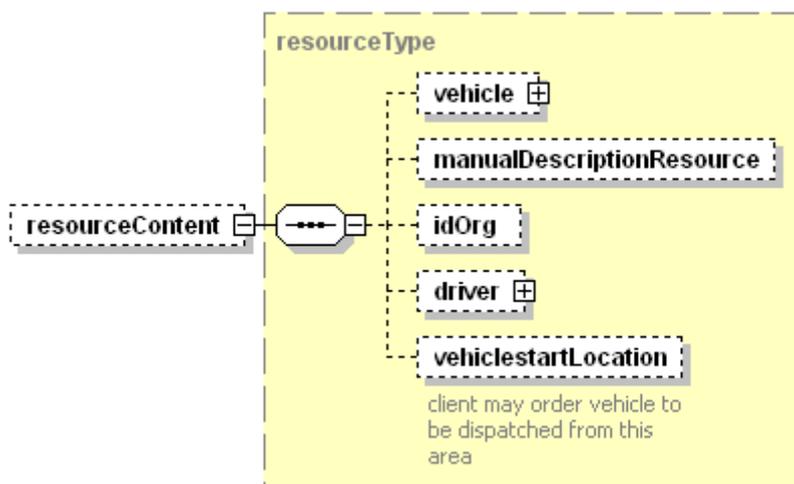
children [price](#) [formOfPayment](#) [exchangeRates](#)

```

source <xs:element name="economyContent" type="economyType" minOccurs="0"/>
    
```

element content/resourceContent

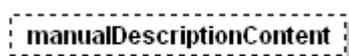
diagram



type [resourceType](#)
 properties isRef 0
 minOcc 0
 maxOcc 1
 content complex
 children [vehicle](#) [manualDescriptionResource](#) [idOrg](#) [driver](#) [vehiclestartLocation](#)
 source `<xs:element name="resourceContent" type="resourceType" minOccurs="0"/>`

element content/manualDescriptionContent

diagram



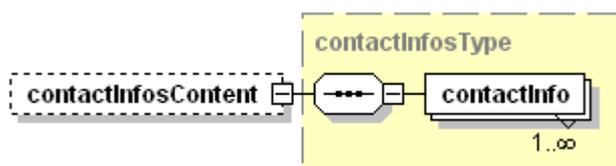
type [manualDescriptionType](#)
 properties isRef 0
 minOcc 0
 maxOcc 1
 content complex
 attributes

Name	Type	Use	Default	Fixed	Annotation
manualText	xs:string	required			

source `<xs:element name="manualDescriptionContent" type="manualDescriptionType" minOccurs="0"/>`

element content/contactInfosContent

diagram

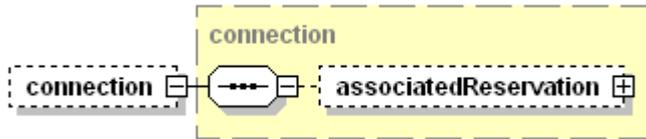


type [contactInfosType](#)
 properties isRef 0
 minOcc 0
 maxOcc 1
 content complex
 children [contactInfo](#)

source `<xs:element name="contactInfosContent" type="contactInfosType" minOccurs="0"/>`

element content/connection

diagram



type [connection](#)

properties

isRef	0
minOcc	0
maxOcc	1
content	complex

children [associatedReservation](#)

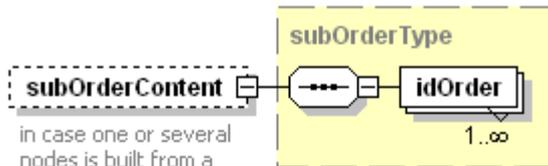
attributes

Name	Type	Use	Default	Fixed	Annotation
connectionType	xs:string	required			
connectionName	xs:string	optional			
connectionArrDep	xs:string	required			

source `<xs:element name="connection" type="connection" minOccurs="0"/>`

element content/subOrderContent

diagram



in case one or several nodes is built from a subOrder in clients system these ordernumbers can be placed here. Observe that the same suborder number may occur in several nodes and the suborder therefore may consists of several nodes. The system of the provider can store these orders separately and can by using message 2040 tell the client its corresponding ordernumbers

type [subOrderType](#)

properties

isRef	0
minOcc	0
maxOcc	1
content	complex

children [idOrder](#)

annotation

documentation

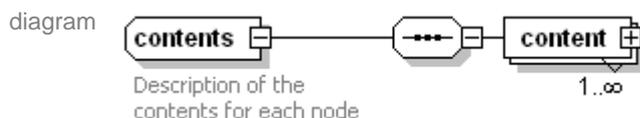
in case one or several nodes is built from a subOrder in clients system these ordernumbers can be placed here. Observe that the same suborder number may occur in several nodes and the suborder therefore may consists of several nodes. The system of the provider can store these orders separately and can by using message 2040 tell the client its corresponding ordernumbers

```

source <xs:element name="subOrderContent" type="subOrderType" minOccurs="0">
  <xs:annotation>
    <xs:documentation>in case one or several nodes is built from a subOrder in clients system these
    ordernumbers can be placed here. Observe that the same suborder number may occur in several nodes
    and the suborder therefore may consists of several nodes. The system of the provider can store these
    orders separately and can by using message 2040 tell the client its corresponding
    ordernumbers</xs:documentation>
  </xs:annotation>
</xs:element>

```

complexType **contents**



children [content](#)

used by element [node/contents](#)

annotation documentation
Description of the contents for each node

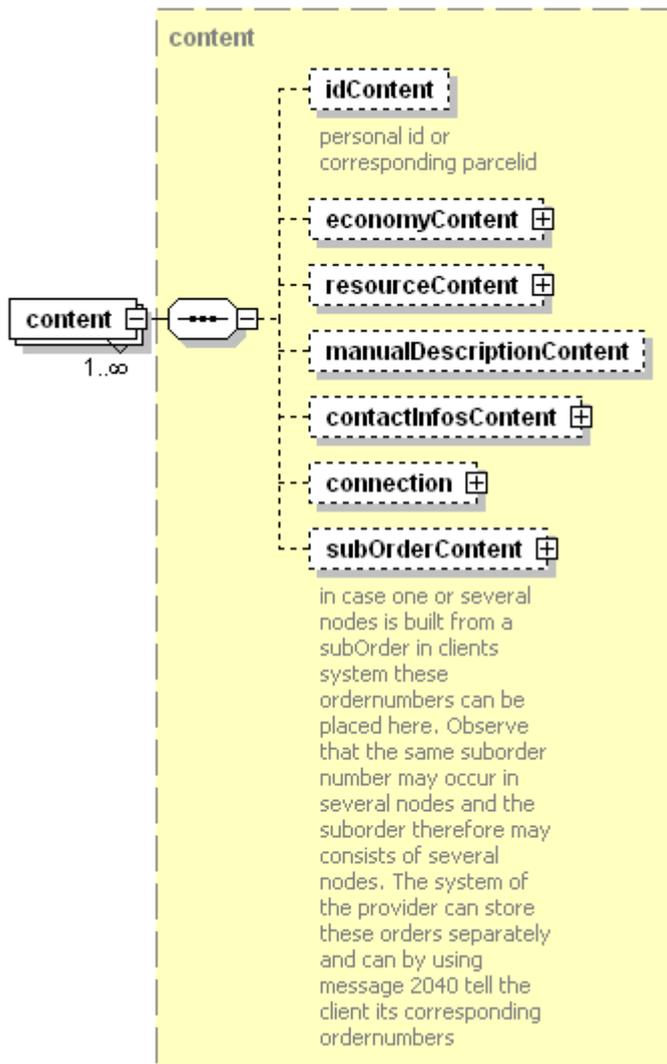
```

source <xs:complexType name="contents">
  <xs:annotation>
    <xs:documentation>Description of the contents for each node</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="content" type="content" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

```

element contents/content

diagram



type [content](#)

properties
 isRef 0
 minOcc 1
 maxOcc unbounded
 content complex

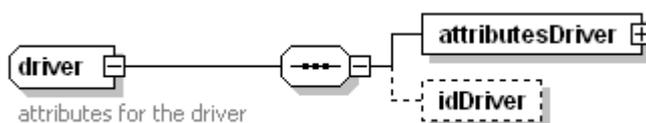
children [idContent](#) [economyContent](#) [resourceContent](#) [manualDescriptionContent](#) [contactInfosContent](#) [connection](#) [subOrderContent](#)

attributes	Name	Type	Use	Default	Fixed	Annotation
	contentType	xs:string	required			
	name	xs:string	required			

source `<xs:element name="content" type="content" maxOccurs="unbounded"/>`

complexType driver

diagram



children [attributesDriver](#) [idDriver](#)

used by element [resourceType/driver](#)

attributes	Name	Type	Use	Default	Fixed	Annotation
	name	xs:string	optional			

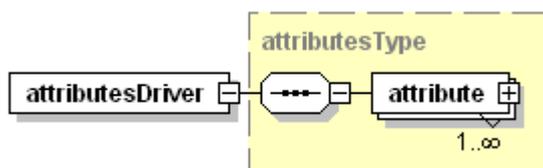
annotation documentation
attributes for the driver

source

```
<xs:complexType name="driver">
  <xs:annotation>
    <xs:documentation>attributes for the driver</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="attributesDriver" type="attributesType"/>
    <xs:element name="idDriver" type="idType" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="name" type="xs:string" use="optional"/>
</xs:complexType>
```

element driver/attributesDriver

diagram



type [attributesType](#)

properties isRef 0
content complex

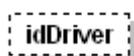
children [attribute](#)

source

```
<xs:element name="attributesDriver" type="attributesType"/>
```

element driver/idDriver

diagram



type [idType](#)

properties isRef 0
minOcc 0
maxOcc 1
content complex

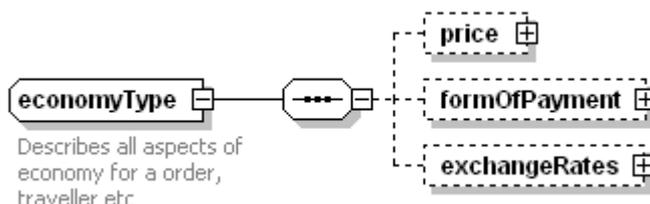
attributes	Name	Type	Use	Default	Fixed	Annotation
	src	xs:string	required			
	id	xs:string	required			
	unique	xs:boolean	optional	false		

source

```
<xs:element name="idDriver" type="idType" minOccurs="0"/>
```

complexType economyType

diagram



children [price](#) [formOfPayment](#) [exchangeRates](#)

used by elements [cancellationConsequence/economyCancellation](#) [content/economyContent](#) [order/economyOrder](#)

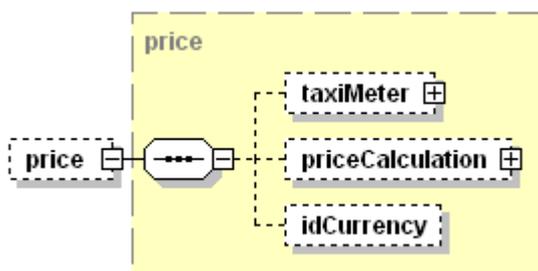
documentation
Describes all aspects of economy for a order, traveller etc

source

```
<xs:complexType name="economyType">
  <xs:annotation>
    <xs:documentation>Describes all aspects of economy for a order, traveller etc</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="price" type="price" minOccurs="0"/>
    <xs:element name="formOfPayment" type="formOfPayment" minOccurs="0"/>
    <xs:element name="exchangeRates" type="exchangeRates" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

element **economyType/price**

diagram



type [price](#)

properties
isRef 0
minOcc 0
maxOcc 1
content complex

children [taxiMeter](#) [priceCalculation](#) [idCurrency](#)

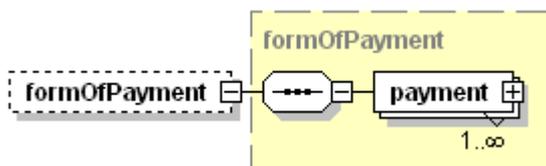
attributes	Name	Type	Use	Default	Fixed	Annotation
	priceResponsi	xs:string	required			
	ble					
	fixedPrice	xs:boolean	required			
	vatIncluded	xs:boolean	required			
	vatPercent	xs:float	optional			
	price	xs:float	optional			
	vat	xs:float	optional			

source

