

Handleiding XML Eindopdracht INF2G



Gemaakt door: David Dorenbos, Ivar Huizing, Michiel Tombergen, Andrej Rodzevic, Thomas Fluit, Joris Rietveld

Versie: 1.0 | 12-04-2019

Omschrijving DTD Eindopdracht XML

Opbouw Garages

Als eerste zal worden gekeken naar de opbouw van het XML-document en het hierbij behorende onderdeel van het DTD. Het XML-document met behulp van een XML openingstag. Hierin aangegeven welke versie van XML wordt gebruikt, en welke encoding wordt gebruikt. Bij deze opdracht is er gekozen voor de encoding UTF-8 omdat dit een standaard is. Hetzelfde geldt voor de gebruikte versie van XML. Hierna wordt het DTD verbonden aan het XML-document zodat het kan worden gebruikt om het XML te valideren.

Nu dat het document is geopend, zal in het XML de groep 'Garages' geopend worden. 'Garages' is een collectie van allemaal verschillende garages. Hierbinnen zit daarom dus Element 'Garage'. Er zal altijd één groep zijn die 'Garages' genoemd is. Hierbinnen kunnen meerdere 'Garage' Elementen zich bevinden. Dit is ook te zien in het DTD die onder het code overzicht van het XML weergegeven staat.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Garages SYSTEM "eindopdracht.dtd">
<Garages>
  <Garage>
    <!-- Dit zijn de gegevens van de garage -->
    <Naam>Angelslo's auto's</Naam>
    <Adres...>
    <Telefoonnummer>06-12345678</Telefoonnummer>
    <Email>autos-angelslo@hotmail.com</Email>
    <Fotos...>
    <Omschrijving>Wij verkopen nieuwe en tweedehands auto's</Omschrijving>
    <Website>www.angelslo-autos.nl</Website>
    <Routebeschrijving>https://goo.gl/maps/vJ9j4VUofS2</Routebeschrijving>

    <!-- Dit zijn de organisaties waarbij de garage is aangesloten -->
    <AangeslotenOrganizaties...>
    <!-- De openingstijden van de verkoop -->
    <VkOpeningstijden format="HH:MM - HH:MM"...>
    <!-- De openings tijden van de werkplaats -->
    <WPOpeningstijden format="HH:MM - HH:MM"...>
    <!-- Dit zijn de sercies die de garage aanbied -->
    <Services...>
    <!-- Dit zijn de de merken waarvan de garage een officiële dealer is-->
    <Automerken...>
    <!-- Dit zijn de faciliteiten die de garage aanbied -->
    <Faciliteiten...>
    <!-- Dit zijn de auto's die de garage verkoopt -->
    <Autos...>
    <!-- Dit zijn de werkzaam heden die de garage aanbied -->
    <Werkzaamheden...>
    <!-- Dit zijn de medewerkers die werkzaam zijn bij de garage. -->
    <Medewerkers...>
  </Garage>
</Garages>
```

Afbeelding 1: overzicht opzet XML

```
<!ELEMENT Garages (Garage+)>
<!-- Dit valideerd de gegevens die voor een garage opgeslagen kunnen worden. -->
<!ELEMENT Garage (Naam, Adres, Telefoonnummer, Email, Fotos,Omschrijving?,Website?,
Routebeschrijving?,AangeslotenOrganizaties?,VkOpeningstijden,WPOpeningstijden,
Services?,Automerken?,Faciliteiten?,Autos?,Werkzaamheden?,Medewerkers)>
<!ELEMENT Naam (#PCDATA)>
<!ELEMENT Adres (Straat, Huisnummer, Postcode, Plaats, Provincie)>
<!ELEMENT Telefoonnummer (#PCDATA)>
<!ELEMENT Email (#PCDATA)>
<!ELEMENT Fotos (Foto*)>
<!ELEMENT Foto (#PCDATA)>
<!ELEMENT Omschrijving (#PCDATA)>
<!ELEMENT Website (#PCDATA)>
<!ELEMENT Routebeschrijving (#PCDATA)>
<!ELEMENT AangeslotenOrganizaties (Organizatie+)>
<!ELEMENT Organizatie (#PCDATA)>
<!ATTLIST Organizatie naam CDATA #REQUIRED>
```

Afbeelding 2: Overzicht onderdeel DTD dat de opzet XML valideert

Binnen in het ‘Garage’ Element staan Elementen die de garage beschrijven. Hiervan zijn er een aantal in het DTD genoteerd als ‘(#PCDATA)’. Dit is omdat het tekstuele data is. De tags die hiervan gebruik maken binnen ‘Garage’ zijn hieronder weergegeven.

