

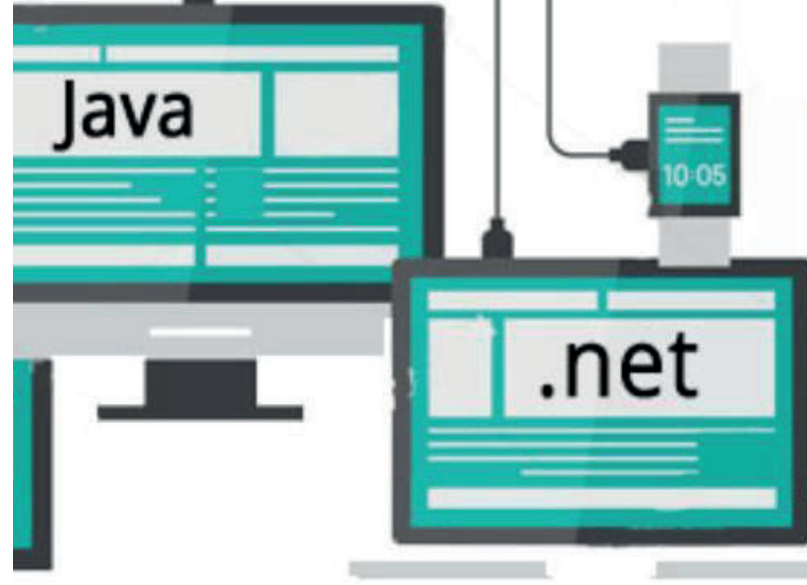
Web Service

BÖLÜM 5

WEB SERVİSLER

ANAHTAR KAVRAMLAR

XML Web Servisleri, WSDL, SOAP, XML RDF, RSS



ÖĞRENME HEDEFLERİ

- 1 XML Web Servislerinin temel kavramlarını ve çalışma prensiplerini anlamak
- 2 XML Web Servislerinin avantajlarını ve dezavantajlarını değerlendirebilmek
- 3 SOAP protokolünün temel bileşenlerini ve çalışma prensiplerini anlamak
- 4 RDF'nin temel kavramlarını ve çalışma prensiplerini anlamak
- 5 RSS'in temel kavramlarını ve çalışma prensiplerini anlamak
- 6 XML Web Servislerini RSS kullanarak yayınlatabilmek

GİRİŞ

Günümüzün dijital dünyasında, birbirleriyle iletişim kuran ve veri alışverişi yapan çok sayıda uygulama bulunmaktadır. Farklı platformlarda ve programlama dillerinde yazılmış bu uygulamaların, veri alışverişini kolaylaştırmak ve standart hale getirmek için **XML Web Servisleri** devreye girer.

XML Web Servisleri, tıpkı internette bir web sitesine erişmek gibi, farklı uygulamaların birbirleriyle iletişim kurmasına ve veri alışverişi yapmasına olanak tanıyan bir teknolojidir. Bu sayede, farklı sistemler arasında veri aktarımı manuel olarak yapılmadan otomatik hale gelir ve hatalar en aza indirilir.

Bu bölümde, XML Web Servislerinin temel kavramlarını, çalışma prensiplerini ve kullanım alanlarını adım adım inceleyeceğiz. Ayrıca, WSDL, SOAP, RDF ve RSS gibi XML Web Servisleri ile ilgili önemli teknolojileri tanımayla çalışacağız.

Hayatın içinden bir örnek vererek başlayalım: Bir kitap satıcısı olduğunuzu hayal edin: Müşterilerinizin internet üzerinden stoklarınızdaki tüm kitapları kolayca sorgulayıp, diledikleri gibi sipariş verebilmelerini ister misiniz? Üstelik bu müşteriler farklı platformlarda ve farklı dillerde yazılmış uygulamalar kullanıyor olsalar bile...

İşte XML Web Servisleri tam da bu noktada devreye giriyor! Bu güçlü teknoloji sayesinde, kitap stoklarınızı farklı uygulamalara açarak, müşterilerinizin platform ve dil sınırlamaları olmadan size ulaşmalarını sağlayabilirsiniz. Bunun için aşağıdaki yöntem izlenebilir.

1. **Kitap Listesi Alma (XML Web Servisleri):** Kitap satış sisteminizde, müşterilerin istek gönderdiği bir web servisi bulunur. Bu web servisi, müşterilerin sorması halinde mevcut kitapların bir listesini XML formatında geri döndürür. Örneğin, müşteri "Tüm kitapları listele" diye bir istek gönderdiğinde, sistem XML formatında bir cevap gönderir. Bu XML dosyasında kitapların isimleri, yazarları, fiyatları gibi bilgiler yer alır.
2. **Müşteri Sorgusu (XML):** Müşteri, kendi uygulamasında bir sorgu oluşturur. Örneğin, "Romanları listele" gibi bir sorgu olabilir. Bu sorgu, XML formatında bir istek olarak hazırlanır. Örneğin:

```
<query>
  <type>roman</type>
</query>
```

3. **Web Servisi İşlemi (XML Web Servisi):** Müşterinin oluşturduğu bu XML isteği, web servisine gönderilir. Web servisi bu isteği alır, içindeki bilgilere bakar ve uygun yanıtı hazırlar. Örneğin, bu sorgu sonucunda XML formatında bir cevap hazırlanır:

```
<books>
  <book>
    <title>Şeker Portakalı</title>
    <author>Jose Mauro de Vasconcelos</author>
    <price>25.00</price>
  </book>
  <book>
    <title>Yaban</title>
    <author>Halit Ziya Uşaklıgil</author>
    <price>20.00</price>
  </book>
  <!-- Diğer kitaplar -->
</books>
```

4. **Müşteriye Yanıt (XML):** Web servisi, müşterinin isteğine uygun olarak hazırladığı XML yanıtını müşteriye gönderir. Müşteri bu yanıtı alır, XML dosyasını işleyerek istediği bilgilere ulaşır.

Bu basit örnekte, XML Web Servisleri sayesinde farklı sistemler arasında iletişim kurulabiliyor. XML formatı, verilerin bir sistemden diğerine taşınmasını sağlayan standart bir formattır ve bu sayede farklı platformlar arasında sorunsuz bir iletişim sağlanabilir.

XML Web Servis Tanım Dili (WSDL)

XML Web Servis Tanım Dili (WSDL), XML tabanlı web servislerinin nasıl kullanılacağını tanımlayan bir standarttır. Şimdi basit bir örnek üzerinden WSDL'yi anlatalım:

Yukarıdaki örnekte verdiğimiz gibi bir kitap satıcısının müşterilerinin XML tabanlı web servisini kullanarak kitapları sorgulaması istendiğini varsayalım. WSDL, bu web servisini tanımlamak için kullanılır.

1. **Servis Tanımı (WSDL):** İlk adım olarak, web servisinin ne tür işlemleri kabul edeceğini ve nasıl kullanılacağını tanımlayan bir WSDL belgesinin oluşturulması gerekir. Bu belge, XML formatında yazılır ve web servisinin işlevselliğini ve erişim yöntemlerini açıklar. Örneğin, bir kitap listesi alma servisinin WSDL belgesi şöyle olabilir:

```
<definitions name="BookService" targetNamespace="http://example.com/books">
  <service name="BookService">
    <port name="BookPort" binding="tns:BookBinding">
      <soap:address location="http://example.com/books/service"/>
    </port>
  </service>
  <binding name="BookBinding" type="tns:BookPortType">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="getBookList">
      <soap:operation soapAction="getBookList"/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
  </binding>
  <portType name="BookPortType">
    <operation name="getBookList">
      <input message="tns:getBookList"/>
      <output message="tns:getBookListResponse"/>
    </operation>
  </portType>
  <message name="getBookList"/>
  <message name="getBookListResponse"/>
</definitions>
```

Bu WSDL belgesi, "BookService" adında bir servisi ve "getBookList" adında bir işlemi tanımlar. Ayrıca, bu servise SOAP (Simple Object Access Protocol) protokolünü kullanarak erişilebileceğini ve bu işlemin giriş ve çıkış parametrelerinin nasıl olacağını belirtir.

