



# NAMAZ VAKİTLERİ REST SERVİSİ

[awqatsalah.diyamet.gov.tr](http://awqatsalah.diyamet.gov.tr)



HAZIRLAYAN

Kamil KIRDI  
Çözümleyici

Diyamet İşleri Başkanlığı  
Din İşleri Yüksek Kurulu

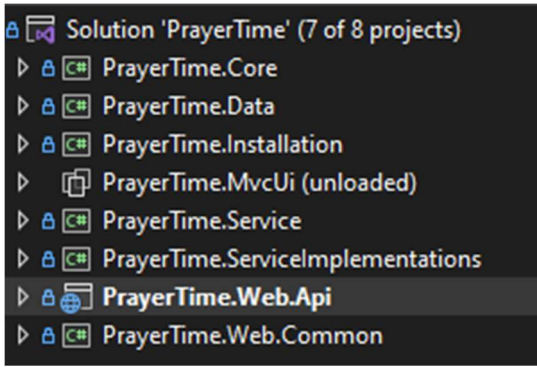
## İçindekiler

1. Genel Bilgiler .....	2
2. Veri Yapısı .....	2
3. Kimliklendirme ve Güvenlik Yapısı .....	4
4. Loglama (Günlük Verisi) Yapısı .....	4
5. İstek Kotası.....	5
6. EndPoint Adres Listesi.....	6
6.1. Auth (Authorization İşlemleri) .....	6
6.1.1. /api/Auth/Login .....	6
6.1.2. /api/Auth/RefreshToken/{RefreshToken} .....	6
6.2. DailyContent (Günlük İçerik İşlemleri) .....	7
6.2.1. /api/DailyContent (Günlük Ayet Hadis Dua Servisi).....	7
6.3. Place (Ülke / Eyalet / Şehir Bilgileri).....	7
6.3.1. /api/Place/Countries (Ülke Listesi) .....	7
6.3.2. /api/Place/States (Eyalet Listesi).....	8
6.3.3. /api/Place/States/{countryId} (Ülkeye Göre Eyalet Bilgileri).....	8
6.3.4. /api/Place/Cities (Şehir Listesi).....	9
6.3.5. /api/Place/Cities/{stateId} (Eyalete Göre Şehir Listesi) .....	9
6.3.6. /api/Place/CityDetail/{cityId} (Şehir Detay Bilgisi).....	9
6.4. PrayerTime (Namaz Vakti Bilgileri) .....	10
6.4.1. /api/PrayerTime/Daily/{cityId} (Şehir'e Ait Günlük Namaz Vakitleri).....	10
6.4.2. /api/PrayerTime/Weekly/{cityId} (Şehir'e Ait Haftalık Namaz Vakitleri) .....	10
6.4.3. /api/PrayerTime/Monthly/{cityId} (Şehir'e Ait Aylık Namaz Vakitleri).....	10
6.4.4. /api/PrayerTime/Eid/{cityId} (Şehir'e Ait Bayram Namazı Vakitleri) .....	11

## 1. Genel Bilgiler

Uygulama API<sup>1</sup> servisi olarak tasarlanmıştır. Yazılım dili olarak C# ile yazılmıştır. Ortam olarak ise açık kaynak yapıya sahip .Net Core<sup>2</sup> teknolojisi kullanılmıştır. Uygulamanın, bilinen bütün veri tabanlarına desteği olmakla birlikte, mevcut PostgreSQL tercih edilmiştir. Veri tabanı ve Uygulama arasındaki iletişim aracı (Object Relational Mapping<sup>3</sup>) olarak açık kaynak yapıya sahip Entity Framework Core<sup>4</sup> tercih edilmiştir.

Ayrıca uygulama geliştirme sürecinde SOLID<sup>5</sup> kurallarına riayet edilmiş, tasarım deseni olarak Repository Pattern<sup>6</sup> kullanılarak bütün bağımlılıklar soyut bir yapıya kavuşturulmuştur. Yine Şekil 1'de görüleceği üzere katmanlı bir mimari modeli benimsenmiş ve her katman yapacağı iş itibarıyla birbirinden soyutlanmıştır. Bu sayede bu uygulamayı ileride geliştirecek yazılımcının kolay bir biçimde projeye entegre olması hedeflenmiştir.



