

RFC 4287

Atom, Eserlerin Ödünleşimli Paylaşım Biçimi

Düzenleyen:
M. Nottingham

Düzenleyen:
R. Sayre

Çeviren:
Nilgün Belma Bugüner

Eylül 2006

Özet

Bu belge, Genel Ağ'da eserlerin birlikleşimini sağlamak için eserin içeriği ve öteverileri hakkında bilgi veren, Atom adı verilen XML tabanlı yapının biçim belirtimidir.

Konu Başlıkları

1. Giriş	4
1.1. Örnekler	4
1.2. İsim Alanı ve Sürüm	5
1.3. Gösterimsel Uzlaşımlar	5
2. Atom Belgeleri	6
3. Ortak Atom Oluşumları	7
3.1. Metin Oluşumları	7
3.1.1. "type" Özniteliği	7
3.2. Kişilik Oluşumları	9
3.2.1. "atom:name" Elemanı	9
3.2.2. "atom:uri" Elemanı	9
3.2.3. "atom:email" Elemanı	9
3.3. Tarih Oluşumları	9
4. Atom İsim Alanındaki Elemanların Tanımları	10
4.1. Taşıyıcı Elemanlar	10
4.1.1. "atom:feed" Elemanı	10
4.1.1.1. Metinsel İçeriğin Sağlanması	11
4.1.2. "atom:entry" Elemanı	11
4.1.3. "atom:content" Elemanı	12
4.1.3.1. İşlem Modeli	13
4.1.4. Örnekler	14
4.2. Öteveri Elemanları	15
4.2.1. "atom:author" Elemanı	15
4.2.2. "atom:category" Elemanı	15
4.2.3. "atom:contributor" Elemanı	15
4.2.4. "atom:generator" Elemanı	16
4.2.5. "atom:icon" Elemanı	16
4.2.6. "atom:id" Elemanı	16
4.2.6.1. atom:id Karşılaştırması	17

4.2.7. "atom:link" Elemanı	17
4.2.8. "atom:logo" Elemanı	19
4.2.9. "atom:published" Elemanı	19
4.2.10. "atom:rights" Elemanı	19
4.2.11. "atom:source" Elemanı	19
4.2.12. "atom:subtitle" Elemanı	20
4.2.13. "atom:summary" Elemanı	20
4.2.14. "atom:title" Elemanı	20
4.2.15. "atom:updated" Elemanı	20
5. Atom Belgelerinin Güvenliği	20
5.1. Sayısal İmzalar	21
5.2. Şifreleme	21
5.3. İmzaladıktan sonra Şifreleme	22
6. Atom Eklentileri	22
6.1. Atom dışı Dillerden Eklentiler	22
6.2. Atom Söz Dağırcığına Eklemeler	22
6.3. Yabancı İmlemenin İşlenmesi	22
6.4. Eklenti Elemanları	22
6.4.1. Basit Eklenti Elemanları	23
6.4.2. Yapılı Eklenti Elemanları	23
7. IANA Değerlendirmeleri	23
7.1. Bağlantı İlişkilerinin Kaydı	24
8. Güvenlik Değerlendirmeleri	24
8.1. HTML ve XHTML İçerik	24
8.2. URI'ler	24
8.3. IRI'ler	25
8.4. Aldatma	25
8.5. Şifreleme ve İmzalama	25
9. Kaynakça	25
9.1. Uyulması Zorunlu Olanlar	25
9.2. Bilgi Niteliğindekiiler	27
A. Katkıda Bulunanlar	28
B. RELAX NG, Bütünleşik Şema	28
C. Belge Hakkında	35

Geçmiş

1.0 İlk çeviri	Eylül 2006	NBB
Standart Aday Adayı Özgün sürüm	Aralık 2005	MN

Sürüm Bilgileri

Ağ Çalışma Grubu
Açıklama İsteği: 4287
Durumu: Standart Aday Adayı

Yasal Uyarı

RFC'lerin yazarlarının hakları [BCP 78^{\(B1\)}](#) ile düzenlenmiştir. Dolayısıyla RFC çevirilerinin çevirmenlerinin haklarını da BCP 78'in düzenlediği kabul edilmiştir.

Bu belge [IETF^{\(B3\)}](#) tarafından yayınlanan resmi RFC 4287'nin **gayriresmi** çevirisidir ve aslının yerine kullanılamaz. Bu çevirinin hiçbir bağlamda ya da koşulda hükmü yoktur. Bu çeviri, anadili Türkçe olan internet kullanıcılarının bu RFC hakkında fikir edinebilmelerini sağlamak amacıyla hazırlanmıştır.

BU BELGE "ÜCRETSİZ" OLARAK RUHSATLANDIĞI İÇİN, İÇERDİĞİ BİLGİLER İÇİN İLGİLİ KANUNLARIN İZİN VERDİĞİ ÖLÇÜDE HERHANGİ BİR GARANTİ VERİLMEMEKTEDİR. AKSİ YAZILI OLARAK BELİRTİLMEDİĞİ MÜDDETÇE TELİF HAKKI SAHİPLERİ VE/VEYA BAŞKA ŞAHISLAR BELGEYİ "OLDUĞU GİBİ", AŞIKAR VEYA ZIMNEN, SATILABİLİRLİĞİ VEYA HERHANGİ BİR AMACA UYGUNLUĞU DA DAHİL OLMAK ÜZERE HİÇBİR GARANTİ VERMEKSİZİN DAĞITMAKTADIRLAR. BİLGİNİN KALİTESİ İLE İLGİLİ TÜM SORUNLAR SİZE AİTTİR. HERHANGİ BİR HATALI BİLGİDEN DOLAYI DOĞABİLECEK OLAN BÜTÜN SERVİS, TAMİR VEYA DÜZELTME MASRAFLARI SİZE AİTTİR.

İLGİLİ KANUNUN İCBAR ETTİĞİ DURUMLAR VEYA YAZILI ANLAŞMA HARİCİNDE HERHANGİ BİR ŞEKİLDE TELİF HAKKI SAHİBİ VEYA YUKARIDA İZİN VERİLDİĞİ ŞEKİLDE BELGEYİ DEĞİŞTİREN VEYA YENİDEN DAĞITAN HERHANGİ BİR KİŞİ, BİLGİNİN KULLANIMI VEYA KULLANILAMAMASI (VEYA VERİ KAYBI OLUŞMASI, VERİNİN YANLIŞ HALE GELMESİ, SİZİN VEYA ÜÇÜNCÜ ŞAHISLARIN ZARARA UĞRAMASI VEYA BİLGİLERİN BAŞKA BİLGİLERLE UYUMSUZ OLMASI) YÜZÜNDEN OLUŞAN GENEL, ÖZEL, DOĞRUDAN YA DA DOLAYLI HERHANGİ BİR ZARARDAN, BÖYLE BİR TAZMİNAT TALEBİ TELİF HAKKI SAHİBİ VEYA İLGİLİ KİŞİYE BİLDİRİLMİŞ OLSA DAHİ, SORUMLU DEĞİLDİR.

Tüm telif hakları aksi özellikle belirtilmediği sürece sahibine aittir. Belge içinde geçen herhangi bir terim, bir ticari isim ya da kuruma itibar kazandırma olarak algılanmamalıdır. Bir ürün ya da markanın kullanılmış olması ona onay verildiği anlamında görülmemelidir.

Bu Belgenin Durumu

Bu belge Genel Ağ standardı olmak üzere standartlaşma aşamasına girmiş bir protokolü belirtir ve geliştirilmesi için Genel Ağ topluluğundan üzerinde tartışılmasını ve önerilerde bulunulmasını talep eder. Bu protokolün durumu ve standartlaşma aşamalarındaki seviyesi için lütfen "Genel Ağ Resmi Protokol Standartları" (STD 1) belgesinin şu an ki sürümüne bakınız. Bu belgenin dağıtımına sınırlama getirilmemiştir.

Copyright © The Internet Society (2001). All Rights Reserved.

1. Giriş

Atom, "ulamalar" (feeds) olarak bilinen bilgilerle ilgili listeleri açıklayan XML'e dayalı bir belge biçimidir. Bir ulama, her biri bir belgenin genişleyebilir öteverilerinden oluşan ve "girdi" adı verilen bir miktar öğeden oluşur. Örneğin her girdinin bir konu başlığı vardır.

Atomun asıl kullanım alanı, sanal günlükler (bloglar), haber başlıkları gibi sanal eserlerin sanalyörelere başka doğrudan kullanıcı arayüzleriyle paylaşımını⁽¹⁾ sağlamaktır.

1.1. Örnekler

Kısa bir tek girdilik Atom Ulama Belgesi:

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">

  <title>Örnek Ulam</title>
  <link href="http://mesela.der/" />
  <updated>2003-12-13T18:30:02Z</updated>
  <author>
    <name>Yaşar Yaşamaz</name>
  </author>
  <id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6</id>

  <entry>
    <title>Atom Pilli Robotlar Amok'a Koşuyor</title>
    <link href="http://mesela.der/2003/12/13/atom03"/>
    <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
    <updated>2003-12-13T18:30:02Z</updated>
    <summary>Bir metin.</summary>
  </entry>

</feed>
```

Biraz daha kapsamlı tek girdilik bir Atom Ulama Belgesi:

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">
  <title type="text">dive into mark </title>
  <subtitle type="html">
    A <em>lot</em> of effort
    went into making this effortless
  </subtitle>
  <updated>2005-07-31T12:29:29Z </updated>
  <id>tag:example.org,2003:3 </id>
```

```
<link rel="alternate" type="text/html"
  hreflang="en" href="http://example.org/" />
<link rel="self" type="application/atom+xml"
  href="http://example.org/feed.atom" />
<rights>Copyright (c) 2003, Mark Pilgrim </rights>
<generator uri="http://www.example.com/" version="1.0">
  Example Toolkit
</generator>
<entry>
  <title>Atom draft-07 snapshot </title>
  <link rel="alternate" type="text/html"
    href="http://example.org/2005/04/02/atom" />
  <link rel="enclosure" type="audio/mpeg" length="1337"
    href="http://example.org/audio/ph34r_my_podcast.mp3" />
  <id>tag:example.org,2003:3.2397 </id>
  <updated>2005-07-31T12:29:29Z </updated>
  <published>2003-12-13T08:29:29-04:00 </published>
  <author>
    <name>Mark Pilgrim </name>
    <uri>http://example.org/ </uri>
    <email>f8dy@example.com </email>
  </author>
  <contributor>
    <name>Sam Ruby </name>
  </contributor>
  <contributor>
    <name>Joe Gregorio </name>
  </contributor>
  <content type="xhtml" xml:lang="en"
    xml:base="http://diveintomark.org/">
    <div xmlns="http://www.w3.org/1999/xhtml">
      <p><i>[Update: The Atom draft is finished.] </i> </p>
    </div>
  </content>
</entry>
</feed>
```

1.2. İsim Alanı ve Sürüm

Bu belirtimde açıklanan XML veri biçimi için XML isim alanı [[W3C.REC-xml-names-19990114](http://www.w3.org/TR/2005/Atom)] adresi:

<http://www.w3.org/2005/Atom>

Kullanışlılık açısından, bu veri biçimine "Atom 1.0" olarak da atıfta bulunulabilir. Bu belirtimin metninde bu amaçla "Atom" kullanılmıştır.

1.3. Gösterimsel Uzlaşımlar

Bu belirtim iki yapay olguya göre uyumluluğu açıklar: Atom Ulama Belgeleri ve Atom Girdi Belgeleri. Ek olarak, Atom işlemcilerle ilgili bazı gereksinimlere de yer verir.

Bu belirtim, yukarıda, *İsim Alanı ve Sürüm* (sayfa: 5) bölümünde belirtilen isim alanı adresinin betimleyicisi olarak "atom:" isim alanı örneğini kullanır. İsim alanı örneğinin seçiminin keyfiliğine ve anlambilimsel açıdan önemsizliğine dikkat ediniz.