```
<xs:element name="price" type="price" minOccurs="0"/>
```

element **economyType/formOfPayment**

diagram



type [formOfPayment](#)

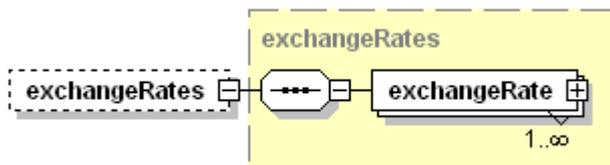
properties
isRef 0
minOcc 0
maxOcc 1
content complex

children [payment](#)

source `<xs:element name="formOfPayment" type="formOfPayment" minOccurs="0"/>`

element `economyType/exchangeRates`

diagram



type [exchangeRates](#)

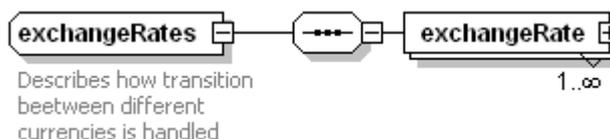
properties
 isRef 0
 minOccurs 0
 maxOccurs 1
 content complex

children [exchangeRate](#)

source `<xs:element name="exchangeRates" type="exchangeRates" minOccurs="0"/>`

complexType `exchangeRates`

diagram



children [exchangeRate](#)

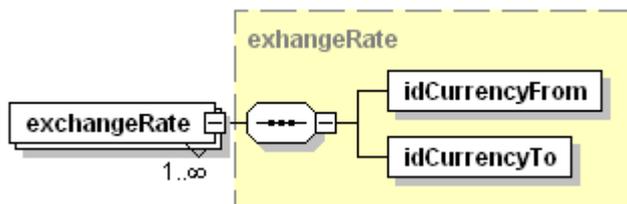
used by element [economyType/exchangeRates](#)

annotation
 documentation
 Describes how transition between different currencies is handled

source `<xs:complexType name="exchangeRates">
 <xs:annotation>
 <xs:documentation>Describes how transition between different currencies is
 handled</xs:documentation>
 </xs:annotation>
 <xs:sequence>
 <xs:element name="exchangeRate" type="exchangeRate" maxOccurs="unbounded"/>
 </xs:sequence>
 </xs:complexType>`

element `exchangeRates/exchangeRate`

diagram



type [exchangeRate](#)

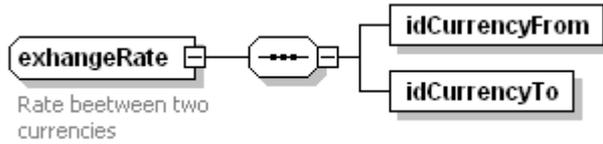
properties
 isRef 0
 minOccurs 1
 maxOccurs unbounded
 content complex

children [idCurrencyFrom](#) [idCurrencyTo](#)

attributes	Name	Type	Use	Default	Fixed	Annotation
	exchangeRate	xs:float	required			
source	<code><xs:element name="exchangeRate" type="exchangeRate" maxOccurs="unbounded"/></code>					

complexType exchangeRate

diagram



children [idCurrencyFrom](#) [idCurrencyTo](#)

used by element [exchangeRates/exchangeRate](#)

attributes	Name	Type	Use	Default	Fixed	Annotation
	exchangeRate	xs:float	required			
annotation	documentation	Rate beetween two currencies				
source	<code><xs:complexType name="exchangeRate"> <xs:annotation> <xs:documentation>Rate beetween two currencies</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="idCurrencyFrom" type="idType"/> <xs:element name="idCurrencyTo" type="idType"/> </xs:sequence> <xs:attribute name="exchangeRate" type="xs:float" use="required"/> </xs:complexType></code>					

element exchangeRate/idCurrencyFrom

diagram



type [idType](#)

properties	isRef	0	content	complex	attributes	Name	Type	Use	Default	Fixed	Annotation
						src	xs:string	required			
						id	xs:string	required			
						unique	xs:boolean	optional	false		
source	<code><xs:element name="idCurrencyFrom" type="idType"/></code>										

element exchangeRate/idCurrencyTo

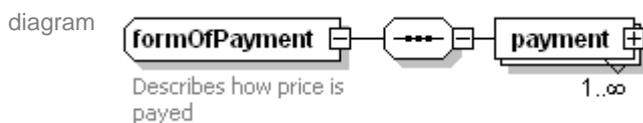
diagram



type [idType](#)

properties	isRef	0	content	complex	attributes	Name	Type	Use	Default	Fixed	Annotation
						src	xs:string	required			
						id	xs:string	required			
						unique	xs:boolean	optional	false		
source	<code><xs:element name="idCurrencyTo" type="idType"/></code>										

complexType formOfPayment



children [payment](#)

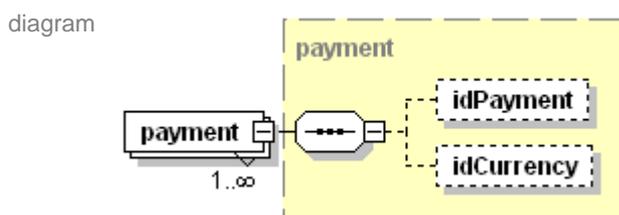
used by element [economyType/formOfPayment](#)

annotation
Describes how price is payed

```

source <xs:complexType name="formOfPayment">
  <xs:annotation>
    <xs:documentation>Describes how price is payed</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="payment" type="payment" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
    
```

element formOfPayment/payment



type [payment](#)

properties
 isRef 0
 minOcc 1
 maxOcc unbounded
 content complex

children [idPayment](#) [idCurrency](#)

attributes	Name	Type	Use	Default	Fixed	Annotation
	amount	xs:float	required			
	paymentType	xs:string	required			
	documentType	xs:string	optional			documentation Cash, Card, Account, Ticket, Document or SocialService Fee
	noOfDocuments	xs:nonNegativeInteger	optional			
	seqno	xs:positiveInteger	optional			

```

source <xs:element name="payment" type="payment" maxOccurs="unbounded"/>
    
```

complexType geographicLocation

diagram

geographicLocation

The coordinates for the address and potentially a known zone

used by elements [addressType/geographicLocation](#) [resourceType/vehiclestartLocation](#)

attributes	Name	Type	Use	Default	Fixed	Annotation
	zone	xs:string	optional			
	typeOfCoordinate	xs:string	required			
	lat	xs:float	required			
	long	xs:float	required			
	precision	xs:integer	required			

annotation documentation

The coordinates for the address and potentially a known zone

source `<xs:complexType name="geographicLocation">`

`<xs:annotation>`

`<xs:documentation>The coordinates for the address and potentially a known zone</xs:documentation>`

`</xs:annotation>`

`<xs:attribute name="zone" type="xs:string" use="optional"/>`

`<xs:attribute name="typeOfCoordinate" type="xs:string" use="required"/>`

`<xs:attribute name="lat" type="xs:float" use="required"/>`

`<xs:attribute name="long" type="xs:float" use="required"/>`

`<xs:attribute name="precision" type="xs:integer" use="required"/>`

`</xs:complexType>`

complexType idType

diagram

idType

General type identification-element

used by elements [addressType/idAddressName](#) [agreement/idAgreement](#) [attribute/idAttribute](#) [addressType/idCommunity](#) [content/idContent](#) [priceCalculation/idCurrency](#) [price/idCurrency](#) [payment/idCurrency](#) [exchangeRate/idCurrencyFrom](#) [exchangeRate/idCurrencyTo](#) [referencesTo/idDriver](#) [driver/idDriver](#) [referencesTo/idMsg](#) [msg/idMsg](#) [referencesTo/idOrder](#) [order/idOrder](#) [msg/orderLink/idOrder](#) [subOrderType/idOrder](#) [resourceType/idOrg](#) [orgType/idOrg](#) [payment/idPayment](#) [product/idProduct](#) [addressType/idStreet](#) [taxiMeter/idTaxa](#) [vehicle/idVehicle](#) [referencesTo/idVehicle](#)

attributes	Name	Type	Use	Default	Fixed	Annotation
	src	xs:string	required			
	id	xs:string	required			
	unique	xs:boolean	optional	false		

annotation documentation

General type identification-element

source `<xs:complexType name="idType">`

`<xs:annotation>`

`<xs:documentation>General type identification-element</xs:documentation>`

`</xs:annotation>`

`<xs:attribute name="src" type="xs:string" use="required"/>`

`<xs:attribute name="id" type="xs:string" use="required"/>`

`<xs:attribute name="unique" type="xs:boolean" use="optional" default="false"/>`

`</xs:complexType>`

complexType manualDescriptionType

diagram

manualDescriptionType

Any text aimed for operators, drivers etc

used by

elements [addressType/manualDescriptionAddress](#) [content/manualDescriptionContent](#)
[msg/manualDescriptionMsg](#) [resourceType/manualDescriptionResource](#)

attributes

Name	Type	Use	Default	Fixed	Annotation
manualText	xs:string	required			

annotation

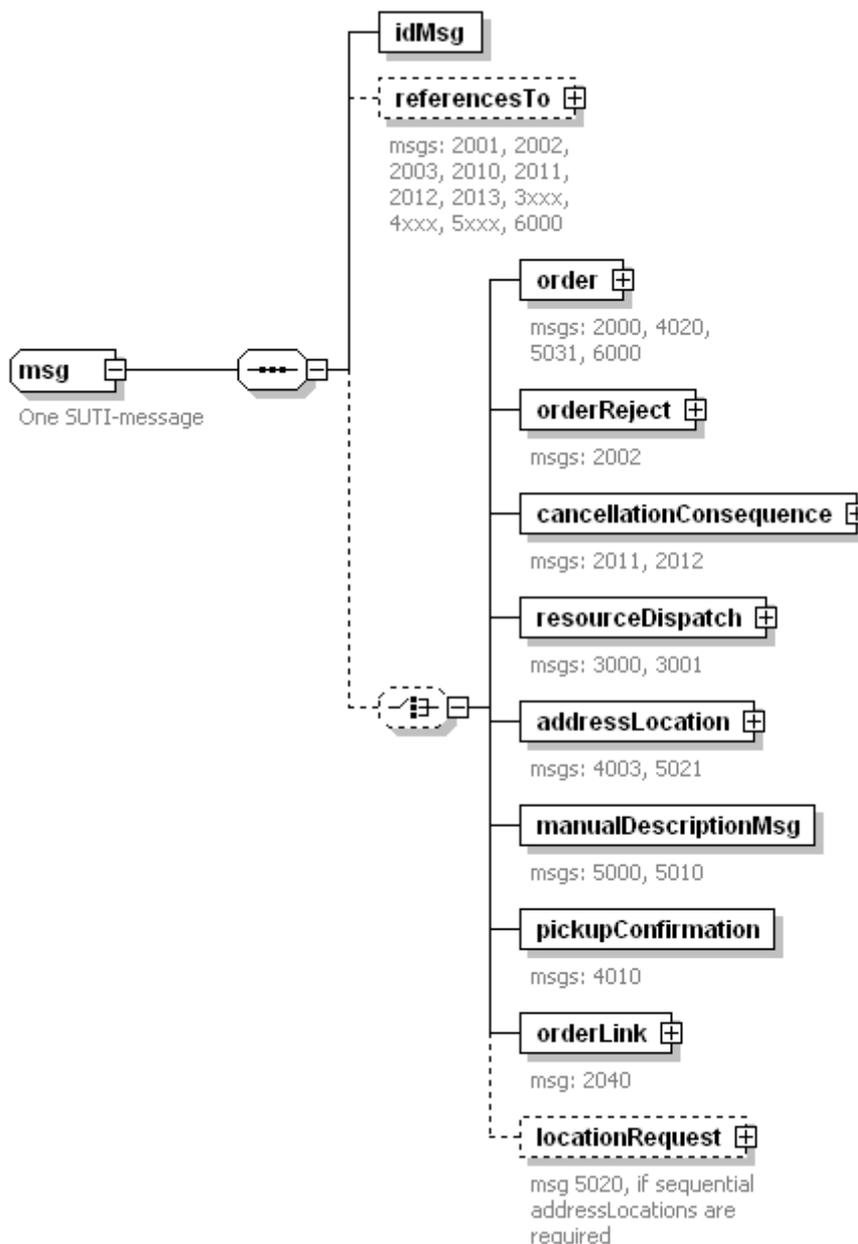
documentation
 Any text aimed for operators, drivers etc

source

