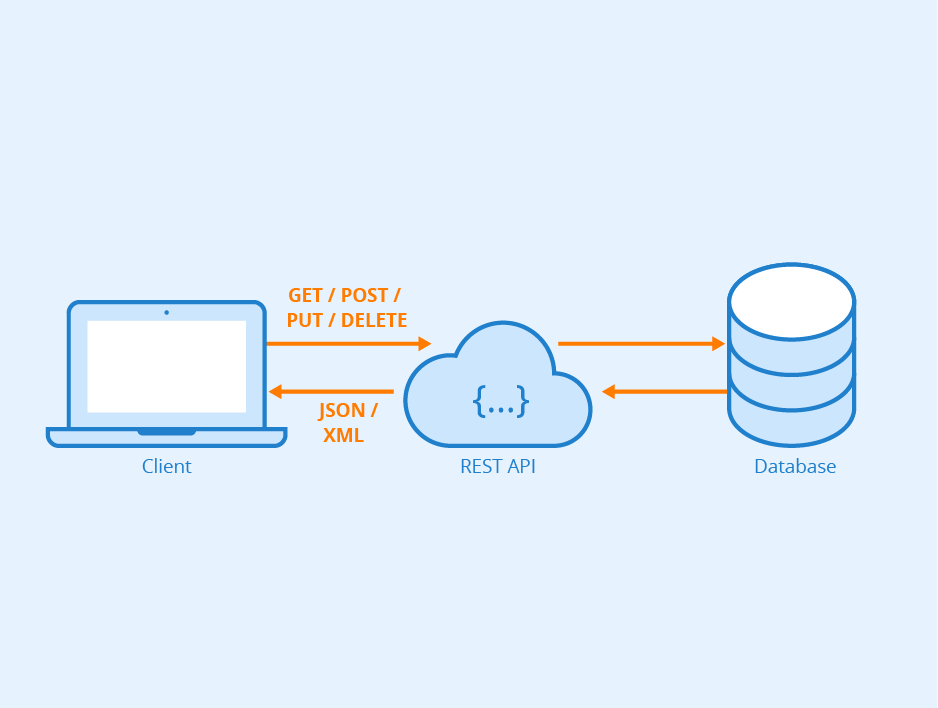
Selamun aleyküm konumuz basit ve yetkilendirmeli bir API hazırlamaktır. Konu içerisinde PDO, json, header ve koşullandırmalar kullanacağız.

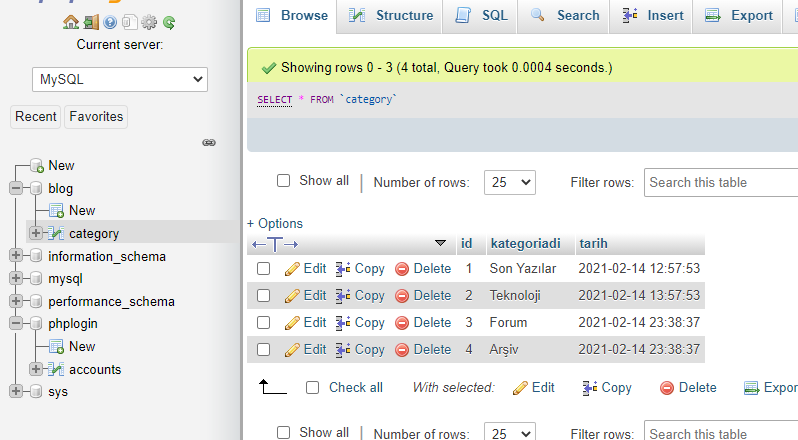
Öncelikle API nedir ve neye hizmet eder?



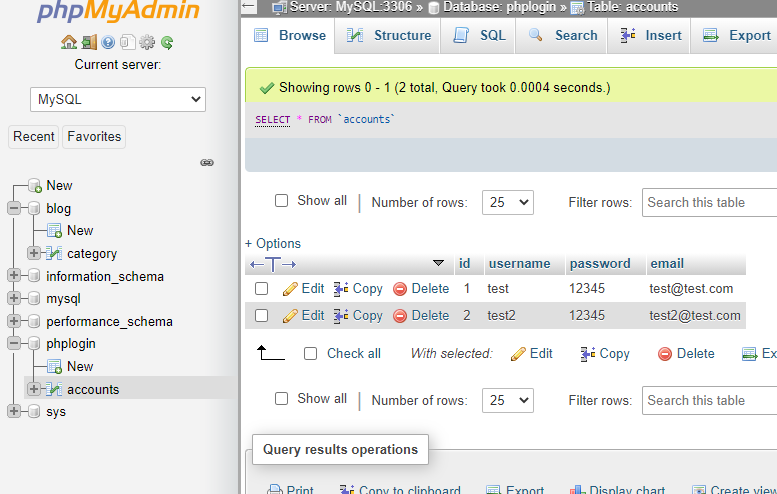
API (application programming interface) kullanıcıya programcının dizayn ettiği şekilde veritabanına istekler göndermesini sağlar. Programcı kullanıcıya API üzerinde kısıtlamalar koyarak ne tür verilere erişebileceğine karar verebilir. Burada ki asıl amaç kullanıcımızın rahat bir şekilde verileri çekip işleyebilmesidir bu yüzden de genelde veriler XML ya da JSON formatında döndürülür.Yapacağımız örnekte JSON kullanacağız ama XML’ye adapte etmek daha zor değildir. Şemada Client REST API ve Database görüyorsunuz aynı sistemi lokal bilgisayarda test edip temel yapıyı bildiğim kadar aktaracağım.

