

MOBİL HABER UYGULAMASI

Mehmet Oğuz Aktaş (150201170), Mert Var (150201222)

oguzaktass@gmail.com, var_mert@hotmail.com

Kocaeli Üniversitesi
Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü

Özet

Projeye, server-client mimarisi ile RESTful web servisi ve Java web uygulamaları için kullanılan frameworkleri araştırarak başladık. Önce Spring Boot ile localhost üzerinden çalışan basit bir REST uygulaması yazdık, bu uygulamayı yazarken Spring Framework, Spring Data JPA, Spring Data REST and HAL Browser, H2 Database, Apache Tomcat teknolojilerini araştırıp öğrendik ve uygulamada kullandık. Server side veritabanı CRUD işlemleri için biri npm kullanarak Angular CLI web ve biri de Java form uygulaması olmak üzere MySQL veritabanı üzerinden çalışan 2 farklı uygulama yazdık, projede birini tercih etmek yerine ikisini birden kullandık. Android mobil uygulama tarafında, RESTful web servis ile mobil uygulama arasında bağlantıyı sağlayabilmek için Retrofit HTTP Client ve GSON kütüphanelerini kullandık. Haber görselleri için Glide ve tarih bilgileri için Prettytime kütüphanelerini kullandık. Haberleri listeleme ve kategoriye göre sıralama işlemleri için MainActivity sınıfı içinde loadJson(), initListener() ve initListener2() metodlarını yazdık. Habere ayrıntılarına gidildiğinde like, dislike ve ana menüye dön seçenekleri ile haberin görüntülenme sayısını 1 arttırarak REST servise PATCH request ile gönderilmesini sağladık.

1. Giriş

Bu projede; server-client mimarisini kullanarak bir veritabanı uygulaması ile istenen özelliklerde bir mobil haber uygulaması yazılması, bu 2 uygulama arasında RESTful web servisi kullanılarak bağlantı kurulması ve mobil uygulama üzerinden beğenme, beğenmeme ve görüntülenme sayısı bilgilerinin RESTful web servis aracılığıyla veritabanına kaydedilmesi ve server tarafında güncellenmesi istenmektedir. Ayrıca mobil uygulamada,

kullanıcının seçimine göre haber kategorilerine göre filtreleme yapılarak haberlerin listelenmesi ve her yeni haber eklendiğinde kullanıcıya bildirim gönderilmesi de istenmektedir.

Projeye önce Spring Boot ile nasıl RESTful web servisi kullanılacağı ile ilgili örnek projeleri ve eğitim kaynaklarını gözden geçirerek başladık. Spring Boot ile RESTful web servisleri kullanımı hakkında birçok örnek proje ve eğitim kaynağı gözden geçirdikten sonra basit bir arayüz tasarladık ve bu kısmı hızlıca tamamlayarak veritabanı uygulamasına geçtik. Veritabanı uygulaması için üzerinde ayrı ayrı çalıştığımız 2 farklı uygulama geliştirdik. npm (Node.js Package Manager) ile çalışan Angular CLI web uygulaması ve Java form uygulaması olmak üzere 2 veritabanı uygulamasından birini projede kullanmak üzere tercih etmek yerine ikisini birden projeye entegre ettik. Android mobil uygulama tarafına geçtiğimizde, bu konuda yeterli deneyimimiz olmadığı için öncelikle Android eğitim kaynaklarını araştırarak başladık. Bazı örnek uygulamaları gözden geçirip biraz fikir sahibi olduktan sonra, basit bir haber listeleme arayüzü tasarlayarak client tarafını yazmaya başladık. RESTful web servis ile bağlantıyı sağlamak için Retrofit HTTP Client kütüphanesini kullandık, kullanıcı tarafından değiştirilen like, dislike ve görüntülenme sayılarını veritabanına kaydettik.