```
<xs:complexType name="manualDescriptionType">
  <xs:annotation>
    <xs:documentation>Any text aimed for operators, drivers etc</xs:documentation>
  </xs:annotation>
  <xs:attribute name="manualText" type="xs:string" use="required"/>
</xs:complexType>
```

complexType msg

diagram



children [idMsg](#) [referencesTo](#) [order](#) [orderReject](#) [cancellationConsequence](#) [resourceDispatch](#) [addressLocation](#) [manualDescriptionMsg](#) [pickupConfirmation](#) [orderLink](#) [locationRequest](#)

used by element [SUTI/msg](#)

attributes	Name	Type	Use	Default	Fixed	Annotation
	msgType	xs:string	required			
	msgName	xs:string	optional			

annotation documentation One SUTI-message

```

source <xs:complexType name="msg">
  <xs:annotation>
    <xs:documentation>One SUTI-message</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="idMsg" type="idType"/>
    <xs:element name="referencesTo" type="referencesTo" minOccurs="0">

```

```

    <xs:annotation>
      <xs:documentation>msgs: 2001, 2002, 2003, 2010, 2011, 2012, 2013, 3xxx, 4xxx, 5xxx,
6000</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:choice minOccurs="0">
    <xs:element name="order" type="order">
      <xs:annotation>
        <xs:documentation>msgs: 2000, 4020, 5031, 6000</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="orderReject" type="orderReject">
      <xs:annotation>
        <xs:documentation>msgs: 2002</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="cancellationConsequence" type="cancellationConsequence">
      <xs:annotation>
        <xs:documentation>msgs: 2011, 2012</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="resourceDispatch" type="resourceType">
      <xs:annotation>
        <xs:documentation>msgs: 3000, 3001</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="addressLocation" type="addressType">
      <xs:annotation>
        <xs:documentation>msgs: 4003, 5021</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="manualDescriptionMsg" type="manualDescriptionType">
      <xs:annotation>
        <xs:documentation>msgs: 5000, 5010</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="pickupConfirmation">
      <xs:annotation>
        <xs:documentation>msgs: 4010</xs:documentation>
      </xs:annotation>
      <xs:complexType/>
    </xs:element>
    <xs:element name="orderLink">
      <xs:annotation>
        <xs:documentation>msg: 2040</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:sequence>
          <xs:element name="idOrder" type="idType" maxOccurs="unbounded">
            <xs:annotation>
              <xs:documentation>ids for the combined order</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="subOrderLink" type="subOrderType" maxOccurs="unbounded">
            <xs:annotation>
              <xs:documentation>ids for all included suborders</xs:documentation>
            </xs:annotation>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="locationRequest" minOccurs="0">
      <xs:annotation>
        <xs:documentation>msg 5020, if sequential addressLocations are required</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:sequence>

```

```

<xs:element name="timeFrom" type="time" minOccurs="0"/>
<xs:element name="timeTo" type="time" minOccurs="0"/>
<xs:element name="interval" minOccurs="0">
  <xs:complexType>
    <xs:attribute name="seconds" type="xs:integer" use="optional"/>
    <xs:attribute name="meter" type="xs:integer" use="optional"/>
  </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:choice>
</xs:sequence>
<xs:attribute name="msgType" type="xs:string" use="required"/>
<xs:attribute name="msgName" type="xs:string" use="optional"/>
</xs:complexType>
    
```

element msg/idMsg

diagram



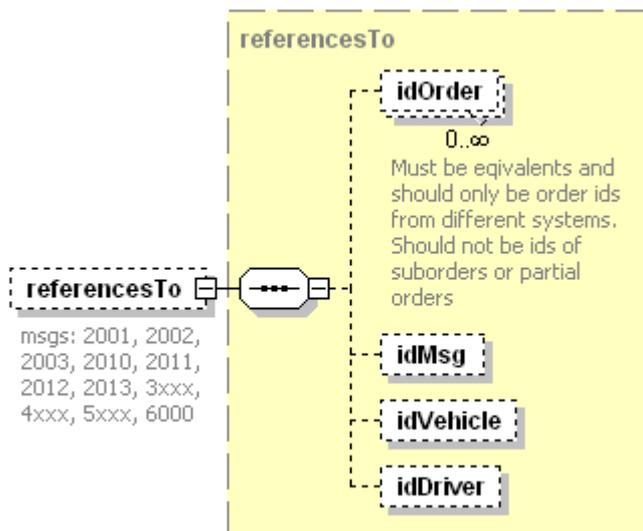
type [idType](#)

properties	isRef	0				
	content	complex				
attributes	Name	Type	Use	Default	Fixed	Annotation
	src	xs:string	required			
	id	xs:string	required			
	unique	xs:boolean	optional	false		

source `<xs:element name="idMsg" type="idType"/>`

element msg/referencesTo

diagram



type [referencesTo](#)

properties	isRef	0
	minOcc	0
	maxOcc	1
	content	complex

children [idOrder](#) [idMsg](#) [idVehicle](#) [idDriver](#)

annotation documentation
 msgs: 2001, 2002, 2003, 2010, 2011, 2012, 2013, 3xxx, 4xxx, 5xxx, 6000

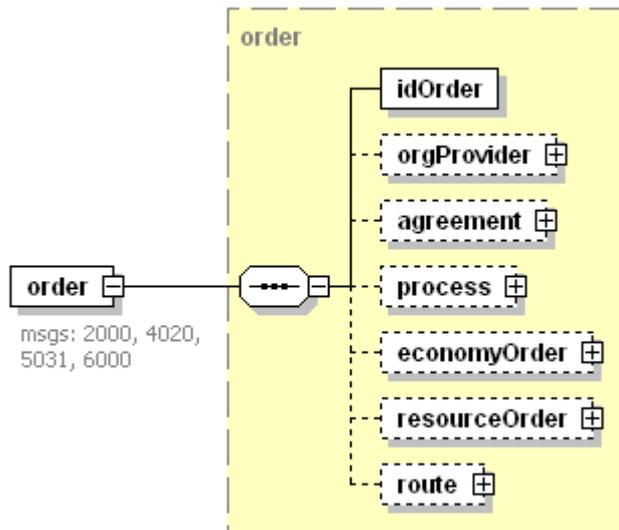
```

source <xs:element name="referencesTo" type="referencesTo" minOccurs="0">
  <xs:annotation>
    <xs:documentation>msgs: 2001, 2002, 2003, 2010, 2011, 2012, 2013, 3xxx, 4xxx, 5xxx,
    6000</xs:documentation>
  </xs:annotation>
</xs:element>

```

element msg/order

diagram



type [order](#)

properties isRef 0
content complex

children [idOrder](#) [orgProvider](#) [agreement](#) [process](#) [economyOrder](#) [resourceOrder](#) [route](#)

annotation documentation
msgs: 2000, 4020, 5031, 6000

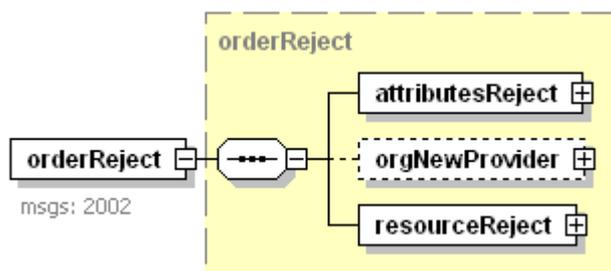
```

source <xs:element name="order" type="order">
  <xs:annotation>
    <xs:documentation>msgs: 2000, 4020, 5031, 6000</xs:documentation>
  </xs:annotation>
</xs:element>

```

element msg/orderReject

diagram



type [orderReject](#)

properties isRef 0
content complex

children [attributesReject](#) [orgNewProvider](#) [resourceReject](#)

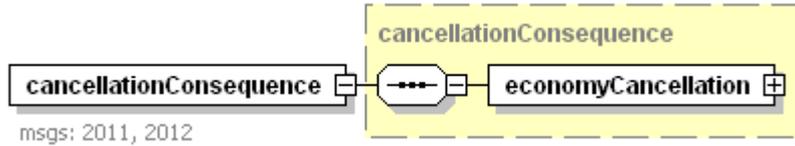
annotation documentation
msgs: 2002

```

source <xs:element name="orderReject" type="orderReject">
  <xs:annotation>
    <xs:documentation>msgs: 2002</xs:documentation>
  </xs:annotation>
</xs:element>
    
```

element msg/cancellationConsequence

diagram



type [cancellationConsequence](#)

properties isRef 0
content complex

children [economyCancellation](#)

attributes	Name	Type	Use	Default	Fixed	Annotation
	cancellationAc	xs:boolean	required			
	ceptance					
	cancellationCo	xs:boolean	required			
	nsequence					

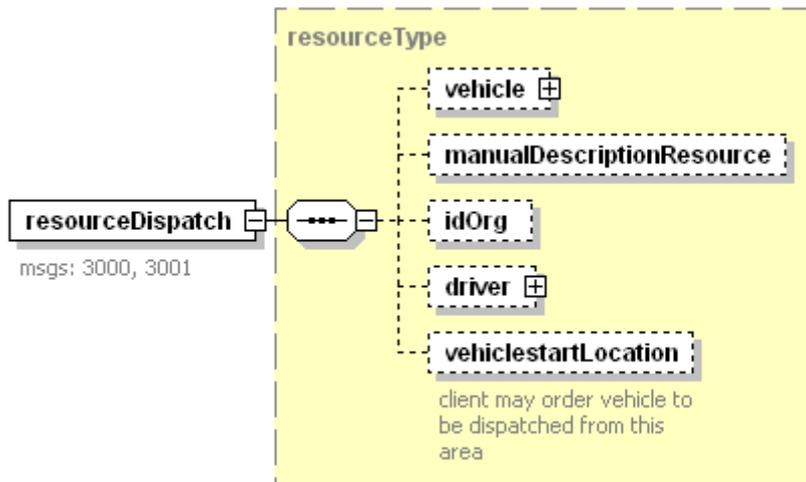
documentation
msgs: 2011, 2012