Atom, XML Bilgi Kümesindeki [*W3C.REC-xml-infosec-20040204*] kurallar kullanılarak belirtilir. Bununla birlikte, bu belirtimde iki bildik kuralı betimlemek için bir kısaltma kullanılmıştır: Eleman Bilgi Öğeleri ve Öznitelik Bilgi Öğeleri'nin betimlediklerini betimlemek üzere "Bilgi Öğesi" terimi kullanılmıştır. Dolayısıyla, bu belirtimdeki her "eleman" terimi Bilgi Kümesi kurallarındaki bir Eleman Bilgi Öğesine bir atıf, her "öznitelik" terimi de bir Öznitelik Bilgi Öğesine bir atıf olacaktır.

Bu belirtimdeki bazı bölümler normdışı bir RELAX NG [*RELAX-NG*] Bütünleşik şema örüntüsüyle gösterilmiştir. Bununla birlikte, bu belirtimin metni bir uyumluluk tanımı içerir. Şemanın tamamı *RELAX NG, Bütünleşik Şema* (sayfa: 28) bölümünde bulunmaktadır.

Bu belirtimin çevirisinde rastlayacağınız *ZORUNLU*, *ÖNERİ* ve *SEÇİMLİK* imleri *BCP 14'ün çevirisinde*^(B11) açıklandığı gibi yorumlanmalıdır.

2. Atom Belgeleri

Bu belirtimde iki çeşit Atom belgesi açıklanmaktadır: Atom Ulama Belgeleri ve Atom Girdi Belgeleri.

Bir Atom Ulama Belgesi, ulama ile ilgili öteverileri ve girdileri (kısmen ya da tamamen) içeren bir Atom ulamından başka birşey değildir. Kökü `atom:feed` elemanıdır.

Bir Atom Girdi Belgesi ise, Atom ulamındaki girdilerden biri olabilecek tek bir Atom girdisi içerir. Kökü `atom:entry` elemanıdır.

```
namespace atom = "http://www.w3.org/2005/Atom"
start = atomFeed | atomEntry
```

Atom belgelerinin her ikisi de XML Bilgi Kümesine göre belirtilmiş, XML 1.0 olarak [*W3C.REC-xml-20040204*] sıradanlaştırılmış ve "application/atom+xml" ortam türü ile betimlenmiştir. Atom Belgeleri iyi biçimli XML olmak zorundadır *ZORUNLU*. Bu belirtim Atom Belgeleri için bir DTD tanımlamaz ve dolayısıyla bu belgelerin geçerliliğinin gerekliliği konusu değildir (XML olarak DTD'ye uyumluluk anlamında geçerlilik).

Atom uluslararası adres (IRI) kullanımını mümkün kılar. Her tektip adresin (URI) [*RFC3986*] aynı zamanda bir uluslararası adres (IRI) olması nedeniyle bir URI, bir IRI'nin betimlediği her yerde kullanılabilir. İki özel durum vardır: (1) bir IRI aynı zamanda bir URI olmadığında, [*RFC3987*]'nin 3.1. bölümündeki adımlar kullanılarak bir URI ile eşlenmeli *ZORUNLU* ve (2) bir IRI'nin bir `atom:id` değeri olarak sunulması halinde ise eşleştirme, bu belgenin *atom:id Karşılaştırması* (sayfa: 17) bölümünde açıklandığı gibi bir karşılaştırma işlemiyle yapılmamalıdır *ZORUNLU*.

Bu belirtim tarafından tanımlanan her elemanın bir `xml:base` özniteliği olabilir *SEÇİMLİK* [*W3C.REC-xmlbase-20010627*]. `xml:base`, bir Atom Belgesinde kullanıldığında, `xml:base` özniteliğinin etkin etki alanı içinde bulunan göreceli başvuruları çözümlemek için temel bir URI (veya IRI) sağlamak suretiyle, [*RFC3986*]'nın 5.1.1. bölümünde açıklanan işlevselliği sunar.

Bu belirtim tarafından tanımlanan her elemanın, içeriği içerildiği elemanın ve çocuklarının anadilini belirten bir `xml:lang` özniteliği olabilir *SEÇİMLİK*. Lisan bağlamı, sadece, bu belirtim tarafından lisana duyarlı ("Language-Sensitive") olarak bildirilmiş elemanlar ve öznitelikler için anlamlıdır. `xml:lang` özniteliğinin yorumlanması ve içeriği ile ilgili gereksinimler XML 1.0 [*W3C.REC-xml-20040204*] belirtiminin 2.12. bölümünde belirtilmiştir.

```
atomCommonAttributes =
  attribute xml:base { atomUri }?,
  attribute xml:lang { atomLanguageTag }?,
  undefinedAttribute*
```

Atom, genişletilebilen bir biçimdir. Atom Belgelerinin nasıl genişletilebildiği bu belgenin [Atom Eklentileri](#) (sayfa: 22) bölümünde açıklanmıştır.

Atom İşlemcileri, bir ulamın içeriğinin sürekli bir görünümünü oluşturken, aynı zamanda, Atom Ulama Belgeleri ve onları başka Atom Ulama Belgeleri ile bir araya getiren Atom Ulama Belgelerinden kaynaklanan durumu kaydedebilir *SEÇİMLİK*. Bir ulamın (eksik girdileri idare ederek, girdileri ve öteverileri güncelleyerek) yeniden oluşturulurken diğerleriyle birleşerek Atom Ulama Belgelerini oluşturması anlamında Atom Ulama Belgeleri bu belirtimin kapsamı dışındadır.

3. Ortak Atom Oluşumları

Atom elemanlarının çoğu bir kaç ortak yapıyı paylaşır. Bu bölümde bu yapılar ve uygun eleman tanımlarına göre elverişli atıflar yapabilmek için bu yapıların gereksinimleri tanımlanmıştır.

Bir eleman, belli bir oluşum çeşidi olarak tanımlandığında, bu oluşumun bu bölümde tanımlanan gereksinimlerinden kendisiyle ilgili olanlarını miras alır.

Bir Tarih oluşumu veya bir uluslararası adres (IRI) içinde boşluk karakterlerinden herhangi birinin olmaması *ZORUNLU* gerektiğine dikkat ediniz. Bazı XML yayınlama gerçeklemleri hatalı olarak değerler arasında boşlukları öntanımlı olarak yerleştirmektedirler. Böyle gerçeklemler geçersiz Atom Belgeleri yayınlacaklardır.

3.1. Metin Oluşumları

Bir Metin (Text) oluşumu insan tarafından okunabilen bir metin içerir. Metin oluşumlarının içeriği lisana duyarlıdır.

```
atomPlainTextConstruct =
  atomCommonAttributes,
  attribute type { "text" | "html" }?,
  text
```

```
atomXHTMLTextConstruct =
  atomCommonAttributes,
  attribute type { "xhtml" },
  xhtmlDiv
```

```
atomTextConstruct = atomPlainTextConstruct | atomXHTMLTextConstruct
```

3.1.1. "type" Özniteliği

Metin oluşumlarının bir "type" özniteliği olabilir *SEÇİMLİK*. Varsa, değeri "text", "html" veya "xhtml" değerlerinden biri olmalıdır *ZORUNLU*. Eğer "type" özniteliği oluşumda yoksa, Atom İşlemciler "text" değerli bir "type" özniteliği varmış gibi davranmalıdırlar *ZORUNLU*. ["atom:content" Elemanı](#) (sayfa: 12) bölümünde tanımlanan `atom:content` elemanının aksine, Metin oluşumlarının "type" özniteliğinin değerleri olarak MIME ortam türleri [[MIMEREG](#)] kullanılmamalıdır *ZORUNLU*.

text değeri

text değeri atom:title örneği:

```
...
<title type="text">
  Küçüktür: &lt;
</title>
...
```

Eğer değer `text` ise Metin oluşumu çocuk eleman içermemelidir *ZORUNLU*. Böyle bir metnin insanlarca okunabilecek tarzda sunulacağı düşünülür. Bu nedenle, Atom İşlemciler boşluk karakterlerinin fazlalık olanlarından kurtulabilir (satırsonu ve satırbaşı karakterleri dahil) ve hizalama, kalın yazıtları gibi teknikleri kullanarak metni gösterebilirler *SEÇİMLİK*.

html değeri

html değeri atom:title örneği:

```
...
<title type="html">
  Küçüktür: &lt;em> &amp;lt; &lt;/em>
</title>
...
```

Eğer "type" özniteliğinin değeri `html` ise, Metin oluşumu çocuk eleman içermemeli *ZORUNLU* ve HTML [[HTML](#)] olarak işlenmeye elverişli olmalıdır *ÖNERİ*. Metin içinde HTML'ye özgü elemanlar kullanılmışsa bunlar HTML'ye özgü biçimde öncelenmelidir *ZORUNLU*; örneğin, "`
`" elemanının yazılışı "`
`" olarak değiştirilmelidir. HTML imleme, önceleme kaldırıldıktan sonra doğrudan bir HTML `<DIV>` elemanı içinde geçerli kalacak şekilde yapılmalıdır *ÖNERİ*. Böyle bir içeriği gösteren Atom İşlemciler kendi gösterimlerinde yardımcı olması için bu imlemeyi kullanabilirler *SEÇİMLİK*.

xhtml değeri

XHTML içerik içinde atom:title örneği:

```
...
<title type="xhtml" xmlns:xhtml="http://www.w3.org/1999/xhtml">
  <xhtml:div>
    Küçüktür: <xhtml:em> &lt; </xhtml:em>
  </xhtml:div>
</title>
...
```

Eğer "type" özniteliğinin değeri `xhtml` ise, Metin oluşumu tek bir XHTML `div` elemanı olmalı *ZORUNLU* ve XHTML olarak işlenmeye elverişli olmalıdır *ÖNERİ*. XHTML `div` elemanının kendisi içeriğin parçası olarak değerlendirilmemelidir *ZORUNLU*. İçeriği gösteren Atom İşlemciler kendi gösterimlerinde yardımcı olması için imlemeyi kullanabilirler *SEÇİMLİK*. "&" ve "<" gibi karakterlerin öncelenmiş sürümleri bu karakterleri temsil eder, dile özgü imlemeyi değil.

Geçerli XHTML içerik örneği:

```
...
<summary type="xhtml">
  <div xmlns="http://www.w3.org/1999/xhtml">
    This is <b>XHTML </b> content.
  </div>
</summary>
...
<summary type="xhtml">
  <xhtml:div xmlns:xhtml="http://www.w3.org/1999/xhtml">
    This is <xhtml:b>XHTML </xhtml:b> content.
  </xhtml:div>
</summary>
...
```


Aşağıdaki örnekte, XHTML isim alanının, belgenin başlarında "xh" önekiyle bağlandığı varsayılmıştır:

```
...
<summary type="xhtml">
  <xh:div>
    This is <xh:b>XHTML </xh:b> content.
  </xh:div>
</summary>
...
```

3.2. Kişilik Oluşumları

Bir Kişilik oluşumu bir şahsı veya tüzelkişiliği tarif eden bir elemandır.

```
atomPersonConstruct =
  atomCommonAttributes,
  (element atom:name { text }
   & element atom:uri { atomUri }?
   & element atom:email { atomEmailAddress }?
   & extensionElement*)
```

Bu belirtim bir Kişilik oluşumunun çocuk elemanlarının birbirlerine göre sıralarıyla ilgili bir atama yapmaz. Kişilik oluşumları ek öteveri elemanlarına izin verir (*Eklenti Elemanları* (sayfa: 22) bölümüne bakınız).

3.2.1. "atom:name" Elemanı

"atom:name" elemanının içeriği kişilik için insan tarafından okunabilen bir isim nakleder. "atom:name" elemanının içeriği lisana duyarlıdır. Kişilik oluşumları ne eksik ne fazla tek bir "atom:name" elemanı içermelidir *ZORUNLU*.

3.2.2. "atom:uri" Elemanı

"atom:uri" elemanının içeriği kişilik ile ilgili bir uluslararası adres (IRI) nakleder. Kişilik oluşumları bir atom:uri elemanı içerebilir *SEÇİMLİK*, ancak birden fazlasını içermemelidir *ZORUNLU*. Bir Kişilik oluşumundaki atom:uri elemanı bir uluslararası adres (IRI) başvurusu [[RFC3987](#)] olmalıdır *ZORUNLU*.

