Interpol Kırmızı Bülten Üzerine Web Scraping

Kerimcan Başarır

June 5, 2023

Contents

Abstract

Proje raporu, Interpol verilerini çekmek için geliştirilen bir web scraping uygulamasını kapsamaktadır. Bu rapor, projenin amaçlarını, kullanılan teknolojileri ve yapılan çalışmanın önemini açıklamaktadır. Web scraping, web sitelerinden veri toplama sürecini otomatikleştiren bir yöntemdir ve bu proje, Interpol'ün resmi web sitesinden suçluların verilerini almayı hedeflemektedir. Bu veriler, uluslararası suçluların bilgilerini, suç türlerini, tutuklama emirlerini ve diğer önemli detayları içermektedir.

Bu projenin temel amacı, Interpol verilerini etkili bir şekilde çekmek ve bu verileri daha fazla analiz veya takip için kullanılabilir hale getirmektir. Çekilen veriler, suçla mücadele, uluslararası işbirliği ve güvenlik alanlarında önemli bilgiler sağlamaktadır. Bu nedenle, bu proje, güvenlik uzmanları, hukuk kurumları, araştırmacılar ve ilgili diğer paydaşlar için büyük bir değer taşımaktadır. Fakat bu projede Yasal hükümlülüklerin dışarısına çıkmadan Interpol'ün public olarak açtığı 160 veri çekildikten sonra **veri analizi yapılmayacaktır.**

Proje, Scrapy adlı Python tabanlı bir web scraping framework'ü kullanılarak geliştirilmiştir. Scrapy, web scraping işlemlerini kolaylaştıran birçok özelliği ve araçları sunan güçlü bir araçtır.

Bu rapor, proje yapısını, veri çekme ve işleme süreçlerini, veri depolama yöntemlerini ve kullanılan veri yapılarını ayrıntılı olarak açıklayacaktır. Ayrıca, proje sürecinde Scrapy ile Beautiful Soup'un karşılaştırılması yapılacak ve hangi durumlarda hangi aracın tercih edilebileceği incelenecektir. Bu karşılaştırma, web scraping işlemlerinin farklı yönlerini anlamak ve en etkili çözümü bulmak için değerli bir bilgi sağlayacaktır.

1 Proje Tanımı

1.1 Kullanılan Teknolojiler

1.1.1 Python

Web kazıma işlemlerini gerçekleştirmek için Scrapy'nin temel programlama dili olarak Python kullanılmıştır. Python, kullanıcı dostu bir dil olup, geniş bir ekosisteme sahiptir ve Scrapy'nin kullanımını kolaylaştırır.

1.1.2 JSON

Veri depolama formatı olarak JSON kullanılmıştır. JSON (JavaScript Object Notation), veri alışverişi için hafif, kolay okunur ve yazılabilir bir formattır. Scrapy, kazıdığı verileri JSON formatında kaydetmek için entegre özellikler sunar.

1.1.3 Scrapy

Scrapy, web kazıma işlemlerini otomatikleştirmek ve web sitelerinden veri kazımak için kullanılan ana teknoloji. Scrapy, güçlü bir API sağlayarak web sayfalarını dolaşma,

içerik indirme, veri işleme ve depolama gibi temel işlevleri sunar.

Bu proje kapsamında, yukarıda bahsedilen teknolojileri kullanarak web sitelerinden veri kazıma işlemleri gerçekleştirilecektir. Scrapy'nin sağladığı özellikler sayesinde verimli, ölçeklenebilir ve esnek bir web kazıma deneyimi elde edilecektir.

Proje gerçekleştirilirken Interpol'ün public olarak sunduğu 160 bültenin API'ı kullanarak spider oluşturuldu. Oluşturulan proje aşağıdaki özelliklere sahiptir:

- Spider adı: "interpolMain"
- İzin verilen domainler: "interpol.int"
- Başlangıç URL'leri: ["https://ws-public.interpol.int/notices/v1/red?page=1"]

Spider, web sitesindeki sayfaları dolaşarak suçlu verilerini çekmekte ve JSON formatında sunmaktadır. Veri çekme işlemi, "parse" ve " parse details" metodları kullanılarak gerçekleştirilmektedir.