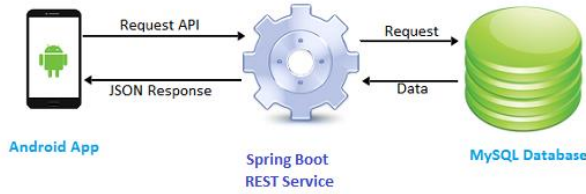
2. Temel Bilgiler

Uygulamayı geliştirirken Spring Boot 2.1.4, Spring Framework 5.1.6, Maven, npm, Angular CLI, RESTful Web Services, MySQL 8.0.15, MySQL JDBC Driver ve MySQL Connector Java, Spring Data JPA, Spring Data REST, Apache Tomcat 9.0.17 teknolojilerini; Java EE Development Kit 8u201 sürümüyle NetBeans 8.2 IDE (NP SpringBoot and Git Remote eklentileriyle birlikte),

IntelliJ IDEA 2019.1.1, Android Studio 3.4 IDE'lerini; REST servis ile mobil uygulama arasındaki bağlantıyı kontrol etmek için Retrofit HTTP Client 2.4.0, haber görsellerini düzenlemek için Glide 4.9.0 ve Java.Date() tipinde nesneleri düzenlemek için Prettytime 4.0.2 kütüphanelerini kullandık. Kullandığımız web teknolojilerinin yanında JSF, Spring MVC, Node.js ve PHP gibi bazı web teknolojilerini de araştırıp ve bazılarını deneyip projede kullanmamayı tercih ettik.

3. Yazılım Mimarisi

Veritabanı CRUD işlemlerinin yapıldığı veritabanı admin uygulaması, Spring Boot RESTful web servis uygulaması ve Android mobil uygulama olmak üzere proje 3 kısımdan oluşuyor. Veritabanı uygulamasını; npm ve Angular CLI ile TypeScript kullanarak bir web uygulaması, Java form uygulaması olmak üzere 2 farklı şekilde yaptık.



Şekil 1: Genel sistem mimarisi

Spring Boot REST serviste News (entity paketi içinde), NewsRepository (repository paketi içinde), NewsController (controller paketi içinde) classlarını kullandık. Ayrıca projeye başlarken uygulamayı yazmak için pom.xml dosyası içinde uygulamayı konfigüre etmek için belirlediğimiz bazı dependency'ler şunlar; spring-boot-starter-data-jpa, spring-boot-starter-data-rest, spring-boot-starter-jdbc, spring-boot-starter-jersey, spring-boot-starter-web, spring-boot-starter-web-services, mysql-connector-java.

Spring Boot uygulaması içinde kullandığımız classlar ve metotlardan bazıları şunlar;

- News: Bu entity class ile application.properties içinde belirlediğimiz MySQL bağlantı bilgileri ile MySQL veritabanı bağlantısı kurarak class içinde @Entity, @Table(name = "news") tanımlarıyla belirttiğimiz id, imageurl, header, content, category, publish_date, likes, dislikes, views sütunlarından oluşan veritabanının otomatik oluşturulmasını sağladık. Bu özellikleri class içinde başına @Column(name

= "id") gibi tanımlama belirterek ayrı ayrı değişken olarak tanımladık. Id özelliğinin başına @Id ve @GeneratedValue(strategy = GenerationType.AUTO) tanımlamalarını belirterek Primary key olarak kullanılmasını sağladık.

- NewsRepository: Spring Boot REST uygulamalarında kalıp olarak kullanılan ve JpaRepository interface'inden extend edilen bu interface ile Spring Data JPA özelliklerinin kullanılmasını sağladık.
- NewsController: @RestController tanımını kullanarak Controller classı olarak belirlediğimiz bu class içinde @RequestMapping, @GetMapping ve @PostMapping tanımlamaları (annotation) ve List<News> tipindeki listNews() metodu ile haber listesine ulaşılabilmesini sağladık.

Veritabanı uygulaması olarak yazdığımız Java form uygulaması için NewsAdmin isimindeki JFrame classını yazdık. Bu class içinde kullandığımız metotlardan bazıları şunlar;

- getConnection(): MySQL veritabanı bağlantısını sağlamak için yazdığımız Connection tipindeki metot.
- addNews(): Haber başlığı, içeriği, resim URL'si, yayınlanma tarihi gibi bilgileri parametre olarak alıp "INSERT INTO" komutuyla veritabanına yeni kayıt eklemeyi sağlayan metot.

