



TechSiN Biliřim özmleri

Ali Nihat Tarlan Cad. Erta Sok. No:10 Kat:4 34572 İerenky Atařehir/İSTANBUL

Tel: +90 216 573 01 84 | Fax: +90 216 573 01 87

www.techsin.com.tr

AEGON Emeklilik ve Hayat

PDF-Service Web API Aıklamaları

İř Dokmanı

Dokman Kodu	:	Aegon-PDF-Service-Manual
Tarih	:	07.11.2018
Srm	:	1.0.0
Hazırlayan(lar)	:	Suat İMAMOĐLU-Gkhan YENER
Onaylayan(lar)	:	İsmail TellioĐlu, FUAT SATIR

Doküman Sürüm Geçmişi

Tarih	Sürüm	Değişiklikleri Yapan	Sürüm Açıklaması
17.05.2018	1.0.0	Suat İMAMOĞLU	Doküman oluşturuldu.
07.11.2018	1.0.1	Gökhan YENER	Doküman güncellendi.

İçerik

Doküman Sürüm Geçmişi	2
1 GENEL TANIMLAR	5
1.1 Amaç	5
1.2 Altyapı	5
1.3 RESTful API	5
1.4 Docker Konfigürasyonu	5
1.5 Symfony Konfigürasyonu	5
1.6 Sigorta Türleri	6
1.7 PDF Dosya Türleri	6
1.8 PDF Dosyaları	7
1.9 Database Yapısı	7
1.9.1 aegon_pdf Tablosu	7
1.9.2 aegon_pdf_log Tablosu	7
1.9.3 Database Tablolarını Oluşturmak için SQL	8
2 KURULUM	9
2.1 Ön Hazırlık	9
2.2 Start	9
2.3 Stop	9
2.4 Moreum Arşivleme Komutu	9
2.5 API Dokümanı (Swagger) Erişimi	10
3 WEBSERVİS KULLANIMI	11
3.1 KONFIGÜRASYON BİLGİLERİNİ SORGULAMA	11
3.1.1 Sigorta Türleri	11
3.1.2 PDF Dosya Tipleri	12
3.2 PDF DOSYASI LİSTELEME	13
3.3 PDF OLUŞTURMA VE GÜNCELLEME	14
3.3.1 Genel Açıklamalar	14

3.3.2	İhtiyaç Teyit Formu.....	16
3.3.3	Teklif Özeti.....	17
3.3.4	Başvuru Formu	18
3.3.5	Bilgilendirme Formu	21
3.3.6	Genel ve Özel Şartlar	21
3.3.7	Police	22
3.3.8	Police Okuma Kılavuzu	23
4	Dikkat Edilmesi Gerekenler ve Alınabilecek Hatalar	23
4.1.1	Teklif Özeti Pdf Oluşturma Hatası :.....	23
4.1.2	400 Bad Request Hatası :.....	23
4.1.3	500 Internal Server Error Hatası :.....	23
4.1.4	Cronjob log hataları	23
4.2	Loglama:	24
4.2.1	Database Loglarından Tekrardan Pdf Üretme	25
5	Web Servis Topoloji ve Mimarisi	26

1 GENEL TANIMLAR

1.1 Amaç

Projenin amacı, “Aegon Kolay Katılım” süreci esnasında müşterinin girmiş olduğu bilgileri derleyip ihtiyaç duyulan PDF dokümanlarının webservis aracılığıyla oluşturulmasıdır.

1.2 Altyapı

Altyapı olarak PHP7, MySQL, NGINX ve Symfony Framework v.3.4 kullanılmıştır. Sistem “docker-compose” desteği sayesinde herhangi bir Linux makinesinde kolayca çalıştırılabilir durumdadır (bk. 2. Kurulum). Docker üzerinden PHP ve NGINX container’leri çalıştırılır; veritabanı (MySQL) ise docker yapısı içerisinde olmayıp, mevcut olan harici bir veritabanı kullanılmıştır.

1.3 RESTful API

Webservis RESTful API olarak tasarlanmıştır. API dokümantasyonu Swagger formatında olup, sistemin çalıştırıldığı makinede erişilebilir durumdadır.

API Dokümantasyonu URL: <http://localhost:8888/doc>

1.4 Docker Konfigürasyonu

Docker konfigürasyonu *docker-compose.yml* dosyası üzerinden yapılır. Özellikle Nginx ayarı olarak hangi portun kullanılacağı önem taşıyabilir. Varsayılan port olarak 8888 ayarlanmıştır. Ancak aynı sunucuda bu port başka bir uygulama için kullanılmaktaysa *docker-compose.yml* dosyası içinde Nginx ayarlarından değiştirilebilir:

```
nginx:
  ports:
    - "8888:80"
```

1.5 Symfony Konfigürasyonu

Symfony projesinin temel konfigürasyonu *app/config/parameters.yml* dosyası içinde yapılır:

```
parameters:
  database_driver: pdo_mysql
  database_host: 10.184.33.111
  database_port: 3306
  database_name: kolaykatilim
  database_user: kolaykatilim
  database_password: inomera
  moreum_archival_wsdl: 'http://10.184.33.34/TerraDocsWS/Archival.svc?wsdl'
  moreum_retrieval_wsdl: 'http://10.184.33.34/TerraDocsWS/Retrieval.svc?wsdl'
```

Öncelikle `database_host`, `database_name`, `database_user` ayarları veritabanı bağlantısı için önemlidir ve kontrol edilmelidir.

Ayrıca `moreum_archival_wsdl` ve `moreum_retrieval_wsdl` ayarları Moreum arşivleme sistemi için önem taşıyor ve kontrol edilmelidir.