```
<!ELEMENT Naam (#PCDATA)>
<!ELEMENT Telefoonnummer (#PCDATA)>
<!ELEMENT Email (#PCDATA)>
<!ELEMENT Foto (#PCDATA)>
<!ELEMENT Omschrijving (#PCDATA)>
<!ELEMENT Website (#PCDATA)>
<!ELEMENT Routebeschrijving (#PCDATA)>
<!ELEMENT Organizatie (#PCDATA)>
```

Afbeelding 3: Overzicht tags die ‘(#PCDATA)’ gebruiken binnen ‘Garage’

Er zijn ook een aantal entiteiten die binnen in garage zullen worden gebruikt. Deze staan hieronder weergegeven.

```
<!--Hier worden de entiteiten van de garage aangemaakt -->
<ENTITY % dagen "(Maandag, Dinsdag, Woensdag, Donderdag, Vrijdag, Zaterdag, Zondag)">
<ENTITY % valuta "(euro|dollar)">
```

Afbeelding 4 :Entiteiten die later zullen worden toegepast binnen in garages

Adres

Hieronder is een voorbeeld van een ‘Adres’ weergegeven. ‘Adres’ is een tag binnen de hiervoor opgezette ‘Garage’. Binnen in garage worden meerdere elementen aangemaakt. Deze elementen zijn: ‘Straat’, ‘Huisnummer’, ‘Postcode’, ‘Plaats’ en ‘Provincie’. Omdat deze Elementen allemaal een tekstuele inhoud hebben, staat er in het DTD ‘(#PCDATA)’ achter.

```
<Adres>
  <Straat>Hoofdstraat</Straat>
  <Huisnummer>25</Huisnummer>
  <Postcode>7821 AS</Postcode>
  <Plaats>Emmen</Plaats>
  <Provincie>Drenthe</Provincie>
</Adres>
```

Afbeelding 5: voorbeeld ‘Adres’ in XML

```
<!ELEMENT Adres (Straat, Huisnummer, Postcode, Plaats, Provincie)>
<!ELEMENT Straat (#PCDATA)>
<!ELEMENT Huisnummer (#PCDATA)>
<!ELEMENT Postcode (#PCDATA)>
<!ELEMENT Plaats (#PCDATA)>
<!ELEMENT Provincie (#PCDATA)>
```

Afbeelding 6: DTD dat ‘Adres’ valideert

Openingtijden Werkplaats en Verkoop

Het element ‘WPOpeningstijden’, binnen in ‘Garage’, staat voor de werkplaats openingstijden en het element ‘VKOpeningstijden’ staat voor de verkoopafdeling openingstijden. Hieronder is een voorbeeld van beide in het XML weergegeven.

```
<!-- De openingstijden van de verkoop -->
<VkOpeningstijden format="HH:MM - HH:MM">
  <Maandag>08:00 - 17:00</Maandag>
  <Dinsdag>08:00 - 17:00</Dinsdag>
  <Woensdag>08:00 - 17:00</Woensdag>
  <Donderdag>08:00 - 20:00</Donderdag>
  <Vrijdag>08:00 - 17:00</Vrijdag>
  <Zaterdag>08:00 - 17:00</Zaterdag>
  <Zondag>12:00 - 17:00</Zondag>
</VkOpeningstijden>

<!-- De openings tijden van de werkplaats -->
<WPOpeningstijden format="HH:MM - HH:MM">
  <Maandag>08:00 - 18:00</Maandag>
```