```

source <xs:element name="cancellationConsequence" type="cancellationConsequence">
  <xs:annotation>
    <xs:documentation>msgs: 2011, 2012</xs:documentation>
  </xs:annotation>
</xs:element>
    
```

element msg/resourceDispatch

diagram



type [resourceType](#)

properties isRef 0
content complex

children [vehicle](#) [manualDescriptionResource](#) [idOrg](#) [driver](#) [vehiclestartLocation](#)

documentation
msgs: 3000, 3001

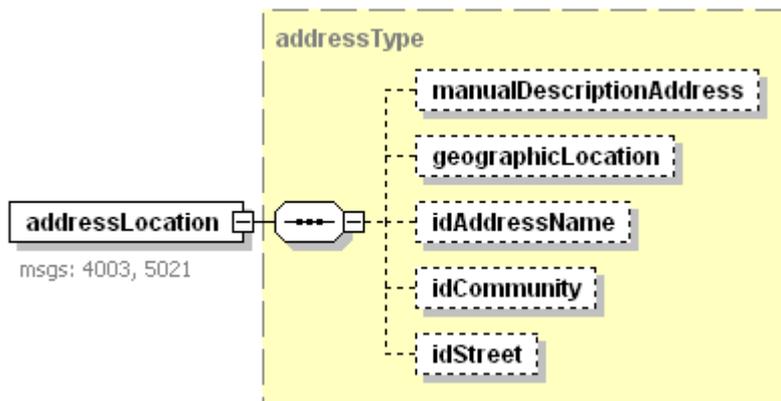
```

source <xs:element name="resourceDispatch" type="resourceType">
  <xs:annotation>
    <xs:documentation>msgs: 3000, 3001</xs:documentation>
  </xs:annotation>
</xs:element>
    
```

</xs:annotation>
</xs:element>

element msg/addressLocation

diagram



type [addressType](#)

properties isRef 0
content complex

children [manualDescriptionAddress](#) [geographicLocation](#) [idAddressName](#) [idCommunity](#) [idStreet](#)

attributes	Name	Type	Use	Default	Fixed	Annotation
	addressName	xs:string	optional			
	street	xs:string	optional			
	streetNo	xs:positiveInteger	optional			
	streetNoLetter	xs:string	optional			
	location	xs:string	optional			
	community	xs:string	optional			
	postalNo	xs:string	optional			
	country	xs:string	optional			

documentation
msgs: 4003, 5021

source <xs:element name="addressLocation" type="addressType">
<xs:annotation>
<xs:documentation>msgs: 4003, 5021</xs:documentation>
</xs:annotation>
</xs:element>

element msg/manualDescriptionMsg

diagram



type [manualDescriptionType](#)

properties isRef 0
content complex

attributes	Name	Type	Use	Default	Fixed	Annotation
	manualText	xs:string	required			

documentation
msgs: 5000, 5010

source <xs:element name="manualDescriptionMsg" type="manualDescriptionType">
<xs:annotation>
<xs:documentation>msgs: 5000, 5010</xs:documentation>
</xs:annotation>
</xs:element>

element msg/pickupConfirmation

diagram

pickupConfirmation

msgs: 4010

properties

isRef 0
content complex

annotation

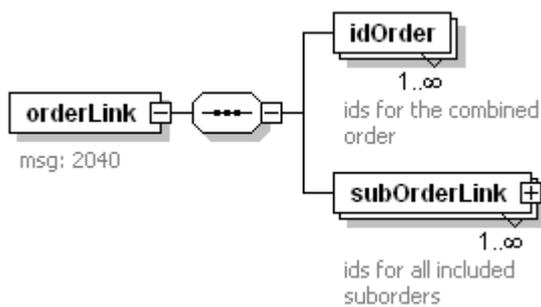
documentation
msgs: 4010

source

```
<xs:element name="pickupConfirmation">
  <xs:annotation>
    <xs:documentation>msgs: 4010</xs:documentation>
  </xs:annotation>
  <xs:complexType/>
</xs:element>
```

element msg/orderLink

diagram



properties

isRef 0
content complex

children

[idOrder](#) [subOrderLink](#)

annotation

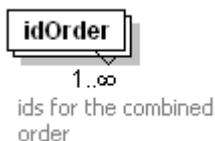
documentation
msg: 2040

source

```
<xs:element name="orderLink">
  <xs:annotation>
    <xs:documentation>msg: 2040</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="idOrder" type="idType" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>ids for the combined order</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="subOrderLink" type="subOrderType" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>ids for all included suborders</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

element msg/orderLink/idOrder

diagram



type [idType](#)

properties
 isRef 0
 minOcc 1
 maxOcc unbounded
 content complex

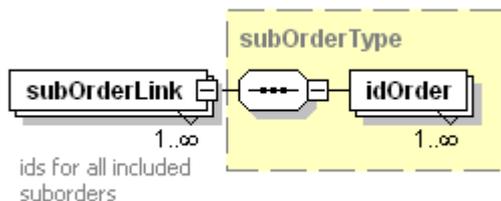
attributes	Name	Type	Use	Default	Fixed	Annotation
	src	xs:string	required			
	id	xs:string	required			
	unique	xs:boolean	optional	false		

annotation
 documentation
 ids for the combined order

```
<xs:element name="idOrder" type="idType" maxOccurs="unbounded">
  <xs:annotation>
    <xs:documentation>ids for the combined order</xs:documentation>
  </xs:annotation>
</xs:element>
```

element msg/orderLink/subOrderLink

diagram



type [subOrderType](#)

properties
 isRef 0
 minOcc 1
 maxOcc unbounded
 content complex

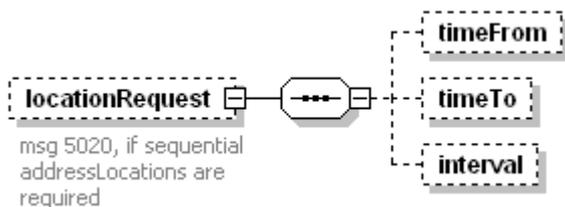
children [idOrder](#)

annotation
 documentation
 ids for all included suborders

```
<xs:element name="subOrderLink" type="subOrderType" maxOccurs="unbounded">
  <xs:annotation>
    <xs:documentation>ids for all included suborders</xs:documentation>
  </xs:annotation>
</xs:element>
```

element msg/locationRequest

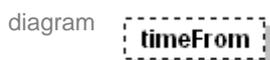
diagram



```

properties      isRef 0
                minOcc 0
                maxOcc 1
                content complex
children timeFrom timeTo interval
annotation documentation
                msg 5020, if sequential addressLocations are required
source <xs:element name="locationRequest" minOccurs="0">
      <xs:annotation>
        <xs:documentation>msg 5020, if sequential addressLocations are required</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:sequence>
          <xs:element name="timeFrom" type="time" minOccurs="0"/>
          <xs:element name="timeTo" type="time" minOccurs="0"/>
          <xs:element name="interval" minOccurs="0">
            <xs:complexType>
              <xs:attribute name="seconds" type="xs:integer" use="optional"/>
              <xs:attribute name="meter" type="xs:integer" use="optional"/>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  
```

element msg/locationRequest/timeFrom



type [time](#)

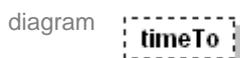
```

properties      isRef 0
                minOcc 0
                maxOcc 1
                content complex
  
```

attributes	Name	Type	Use	Default	Fixed	Annotation
	timeAccuracy	xs:string	required			
	timeZone	xs:integer	required			
	time	xs:dateTime	required			

```
source <xs:element name="timeFrom" type="time" minOccurs="0"/>
```

element msg/locationRequest/timeTo



type [time](#)

```

properties      isRef 0
                minOcc 0
                maxOcc 1
                content complex
  
```

attributes	Name	Type	Use	Default	Fixed	Annotation
	timeAccuracy	xs:string	required			
	timeZone	xs:integer	required			
	time	xs:dateTime	required			

```
source <xs:element name="timeTo" type="time" minOccurs="0"/>
```

element msg/locationRequest/interval

diagram						
properties	isRef	0				
	minOcc	0				
	maxOcc	1				
	content	complex				
attributes	Name	Type	Use	Default	Fixed	Annotation
	seconds	xs:integer	optional			
	meter	xs:integer	optional			
source	<pre> <xs:element name="interval" minOccurs="0"> <xs:complexType> <xs:attribute name="seconds" type="xs:integer" use="optional"/> <xs:attribute name="meter" type="xs:integer" use="optional"/> </xs:complexType> </xs:element> </pre>					

complexType multiDispatch

diagram						
	Presence of this element in order indicates that the order may have been sent to more than one provider					
used by	element	process/multiDispatch				
attributes	Name	Type	Use	Default	Fixed	Annotation
	time	xs:dateTime	required			
annotation	documentation					
	Presence of this element in order indicates that the order may have been sent to more than one provider					
source	<pre> <xs:complexType name="multiDispatch"> <xs:annotation> <xs:documentation>Presence of this element in order indicates that the order may have been sent to more than one provider</xs:documentation> </xs:annotation> <xs:attribute name="time" type="xs:dateTime" use="required"/> </xs:complexType> </pre>					

complexType node

diagram						
	Description of the node. Point is a synonym for node. The node can be either of the types pickup, destination, navigation or action.					
children	addressNode timesNode contents					
used by	elements	route/node pickupConfirmation/node				
attributes	Name	Type	Use	Default	Fixed	Annotation
	nodeSeqno	xs:positiveInteger	required			

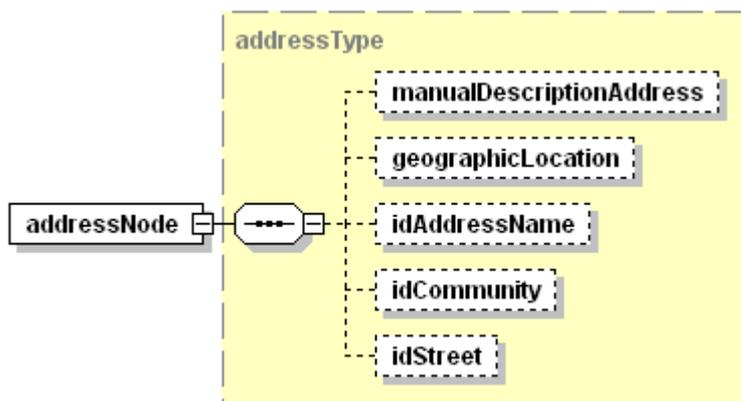
nodeType **xs:string** required

annotation documentation
Description of the node. Point is a synonym for node. The node can be either of the types pickup, destination, navigation or action.