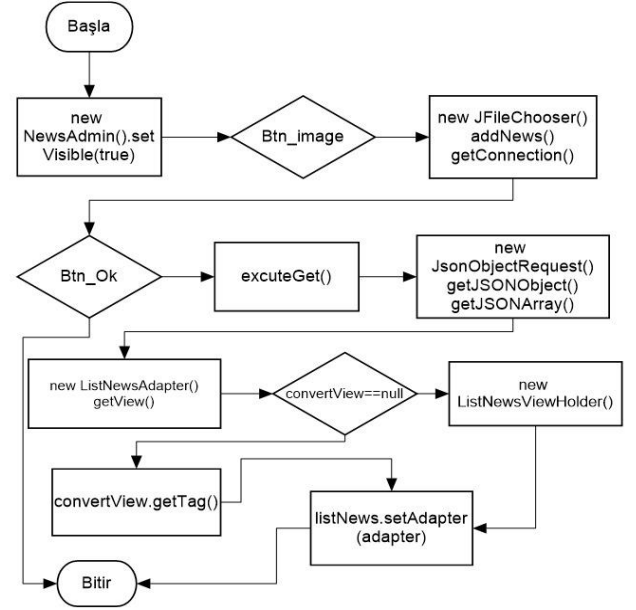
Projeyi yaparken en çok zamanımızı alan kısım olan kısım, Android mobil haber uygulaması kısmı oldu. Android uygulamasında default olarak kullanılanlar dışında build.gradle dosyasında ayıriyetten belirlediğimiz dependency'ler şunlar; com.github.bumptech.glide:glide:4.9.0, com.github.bumptech.glide:compiler:4.7.1, com.squareup.retrofit2:retrofit:2.5.0, com.squareup.retrofit2:converter-gson:2.5.0, org.ocpssoft.prettytime:prettytime:4.0.2.Final. Kullandığımız önemli class, interface ve metotlardan bazıları da şunlar;

- loadJson(): MainActivity içinde bulunan void tipindeki bu metot ile Retrofit Call classını kullanarak GET metodu ile RESTful API'den haberlerin çekilmesini ve oluşturduğumuz Adapter'a kaydedilmesini sağladık. Ayrıca haberler listelendikten sonra initListener() metoduyla üzerine tıklanarak haber ayrıntısına gidilmesini ve initListener2() metoduyla haber kategorisine göre filtreleme yapılmasını

kontrol ettik. Projede kullandığımız en önemli metod buydu.

- **İnitListener():** Uygulama ama menüsünde haberin üzerine tıklandığında açılacak olan haber detayları arayüzünde gösterilmek üzere, loadJson() metoduyla çekilen haber bilgilerinin Intent classı ile NewsDetailActivity Activity classına gönderilmesini sağladık.
- **İnitListener2():** Spinner kullanarak Drop Down Menu'den seçilecek olan Eğitim, Spor, Ekonomi, Gündem kategorilerinin oluşturduğumuz selectCategory değişkenine kaydedilmesini, bu kategoriye göre loadJson() metodunun tekrar çalıştırılarak filtreleme yapılmasını sağladık.
- **RESTClient:** Bu class ile Retrofit ve REST servisin genel özelliklerini tanımladık. BASE_URL değişkeni oluşturarak localhost-ip:8080 şeklinde ana URL'yi belirledik.
- **RESTInterface:** Yine Retrofit kütüphanesi classlarını kullandığımız bu interface içinde, REST servisten GET ve PUT metotları ile bilgi çekme Java metotlarını (getJSON()) ve putNews()), PATCH metodu ile bilgi güncellemeye yarayan patchNews() Java metodunu tanımladık.
- **News:** Bu class ile REST servisten alınan haber bilgilerinin id, category, header, content, url, imagetourl, publishDate, likes, dislikes, views özellikleriyle kaydedilmesini sağladık.
- **Adapter:** RESTful API'den alınan bilgilerin düzenli biçimde arayüzde görünmesini sağlamak için kullandığımız önemli classlardan biri.