1.2 Kullanılan URL'ler ve Sayfa Yapısı:

Proje, aşağıdaki URL'leri kullanarak verileri çekmektedir:

- Başlangıç URL'si: ["https://ws-public.interpol.int/notices/v1/red?page=1"]
- Detay sayfalarının URL'leri: Her suçlu için "self" bağlantısı kullanılarak elde edilmektedir.
- Örnek Detay Sayfa URL'i: ["https://ws-public.interpol.int/notices/v1/red/2023-31094"]

Web sitesinin sayfa yapısı aşağıdaki gibi olup, Spider bu yapıya göre veri çekme işlemini gerçekleştirmektedir:

- Anasayfa
  - Suçlu 1
    - Detay Sayfası
  - Suçlu 2
    - Detay Sayfası
  - ...
  - Suçlu N
    - Detay Sayfası
  - Sonraki Sayfa

Her suçlu için detay sayfasının bağlantısı alınmakta ve bu bağlantılar kullanılarak detay verileri çekilmektedir. Ayrıca, sonraki sayfaya geçmek için "next" bağlantısı kullanılmaktadır.

2 Proje Yapısı

Proje yapısı aşağıdaki bileşenleri içermektedir:

2.1 Spider (InterpolmainSpider)

Bu sınıf, Scrapy Spider sınıfından türetilmiştir ve Interpol verilerini çekmek için kullanılan ana bileşendir. Spider, "interpolMain" adını taşımakta ve "interpol.int" etki alanına izin vermektedir. Başlangıç URL'si "https://ws-public.interpol.int/notices/v1/red?page=1" olarak belirlenmiştir. Spider, *parse* ve *parse_details* adlı iki ana metoda sahiptir.

2.2 start_requests Metodu

start_requests methodu, spider'ın başlangıç noktalarını belirler ve başlangıç isteklerini oluşturur. Bu method, start_urls listesindeki her bir URL için bir istek oluşturur. Her istek için rastgele bir kullanıcı ajanı (User-Agent) seçer ve bu isteği parse metoduna yönlendirir. Bu, başlangıç URL'lerini işlemek ve veri çekmeye başlamak için bir başlangıç noktası sağlar.

```
def start_requests(self):
    for url in self.start_urls:
        user_agent = random.choice(self.USER_AGENTS)
        yield scrapy.Request(url, callback=self.parse, headers={'User-Agent': user_agent})
```

Figure 1: start_requests Metodu

2.3 parse Metodu

parse metodu, Interpol web sitesinden gelen yanıtın içeriğini işlemek için kullanılır. İlk olarak, yanıtın içeriği JSON formatında yüklenir ve içinde "_embedded" ve "notices" anahtarları varsa, bu anahtarların değerleri üzerinde dolaşılır. Her bir suçlu için _links anahtarının değerine göre işlemler yapılır.

Eğer _links içinde "self" anahtarı varsa, suçluyla ilgili detay sayfasının URL'si alınır. Ardından, rastgele bir kullanıcı ajanı (User-Agent) seçilir ve bu ajanla bir HTTP isteği yapılır. İstek sonucunda elde edilen yanıt, parse_details metoduna yönlendirilir.

Eğer _links içinde "images" anahtarı varsa, suçluyla ilgili fotoğraf sayfasının URL'si alınır. Yine rastgele bir kullanıcı ajanı seçilir ve bu ajanla bir HTTP isteği yapılır. İstek sonucunda elde edilen yanıt, parse_photo metoduna yönlendirilir.

parse_details ve parse_photo metodlarından dönen sonuçlar yield ifadesiyle geri döndürülür. Bu sayede, her bir suçlu için detay bilgileri ve fotoğrafları elde edilerek işlenebilir.