2. **Servis Kullanımı (XML İsteği):** Müşteriler, WSDL belgesine dayanarak web servisine erişebilirler. Örneğin, "getBookList" işlemini çağırmak için müşteri bir XML isteği oluşturur:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:web="http://example.com/books">
  <soapenv:Header/>
  <soapenv:Body>
    <web:getBookList/>
  </soapenv:Body>
</soapenv:Envelope>
```

Bu istek, web servisine gönderilir ve belirtilen işlemi (örneğin, "getBookList") çağırır.

3. **Servis Yanıtı (XML Cevabı):** Web servisi, isteğe yanıt olarak bir XML belgesi döndürür. Örneğin, "getBookList" işlemi çağrıldığında, servis mevcut kitapların bir listesini döndürebilir:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:web="http://example.com/books">
  <soapenv:Header/>
  <soapenv:Body>
    <web:getBookListResponse>
      <bookList>
        <book>
          <title>Kitap 1</title>
          <author>Yazar 1</author>
          <price>20.00</price>
        </book>
        <book>
          <title>Kitap 2</title>
          <author>Yazar 2</author>
          <price>25.00</price>
        </book>
        <!-- Diğer kitaplar -->
      </bookList>
    </web:getBookListResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Bu yanıt, istemciye geri döndürülür ve istemci tarafından işlenerek kullanıcıya gösterilir. Bu örnekler, WSDL web servislerinin nasıl kullanılacağını ve müşterilerin bu servislerden nasıl sorgu yapabileceğini anlatmaktadır.

XML Web Servisleri ve WSDL'nin kullanılmasının hem avantajları hem de dezavantajları bulunmaktadır.

Avantajlar:

1. **Platform Bağımsızlık:** XML tabanlı web servislerinin kullanılması, farklı platformlar arasında iletişim kurmayı sağlar. Bu, farklı programlama dilleri ve işletim sistemlerinin birbirleriyle entegre olmasını kolaylaştırır.
2. **İşbirliği Yeteneği:** XML Web Servisleri, farklı şirketlerin ve sistemlerin birlikte çalışmasını sağlar. Bu, işbirliği gerektiren uygulamalar ve sistemler arasında veri paylaşımını kolaylaştırır.
3. **Geniş Kullanılabilirlik:** XML ve WSDL standartları, yaygın olarak desteklenir ve geniş bir kullanılabilirliğe sahiptir. Bu da bu teknolojilerin uygulanmasını ve benimsenmesini kolaylaştırır.
4. **Esneklik:** XML tabanlı web servisleri, veri alışverişinde esneklik sağlar. Veri formatları ve iletişim protokolleri standartlaştırıldığı için, değişiklikler kolayca yapılabilir ve uyum sağlanabilir.

Dezavantajlar:

1. **Performans:** XML Web Servisleri, veri alışverişinde XML kullanması nedeniyle bazen performans sorunlarına neden olabilir. XML'in ağırlığı ve işleme maliyeti, özellikle büyük veri miktarlarıyla çalışırken performansı olumsuz etkileyebilir.
2. **Karmaşıklık:** WSDL belgeleri ve XML yapıları karmaşık olabilir. Bu da uygulamaların geliştirilmesi ve bakımının daha zor olmasına neden olabilir.
3. **Veri Boyutu:** XML tabanlı iletişim, veri boyutunu artırabilir. Bu da ağ trafiğini ve veri transfer maliyetlerini artırabilir.
4. **Alternatif Teknolojiler:** Gelişen teknoloji ile XML tabanlı web servislerine alternatif teknolojiler de ortaya çıkmıştır. Örneğin, JSON tabanlı RESTful servisler, daha hafif ve daha hızlı bir iletişim sağlar.

Bu avantajlar ve dezavantajlar, XML Web Servisleri ve WSDL'nin kullanılmasıyla ilgili karar verirken göz önünde bulundurulması gereken önemli faktörlerdir. Uygulamanın gereksinimleri ve ihtiyaçları doğrultusunda en uygun teknoloji seçilmelidir.

XML SOAP (Simple Object Access Protocol) Basit Nesne Erişim Protokolü

SOAP Nedir?

SOAP, Simple Object Access Protocol'ün (Basit Nesne Erişim Protokolü) kısaltmasıdır ve internet üzerinden uygulamalar arasında bilgi alışverişi yapmak için kullanılan bir **protokoldür**. XML tabanlı bir protokoldür, yani mesajlar XML formatında gönderilir ve alınır. SOAP, istemci-sunucu mimarisini kullanır, yani bir istemci sunucudan bilgi ister ve sunucu bu bilgiyi istemciye geri gönderir.

SOAP'ın temel amacı, farklı programlama dillerinde ve platformlarda yazılmış uygulamaların birbirleriyle sorunsuz bir şekilde iletişim kurabilmesini sağlamaktır. Bunu, mesajları standart bir formatta tanımlayarak ve bu mesajları taşımak için standart bir iletişim protokolü kullanarak yapar.

SOAP ile ne yapılabilir?

- Farklı programlama dilleri ve platformları kullanan uygulamalar arasında veri alışverişi yapılabilir.
- Web servisleri aracılığıyla uygulamalara erişilebilir.

- Uzaktan prosedür çağrılar (RPC) yapılabilir.
- Uzaktan veri erişimi sağlanabilir.

SOAP nasıl çalışır?

SOAP ile bir uygulama, bir sunucuya **SOAP mesajı** gönderir. Bu mesaj, bir **istek** (request) veya bir **yanıt** (response) olabilir. İstek mesajları, sunucudan bilgi veya işlem talep eder.

Yanıt mesajları ise sunucunun isteğe verdiği cevabı içerir.

SOAP mesajları, **zarf** (envelope) ve **gövde** (body) olmak üzere iki bölümden oluşur. Zarf, mesajın kimden kime gönderildiğini, mesajın türünü ve diğer meta verileri içerir. Gövde ise mesajın içeriğini içerir.

İpucu: SOAP mesajları, HTTP veya HTTPS gibi bir protokol aracılığıyla taşınır.

SOAP Örneği:

Kitap satan bir e-ticaret web sitesi olduğunu düşünün. Müşteri, bir ürünü sepete eklemek istediğinde, web sitesi SOAP mesajı kullanarak sunucuya bir istek gönderir. Bu mesaj, ürünün kimliğini ve miktarını içerir. Sunucu, mesajı işler ve müşterinin sepetine ürünü ekler. Ardından, müşteriye bir yanıt mesajı gönderir. Bu mesaj, sepetin içeriğini ve toplam tutarı içerir.

1. **İstek Oluşturma (XML SOAP):** İlk olarak, satın almak istediğin kitapla ilgili bir SOAP isteği oluşturmak gerekir. Bu istek, XML formatında bir SOAP mesajıdır. Örneğin:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <m:buyBook xmlns:m="https://example.com/bookstore">
      <bookID>123456</bookID>
      <quantity>1</quantity>
    </m:buyBook>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

2. Bu SOAP isteği, "buyBook" adında bir işlemi ve bu işlem için gerekli parametreleri içerir.
3. **SOAP İstek Gönderme:** Şimdi bu SOAP isteğini, kitap satıcısının web servisine göndermek gerekir. SOAP isteği, HTTP(S) protokolü üzerinden gönderilir. Web servisi, bu isteği alır ve içindeki işlemi gerçekleştirir.
4. **İşlem Gerçekleştirme:** Web servisi, aldığı SOAP isteğini işler. Örneğin, verilen kitap ID'sine göre stoklarını kontrol eder ve gerekli işlemleri gerçekleştirir (örneğin, kitabı satışa çıkarır veya stoktan düşer).
5. **Cevap Oluşturma (XML SOAP):** İşlem başarıyla gerçekleştirildikten sonra, web servisi bir SOAP yanıtı oluşturur. Bu yanıt da XML formatında bir SOAP mesajıdır. Örneğin:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <m:buyBookResponse xmlns:m="https://example.com/bookstore">
      <status>success</status>
      <message>Kitabınız başarıyla satın alındı.</message>
    </m:buyBookResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```
</SOAP-ENV:Body>  
</SOAP-ENV:Envelope>
```

6. **SOAP Cevabını Alma:** Son olarak, istek doğrultusunda oluşturulan SOAP yanıtı alınır. Bu yanıtı işleyerek, kitabın başarıyla satın alındığını veya bir hata oluştuğu öğrenilebilir.

XML SOAP, web servisleri arasında güvenli ve standart bir iletişim sağlar. İstemci ve sunucu arasındaki veri alışverişi, XML formatında yapılandırılmış SOAP mesajları aracılığıyla gerçekleşir. Bu sayede farklı platformlar arasında tutarlı bir iletişim sağlanabilir. XML SOAP'un bazı avantajları ve dezavantajları vardır:

Avantajlar:

1. **Platform Bağımsızlığı:** XML SOAP, farklı platformlar arasında iletişim kurabilmek için kullanılabilir. Bu, farklı programlama dilleri ve işletim sistemleri arasında uyumluluğu artırır.
2. **Standart Protokol:** XML SOAP, standart bir iletişim protokolüdür. Bu standartlık, farklı sistemler arasında entegrasyonun kolaylaştırılmasına ve uyumluluğun artırılmasına yardımcı olur.
3. **Güvenlik:** XML SOAP, iletişimin güvenliğini sağlamak için çeşitli güvenlik mekanizmalarını destekler. Bu, veri alışverişi sırasında güvenlik risklerini azaltır.
4. **Desteklenen Araçlar:** XML SOAP'u destekleyen birçok geliştirme aracı, kütüphane ve çerçeve bulunmaktadır. Bu araçlar, XML SOAP tabanlı uygulamaların geliştirilmesini kolaylaştırır.

