Patika Dev

Akbank .Net Bootcamp
Final Case Projesi

Masraf Ödeme Sistemi

Proje Raporu

Ali İlman

Giriş

Bootcamp eğitimi sonunda Final Case ödevi olarak verilen “Masraf Ödeme Sistemi“ kapsamında oluşturulmuştur.

Projeye ait dokümantasyon, çalışır halinin ekran görüntüleri ve Loom platformunda benim projeyi anlattığım video mevcuttur.

Postman üzerinden yapılan Api Dokümantasyon :
https://documenter.getpostman.com/view/32455003/2s9YsT68u6

RabbitMQ arayüzü:
https://fish.rmq.cloudamqp.com/#/queues/ocdvcuen/PaymentQueue
"UserName": "ocdvcuen",
"Password": "0Qh_eUzUCDoG-iHudqT8P4NGOAf6P2vf"

Kullanılan Teknolojiler ve Geliştirme Ortamı

Uygulama Visual Studio Code ortamına geliştirilmiştir.
.Net 8 kullanılarak geliştirilmiştir
Veritabanı için MS SQL kullanılmıştır.
Test ve Api yönetimi için Swagger kullanılmıştır.

Kütüphaneler

Veritabanı işlemleri:
"Microsoft.EntityFrameworkCore" Version="8.0.0"
"Microsoft.EntityFrameworkCore.Design" Version="8.0.0"
"Microsoft.EntityFrameworkCore.SqlServer" Version="8.0.0"
"Microsoft.EntityFrameworkCore.Tools" Version="8.0.0"
"Dapper" Version="2.1.28" (sorgular için)

Token:
"Microsoft.AspNetCore.Authentication.JwtBearer" Version="8.0.0

Json dönüşümleri:
"Newtonsoft.Json" Version="13.0.3"

Kuyruk yapısı ile serviler arası iletişim:
"RabbitMQ.Client" Version="6.8.1"

Loglama:
"Serilog.AspNetCore" Version="8.0.0"
"Serilog.AspNetCore" Version="8.0.0"
"Serilog.Settings.Configuration" Version="8.0.0"
"Serilog.Sinks.Console" Version="5.0.1"

"Serilog.Sinks.File" Version="5.0.0"

Model doğrulaması:
"FluentValidation" Version="11.9.0" />
"FluentValidation.AspNetCore" Version="11.3.0" />
"FluentValidation.DependencyInjectionExtensions" Version="11.9.0" />

Mapping:
"AutoMapper" Version="12.0.1"

CQRS pattren için:
"MediatR" Version="12.2.0"

Mimari Tasarım

Proje kapsamında bootcamp sürecinde kullandığımız mimari ile devam ettim. CQRS design pattern kullanarak ilerledim. Modeler için Command ve Query'ler oluşturarak projemdeki işlevleri yönettim.

Projeyi beş parçaya böldüm:

MOS.Api: Wep Api projesi şablonu kullanarak geliştirilen içerisinde Apilere ait controllerin barındığı katman

MOS.Base: Class Library projesi olarak oluşturulan temel sınıfların konumlandırıldığı proje.

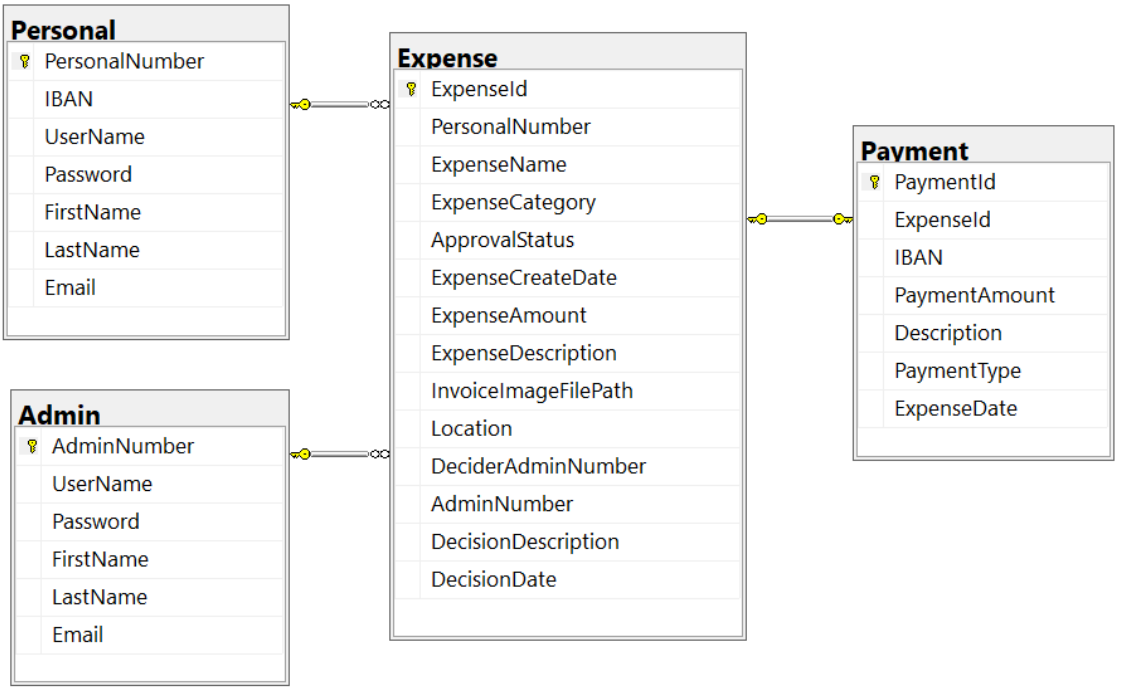
MOS.Business: Cqrs, Command, Query, Mapper, Validator ve Servisleri içeren projedeki ana hesaplamaların yapıldığı katman.

MOS.Data: Veritabanına ait Entity , Migration ve gerekli bağlantı komutlarını içeren projedir.

MOS.Schema: Kullanıcıdan gelen isteklerde ve gidecek cevaplarda kullanılacak sınıfları içeren katmandır.

Veri Modeli

Projeme ait veritabanı tasarımı aşağıdaki gibidir.