Şekil 1 Proje Katman Yapısı

## 2. Veri Yapısı

Kullanacağınız Api servisi bir rest servistir. Yapılan isteklerin tamamının sonucu Şekil 2'de gösterilen IActionResult modeli ile döndürmektedir.

<sup>1</sup> API'ler, iki yazılım bileşeninin belirli tanımlar ve protokoller aracılığıyla birbiriyle iletişim kurmasına olanak tanıyan mekanizmalardır. Örneğin, meteoroloji müdürlüğünün yazılım sistemi, günlük hava durumu verilerini içerir. Telefonunuzdaki hava durumu uygulaması, API'ler aracılığıyla bu sistemle "konuşur" ve telefonunuzda size günlük hava durumu güncellemelerini gösterir.

<sup>2</sup> .NET topluluğu tarafından geliştirilmiş ve geliştirilmeye devam eden, Github üzerinde bulunan açık kaynaklı, çapraz platform geliştirme yapmaya olanak sağlayan bir platformdur. .Net Core çapraz geliştirmeye verdiği olanak sayesinde Windows, macOS ve Linux'u destekler. Ayrıca bulut ve IoT uygulamaları oluşturulmasında kullanılabilir.

<sup>3</sup> Nesne odaklı paradigma kullanarak bir veri tabanından veri sorgulamanıza ve işlemenize izin veren bir tekniktir.

<sup>4</sup> Entity Framework (EF) Core, popüler Entity Framework veri erişim teknolojisinin basit, genişletilebilir, açık kaynaklı ve platformlar arası bir sürümüdür.

<sup>5</sup> SOLID yazılım prensipleri; geliştirilen yazılımın esnek, yeniden kullanılabilir, sürdürülebilir ve anlaşılır olmasını sağlayan, kod tekrarını önleyen ve Robert C. Martin tarafından öne sürülen prensipler bütünüdür.

<sup>6</sup> Repository Design Pattern, veritabanı sorumluluğunu üstlenen sınıfı tasarlarlarken bir standart üzerine oturtmayı hedefleyen ORM(Object Relational Mapping) araçlarıyla kombine edilerek sorgusal anlamda az sayıda metotla yüksek seviyede veri erişim imkanı sağlayan bir strateji üzerine kurulu tasarım desendir.

```
IResult {
  success      boolean
              readOnly: true
  message      string
              nullable: true
              readOnly: true
}
```

Şekil 2- IResult nesnesi

Yapılan istek; sonucunda geriye data döndürüyorsa; bu model içine; *T(Generic)* tipinde Şekil 3'de görülebileceği üzere *Data* property eklenmektedir.

```
Response body
{
  "data": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8  
IzI1b0dHA6Ly9zY2h1bWZzLnhtbHNvYXAub3JnL3dzLzIwMjUvMDUvaWR1b  
aXR5L2NsYm1tcy9yb2x1IjoiaXN0Ij09IiwiaWF0IjoiMj01NzUyMjI1MjI1  
miS_P1QCgCi-IE0VJAHvB2ikitoY",
  "success": true,
  "message": null
}
```

Şekil 3 IResult nesnesinin veri ile birlikte geri döndürdüğü yapı.

Web API geliştirmede en önemli ihtiyaçlardan biri dokümantasyon ihtiyacıdır. Çünkü API metodlarının ne işe yaradığı ve nasıl kullanıldığı dokümantasyon içerisinde anlaşılır olması gerekir. Api dokümantasyonunu el emeği ile yazmak hem zordur hemde güncel tutması imkansızdır. Bir biçimde bu dokümantasyonu güncel olarak üretmek gerekir.