Dezavantajlar:

1. **Karmaşıklık:** XML SOAP mesajları genellikle karmaşık olabilir. Bu, geliştirme sürecini ve mesajların işlenmesini karmaşık hale getirebilir.
2. **Performans:** XML SOAP mesajlarının işlenmesi ve ağ üzerinde iletilmesi bazen diğer daha hafif protokollere göre daha fazla kaynak gerektirebilir. Bu, performans açısından dezavantaj olabilir.
3. **Veri Boyutu:** XML SOAP mesajları genellikle büyük olabilir, çünkü XML formatı metin tabanlıdır ve belirli bir düzende yapılandırılmıştır. Bu, veri alışverişinin ağırlığını artırabilir ve bant genişliği tüketimini artırabilir.
4. **Alternatif Protokollerin Varlığı:** Gelişen teknoloji ile birlikte, XML SOAP'un yerini daha hafif ve daha performanslı protokoller almıştır. Bu alternatif protokoller, özellikle hafif ve dağıtık sistemler için daha uygun olabilir.

XML RDF

RDF (Resource Description Framework), "Kaynak Tanımlama Çerçevesi" anlamına gelir ve web üzerindeki kaynakları tanımlamak için kullanılan bir standarttır. Bu çerçeve, bilgisayarlar tarafından okunacak ve anlaşılacak şekilde özellikle tasarlanmıştır, yani insanlara gösterilmek üzere değildir. RDF, bilgileri XML biçiminde ifade eder ve bu nedenle web tarafından işlenebilir veri olarak kullanılır. Ayrıca, RDF, W3C'nin Anlamsal Web Etkinliğinin bir parçasıdır ve 10 Şubat 2004 tarihli bir W3C Tavsiyesi olarak kabul edilmiştir.

RDF kullanım alanlarına aşağıdaki örnekler gösterilebilir.

- Fiyat ve stok durumu gibi alışveriş öğelerinin özelliklerini açıklama
- Web etkinlikleri için zaman çizelgelerini açıklama
- Web sayfalarına ilişkin bilgilerin açıklanması (içerik, yazar, oluşturulma ve değiştirilme tarihi)
- Web resimleri için içerik ve derecelendirmeyi açıklama
- Arama motorları için içeriği tanımlama
- Elektronik kütüphanelerin tanımlanması

RDF, verilerin anlamını tanımlamak için kullanılır ve genellikle üç temel bileşeni vardır: **kaynaklar** (resources), **özellikler** (properties) ve **değerler** (values). Bu bileşenler şöyle açıklanabilir:

1. **Kaynaklar (Resources):** RDF'de, herhangi bir şeyi temsil eden kaynaklar bulunur. Örneğin, bir kitap, bir web sayfası, bir kişi veya bir etkinlik gibi herhangi bir şey bir kaynak olabilir.
2. **Özellikler (Properties):** Kaynaklar arasındaki ilişkileri tanımlayan özellikler bulunur. Örneğin, bir kitabın başlığı, yazarı veya yayınevi gibi özellikler bir kitap kaynağıyla ilişkilendirilebilir.
3. **Değerler (Values):** Özelliklere atanan değerler bulunur. Örneğin, bir kitabın başlığı "Harry Potter and the Philosopher's Stone" olabilir.

RDF Bildirimleri:

RDF, genellikle Turtle veya RDF/XML gibi formatlarda ifade edilir. Bu formatlar, kaynaklar, özellikler ve değerler arasındaki ilişkileri açıkça belirtmek için kullanılır.

RDF'nin avantajları şunlardır:

- **Esneklik:** RDF, web üzerindeki her türlü bilgiyi temsil etmek için kullanılabilir. Bu, çok çeşitli bilgi türlerini ve ilişkilerini kolayca tanımlama esnekliği sağlar.
- **Bağlantılı Veri:** RDF, veriler arasındaki anlamlı ilişkileri tanımlayarak bağlantılı veri oluşturmayı sağlar. Bu da verilerin daha anlamlı ve işlevsel hale gelmesine yardımcı olur.
- **Makine Okunabilirlik:** RDF, verilerin bilgisayarlar tarafından otomatik olarak işlenebilir ve anlamlandırılabilir olmasını sağlar. Bu, veri entegrasyonu, arama ve analiz gibi işlemleri kolaylaştırır.

Ancak RDF'nin dezavantajları da vardır:

- **Karmaşıklık:** RDF, bazen karmaşık ve anlaşılması zor olabilir, özellikle büyük veri kümeleri için.
- **İşleme Maliyeti:** RDF verilerinin işlenmesi ve depolanması bazen maliyetli olabilir, özellikle büyük ölçekli projelerde.
- **Yetersiz Kullanım:** RDF, genellikle belirli uzmanlık gerektiren alanlarda kullanılır ve genel web uygulamalarında daha az yaygındır. Bu da yaygın kullanımını kısıtlayabilir.

```
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:cd="http://www.recshop.fake/cd#">
```

```

<rdf:Description
rdf:about="http://www.recshop.fake/cd/Empire_Burlesque">
  <cd:artist>Bob Dylan</cd:artist>
  <cd:country>USA</cd:country>
  <cd:company>Columbia</cd:company>
  <cd:price>10.90</cd:price>
  <cd:year>1985</cd:year>
</rdf:Description>

<rdf:Description
rdf:about="http://www.recshop.fake/cd/Hide_your_heart">
  <cd:artist>Bonnie Tyler</cd:artist>
  <cd:country>UK</cd:country>
  <cd:company>CBS Records</cd:company>
  <cd:price>9.90</cd:price>
  <cd:year>1988</cd:year>
</rdf:Description>
</rdf:RDF>

```

Yukarıdaki örnekte: RDF belgesinin ilk satırı XML bildirimidir. XML bildiriminin ardından RDF belgelerinin kök ögesi gelir: <rdf:RDF> .

xmlns :rdf ad alanı, rdf öneğine sahip öğelerin "http://www.w3.org/1999/02/22-rdf-syntax-ns#" ad alanından olduğunu belirtir.

xmlns :cd ad alanı, cd öneğine sahip öğelerin "http://www.recshop.fake/cd#" ad alanından olduğunu belirtir.

< rdf:Description> ögesi, rdf:about özelliği tarafından tanımlanan kaynağın açıklamasını içerir .

Öğeler: <cd:artist>, <cd:country>, <cd:company> vb. kaynağın özellikleridir.

Title	Artist	Country	Company	Price	Year
Empire Burlesque	Bob Dylan	USA	Columbia	10.90	1985
Hide your heart	Bonnie Tyler	UK	CBS Records	9.90	1988

RDF Öğeleri

RDF'nin ana öğeleri, kök öge olan <RDF> ve bir kaynağı tanımlayan açıklama ögesidir.

<rdf:RDF> Ögesi

<rdf:RDF> bir RDF belgesinin kök ögesidir. XML belgesinin bir RDF belgesi olduğunu tanımlar. Ayrıca RDF ad alanına bir referans içerir:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
...Açıklama burada yer alır...
</rdf:RDF>
```

<rdf:Description> Öğesi

XML'de, RDF Description elemanı, bir bilgi kaynağını ve onunla ilgili bilgileri temsil etmek için kullanılır. Bu sayede verilerin anlamını ve bağlamını daha net bir şekilde ifade edebilir.

Neler Yapılabilir?

- **Bilgi Kaynaklarını Tanımlama:** RDF Description elemanını kullanarak bir bilgi kaynağını (örneğin, bir kitap, bir kişi veya bir web sitesi) tanımlayabilirsiniz.
- **Özellikleri Tanımlama:** RDF Description elemanının alt öğeleri aracılığıyla bilgi kaynağına ilişkin özellikleri tanımlayabilirsiniz. Örneğin, bir kitap için yazar, başlık, ISBN ve yayın tarihi gibi bilgileri tanımlayabilirsiniz.
- **İlişkileri Gösterme:** RDF Description elemanındaki özellikler aracılığıyla bilgi kaynakları arasındaki ilişkileri gösterebilirsiniz. Örneğin, bir yazarın yazdığı kitapları veya bir kitap kategorisini ilişkilendirebilirsiniz.

Nasıl Yapılır?

XML'de bir RDF Description elemanı oluşturmak için <rdf:Description> etiketini kullanmanız gerekir. Örnek olarak, bir kitap hakkında bilgi içeren bir XML dosyasında şu şekilde bir kod kullanabilirsiniz:

```
<?xml version="1.0" encoding="UTF-8"?>
<kitaplar xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  <rdf:Description rdf:about="http://www.example.com/kitaplar/1">
    <baslik>Veritabanları ve XML</baslik>
    <yazar>Roger Conklin</yazar>
    <yayinEvi>Addison-Wesley</yayinEvi>
    <yayinTarihi>2003</yayinTarihi>
    <tur>Bilgisayar Bilimi</tur>
    <fiyat>50 TL</fiyat>
    <ozet>Veritabanları ve XML hakkında bir kitap...</ozet>
  </rdf:Description>
</kitaplar>
```

Bu kodda, <rdf:Description> etiketi bir RDF Description elemanı oluşturur. rdf:about özelliği, elemanın temsil ettiği bilgi kaynağının URL'sini belirtir. Bu örnekte, bilgi kaynağı, http://www.example.com/kitaplar/1 URL'sindeki bir kitaptır.

Nitelik Olarak Özellikler

Özellik öğeleri aynı zamanda kaynaklar olarak da tanımlanabilir:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:book="http://www.library.com/book#"
  <rdf:Description rdf:about="http://www.library.com/book/12345">
```

```
<book:author
rdf:resource="http://www.library.com/author/agatha_christie"/>
<book:title>Mavi Ekspres'te Cinayet</book:title>
</rdf:Description>
</rdf:RDF>
```

Yukarıdaki örnekte yazar özelliğinin bir değeri yoktur, ancak yazar hakkında bilgi içeren bir kaynağa referansı vardır.