Admin ve Personal

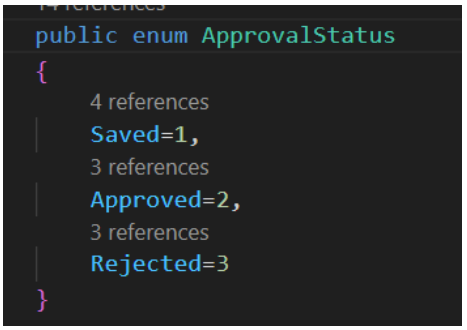
Projede bizden iki farklı çalışan rolünden bahsedilmiş. Bu kapsamda veritabanını oluştururken buna göre hareket etmek istedim. Tek bir çalışanlar tablosu yerine “Admin” ve “Personal” adında 2 farklı tablo oluşturarak bu kişileri burada tuttum. Buradaki temel düşünce her ne kadar bu iki çalışan tipi temelde aynı özelliklere sahip olsa da farklı görev tanımları var. Belki ileride “Personal” tipindeki çalışanlar için şirket dışındaki çalışmalarını tutan ayrı bir istek eklenecek bu tarz istekler “Admin” röleden ayrı olduğu için böyle bir sistem seçtim.
Bu projeyi hali hazırda şirketin sistemine entegre olacak şekilde değil de kendine ait veritabanı olan istendiğinde şirketin diğer servisleri ile iletişime geçebilecek bir tasarım yaptım. (Bildirim ve Ödeme servisleri gibi)

Expense

Personal tarafından yapılan harcamaların tutulduğu tablodur. Buradaki en kritik nokta tüm oluşturulan , onaylanan ve reddedilen kayıtların bir arada tutuluyor olmasıdır.

Personal bir Expense oluştururken kendine ait doldurması gereken alanları girerek kayıt oluşturur. Bundan sonraki süreçte Admin ilgili veriyi çeker ve onay, red durumunu belirler. Admin gerekli bilgileri girer ve süreci tamamlar.

Expense tablosu Personal ile ilişkilidir (1:N) . Bir Personal birden fazla Expence üretebilir. Bir Expence bir Personal tarafından oluşturulabilir.
Expense tablosu Admin ile ilişkilidir (1:N) . Bir Admin birden fazla Expence kararı verebilir. Bir Expence bir Admin tarafından karar atanabilir.



“ApprovalStatus” enum veri yapısı burada Expense’nin durumunu tutar. Karar durumlarında değer set edilir.

Aynı satır üzerindeki verilerde farklı kişilerin işlemler yapıyo olması sonucunda oluşacak sorunları gidermek adına sistem validasyonlar ile korunmaktadır. Her rol sadece kendi ile ilgi sütunlarda değişiklik yapabilmektedir.

“InvoiceImageFilePath” sisteme başka bir api üzerinden yüklenen yüklenebilen dosya (resim pdf vb) yolunu tutar. Bu süreç şimdilik farklı bir Api üzerinden işlemektedir. İstenildiği takdirde CrateExpence Api’sine eklenebilir.

Expense oluşturma silme değiştirme yetkisi sadece Personal e aittir. Personal sadece kendi Expense’lerine erişebilir.Karar verilmiş harcama üzerinde Personal tarafından silme değiştirme işlemleri yapılamaz.

Admin karar verme işlemi yapabilir. Admin tüm expenseleri görüntüleyebilir.

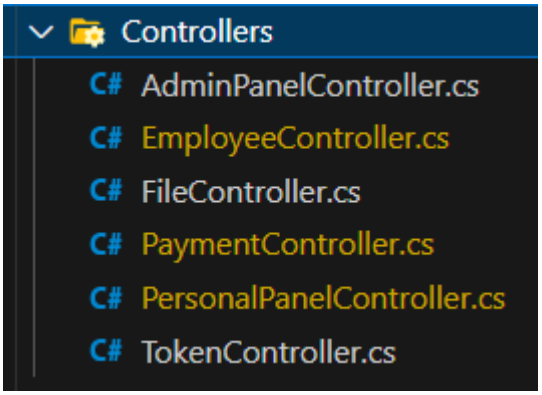
Payment

Onaylanan her ödeme için bir Personal’e bir ödeme emri tanımlanır.
Her Expense için bir Payment ve her Payment için bir Expense olabilir.

Web Api

Postman üzerinden yapılan Api Dokümantasyon :
https://documenter.getpostman.com/view/32455003/2s9YsT68u6

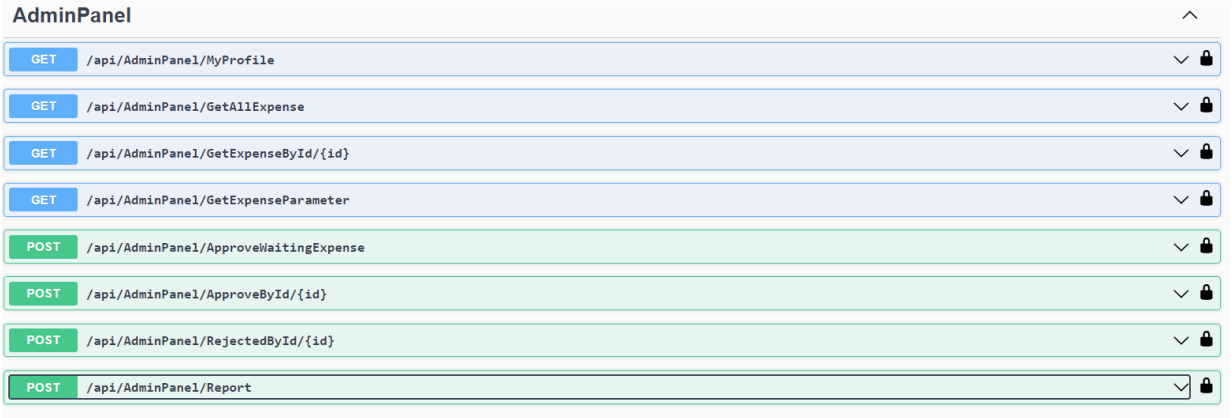
Projede farklı Controller ile Api’ler gruplandırılmıştır.
Rollere göre iki farklı Controller eklenmiştir. Bizden istenen ana senaryonun işlemleri için bu apiler üzerinden işlem yapılacaktır.



Admin Panel

JWT Authorization ile yalnızca Admin rolündeki kullanıcılar bunu kullanabilmektedir.
Expense işlemleri yapmak için hazırlanmış Api'lerden Admin için olanıdır.
Masraf Onaylama sürecinde Admine ait işlevler burada tanımlanmıştır.
Expense için Get / GetByid/ GetByParameter apiler ve bekleyenleri listeleme, Onaylama ve Reddetme işlemleri buradan yapılmaktadır.

Raporlama api de burada yer almaktadır.