Son olarak, get_next_page metodu çağrılarak bir sonraki sayfaya geçiş yapılır. Bu metot, yanıtın içindeki "_links" anahtarının değerine bakarak "next" anahtarının URL'sini alır. Eğer bir sonraki sayfa varsa, rastgele bir kullanıcı ajanı seçilir ve bu ajanla bir HTTP isteği yapılır. İstek sonucunda elde edilen yanıt, tekrar parse metoduyla işlenir.

```
for notice in notices:
    for link_name, link in notice["_links"].items():

        if link_name == "self":
            detail_links = link["href"]
            user_agent = random.choice(self.USER_AGENTS)
            yield scrapy.Request(detail_links, callback=self.parse_details, headers={'User-Agent': user_agent})

        elif link_name == "images":
            photo_links = link["href"]
            user_agent = random.choice(self.USER_AGENTS)
            yield scrapy.Request(photo_links, callback=self.parse_photo, headers={'User-Agent': user_agent})

yield from self.get_next_page(data)
```

Figure 2: parse Metodu

2.4 get_next_page Metodu

get_next_page methodu, mevcut sayfadan sonraki sayfaya geçmek için kullanılır. Verilen veri içinde "_links" bölümünde "next" adında bir bağlantı varsa, bu bağlantıyı kullanarak bir sonraki sayfayı işlemek üzere bir istek yapar ve parse metodunu çağırır. Bu işlem, sayfaların sırayla işlenerek tüm verilerin alınmasını sağlar.

```
def get_next_page(self, data):

    # Sonraki sayfaya geçmek için URL oluşturma
    next_page_link = data["_links"].get("next", {}).get("href")

    if next_page_link:
        user_agent = random.choice(self.USER_AGENTS)

        # Sonraki sayfayı işlemek üzere istek yapar ve parse metodu çağrılır
        yield scrapy.Request(next_page_link, callback=self.parse, headers={'User-Agent': user_agent})
```

Figure 3: get_next_page Metodu

2.5 parse_photo Metodu

parse_photo metodu, Interpol web sitesinden gelen yanıtın içinden fotoğraf verilerini çekmek için kullanılır. İlk olarak, yanıtın içeriği JSON formatında yüklenir ve içinde "_embedded" ve "images" anahtarları varsa, bu anahtarların değerleri üzerinde dolaşılır. Eğer fotoğraf varsa, ilk fotoğrafın "_links" anahtarının değerinden fotoğrafın URL'si alınır. Son olarak, bu URL'ye bir HTTP isteği yapılır ve fotoğraf yanıtı alındığında save_photo metodu çağrılır.

```
def parse_photo(self, response):

    data = json.loads(response.body)

    if "_embedded" in data and "images" in data["_embedded"]:
        images = data["_embedded"]["images"]
        if images:
            photo_url = images[0]["_links"]["self"]["href"]
            user_agent = random.choice(self.USER_AGENTS)
            yield scrapy.Request(photo_url, callback=self.save_photo, headers={'User-Agent': user_agent})
```

Figure 4: parse_photo Metodu

2.6 save_photo Metodu

save_photo metodu, parse_photo metodu tarafından alınan fotoğraf yanıtını işler ve bir dosyaya kaydeder. İlk olarak, yanıtın URL'sinden dosya adı elde edilir. Ardından, bu dosya adıyla bir dosya oluşturulur ve yanıtın içeriği bu dosyaya yazılır. Bu sayede, Interpol web sitesinden çekilen fotoğraflar yerel bir dosyada kaydedilir.

```
def save_photo(self, response):
    filename = response.url.split("/")[-1] + ".jpg"
    with open(filename, "wb") as f:
        f.write(response.body)
    self.log(f"Fotoğraf kaydedildi: {filename}")
```

