

## Açıklama

Şirketlerin ve ya şirket çalışanlarının iş ilan oluşturabilecekleri bir web uygulaması tasarlanmıştır. Uygulamanın Backend’de Spring Boot kullanılırken Frontend’de React kullanılmıştır. Backend’de Restful Web Service çalışmaktadır ve bu service Http Requestleri hem React hem de Postman aracılığı ile yapılmaktadır.

Bu web uygulaması alışıla geldik CRUD(Create-Read-Update-Delete) işlemleri gerçekleştirmek ile birlikte http’ni en çok kullanılan GET, POST,PUT, DELETE gibi fiillerini kullanmaktadır. Aşağıda uygulama için tanımlı olan Requestler verilmiştir.

### Advertisement

- -GET /advertisements
- -Delete /advertisements/{id}
- -POST /advertisements
- -Put /advertisements/{id}

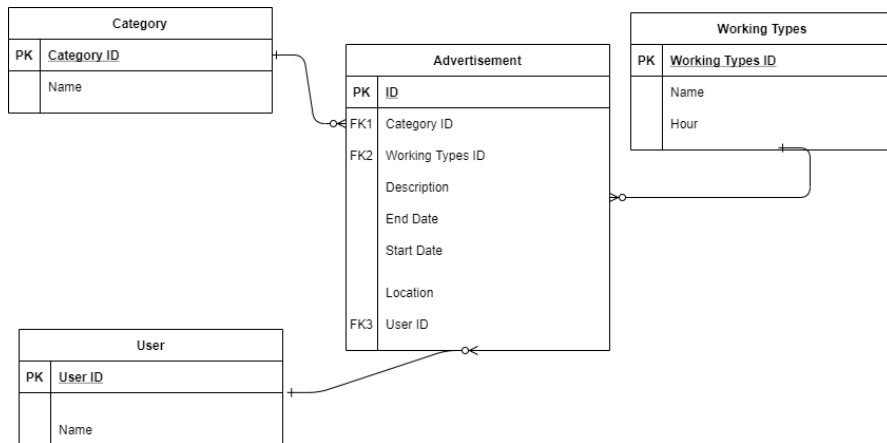
### Category

- -GET /category
- PUT / category/{id}
- -Delete /category /{id}
- -POST / category

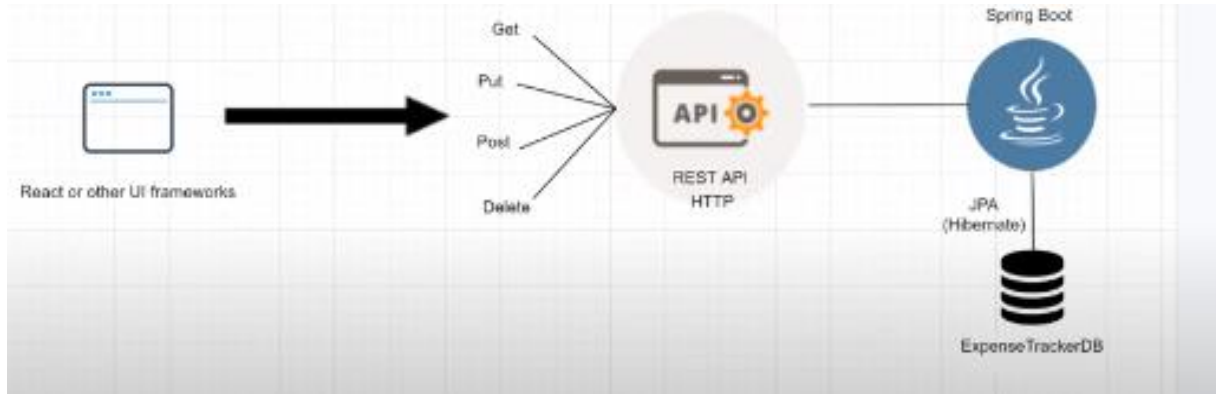
### Working Type

- -GET /workingtypes
- -Delete / workingtypes /{id}
- -PUT / workingtypes /{id}

Bunun dışında bulun tüm veri tabanı bilgiler Sql sorguları yazılarak girilmiştir. Aşağıda veri tabanı ilişkileri ve tablolar verilmiştir.



Aşağıdaki Temsili bir görseldir gerçekleştirdiğimiz web uygulaması ile alakası bulunmamaktadır.



Bunu yanı sıra raporun devamında kullanılan teknolojilerden ve bize kazandırmış olduğu avantajlardan bahsedilmiştir.