1.6 Sigorta Türleri

Webservis, Kolay Katılım projesi kapsamında oluşturulabilen 14 farklı sigorta çeşidi için PDF dosyası üretebilmektedir:

Sigorta Türü	Sigorta Kodu (slug)
5+5 Ödüllü Birikim Hayat Sigortası	5-5-odullu-birikim
Eğitim İçin Hayat Sigortası	egitim-icin-hayat-sigortasi
Ferdi Kazalı Prim İadeli Hayat Sigortası	ferdi-kazali-prim-iadeli
Ferdi Kazalı Prim İadeli Hayat Sigortası (Euro)	ferdi-kazali-prim-iadeli-euro
Grup Turuncu Elma Hayat Sigortası	grup-turuncu-elma
Ödeme Güvence Sigortası	odeme-guvence
Ödüllü Birikim Hayat Sigortası	odullu-birikim
Prim İadeli Hayat Sigortası	prim-iadeli
Prim İadeli Hayat Sigortası (Euro)	prim-iadeli-euro
3 Yıllık Prim İadeli Hayat Sigortası	prim-iadeli-3-yillik
5 Yıllık Prim İadeli Hayat Sigortası	prim-iadeli-5-yillik
Tek Prim Ödemeli Hayat Sigortası	tek-prim-odemeli
5 Yıllık Tek Prim Ödemeli Hayat Sigortası	tek-prim-odemeli-5-yillik
Turuncu Elma Hayat Sigortası	turuncu-elma

1.7 PDF Dosya Türleri

Her sigorta çeşidi için 7 farklı PDF dosyası üretebilmektedir:

PDF Türü	PDF Kodu (slug)
İhtiyaç Teyit Formu	requirements-confirmation
Teklif Özeti	offer-summary
Başvuru Formu	application-form
Bilgilendirme Formu	information-form
Genel ve Özel Şartlar	terms
Poliçe	insurance-policy
Poliçe Okuma Kılavuzu	policy-manual

1.8 PDF Dosyaları

PDF dosyaları proje içerisindeki “web/download” ana klasörü altındaki alt klasörlere kaydedilir. Alt klasör ismi ve dosya ismi veritabanına kaydedilmektedir (“dir_name” ve “file_name” alanlarında).

1.9 Database Yapısı

Webservice ait iki tablo mevcuttur: “aegon_pdf” ve “aegon_pdf_log”.

1.9.1 aegon_pdf Tablosu

aegon_pdf tablosu güncel/geçerli olan PDF dosyalarının bilgilerini tutmaktadır.

Field	Type	Örnek	Açıklama
id	int(11)	1	İntern ID (autoincrement)
moreum_document_id	int(11)	83905	Dosya Moreum’a kaydedildiyse, Moreum’un “DocumentId” değeri, aksi takdirde NULL
personal_id_number	varchar(11)	12345678901	TCKN
pol_id	varchar(64)	2223524	POLID
policy_number	varchar(64)	ROP_AD_3017060001	Poliçe No
insurance_slug	varchar(128)	5-5-odullu-birikim	Sigorta türü kodu (slug)
pdf_type_slug	varchar(128)	01-ihciyac-teyit	PDF türü kodu (slug)
dir_name	varchar(128)	1234567/89	Dosyanın kaydedildiği alt klasör
file_name	varchar(255)	aegon_2223524_5-5-odullu-birikim_01-ihci.pdf	Dosya ismi
json_data	longtext	-	Webservice gönderilen JSON objesi
json_processed	longtext	-	JSON objesinin derlenip PDF oluşturmaya uygun hali
created_at	datetime		Oluşturma tarihi
updated_at	datetime		Güncelleme tarihi

1.9.2 aegon_pdf_log Tablosu

aegon_pdf_log tablosu, **aegon_pdf** tablosunun birebir kopyasıdır. **aegon_pdf** tablosunda her yeni kayıt oluşturulduğunda ve/veya var olan bir kayıt güncellendiğinde, ilgili kaydın birebir kopyası **aegon_pdf_log** tablosuna da yeni bir kayıt olarak kaydedilmektedir.

Böylelikle yeni oluşturulan veya güncellenen PDF dosyalarının geçmişi kayıt altına alınmaktadır.