3.2.3. "atom:email" Elemanı

"atom:email" elemanının içeriği kişilik ile ilgili bir eposta adresi nakleder. Kişilik oluşumları bir atom:email elemanı içerebilir *SEÇİMLİK*, ancak birden fazlasını içermemelidir *ZORUNLU*. İçeriği [[RFC2822](#)]'deki "adres-belirtimi" tanımına uygun olmalıdır *ZORUNLU*.

3.3. Tarih Oluşumları

Bir Tarih oluşumu [[RFC3339](#)]'daki "tarih-saat" tanımına uygun olmalıdır *ZORUNLU*. Buna ek olarak tarih gösterimini saat gösteriminden ayırmak için büyük "T" harfi kullanılmalı *ZORUNLU* ve sayısal zaman dilimi saat farkının yokluğunda bir büyük "Z" harfi bulunmalıdır *ZORUNLU*.

```
atomDateConstruct =
  atomCommonAttributes,
  xsd:dateTime
```

Böyle tarih değerleri şu belirtilmelerle uyumlu olarak oluşur: [[ISO.8601.1988](#)], [[W3C.NOTE-datetime-19980827](#)] ve [[W3C.REC-xmlschema-2-20041028](#)].

Tarih oluşumlarına örnekler:

```
<updated>2003-12-13T18:30:02Z </updated>
<updated>2003-12-13T18:30:02.25Z </updated>
<updated>2003-12-13T18:30:02+01:00 </updated>
<updated>2003-12-13T18:30:02.25+01:00 </updated>
```

Tarih değerleri mümkün olduğunca doğru olmalıdır *ÖNERİ*. Örneğin, aynı gün içinde yayınlanan çeşitli girdilere aynı zaman damgasını uygulamak bir yayın sistemi için genellikle uygunsuzluk olurdu.

4. Atom İsim Alanındaki Elemanların Tanımları

4.1. Taşıyıcı Elemanlar

4.1.1. "atom:feed" Elemanı

"atom:feed" elemanı bir Atom Ulama Belgesinin belge (yani, tepe seviyeden) elemanıdır yani, ulam ile ilgili veri ve öteveri için bir taşıyıcıdır. Çocuk elemanları, öteveri elemanlarını takibeden sıfır veya daha fazla sayıda atom:entry elemanından oluşur.

```
atomFeed =
  element atom:feed {
    atomCommonAttributes,
    (atomAuthor*
      & atomCategory*
      & atomContributor*
      & atomGenerator?
      & atomIcon?
      & atomId
      & atomLink*
      & atomLogo?
      & atomRights?
      & atomSubtitle?
      & atomTitle
      & atomUpdated
      & extensionElement*),
    atomEntry*
  }
```

Bu belirtim, atom:feed içindeki atom:entry elemanları arasında bir öncelik belirtmez.

Bu belirtim tarafından tanımlanan çocuk elemanlar (bu elemanların bazılarının varlığı gereklidir):

- atom:feed elemanının atom:entry çocuklarının her biri en azından birer atom:author elemanı içermedikçe, atom:feed elemanlarının bir veya daha fazla sayıda atom:author elemanı içermesi gerekir *ZORUNLU*.
- atom:feed elemanları herhangi bir sayıda atom:category elemanı içerebilir *SEÇİMLİK*.
- atom:feed elemanları herhangi bir sayıda atom:contributor elemanı içerebilir *SEÇİMLİK*.
- atom:feed elemanlarının birden fazla atom:generator elemanı içermemesi gerekir *ZORUNLU*.
- atom:feed elemanlarının birden fazla atom:icon elemanı içermemesi gerekir *ZORUNLU*.
- atom:feed elemanlarının birden fazla atom:logo elemanı içermemesi gerekir *ZORUNLU*.

- `atom:feed` elemanlarının, ne eksik ne fazla tam olarak bir tane `atom:id` elemanı içermesi gerekir *ZORUNLU*.
- `atom:feed` elemanları, "self" değerli bir `rel` özniteliği olan tek bir `atom:link` elemanı içermelidir *ÖNERİ*. Bu `atom:link` elemanının içeriği Atom Ulama Belgelerinin alınacağı, Atom ulamını sunan yerin tercihli adresidir (URI).
- `atom:feed` elemanları, "alternate" değerli bir `rel` özniteliği olup aynı zamanda `type` ve `hreflang` özniteliklerinde de aynı değerlere sahip birden fazla `atom:link` elemanı içermemelidir *ZORUNLU*.
- `atom:feed` elemanları, yukarıda açıklananlar dışında başka `atom:link` elemanları içerebilir *SEÇİMLİK*.
- `atom:feed` elemanlarının birden fazla `atom:rights` elemanı içermemesi gerekir *ZORUNLU*.
- `atom:feed` elemanlarının birden fazla `atom:subtitle` elemanı içermemesi gerekir *ZORUNLU*.
- `atom:feed` elemanlarının, ne eksik ne fazla tam olarak bir tane `atom:title` elemanı içermesi gerekir *ZORUNLU*.
- `atom:feed` elemanlarının, ne eksik ne fazla tam olarak bir tane `atom:updated` elemanı içermesi gerekir *ZORUNLU*.

Eğer bir Atom Ulama Belgesinde aynı `atom:id` değerli çok sayıda `atom:entry` elemanı görünüyorsa, bunlar aynı girdiyi ifade ediyorlar demektir. Bunların `atom:updated` zaman damgaları farklı olmalıdır *ÖNERİ*.

Eğer bir Atom Ulama Belgesi aynı `atom:id` ile çok sayıda girdi içeriyorsa, Atom İşlemcileri bunların tümünü göstermeyi seçebilecekleri gibi bir kısmını da gösterebilirler. Tipik bir davranış, en yeni `atom:updated` zaman damgasını içeren girdiyi göstermek olurdu.

4.1.1.1. Metinsel İçeriğin Sağlanması

Deneyimler, ulamların metinsel içeriği içermesinin içermemesinden genelde daha faydalı olduğunu öğretmektedir. Bazı uygulamalar (tamamen metin dizinleri üretenler gibi), işlevlerini yerine getirebilmek için az da olsa metin veya (X)HTML içerik gerektirir. Ulam üreticilerinin bu gibi hususlardan haberi olması gerekir. Her `atom:entry` elemanının, içerdiği takdirde `atom:title` ve `atom:content` elemanlarının içeriğinin boş olmaması ve `atom:content` içermemesi halinde, boş olmayan bir `atom:summary` elemanı içermesi tercih edilmelidir. Bununla birlikte, bir `atom:summary` elemanının yokluğu bir hata değildir ve Atom İşlemcilerinin böyle bir yokluk durumunda başarısızlık göstermemesi gerekir *ZORUNLU*.

4.1.2. "atom:entry" Elemanı

`atom:entry` elemanı, girdi ile ilgili veri ve öteveri için bir taşıyıcı görevi görmesi bakımından başlıbaşına bir girdi içerir. Bir `atom:feed` elemanının bir çocuğu olarak karşımıza çıkabileceği gibi tek başına bir Atom Girdi Belgesinin belge (yani, tepe seviyeden) elemanı olarak da karşımıza çıkabilir.

```
atomEntry =
  element atom:entry {
    atomCommonAttributes,
    (atomAuthor*
      & atomCategory*
      & atomContent?
      & atomContributor*
      & atomId
      & atomLink*
      & atomPublished?
      & atomRights?)
```

```

    & atomSource?
    & atomSummary?
    & atomTitle
    & atomUpdated
    & extensionElement*)
  }

```

Bu belirtim `atom:entry` elemanının çocukları arasında bir öncelik belirtmez.

Bu belirtim tarafından tanımlanan çocuk elemanlar (bu elemanların bazılarının varlığı gereklidir):

- Bir `atom:entry` elemanı içinde, bir `atom:author` elemanı içeren bir `atom:source` elemanı veya bir Atom Ulama Belgesi içinde, kendi `atom:author` elemanını içeren bir `atom:feed` elemanı olmadıkça `atom:entry` elemanlarının bir veya daha fazla sayıda `atom:author` elemanı içermesi gerekir ***ZORUNLU***.
- `atom:entry` elemanları herhangi bir sayıda `atom:category` elemanı içerebilir ***SEÇİMLİK***.
- `atom:entry` elemanları birden fazla sayıda `atom:content` elemanı içermemelidir ***ZORUNLU***.
- `atom:entry` elemanları herhangi bir sayıda `atom:contributor` elemanı içerebilir ***SEÇİMLİK***.
- `atom:entry` elemanlarının, ne eksik ne fazla tam olarak bir tane `atom:id` elemanı içermesi gerekir ***ZORUNLU***.
- Hiç `atom:content` çocuğu olmayan `atom:entry` elemanlarının, değeri "alternate" olan bir `rel` özniteliğine sahip en az bir `atom:link` elemanı olmalıdır ***ZORUNLU***.
- `atom:entry` elemanları, "alternate" değerli bir `rel` özniteliği olup aynı zamanda `type` ve `hreflang` öznitiliklerinde de aynı değerlere sahip birden fazla `atom:link` elemanı içermemelidir ***ZORUNLU***.
- `atom:entry` elemanları, yukarıda açıklananlar dışında başka `atom:link` elemanları içerebilir ***SEÇİMLİK***.
- `atom:entry` elemanları birden fazla sayıda `atom:published` elemanı içermemelidir ***ZORUNLU***.
- `atom:entry` elemanları birden fazla sayıda `atom:rights` elemanı içermemelidir ***ZORUNLU***.
- `atom:entry` elemanları birden fazla sayıda `atom:source` elemanı içermemelidir ***ZORUNLU***.
- `atom:entry` elemanları şu iki durumdan birinde bir `atom:summary` elemanı içermelidir ***ZORUNLU***:
 - `atom:entry` elemanı, bir "src" özniteliğine sahip (bu nedenle boştur) bir `atom:content` elemanı içeriyordur.
 - `atom:entry` elemanı Base64 kodlu bir içeriğe sahiptir; yani, `atom:content` elemanının "type" özniteliği bir MIME ortam türüdür [*MIMEREĞ*], ama bir XML ortam türü [*RFC3023*] değildir, "text/" ile başlamıyor ve "/xml" veya "+xml" ile bitmiyordur.
- `atom:entry` elemanları birden fazla sayıda `atom:summary` elemanı içermemelidir ***ZORUNLU***.
- `atom:entry` elemanlarının, ne eksik ne fazla tam olarak bir tane `atom:title` elemanı içermesi gerekir ***ZORUNLU***.
- `atom:entry` elemanlarının, ne eksik ne fazla tam olarak bir tane `atom:updated` elemanı içermesi gerekir ***ZORUNLU***.

4.1.3. "atom:content" Elemanı

`atom:content` elemanı bir girdiyi ya içerir ya da içeriğe bir bağlantı olur. `atom:content` elemanının içeriği lisana duyarlıdır.

```

atomInlineTextContent =
  element atom:content {

```

```

        atomCommonAttributes,
        attribute type { "text" | "html" }?,
        (text)*
    }

atomInlineXHTMLContent =
    element atom:content {
        atomCommonAttributes,
        attribute type { "xhtml" },
        xhtmlDiv
    }

atomInlineOtherContent =
    element atom:content {
        atomCommonAttributes,
        attribute type { atomMediaType }?,
        (text|anyElement)*
    }

atomOutOfLineContent =
    element atom:content {
        atomCommonAttributes,
        attribute type { atomMediaType }?,
        attribute src { atomUri },
        empty
    }

atomContent = atomInlineTextContent
| atomInlineXHTMLContent
| atomInlineOtherContent
| atomOutOfLineContent

```

"type" Özneliği

`atom:content` elemanında `type` özneliğinin değeri `"text"`, `"html"` veya `"xhtml"` olabilir *SEÇİMLİK*. Bu mümkün olmazsa, değerin bir MIME ortam türü sözdizimine uygun olması *ZORUNLU* ama birleşik bir tür olmaması gerekir *ZORUNLU* ([[MIMEREG](#)] belgesinin 4.2.6. bölümüne bakınız). Eğer ne `type` ne de `src` özneliği sağlanmışsa, Atom İşlemcileri `"text"` değerli bir `type` özneliği varmış gibi davranmalıdırlar.

