

# REST Servis Çağrılarında Content Tipinin Belirlenmesi



# Response Content Tipinin Belirlenmesi

- Web isteği sonucunda dönülecek içeriğin formatı default durumda önce **request URI'in uzantısına**, yoksa **format request parametresine**, diğer durumda ise **Accept request header'ına** bakılarak belirlenir
- Bu ayarlar `<mvc:annotation-driven/>` elemanı built-in **ContentNegotiationManager** bean'i register etmektedir

# Response Content

## Tipinin Belirlenmesi: URI Path Ext.

GET /petclinic/pets/123.json

...

HTTP/1.1 200 OK

...

Content-Length:...

**Content-Type: application/json**

{

pet: {id:"123", ...}

}

# Response Content Tipinin Belirlenmesi: Request Parameter



GET /petclinic/pets/123?format=json

...

HTTP/1.1 200 OK

...

Content-Length:...

**Content-Type: application/json**

{

pet: {id:"123", ...}

}

# Response Content Tipinin Belirlenmesi: Accept Header

```
GET /petclinic/pets/123  
...  
Accept: application/json  
...
```

```
HTTP/1.1 200 OK  
...  
Content-Length: ...  
Content-Type: application/json  
  
{  
    pet: {id:"123", ...}  
}
```

# ContentNegotiationManager

## Konfigürasyonu

```
<mvc:annotation-driven content-negotiation-manager="contentNegotiationManager"/>
```

```
<bean id="contentNegotiationManager"  
class="org.springframework.web.accept.ContentNegotiationManagerFactoryBean">
```

```
  <property name="favorPathExtension" value="true"/>
```

```
  <property name="favorParameter" value="false"/>
```

```
  <property name="ignoreAcceptHeader" value="false"/>
```

```
  <property name="mediaTypes">
```

```
    <map>
```

```
      <entry key="json" value="application/json"/>
```

```
      <entry key="xml" value="application/xml"/>
```

```
      <entry key="html" value="text/html"/>
```

```
    </map>
```

```
  </property>
```

```
  <property name="useJaf" value="false"/>
```

```
  <property name="defaultContentType" value="text/html"/>
```

```
</bean>
```

1. request URI uzantısı'na bakılır
2. Accept header değerine bakılır
3. "format" request param değerine bakılır

MediaTypes mevcut değil ise Java Activation Framework Kütüphanesi classpath'de mevcut ise mime type belirlemek için kullanılabilir

# Response'un Farklı Tiplerde Üretilmesine Örnek

```
@RequestMapping(value="/owners/{id}", produces={"application/json", "application/xml"})  
@ResponseBody  
public Owner getOwner(@PathVariable Long id) {  
    return petClinicService.getOwner(id);  
}
```

http://localhost:8080/petclinic/mvc/owners/1.json -> application/json  
http://localhost:8080/petclinic/mvc/owners/1.xml -> application/xml  
  
http://localhost:8080/petclinic/mvc/owners/1.html -> text/html

```
@RequestMapping("/owners/{id}")  
@ResponseBody  
public String getOwnerAsHtml(@PathVariable Long id) {  
    StringBuilder builder = new StringBuilder();  
    Owner owner = petClinicService.getOwner(id);  
    builder.append("<html><body>");  
    builder.append(owner.getFirstName() + " " + owner.getLastName() + "<br>");  
    builder.append("</body></html>");  
    return builder.toString();  
}
```

# İletişim

- **Harezmi** Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- <http://www.java-egitimleri.com>
- [info@java-egitimleri.com](mailto:info@java-egitimleri.com)