1.9.3 Database Tablolarını Oluşturmak için SQL

MySQL database tablolarını oluşturmak için aşağıdaki SQL kodları çalıştırılabilir:

```
CREATE TABLE `aegon_pdf` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `personal_id_number` varchar(11) COLLATE utf8_unicode_ci NOT NULL,  
  `insurance_slug` varchar(128) COLLATE utf8_unicode_ci NOT NULL,  
  `pdf_type_slug` varchar(128) COLLATE utf8_unicode_ci NOT NULL,  
  `dir_name` varchar(128) COLLATE utf8_unicode_ci NOT NULL,  
  `file_name` varchar(255) COLLATE utf8_unicode_ci NOT NULL,  
  `json_data` longtext COLLATE utf8_unicode_ci,  
  `json_processed` longtext COLLATE utf8_unicode_ci,  
  `created_at` datetime DEFAULT NULL,  
  `updated_at` datetime DEFAULT NULL,  
  `moreum_document_id` int(11) DEFAULT NULL,  
  `pol_id` varchar(64) COLLATE utf8_unicode_ci NOT NULL,  
  `policy_number` varchar(64) COLLATE utf8_unicode_ci DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  KEY `idx_pdf_personal_id_number` (`personal_id_number`),  
  KEY `idx_pdf_insurance_slug` (`insurance_slug`),  
  KEY `idx_pdf_pdf_type_slug` (`pdf_type_slug`),  
  KEY `idx_pdf_dir_name` (`dir_name`),  
  KEY `idx_pdf_file_name` (`file_name`),  
  KEY `idx_pdf_created_at` (`created_at`),  
  KEY `idx_pdf_updated_at` (`updated_at`),  
  KEY `idx_pdf_moreum_document_id` (`moreum_document_id`),  
  KEY `idx_pdf_pol_id` (`pol_id`),  
  KEY `idx_pdf_policy_number` (`policy_number`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;  
  
CREATE TABLE `aegon_pdf_log` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `personal_id_number` varchar(11) COLLATE utf8_unicode_ci NOT NULL,  
  `insurance_slug` varchar(128) COLLATE utf8_unicode_ci NOT NULL,  
  `pdf_type_slug` varchar(128) COLLATE utf8_unicode_ci NOT NULL,  
  `dir_name` varchar(128) COLLATE utf8_unicode_ci NOT NULL,  
  `file_name` varchar(255) COLLATE utf8_unicode_ci NOT NULL,  
  `json_data` longtext COLLATE utf8_unicode_ci,  
  `json_processed` longtext COLLATE utf8_unicode_ci,  
  `created_at` datetime DEFAULT NULL,  
  `updated_at` datetime DEFAULT NULL,  
  `moreum_document_id` int(11) DEFAULT NULL,  
  `pol_id` varchar(64) COLLATE utf8_unicode_ci NOT NULL,  
  `policy_number` varchar(64) COLLATE utf8_unicode_ci DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  KEY `idx_pdf_log_personal_id_number` (`personal_id_number`),  
  KEY `idx_pdf_log_insurance_slug` (`insurance_slug`),  
  KEY `idx_pdf_log_pdf_type_slug` (`pdf_type_slug`),  
  KEY `idx_pdf_log_dir_name` (`dir_name`),  
  KEY `idx_pdf_log_file_name` (`file_name`),  
  KEY `idx_pdf_log_created_at` (`created_at`),  
  KEY `idx_pdf_log_updated_at` (`updated_at`),  
  KEY `idx_pdf_moreum_document_id` (`moreum_document_id`),  
  KEY `idx_pdf_pol_id` (`pol_id`),  
  KEY `idx_pdf_policy_number` (`policy_number`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```


2 KURULUM

2.1 Ön Hazırlık

Uygulamanın çalıştırılacağı sunucuda Docker altyapısının kurulmuş olması gerekir.

Uygulama ZIP dosyası halinde teslim edilir. Bu ZIP dosyası sistemin çalışacağı sunucuya yüklenip öngörülen klasör içinde açılmalıdır ("extract" edilmelidir).

Örn.

```
cp pdf-service-1.0.0.zip /var/www  
cd /var/www/  
unzip pdf-service-1.0.0.zip
```

2.2 Start

Uygulamayı çalıştırmak için konsol ("command line" - CLI) ile önceki adımda extract edilmiş olan uygulamanın ana klasörüne geçip *app-start.sh* scripti çalıştırılmalıdır.

Örn.

```
cd /var/www/pdf-service-1.0.0/  
./app-start.sh
```

Bu şekilde iki tane Docker container'i çalıştırılmış olacaktır (nginx ve php).

2.3 Stop

Uygulamayı durdurmak için yine konsol ile uygulamanın ana klasörüne geçip, *app-stop.sh* scripti çalıştırılmalıdır:

Örn.

```
cd /var/www/pdf-service-1.0.0/  
./app-stop.sh
```

2.4 Moreum Arşivleme Komutu

Servis üzerinden oluşturulan tüm PDF dosyalarının ayrıca Moreum Arşivleme sistemine aktarılması gerekmektedir.

Bu işlem için belirli aralıklarla bir komutun çalıştırılması gerekmektedir. Bunun için servisin çalıştırıldığı makinede (“docker host”) bir cronjob’un ayarlanması gerekli. Bu cronjob örn. 5 veya 10 dakikada bir çalışmalıdır.

Bu şekilde ayarlanmış olan cronjob, şu komutları çalıştırmalıdır:

```
cd /var/www/pdf-service-1.0.0/  
docker-compose exec php bin/console app:moreum:archival --env=prod
```

Bu komut henüz arşivlenmemiş olan tüm PDF dosyalarını Moreum’a aktarır ve aktarılan her PDF dosyası için veritabanındaki aegon_pdf tablosunda moreum_document_id alanına Moreum’dan dönen ID’yi kaydeder.

2.5 API Dokümanı (Swagger) Erişimi

Detaylı API dokümantasyonu Swagger formatında olup, sistemin çalıştırıldığı makinede erişilebilir durumdadır:

<http://localhost:8888/doc>

3 WEBSERVİS KULLANIMI

Webservis kullanımı öncelikle API Dokümantasyonu sayfasından deneyimlenebilir ve test edilebilir:

<http://localhost:8888/doc>

Bu sayfada tüm API endpoint'leri ve kullanılan model yapısı, örnek sorgular vs. açıklanmıştır. Webservisin sunduğu tüm endpointler için detaylı örnek sorgular mevcut olup, bu sayfa üzerinden doğrudan test edilebilir.

3.1 KONFIGÜRASYON BİLGİLERİNİ SORGULAMA

İlk aşamada webservisin sunduğu "Sigorta Türlerini" ve "PDF Dosya Tiplerini" sorgulamak gerekebilir. Bu bilgiler diğer sorguları oluşturmak için gereklidir.

3.1.1 Sigorta Türleri

Sigorta türlerini sorgulamak için kullanılan endpoint:

Metod	URL
GET	http://localhost:8888/api/configuration/insurance-slugs
Response	<pre>{ "5-5-odullu-birikim": "5+5 Ödüllü Birikim Hayat Sigortası", "egitim-icin-hayat-sigortasi": "Eğitim İçin Hayat Sigortası", "ferdi-kazali-prim-iadeli": "Ferdî Kazalı Prim İadeli Hayat Sigortası", "grup-turuncu-elma": "Grup Turuncu Elma Hayat Sigortası", "odeme-guvence": "Ödeme Güvence Sigortası", "odullu-birikim": "Ödüllü Birikim Hayat Sigortası", "prim-iadeli": "Prim İadeli Hayat Sigortası", "prim-iadeli-3-yillik": "3 Yıllık Prim İadeli Hayat Sigortası", "prim-iadeli-5-yillik": "5 Yıllık Prim İadeli Hayat Sigortası", "tek-prim-odemeli": "Tek Prim Ödemeli Hayat Sigortası", "tek-prim-odemeli-5-yillik": "5 Yıllık Tek Prim Ödemeli Hayat Sigortası", "turuncu-elma": "Turuncu Elma Hayat Sigortası" }</pre>

3.1.2 PDF Dosya Tipleri

Webservisin oluşturabildiği PDF dosya tiplerini sorgulamak için:

Metod	URL
GET	http://localhost:8888/api/configuration/pdf-type-slugs
Response	<pre>{ "requirements-confirmation": "İhtiyaç Teyit Formu", "offer-summary": "Teklif Özeti", "application-form": "Başvuru Formu", "information-form": "Bilgilendirme Formu", "terms": "Genel ve Özel Şartlar", "insurance-policy": "Poliçe", "policy-manual": "Poliçe Okuma Kılavuzu" }</pre>

3.2 PDF DOSYASI LİSTELEME

Herhangi bir müşteriye ait daha önceden oluşturulmuş olan PDF dosyalarını listelemek için kullanılır.

Metod	URL
GET	http://localhost:8888/api/pdf/get-by-customer/{personalIdNumber}/{polId}/{insuranceSlug}

Parametre	Açıklama
personalIdNumber	Sigortalının TC Kimlik No'su
polId	Sigorta POLID'si
insuranceSlug	Sigorta türünün kodu

Örnek	http://localhost:8888/api/pdf/get-by-customer/12345678901/2223524/5-5-odullu-birikim
Response	<pre>[{ "id": 73, "moreumDocumentId": 84249, "personalIdNumber": "12345678901", "polId": "2223524", "policyNumber": "ROP_AD_3017060001", "insuranceSlug": "5-5-odullu-birikim", "pdfTypeSlug": "06-police", "dirName": "1234567/89", "fileName": "aegon_2223524_5-5-odullu-birikim_06-polc.pdf", "url": "/download/1234567/89/aegon_2223524_5-5-odullu-birikim_06-polc.pdf" }]</pre>

Bu örnekte oluşturulmuş olan PDF dosyasını şu URL'den indirmek mümkündür:

http://localhost:8888/download/1234567/89/aegon_2223524_5-5-odullu-birikim_06-polc.pdf

Dosyalara ayrıca webservisin çalıştırıldığı sunucuda (docker host) şu klasörden de ulaşılabilir:

Örn.:

```
/var/www/pdf-service-1.0.0/web/
download/1234567/89/aegon_2223524_5-5-odullu-birikim_06-polc.pdf
```

3.3 PDF OLUŞTURMA VE GÜNCELLEME

3.3.1 Genel Açıklamalar

3.3.1.1 Endpointler ve POST / PUT Metodları

PDF dosyası oluşturmak için her PDF Dosya Türüne özel ayrı bir endpoint bulunmaktadır.

PDF oluşturma veya güncelleme için yapılan tüm istekler POST veya PUT metoduyla yapılmalıdır.

Her endpoint, request metoduna (POST veya PUT) bağlı olarak, RESTful API yapısına uygun olacak şekilde, yeni bir dosya oluşturur (POST = create) veya var olan bir dosyayı günceller (PUT = update).



Henüz oluşturulmamış olan bir PDF dosyası için PUT (update) metodu kullanılamaz, aksi takdirde sistem hata verir. Aynı şekilde, var olan bir PDF dosyası için POST (create) metodu kullanılamaz, yoksa sistem hata verir.

Tüm endpointlerin detayları örnekleriyle beraber API dokümantasyon sayfasında (<http://localhost:8888/doc>) belirtilmiştir.

Aşağıdaki açıklamalar, her PDF türü için özellikle dikkat edilmesi gereken ayrıntıları içerir.

3.3.1.2 Model Yapısı

Her isteğin request body kısmında, PDF dosyasının içeriğini belirleyen JSON objesi bulunmalıdır. Bu JSON objeleri, her PDF türü için farklıdır ve API dokümantasyon sayfasında ilgili endpointun altında örnekleriyle beraber detaylıca açıklanmıştır:

PDF Edit - Requirements Confirmation (İhtiyaç Teyit) ▾

POST /api/pdf/requirements-confirmation/{personalIdNumber} Creates a new Requirements Confirmation PDF file (İhtiyaç Teyit)

Parameters Try it out

Name	Description
personalIdNumber ★ required string (path)	Personal ID Number (TC Kimlik No) of the insured person (e.g. '12345678901')
body ★ required (body)	PDF object that needs to be created Example Value Model