RDF Konteynerleri

XML'de, RDF konteynerleri, birden fazla değeri tek bir özellikte tutmak için kullanılır. Bu sayede, karmaşık ilişkileri ve verileri daha kolay bir şekilde temsil edebilirsiniz.

RDF'de, bilgi kümelerini ve sıralı veya sırasız ilişkileri temsil etmek için üç konteyner türü kullanılır:

RDF'de, bilgi nesnelerini organize etmek ve ilişkilerini göstermek için konteynerler kullanılabilir. En yaygın kullanılan üç konteyner türü şunlardır:

1. <Bag>:

- **Tanım:** Bir <Bag>, sıralı veya sırasız olabilen bir değer kümesi tanımlar.
- **Kullanım:** Bir nesnenin birden fazla değere sahip olduğu durumlarda kullanılır. Örneğin, bir kitap için birden fazla yazar veya bir film için birden fazla tür belirtilebilir.

Örnek:

```
<rdf:Description rdf:about="http://www.example.com/kitaplar/12345">
  <dc:creator rdf:resource="http://www.example.com/yazarlar/agatha_christie"/>
  <dc:creator rdf:resource="http://www.example.com/yazarlar/hercule_poirot"/>
  <dc:title>Mavi Ekspres'te Cinayet</dc:title>
</rdf:Description>
```

2. <Seq>:

- **Tanım:** Bir <Seq>, sıralı bir değer kümesi tanımlar.
- **Kullanım:** Bir nesnenin birden fazla değere sahip olduğu ve bu değerlerin sıranın önemli olduğu durumlarda kullanılır. Örneğin, bir kitabın bölümlerini veya bir filmin sahnelerini sıralı olarak listelemek için kullanılabilir.

Örnek:

```
<rdf:Description rdf:about="http://www.example.com/filmler/12345">
  <dc:subject rdf:resource="http://www.example.com/konular/gizem"/>
  <dc:subject rdf:resource="http://www.example.com/konular/macera"/>
  <dc:description>Bir grup arkadaşın ıssız bir adaya düşmesi ve hayatta kalmak için mücadele etmesi hakkında bir film...</dc:description>
</rdf:Description>
```

3. <Alt>:

- **Tanım:** Bir <Alt>, alternatif değerleri temsil eden bir değer kümesi tanımlar.
- **Kullanım:** Bir nesnenin birden fazla değere sahip olduğu ve bu değerlerin hangisinin kullanılacağını belirsiz olduğu durumlarda kullanılır. Örneğin, bir kitabın farklı baskılarının ISBN numaralarını veya bir filmin farklı dillerdeki isimlerini listelemek için kullanılabilir.

Örnek:

```
<rdf:Description rdf:about="http://www.example.com/kitaplar/12345">
  <dc:identifier rdf:resource="http://www.example.com/isbn/978-0-14-035697-5"/>
  <dc:identifier rdf:resource="http://www.example.com/isbn/978-0-14-035698-2"/>
  <dc:title>Mavi Ekspres'te Cinayet</dc:title>
</rdf:Description>
```

RDF Koleksiyonları

RDF'de, bilgi nesnelerini organize etmek ve ilişkilerini göstermek için koleksiyonlar kullanılabilir. Koleksiyonlar, birden fazla bilgi nesnesini tek bir birim halinde gruplandırmak için kullanılır. Bu sayede, bilgi nesneleri arasındaki ilişkileri daha net bir şekilde ifade etmek ve verilerin daha kolay işlenmesini sağlamak mümkün hale gelir.

Kullanım Alanları:

- **Listeler:** Bir nodanın birden fazla sıralı değere sahip olduğu durumlarda kullanılır. Örneğin, bir kitabın yazarlarını veya bir filmin türlerini listelemek için kullanılabilir.
- **Setler:** Bir nodanın birden fazla sırasız değere sahip olduğu durumlarda kullanılır. Örneğin, bir kitabın konularını veya bir filmin oyuncularını listelemek için kullanılabilir.
- **Torbalar:** Bir nodanın birden fazla sırasız ve tekrarlanabilen değere sahip olduğu durumlarda kullanılır. Örneğin, bir kitabın baskılarının ISBN numaralarını veya bir filmin farklı dillerdeki isimlerini listelemek için kullanılabilir.

rdf:parseType="Collection" Özelliği Nedir?

rdf:parseType="Collection", bir RDF noda bağlı bir XML ögesinin birden fazla RDF terimini temsil ettiğini göstermek için kullanılır. Bu, bir nodanın tek bir değer yerine bir koleksiyonla ilişkilendirilmesini sağlar.

Kullanım Alanları:

- **Listeler:** Bir nodanın birden fazla sıralı değere sahip olduğu durumlarda kullanılır. Örneğin, bir kitabın yazarlarını veya bir filmin türlerini listelemek için kullanılabilir.
- **Setler:** Bir nodanın birden fazla sırasız değere sahip olduğu durumlarda kullanılır. Örneğin, bir kitabın konularını veya bir filmin oyuncularını listelemek için kullanılabilir.
- **Torbalar:** Bir nodanın birden fazla sırasız ve tekrarlanabilen değere sahip olduğu durumlarda kullanılır. Örneğin, bir kitabın baskılarının ISBN numaralarını veya bir filmin farklı dillerdeki isimlerini listelemek için kullanılabilir.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description rdf:about="http://www.example.com/kitaplar/12345">
    <dc:creator rdf:parseType="Collection">
      <rdf:li
rdf:resource="http://www.example.com/yazarlar/agatha_christie"/>
      <rdf:li
rdf:resource="http://www.example.com/yazarlar/hercule_poirot"/>
    </dc:creator>
    <dc:title>Mavi Ekspres'te Cinayet</dc:title>
  </rdf:Description>
</rdf:RDF>
```

Bu örnekte, dc:creator özelliği, rdf:parseType="Collection" özelliğini kullanarak bir liste tanımlamak için kullanılır. Bu liste, kitabın iki yazarını içerir. rdf:li etiketleri, listenin öğelerini temsil eder.

Faydaları:

- **Veri Esnekliği:** rdf:parseType="Collection", bir nodanın birden fazla değere sahip olmasını sağlayarak veri esnekliği sağlar.
- **Veri Temsili:** rdf:parseType="Collection", bilgi nesnelerini daha net bir şekilde temsil etmenizi sağlar.
- **Veri İşleme:** rdf:parseType="Collection", verilerin daha kolay işlenmesini ve sorgulanmasını sağlar.

RDF Şeması ve Uygulama Sınıfları

RDF Şeması (RDFS):

RDF Şeması (RDFS), RDF'de bilgi nesneleri ve ilişkileri tanımlamak için kullanılan bir meta veri dilidir. RDFS, temel ontoloji kavramlarını ve ilişkilerini tanımlamak için kullanılır. Bunlar şunları içerir:

- **Sınıflar:** Gerçek dünyadaki nesneleri veya kavramları temsil eder.
- **Özellikler:** Sınıflar arasındaki ilişkileri ve sınıfların özelliklerini temsil eder.
- **Alt Sınıflar:** Bir sınıfın, daha genel bir sınıftan türediğini gösterir.
- **Özellik Kısıtlamaları:** Özelliklerin değerlerinin hangi türde olması gerektiğini belirtir.

Uygulama Sınıfları:

Uygulama sınıfları, RDFS'de tanımlanan temel kavramları kullanarak belirli bir alan veya uygulama için bilgi nesnelerini ve ilişkilerini tanımlamak için kullanılır. Uygulama sınıfları, RDFS'nin sunduğundan daha fazla ayrıntı ve özel bilgi sağlayarak daha karmaşık ve nüanslı bilgi modelleri oluşturmak için kullanılabilir.

Sınıflar, Özellikler, Alt Sınıflar ve Özellik Kısıtlamaları Tablosu

Terim	Tanım	Örnek Kullanım
-------	-------	----------------

Sınıf	Gerçek dünyadaki nesneleri veya kavramları temsil eder.	Kitap, Film, Kişi gibi
Özellik	Sınıflar arasındaki ilişkileri ve sınıfların özelliklerini temsil eder.	yazar, tür, ad, soyad gibi
Alt Sınıf	Bir sınıfın, daha genel bir sınıftan türediğini gösterir.	Roman sınıfı, Kitap sınıfının alt sınıfıdır.
Özellik Kısıtlaması	Özelliklerin değerlerinin hangi türde olması gerektiğini belirtir.	yazar özelliğinin değeri Kişi sınıfından bir nesne olmalıdır.

Ek Bilgiler:

- Bir sınıf birden fazla alt sınıfa sahip olabilir.
- Bir özellik birden fazla sınıf tarafından kullanılabilir.
- Bir özellik kısıtlaması, bir özelliğin değerinin veri türünü, aralığını veya formatını belirleyebilir.

Örnek:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xml:base="http://www.animals.fake/animals#">
  <rdf:Description rdf:ID="animal">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  </rdf:Description>
  <rdf:Description rdf:ID="horse">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="#animal"/>
  </rdf:Description>
</rdf:RDF>
```

Yukarıdaki örnekte, RDF (Resource Description Framework) ile hayvanları ve sınıflandırmalarını tanımlayan bir şema oluşturmaktadır. "Hayvan" adında bir üst sınıf ve "at" adında bir alt sınıf tanımlanır. Bu, tüm atların hayvan sınıfından türediğini ve hayvanların özelliklerini paylaştıklarını gösterir. Kod, RDF ve RDFS (RDF Şeması) için ad alanları kullanır ve göreceli URI'leri çözmek için bir taban URI belirler.