source `<xs:complexType name="node">`
`<xs:annotation>`
`<xs:documentation>Description of the node. Point is a synonym for node. The node can be either of the`
`types pickup, destination, navigation or action.</xs:documentation>`
`</xs:annotation>`
`<xs:sequence>`
`<xs:element name="addressNode" type="addressType"/>`
`<xs:element name="timesNode" type="timesType"/>`
`<xs:element name="contents" type="contents"/>`
`</xs:sequence>`
`<xs:attribute name="nodeSeqno" type="xs:positiveInteger" use="required"/>`
`<xs:attribute name="nodeType" type="xs:string" use="required"/>`
`</xs:complexType>`

element node/addressNode

diagram



type [addressType](#)

properties isRef 0
content complex

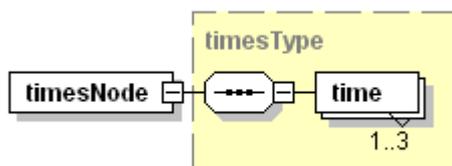
children [manualDescriptionAddress](#) [geographicLocation](#) [idAddressName](#) [idCommunity](#) [idStreet](#)

attributes	Name	Type	Use	Default	Fixed	Annotation
	addressName	xs:string	optional			
	street	xs:string	optional			
	streetNo	xs:positiveInteger	optional			
	streetNoLetter	xs:string	optional			
	location	xs:string	optional			
	community	xs:string	optional			
	postalNo	xs:string	optional			
	country	xs:string	optional			

source `<xs:element name="addressNode" type="addressType"/>`

element node/timesNode

diagram

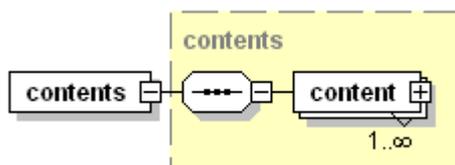


type [timesType](#)

properties isRef 0
content complex
children [time](#)
source `<xs:element name="timesNode" type="timesType"/>`

element node/contents

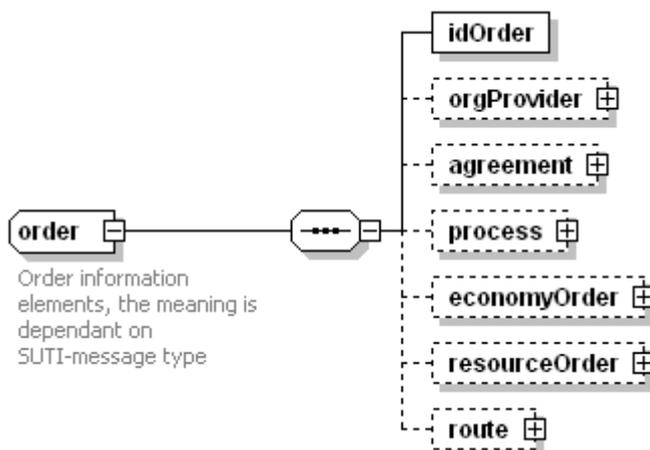
diagram



type [contents](#)
properties isRef 0
content complex
children [content](#)
source `<xs:element name="contents" type="contents"/>`

complexType order

diagram

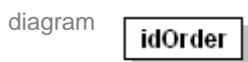


children [idOrder](#) [orgProvider](#) [agreement](#) [process](#) [economyOrder](#) [resourceOrder](#) [route](#)
used by element [msg/order](#)
annotation documentation
Order information elements, the meaning is dependant on SUTI-message type
source `<xs:complexType name="order">
<xs:annotation>
<xs:documentation>Order information elements, the meaning is dependant on SUTI-message type</xs:documentation>
</xs:annotation>
<xs:sequence>
<xs:element name="idOrder" type="idType"/>
<xs:element name="orgProvider" type="orgType" minOccurs="0"/>
<xs:element name="agreement" type="agreement" minOccurs="0"/>
<xs:element name="process" minOccurs="0">
<xs:complexType>
<xs:complexContent>
<xs:extension base="process"/>
</xs:complexContent>
</xs:complexType>
</xs:sequence>
</xs:complexType>`

```

<xs:element name="economyOrder" type="economyType" minOccurs="0"/>
<xs:element name="resourceOrder" type="resourceType" minOccurs="0"/>
<xs:element name="route" type="route" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
    
```

element order/idOrder



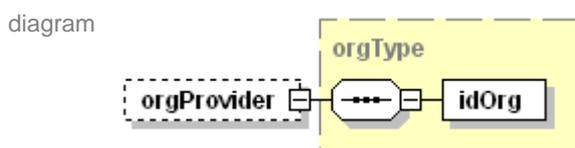
type [idType](#)

properties isRef 0
content complex

attributes	Name	Type	Use	Default	Fixed	Annotation
src		xs:string	required			
id		xs:string	required			
unique		xs:boolean	optional	false		

source `<xs:element name="idOrder" type="idType"/>`

element order/orgProvider



type [orgType](#)

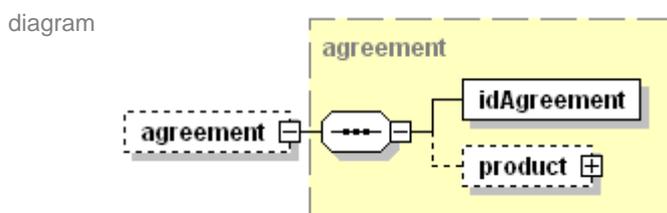
properties isRef 0
minOcc 0
maxOcc 1
content complex

children [idOrg](#)

attributes	Name	Type	Use	Default	Fixed	Annotation
name		xs:string	required			

source `<xs:element name="orgProvider" type="orgType" minOccurs="0"/>`

element order/agreement



type [agreement](#)

properties isRef 0
minOcc 0
maxOcc 1
content complex

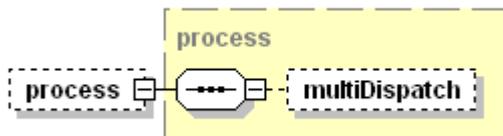
children [idAgreement](#) [product](#)

attributes	Name	Type	Use	Default	Fixed	Annotation
name		xs:string	optional			

source `<xs:element name="agreement" type="agreement" minOccurs="0"/>`

element **order/process**

diagram



type extension of [process](#)

properties
 isRef 0
 minOcc 0
 maxOcc 1
 content complex

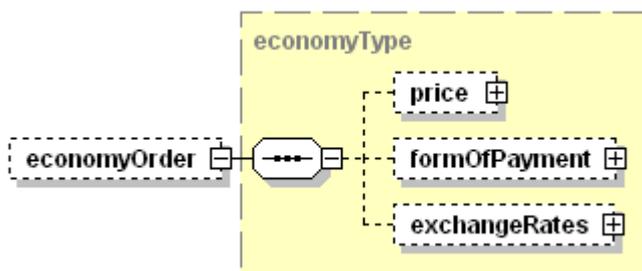
children [multiDispatch](#)

attributes	Name	Type	Use	Default	Fixed	Annotation
	dispatch	xs:boolean	required			
	dispatchResponsible	xs:string	required			
	trafficControl	xs:boolean	required			
	report	xs:boolean	required			
	preorderedVehicle	xs:boolean	required			
	allowRouting	xs:boolean	required			
	automaticStatus	xs:boolean	optional			
	statusDistance	xs:nonNegativeInteger	optional			

source `<xs:element name="process" minOccurs="0">
 <xs:complexType>
 <xs:complexContent>
 <xs:extension base="process"/>
 </xs:complexContent>
 </xs:complexType>
 </xs:element>`

element **order/economyOrder**

diagram



type [economyType](#)

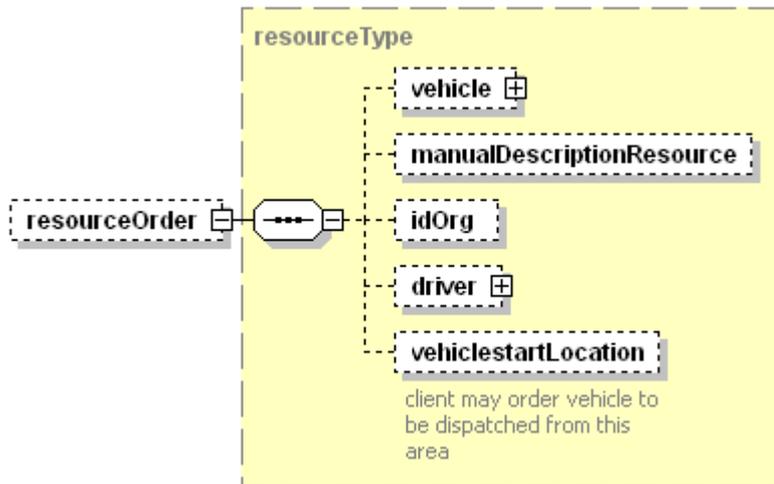
properties
 isRef 0
 minOcc 0
 maxOcc 1
 content complex

children [price](#) [formOfPayment](#) [exchangeRates](#)

source `<xs:element name="economyOrder" type="economyType" minOccurs="0"/>`

element `order/resourceOrder`

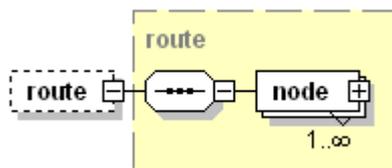
diagram



type [resourceType](#)
 properties isRef 0
 minOcc 0
 maxOcc 1
 content complex
 children [vehicle](#) [manualDescriptionResource](#) [idOrg](#) [driver](#) [vehiclestartLocation](#)
 source `<xs:element name="resourceOrder" type="resourceType" minOccurs="0"/>`

element `order/route`

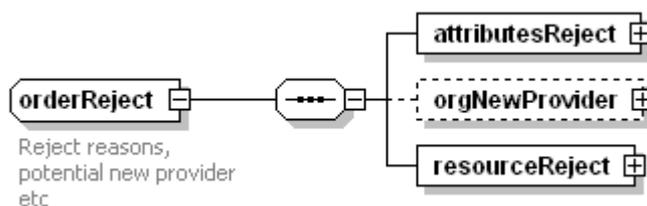
diagram



type [route](#)
 properties isRef 0
 minOcc 0
 maxOcc 1
 content complex
 children [node](#)
 source `<xs:element name="route" type="route" minOccurs="0"/>`

complexType `orderReject`

diagram



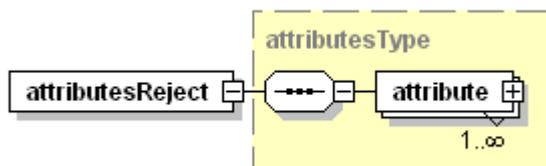
children [attributesReject](#) [orgNewProvider](#) [resourceReject](#)
 used by element [msg/orderReject](#)

```

annotation documentation
Reject reasons, potential new provider etc
source <xs:complexType name="orderReject">
  <xs:annotation>
    <xs:documentation>Reject reasons, potential new provider etc</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="attributesReject" type="attributesType"/>
    <xs:element name="orgNewProvider" type="orgType" minOccurs="0"/>
    <xs:element name="resourceReject" type="resourceType"/>
  </xs:sequence>
</xs:complexType>
    
```

element orderReject/attributesReject

diagram

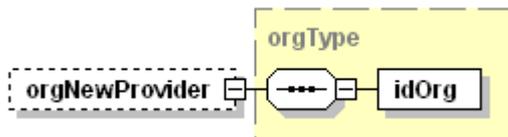


```

type attributesType
properties isRef 0
           content complex
children attribute
source <xs:element name="attributesReject" type="attributesType"/>
    
```

element orderReject/orgNewProvider

diagram

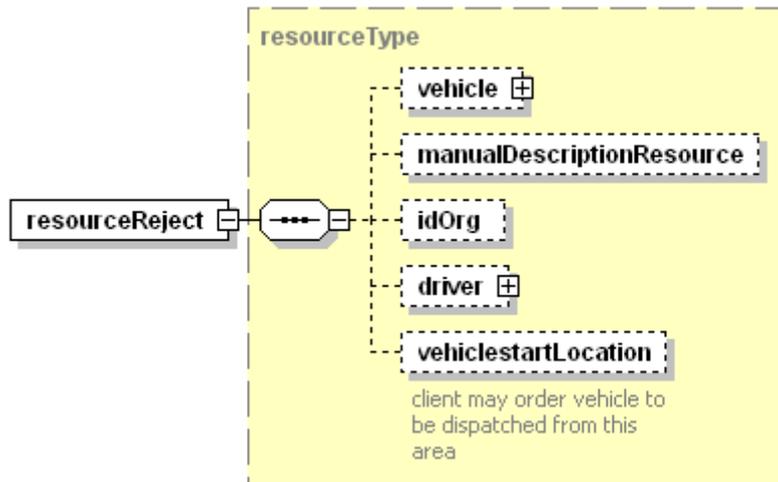


```

type orgType
properties isRef 0
           minOcc 0
           maxOcc 1
           content complex
children idOrg
attributes
  Name      Type      Use      Default      Fixed      Annotation
  name      xs:string  required
source <xs:element name="orgNewProvider" type="orgType" minOccurs="0"/>
    
```

element `orderReject/resourceReject`

diagram



type [resourceType](#)