Request URL

```
http://localhost:5203/api/AdminPanel/Report
```

Server response

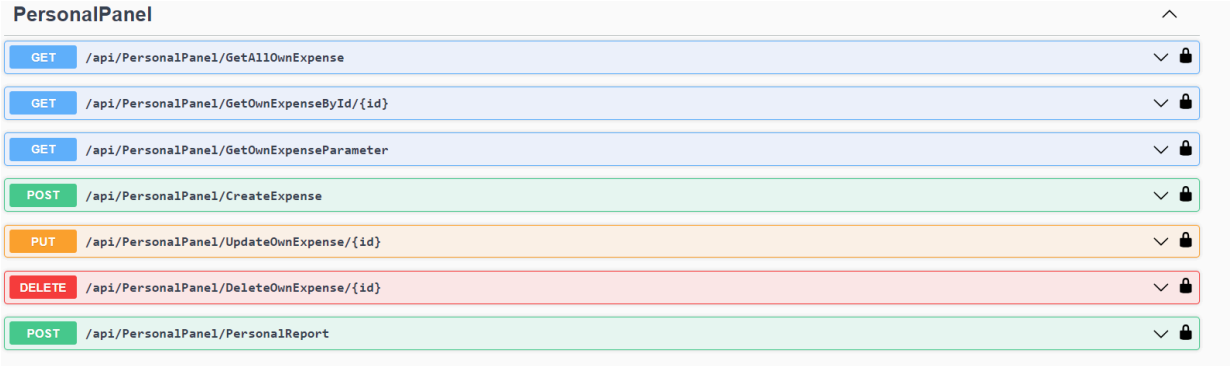
Code	Details
200	Response body


```
{
  "serverDate": "2024-01-19T23:55:02.6976499Z",
  "referenceNo": "b67747a2-2d1e-42f2-8fa6-12f225592fd5",
  "success": true,
  "message": "Success",
  "response": {
    "raporName": "Report__2023-01-01__2024-01-19__Activity_Report",
    "reportCrateDate": "2024-01-20T02:55:02.6974401+03:00",
    "startTheDate": "2023-01-01T00:00:00",
    "endTheDate": "2024-01-19T23:54:16.039Z",
    "totalMoneySpentByPersonals": 28103,
    "totalApprovedSpentMoney": 3653,
    "totalRejectedSpentMoney": 5750,
    "reportEachPersonalList": [
      {
        "personalResponse": {
          "iban": "12345678981234456798",
          "personalNumber": 1,
          "userName": "ferdikadi",
          "firstName": "Ferdi",
          "lastName": "Kadi",
          "email": "ferdi.kadi@akbank.com"
        },
        "moneySpentByPersonal": 8200,
        "approvedSpentMoney": 2153,
        "rejectedSpentMoney": 3000,
        "waitingExpenseList": [
          {
            "expenseId": 7
```


PersonalPanel

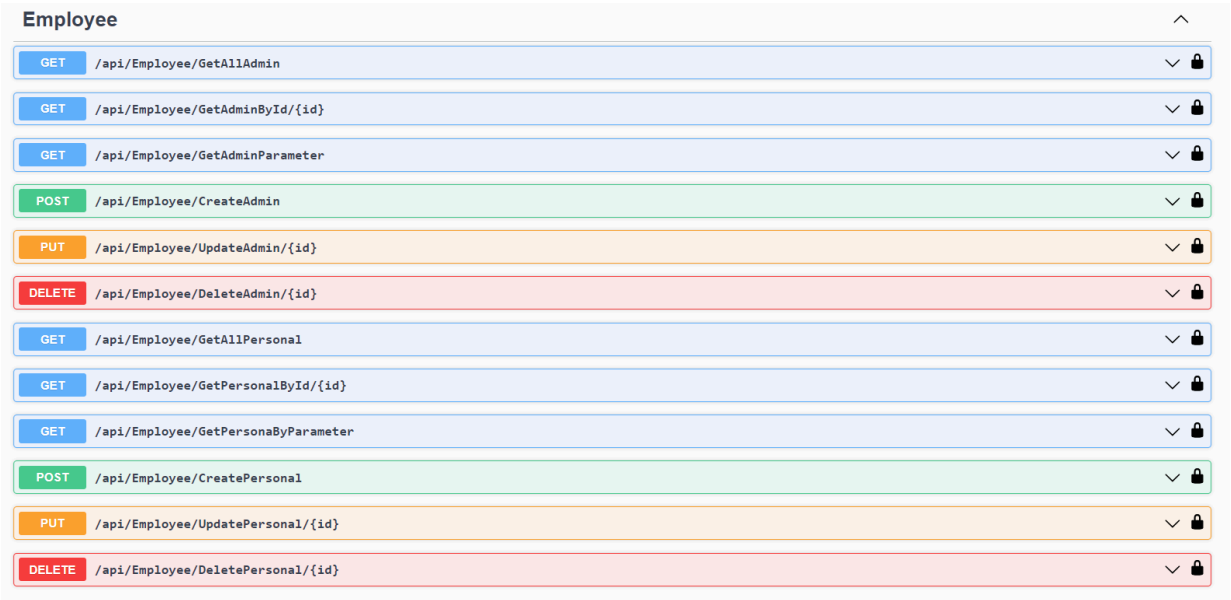
JWT Authorization ile yalnızca Personal rolündeki kullanıcılar bunu kullanabilmektedir.
Expense işlemleri yapmak için hazırlanmış Api'lerden Personaller için olanıdır.
Personaller yalnızca kendi Expense'leri üzerinde işlem yapabilmektedirler.

Personal kendine ait harcamaları raporlayabilir.



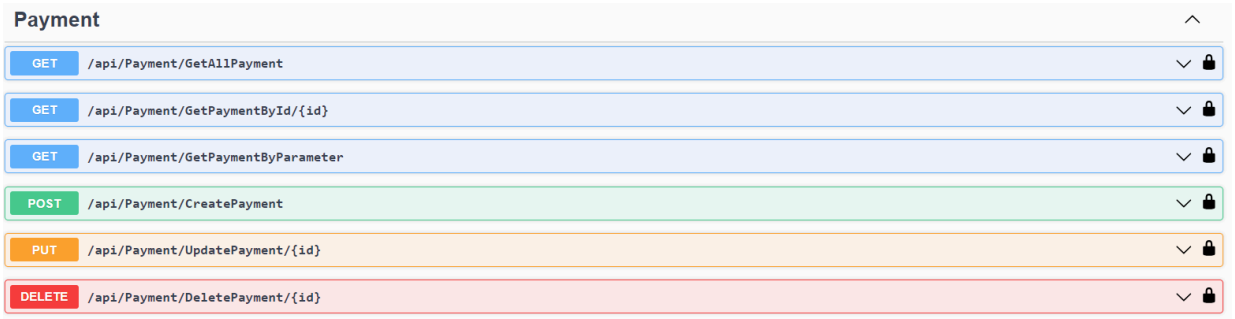
Employee

JWT Authorization ile yalnızca Admin rolündeki kullanıcılar bunu kullanabilmektedir.
Admin ve Personal rolleri için Command ve Query İşlemlerinin yürütüldüğü yerdir.