Aynı Örneğin Daha Kısa Yazılışı:

RDFS sınıfı bir RDF kaynağı olduğundan, yukarıdaki örneği rdf:Description yerine rdfs:Class kullanarak kısaltabilir ve rdf:type bilgisini bırakabiliriz:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xml:base="http://www.animals.fake/animals#">
```

```

<rdfs:Class rdf:ID="animal"/>
<rdfs:Class rdf:ID="horse">
  <rdfs:subClassOf rdf:resource="#animal"/>
</rdfs:Class>
</rdf:RDF>

```

RDFS / RDF Sınıfları

Sınıf	Tanım	Alt Sınıfı
rdfs:Class	Gerçek dünyadaki nesneleri veya kavramları temsil eder.	rdfs:Datatype (Veri Tipleri) rdfs:Resource (Tüm Kaynaklar) rdfs:Literal (Değerler - Metin ve Sayılar) rdfs:Container (Konteynerler) rdfs:List (Listeler) rdfs:Property (Özellikler) rdfs:Statement (İfadeler)
rdfs:Datatype	Veri tiplerini (örneğin, string, integer, date) temsil eder.	Yok
rdfs:Resource	Tüm RDF kaynaklarını temsil eder.	rdfs:Literal (Değerler - Metin ve Sayılar) rdfs:Container (Konteynerler)
rdfs:Literal	Metin veya sayısal değerleri temsil eder.	Yok
rdfs:Container	Diğer RDF kaynaklarını (örneğin, listeler, torbalar, diziler) gruplandırmak için kullanılır.	rdfs:List (Listeler) rdfs:Bag (Sırasız Konteynerler) rdfs:Seq (Sıralı Konteynerler)
rdfs:List	Sıralı bir kaynak koleksiyonunu temsil eder.	Yok
rdfs:Property	Sınıflar arasındaki ilişkileri veya bir sınıfın özelliklerini temsil eder.	Yok
rdfs:Statement	Bir özne, yüklem ve nesneden oluşan bir RDF ifadesini temsil eder.	Yok
rdfs:Alt	Bir kaynağın alternatif değerlerini temsil eder.	Yok

rdfs:Bag	Sırasız ve tekrarlanabilen bir kaynak koleksiyonunu temsil eder.	Yok
rdfs:Seq	Sıralı ve tekrarlanmayan bir kaynak koleksiyonunu temsil eder.	Yok
rdfs:XMLLiteral	XML biçimindeki değerleri temsil eder.	Alt sınıfı yoktur, rdfs:Literal sınıfının bir alt türü olarak düşünülebilir.

RDFS / RDF Özellikleri

Terim	Açıklaması	Örnek
rdfs:domain	Özelliğin uygulandığı kaynak sınıfı	Yazar özelliğinin domain'i Kişi sınıfı olabilir.
rdfs:range	Özelliğin değerlerinin ait olduğu sınıf	Yazar özelliğinin range'i Kişi sınıfı olabilir.
rdfs:subPropertyOf	Bir özelliğin başka bir özelliğin alt özelliği olduğunu gösterir	tür özelliği, kategorisi özelliğinin alt özelliği olabilir.
rdfs:subClassOf	Bir sınıfın başka bir sınıfın alt sınıfı olduğunu gösterir	Roman sınıfı, Kitap sınıfının alt sınıfı olabilir.
rdfs:comment	Kaynağın insan tarafından okunabilir açıklaması	Kitap sınıfının yorumu "Farklı konularda yazılmış eserler" olabilir.
rdfs:label	Kaynağın insan tarafından okunabilir etiket (adı)	Kitap sınıfının label'ı "Kitap" olabilir.
rdfs:isDefinedBy	Kaynağın tanımlandığı kaynak	Kitap sınıfının tanımı, bir ontoloji dosyasında bulunabilir.
rdfs:seeAlso	Kaynak hakkında ek bilgi	Kitap sınıfı için Yayıncı sınıfına bakın bilgisi bulunabilir.
rdfs:member	Bir koleksiyonun üyesi	Yazarlar koleksiyonunun bir üyesi Agatha Christie olabilir.
rdf:first	Listenin ilk elemanı	Yazarlar listesindeki ilk yazar Agatha Christie olabilir.

rdf:rest	Listenin geri kalanı	Yazarlar listesinde Agatha Christie'den sonraki yazarlar listenin geri kalanını oluşturur.
rdf:subject	Bir RDF ifadesindeki özne	Agatha Christie yazdı ifadesinde özne Agatha Christie olabilir.
rdf:predicate	Bir RDF ifadesindeki yüklem	Agatha Christie yazdı ifadesinde yüklem yazdı fiilidir.
rdf:object	Bir RDF ifadesindeki nesne	Agatha Christie yazdı ifadesinde nesne yazılan eser olabilir.
rdf:value	Değerler için kullanılan özellik	fiyat özelliği bir ürünün fiyatını tutar.
rdf:type	Kaynağın ait olduğu sınıf	Agatha Christie örneği Kişi sınıfına ait olabilir.

RDF Nitelikleri

Nitelik	Açıklama	Örnek
rdf:about	Tanımlanan kaynağı belirtir.	rdf:about="#kitap1" (kitap1 kimliği)
rdf:Description	Kaynak tanımı için kullanılır.	Her kaynak <rdf:Description> içinde tanımlanır.
rdf:resource	Kaynağı tanımlar.	dc:creator="yazarlar/agatha_christie"
rdf:datatype	Veri türünü tanımlar.	yas xsd:date (tarih formatı)
rdf:ID	Benzersiz kimlik tanımlar.	rdf:ID="kitapX" (kitapX kimliği)
rdf:li	Liste ögesini tanımlar.	<rdf:li>Yazar Adı</rdf:li>
rdf:nodeID	Düğüm kimliğini tanımlar.	rdf:nodeID="yazarY" (yazarY kimliği)
rdf:RDF	RDF belgesinin köküdür.	Her RDF belgesi rdf:RDF ile başlar.
xml:base	XML tabanını tanımlar.	Göreceli URI'lerin çözümü için.
xml:lang	Dil bilgisini tanımlar.	Çok dilli içeriklerde kullanılır.

XML RSS (Really Simple Syndication)

RSS Nedir?

RSS, Really Simple Syndication'ın kısaltmasıdır ve Türkçe'ye Gerçekten Basit Dağıtım olarak çevrilebilir. İnternette sık güncellenen web sitelerinin (haber siteleri, bloglar, podcastler vb.)

yeni içeriklerini takip etmenizi sağlayan bir sistemdir. Kullandığı dosya biçimleri **.rss** ve **.xml**'dir. RSS; web geliştiriciler tarafından RSS akışı oluşturacak .rss veya .xml dosyaların web sitelerinde oluşturulmasının ardından istemci tarafında yani web ziyaretçileri tarafından da RSS okuyucu yazılımlara söz konusu dosyaların bağlantısının eklenmesiyle süreç tamamlanmaktadır. RSS kaynağı sağlayan internet sitelerinde genellikle şu simgeler eklenir.



RSS kısaltmasının açılımı ve zaman içinde gelişimi şöyledir:

- Rich Site Summary (RSS 0.91) (Zengin Site Özeti)
- RDF Site Summary (RSS 0.9 and 1.0) (RDF Site Özeti)
- Really Simple Syndication (RSS 2.0.0) (Çok Basit Besleme)

RSS, XML tabanlı bir dosya formatı kullanır. Bu dosya, sitenin güncel içeriklerinin başlıklarını, özetlerini, bağlantı adreslerini ve diğer bazı bilgilerini içerir. RSS okuyucu adı verilen bir program aracılığıyla bu dosyayı takip ederek, web sitesini ziyaret etmeden yeni içeriklerden haberdar olabilirsiniz.

RSS Nasıl Çalışır?

1. Abone Olma: İlgi duyduğunuz web sitenin RSS beslemesine abone olursunuz. Bunu, sitenin RSS bağlantısını kopyalayıp RSS okuyucunuza yapıştırarak veya sitenin sunduğu abonelik seçeneklerini kullanarak yapabilirsiniz.
2. RSS Beslemesini Oku: RSS okuyucu, abone olduğunuz sitelerin RSS beslemelerini otomatik olarak kontrol eder. Yeni içerikler eklendiğinde, RSS okuyucuda bir bildirim alırsınız.
3. İçeriği Görüntüleme: Bildirime tıkladığınızda, RSS okuyucuda içeriğin başlığını, özetini ve bağlantısını görebilirsiniz. İstediğiniz içeriğin bağlantısına tıklayarak web sitesine gidip içeriği tam olarak okuyabilirsiniz.

RSS Örneği

Aşağıda yer alan örnek RSS Belgesindeki ilk satır - XML bildirimi - belgede kullanılan XML sürümünü ve karakter kodlamasını tanımlar. Örneğimizde verilen belge XML'in 1.0 versiyonuna uygundur ve UTF-8 karakter setini kullanır.

Sonraki satır, bunun bir RSS belgesi olduğunu tanımlayan RSS beyanıdır (Örneğimizde RSS sürüm 2.0).

Sonraki satır <channel> ögesini içerir. Bu öge RSS beslemesini tanımlamak için kullanılır.

<channel> ögesinin gerekli üç alt ögesi vardır:

<title> - Kanalin başlığını tanımlar (örn. **Örnek Blog**)

<link> - Kanala olan köprüyü tanımlar (örn. **<https://www.ornekblog.com>**)

<açıklama> - Kanalı açıklar (örn. **Teknoloji ve Bilim Hakkında Bilgiler**)

Her <channel> elemanı bir veya daha fazla <item> elemanına sahip olabilir.

Her <item> ögesi, RSS beslemesindeki bir makaleyi veya "hikayeyi" tanımlar.

<item> ögesinin gerekli alt ögeleri vardır:

<title> - Öğenin başlığını tanımlar (örn. **Yapay Zeka Gelecekte Nasıl Bir Rol Oynayacak?**)

<link> - Öğeye olan köprüyü tanımlar (örn. **<https://www.ornekblog.com/yapay-zeka-gelecekte-nasil-bir-rol-oyunayacak>**)

<açıklama> - Öğeyi açıklar (örn. **Yapay zekanın gelişimi ve gelecekte insanlık üzerinde yaratacağı etkiler hakkında bir makale.**)

<pubDate> Öğenin yayınlanma zamanını tanımlar. (örn. **Mon, 04 May 2024 15:30:00 +0300**)

Son olarak son iki satır <channel> ve <rss> ögelerini kapatır.