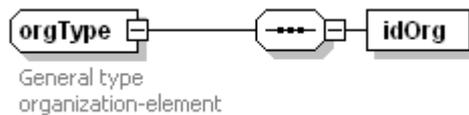
properties isRef 0
content complex

children [vehicle](#) [manualDescriptionResource](#) [idOrg](#) [driver](#) [vehiclestartLocation](#)

source `<xs:element name="resourceReject" type="resourceType"/>`

complexType `orgType`

diagram



children [idOrg](#)

used by elements [orderReject/orgNewProvider](#) [order/orgProvider](#) [SUTI/orgReceiver](#)
[associatedReservation/orgReservation](#) [SUTI/orgSender](#)

attributes	Name	Type	Use	Default	Fixed	Annotation
	name	xs:string	required			

annotation documentation
General type organization-element

source `<xs:complexType name="orgType">
<xs:annotation>
<xs:documentation>General type organization-element</xs:documentation>
</xs:annotation>
<xs:sequence>
<xs:element name="idOrg" type="idType"/>
</xs:sequence>
<xs:attribute name="name" type="xs:string" use="required"/>
</xs:complexType>`

element `orgType/idOrg`

diagram



type [idType](#)

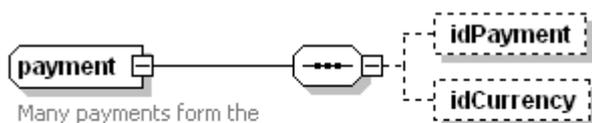
properties isRef 0
content complex

attributes	Name	Type	Use	Default	Fixed	Annotation
	src	xs:string	required			
	id	xs:string	required			
	unique	xs:boolean	optional	false		

source `<xs:element name="idOrg" type="idType"/>`

complexType **payment**

diagram



Many payments form the formOfPayment. The types of payment can be Cash, Card, Account, Ticke, Document or SocialServiceFee

children [idPayment](#) [idCurrency](#)

used by element [formOfPayment/payment](#)

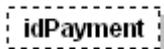
attributes	Name	Type	Use	Default	Fixed	Annotation
	amount	xs:float	required			
	paymentType	xs:string	required			documentation Cash, Card, Account, Ticket, Document or SocialService Fee
	documentType	xs:string	optional			
	noOfDocuments	xs:nonNegativeInteger	optional			
	seqno	xs:positiveInteger	optional			

annotation documentation
Many payments form the formOfPayment. The types of payment can be Cash, Card, Account, Ticke, Document or SocialServiceFee

source `<xs:complexType name="payment">
 <xs:annotation>
 <xs:documentation>Many payments form the formOfPayment. The types of payment can be Cash, Card, Account, Ticke, Document or SocialServiceFee</xs:documentation>
 </xs:annotation>
 <xs:sequence>
 <xs:element name="idPayment" type="idType" minOccurs="0"/>
 <xs:element name="idCurrency" type="idType" minOccurs="0"/>
 </xs:sequence>
 <xs:attribute name="amount" type="xs:float" use="required"/>
 <xs:attribute name="paymentType" type="xs:string" use="required">
 <xs:annotation>
 <xs:documentation>Cash, Card, Account, Ticket, Document or SocialServiceFee</xs:documentation>
 </xs:annotation>
 </xs:attribute>
 <xs:attribute name="documentType" type="xs:string" use="optional"/>
 <xs:attribute name="noOfDocuments" type="xs:nonNegativeInteger" use="optional"/>
 <xs:attribute name="seqno" type="xs:positiveInteger" use="optional"/>
 </xs:complexType>`

element `payment/idPayment`

diagram



type [idType](#)

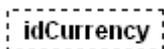
properties
 isRef 0
 minOcc 0
 maxOcc 1
 content complex

attributes	Name	Type	Use	Default	Fixed	Annotation
	src	xs:string	required			
	id	xs:string	required			
	unique	xs:boolean	optional	false		

source `<xs:element name="idPayment" type="idType" minOccurs="0"/>`

element `payment/idCurrency`

diagram



type [idType](#)

properties
 isRef 0
 minOcc 0
 maxOcc 1
 content complex

attributes	Name	Type	Use	Default	Fixed	Annotation
	src	xs:string	required			
	id	xs:string	required			
	unique	xs:boolean	optional	false		

source `<xs:element name="idCurrency" type="idType" minOccurs="0"/>`

complexType `pickupConfirmation`

diagram



Node and content

children [node](#)

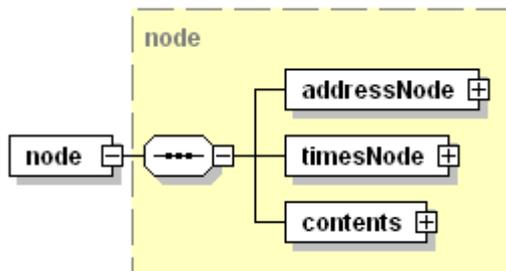
attributes	Name	Type	Use	Default	Fixed	Annotation
	pickupConfirmation	xs:boolean	required			

annotation
 documentation
 Node and content

source `<xs:complexType name="pickupConfirmation">
 <xs:annotation>
 <xs:documentation>Node and content</xs:documentation>
 </xs:annotation>
 <xs:sequence>
 <xs:element name="node" type="node"/>
 </xs:sequence>
 <xs:attribute name="pickupConfirmation" type="xs:boolean" use="required"/>
 </xs:complexType>`

element pickupConfirmation/node

diagram



type [node](#)

properties isRef 0
content complex

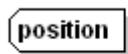
children [addressNode](#) [timesNode](#) [contents](#)

attributes	Name	Type	Use	Default	Fixed	Annotation
	nodeSeqno	xs:positiveInteger	required			
	nodeType	xs:string	required			

source `<xs:element name="node" type="node"/>`

complexType position

diagram



The characteristics of a seat position

used by element [seats/position](#)

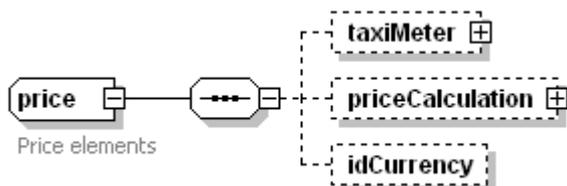
attributes	Name	Type	Use	Default	Fixed	Annotation
	row	xs:positiveInteger	required			
	seat	xs:string	optional			
	direction	xs:string	optional			
	legSpace	xs:string	optional			
	access	xs:nonNegativeInteger	optional			

annotation documentation
The characteristics of a seat position

source `<xs:complexType name="position">
<xs:annotation>
<xs:documentation>The characteristics of a seat position</xs:documentation>
</xs:annotation>
<xs:attribute name="row" type="xs:positiveInteger" use="required"/>
<xs:attribute name="seat" type="xs:string" use="optional"/>
<xs:attribute name="direction" type="xs:string" use="optional"/>
<xs:attribute name="legSpace" type="xs:string" use="optional"/>
<xs:attribute name="access" type="xs:nonNegativeInteger" use="optional"/>
</xs:complexType>`

complexType price

diagram



children [taxiMeter](#) [priceCalculation](#) [idCurrency](#)

used by element [economyType/price](#)

attributes	Name	Type	Use	Default	Fixed	Annotation
	priceResponsible	xs:string	required			
	fixedPrice	xs:boolean	required			
	vatIncluded	xs:boolean	required			
	vatPercent	xs:float	optional			
	price	xs:float	optional			
	vat	xs:float	optional			

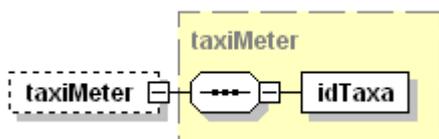
annotation documentation Price elements

```

source <xs:complexType name="price">
  <xs:annotation>
    <xs:documentation>Price elements</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="taxiMeter" type="taxiMeter" minOccurs="0"/>
    <xs:element name="priceCalculation" type="priceCalculation" minOccurs="0"/>
    <xs:element name="idCurrency" type="idType" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="priceResponsible" type="xs:string" use="required"/>
  <xs:attribute name="fixedPrice" type="xs:boolean" use="required"/>
  <xs:attribute name="vatIncluded" type="xs:boolean" use="required"/>
  <xs:attribute name="vatPercent" type="xs:float" use="optional"/>
  <xs:attribute name="price" type="xs:float" use="optional"/>
  <xs:attribute name="vat" type="xs:float" use="optional"/>
</xs:complexType>
    
```

element price/taxiMeter

diagram



type [taxiMeter](#)

properties isRef 0
minOcc 0
maxOcc 1
content complex

children [idTaxa](#)

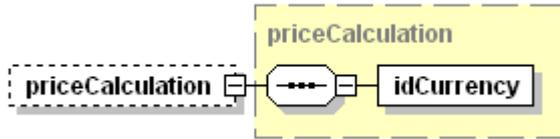
attributes	Name	Type	Use	Default	Fixed	Annotation
	activate	xs:boolean	required			

```

source <xs:element name="taxiMeter" type="taxiMeter" minOccurs="0"/>
    
```

element price/priceCalculation

diagram



type [priceCalculation](#)

properties
 isRef 0
 minOcc 0
 maxOcc 1
 content complex

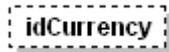
children [idCurrency](#)

attributes	Name	Type	Use	Default	Fixed	Annotation
	vatIncluded	xs:boolean	required			
	vatPercent	xs:float	optional			
	startFee	xs:float	required			
	kmFee	xs:float	required			
	minuteFee	xs:float	required			
	formula	xs:string	required			

source `<xs:element name="priceCalculation" type="priceCalculation" minOccurs="0"/>`

element price/idCurrency

diagram



type [idType](#)

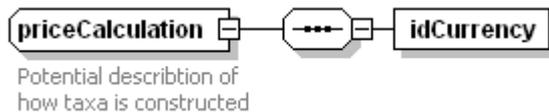
properties
 isRef 0
 minOcc 0
 maxOcc 1
 content complex

attributes	Name	Type	Use	Default	Fixed	Annotation
	src	xs:string	required			
	id	xs:string	required			
	unique	xs:boolean	optional	false		

source `<xs:element name="idCurrency" type="idType" minOccurs="0"/>`

complexType priceCalculation

diagram



children [idCurrency](#)

used by element [price/priceCalculation](#)

attributes	Name	Type	Use	Default	Fixed	Annotation
	vatIncluded	xs:boolean	required			
	vatPercent	xs:float	optional			
	startFee	xs:float	required			
	kmFee	xs:float	required			
	minuteFee	xs:float	required			
	formula	xs:string	required			

annotation
 documentation
 Potential description of how taxa is constructed

source `<xs:complexType name="priceCalculation">`

```

<xs:annotation>
  <xs:documentation>Potential description of how taxa is constructed</xs:documentation>
</xs:annotation>
<xs:sequence>
  <xs:element name="idCurrency" type="idType"/>
</xs:sequence>
<xs:attribute name="vatIncluded" type="xs:boolean" use="required"/>
<xs:attribute name="vatPercent" type="xs:float" use="optional"/>
<xs:attribute name="startFee" type="xs:float" use="required"/>
<xs:attribute name="kmFee" type="xs:float" use="required"/>
<xs:attribute name="minuteFee" type="xs:float" use="required"/>
<xs:attribute name="formula" type="xs:string" use="required"/>
</xs:complexType>