Hem insanların hemde bilgisayarlara kaynak koda erişmeden RestApi lerin özelliklerini görmesine, incelemesine ve anlamasına olanak sağlayan Swagger<sup>7</sup> entegrasyonu belirli bir süreliğine açık tutulacaktır.<sup>8</sup>

<sup>7</sup> Swagger UI, oluşturduğumuz API'lar ile ilgili bilgileri görselleştirmemiz ve otomatik dokümantasyon oluşturabilmemize yarayan yardımcı bir araçtır.

<sup>8</sup> Bu süreç içerisinde <https://awqatsalah.diyamet.gov.tr/swagger/index.html> adresinden swaggere erişim sağlanabilir.

### 3. Kimliklendirme ve Güvenlik Yapısı

Bu servis JWT<sup>9</sup> (JSON Web Tokens) yapısını kullanarak kullanıcı doğrulama (kimliklendirme / yetkilendirme) işlemi yapmaktadır. Bu senaryoda sisteme kullanıcı adı ve şifre gönderilir, sistem başarılı bir kullanıcı girişinden sonra istemciye Access Token<sup>10</sup> ve Refresh Token<sup>11</sup> verir. Access Token'in süresi 30 dakika iken Refresh Token'in süresi +15 dakikadır. İstemcinin yapacağı her istekte bu Access Token ile gelmesi beklenir.

Örnek	Kullanım:	Authorization:	Bearer

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW11cy54bWxzbnR5Z2FwLm9yZy93cy8yMDA1LzA1L2lkZW50aXR5L2NsYWltcy9uYW1laWRlbnRpdHkiOiJ0MWF1OGQ0LTE1YWYtNGQxNy1hODIyLWVmNjYwMTRmMTY0OSIsImh0dHA6Ly9zY2hlbWZlLnhtbHNvYXAub3JnL3dzLzIwMDUvMDUvaWRlbnRpdHkvY2xhaW1zL2VtYWlsYWRkcmVzcyI6ImthbWlsLmtpcmRpdG9rpeWV4ZXQuZ292LnRyIiwiaHR0cDovL3NjaGVtYXMubWljcm9zb2Z0LmNvbS93cy8yMDA1LzA1L2lkZW50aXR5L2NsYWltcy9yb2x1IjoiaWFuY291LmRpeWV4ZXQuZ292LnRyIiwiaWF0IjoiYXV0aW11LmRpeWV4ZXQuZ292LnRyIn0.zaF92gi9VNvx7IVLpInscG0uH5SHq2BqR7UHI13MVAg
```

Bu senaryoda İstemcinin isteklerinin hızlı ve olması için, istemci tarafından alınan Access Token'in her istekten önce Expire Time (geçerlilik süresi)'ni kontrol etmesi istenir eğer süre geçmiş ve 15 dakikayı aşmamış ise Refresh Token mekanizmasını kullanması ideal olanıdır.

Bu token, istemciye ait Guid tipinde ve aynıysından ikinci kez rastgele üretilmesi mümkün olmayan bir değer içerir. Bu token gizli bir anahtar ile sunucu tarafında doğrulandığı için manipüle edilemez. Ayrıca istemcinin rolüne bağlı endpoint yetkileri her istek sırasında yeniden veri tabanından kontrol edildiği için, mevcut isteğini asla yetkisiz olduğu bir metoda yapamayacağından yetkisiz erişimin önüne geçilmiş olur. Yukarıda belirtildiği gibi geçerlilik süresi 30 dakika olarak ayarlanmıştır. Bu süre sonunda bu tokenle gelen isteklere cevap verilmez. Böylece de güvenlik sağlanmış olmaktadır.

### 4. Loglama (Günlük Verisi) Yapısı

Bu servis standart olarak her bir isteği loglar. Bu sayede çeşitli istatistiki bilgilerin yanı sıra sistem tarafında istemcilerin yapmış oldukları isteklerin detayları, karşılaştığı hatalar veya yetkisiz erişim istekleri takip edilebilmektedir.

Log tutucusu olarak, Windows ve Linux(Docker) üzerinde çalışabilen ve verileri çeşitli çizelgeler ile Şekil 4'teki gibi görselleştirebilen Seq<sup>12</sup> uygulaması tercih edilmiştir.

<sup>9</sup> JWT(JSON Web Tokens), bir RFC7519 endüstri standartıdır. JWT, kullanıcının doğrulanması, web servis güvenliği, bilgi güvenliği gibi birçok konuda kullanılabilir.

<sup>10</sup> Bir kaynağa ulaşmak için verilmiş belirteçtir.

<sup>11</sup> Refresh Token authorization server tarafından mevcutta olan Access Token'in expire süresi sona ermeye yaklaştığında veya sona erdiğinde, yeni bir Access Token elde edebilmek için client'a verilen tokendir.

<sup>12</sup> Seq, modern yapılandırılmış günlük verileri için özel olarak oluşturulmuş akıllı arama, analiz ve uyarı sunucusudur. <https://datalust.co/>



Şekil 4 Günlükleme Uygulaması Seq'e ait örnek bir dashbord

## 5. İstek Kotası

Bu servis yazılım geliştiricilerine yönelik olarak hazırlanmıştır. Ticari veya bireysel kullanım fark etmeksizin kullanımlar, rollere göre çeşitli sınırlandırmalara sahiptir.

Bu amaçla 2 tane rol tanımlanmıştır.

### 1. Developer Rolü

İlgili api servisini sistemine entegre edecek kullanıcılar için, bütün endpointleri, günlük 100 kez tüketebilecek şekilde hak tanıyan roldür. Bu röle sahip kullanıcılar belirli bir sürenin sonunda Standart Role taşınır.

### 2. Standart Rol

Bu röle sahip kullanıcılar tükettikleri her bir endpoint için, günlük, aylık veya yıllık periyotta değişen haklara sahiptir. Genel olarak günlük bütün endpointler (parametre istenen endpointlerin her biri parametresi ile birlikte) 5 istekle sınırlıdır. Bu nedenle ilgili servisi kullanan geliştiriciler mobil veya web uygulamaları için kendi imkanları ile bu verileri host ederek kullanabileceklerdir.

## 6. EndPoint Adres Listesi

### 6.1.Auth (Authorization İşlemleri)

#### 6.1.1. /api/Auth/Login

İstemcinin sisteme giriş yapmasını sağlar. Kullanıcı adı ve şifre doğru olduğunda Şekil 2'deki IResult nesnesinin içerisinde bir Token döndürür. Tokenin süresi 45 dakikadır.

Post metodu ile çağırımı destekler. Aşağıda şekildeki gibi bir model beklemektedir.