"src" Özneliği

`atom:content` bir `src` özneliğine sahip olabilir *SEÇİMLİK*, ancak, değerinin bir uluslararası adres (IRI) [[RFC3987](#)] olması gerekir *ZORUNLU*. `src` özneliği sağlanmışsa, `atom:content` içeriği boş olmalıdır *ZORUNLU*. Atom İşlemcileri içeriği almak için bu IRI'yi kullanabilir *SEÇİMLİK* veya uzak içeriği yoksaymayı ya da yerel içerikten farklı bir anlamda içermeyi seçebilir *SEÇİMLİK*.

`src` özneliği sağlanmışsa, `type` özneliği de sağlanmalı *ÖNERİ* ancak değeri `"text"`, `"html"` veya `"xhtml"` değil bir MIME ortam türü [[MIMEREG](#)] olmalıdır *ZORUNLU*. Değer tavsiye niteliğindedir; içeriği sağlayan sunucu ayrıca bir ortam türü de sağlıyorsa, sunucu tarafından sağlanan ortam türü yetkindir.

4.1.3.1. İşlem Modeli

Atom Belgelerinin aşağıdaki kurallara uygun olması gerekir ***ZORUNLU***. Atom İşlecilerinin `atom:content` içeriğini uygulanabilir olan ilk kurala göre yorumlaması gerekir ***ZORUNLU***.

1. `type` değeri `"text"` ise, `atom:content` çocuk elemanlar içermemelidir ***ZORUNLU***. Böyle bir metnin insanlarca okunabilecek şekilde içerileceği düşünülür. Bu nedenle, Atom İşlecileri fazla boşlukları atabilir ve metni hizalamak ve özel yazıtları kullanmak gibi tekniklerle göze daha hoş gösterebilir ***SEÇİMLİK***.
2. `type` değeri `"html"` ise, `atom:content` çocuk elemanlar içermemeli ***ZORUNLU*** ve içerik HTML [[HTML](#)] olarak işlenmeye uygun olmalıdır ***ÖNERİ***. HTML imleme öncelenmelidir; örneğin, `"
"` yerine `"
"` yazılmalıdır. HTML imleme, önceleme kaldırıldıktan sonra doğrudan bir HTML `<DIV>` elemanı içinde geçerli kalacak şekilde yapılmalıdır ***ÖNERİ***. İçeriği gösteren Atom İşlecileri imlemeyi gösterime yardımcı olarak kullanabilirler ***SEÇİMLİK***.
3. `type` değeri `"xhtml"` ise, `atom:content` çocuk eleman olarak tek bir XHTML `div` elemanı içermelidir ***ZORUNLU***. İçerik, XHTML olarak işlenmeye elverişli olmalıdır ***ÖNERİ***. XHTML `div` elemanının kendisi içeriğin parçası olarak ele alınmamalıdır ***ZORUNLU***. İçeriği gösteren Atom İşlecileri imlemeyi gösterime yardımcı olarak kullanabilirler ***SEÇİMLİK***. `"&"` ve `"<"` gibi karakterlerin öncelenmiş sürümleri imleme belirtmez, karakterlerin kendilerini ifade ederler.
4. `type` değeri bir XML ortam türü [[RFC3023](#)] ise veya `"+xml"` veya `"/xml"` (harf büyüklüğüne du-yarsız) ile bitiyorsa, `atom:content` çocuk elemanlar içerebilir ***SEÇİMLİK*** ancak, belirtilen ortam türünde işlenmeye elverişli olmalıdır ***ÖNERİ***. Eğer `"src"` özniteliği sağlanmamışsa, bu normalde, `atom:content`'in belirtilen türdeki XML belgenin kök elemanı olarak tek bir çocuk eleman içerdiği anlamına gelir.
5. `type` değeri `"text/"` (harf büyüklüğüne duyarsız) ile başlıyorsa, `atom:content` çocuk elemanlar içeremez ***ZORUNLU***.
6. `type` özniteliğinin diğer tüm değerleri için `atom:content` içeriğinin [[RFC3548](#)] belgesinin 3. bölümünde açıklandığı gibi Base64 kodlu geçerli bir içerik olması gerekir ***ZORUNLU***. Kodu çözüldüğünde, belirtilen ortam türünde işlenmeye elverişli olmalıdır ***ÖNERİ***. Bu durumda, `atom:content` içinde, Base64 kod-lamadaki karakterlerin önüne ve ardına boşluklar eklenmiş ve satırlar tek bir satırsonu karakteri (U+000A) ile ayrılmış olabilir ***SEÇİMLİK***.

4.1.4. Örnekler

XHTML içinde:

```
...
<content type="xhtml">
  <div xmlns="http://www.w3.org/1999/xhtml">
    This is <b>XHTML </b> content.
  </div>
</content>
...
<content type="xhtml">
  <xhtml:div xmlns:xhtml="http://www.w3.org/1999/xhtml">
    This is <xhtml:b>XHTML </xhtml:b> content.
  </xhtml:div>
</content>
...
```

Aşağıdaki örnekte, XHTML isim alanının, belgenin başlarında "xh" öneğine bağlandığı varsayılmıştır:

```

...
<content type="xhtml">
  <xh:div>
    This is <xh:b>XHTML </xh:b> content.
  </xh:div>
</content>
...

```

4.2. Öteveri Elemanları

4.2.1. "atom:author" Elemanı

"atom:author" elemanı, girdinin veya ulamın yazarını belirten bir Kişilik oluşumdur.

```
atomAuthor = element atom:author { atomPersonConstruct }
```

Eğer bir atom:entry elemanının atom:author çocukları yoksa, atom:source çocuklarının atom:author çocukları uygulamaya konu olurlar. Bir Atom Ulama Belgesinde, girdi için eğer yukarıda açıklanan yerlerde hiç atom:author elemanları yoksa içerilen atom:feed elemanının atom:author çocukları uygulamaya konu olurlar.

4.2.2. "atom:category" Elemanı

"atom:category" elemanı bir girdi veya ulam ile ilişkili bir kategori hakkında bilgi nakleder. Bu belirtim bu elemanın içeriğine (varsa) hiçbir anlam atamaz.

```

atomCategory =
  element atom:category {
    atomCommonAttributes,
    attribute term { text },
    attribute scheme { atomUri }?,
    attribute label { text }?,
    undefinedContent
  }

```

"term" Özniteliği

term özniteliği, girdi veya ulamın içinde bulunduğu kategoriyi betimleyen bir dizgedir. Her "atom:category" elemanının bir term özniteliği olmalıdır *ZORUNLU*.

"scheme" Özniteliği

scheme özniteliği, bir kategorileme şemasını betimleyen bir uluslararası adrestir (IRI). Her "atom:category" elemanının bir scheme özniteliği olabilir *SEÇİMLİK*.

"label" Özniteliği

label özniteliği, son kullanıcı uygulamalarında göstermek için insanlarca okunabilen bir etiket içerir. label özniteliğinin içeriği lisana duyarlıdır. "&," ve "<" gibi öğeler onlara karşılık gelen karakterleri (sırayla "&" ve "<") temsil ederler, bir imlemeyi değil. Her "atom:category" elemanının bir label özniteliği olabilir *SEÇİMLİK*.

4.2.3. "atom:contributor" Elemanı

"atom:contributor" elemanının içeriği, ulam veya girdiye katkıda bulunan bir şahsı veya tüzelkişiyi belirten bir Kişilik oluşumdur.

```
atomContributor = element atom:contributor { atomPersonConstruct }
```

4.2.4. "atom:generator" Elemanı

"atom:generator" elemanının içeriği, hata ayıklama veya başka bir amaçla bir ulam üretmekte kullanılacak aracı betimler.

```
atomGenerator = element atom:generator {
    atomCommonAttributes,
    attribute uri { atomUri }?,
    attribute version { text }?,
    text
}
```

Kullanıldığında bu elemanın içeriği ulam üreticinin insanlarca okunabilen isminden oluşan bir dizge olmalıdır *ZORUNLU*. "&" ve "<" gibi öğeler onlara karşılık gelen karakterleri (sırayla "&" ve "<") temsil ederler, bir imlemeyi değil.

atom:generator elemanı, değeri bir IRI [[RFC3987](#)] başvurusu olması gereken *ZORUNLU* bir "uri" özniteliği içerebilir *SEÇİMLİK*. Başvuru çözümlendiğinde sonuçlanan URI (gerekliyse, bir IRI'den eşlenerek) üreticinin amacına uygun bir sunum sağlamalıdır *ÖNERİ*.

atom:generator elemanı, ulam üreticinin sürümünü belirten bir "version" özniteliği içerebilir *SEÇİMLİK*.

4.2.5. "atom:icon" Elemanı

atom:icon elemanı, bir ulam için görsel simgeli betimleme sağlayan bir imgeyi betimleyen bir IRI [[RFC3987](#)] başvurusu içerir.

```
atomIcon = element atom:icon {
    atomCommonAttributes,
    (atomUri)
}
```

Bu imge, bir en-boy oranına sahip olmalı *ÖNERİ* ve küçük boyutta gösterilmeye elverişli olmalıdır *ÖNERİ*.

4.2.6. "atom:id" Elemanı

"atom:id" elemanı, bir ulam veya girdi için kalıcı ve evrensel olarak eşsiz bir betimleyici nakleder.

```
atomId = element atom:id {
    atomCommonAttributes,
    (atomUri)
}
```

İçeriği [[RFC3987](#)]’ye göre tanımlanmış bir IRI olmalıdır *ZORUNLU*. "IRI" tanımının görelî başvuruları dışladığına dikkat ediniz. "IRI" başvuru oluşturabilen bir şema kullanabildiğinden, Atom İşlemcileri onu çözümlenerek başvuru olabilir varsaymamalıdır *ZORUNLU*.

Bir Atom belgesi, yeniden konumlandığında, yansındığında, ödünleşerek paylaşıldığında, yeniden yayınlandığında, ihraç ya da ithal edildiğinde, belgenin atom:id elemanının içeriği değişmemelidir *ZORUNLU*. Başka bir deyişle, bir atom:id içeriğinin, belli bir Atom ulamı veya girdisinin tüm örnekleriyle ilgisi sürmeli; düzeltmeler aynı atom:id içerikli kalmalıdır. Her atom:id elemanının ilişkili olduğu özkaynak ile birlikte saklanması önerilir.

Bir atom:id elemanının içeriğinin eşsizliğinden emin olunacak bir yöntemle oluşturulması gerekir *ZORUNLU*.

URI'lere eşlendiğinde veya başvuru olarak çözümlendiğinde eşdeğer olması gereken IRI'ler arasındaki belirsizlik riskinden dolayı, atom:id elemanları üretilirken aşağıdaki normalleştirme stratejisi uygulanmalıdır *ÖNERİ*:

- Şema küçük harflerden oluşur.
- Varsa, konak küçük harflerden oluşur.
- Gerekli olduğu yerde sadece kısmen kodlama yapılır.
- Kimi kodlama yapılırken A'dan F'ye kadar büyük harfler kullanılır.
- Yollarda noktalarla ayrılmış parçaların görünmesinden kaçınılır.
- Bir öntanımlı yetkili tanımlayan şemalar için, eğer öntanımlama isteniyorsa boş yetkili kullanılır.
- Bir "/" yoluna eşdeğer olarak boş bir yol tanımlayan şemalar için "/" kullanılır.
- Bir port tanımlayan şemalar için, eğer öntanımlı bir tane isteniyorsa, hiçbir port belirtilmez.
- Boş veridilimi betimleyicilerinden ve sorgularından kaçınılır.
- IRI'nin tüm bileşenlerinin uygun karakterlerle (örn, NFC veya NFKC kullanarak) normalleştirildiğinden emin olunur.