```

element priceCalculation/idCurrency

diagram



type [idType](#)

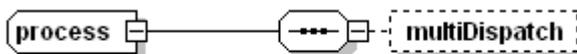
properties isRef 0
content complex

attributes	Name	Type	Use	Default	Fixed	Annotation
	src	xs:string	required			
	id	xs:string	required			
	unique	xs:boolean	optional	false		

source `<xs:element name="idCurrency" type="idType"/>`

complexType process

diagram



When order is sent this element describes how the client and provider will process order. Required only in first order.

children [multiDispatch](#)

used by element [order/process](#)

attributes	Name	Type	Use	Default	Fixed	Annotation
	dispatch	xs:boolean	required			
	dispatchResponsible	xs:string	required			
	trafficControl	xs:boolean	required			
	report	xs:boolean	required			
	preorderedVehicle	xs:boolean	required			
	allowRouting	xs:boolean	required			
	automaticStatus	xs:boolean	optional			
	statusDistance	xs:nonNegativeInteger	optional			

annotation documentation
When order is sent this element describes how the client and provider will process order. Required only in first order.

source `<xs:complexType name="process">`

```

<xs:annotation>
  <xs:documentation>When order is sent this element describes how the client and provider will process order. Required only in first order.</xs:documentation>
</xs:annotation>

```

```

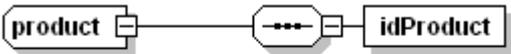
<xs:sequence>
  <xs:element name="multiDispatch" type="multiDispatch" minOccurs="0"/>
</xs:sequence>
<xs:attribute name="dispatch" type="xs:boolean" use="required"/>
<xs:attribute name="dispatchResponsible" type="xs:string" use="required"/>
<xs:attribute name="trafficControl" type="xs:boolean" use="required"/>
<xs:attribute name="report" type="xs:boolean" use="required"/>
<xs:attribute name="preorderedVehicle" type="xs:boolean" use="required"/>
<xs:attribute name="allowRouting" type="xs:boolean" use="required"/>
<xs:attribute name="automaticStatus" type="xs:boolean" use="optional"/>
<xs:attribute name="statusDistance" type="xs:nonNegativeInteger" use="optional"/>
</xs:complexType>

```

element process/multiDispatch

diagram						
type	multiDispatch					
properties	isRef	0				
	minOcc	0				
	maxOcc	1				
	content	complex				
attributes	Name	Type	Use	Default	Fixed	Annotation
	time	xs:dateTime	required			
source	<code><xs:element name="multiDispatch" type="multiDispatch" minOccurs="0"/></code>					

complexType product

diagram						
	Refers to a product description/definition within the agreement					
children	idProduct					
used by	element	agreement/product				
attributes	Name	Type	Use	Default	Fixed	Annotation
	name	xs:string	optional			
annotation	documentation Refers to a product description/definition within the agreement					
source	<pre> <xs:complexType name="product"> <xs:annotation> <xs:documentation>Refers to a product description/definition within the agreement</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="idProduct" type="idType"/> </xs:sequence> <xs:attribute name="name" type="xs:string" use="optional"/> </xs:complexType> </pre>					

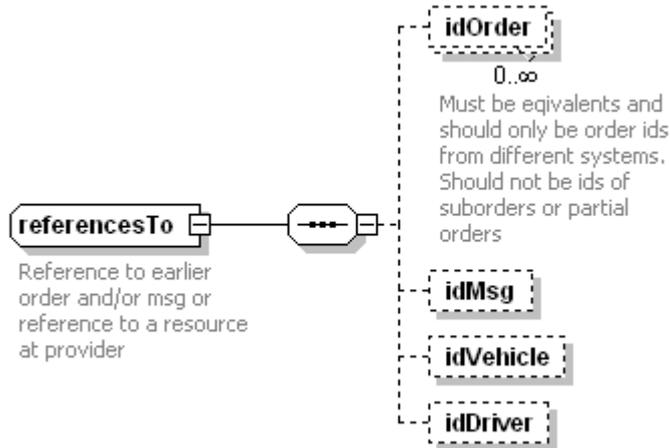
element product/idProduct

diagram		
type	idType	
properties	isRef	0
	content	complex

attributes	Name	Type	Use	Default	Fixed	Annotation
	src	xs:string	required			
	id	xs:string	required			
	unique	xs:boolean	optional	false		
source	<code><xs:element name="idProduct" type="idType"/></code>					

complexType referencesTo

diagram



children [idOrder](#) [idMsg](#) [idVehicle](#) [idDriver](#)

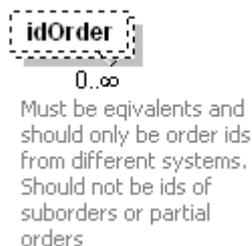
used by element [msg/referencesTo](#)

annotation documentation
Reference to earlier order and/or msg or reference to a resource at provider

source `<xs:complexType name="referencesTo">`
`<xs:annotation>`
`<xs:documentation>Reference to earlier order and/or msg or reference to a resource at provider</xs:documentation>`
`</xs:annotation>`
`<xs:sequence>`
`<xs:element name="idOrder" type="idType" minOccurs="0" maxOccurs="unbounded">`
`<xs:annotation>`
`<xs:documentation>Must be equivalents and should only be order ids from different systems. Should not be ids of suborders or partial orders</xs:documentation>`
`</xs:annotation>`
`</xs:element>`
`<xs:element name="idMsg" type="idType" minOccurs="0"/>`
`<xs:element name="idVehicle" type="idType" minOccurs="0"/>`
`<xs:element name="idDriver" type="idType" minOccurs="0"/>`
`</xs:sequence>`
`</xs:complexType>`

element referencesTo/idOrder

diagram



type [idType](#)

properties isRef 0

	minOcc	0				
	maxOcc	unbounded				
	content	complex				
attributes	Name	Type	Use	Default	Fixed	Annotation
	src	xs:string	required			
	id	xs:string	required			
	unique	xs:boolean	optional	false		
annotation	documentation	Must be equivalents and should only be order ids from different systems. Should not be ids of suborders or partial orders				
source	<pre><xs:element name="idOrder" type="idType" minOccurs="0" maxOccurs="unbounded"> <xs:annotation> <xs:documentation>Must be equivalents and should only be order ids from different systems. Should not be ids of suborders or partial orders</xs:documentation> </xs:annotation> </xs:element></pre>					

element referencesTo/idMsg

diagram						
type	idType					
properties	isRef	0				
	minOcc	0				
	maxOcc	1				
	content	complex				
attributes	Name	Type	Use	Default	Fixed	Annotation
	src	xs:string	required			
	id	xs:string	required			
	unique	xs:boolean	optional	false		
source	<pre><xs:element name="idMsg" type="idType" minOccurs="0"/></pre>					

element referencesTo/idVehicle

diagram						
type	idType					
properties	isRef	0				
	minOcc	0				
	maxOcc	1				
	content	complex				
attributes	Name	Type	Use	Default	Fixed	Annotation
	src	xs:string	required			
	id	xs:string	required			
	unique	xs:boolean	optional	false		
source	<pre><xs:element name="idVehicle" type="idType" minOccurs="0"/></pre>					

element referencesTo/idDriver

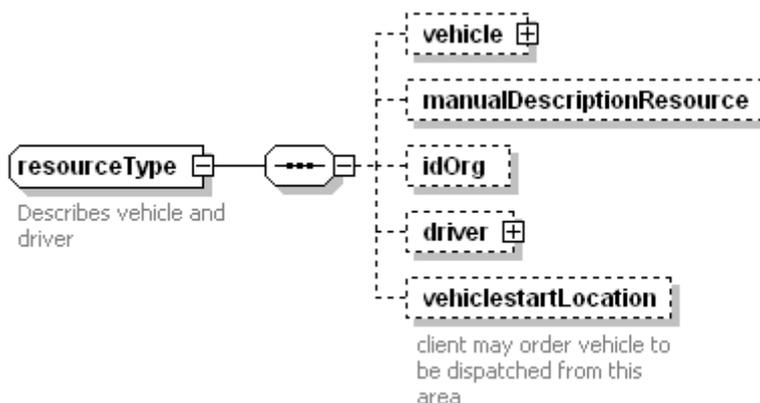
diagram						
type	idType					
properties	isRef	0				
	minOcc	0				
	maxOcc	1				
	content	complex				

attributes	Name	Type	Use	Default	Fixed	Annotation
	src	xs:string	required			
	id	xs:string	required			
	unique	xs:boolean	optional	false		

source `<xs:element name="idDriver" type="idType" minOccurs="0"/>`

complexType resourceType

diagram



children [vehicle](#) [manualDescriptionResource](#) [idOrg](#) [driver](#) [vehiclestartLocation](#)

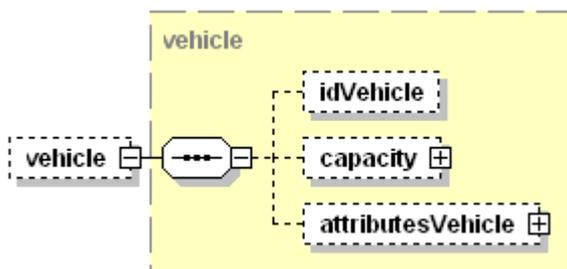
used by elements [content/resourceContent](#) [msg/resourceDispatch](#) [order/resourceOrder](#) [orderReject/resourceReject](#)

annotation documentation
Describes vehicle and driver

source `<xs:complexType name="resourceType">`
`<xs:annotation>`
`<xs:documentation>Describes vehicle and driver</xs:documentation>`
`</xs:annotation>`
`<xs:sequence>`
`<xs:element name="vehicle" type="vehicle" minOccurs="0"/>`
`<xs:element name="manualDescriptionResource" type="manualDescriptionType" minOccurs="0"/>`
`<xs:element name="idOrg" type="idType" minOccurs="0"/>`
`<xs:element name="driver" type="driver" minOccurs="0"/>`
`<xs:element name="vehiclestartLocation" type="geographicLocation" minOccurs="0">`
`<xs:annotation>`
`<xs:documentation>client may order vehicle to be dispatched from this area</xs:documentation>`
`</xs:annotation>`
`</xs:element>`
`</xs:sequence>`
`</xs:complexType>`

element resourceType/vehicle

diagram

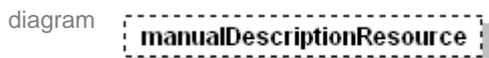


type [vehicle](#)

properties isRef 0
minOcc 0
maxOcc 1

content complex
 children [idVehicle](#) [capacity](#) [attributesVehicle](#)
 source `<xs:element name="vehicle" type="vehicle" minOccurs="0"/>`

element resourceType/manualDescriptionResource



type [manualDescriptionType](#)
 properties isRef 0
 minOcc 0
 maxOcc 1
 content complex
 attributes