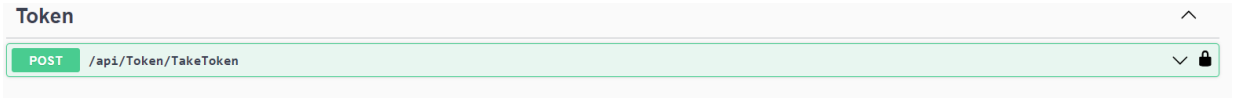
Payment

JWT Authorization ile yalnızca Admin rolündeki kullanıcılar bunu kullanabilmektedir.
Payment için Command ve Query İşlemlerinin yürütüldüğü yerdir



Token

Sisteme login olmak ve token almak için kullanılan apidir.

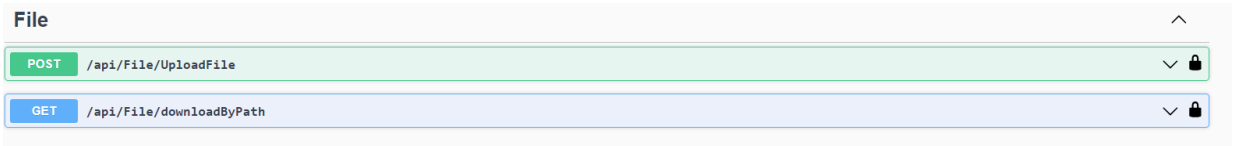


File

JWT Authorization ile rol farketmeden kullanılabilmektedir

.Sisteme yüklenmesi gereken dosyalar önce buradan yüklenir geri dönüş değeri olarak dosya yolu alınır. Expence crate işlemi sırasında yol databaseye kaydedilir.

İstenildiği zaman dosya yolu ile dosya sistemden indirilebilir.



Harici Servis Entegrasyonları

Proje kapsamında harici servis olarak sadece RabbitMQ kullanılmıştır.

RabbitMQ

Popüler bir kuyruk servisi olan RabbitMQ, projede servisler arası iletişimi sağlamak amacıyla konumlandırılmıştır.
Bu kapsamda iki farklı kuyruk tanımlanmıştır. Bu kuyruklar projenin ana isteri olan masraf onaylama işlemleri için değil onay ve red sonrası için oluşan senaryolar için düşünülmektedir.

“NotificationQueue” ve “PaymentQueue” kuyrukları kullanılmaktadır.
NotificationQueue : Sistemde hali hazırda var olduğu varsayılan anlık bildirim servisine veya basit bir smtp servisiyle iletişime geçilmesi için düşünülmüştür. Bildirim servisi proje kapsamında olmadığı için eklenmemiştir.

Personal'in eklediği Expense hakkında karar verildiği durumlarda bildirim nesnesi oluşturulur json şekline çevrilir ve kuyruğa alınır.

PaymentQueue: Şirket kapsamına hali hazırda var olduğu varsayılan muhasebe veya maaş ödeme emirlerini oluşturan başka ile iletişime geçecek olan servisle haberleşmesi düşünülerek tasarlanmıştır. Burada Expense Onaylandığı zaman oluşturulan payment id kuyruğa eklenir ve ilgili servise iletilir.(İlgili servisin bizim veritabanına erişimi olduğunu varsayıyoruz. Aksi takdirde Payment nesnesini gönderirdik).

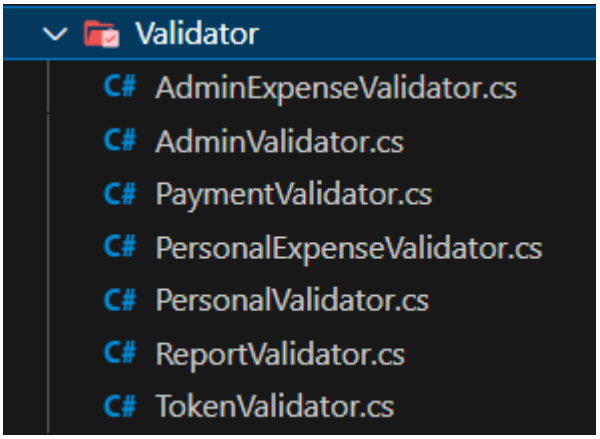
RabbitMQ arayüzü:
https://fish.rmq.cloudamqp.com/#/queues/ocdvcuen/PaymentQueue
"UserName": "ocdvcuen",
"Password": "0Qh_eUzUCDoG-iHudqT8P4NGOAf6P2vf"

Rapor

Projeki bir diğer kapsamlı ister raporlama. Proje kapsamında rapor için Dapper kullanımı istenmekte. Burada rapor oluşturmak için gerekli sorguları yazarken bu kullanılmıştır.
Rapor için Admin kullanıcıdan rapor başlangıç tarihi, rapor bitiş tarihi ve istenilen personal numalarının bir listesini alıyoruz.
Değerlerin null gelmesi durumunda ilgili alan her şeyi kapsayacak şekilde set edilmektedir. (Liste null ise tüm Personal'ler getirilir)

Dönüş değeri olarak toplam harcama, onaylanan harcama, reddedilen harcama gibi değerleri ve personal bazlı harcamalar, toplam harcama, onaylanan, reddedilen harcama değerlerini içerek şekilde verilmektedir.
Rapor oluşturmak için verileri json formatta geri dönmekdeyim.
Ek kısmında örnek verilmiştir.Görmek için tıklayınız.

Validasyonlar



Proje kapsamında Fluent validasyon kullanılmıştır.

Kullanıcı tarafından gelen bütün istekler validasyondan geçmektedir.