4.2.6.1. **atom:id** Karşılaştırması

`atom:id` elemanlarının örnekleri bir ulam veya girdinin daha önceki biriyle aynı olup olmadığını saptamak için karşılaştırılabilir. İşlemciler `atom:id` elemanlarını karakter karakter, harf büyüklüğüne duyarlı olarak karşılaştırmalıdır ***ZORUNLU***. Karşılaştırma işlemleri yalnızca IRI karakter dizgelerine dayanmalı ***ZORUNLU*** ve IRI'leri başvuru olarak çözümlemelere veya onlara eşlenen URI'lere bel bağlanmamalıdır ***ZORUNLU***.

Sonuç olarak, aynı özkaynağa çözümlenen fakat karakter karakter birbirinin eşi olmayan iki IRI, betimleyici karşılaştırması bakımından farklı sayılırlar.

Örneğin, bunlar aslında birbirlerinden sadece harf büyüklükleri bakımından farklı olan dört ayrı betimleyicidir:

```
http://www.example.org/thing
http://www.example.org/Thing
http://www.EXAMPLE.org/thing
HTTP://www.example.org/thing
```

Benzer şekilde, bunlar da, IRI % öncelemesi karşılaştırmının amacına uygun olarak önem kazandığından aslında üç ayrı betimleyicidir:

```
http://www.example.com/~bob
http://www.example.com/%7ebob
http://www.example.com/%7Eb0b
```

4.2.7. **"atom:link"** Elemanı

Bir `"atom:link"` elemanı bir ulam veya girdiden bir sanaldoku özkaynağına bir başvuru tanımlar. Bu belirtim, bu elemanın içeriğine bir anlam atfetmez.

```
atomLink =
  element atom:link {
    atomCommonAttributes,
    attribute href { atomUri },
    attribute rel { atomNCName | atomUri }?,
    attribute type { atomMediaType }?,
    attribute hreflang { atomLanguageTag }?,
    attribute title { text }?,
    attribute length { text }?,
    undefinedContent
  }
```

"href" Özniteliği

"href" özniteliği bağlantının IRI'sini içerir. Her `atom:link` elemanının, değeri bir IRI başvurusu [[RFC3987](#)] olması gereken *ZORUNLU* bir `href` özniteliği olmalıdır *ZORUNLU*.

"rel" Özniteliği

Her `atom:link` elemanının, bağlantı ilişki türünü belirten bir `rel` özniteliği olabilir *SEÇİMLİK*. Eğer bir `rel` özniteliği belirtilmemişse, `atom:link` elemanı için bağlantı ilişki türü "alternate" imiş gibi yorum yapılmalıdır *ZORUNLU*.

`rel` değeri, boş olmayan ve, ya bir "." içermeksizin sıfır uzunlukta olmayan bir veribölütü ya da [[RFC3987](#)]den biri IRI ürünü olan bir dizge olmalıdır *ZORUNLU*. Basit bir isimden başka bir görelî başvuru kullanımına izin verilmediğine dikkat ediniz. Eğer bir isim verilmişse, gerçeklenimlerin, bağlantı ilişki türünün IANA Bağlantı İlişkileri Sicilinde (bkz, [IANA Değerlendirmeleri](#) (sayfa: 23)) kayıtlı isimle ve dolayısıyla "http://www.iana.org/assignments/relation/" dizgesine `rel` değeri eklenerek oluşturulan IRI'yle aynı olduğunu varsaymaları gerekir *ZORUNLU*. `rel` değeri bağlantının anlamını açıklar fakat Atom İşlemciler her hangi bir davranışsal gereksinim dayatmaz.

Bu belge Bağlantı İlişkileri Sicili için beş ilkendirme değeri tanımlar:

1. "alternate" değeri, `href` özniteliğinin değerindeki IRI'nin taşıyıcı eleman tarafından açıklanan özkaynağın bir başka sürümünü betimlediği anlamına gelir.
2. "related" değeri, `href` özniteliğinin değerindeki IRI'nin taşıyıcı eleman tarafından açıklanan özkaynakla ilgili bir özkaynağı betimlediği anlamına gelir. Örneğin, "http://search.example.com" adresindeki bir arama motorunun başarımının anlatıldığı bir sanalyöre ulamında `atom:feed` şöyle bir çocuk içermelidir:

```
<link rel="related" href="http://search.example.com/" />
```

Eşdeğer bir bağlantı, aynı arama motorundan bahsedilen bir `atom:entry`'nin çocuğu olarak da karşımıza çıkabilir.

3. "self" değeri, `href` özniteliğinin değerindeki IRI'nin taşıyıcı elemene eşdeğer bir özkaynağı betimlediği anlamına gelir.
4. "enclosure" değeri, `href` özniteliğinin değerindeki IRI'nin oldukça büyük olması muhtemel ve bu bakımdan özel olarak ele alınması gerekebilecek bir ilişkili özkaynağı betimlediği anlamına gelir. `rel="enclosure"` içeren bir `atom:link` elemanı için bir `length` özniteliği sağlanmış olmalıdır *ÖNERİ*.
5. "via" değeri, `href` özniteliğinin değerindeki IRI'nin taşıyıcı elemenda sağlanan bilginin kaynağı olan bir özkaynağı betimlediği anlamına gelir.

"type" Özniteliği

`atom:link` elemanının "type" özniteliğinin değeri tavsiye niteliğinde bir ortam türüdür: `href` özniteliğinin değeri başvuru olarak çözümlendiğinde döneceği umulan gösterimin türü hakkında bir ipucudur. `type` özniteliğinin gösterim ile dönen asıl ortam türünü değiştiren nitelikte olmadığına dikkat ediniz. Her `atom:link` elemanının, değeri bir MIME ortam türü [[MIMEREG](#)] sözdizimine uygun olması gereken *ZORUNLU* bir `type` özniteliği olabilir *SEÇİMLİK*.

"hreflang" Özniteliği

"hreflang" özniteliğinin içeriği `href` özniteliğinin gösterdiği özkaynağın dilini açıklar. `rel="alternate"` ile beraber kullanıldığında, girdinin çevrilmiş bir sürüm olduğu anlamına gelir. Her `atom:link` elemanının, değeri bir lisan yaftası [[RFC3066](#)] olması gereken *ZORUNLU* bir `hreflang` özniteliği olabilir *SEÇİMLİK*.

"title" Özniteliği

"title" özniteliği bağlantı hakkında insanlarca okunabilir bilgi nakleder. "title" özniteliğinin içeriği lisana duyarlıdır. "&" ve "<" gibi öğeler onlara karşılık gelen karakterleri (sırayla "&" ve "<") temsil ederler, bir imlemeyi değil. Her `atom:link` elemanının bir `title` özniteliği olabilir *SEÇİMLİK*.

"length" Özniteliği

"length" özniteliği bağlantılı içeriğin sekiz bitlik baytlar cinsinden tavsiye niteliğindeki uzunluğunu belirtir; `href` özniteliğindeki IRI bir başvuru olarak çözümlendiğinde veya bir URI ile eşleştiğinde dönen verinin uzunluğu hakkında bir ipucudur. `length` özniteliğinin, taşıyıcı protokol tarafından raporlanan asıl veri uzunluğunu değiştirir nitelikte olmadığına dikkat ediniz. Her `atom:link` elemanının bir `length` özniteliği olabilir *SEÇİMLİK*.

4.2.8. "atom:logo" Elemanı

"atom:logo" elemanının içeriği, ulam için görsel kimlik sağlayan bir imgeyi betimleyen bir IRI başvurusudur [RFC3987].

```
atomLogo = element atom:logo {
    atomCommonAttributes,
    (atomUri)
}
```

İmgenin en-boy oranı 2'ye 1 olmalıdır *ÖNERİ*.

4.2.9. "atom:published" Elemanı

"atom:published" elemanı girdinin yaşamının başlarındaki bir olayla ilgili bir anı belirten bir Tarih oluşumdur.

```
atomPublished = element atom:published { atomDateConstruct }
```

Genellikle, `atom:published` özkaynağın ilk işe yarar oluşu veya ilk oluşturuluşu ile ilgilidir.

4.2.10. "atom:rights" Elemanı

"atom:rights" elemanı bir ulam veya girdinin içindeki veya üzerindeki fikri haklar hakkında bilgi nakleden bir Metin oluşumdur.

```
atomRights = element atom:rights { atomTextConstruct }
```

`atom:rights` elemanı, makineler tarafından okunabilir lisanslama bilgisini nakletmekte kullanılmamalıdır *ÖNERİ*.

Eğer bir `atom:entry` elemanı bir `atom:rights` elemanı içermiyorsa, taşıyıcı `atom:feed` elemanının bir `atom:rights` elemanı eğer varsa onun içeriğinin girdiye uygulanacağı varsayılır.

4.2.11. "atom:source" Elemanı

Eğer bir `atom:entry` elemanı bir ulamdan diğerine kopyalanmışsa, kaynak `atom:feed` elemanının öteverisi (`atom:entry` elemanlarından başka `atom:feed`'in tüm çocuk elemanları) kopya girdiye bir `atom:source` elemanı eklenerek onun içinde veya bir `atom:source` elemanı zaten varsa ona çocukları olarak eklenerek korunabilir *SEÇİMLİK*. Eğer kaynak `atom:feed` elemanı, `atom:author`, `atom:contributor`, `atom:rights` veya `atom:category` çocuk elemanlarından bazılarını içeriyorsa ve bu çocuk elemanlar kaynak `atom:entry` içinde mevcut değilse böyle öteverinin de korunması gerekir *ÖNERİ*.

```
atomSource =
    element atom:source {
```

```

    atomCommonAttributes,
    (atomAuthor*
     & atomCategory*
     & atomContributor*
     & atomGenerator?
     & atomIcon?
     & atomId?
     & atomLink*
     & atomLogo?
     & atomRights?
     & atomSubtitle?
     & atomTitle?
     & atomUpdated?
     & extensionElement*)
  }

```

`atom:source` elemanı, bir girdinin kaynak ulamı hakkındaki bilgi kalırken farklı ulamlardaki girdilerin ortaklaşılmasını mümkün kılmak için tasarlanmıştır. Bu sebeple, böyle ortak değerleri işleme sokan Atom İşlemcilerinin `atom:source` elemanında en azından ulam seviyesinde gerekli Öteveri elemanlarını (`atom:id`, `atom:title` ve `atom:updated`) içermesi gerekir *ÖNERİ*.

4.2.12. "atom:subtitle" Elemanı

"`atom:subtitle`" elemanı, bir ulam için altbaşlık veya insanlarca okunabilen bir açıklama nakleden bir Metin oluşumdur.

```
atomSubtitle = element atom:subtitle { atomTextConstruct }
```

4.2.13. "atom:summary" Elemanı

"`atom:summary`" elemanı, bir girdiden kısa bir alıntı veya bir özet nakleden bir Metin oluşumdur.

```
atomSummary = element atom:summary { atomTextConstruct }
```

Atom İşlemcilerinin olmadığı halde yararlı bir özet varmış gibi kabul edecek olmaları sebebiyle `atom:summary` elemanının, `atom:title` veya `atom:content` elemanını yineleyen bir içeriği nakletmesi tavsiye edilmez.

4.2.14. "atom:title" Elemanı

"`atom:title`" elemanı, bir ulam veya girdi için insanlarca okunabilen başlık nakleden bir Metin oluşumdur.

```
atomTitle = element atom:title { atomTextConstruct }
```

4.2.15. "atom:updated" Elemanı

"`atom:updated`" elemanı, bir girdi veya ulam yayıncı için önemli sayılabilecek değişikliklere uğradığında bunun en sonuncusunun tarihini belirten bir Tarih oluşumdur. Bu bakımdan tüm değişiklikler `atom:updated` değerinde bir değişiklik ile sonuçlanmaz.

```
atomUpdated = element atom:updated { atomDateConstruct }
```

Yayıncılar bu elemanın değerini biraz ileri alabilirler *SEÇİMLİK*.

5. Atom Belgelerinin Güvenliği

Atom, XML'e dayalı bir biçim olduğundan içeriğinin güvenliği için mevcut XML güvenlik mekanizmaları kullanılabilir.