4. Akış Şeması



Şekil 2: Akış şeması

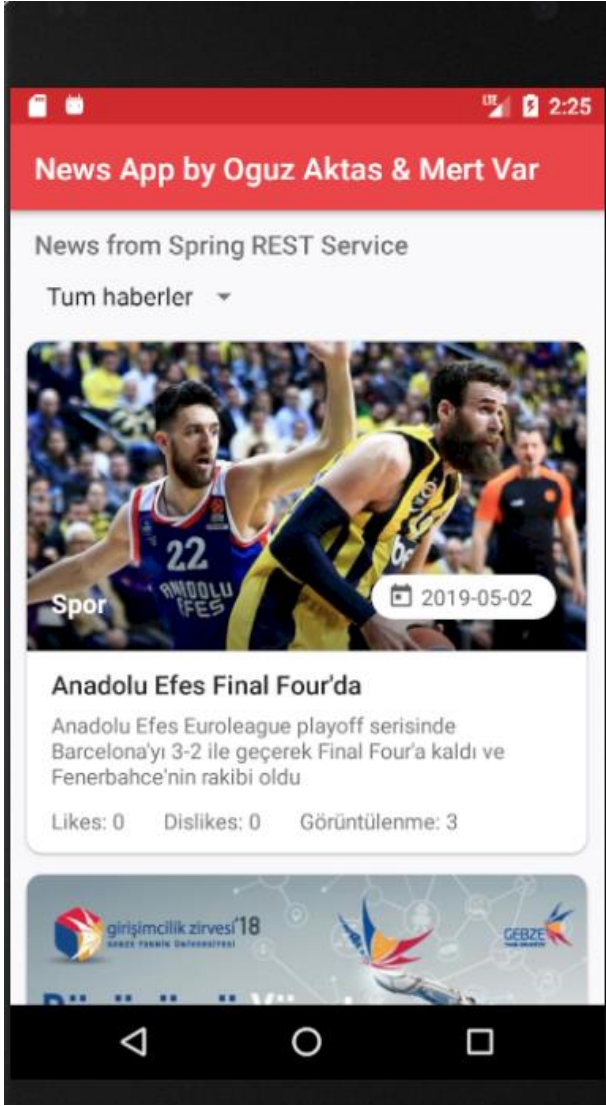
5. Kullanıcı Kataloğu

Spring Boot ile oluşturduğumuz RESTful API'nin genel görünümü şu şekilde;

```
localhost:8080/news x +
< - > ↺ ↻ ↶ ↷ ⓘ localhost:8080/news
{
  "_embedded": {
    "news": [ {
      "imageurl": "https://cdn2.iddaa.com/iddaa/sport-news/sporhaber.jpg",
      "header": "Fenerbahce'nin Yeni Transferi",
      "content": "Fenerbahcenin yeni transferi bugun Istanbula geldi. Fener",
      "likes": 0,
      "dislikes": 1,
      "views": 12,
      "publishDate": "2019-04-22",
      "id": 1,
      "_links": {
        "self": {
          "href": "http://localhost:8080/news/1"
        }
      },
      "news": {
        "href": "http://localhost:8080/news/1"
      }
    }, {
      "imageurl": "http://marastahaber.com/wp-content/uploads/2016/07/ekono",
      "header": "Dolar Ne Kadar Oldu?",
      "content": "Piyasaların tatil olduğu bugünlerde dolar 5,96 lira sevi",
      "category": "Ekonomi",
      "likes": 1,
      "dislikes": 0,
      "views": 7,
      "publishDate": "2019-04-25",
      "id": 2,
      "_links": {
        "self": {
          "href": "http://localhost:8080/news/2"
        }
      },
      "news": {
        "href": "http://localhost:8080/news/2"
      }
    }
  ]
}
```

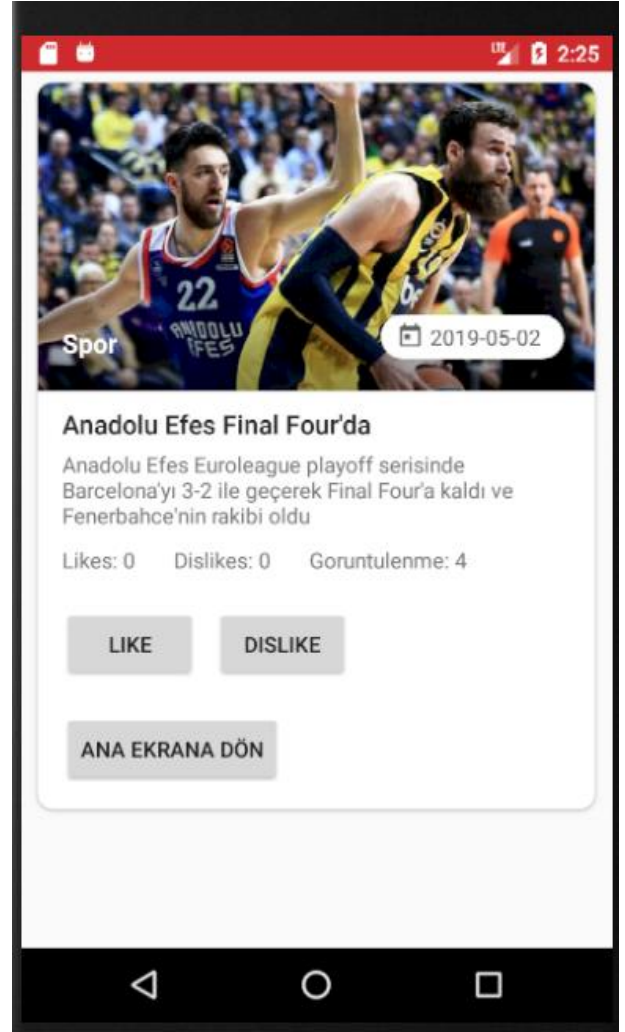
Şekil 3: RESTful API

Buradan <http://localhost:8080/news/1> gibi id numarası verilerek URI ile haberlere ulaşılabilir. Android uygulamasında ana ekran şu şekilde;