```
{  
  "email": "user@example.com",  
  "password": "string"  
}
```

Dönüş örneği aşağıdaki gibidir.

```
{  
  "data": {  
    "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ3Y29udHIiuyuyfQ.uuQMMzY10aZJa08BEE9u1Tg0-M",  
    "refreshToken": "4600ABDB8F396FC87FA4C81ABC6DE9087C46351D0AB6FD8AE7591FFC646647050AC2"  
  },  
  "success": true,  
  "message": null  
}
```

#### 6.1.2. /api/Auth/RefreshToken/{RefreshToken}

İstemciye login olunduktan sonra Acces Token ile birlikte Resfresh Token'da verilir. Resfresh Token'ın süresi Acces Token'in süresinin bitiminden 15 dakika sonrasıdır. İstemci isteği sırasından Acces Token'in bitiş süresini kontrol eder. Bu süre yaklaştı veya bitti ise (+15 dakika) bu endpointe Resfresh Token'ı ile istek yaparak yeniden Auth işlemine gerek kalmaksızın yeni bir tane Acces Token ile birlikte Resfresh Token alır.

GET metodu ile çağırımı destekler.

## 6.2.DailyContent (Günlük İçerik İşlemleri)

### 6.2.1. /api/DailyContent (Günlük Ayet Hadis Dua Servisi)

Yılın ilgili gününe ait Ayet, Hadis ve Dua bilgilerini istemciye döndürür.

Get metodu ile çağrılabilir. Herhangi bir parametreye ihtiyaç duymaz.

IResult nesnesinin içerisinde ki dönüş örneği aşağıdaki gibidir.