Bu kısımda bahsetmem gereken ana husus şudur. Conrtoller tarafında işlem yapılmadan önce kontrol edilmektedir.
Bu kısımda aslında manuel kontrol değil otomatik olarak kontrol etmek için çalışma

yapılmıştır. Bu kısımdaki kodlarda oluşan tam çalışmama sorunu nedeniyle manuel yollarla halledilmiştir.

```
[HttpPost("CreateExpense")]
0 references
public async Task<ApiResponse<ExpenseResponse>> CreateExpense([FromBody] PersonalExpenseRequest Expense)
{
    PersonalExpenseValidator validations = new();
    validations.ValidateAndThrow(Expense);

    string PersonalNumber = (User.Identity as ClaimsIdentity).FindFirst("Id")?.Value;
    var operation = new CreateExpenseCommand(PersonalNumber: int.Parse(PersonalNumber), Expense);
    var result = await mediator.Send(operation);
    return result;
}
```

```
5 references
public class AdminExpenseValidator : AbstractValidator<AdminExpenseRequest>
{
    2 references
    public AdminExpenseValidator()
    {
        RuleFor(x => x.DecisionDescription)
            .NotEmpty()
            .MaximumLength(255).WithMessage("DecisionDescription can be up to 255 characters");
    }
}
```

```
public class AdminValidator: AbstractValidator<AdminRequest>
{
    0 references
    public AdminValidator()
    {
        RuleFor(x => x.UserName)
            .NotEmpty()
            .MinimumLength(3).WithMessage("UserName can be at least 3 characters")
            .MaximumLength(50).WithMessage("UserName can be a maximum of 50 characters")
            .Matches("^[a-zA-Z0-9]*$").WithMessage("UserName must contain only letters and numbers.");

        RuleFor(x => x.Password)
            .NotEmpty()
            .MinimumLength(6).WithMessage("Password can be at least 6 characters");

        RuleFor(x => x.FirstName)
            .NotEmpty()
            .MaximumLength(50).WithMessage("FirstName can be a maximum of 50 characters");

        RuleFor(x => x.LastName)
            .NotEmpty()
            .MaximumLength(50).WithMessage("Surname can be a maximum of 50 characters");

        RuleFor(x => x.Email)
            .NotEmpty()
            .EmailAddress().WithMessage("Please enter a valid e-mail address.")
            .MaximumLength(100).WithMessage("Email address can be up to 100 characters");
    }
}
```

```
5 references
public class PaymentValidator : AbstractValidator<PaymentRequest>
{
    2 references
    public PaymentValidator()
    {
        RuleFor(x => x.IBAN)
            .NotEmpty()
            .Length(26).WithMessage("IBAN must be 26 characters");

        RuleFor(x => x.PaymentAmount)
            .GreaterThan(0).WithMessage("PaymentAmount must be greater than 0");

        RuleFor(x => x.Description)
            .NotEmpty()
            .MaximumLength(200).WithMessage("Description must be at most 200 characters.");

        RuleFor(x => x.PaymentType)
            .NotEmpty()
            .MaximumLength(30).WithMessage("PaymentType must be at most 30 characters.");

        RuleFor(x => x.ExpenseDate)
            .NotEmpty();

    }
```

```
5 references
public class PersonalExpenseValidator : AbstractValidator<PersonalExpenseRequest>
{
    2 references
    public PersonalExpenseValidator()
    {
        RuleFor(x => x.ExpenseName)
            .NotEmpty()
            .MaximumLength(50).WithMessage("ExpenseName can be up to 50 characters");

        RuleFor(x => x.ExpenseCategory)
            .NotEmpty()
            .MaximumLength(50).WithMessage("ExpenseCategory name can be up to 50 characters");

        RuleFor(x => x.ExpenseAmount)
            .GreaterThan(0).WithMessage("ExpenseAmount must be greater than 0.");

        RuleFor(x => x.ExpenseDescription)
            .NotEmpty()
            .MaximumLength(100).WithMessage("ExpenseDescription can be up to 100 characters");

        RuleFor(x => x.InvoiceImageFilePath)
            .NotEmpty()
            .MaximumLength(150).WithMessage("InvoiceImageFilePath can be up to 150 characters");

        RuleFor(x => x.Location)
            .MaximumLength(150).WithMessage("Location can be up to 150 characters");
    }
```

```
public class PersonalValidator : AbstractValidator<PersonalRequest>
{
    0 references
    public PersonalValidator()
    {
        RuleFor(x => x.UserName)
            .NotEmpty()
            .MinimumLength(3).WithMessage("UserName can be at least 3 characters")
            .MaximumLength(50).WithMessage("UserName can be a maximum of 50 characters")
            .Matches("^[a-zA-Z0-9]*$").WithMessage("UserName must contain only letters and numbers.");

        RuleFor(x => x.Password)
            .NotEmpty()
            .MinimumLength(6).WithMessage("Password can be at least 6 characters");

        RuleFor(x => x.FirstName)
            .NotEmpty()
            .MaximumLength(50).WithMessage("FirstName can be a maximum of 50 characters");

        RuleFor(x => x.LastName)
            .NotEmpty()
            .MaximumLength(50).WithMessage("Surname can be a maximum of 50 characters");

        RuleFor(x => x.Email)
            .NotEmpty()
            .EmailAddress().WithMessage("Please enter a valid e-mail address.")
            .MaximumLength(100).WithMessage("Email address can be up to 100 characters");

        RuleFor(x => x.IBAN).NotEmpty().Length(26).WithMessage("IBAN must be 26 characters.");
```