Figure 5: save_photo Metodu

2.7 parse_details Metodu

Bu metod, suçlu ayrıntılarının bulunduğu sayfayı işlemektedir. Yanıt verisi JSON formatında yüklenmekte ve InterpolscrapingItem nesnesi oluşturulmaktadır. Suçluyla ilgili detaylar, ilgili alanlara eklenmekte ve oluşturulan nesne yield edilmektedir.

```python
def parse_details(self, response):

    # Detay sayfasının yanıt verisini JSON formatında yükler
    data = json.loads(response.body)

    # InterpolscrapingItem nesnesi oluşturur
    item = InterpolscrapingItem()

    # Detayları ilgili alanlara ekler
    item['entity_id'] = data.get('entity_id')
    item['name'] = data.get('name')
    item['forename'] = data.get('forename')
    item['weight'] = data.get('weight')
    item['date_of_birth'] = data.get('date_of_birth')
    item['languages_spoken_ids'] = data.get('languages_spoken_ids')
    item['nationalities'] = data.get('nationalities')
    item['sex_id'] = data.get('sex_id')
    item['country_of_birth_id'] = data.get('country_of_birth_id')
    item['distinguishing_marks'] = data.get('distinguishing_marks')
    item['eyes_colors_id'] = data.get('eyes_colors_id')
    item['hairs_id'] = data.get('hairs_id')
    item['place_of_birth'] = data.get('place_of_birth')

    yield item
```

Figure 6: parse_details Metodu

2.8 InterpolscrapingItem Sınıfı

Bu sınıf, Spider tarafından çekilen verilerin saklanması için kullanılan bir veri modelidir. Interpol verilerinin çeşitli özelliklerini temsil eden alanlar içermektedir. Bu alanlar, çekilen verilerin tutulduğu ve işlendiği yerdir.

Proje yapısı, Spider ve InterpolscrapingItem arasındaki ilişkiyi sağlamaktadır. Spider, Scrapy çerçevesinin sunduğu özellikleri kullanarak verileri çekmekte ve parse_details metodunu çağırarak ayrıntıları işlemektedir. Daha sonra, parse_details metodundan elde edilen InterpolscrapingItem nesneleri, çekilen verileri temsil eden özelliklere sahip olarak saklanmaktadır. Bu yapı, Interpol verilerinin etkili bir şekilde çekilmesini ve saklanmasını sağlamaktadır.

```python
class InterpolscrapingItem(scrapy.Item):

    # Suçluyla ilgili verileri temsil eden Scrapy öğesi (item) tanımlanır

    name = scrapy.Field()
    forename = scrapy.Field()
    weight = scrapy.Field()
    date_of_birth = scrapy.Field()
    entity_id = scrapy.Field()
    languages_spoken_ids = scrapy.Field()
    nationalities = scrapy.Field()
    sex_id = scrapy.Field()
    country_of_birth_id = scrapy.Field()
    distinguishing_marks = scrapy.Field()
    eyes_colors_id = scrapy.Field()
    hairs_id = scrapy.Field()
    place_of_birth = scrapy.Field()
    charge = scrapy.Field()
```

Figure 7: InterpolscrapingItem Sınıfı

3 Sonuç

entity_id	name	forename	weight	date_of_birth	languages_spoken_ids	nationalities	sex_id	country_of_birth_id	distinguishing_marks	eyes_colors_id	hairs_id	place_of_birth
2023/30656	SALAME	RIAD TOUFIK	-	1950/07/17	ENG, ARA, FRE	FR, LB	M	LB	-	-	-	Antelias
2023/30297	LEMUS BARAHONA	ELMER MANUEL	-	1999/09/12	SPA	SV	M	SV	-	-	-	Carolina, San Miguel
2023/28202	DEMERITTE	BYRON BRADLEY	63	1993/02/23	ENG	BS	M	BS	-	BROD	BLA	NEW PROVIDENCE
2023/24847	PERLA RUBIO	PEDRO OSMAR	-	1986/03/04	SPA	SV	M	SV	-	-	-	Municipio El Carmen, departamento La Unión
2023/26387	RUBIO PERLA	JOSE ROGELIO MISAEL	-	1980/10/16	SPA	SV	M	SV	-	-	-	Municipio el Carmen, departamento La Unión
2023/20396	MAGOMADOV	MOVSAR ZOUBAIROVITCH	-	1991/06/22	SPA, FRE, RUS	RU	M	RU	bald	BLUL	-	-
2023/27985	MARTÍNEZ RODRÍGUEZ	EDGARDO	-	1990/03/29	SPA	SV	M	SV	-	-	-	municipio de Santa Tecla, departamento de La Libertad
2023/27683	MACZ GUERRA	JOSE EDUARDO	-	1990/09/28	SPA	GT	M	GT	-	BROD	BLA	Guatemala
2023/30311	SILVA BRAVO	VERONICA MARIA	85	1973/04/04	ENG, SPA	NI	F	NI	-	BRO	BLA	Managua
2023/13927	TURASHEV	IGOR	81	1981/06/15	-	RU	M	RU	-	BRO	BRO	Yoshkar-Ola

