

# XML Schema Documentation

## Table of Contents

- [Schema Document Properties](#)
- [Global Declarations](#)
  - [Element: \*\*zmluvy\*\*](#)
- [Global Definitions](#)
  - [Complex Type: \*\*dodatkyType\*\*](#)
  - [Complex Type: \*\*dodatokType\*\*](#)
  - [Complex Type: \*\*prilohaType\*\*](#)
  - [Complex Type: \*\*prilohyType\*\*](#)
  - [Complex Type: \*\*zmluvaType\*\*](#)
  - [Simple Type: \*\*datumType\*\*](#)

[top](#)

## Schema Document Properties

Target Namespace	None
Element and Attribute Namespaces	<ul style="list-style-type: none"><li>• Global element and attribute declarations belong to this schema's target namespace.</li><li>• By default, local element declarations belong to this schema's target namespace.</li><li>• By default, local attribute declarations have no namespace.</li></ul>

## Declared Namespaces

Prefix	Namespace
xml	http://www.w3.org/XML/1998/namespace
xs	http://www.w3.org/2001/XMLSchema

### Schema Component Representation

```
<xs:schema elementFormDefault="qualified">
  ...
</xs:schema>
```

[top](#)

## Global Declarations

Element: **zmluvy**

Name	zmluvy
Type	Locally-defined complex type
Nillable	no
Abstract	no
Diagram	

### XML Instance Representation

```
<zmluvy>
  <zmluva> zmluvaType </zmluva> [1]
</zmluvy>
```

### Schema Component Representation

```
<xs:element name="zmluvy">
  <xs:complexType>
    <xs:sequence minOccurs="1">
      <xs:element name="zmluva" type="zmluvaType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

[top](#)

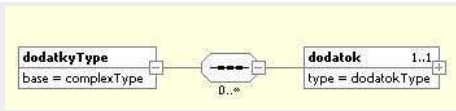
## Global Definitions

Complex Type: **dodatkyType**

Super-types:	None
Sub-types:	None

Name	dodatkyType
Abstract	no

Diagram



XML Instance Representation

```
<...>
  Start Sequence [0..*]
    <dodatok> dodatokType </dodatok> [1]
  End Sequence
</...>
```

Schema Component Representation

```
<xs:complexType name="dodatkyType">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:element name="dodatok" type="dodatokType"/>
  </xs:sequence>
</xs:complexType>
```

[top](#)

Complex Type: **dodatokType**

Super-types:	None
Sub-types:	None

Name	dodatokType
Abstract	no
Diagram	

XML Instance Representation

```
<...>
  <id_dodatku> xs:nonNegativeInteger </id_dodatku> [0..1] ?
  <cislo_dodatku> xs:string </cislo_dodatku> [1] ?
  <obstaravatel> xs:string </obstaravatel> [1] ?
  <dodavatel> xs:string </dodavatel> [1] ?
  <dodavatel_ico> xs:string </dodavatel_ico> [1] ?
  <datum_ucinnosti> datumType </datum_ucinnosti> [0..1] ?
  <datum_platnosti_do> datumType </datum_platnosti_do> [0..1] ?
  <predmet_dodatku> xs:string </predmet_dodatku> [1] ?
  <suma> xs:float </suma> [1] ?
  <poznamka> xs:string </poznamka> [0..1] ?
  <prilohy> prilohyType </prilohy> [1..*] ?
</...>
```

Schema Component Representation

```
<xs:complexType name="dodatokType">
  <xs:sequence>
    <xs:element name="id_dodatku" type="xs:nonNegativeInteger" minOccurs="0"/>
    <xs:element name="cislo_dodatku" type="xs:string"/>
    <xs:element name="obstaravatel" type="xs:string"/>
    <xs:element name="dodavatel" type="xs:string"/>
    <xs:element name="dodavatel_ico" type="xs:string"/>
    <xs:element name="datum_ucinnosti" type="datumType" minOccurs="0"/>
    <xs:element name="datum_platnosti_do" type="datumType" minOccurs="0"/>
    <xs:element name="predmet_dodatku" type="xs:string"/>
    <xs:element name="suma" type="xs:float"/>
    <xs:element name="poznamka" type="xs:string"/>
    <xs:element name="prilohy" type="prilohyType"/>
  </xs:sequence>
</xs:complexType>
```

```
<xs:element name="poznamka" type="xs:string" minOccurs="0"/>
<xs:element name="prilohy" type="prilohyType" minOccurs="1" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
```

[top](#)

Complex Type: **prilohaType**

Super-types:	None
Sub-types:	None

Name	prilohaType
Abstract	no
Diagram	

```
<...>
  <id_priloha> xs:nonNegativeInteger </id_priloha> [0..1] ?
  <nazov_suboru> xs:string </nazov_suboru> [1] ?
  <popis> xs:string </popis> [1] ?
  <subor_prilohy> xs:base64Binary </subor_prilohy> [1] ?
</...>
```

```
<xs:complexType name="prilohaType">
  <xs:sequence>
    <xs:element name="id_priloha" type="xs:nonNegativeInteger" minOccurs="0"/>
    <xs:element name="nazov_suboru" type="xs:string"/>
    <xs:element name="popis" type="xs:string"/>
    <xs:element name="subor_prilohy" type="xs:base64Binary"/>
  </xs:sequence>
</xs:complexType>
```

[top](#)

Complex Type: **prilohyType**

Super-types:	None
Sub-types:	None

Name	prilohyType
Abstract	no
Diagram	

```
<...>
  Start Sequence [0..*]
  <priloha> prilohaType </priloha> [1]
  End Sequence
</...>
```

```
<xs:complexType name="prilohyType">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:element name="priloha" type="prilohaType"/>
  </xs:sequence>
</xs:complexType>
```

[top](#)

Complex Type: **zmluvaType**

Super-types:	None
Sub-types:	None

Name	zmluvaType
Abstract	no
Diagram	<pre> classDiagram     class zmluvaType {         &lt;&lt;complexType&gt;&gt;         id_zmluvy 0..1 type = nonNegativeInteger         cislo_zmluvy 1..1 type = string         obstaravatel 1..1 type = string         dodavatel 1..1 type = string         dodavatel_ico 1..1 type = string         predmet_zmluvy 1..1 type = string         datum_ucinnosti 0..1 type = datumType         datum_platnosti_do 0..1 type = datumType         suma_zmluva 1..1 type = float         suma_spolu 1..1 type = float         poznamka 0..1 type = string         prilohy 1..00 type = prilohyType         dodatky 0..00 type = dodatkyType     }     </pre>

### XML Instance Representation

### Schema Component Representation

[top](#)

## Simple Type: `datumType`

Super-types:	<a href="#">xs:string</a> < <b>datumType</b> (by restriction)
Sub-types:	None

<b>Name</b>	datumType
<b>Content</b>	<ul style="list-style-type: none"> <li>Base XSD Type: string</li> <li><i>pattern</i> = ([1-2][0-9][0-9][0-9])-(01 02 03 04 05 06 07 08 09 10 11 12)-(01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31)</li> </ul>

Schema Component Representation

```
<xs:simpleType name="datumType">
  <xs:restriction base="xs:string">
    <xs:pattern value="([1-2][0-9][0-9]
[0-9])-(01|02|03|04|05|06|07|08|09|10|11|12)-(01|02|03|04|05|06|07|08|09|10|11|12|13|14|15|16|17|18|19|20|21|22|23|24|25|26|27|28|29|30|31)"/>
  </xs:restriction>
</xs:simpleType>
```

[top](#)