```
{
  "data": {
    "id": 333,
    "dayOfYear": 333,
    "verse": "\"Gökleri, yeri ve bu ikisi içinde yaydığı canlıları yaratması, O'nun varlığının delillerindendir. O, dilediği zaman, onları bir araya getirmeye de gücü yetendir.\" ",
    "verseSource": "(Şu'arâ, 42/29)",
    "hadith": "\"Küçüklerimize merhamet etmeyen, büyüklerimize saygı göstermeyen bizden değildir.\" ",
    "hadithSource": "(Tirmizî, \"Birr \", 15)",
    "pray": "\"Bizleri, insanlığın barış ve huzuru için hayrın anahtarı şerrin kilidi olanlardan eyle Allah'ım.\" ",
    "praySource": null
  },
  "success": true,
  "message": null
}
```

## 6.3.Place (Ülke / Eyalet / Şehir Bilgileri)

### 6.3.1. /api/Place/Countries (Ülke Listesi)

Serviste kaydı bulunan Ülke listesini döndürür. Get metodu ile çağrılabilir. Herhangi bir parametreye ihtiyaç duymaz.

Dönüş örneği aşağıdaki gibidir.

```
{
  "data": [
    {
      "id": 1,
      "code": "NORTH CYPRUS",
      "name": "KUZAY KIBRIS"
    },
    {
      "id": 2,
      "code": "TURKEY",
      "name": "TÜRKİYE"
    }
  ],
  "success": true,
  "message": null
}
```



### 6.3.2. /api/Place/States (Eyalet Listesi)

Serviste kaydı bulunan Eyalet (Türkiye için Şehir) listesini döndürür. Get metodu ile çağrılabilir. Herhangi bir parametreye ihtiyaç duymaz.

Dönüş örneği aşağıdaki gibidir.

```
{
  "data": [
    {
      "id": 500,
      "code": "ADANA",
      "name": "ADANA"
    },
    {
      "id": 854,
      "code": "BREMEN",
      "name": "BREMEN"
    }
  ],
  "success": true,
  "message": null
}
```

### 6.3.3. /api/Place/States/{countryId} (Ülkeye Göre Eyalet Bilgileri)

/api/Place/Countries servisinden dönen listedeki Ülkelerden seçilen ID değerine bağlı Eyalet (Türkiye için Şehir) bilgilerini döndürür. {countryId} yerinde bu bilgi gönderilmelidir. Get metodu ile çağrılır.

Dönüş değeri 6.3.2 başlığında ki dönüş değeri yapısı ile aynıdır.

#### 6.3.4. /api/Place/Cities (Şehir Listesi)

Serviste kaydı bulunan Şehir (Türkiye için İlçe) listesini döndürür. Get metodu ile çağrılabilir. Herhangi bir parametreye ihtiyaç duymaz.

Dönüş değeri 6.3.2 başlığında ki dönüş değeri yapısı ile aynıdır.

#### 6.3.5. /api/Place/Cities/{stateId} (Eyalete Göre Şehir Listesi)

/api/Place/States servisinde dönen listedeki Eyalete (Türkiye için Şehir) ait ID değerine bağlı Şehir (Türkiye için İlçe) listesini döndürür. Get metodu ile çağrılabilir. {stateId} yerinde bu bilgi gönderilmelidir

Dönüş değeri 6.3.2 başlığında ki dönüş değeri yapısı ile aynıdır.

#### 6.3.6. /api/Place/CityDetail/{cityId} (Şehir Detay Bilgisi)

/api/Place/Cities/{stateId} servisinde dönen listeden seçilen Şehre (Türkiye için İlçe) ait ID değeri ile Şehre (Türkiye için İlçe) ait Detay bilgisi döndürür. Get metodu ile çağrılabilir.

Dönüş örneği aşağıdaki gibidir.