```
RequirementsConfirmationPdf ▾ {
  description: İhtiyaç Teyit
  insurance_slug: string
    example: 5-5-odullu-birikim
    maxLength: 64
    minLength: 64
    Sigorta Türü Kodu (insuranceSlug)
```

Tüm JSON objelerinin (Model) ortak ve zorunlu alanları şunlardır:

insurance_slug	Sigorta türü kodu (slug), örn. "5-5-odullu-birikim"
pol_id	Sigortanın POLID'si
policy_number	Sigortanın Poliçe No'su
back_url	Geri gitme linki (PDF dosyasının her sayfasının alt kısmında "İndex" yazılı bir imaj linki basılır). Kullanıcı PDF dosyasını tarayıcı içinde açtığında bu linke tıklayınca geldiği sayfaya geri gidecektir.

Tüm endpoint URL'lerinin ortak parametresi şudur:

personalIdNumber	Sigortalı kişinin TC Kimlik No'su
------------------	-----------------------------------

3.3.2 İhtiyaç Teyit Formu

İhtiyaç Teyit Formu oluştururken gönderilen JSON objesinde özellikle dikkat edilmesi gereken alanlar:

POST/PUT	/api/pdf/requirements-confirmation/{personalIdNumber}
insurer_sign_image_url	Sigorta ettiren kişinin imzasının bulunduğu bir imajın URL'sidir. Bu URL'ye webservisin kurulu olduğu sunucudan erişilebilmelidir. Bu imaj, PDF'in imza kısmına basılacaktır.



The image shows a template for an insurance confirmation form. On the left, there is a circular logo with a stylized 'A' and a checkmark. To the right of the logo, the text 'SİGORTA ETTİREN' is displayed. Further right, the fields 'Ad Soyad: Jane Doe' and 'Tarih: 01.01.1980' are shown. A red arrow points from the text 'SİGORTA ETTİREN' to a rectangular box on the right, which contains a handwritten signature 'JD'.

3.3.3 Teklif Özeti

Teklif Özeti endpoint'u, aslında var olan ve farklı bir webservis tarafından üretilmiş olan teklif özeti PDF dosyalarını *Kolay Kullanım* projesi tasarımıyla süslemektedir. Dolayısıyla bu endpoint'e var olan PDF dosyasının URL'si bildirilmelidir ([offer_summary_pdf_url](#)).

Dikkat edilmesi gereken alanlar:

POST/PUT	/api/pdf/offer-summary/{personalIdNumber}
offer_summary_pdf_url	Var olan teklif özeti PDF dosyasının URL'si bu alana yazılmalıdır. İlgili URL'ye webservisin kurulu olduğu sunucudan erişilebilmelidir.
insurer_initials_image_url	Sigorta ettiren kişinin parafının bulunduğu bir imajın URL'sidir. Bu URL'ye webservisin kurulu olduğu sunucudan erişilebilmelidir. Bu imaj, tüm PDF sayfalarının alt kısmına basılacaktır.



3.3.4 Başvuru Formu

Başvuru Formu diğer PDF türlerine nazaran daha karmaşık olan bir PDF dosyası tipidir. Gönderilen JSON objesinin yapısına göre PDF'in sayfa sayısı, görsel yapısı ve içeriği büyük oranda değişmektedir. Bu yüzden aşağıdaki açıklamaların özenle dikkate alınması gerekir.

POST/PUT	/api/pdf/application-form/{personalIdNumber}
guarantees	Array halinde 1 veya daha fazla teminat bilgisini içerir (<i>ApplicationFormGuarantee</i>).
credit_cards	Array halinde 1 veya (maks.) 2 kredi kartı bilgisini içerir (<i>ApplicationFormCreditCard</i>). PDF dosyasına en fazla 2 tane kredi kartı bilgisi basılabilir. Bunlardan birinin <i>is_preferred</i> bilgisi <i>true</i> (boolean) olmalıdır.
endorseds	Array halinde 1 veya daha fazla lehdar bilgisini içerir (<i>ApplicationFormEndorsed</i>).
mediator	Aracı bilgilerini içerir (<i>ApplicationFormMediator</i>).
insured_person ⁽¹⁾	Sigortalanan kişinin bilgilerini içerir (<i>ApplicationFormInsuredPerson</i>). <i>Null olamaz</i> ⁽¹⁾ .
insurer_person ⁽¹⁾	Sigortalayan kişinin (eğer gerçek kişiye) bilgilerini içerir (<i>ApplicationFormInsurerPerson</i>). <i>Null olabilir</i> ⁽¹⁾ .
insurer_company ⁽¹⁾	Sigortalayan kişinin (eğer tüzel kişiye, yani şirket ise) bilgilerini içerir (<i>ApplicationFormInsurerPerson</i>). <i>Null olabilir</i> ⁽¹⁾ .
lifestyle	Yaşam tarzı bilgilerini içerir (<i>ApplicationFormLifestyle</i>). <i>health_declaration</i> bilgisi ("Sağlık Beyan Metni") null olduğu takdirde bu alan ve başlığı basılmaz.
health_basic ⁽²⁾⁽³⁾	5 sorudan oluşan sağlık bilgilerini içerir (<i>ApplicationFormHealthBasic</i>). <i>Null olabilir</i> ⁽²⁾ .
health_extended ⁽²⁾⁽³⁾	12 sorudan oluşan sağlık bilgilerini içerir (<i>ApplicationFormHealthExtended</i>). <i>Null olabilir</i> ⁽²⁾ .
confirmation ⁽⁴⁾	<i>ApplicationFormConfirmation</i> objesini içerir (bk. aşağıdaki açıklamalar)

(1) insured_person, insurer_person, insurer_company Kombinasyonları:

Bu alanların kombinasyonlarına göre "Sigortalayan" (gerçek/tüzel şahıs) bilgileri oluşturulur:

- insurer_person = null AND insurer_company = null*
Sigortalayan = Sigortalanan kabul edilir (yani sigortalayan kişi, kendisi için sigorta oluşturmuştur)
- insurer_person != null AND insurer_company = null*
Sigortalayan kişi gerçek şahıs ve sigortalanan şahıstan farklı biridir
- insurer_person = null AND insurer_company != null*
Sigortalayan tüzel kişidir (şirket)

(2) `health_basic`, `health_extended` Kombinasyonları:

Bu alanların kombinasyonlarına göre 5 sorulu veya 12 sorulu sağlık bilgileri basılır:

- `health_basic != null`
5 sorudan oluşan “Sağlık Bilgileri” basılır
- `health_basic = null AND health_extended != null`
12 sorudan oluşan “Sağlık Bilgileri” basılır

(3) “Sağlık Bilgileri” sayfasının altına bazı durumlarda *sürprim hesaplamaları* basılmaktadır.