## Maven



Apache firması tarafından geliştirmiş olduğu bir teknolojidir. Maven, yazılım inşa sürecini otomatikleştirmeyi sağlayan bir araçtır. Projenin bağımlıklarını yönetmemizi kolaylaştırır. Maven build işlemlerimizi sadeleştirmemizi, standart oluşturmamızı ve projelerin kütüphane bağımlılığını ortadan kaldırmamızı sağlar. Basit bir ifade ile projelerimize artık kütüphaneleri teker teker yüklemek yerine dependency (bağımlılık) yükleriz.

## Pom Nedir?

Proje Nesnesi Modeli(Project Object Model) proje hakkında bilgileri ve konfigürasyon detaylarının tutulduğu XML dosyasıdır ve maven tarafından yazılımın inşası için kullanılır. Pom ayrıca hedefleri ve eklentileri içerir. Maven bir işlemi gerçekleştirirken ilgili dizinde POM dosyasını arar ve gerekli yapılandırma bilgilerini alır ve bu doğrultuda işlemi gerçekleştirir.



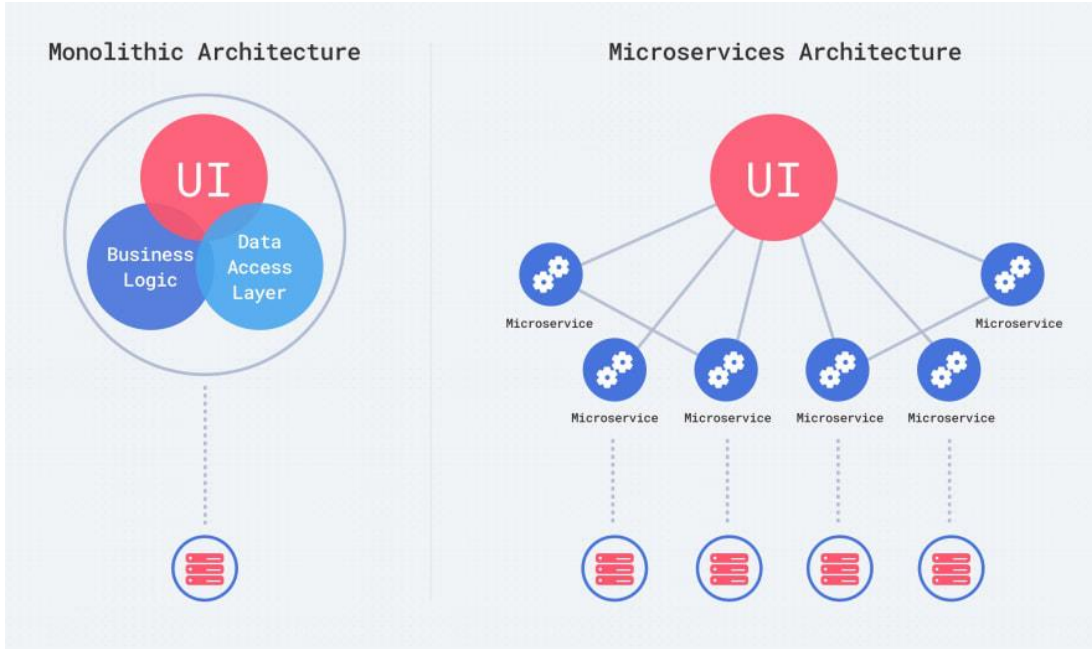
## Spring Boot Nedir?

Spring Boot, Spring tabanlı web uygulamaları geliştirmemize olanak tanıyan bir frameworkdür. Spring Boot ile gereksiz kod kalıplarından kurtulur ve sadece gerekli olan kodları yazarız. Spring Boot'un bize sağlamış olduğu iki önemli avantaj vardır. Bunlardan birincisi Spring Boot bizi XML Konfigürasyonu ile uğraşmaktan kurtarır. Arka planda tanımlı olan kütüphaneler yardımıyla bizim için bu işi Spring Boot halleder. İkincisi ise Spring Boot Web sunucusu olarak tomcat ile birlikte gelir. Böylelikle web uygulamamızı sanki java uygulamalarımız çalıştırıyor gibi çalıştırabiliriz.

Spring Boot uygulama geliştirme sürecimizi hızlandırmaya yardımcı olur. Bu nedenledir ki microservice mimarilerinde yaygın bir şekilde kullanılır.