```

    <Dinsdag>08:00 - 18:00</Dinsdag>
    <Woensdag>08:00 - 18:00</Woensdag>
    <Donderdag>08:00 - 21:00</Donderdag>
    <Vrijdag>08:00 - 18:00</Vrijdag>
    <Zaterdag>08:00 - 17:00</Zaterdag>
    <Zondag>12:00 - 17:00</Zondag>
  </WPOpeningstijden>

```

Afbeelding 7: Openingstijden in het XML

‘VKOpeningstijden’ en ‘WPOpeningstijden’ maken beide gebruik van de eerder gedefinieerde elementen van dagen. Deze worden hier ook direct aangemaakt in het DTD. Alle dagen zijn tekstueel en dus zijn ze, zoals hieronder te zien is, gedefinieerd als ‘(#PCDATA)’. Om een formaat weer te geven voor de dagen, is er een formaat weergegeven. Dit formaat is in het XML: "HH:MM - HH:MM". Dit is verplicht binnen een dag te staan en heeft dus in het DTD een ‘#REQUIRED’ erbij staan.

```

<!-- Dit valideert de openings tijden van de werkplaats en verkoop -->
<!ELEMENT WPOpeningstijden %dagen;>
<!ATTLIST WPOpeningstijden format CDATA #REQUIRED>
<!ELEMENT VkOpeningstijden %dagen;>
<!ATTLIST VkOpeningstijden format CDATA #REQUIRED>
<!ELEMENT Maandag (#PCDATA)>
<!ELEMENT Dinsdag (#PCDATA)>
<!ELEMENT Woensdag (#PCDATA)>
<!ELEMENT Donderdag (#PCDATA)>
<!ELEMENT Vrijdag (#PCDATA)>
<!ELEMENT Zaterdag (#PCDATA)>
<!ELEMENT Zondag (#PCDATA)>

```

Afbeelding 8: Openingstijden in het DTD

Services

Binnen het element ‘Services’ kunnen zich meerdere services bevinden. Deze worden met het ‘Service’ element aangemaakt. In het onderstaande voorbeeld zijn de services dus bijvoorbeeld huren en lenen. Omdat ‘Service’ tekstuele data bevat, is het gedefinieerd als ‘(#PCDATA)’.

```

<!-- Dit zijn de services die de garage aanbied -->
<Services>
  <Service>Huren</Service>
  <Service>lenen</Service>
</Services>

```

Afbeelding 9: Services in het XML

```

<!-- Dit valideert de Services -->
<!ELEMENT Services (Service+)>
<!ELEMENT Service (#PCDATA)>

```

Afbeelding 10: Services in het DTD

Automerken

De garage kan in verschillende automerken handelen. Hierom kunnen zich in het element ‘Automerken’, zich dus meerdere elementen van ‘Automerk’ bevinden. Hieronder staat een voorbeeld weergegeven van een garage die in de merken Ford, Kia en Mercedes handelt. Omdat een ‘Automerk’ element tekstuele informatie bevat, is het gedefinieerd in het DTD als ‘(#PCDATA)’.

```

<!-- Dit zijn de de merken waarvan de garage een officiële dealer is-->
<Automerken>
  <Automerk>Ford</Automerk>
  <Automerk>Kia</Automerk>
  <Automerk>Mercedes</Automerk>
</Automerken>

```

Afbeelding 11: XML van ‘Automerken’

```

<!-- Dit valideert de de beschikbare auto merken. -->
<!ELEMENT Automerken (Automerk+)>
<!ELEMENT Automerk (#PCDATA)>

```

Afbeelding 12: DTD van ‘Automerken’

Faciliteiten

Er kunnen verschillende faciliteiten zijn binnen in een ‘Garage’. Hierom is er gekozen om alles optioneel te maken. Dit is omdat er een garage kan zijn met een wasstraat, maar bijvoorbeeld zonder een pomp om aan te tanken. De ‘Wasbox’ en ‘Wasstraat’ hebben geen tekstuele informatie erin staan en dus is het gedefinieerd als ‘EMPTY’.

Een faciliteit van een garage kan ook een collectie aan ‘Pompen’ zijn. Hierin zitten dan meerdere ‘Pomp’ elementen die verschillende soorten tankbare ‘Brandstoffen’ kunnen afgeven. Deze mogelijke ‘Brandstoffen’ zijn: ‘Gas’, ‘Diesel’ en ‘Benzine’.