Bu hesaplamaların basılması için, `health_basic` (`ApplicationFormHealthBasic`) veya `health_extended` (`ApplicationFormHealthExtended`) objesinin şu alanlarının boş bırakılmaması veya null olmaması gerekir:

<code>additional_guarantees_explanation</code>	Sürprim Teminat Tablosu Açıklaması
<code>additional_guarantees_name_title</code>	Sürprim Teminat Tablosu (1. Başlık): Teminat İsmi
<code>additional_guarantees_before_title</code>	Sürprim Teminat Tablosu (2. Başlık): Uygulama Öncesi
<code>additional_guarantees_percent_title</code>	Sürprim Teminat Tablosu (3. Başlık): Sürprim Oranı
<code>additional_guarantees_after_title</code>	Sürprim Teminat Tablosu (4. Başlık): Uygulama Sonrası
<code>additional_guarantees_table_rows</code>	Array şeklinde 1 veya daha fazla teminat bilgisi içerir (<code>HealthAdditionalGuaranteesTableRow</code>).

Ön Değerlendirme

Başvurunuzla istinaden yapılan ön değerlendirme sonuçları ...

Yeni Teminat Bilgilendirme ve Onay

Teminat	Uygulama Öncesi	Sürprim Oranı	Prim Artışı
Vefat Teminatı	1.000 TL	%10	1.000 TL
Vefat Teminatı	1.000 TL	%10	1.000 TL

(4) **ApplicationFormConfirmation** objesi için dikkat edilmesi gereken alanlar:

<code>insurer_sign_image_url</code>	Sigorta ettiren kişinin imzasının bulunduğu bir imajın URL'sidir. Bu URL'ye webservisin kurulu olduğu sunucudan erişilebilmelidir. Bu imaj, PDF'in son sayfasına basılacaktır. Null olabilir veya boş bırakılabilir.
<code>insurer_initials_image_url</code>	Sigorta ettiren kişinin parafının bulunduğu bir imajın URL'sidir. Bu URL'ye webservisin kurulu olduğu sunucudan erişilebilmelidir. Bu imaj, tüm PDF sayfalarının alt kısmına basılacaktır.
<code>company_sign_image_url</code>	Sigorta ettiren şirketin imzasının bulunduğu bir imajın URL'sidir. Bu URL'ye webservisin kurulu olduğu sunucudan erişilebilmelidir. Bu imaj, PDF'in son sayfasına basılacaktır. Null olabilir veya boş bırakılabilir.

`insurer_sign_image_url` ve `company_sign_image_url` boş/null olmadığı durumda PDF'in son sayfası şu şekilde basılır:



`insurer_sign_image_url` ve `insurer_sign_image_url` boş/null olduğu durumda PDF'in son sayfası şu şekilde basılır:



3.3.5 Bilgilendirme Formu

Bilgilendirme Formu oluştururken gönderilen JSON objesinde özellikle dikkat edilmesi gereken alanlar:

POST/PUT	/api/pdf/information-form/{personalIdNumber}
insurer_sign_image_url	Sigorta ettiren kişinin imzasının bulunduğu bir imajın URL'sidir. Bu URL'ye webservisin kurulu olduğu sunucudan erişilebilmelidir. Bu imaj, PDF'in son sayfasına basılacaktır.
insurer_initials_image_url	Sigorta ettiren kişinin parafının bulunduğu bir imajın URL'sidir. Bu URL'ye webservisin kurulu olduğu sunucudan erişilebilmelidir. Bu imaj, tüm PDF sayfalarının alt kısmına basılacaktır.
company_sign_image_url	Sigorta ettiren şirketin imzasının bulunduğu bir imajın URL'sidir. Bu URL'ye webservisin kurulu olduğu sunucudan erişilebilmelidir. Bu imaj, PDF'in son sayfasına basılacaktır.

3.3.6 Genel ve Özel Şartlar


Genel ve Özel Şartlar PDF dosyasını oluştururken default değerlerin dışında özellikle dikkat edilmesi gereken alanlar bulunmamaktadır.

3.3.7 Poliçe

Poliçe dosyası oluştururken JSON objesinde özellikle dikkat edilmesi gereken alanlar:

POST/PUT	/api/pdf/policy/{personalIdNumber}
details	Array şeklinde 1. sayfaya basılan poliçe detaylarını içerir (<i>PolicyDetay</i>).
guarantees	Array şeklinde poliçe teminatlarını içerir (<i>PolicyGuarantee</i>).
payback_years	Array şeklinde yıllık geri ödeme tarihlerini içerir (<i>PolicyPaybackYear</i>). Sadece “Eğitim İçin Hayat Sigortası” için geçerlidir.
redemption_fees	Array şeklinde “Matematik Karşılık ve İştirak Bedeli” tablosunun satırlarını içerir (<i>PolicyRedemptionFee</i>). Sadece “Ferdî Kazalı Prim İadeli Hayat Sigortası” ve “Prim İadeli Hayat Sigortası” için geçerlidir.
endorseds	Array şeklinde poliçe lehdarlarının bilgilerini içerir (<i>PolicyEndorsed</i>).
mediator_details ⁽¹⁾	Array şeklinde poliçe aracı bilgileri tablosunun satırlarını içerir (<i>PolicyMediatorDetail</i>) ⁽¹⁾ .
is_signed	Boolean olarak true gönderilirse, aşağıdaki imajlar için geçerli URL’ler gönderilmelidir. Aksi takdirde aşağıdaki imza resimleri PDF’e basılmaz.
company_sign_image_url	Sigorta ettiren kişinin imzasının bulunduğu bir imajın URL’sidir. Bu URL’ye webservisin kurulu olduğu sunucudan erişilebilmelidir. Bu imaj, PDF’in son sayfasına basılacaktır.
aegon_sign_image_url	Aegon şirket yetkilisinin imzasının bulunduğu bir imajın URL’sidir. Bu URL’ye webservisin kurulu olduğu sunucudan erişilebilmelidir. Bu imaj, PDF’in son sayfasına basılacaktır.

- (1) **PolicyMediatorDetail** objeleri **mediator_details** alanında array şeklinde gönderilir ve sonuç olarak şu şekilde PDF’e basılır:

	Aracı Bilgileri
Acente:	Acente
Acente Adı:	Örnek Sigorta Acentesi Ltd.
Acente Aracı Adı:	John Doe
Acente Levha Kayıt No:	1234567890
HAYMER Tarife Kodu:	1234567890
Poliçe No:	1234567890

3.3.8 Poliçe Okuma Kılavuzu

Poliçe Okuma Kılavuzu PDF dosyasını oluştururken default değerlerin dışında özellikle dikkat edilmesi gereken alanlar bulunmamaktadır.

4 Dikkat Edilmesi Gerekenler ve Alınabilecek Hatalar

4.1.1 Teklif Özeti Pdf Oluşturma Hatası :

Teklif özeti pdf oluşturulurken, İnomera'ya ait olan offer_summary_pdf_url kısmındaki linkin doğru olduğu kontrol edilmelidir. Bir örneği aşağıda yer almaktadır.