```
<?xml version="1.0" encoding="UTF-8"?>
<rss version="2.0">
<channel>
  <title>Örnek Blog</title>
  <link>https://www.ornekblog.com</link>
  <description>Teknoloji ve Bilim Hakkında Bilgiler</description>
<item>
  <title>Yapay Zeka Gelecekte Nasıl Bir Rol Oynayacak?</title>
  <link>https://www.ornekblog.com/yapay-zeka-gelecekte-nasil-bir-rol-oyunayacak</link>
  <description>Yapay zekanın gelişimi ve gelecekte insanlık üzerinde yaratacağı etkiler hakkında bir makale.</description>
  <pubDate>Mon, 04 May 2024 15:30:00 +0300</pubDate>
</item>
<item>
  <title>Yeni Uzay Teleskobu Başarıyla Fırlatıldı</title>
  <link>https://www.ornekblog.com/yeni-uzay-teleskobu-basariyla-firlatildi</link>
  <description>Yeni uzay teleskobunun özellikleri ve evreni keşfetmede nasıl yardımcı olacağı hakkında bir makale.</description>
  <pubDate>Sun, 29 Apr 2024 12:00:00 +0300</pubDate>
</item>
</channel>
</rss>
```

Bu kodda, **channel** ögesi blogun adını, bağlantısını ve açıklamasını içerir. **item** ögeleri ise her bir yeni makaleyi temsil eder ve makalenin başlığını, bağlantısını, özetini ve yayınlanma tarihini içerir.

RSS'in Faydaları

- **Zaman Tasarrufu:** RSS sayesinde, takip ettiğiniz tüm sitelerin yeni içeriklerini tek bir yerden takip edebilir ve zamandan tasarruf edebilirsiniz.

- **Güncel Kalma:** RSS, yeni içeriklerden anında haberdar olmanızı sağlayarak güncel kalmanıza yardımcı olur.
- **Bilgi Düzenleme:** RSS, ilgi alanlarınıza göre bilgi almanızı ve gereksiz içeriklerden uzak durmanızı sağlar.

RSS Kullanmanın Dezavantajları

- **Teknik Gereklilik:** RSS kullanmak için bir RSS okuyucu programı kurmanız gerekir.
- **Bakım Zorluğu:** Bazı siteler RSS beslemelerini güncellemeyi bırakabilir veya formatlarını değiştirebilir.
- **Bilgi Kirliliği:** Çok fazla RSS beslemesine abone olmak, bilgi kirliliğine yol açabilir.

RSS Kimler İçin Faydalıdır?

RSS, web sitelerini **sık sık güncelleyen** içerik üreticileri için oldukça faydalı bir araçtır.

- **Haber siteleri:** RSS, haber sitelerinin güncel haberlerini, başlıkları, tarihlerini ve özetlerini takip etmenizi sağlar. Bu sayede, web sitesini ziyaret etmeden bile en son gelişmelerden haberdar olabilirsiniz.
- **Şirketler:** Şirketler, RSS kanalları aracılığıyla yeni ürünlerini, haberlerini ve güncellemelerini duyurabilirler. Bu sayede müşterileri ve iş ortakları ile daha kolay iletişim kurabilirler.
- **Takvimler:** RSS, yaklaşan etkinlikleri, önemli günleri ve randevuları takip etmenizi sağlar. Bu sayede zamanınızı daha iyi planlayabilir ve önemli bir şeyi kaçırma riskini en aza indirebilirsiniz.
- **Bloglar ve forumlar:** Bloglar ve forumlar da RSS kanalları aracılığıyla yeni içeriklerini takipçileriyle paylaşabilirler. Bu sayede içerik üreticileri, takipçileriyle daha fazla etkileşime girebilir ve onları sitenin güncel gelişmelerinden haberdar edebilirler.

Önemli Hatırlatma: RSS kullanmak için bir RSS okuyucu programı kurmanız gerekir. Feedly, Inoreader gibi birçok farklı RSS okuyucu programı mevcuttur.

Hangi RSS Versiyonu Kullanılmalı?

RSS 0.91 ve RSS 2.0, RSS 1.0'a göre daha kolay anlaşılır ve kullanılır. Bu bölümde verilen bilgiler de **RSS 2.0** versiyonuna dayanmaktadır.

RSS 2.0, basit ve katı sözdizimi kurallarına sahiptir. Bu nedenle, yeni başlayanlar için daha uygundur.

RSS, resmi bir web standardı değildir. Fakat, birçok web sitesi ve uygulama tarafından kullanılmaktadır.

Günümüzde kullanılan RSS versiyonları:

- **%50:** RSS 0.91
- **%25:** RSS 1.0
- **%25:** RSS 0.9x ve RSS 2.0 (diğer versiyonlar)

RSS Nasıl Çalışır?

- **RSS kullanımı için:**

1. Bir RSS belgesi oluşturulur ve .xml uzantısıyla kaydedilir.
2. Dosya web sitesine yüklenir.
3. Bir RSS okuyucuyu yüklenir veya çevrimiçi yazılımlara kaydolunur.
4. Okuyucu, web sitenizi her gün tarar ve yeni RSS belgelerini otomatik olarak bulur.
5. Takipçiler, RSS toplayıcı aracılığıyla yeni içeriklere bağlantıyı bulabilirler.

RSS XML olduğundan şu hususlar unutulmamalıdır:

- Tüm öğelerin bir kapanış etiketi olmalıdır.
- Öğeler büyük/küçük harfe duyarlıdır.
- Öğeler düzgün bir şekilde iç içe yerleştirilmelidir.
- Özellik değerleri her zaman alıntılanmalıdır.

Yorum satırları

RSS'de yorum yazmanın sözdizimi HTML'deki gibidir.