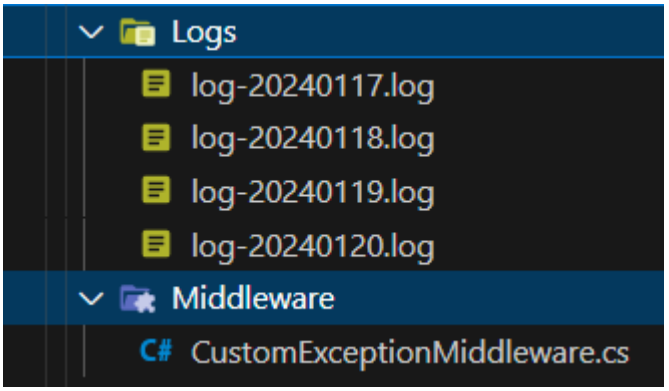
```
public class ReportValidator : AbstractValidator<ReportRequest>
{
    1 reference
    public ReportValidator()
    {
        RuleFor(x => x)
        .Must(x => Datetimevalidate(x.StartExpenceDate,x.EndExpenceDate))
        .WithMessage(" The StartExpenceDate must be less than the enddate - The StartExpenceDate must be less than the D
        //içeriklerin boş geliği durumlar için senaryolar geliştirilmiştir. O yüzden konraol edilmemiştir.
    }
    1 reference
    private bool Datetimevalidate(DateTime? StartExpenceDate, DateTime? EndExpenceDate)
    {
        //ikisin de null oluğu durum için işlem yapılabilir durumdadır.
        //ikisinin de null olamdığı durumlarda bu koşullar sağlanmalıdır
        if (!(StartExpenceDate == null && EndExpenceDate == null) &&
         (StartExpenceDate > EndExpenceDate ||
          StartExpenceDate > DateTime.Now.AddMinutes(1) ||
          EndExpenceDate > DateTime.Now.AddMinutes(1)))
        {
            return false;
        }
        return true;
    }
}
```

```
3 references
public class ReportRequestForOnePersonalValidator : AbstractValidator<ReportRequestForOnePersonal>
{
    1 reference
    public ReportRequestForOnePersonalValidator()
    {
        RuleFor(x => x)
        .Must(x => Datetimevalidate(x.StartExpenceDate,x.EndExpenceDate))
        .WithMessage(" The StartExpenceDate must be less than the enddate - The StartExpenceDate must be less than t
    }
    1 reference
    private bool Datetimevalidate(DateTime? StartExpenceDate, DateTime? EndExpenceDate)
    {
        //ikisin de null oluğu durum için işlem yapılabilir durumdadır.
        //ikisinin de null olamdığı durumlarda bu koşullar sağlanmalıdır
        if (!(StartExpenceDate == null && EndExpenceDate == null) &&
         (StartExpenceDate > EndExpenceDate ||
          StartExpenceDate > DateTime.Now.AddMinutes(1) ||
          EndExpenceDate > DateTime.Now.AddMinutes(1)))
        {
            return false;
        }
        return true;
    }
}
```

```
3 references
public class TokenValidator : AbstractValidator<TokenRequest>
{
    1 reference
    public TokenValidator()
    {
        RuleFor(x => x.UserName).NotEmpty().WithMessage("Username is mus be not null!!")
        .MinimumLength(5).MaximumLength(50);
        RuleFor(x => x.Password).NotEmpty().WithMessage("Username is mus be not null!!")
        .MinimumLength(5).MaximumLength(50);
    }
}
```

Global Exception Handler Middleware ve Logging

Projede eklenen Middleware kapsamında hata ayıklama yapılmaktadır. Gelen istekler loglanarak kaydedilmektedir. Log dosyaları günlük olarak tutulup otomatik olarak oluşturulmaktadır.



Ek:

Rapor için örnek json verisi:


```
{
  "serverDate": "2024-01-19T23:55:02.6976499Z",
  "referenceNo": "b67747a2-2d1e-42f2-8fa6-12f225592fd5",
  "success": true,
  "message": "Success",
  "response": {
    "raporName": "Report__2023-01-01__2024-01-19__Activity_Report",
    "reportCrateDate": "2024-01-20T02:55:02.6974401+03:00",
    "startTheDate": "2023-01-01T00:00:00",
    "endTheDate": "2024-01-19T23:54:16.039Z",
    "totalMoneySpentByPersonals": 28103,
    "totalApprovedSpentMoney": 3653,
    "totalRejectedSpentMoney": 5750,
    "reportEachPersonalList": [
      {
        "personalResponse": {
          "iban": "12345678981234456798",
          "personalNumber": 1,
          "userName": "ferdikadi",
          "firstName": "Ferdi",
          "lastName": "Kadi",
          "email": "ferdi.kadi@akbank.com"
```

```
      },
      "moneySpentByPersonal": 8200,
      "approvedSpentMoney": 2153,
      "rejectedSpentMoney": 3000,
      "waitingExpenseList": [
       {
        "expenseId": 7,
        "personalNumber": 1,
        "approvalStatus": 1,
        "expenseCreateDate": "2023-01-17T15:19:53.3943908",
        "expenseAmount": 1500,
        "expenseDescription": "A otelde konaklama ücreti",
        "invoiceImageFilePath":
"/uploads/799f7b3d-7b46-4728-a756-3445f0ff97d1_fatura-av-asilcan-tuzcu.jpg",
        "location": "esenyurt",
        "decisionDescription": null,
        "decisionDate": null,
        "payment": null
       },
       {
        "expenseId": 9,
        "personalNumber": 1,
        "approvalStatus": 1,
        "expenseCreateDate": "2023-03-17T15:51:33.1015422",
        "expenseAmount": 5000,
        "expenseDescription": "Temizlik malzemesi temini",
        "invoiceImageFilePath":
"/uploads/799f7b3d-7b46-4728-a756-3445f0ff97d1_fatura-av-asilcan-tuzcu.jpg",
        "location": "bornova",
        "decisionDescription": null,
        "decisionDate": null,
        "payment": null
       },
       {
        "expenseId": 11,
        "personalNumber": 1,
        "approvalStatus": 1,
        "expenseCreateDate": "2023-05-17T15:57:51.6795154",
        "expenseAmount": 600,
        "expenseDescription": "A otelde konaklama ücreti",
        "invoiceImageFilePath":
"/uploads/88ff7d06-1be3-45d2-996d-6067c4938e67_300_Akbank_Odev-1_2241a85fb3444bc
4bfafafd7def020fd.pdf",
        "location": "ankara",
        "decisionDescription": null,
        "decisionDate": null,
        "payment": null
       },
```

```
        {
          "expenseId": 14,
          "personalNumber": 1,
          "approvalStatus": 1,
          "expenseCreateDate": "2023-10-17T16:18:57.8016182",
          "expenseAmount": 200,
          "expenseDescription": "öğlen yemeği",
          "invoiceImageFilePath":
"/uploads/88ff7d06-1be3-45d2-996d-6067c4938e67_300_Akbank_Odev-1_2241a85fb3444bc
4bfafafd7def020fd.pdf",
          "location": "eskişehir",
          "decisionDescription": null,
          "decisionDate": null,
          "payment": null
        },
        {
          "expenseId": 16,
          "personalNumber": 1,
          "approvalStatus": 1,
          "expenseCreateDate": "2023-12-17T16:35:52.2276067",
          "expenseAmount": 900,
          "expenseDescription": "öğlen yemeği",
          "invoiceImageFilePath":
"/uploads/88ff7d06-1be3-45d2-996d-6067c4938e67_300_Akbank_Odev-1_2241a85fb3444bc
4bfafafd7def020fd.pdf",
          "location": "samsun",
          "decisionDescription": null,
          "decisionDate": null,
          "payment": null
        }
      ],
      "aproveedExpenseList": [
        {
          "expenseId": 1,
          "personalNumber": 1,
          "approvalStatus": 2,
          "expenseCreateDate": "2024-01-16T17:48:49.0945033",
          "expenseAmount": 2000,
          "expenseDescription": "Muğlaya giderken yakıt aldım",
          "invoiceImageFilePath":
"/uploads/799f7b3d-7b46-4728-a756-3445f0ff97d1_fatura-av-asilcan-tuzcu.jpg",
          "location": "Gulf petrol",
          "decisionDescription": "Harcamanız onaylanmış ve ödeme emri tanımlanmıştır. Onay
açılaması: onayladım",
          "decisionDate": "2024-01-16T17:58:31.7745024",
          "payment": null
        },
        {
```

```
          "expenseId": 21,
          "personalNumber": 1,
          "approvalStatus": 2,
          "expenseCreateDate": "2024-01-18T22:57:12.0307021",
          "expenseAmount": 153,
          "expenseDescription": "Akşam yemepi ücreti",
          "invoiceImageFilePath": "filepath",
          "location": "aile evi",
          "decisionDescription": "Harcamanız onaylanmış ve ödeme emri tanımlanmıştır. Onay
açılaması: Yemek harcamanız onaylanmıştır",
          "decisionDate": "2024-01-18T22:59:23.514863",
          "payment": null
        }
      ],
      "paymentList": [
        {
          "paymentId": 1,
          "expenseId": 1,
          "expense": null,
          "iban": "12345678981234456798",
          "paymentAmount": 2000,
          "description": "1 nolu şirket harcamanızın ücreti. Onay açıklaması:onayladım",
          "paymentType": "EFT",
          "expenseDate": "2024-01-16T17:58:31.774806"
        },
        {
          "paymentId": 4,
          "expenseId": 21,
          "expense": null,
          "iban": "12345678981234456798258763",
          "paymentAmount": 153,
          "description": "21 nolu şirket harcamanızın ücreti. Onay açıklaması:Yemek harcamanız
onaylanmıştır",
          "paymentType": "EFT",
          "expenseDate": "2024-01-18T22:59:26.4785422"
        }
      ],
      "rejecetExpenseList": [
        {
          "expenseId": 4,
          "personalNumber": 1,
          "approvalStatus": 3,
          "expenseCreateDate": "2024-01-17T15:14:36.4007713",
          "expenseAmount": 1500,
          "expenseDescription": "öğlen yemeği",
          "invoiceImageFilePath":
"/uploads/799f7b3d-7b46-4728-a756-3445f0ff97d1_fatura-av-asilcan-tuzcu.jpg",
          "location": "izmir",
```

```
          "decisionDescription": "Harcamanız Reddedilmiştir. Açıkama: Öğlen yemeği için 1500 lira harcama keyfidir",
          "decisionDate": "2024-01-18T17:04:42.6825662",
          "payment": null
        },
        {
          "expenseId": 6,
          "personalNumber": 1,
          "approvalStatus": 3,
          "expenseCreateDate": "2024-01-17T15:18:10.8058693",
          "expenseAmount": 1500,
          "expenseDescription": "öğlen yemeği",
          "invoiceImageFilePath": "/uploads/799f7b3d-7b46-4728-a756-3445f0ff97d1_fatura-av-asilcan-tuzcu.jpg",
          "location": "beşiktaş",
          "decisionDescription": "Harcamanız Reddedilmiştir. Açıkama: 1500 lira öğlen yemeği keyfi bir harcamadır",
          "decisionDate": "2024-01-18T17:11:04.1475092",
          "payment": null
        }
      ]
    },
    {
      "personalResponse": {
        "iban": "56412345678981234456798",
        "personalNumber": 2,
        "userName": "ardagul",
        "firstName": "Arda",
        "lastName": "Gul",
        "email": "Arda.gul@akbank.com"
      },
      "moneySpentByPersonal": 1700,
      "approvedSpentMoney": 0,
      "rejectedSpentMoney": 2750,
      "waitingExpenseList": [
        {
          "expenseId": 8,
          "personalNumber": 2,
          "approvalStatus": 1,
          "expenseCreateDate": "2023-02-17T15:20:58.6071293",
          "expenseAmount": 550,
          "expenseDescription": "öğlen yemeği",
          "invoiceImageFilePath": "/uploads/799f7b3d-7b46-4728-a756-3445f0ff97d1_fatura-av-asilcan-tuzcu.jpg",
          "location": "eminönü",
          "decisionDescription": null,
          "decisionDate": null,