Ulamaların ve/veya girdilerin üreticileri ve ulamları ve/veya girdileri toplayan araçlar şifreleme ve imzalama için sebepler olduğundan yoksa içeriğin korunmamış olacağından bahsedebilirler. Örneğin, bir tüccar ürün için indirim kuponu içeren bir iletiyi sayısal olarak imzalayabilir. Müşteri ekstrelerini teslim ederken Atom kullanan bir banka, müşterisinin parasal bilgilerini korumak ve onların doğruluğunu kanıtlamak için bu iletileri şifrelemeyi ve imzalamayı çok isteyebilir. Araçlar, edilgen bir gözlemcinin alıcıların ilgilendiği konular hakkında fikir edinmemesi için topladıkları ulamları şifrelemek isteyebilirler. Böyle daha pek çok örnek verilebilir.

Bu bölümdeki algoritma gereksinimleri Atom İşlemcilerle ilişkindir. Bir alıcının en azından belirtilen şifreleme algoritmasını kullanarak iletileri elde edebilir olmasını gerektirirler. Bu gereksinimler göndericinin seçebileceği algoritmaları sınırlamaz.

5.1. Sayısal İmzalar

Bir Atom Belgesinin tepe elemanı (yani, Atom Ulama Belgesi için `atom:feed`, Atom Girdi Belgesi için `atom:entry`) veya bir `atom:entry` elemanı, [[W3C.REC-xmlsig-core-20020212](#)] XML-İmza Sözdizimi ve İşlemleri Belgesi tarafından açıklandığı gibi bir Zarflı İmza içerebilir *SEÇİMLİK*.

Atom İşlemcileri böyle bir imzayı doğrulama yeteneğine sahip olmadıkları gerekçesi ile böyle bir Atom Belgesini reddetmemelidirler *ZORUNLU*; işleme devam etmelidirler *ZORUNLU* ancak, kullanıcıyı imzayı doğrulayamadıkları konusunda bilgilendirmelidirler *SEÇİMLİK*.

Başka bir deyişle, belge elemanının bir çocuğu olarak yerel ismi "Signature" olan ve isim alanı olarak "http://www.w3.org/2000/09/xmlsig#" URI'li bir elemanın varlığında, Atom İşlemci onun varlığından dolayı başarısızlığa uğramamalıdır *ZORUNLU*.

Atom Belgesindeki diğer elemanlar kendi tanımlarında açıkça böyle bir yeteneği belirtmedikleri takdirde imzalanmamalıdır *ZORUNLU*.

[[W3C.REC-xmlsig-core-20020212](#)] belgesinin 6.5.1. bölümü Meşru Olarak XML [[W3C.REC-xml-c14n-20010315](#)] için destek gerektirir. Bununla birlikte, çoğu gerçeklenimci, imzalı XML belgelerin diğer XML belgelerin imzalarını bozdukları gerekçesiyle onu kullanmazlar. Bu nedenle, imzalı Atom Belgelerinin doğrulayan Atom İşlemcilerinin, Ayrıcalıklı XML Meşrulaştırma [[W3C.REC-xml-exc-c14n-20020718](#)] belgesinde belirtildiği gibi, "http://www.w3.org/2001/10/xml-exc-c14n#" URI'si tarafından betimlenen ayrıcalıklı XML meşrulaştırma yöntemi ile meşrulaştırılabilmeleri gerekir *ZORUNLU*.

Toplayıcılar, vb. gibi araçlar kendi `atom:source` elemanını içermeyen girdiler için bir `atom:source` elemanı eklemek gereği duyabilirler. Böyle bir girdinin imzalı olması halinde, ekleme imzayı bozacaktır. Bu nedenle, tek tek imzalanmış girdilerin yayıncıları kesinlikle, bu girdiler imzalanmadan önce onlara bir `atom:source` elemanı eklemeyi düşünmelidirler *ÖNERİ*. Gerçeklenimcilerin ayrıca, "xml:" isim alanında meşrulaştırma ile etkileşen imleme kullanımı ile ilgili konulardan haberdar olması gerekir.

[[W3C.REC-xmlsig-core-20020212](#)] belgesinin 4.4.2. bölümü DSA imzaları ve RSA imzaları için destek önerileri için destek gerektirir. Bununla birlikte, DSA'ya nazaran RSA pazarının daha büyük rağbet görmesinden dolayı, imzalı Atom Belgelerini doğrulayan Atom İşlemcilerinin RSA imzalarını doğrulayabilmesi gerekir *ZORUNLU*, fakat DSA imzalarını doğrulayabilmeleri gerekli değildir. İleti kimlik kanıtlama kodu (İKKK) için anahtar malzemesinin kimlik kanıtlaması gerektiği gibi yapılamazsa, Atom Belgeleri imzalar için İKKK'ları kullanmamaları *ÖNERİ* gerektiğinden dolayı güvenlik kaygıları ortaya çıkabilir.

5.2. Şifreleme

Bir Atom Belgesinin tepe elemanı (yani, Atom Ulama Belgesi için `atom:feed`, Atom Girdi Belgesi için `atom:entry`), XML Şifreleme Sözdizimi ve İşlemleri [[W3C.REC-xmlenc-core-20021210](#)] belgesi tarafından açıklanan mekanizmalar kullanılarak şifrelenebilir *SEÇİMLİK*.

[[W3C.REC-xmlenc-core-20021210](#)] belgesinin 5.1. bölümü TripleDES, AES-128 ve AES-256 için destek gerektirir. Atom Belgelerinin şifrelerini çözen Atom İşlemciler Şifre Bloku Zincirleme kipinde (Cipher Block Chaining – CBC) AES-128 ile şifre çözebilir *ZORUNLU*.

[[W3C.REC-xmlenc-core-20021210](#)] belgesine dayalı şifreleme özgün belgenin bütünlüğünden emin olmak için yeterli değildir. Bir iletinin şifresinin çözümlenememesine veya çözümlense bile içeriğin anlam değiştirmesine yol açtığı bilinen şifreleme saldırıları vardır. Bu nedenle, Atom belgelerinin şifrelerini çözümlen Atom İşlemcileri belgede (eğer varsa) belge bitözetini ya da imza bitözetini doğrulayarak şifreli belgenin bütünlüğünü sınamalıdır *ÖNERİ*.

5.3. İmzaladıktan sonra Şifreleme

Bir Atom Belgesi hem imzalı hem de şifreli olması gerektiğinde, genellikle en iyisi belgeyi önce imzalamak sonra da imzalı belgeyi şifrelemektir. Bu, imzalı ögenin kimliği de dahil olmak üzere tüm bilgiyi şifreleyerek asıl belgenin bütünlüğünü sağlar. Kimlik kanıtlama için İKKK kullanılmışsa, belgede izlenecek sıra imzaladıktan sonra şifrelemek olmalı *ZORUNLU* , başka bir yol kullanılmamalıdır.

6. Atom Eklentileri

6.1. Atomdışı Dillerden Eklentiler

Bu belirtim Atomun XML sözcük dağarcığını açıklar. Diğer imleme dillerinden sözcükler ("yabancı imleme") Atom Belgelerinde kullanılabilir. `atom:content` elemanının böyle bir içeriği içererek desteklemek için tasarlandığına dikkat ediniz.

6.2. Atom Söz Dağarcığına Eklèmeler

Atom isim alanı Atomun gelecekteki ileriye dönük uyumlu değişikliklerine ayrılmıştır. Bu belirtimin gelecekteki sürümleri Atom sözcük dağarcığına yeni elemanlar ve öznitelikler ekleyebilir. Belirtimin bu sürümüyle uyumlu olmak üzere geliştirilmiş yazılımlar böyle bir belgeyi doğru işleyemeyecek ve aslında onu dil hatalarından ayıramayacaktır. Bu değerlendirmenin amaçları doğrultusunda Atom sözcük dağarcığında olmayan dil belirtilerinden "yabancı imleme" olarak bahsedilecektir.

6.3. Yabancı İmlemenin İşlenmesi

Atom İşlemcileri, bu belirtim göre meşru olan bir yerde yabancı imleme saptadığında işlemi durdurmamalı veya bir hatayı sinyallememelidirler *ZORUNLU*. Atom İşlemcilerinin yabancı imlemeyi doğru olarak işlemesi mümkün olabildiği bir durum olabilir ve bunu yapar. Aksi takdirde, böyle bir imleme "bilinmeyen yabancı imleme" adını alır.

Bilinmeyen yabancı imleme `atom:entry`, `atom:feed` veya bir Kişilik oluşumunun bir çocuğu olarak saptandığında, Atom İşlemcileri imlemeyi ve metinsel içeriğini görmezden gelebilir *SEÇİMLİK* ama imlemenin varlığının bir sonucu olarak onların davranışlarını değiştiremez *ZORUNLU*.

Bilinmeyen yabancı imleme `atom:content` elemanında veya bir Metin oluşumunda saptandığında, yazılım imlemeyi yoksaymalı ve yabancı elemanların metin içeriklerini onları sarmalayan bir imleme yokmuşçasına işleme sokmalıdır *ÖNERİ*.

6.4. Eklenti Elemanları

Atom, açıkça yasaklandığı yerler dışında Atom belgesinin her yerinde yabancı işlemeye izin verir. `atom:entry`, `atom:feed`, `atom:source` elemanlarının çocukları ve Kişilik oluşumları Öteveri elemanları sayılırlar ve aşağıda açıklanmışlardır. Kişilik oluşumlarının çocuk elemanları oluşuma uygulanacaklar olarak ele alınır. Bunlar dışında kalan yabancı imlemenin rolü bu belirtim tarafından tanımlanmamıştır.

6.4.1. Basit Eklenti Elemanları

Bir Basit Eklenti Elemanının öznitelikleri ve çocuk elemanları olmamalıdır ***ZORUNLU***. Eleman karakter veri içerebileceği gibi boş da olabilir ***SEÇİMLİK***. Basit Eklenti Elemanları lisana duyarlı değildir.

```
simpleExtensionElement =
  element * - atom:* {
    text
  }
```

Eleman, onu içeren ata elemanın basit bir özelliği (veya isim-değer çifti) olarak yorumlanabilir. Elemanın isim alanı URI'si ile elemanın yerel isminden oluşan çift özelliğin ismi olarak yorumlanabilir. Elemanın karakter verisinden oluşan içeriği özelliğin değeri olarak yorumlanabilir. Eğer eleman boşsa, özellik değeri bir boş dizge olarak yorumlanabilir.

6.4.2. Yapılı Eklenti Elemanları

Bir Yapılı Eklenti elemanının tepe elemanı en azından bir öznitelik veya bir çocuk eleman içermelidir ***ZORUNLU***. Öznitelikleri olabilir ***SEÇİMLİK***, iyi biçimli XML içeriği (karakter veri içererek) olabilir ***SEÇİMLİK*** veya boş olabilir ***SEÇİMLİK***. Yapılı Eklenti elemanları Lisana duyarlıdır.

```
structuredExtensionElement =
  element * - atom:* {
    (attribute * { text }+,
      (text|anyElement)*)
    | (attribute * { text }*,
      (text?, anyElement+, (text|anyElement)*))
  }
```

Bir Yapılı Eklenti elemanının yapısı onun çocuk elemanlarının sıralamasını da içererek önemli olabilir.

Bu belirtim bir Yapılı Eklenti eleman yorumlaması sağlamaz. Elemanın içerdiği XML'in sözdizimi (ve elemanın içerdiği elemanlarla olan ilgisinin nasıl yorumlanacağı) Atom eklentisinin belirtimi tarafından tanımlanır.