```
<!-- Dit zijn de faciliteiten die de garage aanbied -->
<Faciliteiten>
  <Pompen>
    <Pomp>
      <Brandstoffen>
        <Benzine />
        <Gas />
        <Diesel />
      </Brandstoffen>
    </Pomp>
  </Pompen>
  <Wasbox/>
</Faciliteiten>
```

Afbeelding 13: Voorbeeld van ‘Faciliteiten’ in het XML

```
<!-- Dit valideert de beschikbare faciliteiten. -->
<!ELEMENT Faciliteiten (Pompen?,Wasbox?,Wasstraat?,Winkel?)>
<!ELEMENT Pompen (Pomp*)>
<!ELEMENT Pomp (Brandstoffen)>
<!ELEMENT Wasbox EMPTY>
<!ELEMENT Wasstraat EMPTY>
```

Afbeelding 14: Voorbeeld van ‘Faciliteiten’ in het DTD

In het geval dan dit voorbeeld is er geen winkel beschikbaar. Als deze er wel zou wezen, dan zou er in de winkel zich een collectie ‘Producten’ bevinden. In deze collectie zal dan een aantal elementen zitten die tekstuele informatie bevat en dus is gedefinieerd als ‘(#PCDATA)’

```
<!-- Dit valideert de gegevens van een winkel. -->
<!ELEMENT Winkel (Producten*)>
<!ELEMENT Producten (Product+)>
<!ELEMENT Product (#PCDATA)>
```

Afbeelding 15: Dit is een voorbeeld van een winkel

Auto's

Het element ‘Autos’ is een collectie waarin meerdere ‘Auto’ elementen kunnen zitten. Een auto heeft verschillende informatie erin opgeslagen. De elementen ‘Kenteken’, ‘Automerik’ en ‘Type’ zijn allemaal pure tekstuele gegevens en zijn dus gedefinieerd als ‘(#PCDATA)’.

De elementen ‘Bouwjaar’ en ‘Prijs’ bevatten ook tekstuele informatie, en zijn daarom dus ook gedefinieerd als ‘(#PCDATA)’. Het enige verschil is dat het element ‘Bouwjaar’ een format heeft van “yyyy”, en het element ‘Prijs’ moet een valuta aangeven die wordt toegepast voor de prijs. In het geval van dit voorbeeld is het “euro”, maar het kan ook “dollar” zijn. Dit is in het DTD te zien.

Dan is er het element ‘Brandstof’. In dit element moet zich één vorm van brandstof bevinden. Dit moet ‘Benzine’, ‘Gas’ of ‘Diesel’ zijn. Dit is omdat een auto natuurlijk niet op verschillende soorten brandstof kan rijden.

‘DatumAPK’ is een element waarin de laatste APK-Keuring wordt weergegeven. De datum wordt ingevoerd als tekst onder de elementen ‘Jaar’, ‘Maand’ en Dag. Deze elementen moeten aanwezig zijn. Ook zijn ze tekstueel en dus moeten ze gedefinieerd worden met ‘(#PCDATA)’.

Als laatste is er binnen in een ‘Auto’ een collectie aanwezig van ‘Fotos’. Binnen in deze collectie kunnen er meerdere ‘Foto’ elementen zitten. Hierin wordt de link naar de foto opgeslagen om later bijvoorbeeld weer te geven in een pdf, of op een website. De ‘Foto’ tag is dus tekstueel en dus moet het weer gedefinieerd worden met een ‘(#PCDATA)’

```
<!-- Dit zijn de auto's die de garage verkoopt -->
<Autos>
  <Auto>
    <Kenteken>98-RR-XX</Kenteken>
    <Automerk>Mazda</Automerk>
    <Type>Cabrio</Type>
    <Bouwjaar format="yyyy">2005</Bouwjaar>
    <Brandstof>
      <Benzine />
    </Brandstof>
    <Prijs valuta="euro">10000</Prijs>
    <!-- Dit is de datum waarop de APK afloopt. -->
    <DatumAPK>
      <Jaar>2019</Jaar>
      <Maand>01</Maand>
      <Dag>20</Dag>
    </DatumAPK>