Şekil 4: Android haber uygulaması ana ekran

Üstte bulunan Drop Down Menu'den haber kategorisi (Spor, Ekonomi, Eğitim, Gündem) seçilerek filtreleme yapılabilir. Bir haberin üzerine tıklandığında ise haber ayrıntısına gidilir ve şu menü açılır;



Şekil 5: Android haber ayrıntıları ekranı

6. Sonuçlar

Projeyi geliştirirken gereksiz kod tekrarları yapmamaya ve gereksiz yere bellekten alan ayırmamaya, her aşamada kod yazarken algoritma karmaşıklığına ve programın hızlı çalışmasına dikkat ettik. Günümüzde yazılım sektöründe kullanılan ve daha önce pek kullanmadığımız Maven, Spring Boot, Spring MVC, RESTful Web Services, Angular, node.js gibi Java EE ve diğer web teknolojilerini araştırıp kullanarak bu konudaki bilgi düzeyimizi arttırdık. Aynı şekilde pek deneyimimiz olmayan Android uygulama geliştirme konusunda da eğitim kaynaklarından ve örnek projelerden yararlanarak bilgi düzeyimizi arttırdık ve belli bir düzeyde deneyim kazandık. Bu açıdan projenin, bitirmek için fazla zaman olmaması ve zorlanmamıza rağmen bize oldukça yararlı olduğunu söyleyebiliriz.

7. Kaynakça

- [1] <https://stackoverflow.com/>
- [2] <https://developer.android.com/docs>
- [3] <https://spring.io/guides>
- [4] <https://mvnrepository.com>
- [5] <https://docs.spring.io/spring-boot/docs/2.1.4.RELEASE/reference/htmlsingle/> (Erişim tarihi: 25.04.2019)
- [6] https://www.tutorialspoint.com/spring_boot/ (Erişim tarihi: 25.04.2019)
- [7] <https://www.baeldung.com/rest-with-spring-series> (Erişim tarihi: 25.04.2019)
- [8] <https://medium.com/kodgemisi/spring-boot-ile-%C3%B6rnek-web-uygulamas%C4%B1-914c94c9099f> (Erişim tarihi: 26.04.2019)
- [9] <https://www.baeldung.com/spring-boot-angular-web> (Erişim tarihi: 26.04.2019)
- [10] https://www.youtube.com/playlist?list=PLYxzS5yYQI31tSVpCDsI6Y2J_YjHhE8 (Erişim tarihi: 26.04.2019)
- [11] <https://www.udemy.com/android-studio-ile-uygulama-gelistirme-egitimi-2/> (Erişim tarihi: 26.04.2019)
- [12] <https://www.mobilhanem.com/android-egitimleri/> (Erişim tarihi: 26.04.2019)
- [13] https://www.tutorialspoint.com/android/android_json_parser.htm (Erişim tarihi: 27.04.2019)
- [14] www.androiddeft.com/retrofit-android/ (Erişim tarihi: 27.04.2019)
- [15] <https://square.github.io/retrofit/2.x/retrofit/retrofit2/Call.html> (Erişim tarihi: 27.04.2019)
- [16] <https://code.tutsplus.com/tutorials/sending-data-with-retrofit-2-http-client-for-android--cms-27845> (Erişim tarihi: 27.04.2019)
- [17] <https://square.github.io/retrofit/2.x/retrofit/retrofit2/Call.html> (Erişim tarihi: 27.04.2019)
- [18] <http://www.jsonschema2pojo.org/> (Erişim tarihi: 27.04.2019)
- [19] <https://guides.codepath.com/android/Displaying-Images-with-the-Glide-Library> (Erişim tarihi: 28.04.2019)
- [20] <https://www.mkyong.com/android/android-spinner-drop-down-list-example/> (Erişim tarihi: 28.04.2019)