```
"offer_summary_pdf_url":"https://teklif.aegon.com.tr//Teklif/ROL5_5/OdulluBirikim5Arti5Sonuc.aspx?  
TeklifID=ROL5_5_24102018_88&Hash=774C440204F269D75C8EE4783963FA  
2226AD303E79CAED0CD6EA1C80245F64F4&Mode=4",
```

4.1.2 400 Bad Request Hatası :

Bu hata oluştuysa, oluşturulmak istenen pdf dosyası, daha önceden oluşturulmuş anlamına gelmektedir. Bu işlemin doğru bir şekilde çalışması için, pdf'i tekrardan oluşturmak yerine (POST), güncelleme(PUT) yapılması gerekmektedir.

- POST metoduyla yeni kayıt ve PDF oluşturulur
- PUT metoduyla var olan kayıt/PDF güncellenir

```
(Symfony\\Component\\HttpKernel\\Exception\\BadRequestHttpException(co  
> de: 0): A PDF file for the given personalIdNumber \"18053100110\",  
> polId \"2290676\", insuranceSlug \"prim-iadeli\" and pdfTypeSlug  
> \"06-police\" already exists. Please update it instead of creating a  
> new one. at
```

4.1.3 500 Internal Server Error Hatası :

Bu hata, POST yada PUT metoduyla, pdf servisine gönderilen parametrelerle ilgili bir hatadır. Dolayısıyla gönderilen parametreleri kontrol ediniz. Bu konuda detaylı bilgiye pdf servisi dokümanından ulaşabilirsiniz.

API Dokümantasyonu URL: <http://localhost:8888/doc>

4.1.4 Cronjob log hataları

- Aşağıdaki gibi , “No such file or directory in” hatası oluştuğunda, pdf dosyasının kaydının database de oluştuğunu, fakat pdf dosyasının web/download (diskte) klasöründe olmadığını göstermektedir. Dolayısıyla Moreum arşivleme sistemine pdf dosyasının aktarılmadığını

göstermektedir. Çözüm için, database den olmayan pdf lerin kaydı silinebilir yada aegon_pdf tablosundaki is_archival_required alanı 0 yapılabilir.

Warning:

file_get_contents(/var/www/symfony/web/download/2324612/85/aegon_2289448_turuncu-elma_01-ihtc.pdf): failed to open stream: No such file or directory in
/var/www/symfony/src/AppBundle/Command/AppMoreumArchivalCommand.php on line 176

- b) Aşağıdaki gibi “Undefined Index” hatası durumunda,

PHP Notice: Undefined index: Poliçe Okuma Kılavuzu in

/var/www/symfony/src/AppBundle/Command/AppMoreumArchivalCommand.php on line 183

Moreum tarafında gönderilmek istenen arşivleme sistemi ile aşağıdaki kısımda yer alan eşleşmelerde hata olduğunu göstermektedir. Bu durumda Moreum arşivleme sisteminde bir değişikliğe gidilmiş olabilir. Sağ taraftaki parametreler Moreum tarafından kontrol edilmelidir.

'01-ihtiyac-teyit' : 'IHTIYAC_TEYIT'
'02-teklif-ozet': 'TEKLIF_OZET'
'03-basvuru-formu': 'HAYAT_BASVURU'
'04-bilgilendirme-formu': 'BILGILENDIRME'
'05-ozel-ve-genel-sartlar': 'OZELGENELSARTLAR'
'06-police': 'POLICE'
'07-police-okuma-kilavuzu': 'POLICEOKUMAKILAVUZU'

- c) Karşılaşılabilecek bir diğer problem, Arşivleme sistemi için Moreum bağlantı parametreleridir. Herhangi bir hata durumunda filesystem deki app/config/parameters.yml da yer alan aşağıdaki bilgiler kontrol edilmelidir.

#Moreum Test Sunucusu

#moreum_archival_wsdl: 'http://moreum-test.aegon.com.tr/TerraDocsWS/Archival.svc?wsdl'

#moreum_retrieval_wsdl: 'http://moreum-test.aegon.com.tr/TerraDocsWS/Retrieval.svc?wsdl'

Moreum Production Sunucusu

moreum_archival_wsdl: 'http://10.184.33.34/TerraDocsWS/Archival.svc?wsdl'

moreum_retrieval_wsdl: 'http://10.184.33.34/TerraDocsWS/Retrieval.svc?wsdl'

4.2 Loglama:

Log dosyaları 3 farklı şekilde loglanmaktadır.

- a) **Database:** Database de, bugüne kadar Moreum arşivleme sistemine aktarılmış, aktarılmamış bütün pdf lerin log ları **aegon_pdf_log** tablosunda yer almaktadır.

- b) **Filesistem:** var/log klasöründe yer almaktadır. Oluşturulmak istenen her pdf dosyası için başarılı olduğunda info mesajı, başarısız olduğunda ise başarısızlıkla ilgili error mesajı, log dosyasına işlenmektedir.
- c) **Cronjob :** Cronjob çalıştığında Moreum arşivleme sistemine başarılı ve başarısız olarak gönderilen pdf sayısı, ayrıca pdf gönderilemediği takdirde hata mesajı, 4.1.4 de ifade edildiği şekilde loglarda işlenmektedir.

4.2.1 Database Loglarından Tekrardan Pdf Üretme

Database logları aegon_pdf_log tablosunda tutulmaktadır. Bu tabloda servise gelen bütün request ler kayıt edilmektedir. Bu log kayıtlarından tekrardan pdf üretebilmek için, belirli aralıklarla bir komutun çalıştırılması gerekmektedir. Bunun için servisin çalıştırıldığı makinede ("docker host") bir cronjob'un ayarlanabilir. Bu cronjob örn. 12 veya 24 saatte bir çalışabilir.

Bu şekilde ayarlanmış olan cronjob, şu komutları çalıştırmalıdır:

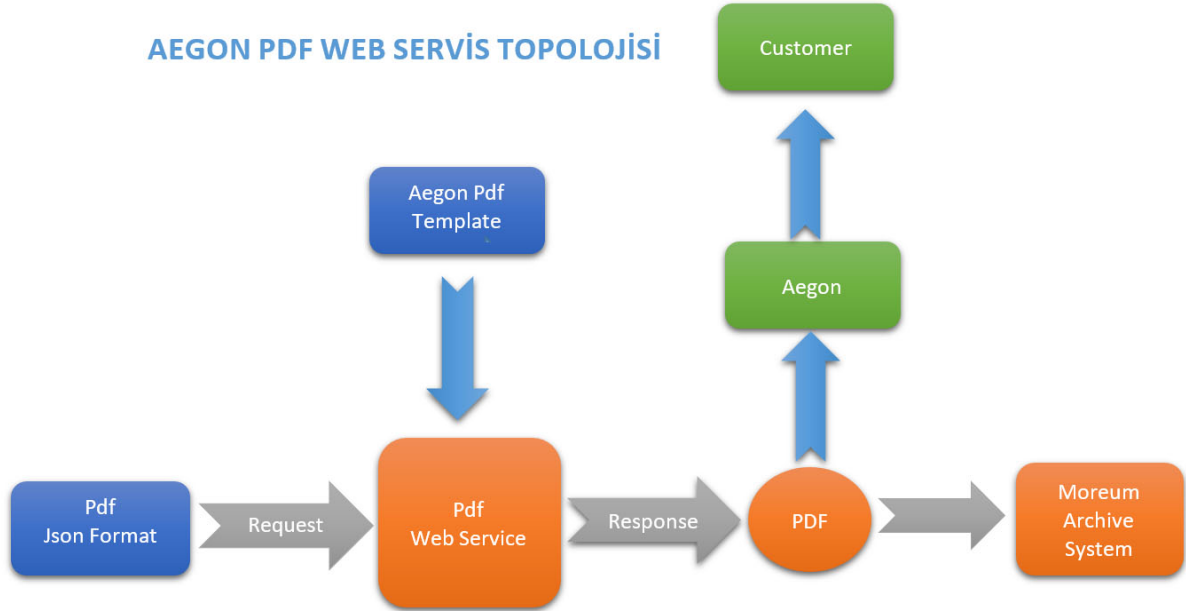
```
cd /var/www/pdf-service-1.0.0/  
docker-compose exec php bin/console app:pdf-regenerate -env=prod
```

Konsol ekranından aşağıdaki şekilde oluşturulan pdfler izlenebilir.