## Microservice Mimarisi

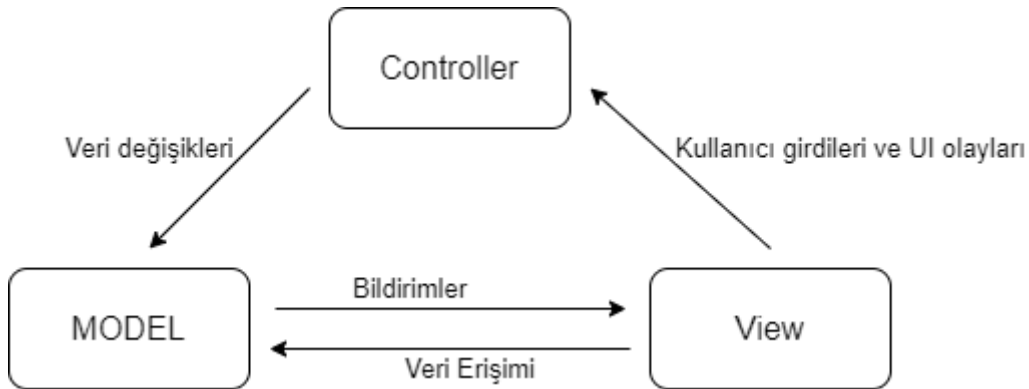
Microservice mimaris bir yazılım geliştirme tekniğidir. Servis odaklı bir mimari (SOA) üzerine kurulmuş bir mimaridir. Micro servisler bize büyük kapsamlı uygulamalarımızı ayrıştırmamıza ve küçük parçalara bölmemize olanak tanır. Bu küçük parçalar belirli bir özelliği ya da fonksiyonu sağlayan, birbirinden bağımsız, görevleri kesin tanımlanmış yazılım parçalarıdır.



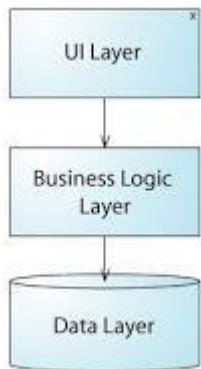
Bu mimarinin yazılımı küçük parçalara bölüyor olmasını bize getirmiş olduğu çok büyük avantajlar vardır. Yazılım küçük parçalara ayrılması ile kolay yönetilebilir hale gelir ve aynı zamanda ölçeklenmesi de bir o kadar kolay olacaktır.

## MVC ve Katmanlı Mimari

MVC(Model View Controller) uygulamamızı 3 ayrı mantıksal bileşene ayıran bir mimari desenidir. Burada Model kullanıcı ara yüzü yani View tarafından görüntülenen veriyi ifade eder. View kullanıcıya gösterilecek olan veriye model üzerinden erişir. Controller ise burada interface gibi çalışır ve model ile view arasındaki iş mantığını yürütür. Kullanıcı tarafından gelen istekler Controller tarafından karşılanır,yönetilir ve ilgili model verisinide içerecek şekilde View'a yönlendirilir.



MVC ve frontcontroller'ın etkisi ile Spring projelerinde çok katmanlı mimari yaygın olarak kullanılmaktadır. Çok katmanlı mimari karmaşık sistemleri alt katmanlara ayırarak daha kolay yönetilebilir hale getirmektedir.



UI, Servis ve Veri erişimi gibi 3 ana katmanından oluşmaktadır. UI katmanı kullanıcı isteklerinin işlenmesi ve kullanıcı arayüzü oluşturmaktan sorumludur.

Servis katmanının uygulamanın iş mantığının çalıştığı ve işlerin yürütüldüğü katmandır. İstemciden(client) gelen çağrılara cevap verir.

Veriye erişim katmanıdır. Bu katmanda veriye erişim, veriyi saklama ve güncelleme gibi işlemler gerçekleştirilir.

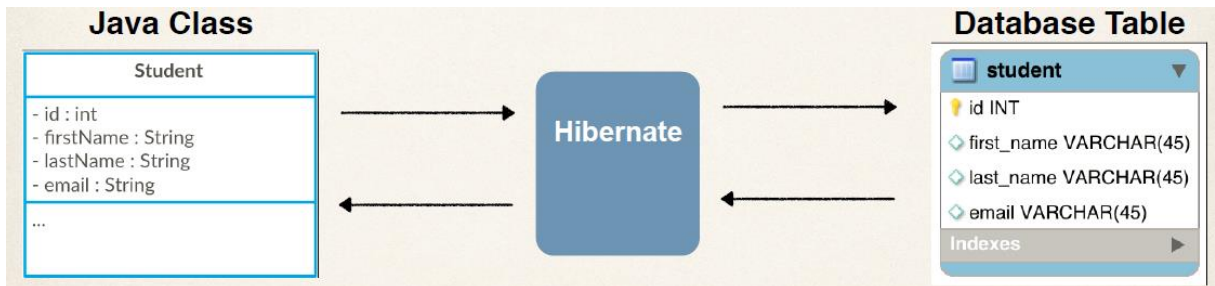
Her katman bir altında bulunan katman ile erişim halindedir ve erişim arayüzler üzerinde gerçekleştirilir.