```
<!-- Bu bir RSS yorum satırıdır -->
```

RSS Öğeleri

<channel> Öğesi:

RSS beslemesinin temelini oluşturan <channel> öğesi, beslemenin başlığını, bağlantısını ve açıklamasını içerir. Bu öğe, beslemeyi tanımlayan ve diğer öğelerin ana çerçevesini oluşturan önemli bir öğedir.

<channel> Öğesinin Gerekli Alt Öğeleri:

1. **<title>:** Bu alt öğe, RSS beslemesinin başlığını tanımlar ve beslemenin içeriğini özetleyen kısa bir metin içerir. Başlık, beslemenin tarayıcıda veya RSS okuyucu programında nasıl gösterileceğini belirler.
2. **<link>:** Bu alt öğe, RSS beslemesinin URL adresini (web sitesi bağlantısını) tanımlar. Kullanıcılar, bu bağlantıyı tıklayarak beslemenin kaynağına, yani web sitesine gidebilirler.
3. **<description>:** Bu alt öğe, RSS beslemesini kısaca açıklar. Açıklama metni, beslemenin içeriğini ve sunduğu konuları hakkında genel bir bilgi verir.

<channel> Öğesinin İsteğe Bağlı Alt Öğeleri:

<channel> öğesi, gerekli alt öğelere ek olarak, besleme hakkında daha fazla bilgi sağlayan çeşitli isteğe bağlı alt öğelere de sahip olabilir. Bu alt öğelerden bazı önemlileri şunlardır:

<category> Öğesi

<category> alt öğesi feed'iniz için bir kategori belirtmek amacıyla kullanılır. Bu öğe ile RSS toplayıcılarının siteleri kategoriye göre gruplandırmasına olanak tanır.

Yukarıdaki RSS belgesinin kategorisi şöyle olabilir:

`<category>Teknoloji ve Bilim</category>`

<telif hakkı> Ögesi

`<copyright>` alt ögesi, telif hakkıyla korunan materyal hakkında bilgi verir.

Yukarıdaki RSS belgesinin telif hakkı şöyle olabilir:

`<copyright>Tüm hakları Örnek Blog'a aittir. ©2024 </copyright>`

<image> Ögesi

`<image>` alt ögesi, toplayıcılar bir yayın sunduğunda bir görselin görüntülenmesine olanak tanır.

`<image>` ögesinin gerekli üç alt ögesi vardır:

- `<url>` - Resmin URL'sini tanımlar
- `<title>` - Görüntünün gösterilememesi durumunda görüntülenecek metni tanımlar
- `<link>` - Kanalı sunan web sitesine giden köprüyü tanımlar

Yukarıdaki RSS belgesinin resmi şöyle olabilir:

```
<image>
  <url>https://www.ornekblog.com/logo.gif</url>
  <title>Örnek Blog</title>
  <link>https://www.ornekblog.com</link>
</image>
```

<dil> Ögesi

`<language>` alt ögesi, belgenizi yazmak için kullanılan dili belirtmek için kullanılır.

`<language>` ögesi, RSS toplayıcılarının siteleri dile göre gruplandırmasına olanak tanır.

Yukarıdaki RSS belgesinin dili şöyle olabilir:

`<language>tr-TR</language>`

<item> Ögesi

RSS beslemeleri, `<channel>` ögesi içinde bir veya daha fazla `<item>` ögesi içerir. Her `<item>` ögesi, beslemedeki bir makaleyi veya "hikayeyi" tanımlar ve bu makaleyle ilgili bilgileri içerir.

```
<?xml version="1.0" encoding="UTF-8"?>
<rss version="2.0">
  <channel>
    <title>Örnek Blog</title>
    <link>https://www.ornekblog.com</link>
    <description>Teknoloji ve Bilim Hakkında Bilgiler</description>
    <item>
      <title>Yeni Uzay Teleskobu Başarıyla Fırlatıldı</title>
```

```
<link>https://www.ornekblog.com/yeni-uzay-teleskobu-basariyla-  
firlatildi</link>  
<description>Yeni uzay teleskobunun özellikleri ve evreni keşfetmede nasıl  
yardımcı olacağı hakkında bir makale.</description>  
<pubDate>Sun, 29 Apr 2024 12:00:00 +0300</pubDate>  
</item>  
</channel>  
</rss>
```

<item> öğesinin gerekli üç alt öğesi vardır:

- <title> - Öğenin başlığını tanımlar
- <link> - Öğeye olan köprüyü tanımlar
- <açıklama> - Öğeyi açıklar

Ayrıca <item> öğesinin çeşitli isteğe bağlı alt öğeleri vardır. Aşağıda en önemlilerini açıklayacağız.

<yazar> Öğesi

<author> alt öğesi, bir öğenin yazarının e-posta adresini belirtmek için kullanılır.

Not: Spam e-postaları önlemek için bazı geliştiriciler <author> öğesini eklemeyi tercih etmemektedir.

<enclosure> Öğesi

<enclosure> alt öğesi, bir medya dosyasının bir öğeye dahil edilmesini sağlar.

<enclosure> öğesinin üç gerekli özelliği vardır:

- url - Medya dosyasının URL'sini tanımlar
- uzunluk - Medya dosyasının uzunluğunu (bayt cinsinden) tanımlar
- type - Medya dosyasının türünü tanımlar

Yukarıdaki RSS belgesindeki öğenin içerdiği bir medya dosyası şunlar olabilir:

```
<enclosure url="https://www.ornekblog.com/rss.mp3" length="5000" type="audio/mpeg"  
>
```

RSS Okuyucular

RSS Akışlarını okumak için bir RSS Okuyucu kullanmanız gerekir. RSS okuyucular birçok farklı cihaz ve işletim sistemi için mevcuttur.

Farklı RSS Okuyucu Türleri:

- **Web Hizmetleri Olarak Çalışanlar:** Bu okuyuculara internet tarayıcınız üzerinden erişebilirsiniz.
- **Pencerelerle Sınırlı Olanlar:** Bu okuyucular Windows işletim sistemine özeldir.
- **Mac, PDA veya UNIX ile Sınırlı Olanlar:** Bu okuyucular Mac, PDA veya UNIX işletim sistemlerine özeldir.

Bazı Popüler RSS Okuyucular:

- **QuietRSS:** Açık kaynaklı, platformlar arası bir RSS/Atom haber akışı okuyucusu
- **FeedReader:** Çok sayıda feed'i kolayca işleyen basit, anlaşılır bir feed okuyucu

İpucu: Çoğu tarayıcıda yerleşik bir RSS Okuyucu bulunur. RSS beslemeleri sunan bir web sitesine giderseniz, adres çubuğunda veya araç çubuğunda bir RSS simgesi göreceksiniz. Farklı yayınların listesini görüntülemek için simgeye tıklayın. Okumak istediğiniz feed'i seçin.

<channel> Öğesinin alt öğeleri

Öge	Açıklama	Zorunlu mu?
<category>	RSS beslemesindeki içeriklerin kategorilerini (konularını) belirtir. Örneğin; haber, spor, teknoloji gibi	Hayır
<cloud>	RSS beslemesinde güncelleme olduğunda haberdar edilmek için kullanılabilecek işlemleri tanımlar	Hayır
<copyright>	Telif hakkı alınmış materyaller hakkında bilgi verir	Hayır
<description>	RSS beslemesini kısaca açıklar	Evet
<docs>	RSS beslemesinde kullanılan formatın dokümantasyonuna giden bir bağlantı URL'si belirtir	Hayır
<generator>	RSS beslemesini oluşturmak için kullanılan programı belirtir	Hayır
<image>	Toplayıcılar bir RSS beslemesi gösterirken görüntülenebilecek bir resim sağlar.	Hayır
<language>	RSS beslemesinin yazıldığı dili belirtir	Hayır
<lastBuildDate>	RSS beslemesindeki içeriğin en son değiştirilme tarihini ve saatini tanımlar	Hayır
<link>	RSS beslemesine bağlantı adresini (URL) tanımlar	Evet
<managingEditor>	RSS beslemesindeki içeriklerin editörü ile iletişime geçmek için e-posta adresini belirtir	Hayır
<pubDate>	RSS beslemesindeki içeriklerin en son yayınlanma tarihini ve saatini tanımlar	Hayır
<rating>	RSS beslemesinin PICS (Platform for Internet Content Selection) derecelendirmesini belirtir	Hayır
<skipDays>	Toplayıcıların RSS beslemesini hangi günlerde güncellemeyi atlamasını istediğinizi belirtir	Hayır
<skipHours>	Toplayıcıların RSS beslemesini hangi saatlerde güncellemeyi atlamasını istediğinizi belirtir	Hayır
<textInput>	RSS beslemesi ile birlikte görüntülenmesi gereken bir metin giriş alanı belirtir	Hayır

<title>	RSS beslemesinin başlığını tanımlar	Evet
<ttl>	RSS beslemesinin kaynak yenilenmeden önce önbellekte tutulabileceği dakika sayısını belirtir	Hayır
<webMaster>	RSS beslemesinin web yöneticisi ile iletişime geçmek için e-posta adresini belirtir	Hayır

<item> Öğesinin alt öğeleri

Element	Açıklama
<author>	İsteğe bağlıdır. Öğenin yazarının e-posta adresini belirtir.
<category>	İsteğe bağlıdır. Öğenin ait olduğu bir veya daha fazla kategoriye tanımlar.
<comments>	İsteğe bağlıdır. Öğenin, o öğe hakkındaki yorumlara bağlantı kurmasını sağlar.
<description>	Zorunludur. Öğeyi açıklar.
<enclosure>	İsteğe bağlıdır. Öğeye bir medya dosyası eklenmesine izin verir.
<guid>	İsteğe bağlıdır. Öğe için benzersiz bir tanımlayıcı tanımlar.
<link>	Zorunludur. Öğeye giden hiperlinki tanımlar.
<pubDate>	İsteğe bağlıdır. Öğe için son yayınlanma tarihini tanımlar.
<source>	İsteğe bağlıdır. Öğe için üçüncü taraf bir kaynak belirtir.
<title>	Zorunludur. Öğenin başlığını tanımlar.

BÖLÜM ÖZETİ

Web servisleri, farklı platformlarda ve programlama dillerinde yazılmış uygulamalar arasında veri alışverişini kolaylaştırmak için kullanılan X emel tabanlı bir teknolojidir. Bu servisler, farklı sistemler arasında iletişim kurmayı sağlar ve veri aktarımını otomatikleştirir, hataları en aza indirir. Temel olarak, X emel Web Servisleri, müşterilerin isteklerini işleyen ve uygun yanıtları döndüren bir yapıya sahiptir. X emel Web Servisleri'nin kullanımıyla, örneğin bir kitap satıcısı olarak, müşterilerinizin stoklarınızdaki kitapları sorgulaması ve sipariş vermesi mümkün hâle gelir. Bu sayede, farklı platformlarda ve dillerde yazılmış uygulamaları kullanan müşterilere kolayca hizmet verebilirsiniz. Örneğin, müşterilerin bir istekte bulunması durumunda, sistem X emel formatında bir cevap gönderir ve bu cevapta kitapların isimleri, yazarları ve fiyatları gibi bilgiler yer alır. X emel Web Servislerinin kullanımı için X emel Web Servis Tanım Dili (WSDL) önemlidir. WSDL, bir web servisinin işlevselliğini ve erişim yöntemlerini tanımlayan bir standarttır. Örneğin, bir kitap listesi alma servisinin WSDL belgesi, servisin adını, işlemlerini ve erişim protokollerini belirtir. X emel SOAP (Simple Object Access Protocol), internet üzerindeki uygulamalar arasında bilgi alışverişini yapmak için kullanılan bir protokoldür. SOAP, X emel tabanlı bir protokol olup, istemci-sunucu mimarisini kullanır. Müşteriler, SOAP mesajları aracılığıyla sunuculara istek gönderir ve sunucular da bu isteklere yanıt verir. SOAP mesajları, zarf ve gövde olmak üzere iki bölümden oluşur ve http veya https gibi protokoller aracılığıyla taşınır. X emel SOAP'un avantajları arasında platform bağımsızlığı, standart protokol olması ve güvenlik sağlaması bulunurken, karmaşıklık, performans ve veri boyutu gibi dezavantajlar da vardır. Bu nedenle, X emel Web Servisleri ve X emel SOAP'un kullanımı, uygulamanın gereksinimleri ve ihtiyaçlarına bağlı olarak değerlendirilmelidir. RDF (Resource Description Framework), web üzerindeki kaynakları tanımlamak için kullanılan ve bilgisayarlar tarafından işlenebilir veri olarak kullanılan bir standarttır. RDF, W3C'nin Anlamsal Web Etkinliğinin bir parçasıdır ve 2004 yılında kabul edilmiştir. RDF'nin kullanım alanları arasında alışveriş öğelerinin özelliklerini açıklama, web etkinliklerinin zaman çizelgelerini belirtme, web sayfalarına ilişkin bilgilerin açıklanması gibi örnekler bulunmaktadır. RDF'nin temel bileşenleri kaynaklar, özellikler ve değerlerdir. RDF, genellikle Turtle veya RDF/X emel gibi formatlarda ifade edilir ve veri esnekliği, bağlantılı veri oluşturma ve makine okunabilirlik gibi avantajlara sahiptir. Ancak, karmaşıklık, işleme maliyeti ve yetersiz kullanım gibi dezavantajları da vardır. RDF öğeleri arasında RDF Description elemanı, kaynakları ve ilişkili bilgileri temsil etmek için kullanılır. Bu elemanla bilgi kaynaklarını tanımlama, özellikleri belirtme ve ilişkileri gösterme gibi işlemler yapılabilir. RDF'de konteynerler ve koleksiyonlar da bilgi nesnelerini organize etmek ve ilişkilerini göstermek için kullanılır. RDF Şeması (RDFS), RDF'de kullanılan meta veri dilidir ve temel ontoloji kavramlarını tanımlar. Sınıflar, özellikler, alt sınıflar ve özellik kısıtlamaları gibi kavramlar RDFS içinde tanımlanır. Bu kavramlar, belirli bir alan veya uygulama için bilgi nesnelerini ve ilişkilerini tanımlamak için kullanılan uygulama sınıfları tarafından kullanılabilir. RSS (Really Simple Syndication), web sitelerinin güncel içeriklerini takip etmek için kullanılan bir sistemdir.

RSS, web sitelerinin sık gncellenen ieriklerini kullanıcıların RSS okuyucuları aracılığıyla takip etmelerine olanak tanır. Bu sayede kullanıcılar, siteleri ziyaret etmeden yeni ieriklerden haberdar olabilirler. RSS belgeleri, bir web sitesinin gncel ieriğini ieren ve ve gibi ğeleri ieren X emel belgeleridir. ğesi, RSS beslemesinin başlığını, bağlantısını ve açıklamasını ierirken, ğesi, bir makalenin veya ieriğin başlığını, bağlantısını, zetini ve yayınlanma tarihini ierir. RSS'in avantajları arasında zaman tasarrufu, gncel kalma ve bilgi dzenleme bulunurken, dezavantajları arasında teknik gereklilik, bakım zorluğu ve bilgi kirliliğı yer alır. RSS, haber siteleri, şirketler, takvimler ve bloglar gibi web siteleri iin faydalı bir araçtır.

GÖZDEN GEÇİRELİM

1. XML Web Servisleri, hangi amaçla kullanılır?

- A) Müşteri hizmetleri için
- B) Veri alışverişi ve iletişim için
- C) Oyun geliştirme için
- D) Reklam amaçlı kullanım için
- E) Haber alma amaçlı kullanım için

2. XML Web Servisleri, farklı uygulamalar arasında hangi veri alışverişini sağlar?

- A) Ses dosyaları
- B) Görüntü dosyaları
- C) Metin dosyaları
- D) XML formatında veri
- E) Video dosyaları

3. SOAP, hangi amaçla kullanılır?

- A) XML dökümanları oluşturmak için
- B) İşletim sistemi oluşturmak için
- C) Görsel öğeleri oluşturmak için
- D) Oyun geliştirmek için
- E) Veri transferi için XML tabanlı bir protokol

4. WSDL nedir?

- A) Web Sitesi Dizini
- B) Web Servis Dökümanı
- C) Yazılım Geliştirme Dili
- D) Web Servis Tanım Dili
- E) XML Tanım Dili

5. RDF Şeması (RDFS) ne için kullanılır?

- A) Bir web sayfasının tasarımını oluşturmak için
- B) Bir veritabanı şemasını tasarlamak için
- C) RDF'de bilgi nesneleri ve ilişkileri tanımlamak için
- D) Bir sunucu ağını kurmak için
- E) Bir dosya sistemini organize etmek için

6. RSS nasıl çalışır?

- A) Web siteleri RSS beslemelerini oluşturur ve kullanıcılar RSS okuyucu yazılımlarıyla bu beslemeleri takip eder.
- B) Kullanıcılar RSS beslemelerini oluşturur ve web siteleri bu beslemeleri kullanarak içeriklerini günceller.
- C) RSS beslemeleri, web sitelerinin içeriklerini otomatik olarak günceller.
- D) RSS beslemeleri, web sitelerinin görünümünü özelleştirmek için kullanılır.
- E) RSS beslemeleri, web sitelerinin güvenliğini artırmak için kullanılır.

7. RSS'de hangi versiyonlar kullanılır?

- A) RSS 0.91, RSS 1.0 ve RSS 2.0
- B) RSS 2.0, RSS 3.0 ve RSS 4.0
- C) RSS 1.0, RSS 2.0 ve RSS 3.0
- D) RSS 0.91, RSS 2.0 ve RSS 3.2
- E) RSS 0.91, RSS 1.0 ve RSS 2.0.1

Cevap Anahtarı

1	2	3	4	5	6	7
B	D	E	D	C	A	A

KAYNAKÇA

[1] Learn to code. W3Schools Online Web Tutorials. (n.d.). <https://www.w3schools.com/>

FAYDALI KAYNAKLAR VE INTERNET KAYNAKLARI

Dimes, T., & L., C. (2017). PHP. Babelcube Inc.

PHP and JSON. (n.d.). https://www.w3schools.com/php/php_json.asp

PHP Basics. (n.d.). Beginning PHP and MySQL, 55–112. https://doi.org/10.1007/978-1-4302-0299-8_3

XML – Web Services (n.d.). https://www.w3schools.com/xml/xml_services.asp

XML – WSDL (n.d.). https://www.w3schools.com/xml/xml_wsdl.asp

XML – SOAP (n.d.). https://www.w3schools.com/xml/xml_soap.asp

XML – RDF (n.d.). https://www.w3schools.com/xml/xml_rdf.asp

XML – RSS (n.d.). https://www.w3schools.com/xml/xml_rss.asp