7. IANA Değerlendirmeleri

Bir Atom Belgesi, XML 1.0 olarak sıradanlaştırıldığında, aşağıdaki ortam türü ile betimlenebilir:

```
MIME ortam türü ismi: application
MIME alt tür ismi: atom+xml
Zorunlu parametreler: Yok.
Seçimlik parametreler:
  "karakter kümesi": Bu parametre [RFC3023]'te belirtildiği gibi
  "application/xml" ortam türünün karakter kümesi
  parametresine eşdeğer bir anlamsallığa sahiptir.
Kodlama değerlendirmeleri: [RFC3023]'ün 3.2. bölümünde
  açıklandığı gibi bunlar "application/xml" ile aynıdır.
Güvenlik değerlendirmeleri: Bu belirtimde tanımlandığı gibidir.
Ek olarak, kendisi "+xml" uzlaşımını kullanan ortam türü olarak
[RFC3023]'ün 10. bölümünde açıklanan
```


güvenlik değerlendirmelerini paylaşır.
 Birlikte çalışabilirlik değerlendirmeleri: Bilinen bir
 birlikte çalışabilirlik uzlaşımı yoktur.
 Yayınlanmış belirtim: Bu belirtim.
 Bu ortam türünü kullanan uygulama: Bu ortam türünü kullanan
 bir uygulama henüz bilinmiyor.

Ek Bilgi:

Sihirli sayı(lar): [[RFC3023](#)]'ün 3.2. bölümünde
 "application/xml" için belirtildiği gibidir.
 Dosya uzantısı: .atom
 Veridilimi betimleyiciler: [[RFC3023](#)]'ün 5. bölümünde
 "application/xml" için belirtildiği gibidir.
 Temel URI: [[RFC3023](#)]'ün 6. bölümünde
 belirtildiği gibidir.
 Macintosh Dosya Türü kodu: TEXT
 Bilgi almak için aranacak kişi ve eposta adresi:
 Mark Nottingham <email>mnot@pobox.com </email>
 Düşünülen kullanım: COMMON
 Yazar/Değişiklik denetleyici: IESG

7.1. Bağlantı İlişkilerinin Kaydı

Bu sicil kaydı IANA tarafından sağlanmıştır ve başlangıç olarak beş değer içerir: "alternate", "related", "self", "enclosure" ve "via". Yeni atamalar [[RFC2434](#)]'te açıklandığı gibi IESG Onayına konu olurlar. IESG'ye onay istemek için sevkedilmek üzere, istekler eposta ile IANA'ya yapılmalıdır. İstek için şu şablon kullanılmalıdır:

- Öznitelik Değeri: ("[rel](#)" [Özniteliği](#) (sayfa: 18) bölümünde verilen sözdizimi kuralına uygun olarak "[rel](#)" özniteliği için bir değer)
- Açıklama:
- Umulan gösterim karakteristikleri:
- Güvenlik değerlendirmeleri:

8. Güvenlik Değerlendirmeleri

8.1. HTML ve XHTML İçerik

`atom:content` ve Metin oluşumları HTML ve XHTML teslimatını mümkün kılar. Bu dillerdeki çoğu eleman istemcileri bir veya daha fazla saldırı türüne açmalarından dolayı 'güvensiz' sayılır. Atom işleyen yazılım gerçek-
 lenimcileri Atom belgelerinde gelen (X)HTML'yi işlerken her eleman türünün elde edilmesini çok dikkatle ele
 almalıdırlar. Size rehber olması için [[RFC2854](#)] ve [[HTML](#)] belgelerinin güvenlik bölümlerine bakınız.

Atom İşlemcileri IMG, SCRIPT, EMBED, OBJECT, FRAME, FRAMESET, IFRAME, META ve LINK elemanlarının
 güvenliğine gereken dikkati ayrı ayrı göstermelidir ama diğer elemanların da ayrıca olumsuz güvenlik özellikleri
 olabilir.

(X)HTML doğrudan ya da dolaylı olarak çalıştırılabilir içeriğe başvurabilir.

8.2. URI'ler

URI'leri Atom İşlemciler elde eder. [\[RFC3986\]](#)'nın 7. bölümüne bakınız.

8.3. IRI'ler

IRI'leri Atom İşlemciler elde eder. [\[RFC3987\]](#)'nin 8. bölümüne bakınız.

8.4. Aldatma

Atom İşlemcileri, saldırganın başka bir ulamın `atom:id`'sini yineleyen tahrif edilmiş bir `atom:source` elemanını kullanarak başka bir ulamdaki bir girdinin `atom:id` değeriyle bir `atom:entry` yayınlamak şeklindeki aldatma saldırıları olasılığının farkında olmalıdır. Örneğin, Atom İşlemci, eşdeğer `atom:id` değerli bir girdi kümesinden sadece bir girdi göstererek yinelenmiş girdi gösterimini bastırabilir. Bu durumda, Atom İşlemci onları yinelenmiş varsaymadan önce aynı yayınlayıcıdan kaynaklanan girdiler olup olmadıklarını saptayacak adımları ayrıca atabilir.

8.5. Şifreleme ve İmzalama

Atom Belgeleri [\[W3C.REC-xmlenc-core-20021210\]](#) ve [\[W3C.REC-xmlsig-core-20020212\]](#) kullanılarak şifrelenebilir, imzalanabilir ve kullanımlarına göre güvenlik değerlendirmelerine örtük konu olurlar.

Sayısal imzalar kimlik kanıtlama, ileti bütünlüğü ve inkarsız kaynak kanıtı sağlar. Şifreleme ise veri mahremiyeti sağlar.

9. Kaynakça

9.1. Uyulması Zorunlu Olanlar

1. [HTML]
HTML 4.01 Belirtimi -- **HTML 4.01 Specification** -- Raggett, D., Hors, A. ve I. Jacobs -- W3C REC [REC-html401-19991224^{\(B69\)}](#) -- Aralık 1999
2. [MIMereg]
Ortam Türü Belirtilimleri ve Kayda Alma Yordamları -- **Media Type Specifications and Registration Procedures** -- Freed, N. ve J. Klensin -- BCP 13, RFC 4288 -- Aralık 2005
3. [RFC2119]
RFC'lerde Gereksinim Seviyelerini Belirtmek için Kullanılan Anahtar Sözcükler -- **Key words for use in RFCs to Indicate Requirement Levels** -- Bradner, S. -- BCP 14, [RFC 2119^{\(B70\)}](#) -- Mart 1997
4. [RFC2822]
Genel Ağ İleti Biçimi -- **Internet Message Format** -- Resnick, P. -- [RFC 2822^{\(B71\)}](#) -- Nisan 2001
5. [RFC2854]
'text/html' Ortam Türü -- **The 'text/html' Media Type** -- Connolly, D. ve L. Masinter -- RFC 2854 -- Haziran 2000
6. [RFC3023]
XML Ortam Türleri -- **XML Media Types** -- Murata, M., St. Laurent, S. ve D. Kohn -- RFC 3023 -- Ocak 2001

7. [RFC3066]
Dillerin Betimlenmesi için Yaftalar -- **Tags for the Identification of Languages** -- Alvestrand, H. -- BCP 47, RFC 3066 -- Ocak 2001
8. [RFC3339]
Genel Ağ'da Tarih ve Saat: Zaman Damgaları -- **Date and Time on the Internet: Timestamps** -- Klyne, G. ve C. Newman -- RFC 3339 -- Temmuz 2002
9. [RFC3548]
Base16, Base32 ve Base64 Veri Kodlamaları -- **The Base16, Base32, and Base64 Data Encodings** -- Josefsson, S. -- RFC 3548 -- Temmuz 2003
10. [RFC3986]
Tektip Özkaynak Betimleyici (URI): Genel Sözdizimi -- **Uniform Resource Identifier (URI): Generic Syntax** -- Berners-Lee, T., Fielding, R. ve L. Masinter -- STD 66, RFC 3986 -- Ocak 2005
11. [RFC3987]
Uluslararasılaştırılmış Özkaynak Betimleyiciler (IRI'ler) -- **Internationalized Resource Identifiers (IRIs)** -- Duerst, M. and M. Suignard -- RFC 3987 -- Ocak 2005
12. [W3C.REC-xml-20040204]
Genişleyebilir İmlleme Dili (XML) 1.0 (Üçüncü Baskı) -- **Extensible Markup Language (XML) 1.0 (Third Edition)** -- Yergeau, F., Paoli, J., Sperberg-McQueen, C., Bray, T. ve E. Maler -- W3C REC [REC-xml-20040204](#) ^(B72) -- Şubat 2004
13. [W3C.REC-xml-c14n-20010315]
Meşru Olarak XML Sürüm 1.0 -- **Canonical XML Version 1.0** -- Boyer, J. -- W3C REC [REC-xml-c14n-20010315](#) ^(B73) -- Mart 2001
14. [W3C.REC-xml-exc-c14n-20020718]
Ayrıcalıklı XML Meşrulaştırma Sürüm 1.0 -- **Exclusive XML Canonicalization Version 1.0** -- Eastlake, D., Boyer, J. ve J. Reagle -- W3C REC [REC-xml-exc-c14n-20020718](#) ^(B74) -- Temmuz 2002
15. [W3C.REC-xml-infoset-20040204]
XML Bilgi Kümesi (İkinci Baskı) -- **XML Information Set (Second Edition)** -- Cowan, J. ve R. Tobin -- W3C REC [REC-xml-infoset-20040204](#) ^(B75) -- Şubat 2004
16. [W3C.REC-xml-names-19990114]
XML'de İsim Alanları -- **Namespaces in XML** -- Hollander, D., Bray, T. ve A. Layman -- W3C REC [REC-xml-names-19990114](#) ^(B76) -- Ocak 1999
17. [W3C.REC-xmlbase-20010627]
XML Üssü : -) -- **XML Base** -- Hollander, D., Bray, T. ve A. Layman -- W3C REC [REC-xmlbase-20010627](#) ^(B77) -- Temmuz 2001
18. [W3C.REC-xmlsig-core-20020212]
XML-İmza Sözdizimi ve İşlemleri -- **XML-Signature Syntax and Processing** -- Solo, D., Reagle, J. ve D. Eastlake -- W3C REC [REC-xmlsig-core-20020212](#) ^(B78) -- February 2002
19. [W3C.REC-xmlenc-core-20021210]
XML Şifreleme Sözdizimi ve İşlemleri -- **XML Encryption Syntax and Processing** -- Reagle, J. ve D. Eastlake -- W3C REC [REC-xmlenc-core-20021210](#) ^(B79) -- Aralık 2002

20. [XHTML]
XHTML[TM]'nin Modülerleştirilmesi -- **Modularization of XHTML[TM]** -- Althaim, M., Boumphrey, F., McCarron, S., Dooley, S., Schnitzenbaumer, S. ve T. Wugofski -- W3C REC [REC-xhtml-modularization-20010410](#)^(B80) -- Nisan 2001

9.2. Bilgi Niteliğindeki

1. [ISO.8601.1988]
ISO Standardı 8601, Tarih elemanları ve değişim biçimleri – Bilgi Değişimi – Tarihlerin ve saatlerin gösterimleri -- **ISO Standard 8601, Data elements and interchange formats – Information interchange – Representation of dates and times** -- -- International Organization for Standardization -- Haziran 1988
2. [RELAX-NG]
RELAX NG Bütünleşik Sözdizimi -- **RELAX NG Compact Syntax** -- Clark, J. -- <http://www.oasis-open.org/committees/relax-ng/compact-20021121.html> -- Aralık 2001
3. [RFC2434]
RFC'lerde IANA Değerlendirmeleri Bölümünü Yazmak için Kılavuz -- **Guidelines for Writing an IANA Considerations Section in RFCs** -- Narten, T. ve H. Alvestrand -- BCP 26, RFC 2434 -- Ekim 1998
4. [W3C.NOTE-datetime-19980827]
Tarih ve Saat Biçimleri -- **Date and Time Formats** -- Wolf, M. ve C. Wicksteed -- W3C NOTE [NOTE-datetime-19980827](#)^(B82) -- Ağustos 1998
5. [W3C.REC-xmlschema-2-20041028]
XML Şema 2. Kısım: Veritürleri — İkinci Baskı -- **XML Schema Part 2: Datatypes Second Edition** -- Malhotra, A. ve P. Biron -- W3C REC [REC-xmlschema-2-20041028](#)^(B83) -- Ekim 2004

A. Katkıda Bulunanlar

Bu kişiler bu belgenin başlangıç niteliğindeki sürümlerine katkıda bulundular: Tim Bray, Mark Pilgrim, and Sam Ruby. Norman Walsh Relax NG şemasını sağladı. İçerik ve kavramlar Atom topluluğunun ve Atompub Çalışma Grubunun bir ürünüdür.