```
{
  "data": {
    "id": "17885",
    "name": "DEVREKANİ",
    "code": null,
    "geographicQiblaAngle": "164",
    "distanceToKaaba": "2312",
    "qiblaAngle": "159",
    "city": "KASTAMONU",
    "cityEn": null,
    "country": "TÜRKİYE",
    "countryEn": "TÜRKİYE"
  },
  "success": true,
  "message": null
}
```

## 6.4. PrayerTime (Namaz Vakti Bilgileri)

6.4.1. `/api/PrayerTime/Daily/{cityId}` (Şehir'e Ait Günlük Namaz Vakitleri)  
`/api/Place/Cities/{stateId}` servisinden dönen listedeki İlçeye ait ID değerine bağlı günlük namaz vaktini döndürür. Get metodu ile çağrılabilir.

Dönüş örneği aşağıdaki gibidir.

```
{
  "data": [
    {
      "shapeMoonUrl": "http://namazvakti.diyabet.gov.tr/images/r5.gif",
      "fajr": "06:11",
      "sunrise": "07:42",
      "dhuhr": "12:38",
      "asr": "15:01",
      "maghrib": "17:23",
      "isha": "18:49",
      "astronomicalSunset": "17:16",
      "astronomicalSunrise": "07:49",
      "hijriDateShort": "5.5.1444",
      "hijriDateShortIso8601": null,
      "hijriDateLong": "5 Cemaziyevvel 1444",
      "hijriDateLongIso8601": null,
      "qiblaTime": "11:31",
      "gregorianDateShort": "29.11.2022",
      "gregorianDateShortIso8601": "29.11.2022",
      "gregorianDateLong": "29 Kasım 2022 Salı",
      "gregorianDateLongIso8601": "2022-11-29T00:00:00.0000000+03:00",
      "greenwichMeanTimeZone": 3
    }
  ],
  "success": true,
  "message": null
}
```

6.4.2. `/api/PrayerTime/Weekly/{cityId}` (Şehir'e Ait Haftalık Namaz Vakitleri)  
`/api/Place/Cities/{stateId}` servisinden dönen listedeki İlçeye ait ID değerine bağlı haftalık namaz vaktini döndürür. Get metodu ile çağrılabilir.

Dönüş değeri 6.4.1 başlığında ki dönüş değeri yapısı ile aynıdır.

6.4.3. `/api/PrayerTime/Monthly/{cityId}` (Şehir'e Ait Aylık Namaz Vakitleri)  
`/api/Place/Cities/{stateId}` servisinden dönen listedeki İlçeye ait ID değerine karşılık gelen yere ait aylık namaz vaktini döndürür. Get metodu ile çağrılabilir.

Dönüş değeri 6.4.1 başlığındaki dönüş değeri ile aynıdır.

**6.4.4. /api/PrayerTime/Eid/{cityId} (Şehir'e Ait Bayram Namazı Vakitleri)**  
/api/Place/Cities/{stateId} servisinden dönen listedeki İlçeye ait ID değerine bağlı bayram namaz vaktini döndürür. Get metodu ile çağrılabilir.

Dönüş örneği aşağıdaki gibidir.

```
{  
  "data": {  
    "eidAlAdhaHijri": "10 Zilhicce 1443",  
    "eidAlAdhaTime": "05:54:00",  
    "eidAlAdhaDate": "9 Temmuz 2022 Cumartesi",  
    "eidAlFitrHijri": "10 Zilhicce 1443",  
    "eidAlFitrTime": "05:54:00",  
    "eidAlFitrDate": "9 Temmuz 2022 Cumartesi"  
  },  
  "success": true,  
  "message": null  
}
```

**6.4.5. /api/PrayerTime/Ramadan/{cityId} (Şehir'e Ait Ramazan İmsakiyesi)**  
/api/Place/Cities/{stateId} servisinden dönen listedeki İlçeye ait ID değerine karşılık gelen yere ait Ramazan İmsakiyesini döndürür. Get metodu ile çağrılabilir.

Dönüş değeri 6.4.1 başlığında ki dönüş değeri yapısı ile aynıdır.