```
pdf is regenerating ...id:13, filename: aegon_2223524_tek-prim-odemeli_03-basv.pdf, folder: 8/93  
pdf regenerated ...id:13, filename: aegon_2223524_tek-prim-odemeli_03-basv.pdf, folder: 8/93  
pdf is regenerating ...id:14, filename: aegon_2223524_tek-prim-odemeli-5-yillik_03-basv.pdf, folder: 89/23  
pdf regenerated ...id:14, filename: aegon_2223524_tek-prim-odemeli-5-yillik_03-basv.pdf, folder: 89/23  
pdf is regenerating ...id:15, filename: aegon_2223524_turuncu-elma_01-ihtc.pdf, folder: 77/20  
pdf regenerated ...id:15, filename: aegon_2223524_turuncu-elma_01-ihtc.pdf, folder: 77/20  
pdf is regenerating ...id:16, filename: aegon_2223524_tek-prim-odemeli-5-yillik_03-basv.pdf, folder: 89233/53  
pdf regenerated ...id:16, filename: aegon_2223524_tek-prim-odemeli-5-yillik_03-basv.pdf, folder: 89233/53  
pdf is regenerating ...id:18, filename: aegon_2223524_turuncu-elma_01-ihtc.pdf, folder: 772/3  
pdf regenerated ...id:18, filename: aegon_2223524_turuncu-elma_01-ihtc.pdf, folder: 772/3  
pdf is regenerating ...id:19, filename: aegon_2223524_egitim-icin-hayat-sigortasi_06-polc.pdf, folder: 934/34  
pdf regenerated ...id:19, filename: aegon_2223524_egitim-icin-hayat-sigortasi_06-polc.pdf, folder: 934/34  
pdf is regenerating ...id:20, filename: aegon_2223524_egitim-icin-hayat-sigortasi_06-polc.pdf, folder: 9343/44  
pdf regenerated ...id:20, filename: aegon_2223524_egitim-icin-hayat-sigortasi_06-polc.pdf, folder: 9343/44
```

Hata durumunda üretilmeyen pdf ekrana yansıyacaktır, bu durumda databaseden id numarasına göre kaydın parametreleri kontrol edilmelidir.

5 Web Servis Topoloji ve Mimarisi



PDF Web Servisi, RESTful API olarak tasarlanmıştır. API dokümantasyonu Swagger formatında olup, altyapı olarak;

- PHP7
- SYMFONY FREAMWORK
- MYSQL
- DOCKER
 - NGNIX

kullanılmıştır.

Servis, gelen json formatındaki requestleri , aegon'a ait olan pdf template üzerindeki karşılıklarına yerleştirerek yeni bir pdf oluşturmaktadır. Ayrıca oluşturulan bu pdf ler, cronjob ile belirli zaman aralığında, pdf in sigorta türüne göre Moreum arşivleme sistemine eklenmektedir.