**API** Nedir?

**API (Application Programming Interface)** bir uygulamaya ait yeteneklerin, başka bir uygulama kullanılabilmesi için, yeteneklerini paylaşan uygulamanın sağladığı arayüzdür. Basitçe A uygulamasının özelliklerini B uygulamasında kullanabilmeyi sağlayan yapı API olarak adlandırılır.

API kullanımın asıl amacı bir uygulamanın bütün veya bazı metotlarını diğer uygulamalara kullanıma açarak uzaktan gelecek veri ve bilgi taleplerini kolayca ve hızlıca karşılamaktadır.

**Repository Pattern** Nedir?

Veri merkezli (veri merkezli olmak zorunda değil genellikle) uygulamalarda veriye erişimin ve yönetimin tek noktaya indirilmesini sağlayan bir tasarım desenidir. Bu patternde sürekli olarak kullanacağımız bilgileri içeren veya işleyen bir metod kuruludur, bu sayede sürekli olarak pattern içerisindeki kodla uğraşmak zorunda kalmayız.

**Web Scraping** Nedir?

Web scraping işlemi botları kullanarak bir websitesinden bilgi çekme işlemidir.

**SQL** **Joins**

İki farklı kaydı (tablo vs) birleştirdiğimizde SQL metodunu kullanarak farklı biçimlerde tablo verilerini ele alabiliriz, iki tablodaki ortak değerler işlenecekse bu inner join, soldaki kayıtlar gelsin sağ tablodakiler gelmesin istersek left join (buna müşteri-sipariş ilişkisi örnek verilebilir, müşteri sol tabloyken siparişler sağ tablodur), sağdaki veri tablosu alınacaksa right join, verilerin tamamı alındığı takdirde ise full outer join (birbiriyle ilişkili/ilişkisiz tüm veriler) olarak adlandırılır.

**@staticmethod**

Bu metot türü dışarıdan değişken ve fonksiyon değeri alamaz, yalnızca kendi içerisindeki verileri kullanabilir. Bu yöntemi kullanmak self argüman kullanımını ortadan kaldırdığı gibi bellek kullanımını da azaltır.

**Statik ve Dinamik Diller** Değişken (Variable) Yapısı

Statik tipteki dillerde bir değişkene string değer atadığımızda bu değeri başka bir str, float, int vs değerlerine dönüştürdüğümüz takdirde bu yapıdaki diller error verecektir.

Halbuki dinamik tipteki dillerde bu değişkenler, aynı olan ilk değişken üzerinde işlemlerini tamamladıktan sonra sıra diğer aynı değişkene gelince rahatlıkla onun üzerinden işlemlerini devam ettirip ilerleyebilirler.

Statik dillerin avantajı belirlediğimiz değişken türü ve değişkenin tek seferlik kullanımı sayesinde o değişken sistemde daha az yer kaplayarak sistemin daha hızlı çalışmasını sağlar.

**Strongly-Typed** **Dil** Yapısı

Strongly-typed diller bir değişkenin aynı anda birden fazla yapıdaki değişken tiplerini alamadığı dilleri oluşturur. Örneğin bir değişken string yapıdan oluşuyor, biz bu yapıya bir integer eklemeye kalkarsak bu yapıdaki sistemler error verecektir.

**Interpreted ve Compiled Diller** Arasındaki Fark

Interpreted dillerde yazılan kod, bir aracı yardımıyla bilgisayar diline dönüştürülerek bilgisayar tarafından işlem yaptırılır (yani arada bir aracı vardır, interpreter). Compiled dillerde ise bir aracı olmadan kod yazıldığı şekilde bilgisayar diline çevirilir. Interpreted dillerin avantajı kod oluşturması kolay fakat executing işlemi yavaştır. Compiled dillerde ise kod oluşturması daha zor fakat executing işlemi hızlıdır.

**Repository** Nedir

Repository, bilgi birikiminin organize bir şekilde tutulduğu ve manüpile edilebildiği bir Merkez depodur. Bu depo genellikle bilgisayar belleği halinde vuku bulur.

**Aggregation 🡪** Kümelenme (topluluk), genellikle bilgi kümeleri için kullanılır.