## JPA ve Hibernate

JPA(Java Persistence API) java nesnelerimizi veri tabanında ilişkisel bir tabloya eklenebilmesini, o tablodan okunabilmesine ve güncelleme yapılabilmesine yani alışık olduğumuz veri tabanı işlemlerini yönetilebilesini sağlar. Object-Relation Mapping yapmamızı olanak sağlar. Yani objelerimiz ile veritabanında sistemimizdeki tablolar arasında köprü görevi kurulmasını sağlar. Veritabanında bulunan tablolar uygulamamızdaki sınıflara (class), tablodaki klonlarımız ise sınıfın her bir değişkenine (field) karşı düşmektedir. Böylelikle orm aracılığı ile karmaşık sql sorgularında kurtulmuş ve işlem yükümüzü hafifletmiş oluruz.



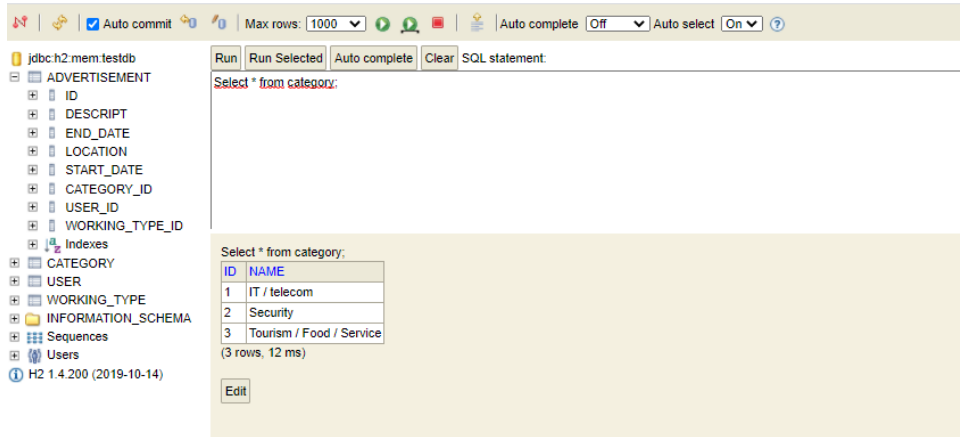
Hibernate ise Java programlama dili için geliştirilmiş olan bir ORM çözümüdür. Veritabanı ile ve sql sorguları ile daha az uğraşmamızı sağlar. Tanımlı anotasyonları kullanarak veritabanı kısıtlarımızı sınıflar üzerinde tanımlar. Aynı zamanda veritabanına ait ilişkilerde sınıflar üzerinde tanımlanmaktadır.



## H2 Database

H2 bir gömülü (in memory) veritabanı çeşididir. Uygulamaların ilk geliştirme süreçlerinde ve test amacı ile kullanılmaktadır. Her uygulama başlatıldığında kayıtlar silinmektedir. H2

veritabanı tarayıcı üzerinde çalışan bir console ekranı da sunmaktadır. Bu açıdan incelendiğinde uygulamamın yayını sırasında kullanmaya uygun olmayan bir veritabanı olsa da projelerin geliştirme aşamasında kullanmak için oldukça uygun bir veri tabanıdır.



## React

React interactive kullanıcı arayüzleri (UI) oluşturmamızı sağlayan bir JavaScript kütüphanesidir. Sanılanın aksine React bir framework değil kütüphanedir. React Facebook tarafında geliştirilmiştir ve 2013’de open source olarak piyasaya sürülmüştür. React MVC uygulamalarımızın View kısmını ifade etmektedir. Peki ama Reacta bu popülerliğini getiren yani onu bu kadar kullanışlı kılan şeyler nedir?

React Virtual DOM yapısına sahiptir, bu sanal DOM gerçek DOM yapısının bir yansımasıdır. Dom üzerinde sürekli değişiklik yapmak tüm DOM ağacının tekrardan üzerinden geçirilmesini gerektirdiği için masraflı bir iştir bu yüzden Virtual DOM kullanılır. Virtual DOM diff algoritması kullanarak gerçek DOM üzerinde değişikliği minimum adımda gerçekleştirmeye çalışır. Bu sayede daha az işlemle ile gerçek DOM üzerinde değişiklikler yapılmış olur.