          "payment": null
```

```
        },
        {
          "expenseId": 13,
          "personalNumber": 2,
          "approvalStatus": 1,
          "expenseCreateDate": "2023-07-17T16:09:18.1391083",
          "expenseAmount": 300,
          "expenseDescription": "Market malzemesi temini",
          "invoiceImageFilePath":
"/uploads/88ff7d06-1be3-45d2-996d-6067c4938e67_300_Akbank_Odev-1_2241a85fb3444bc
4bfafafd7def020fd.pdf",
          "location": "aydın",
          "decisionDescription": null,
          "decisionDate": null,
          "payment": null
        },
        {
          "expenseId": 19,
          "personalNumber": 2,
          "approvalStatus": 1,
          "expenseCreateDate": "2024-01-10T17:20:23.7100974",
          "expenseAmount": 850,
          "expenseDescription": "A otelde konaklama ücreti",
          "invoiceImageFilePath":
"/uploads/88ff7d06-1be3-45d2-996d-6067c4938e67_300_Akbank_Odev-1_2241a85fb3444bc
4bfafafd7def020fd.pdf",
          "location": "buca",
          "decisionDescription": null,
          "decisionDate": null,
          "payment": null
        }
      ],
      "aproveedExpenseList": [],
      "paymentList": [],
      "rejecetExpenseList": [
        {
          "expenseId": 2,
          "personalNumber": 2,
          "approvalStatus": 3,
          "expenseCreateDate": "2024-01-16T17:53:01.7329286",
          "expenseAmount": 1000,
          "expenseDescription": "Muğlaya giderken yakıt aldım",
          "invoiceImageFilePath":
"/uploads/799f7b3d-7b46-4728-a756-3445f0ff97d1_fatura-av-asilcan-tuzcu.jpg",
          "location": "Gulf petrol",
          "decisionDescription": "Harcamanız Reddedilmiştir. Açıkama: Bu harcama şirket
harcaması kapsamına giremez",
          "decisionDate": "2024-01-17T00:38:31.640193",
```

```
          "payment": null
        },
        {
          "expenseId": 10,
          "personalNumber": 2,
          "approvalStatus": 3,
          "expenseCreateDate": "2023-04-17T15:54:09.672294",
          "expenseAmount": 1750,
          "expenseDescription": "öğlen yemeği",
          "invoiceImageFilePath":
"/uploads/88ff7d06-1be3-45d2-996d-6067c4938e67_300_Akbank_Odev-1_2241a85fb3444bc
4bfafafd7def020fd.pdf",
          "location": "bodrum",
          "decisionDescription": "Harcamanız Reddedilmiştir. Açıkama: Öğlen yemeği için 1750
lira harcama keyfidir ",
          "decisionDate": "2024-01-18T17:05:38.42002",
          "payment": null
        }
      ]
    },
    {
      "personalResponse": {
        "iban": "12334567898564512344567 98",
        "personalNumber": 3,
        "userName": "sebastiansimanski",
        "firstName": "Sebastian",
        "lastName": "simanski",
        "email": "sebastian.simanski@akbank.com"
      },
      "moneySpentByPersonal": 2950,
      "approvedSpentMoney": 0,
      "rejectedSpentMoney": 0,
      "waitingExpenseList": [
        {
          "expenseId": 3,
          "personalNumber": 3,
          "approvalStatus": 1,
          "expenseCreateDate": "2024-01-17T14:13:35.446216",
          "expenseAmount": 1500,
          "expenseDescription": "A otelde konaklama ücreti",
          "invoiceImageFilePath":
"/uploads/799f7b3d-7b46-4728-a756-3445f0ff97d1_fatura-av-asilcan-tuzcu.jpg",
          "location": "Muğla",
          "decisionDescription": null,
          "decisionDate": null,
          "payment": null
        },
        {
```

```
          "expenseId": 15,
          "personalNumber": 3,
          "approvalStatus": 1,
          "expenseCreateDate": "2023-11-17T16:29:45.9284053",
          "expenseAmount": 1450,
          "expenseDescription": "A otelde konaklama ücreti",
          "invoiceImageFilePath":
"/uploads/88ff7d06-1be3-45d2-996d-6067c4938e67_300_Akbank_Odev-1_2241a85fb3444bc
4bfafafd7def020fd.pdf",
          "location": "gaziantep",
          "decisionDescription": null,
          "decisionDate": null,
          "payment": null
        }
      ],
      "aproveedExpenseList": [],
      "paymentList": [],
      "rejecetExpenseList": []
    },
    {
      "personalResponse": {
        "iban": "12345678981541654234456798",
        "personalNumber": 5,
        "userName": "edinceko",
        "firstName": "Edin",
        "lastName": "Ceko",
        "email": "Edin.ceko@akbank.com"
      },
      "moneySpentByPersonal": 5000,
      "approvedSpentMoney": 1500,
      "rejectedSpentMoney": 0,
      "waitingExpenseList": [
        {
          "expenseId": 17,
          "personalNumber": 5,
          "approvalStatus": 1,
          "expenseCreateDate": "2024-01-01T16:40:28.0070741",
          "expenseAmount": 5000,
          "expenseDescription": "Temizlik malzemesi temini",
          "invoiceImageFilePath":
"/uploads/88ff7d06-1be3-45d2-996d-6067c4938e67_300_Akbank_Odev-1_2241a85fb3444bc
4bfafafd7def020fd.pdf",
          "location": "bursa",
          "decisionDescription": null,
          "decisionDate": null,
          "payment": null
        }
      ],
```

```
      "aproveedExpenseList": [
        {
          "expenseId": 5,
          "personalNumber": 5,
          "approvalStatus": 2,
          "expenseCreateDate": "2024-01-17T15:17:02.4062304",
          "expenseAmount": 1500,
          "expenseDescription": "ofis malzemesi temini",
          "invoiceImageFilePath":
"/uploads/799f7b3d-7b46-4728-a756-3445f0ff97d1_fatura-av-asilcan-tuzcu.jpg",
          "location": "kadıköy",
          "decisionDescription": "Harcamanız onaylanmış ve ödeme emri tanımlanmıştır. Onay
açılaması: string",
          "decisionDate": "2024-01-17T17:07:13.7378447",
          "payment": null
        }
      ],
      "paymentList": [
        {
          "paymentId": 2,
          "expenseId": 5,
          "expense": null,
          "iban": "12345678981234456798",
          "paymentAmount": 1500,
          "description": "5 nolu şirket harcamanızın ücreti. Onay açıklaması:string",
          "paymentType": "EFT",
          "expenseDate": "2024-01-17T17:07:13.7384335"
        }
      ],
      "rejecetExpenseList": []
    },
    {
      "personalResponse": {
        "iban": "158753138484531551534843 51",
        "personalNumber": 6,
        "userName": "dusantadic",
        "firstName": "Dusan",
        "lastName": "Tadic",
        "email": "dusan.tadic@akbank.com"
      },
      "moneySpentByPersonal": 850,
      "approvedSpentMoney": 0,
      "rejectedSpentMoney": 0,
      "waitingExpenseList": [
        {
          "expenseId": 12,
          "personalNumber": 6,
          "approvalStatus": 1,
```

```
          "expenseCreateDate": "2023-06-17T16:05:20.2711619",
          "expenseAmount": 750,
          "expenseDescription": "öğlen yemeği",
          "invoiceImageFilePath":
"/uploads/88ff7d06-1be3-45d2-996d-6067c4938e67_300_Akbank_Odev-1_2241a85fb3444bc
4bfafafd7def020fd.pdf",
          "location": "Muğla",
          "decisionDescription": null,
          "decisionDate": null,
          "payment": null
        },
        {
          "expenseId": 18,
          "personalNumber": 6,
          "approvalStatus": 1,
          "expenseCreateDate": "2024-01-05T17:04:12.4843858",
          "expenseAmount": 100,
          "expenseDescription": "öğlen yemeği",
          "invoiceImageFilePath":
"/uploads/88ff7d06-1be3-45d2-996d-6067c4938e67_300_Akbank_Odev-1_2241a85fb3444bc
4bfafafd7def020fd.pdf",
          "location": "kocaeli",
          "decisionDescription": null,
          "decisionDate": null,
          "payment": null
        }
      ],
      "aproveedExpenseList": [],
      "paymentList": [],
      "rejecetExpenseList": []
    }
  ]
 }
}
```