Name	Type	Use	Default	Fixed	Annotation
manualText	xs:string	required			

source `<xs:element name="manualDescriptionResource" type="manualDescriptionType" minOccurs="0"/>`

element resourceType/idOrg

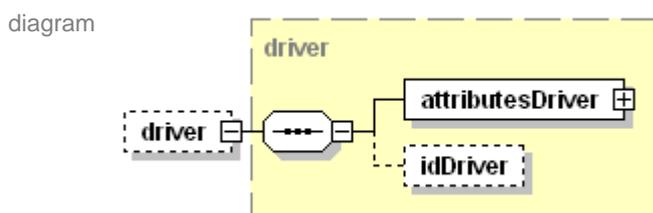


type [idType](#)
 properties isRef 0
 minOcc 0
 maxOcc 1
 content complex
 attributes

Name	Type	Use	Default	Fixed	Annotation
src	xs:string	required			
id	xs:string	required			
unique	xs:boolean	optional	false		

source `<xs:element name="idOrg" type="idType" minOccurs="0"/>`

element resourceType/driver



type [driver](#)
 properties isRef 0
 minOcc 0
 maxOcc 1
 content complex
 children [attributesDriver](#) [idDriver](#)
 attributes

Name	Type	Use	Default	Fixed	Annotation
name	xs:string	optional			

source `<xs:element name="driver" type="driver" minOccurs="0"/>`

element resourceType/vehiclestartLocation

diagram



type [geographicLocation](#)

properties	isRef	minOcc	maxOcc	content
	0	0	1	complex

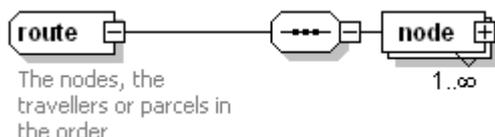
attributes	Name	Type	Use	Default	Fixed	Annotation
	zone	xs:string	optional			
	typeOfCoordinate	xs:string	required			
	lat	xs:float	required			
	long	xs:float	required			
	precision	xs:integer	required			

documentation
client may order vehicle to be dispatched from this area

```
source <xs:element name="vehiclestartLocation" type="geographicLocation" minOccurs="0">
  <xs:annotation>
    <xs:documentation>client may order vehicle to be dispatched from this area</xs:documentation>
  </xs:annotation>
</xs:element>
```

complexType route

diagram



The nodes, the travellers or parcels in the order

children [node](#)

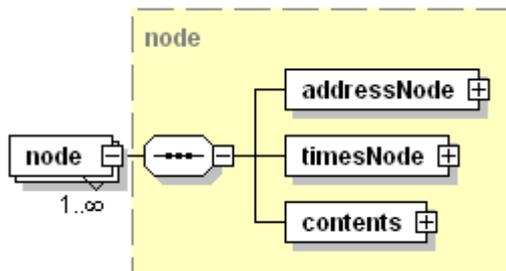
used by element [order/route](#)

documentation
The nodes, the travellers or parcels in the order

```
source <xs:complexType name="route">
  <xs:annotation>
    <xs:documentation>The nodes, the travellers or parcels in the order</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="node" type="node" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

element route/node

diagram



type [node](#)

properties
 isRef 0
 minOcc 1
 maxOcc unbounded
 content complex

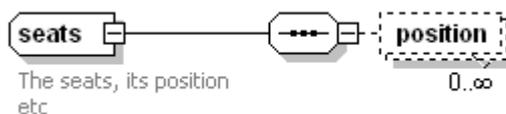
children [addressNode](#) [timesNode](#) [contents](#)

attributes	Name	Type	Use	Default	Fixed	Annotation
	nodeSeqno	xs:positiveInteger	required			
	nodeType	xs:string	required			

source `<xs:element name="node" type="node" maxOccurs="unbounded"/>`

complexType seats

diagram



children [position](#)

used by element [capacity/seats](#)

attributes	Name	Type	Use	Default	Fixed	Annotation
	noOfSeats	xs:nonNegativeInteger	required			

annotation documentation
 The seats, its position etc

source `<xs:complexType name="seats">
 <xs:annotation>
 <xs:documentation>The seats, its position etc</xs:documentation>
 </xs:annotation>
 <xs:sequence>
 <xs:element name="position" type="position" minOccurs="0" maxOccurs="unbounded"/>
 </xs:sequence>
 <xs:attribute name="noOfSeats" type="xs:nonNegativeInteger" use="required"/>
 </xs:complexType>`

element seats/position

diagram



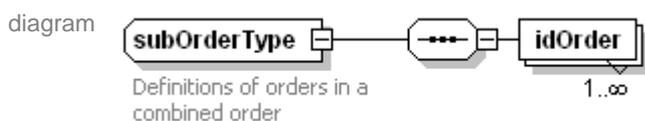
type [position](#)

properties
 isRef 0
 minOcc 0

attributes	Name	Type	Use	Default	Fixed	Annotation
	row	xs:positiveInteger	required			
	seat	xs:string	optional			
	direction	xs:string	optional			
	legSpace	xs:string	optional			
	access	xs:nonNegativeInteger	optional			

source `<xs:element name="position" type="position" minOccurs="0" maxOccurs="unbounded"/>`

complexType subOrderType



children [idOrder](#)

used by elements [content/subOrderContent](#) [msg/orderLink/subOrderLink](#)

annotation documentation
Definitions of orders in a combined order

source `<xs:complexType name="subOrderType">
 <xs:annotation>
 <xs:documentation>Definitions of orders in a combined order</xs:documentation>
 </xs:annotation>
 <xs:sequence>
 <xs:element name="idOrder" type="idType" maxOccurs="unbounded"/>
 </xs:sequence>
 </xs:complexType>`

element subOrderType/idOrder



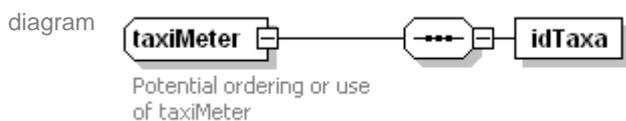
type [idType](#)

properties isRef 0
 minOcc 1
 maxOcc unbounded
 content complex

attributes	Name	Type	Use	Default	Fixed	Annotation
	src	xs:string	required			
	id	xs:string	required			
	unique	xs:boolean	optional	false		

source `<xs:element name="idOrder" type="idType" maxOccurs="unbounded"/>`

complexType taxiMeter



children [idTaxa](#)

used by element [price/taxiMeter](#)

attributes	Name	Type	Use	Default	Fixed	Annotation
	activate	xs:boolean	required			
annotation	documentation					Potential ordering or use of taxiMeter
source	<pre><xs:complexType name="taxiMeter"> <xs:annotation> <xs:documentation>Potential ordering or use of taxiMeter</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="idTaxa" type="idType"/> </xs:sequence> <xs:attribute name="activate" type="xs:boolean" use="required"/> </xs:complexType></pre>					

element taxiMeter/idTaxa



type [idType](#)

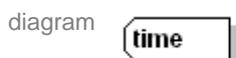
properties isRef 0
content complex

attributes	Name	Type	Use	Default	Fixed	Annotation
	src	xs:string	required			
	id	xs:string	required			
	unique	xs:boolean	optional	false		

source

```
<xs:element name="idTaxa" type="idType"/>
```

complexType time



Should normally be the local time.

used by elements [timesType/time](#) [msg/locationRequest/timeFrom](#) [msg/locationRequest/timeTo](#)

attributes	Name	Type	Use	Default	Fixed	Annotation
	timeAccuracy	xs:string	required			
	timeZone	xs:integer	required			
	time	xs:dateTime	required			

annotation documentation
Should normally be the local time.

source

```
<xs:complexType name="time">
  <xs:annotation>
    <xs:documentation>Should normally be the local time.</xs:documentation>
  </xs:annotation>
  <xs:attribute name="timeAccuracy" type="xs:string" use="required"/>
  <xs:attribute name="timeZone" type="xs:integer" use="required"/>
  <xs:attribute name="time" type="xs:dateTime" use="required"/>
</xs:complexType>
```

complexType timesType



Scheduled time and/or estimated time and/or actual time

1..3

children [time](#)

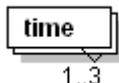
used by elements [node/timesNode](#) [associatedReservation/timesReservation](#)

annotation documentation Scheduled time and/or estimated time and/or actual time

source

```
<xs:complexType name="timesType">
  <xs:annotation>
    <xs:documentation>Scheduled time and/or estimated time and/or actual time</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="time" type="time" maxOccurs="3"/>
  </xs:sequence>
</xs:complexType>
```

element timesType/time

diagram 

type [time](#)

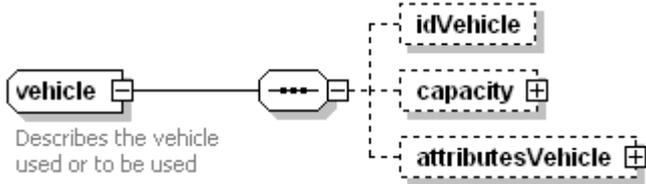
properties isRef 0
minOcc 1
maxOcc 3
content complex

attributes	Name	Type	Use	Default	Fixed	Annotation
	timeAccuracy	xs:string	required			
	timeZone	xs:integer	required			
	time	xs:dateTime	required			

source

```
<xs:element name="time" type="time" maxOccurs="3"/>
```

complexType vehicle

diagram 

children [idVehicle](#) [capacity](#) [attributesVehicle](#)

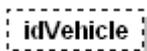
used by element [resourceType/vehicle](#)

annotation documentation Describes the vehicle used or to be used

source

```
<xs:complexType name="vehicle">
  <xs:annotation>
    <xs:documentation>Describes the vehicle used or to be used</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="idVehicle" type="idType" minOccurs="0"/>
    <xs:element name="capacity" type="capacity" minOccurs="0"/>
    <xs:element name="attributesVehicle" type="attributesType" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

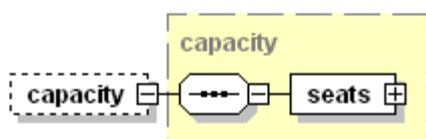
element vehicle/idVehicle

diagram 

type	idType					
properties	isRef	0				
	minOcc	0				
	maxOcc	1				
	content	complex				
attributes	Name	Type	Use	Default	Fixed	Annotation
	src	xs:string	required			
	id	xs:string	required			
	unique	xs:boolean	optional	false		
source	<code><xs:element name="idVehicle" type="idType" minOccurs="0"/></code>					

element **vehicle/capacity**

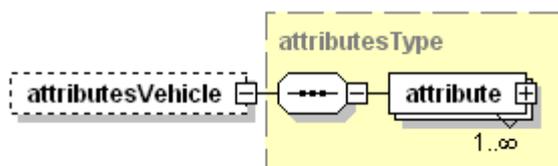
diagram



type	capacity					
properties	isRef	0				
	minOcc	0				
	maxOcc	1				
	content	complex				
children	seats					
attributes	Name	Type	Use	Default	Fixed	Annotation
	luggageArea	xs:float	optional			
	fullArea	xs:float	optional			
	stretcherArea	xs:float	optional			
source	<code><xs:element name="capacity" type="capacity" minOccurs="0"/></code>					

element **vehicle/attributesVehicle**

diagram



type	attributesType					
properties	isRef	0				
	minOcc	0				
	maxOcc	1				
	content	complex				
children	attribute					
source	<code><xs:element name="attributesVehicle" type="attributesType" minOccurs="0"/></code>					