Table 1: Veri Tablosu

(*Raporda, bulunan suçlu bilgileri ve fotoğrafları, kamuoyuyla paylaşılmış ve herkesin erişebileceği bir platform olan Interpol'ün resmi internet sitesi interpol.int üzerinde yayınlanmıştır. Bu veriler, Interpol tarafından sağlanan güvenilir ve resmi kaynaklardan derlenmiştir.*)

Proje kapsamında, toplamda 160 suçlu verisi çekilmiştir. Her suçlu verisi, 13 farklı özellikten oluşmaktadır. Bu özellikler arasında suçlu kimliği (*entity_id*), isim (*name*), soyisim (*forename*), kilo (*weight*), doğum tarihi (*date_of_birth*), konuşulan dillerin kimlikleri (*languages_spoken_ids*), uyruklar (*nationalities*), cinsiyet kimliği (*sex_id*), doğum yeri ülke kimliği (*country_of_birth_id*), ayırt edici işaretler (*distinguishing_marks*), göz rengi kimliği (*eyes_colors_id*), saç rengi kimliği (*hairs_id*) ve doğum yeri (*place_of_birth*) bulunmaktadır.

Projenin sonuçları değerlendirildiğinde, Interpol web sitesinden başarıyla suçlu verilerinin çekildiği görülmektedir. Bu veriler, uluslararası suçluların profil bilgilerini içermekte ve güvenlik birimlerinin suçluları tanımlamak ve izlemek için kullanabilecekleri değerli bilgiler sunmaktadır.

Figure 8: Örnek fotoğraflar

Bu proje, güvenlik birimlerine ve kolluk kuvvetlerine uluslararası suçluların izlenmesi ve yakalanması konusunda yardımcı olabilecek bir kaynak sağlamaktadır. Elde edilen veriler, suçlu profillerinin oluşturulması, arama çalışmalarının yönlendirilmesi ve potansiyel risklerin belirlenmesi gibi amaçlarla kullanılabilir.

4 Projenin Geleceği

4.1 Veri Tabanına Aktarım ve Düzenleme

İkinci aşamada, projede toplanan tüm verilerin veri tabanına aktarılması ve uygun bir şekilde düzenlenmesi hedeflenmektedir. Veri tabanı, suçlu detaylarının ve görsellerinin güvenli bir şekilde depolanması ve yönetilmesi için kullanılacaktır. Verilerin doğru bir şekilde aktarılması, veri tabanının etkin kullanımını sağlayacak ve suçlu verilerinin daha kolay erişilebilir olmasını sağlayacaktır. Veri tabanındaki düzenleme işlemleri, verilerin tutarlılığını ve doğruluğunu koruyacak şekilde gerçekleştirilecektir.

(*Bu aşamada veritabanı işlemleri için Psycopg2 Kullanılacaktır. Psycopg2, Python programlama dili için geliştirilmiş PostgreSQL veritabanı bağdaştırıcısıdır.*)

Bu sonraki aşamalar, projenin kapsamını genişletecek ve Interpol suçlu veri tabanının daha işlevsel hale getirilmesini sağlayacaktır. Görsel verilerin çekilmesi, suçluların tanımlanmasını ve teşhisini kolaylaştıracak, veri tabanına aktarım ve düzenleme ise verilerin daha iyi yönetilmesini ve erişilebilirliğini sağlayacaktır.