React bileşenlerden (component) oluşur. Componentlar, uygulamamızı küçük ve tekrar kullanılabilir parçalara ayırmamızı sağlar. Bir tane root Component bulunur ve bu component’ın child componentları bulunur. React componentları birleşerek asıl Frontend uygulamamızı oluşturur. Aşağıdaki görselde bu durum illüstre edilmiştir.



Bu componentlar iki önemli parçadan oluşur bunlar state ve renderdır. State ilgili component'ın dahili saklama alanıdır, dışardan bu alana dışardan erişilemez ve değiştirilemez. State yapısına sahip olan componentlere statefull adı edilirken, state sahip olmayan componentlere stateless denir.

React backend'de çalışan servisimize asenkron HttpRequestlerinde bulunur ve bu requestlerin sonucunda elde ettiği veriyi state bilgisine kayıt ederek kullanıcıya gösterir ya da kullanıcıdan alınan ilgili bilgileri işlenmesi için backend'e Post Request ile gönderir.



## Postman

Postman yazmış olduğumuz backend servisleri test etmek, dökümanete etmek, servisin cevaplarını incelemek için kullanırız. API test etmek için bütün HttpRequestlerinde bulunabiliriz. Aşağıda **Advertisements** için gerçekleştirilen GET Request ve dönen Response gösterilmiştir.

The screenshot shows a Postman interface with a GET request to `localhost:8080/advertisements`. The request is labeled "Request" with a red bracket. The response is a JSON object, labeled "Response" with a red bracket. The response status is 200 OK, Time: 2.88 s, Size: 688 B. The response body is displayed in JSON format:

```
1 {
2   "startDate": "2020-08-27T11:40:28.868+00:00",
3   "endDate": "2020-08-27T11:40:28.868+00:00",
4   "descript": "Software Developer",
5   "location": "New york",
6   "category": {
7     "id": 1,
8     "name": "IT / telecom"
9   },
10  },
11  "workingType": {
12    "id": 1,
13    "workingType": "Full Time ",
14    "workingHour": 9
15  }
```

## UYGULAMAYA İLİŞKİN GÖRESELLER

Aşağıdaki görselde yeni kayıt'ın nasıl ekleneceği gösterilmiştir.

[Job advertisement](#) [Home](#) [Categories-Types](#) [Advertisement](#) [Company](#)

### Advertisement

Description:

Category:

DeadLine:

Working Type:

Location:

*Yeni kayıt*

#### Advertisement List

Description	Location	End Date	Type	Category	Action
Software Developer	New york	27/08/2020	Full Time	IT / telecom	<input type="button" value="Delete"/>
Daily Waiter	Los Angeles	27/08/2020	Part Time	Tourism / Food / Service	<input type="button" value="Delete"/>

Aşağıda yeni kayıt ile birlikte kayıt silme işleminin nasıl gerçekleştirileceği gösterilmiştir.

[Job advertisement](#) [Home](#) [Categories-Types](#) [Advertisement](#) [Company](#)

### Advertisement

Description:

Category:

DeadLine:

Working Type:

Location:

#### Advertisement List

Description	Location	End Date	Type	Category	Action
Software Developer	New york	27/08/2020	Full Time	IT / telecom	<input type="button" value="Delete"/>
Daily Waiter	Los Angeles	27/08/2020	Part Time	Tourism / Food / Service	<input type="button" value="Delete"/>
Daily Security	İstanbul	27/08/2020	Part Time	Security	<input type="button" value="Delete"/>



## Advertisement

Description:

Category: IT / telecom

DeadLine: 08/27/2020

Working Type: Full Time

Location:

Save

Cancel

### Advertisement List

Description	Location	End Date	Type	Category	Action
Daily Waiter	Los Angeles	27/08/2020	Part Time	Tourism / Food / Service	Delete
Daily Security	İstanbul	27/08/2020	Part Time	Security	Delete

İlgili kategorilerin ve Çalışma Tiplerinin listelenmesi (aynı zamanda select tagları içinde bu bilgiler asenkron bir şekilde çekilmektedir.)

### Category List

ID	Name
1	IT / telecom
2	Security
3	Tourism / Food / Service

### Working Types List

ID	Name	Hour
1	Full Time	9
2	Part Time	4
3	Intern	12

## Kaynakça

[https://dev.to/alex\\_barashkov/microservices-vs-monolith-architecture-4l1m](https://dev.to/alex_barashkov/microservices-vs-monolith-architecture-4l1m)

<http://kuryaz.com/2017/09/19/katmanli-mimari/>

<https://okanuzun.com/hibernate-ve-spring-data-jpa-arasindaki-farklar/>

<https://koraypeker.com/2018/03/11/derinlemesine-apache-maven-i/>