Atompub Çalışma Grubu, bu belgeyi fikirleriyle kelimelendiren düzinelerce çok etkin destekçilere sahip:

Danny Ayers, James Aylett, Roger Benningfield, Arve Bersvendsen, Tim Bray, Dan Brickley, Thomas Broyer, Robin Cover, Bill de hOra, Martin Duerst, Roy Fielding, Joe Gregorio, Bjoern Hoehrmann, Paul Hoffman, Anne van Kesteren, Brett Lindsley, Dare Obasanjo, David Orchard, Aristotle Pagaltzis, John Panzer, Graham Parks, Dave Pawson, Mark Pilgrim, David Powell, Julian Reschke, Phil Ringnalda, Antone Roundy, Sam Ruby, Eric Scheid, Brent Simmons, Henri Sivonen, Ray Slakinski, James Snell, Henry Story, Asbjorn Ulsberg, Walter Underwood, Norman Walsh, Dave Winer ve Bob Wyman.

B. RELAX NG, Bütünleşik Şema

Bu ek bölüm bilgi mahiyetindedir.

Relax NG şeması, belirtimin bu düzenlemesinde tanımlanmamış olan Atom isim alanındaki elemanları açıkça dışlar. Böyle bir imlemeyi saptayan Atom İşlecileri için gereksinimlerden [Atom Söz Dağarcığına Eklemler](#) (sayfa: 22) ve [Yabancı İmlenin İşlenmesi](#) (sayfa: 22) bölümlerinde bahsedilmiştir.

```
# -*- rnc -*-
# Atom Biçimi Belirtimi Sürüm 11 için
# RELAX NG Bütünleşik Sözdizimi Bilgisi

namespace atom = "http://www.w3.org/2005/Atom"
namespace xhtml = "http://www.w3.org/1999/xhtml"
namespace s = "http://www.ascc.net/xml/schematron"
namespace local = ""

start = atomFeed | atomEntry

# Ortak öznitelikler

atomCommonAttributes =
  attribute xml:base { atomUri }?,
  attribute xml:lang { atomLanguageTag }?,
  undefinedAttribute*

# Metin oluşumları

atomPlainTextConstruct =
  atomCommonAttributes,
  attribute type { "text" | "html" }?,
  text

atomXHTMLTextConstruct =
  atomCommonAttributes,
  attribute type { "xhtml" },
  xhtmlDiv
```

```
atomTextConstruct = atomPlainTextConstruct | atomXHTMLTextConstruct

# Kişilik oluşumları

atomPersonConstruct =
    atomCommonAttributes,
    (element atom:name { text }
     & element atom:uri { atomUri }?
     & element atom:email { atomEmailAddress }?
     & extensionElement*)

# Tarih oluşumları

atomDateConstruct =
    atomCommonAttributes,
    xsd:dateTime

# atom:feed

atomFeed =
    [
        s:rule [
            context = "atom:feed"
            s:assert [
                test = "atom:author or not (atom:entry[not (atom:author)])"
                "Bir atom:feed elemanının tüm atom:entry çocukları bir "
                ~ "atom:author elemanı içermedikçe, bir atom:author "
                ~ "elemanının olması gerekir."
            ]
        ]
    ]
    element atom:feed {
        atomCommonAttributes,
        (atomAuthor*
         & atomCategory*
         & atomContributor*
         & atomGenerator?
         & atomIcon?
         & atomId
         & atomLink*
         & atomLogo?
         & atomRights?
         & atomSubtitle?
         & atomTitle
         & atomUpdated
         & extensionElement*),
        atomEntry*
    }
```

```

# atom:entry

atomEntry =
[
  s:rule [
    context = "atom:entry"
    s:assert [
      test = "atom:link[@rel='alternate'] "
      ~ "or atom:link[not(@rel)] "
      ~ "or atom:content"
      "Bir atom:entry elemanının rel özneliği 'alternate' "
      ~ "olan en azından bir atom:link elemanı veya bir "
      ~ "atom:content elemanı olması gerekir."
    ]
  ]
]
s:rule [
  context = "atom:entry"
  s:assert [
    test = "atom:author or "
    ~ "../atom:author or atom:source/atom:author"
    "Eğer ulamı içermiyorsa, bir atom:entry elemanının bir "
    ~ "atom:author çocuğu olması gerekir."
  ]
]
]
element atom:entry {
  atomCommonAttributes,
  (atomAuthor*
  & atomCategory*
  & atomContent?
  & atomContributor*
  & atomId
  & atomLink*
  & atomPublished?
  & atomRights?
  & atomSource?
  & atomSummary?
  & atomTitle
  & atomUpdated
  & extensionElement*)
}

# atom:content

atomInlineTextContent =
  element atom:content {
    atomCommonAttributes,
    attribute type { "text" | "html" }?,

```

```
        (text)*
    }

atomInlineXHTMLContent =
    element atom:content {
        atomCommonAttributes,
        attribute type { "xhtml" },
       /xhtmlDiv
    }

atomInlineOtherContent =
    element atom:content {
        atomCommonAttributes,
        attribute type { atomMediaType }?,
        (text|anyElement)*
    }

atomOutOfLineContent =
    element atom:content {
        atomCommonAttributes,
        attribute type { atomMediaType }?,
        attribute src { atomUri },
        empty
    }

atomContent = atomInlineTextContent
| atomInlineXHTMLContent
| atomInlineOtherContent
| atomOutOfLineContent

# atom:author

atomAuthor = element atom:author { atomPersonConstruct }

# atom:category

atomCategory =
    element atom:category {
        atomCommonAttributes,
        attribute term { text },
        attribute scheme { atomUri }?,
        attribute label { text }?,
        undefinedContent
    }

# atom:contributor

atomContributor = element atom:contributor { atomPersonConstruct }
```

```
# atom:generator

atomGenerator = element atom:generator {
    atomCommonAttributes,
    attribute uri { atomUri }?,
    attribute version { text }?,
    text
}

# atom:icon

atomIcon = element atom:icon {
    atomCommonAttributes,
    (atomUri)
}

# atom:id

atomId = element atom:id {
    atomCommonAttributes,
    (atomUri)
}

# atom:logo

atomLogo = element atom:logo {
    atomCommonAttributes,
    (atomUri)
}

# atom:link

atomLink =
    element atom:link {
        atomCommonAttributes,
        attribute href { atomUri },
        attribute rel { atomNCName | atomUri }?,
        attribute type { atomMediaType }?,
        attribute hreflang { atomLanguageTag }?,
        attribute title { text }?,
        attribute length { text }?,
        undefinedContent
    }

# atom:published

atomPublished = element atom:published { atomDateConstruct }

# atom:rights
```



```
atomRights = element atom:rights { atomTextConstruct }

# atom:source

atomSource =
  element atom:source {
    atomCommonAttributes,
    (atomAuthor*
      & atomCategory*
      & atomContributor*
      & atomGenerator?
      & atomIcon?
      & atomId?
      & atomLink*
      & atomLogo?
      & atomRights?
      & atomSubtitle?
      & atomTitle?
      & atomUpdated?
      & extensionElement*)
  }

# atom:subtitle

atomSubtitle = element atom:subtitle { atomTextConstruct }

# atom:summary

atomSummary = element atom:summary { atomTextConstruct }

# atom:title

atomTitle = element atom:title { atomTextConstruct }

# atom:updated

atomUpdated = element atom:updated { atomDateConstruct }

# Düşük seviyeden basit türler

atomNCName = xsd:string { minLength = "1" pattern = "[^:]*" }

# Her ortam türü en azından bir "/" içerir
atomMediaType = xsd:string { pattern = ".+/.+" }

# RFC 3066'da tanımlandığı gibi
atomLanguageTag = xsd:string {
  pattern = "[A-Za-z]{1,8}(-[A-Za-z0-9]{1,8})*"
```

```
}

# Unconstrained; it's not entirely clear how IRI fit into
# xsd:anyURI so let's not try to constrain it here
atomUri = text

# Her eposta adresi en azından bir @ içerir
atomEmailAddress = xsd:string { pattern = ".+@.+" }

# Basit Eklenti

simpleExtensionElement =
    element * - atom:* {
        text
    }

# Yapılı Eklenti

structuredExtensionElement =
    element * - atom:* {
        (attribute * { text }+,
         (text|anyElement)*)
        | (attribute * { text }*,
         (text?, anyElement+, (text|anyElement)*))
    }

# Diğer Genişletmeler

extensionElement =
    simpleExtensionElement | structuredExtensionElement

undefinedAttribute =
    attribute * - (xml:base | xml:lang | local:*) { text }

undefinedContent = (text|anyForeignElement)*

anyElement =
    element * {
        (attribute * { text }
         | text
         | anyElement)*
    }

anyForeignElement =
    element * - atom:* {
        (attribute * { text }
         | text
         | anyElement)*
    }
```

```
# XHTML

anyXHTML = element xhtml:* {
  (attribute * { text }
   | text
   | anyXHTML)*
}

xhtmlDiv = element xhtml:div {
  (attribute * { text }
   | text
   | anyXHTML)*
}

# EOF
```

C. Belge Hakkında

Yazarların Adresleri

Mark Nottingham (editör)

Eposta: <mnot (at) pobox.com>

URI: <http://www.mnot.net/>

Robert Sayre (editör)

Eposta: <rfsayre (at) boswijck.com>

URI: <http://boswijck.com>

Tam Telif Hakkı Beyanı

Copyright © The Internet Society (2005).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Fikri Mülkiyet

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the

extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Bilgi için

RFC Editor^(B88) işlevinin mali desteği şu an Genel Ağ Derneği (The Internet Society) tarafından sağlanmaktadır.

Notlar

Belge içinde dipnotlar ve dış bağlantılar varsa, bunlarla ilgili bilgiler bulundukları sayfanın sonunda dipnot olarak verilmeyip, hepsi toplu olarak burada listelenmiş olacaktır.

(B1) <ftp://ftp.rfc-editor.org/in-notes/bcp/bcp78.txt>

(B3) <http://www.ietf.org/>

(1) Ç.N. – Belgede kullanılan "syndication" sözcüğü bir yazarın bir eserini aynı anda çok sayıda yayınevine satması anlamına gelen bir teknik terimdir. Bu terimin Türkçe karşılığı olarak bu çeviride "eserlerin ödünleşimli paylaşımı" deyimini kullanılmıştır.

(B11) [../rfc/rfc2119.pdf](http://rfc/rfc2119.pdf)

(B69) <http://www.w3.org/TR/1999/REC-html401-19991224>

(B70) [../rfc/rfc2119.pdf](http://rfc/rfc2119.pdf)

(B71) [../rfc/rfc2822.pdf](http://rfc/rfc2822.pdf)

(B72) <http://www.w3.org/TR/2004/REC-xml-20040204>

(B73) <http://www.w3.org/TR/2001/REC-xml-c14n-20010315>

(B74) <http://www.w3.org/TR/2002/REC-xml-exc-c14n-20020718>

(B75) <http://www.w3.org/TR/2004/REC-xml-info-set-20040204>

(B76) <http://www.w3.org/TR/1999/REC-xml-names-19990114>

(B77) <http://www.w3.org/TR/2001/REC-xmlbase-20010627>

(B78) <http://www.w3.org/TR/2002/REC-xml-dsig-core-20020212>

(B79) <http://www.w3.org/TR/2002/REC-xml-enc-core-20021210>

(B80) <http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410>

(B82) <http://www.w3.org/TR/1998/NOTE-datettime-19980827>

(B83) <http://www.w3.org/TR/2004/REC-xmlschema-2-20041028>

(B88) <http://www.rfc-editor.org/>

Bu dosya (rfc4287.pdf), belgenin XML biçiminin T_EXLive ve belgeler-xsl paketlerindeki araçlar kullanılarak PDF biçimine dönüştürülmesiyle elde edilmiştir.

17 Ocak 2007