    <Fotos>
      <Foto>Front.jpg</Foto>
      <Foto>Back.jpg</Foto>
      <Foto>Inside.jpg</Foto>
      <Foto>Motor.jpg</Foto>
    </Fotos>
  </Auto>
</Autos>
```

Afbeelding 16: een voorbeeld van een ‘Auto’ binnen in een verzameling ‘Autos’ in het XML

```
<!-- Dit valideert de gegevens van auto's -->
<!ELEMENT Autos (Auto+)>
<!ELEMENT Auto (Kenteken, Automerk, Type, Bouwjaar, Brandstof, Prijs, DatumAPK, Fotos)>
<!ELEMENT Kenteken (#PCDATA)>
<!ELEMENT Type (#PCDATA)>
<!ELEMENT Bouwjaar (#PCDATA)>
<!ATTLIST Bouwjaar format CDATA #REQUIRED>
<!ELEMENT Brandstoffen (Benzine?, Gas?, Diesel?)>
<!ELEMENT Brandstof (Benzine|Gas|Diesel)>

<!ELEMENT Benzine (#PCDATA)>
<!ELEMENT Gas (#PCDATA)>
<!ELEMENT Diesel (#PCDATA)>

<!ELEMENT Prijs (#PCDATA)>
<!ATTLIST Prijs valuta %valuta; #REQUIRED>
```

Afbeelding 17: DTD validatie voor ‘Autos’

Werkzaamheden

Het element ‘Werkzaamheden’ is een collectie van meerdere elementen genaamd ‘Werkzaamheid’. Er kunnen dus meerdere elementen van ‘Werkzaamheid’ binnen in ‘Werkzaamheden’ zitten. Elke ‘Werkzaamheid’ heeft tekstuele informatie om te beschrijven wat voor werkzaamheid het is en dus wordt het gedefinieerd als ‘(#PCDATA)’.

```
<!-- Dit zijn de werkzaamheden die de garage aanbied -->
<Werkzaamheden>
  <Werkzaamheid>onderhoud</Werkzaamheid>
  <Werkzaamheid>APKKeuring</Werkzaamheid>
  <Werkzaamheid>BandenService</Werkzaamheid>
</Werkzaamheden>
```

Afbeelding 18: Werkzaamheden in het XML

```
<!-- Dit valideert de werkzaamheden. -->
<!ELEMENT Werkzaamheden (Werkzaamheid+)>
<!ELEMENT Werkzaamheid (#PCDATA)>
```

Afbeelding 19: Werkzaamheden in het DTD

Medewerkers

Als laatste zit er binnen elke ‘Garage’ een collectie aan ‘Medewerkers’. Binnen deze collectie zitten meerdere ‘Medewerker’ elementen. Er kunnen dus meerdere ‘Medewerker’ elementen in een collectie van ‘Medewerkers’ wezen. Elke ‘Medewerker’ heeft drie elementen: ‘Naam’, ‘Foto’ (een link naar een foto van de medewerker) en ‘Functie’. Deze zijn allemaal tekstuele informatie en zijn dus gedefinieerd als ‘(#PCDATA)’.

```
<!-- Dit zijn de medewerkers die werkzaam zijn bij de garage. -->
<Medewerkers>
  <Medewerker>
    <Naam>Jos Harmsen</Naam>
    <Foto>jos.jpg</Foto>
    <Functie>Pompbediende</Functie>
  </Medewerker>
  <Medewerker>
    <Naam>Gerard Provrex</Naam>
    <Foto>me.jpg</Foto>
    <Functie>Monteur</Functie>
  </Medewerker>
</Medewerkers>
```

Afbeelding 20: Voorbeeld ‘Medewerkers’ in het XML

```
<!-- Dit valideert de medewerkers. -->
<!ELEMENT Medewerkers (Medewerker+)>
<!ELEMENT Medewerker (Naam, Foto , Functie)>
<!ELEMENT Functie (#PCDATA)>
```

Afbeelding 21: Validatie van ‘Medewerkers’ in het DTD