**Veri Tabanı kurulumu:**

Veri tabanı olarak mysql seçtim bağlantı şekli olarakta PDO sınıfını kullanacağız. Öncelikle lokal serverimizi başlatıp tarayıcıdan phpmyadmin’e girelim.



Sol menüde görüldüğü gibi blog adında bir veritabanı onun altında category isminde bir tablo oluşturduk. Elimizde bir blog sitesi olduğunu hayal edip telefon için uygulama yapacakmış gibi senaryo düşünebilirsiniz. Deneme amaçlı kayıtlar girdik. Yine sol tarafta phplogin adında veritabanı açıp accounts adında tablo oluşturalım. Bu da API iznini vereceğimiz kullanıcılar olacak.

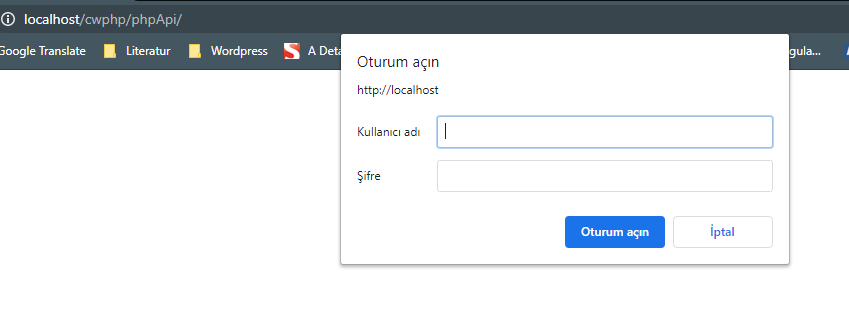


2 adet test kullanıcısı hazırladım bunları paylaşağımız kişi sayısına göre çoğaltabiliriz.

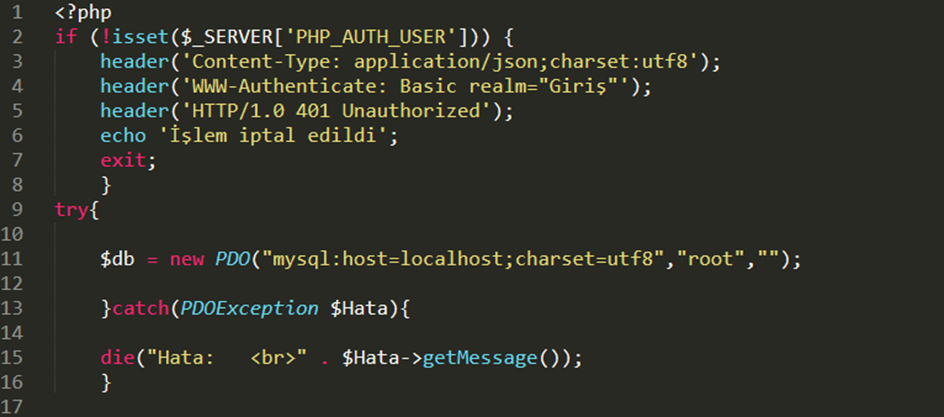
Şimdi veritabanı tamam kod kısmına geçelim. Bir tane php dosyası oluşturalım lokalde olduğumuz için /www/.. gibi bir dosya dizininde olmalı.



Şimdi sitemiz tarayıcıda ilk açıldığında $\_SERVER değişkeni belli ön tanımlı parametrelerle gelir. PHP\_AUTH\_USER dizinimizde önceden giriş yapmış kullanıcı var mı yok mu değer olarak tutar isset() fonksiyonu ilede bize kullanıcı yoksa bile null hatası verme demiş oluyoruz. Tabiki ilk girişimizde kullanıcı tanımlı olmadığı için koşulumuz çalışacak ve bizden Basic Auth isteyecek , eğer iptal der isek İşlem iptal edildi deyip exit ile kullanıcı ile ilişkimiz kesilecek. Eğer kullanıcı tarayıcıyı kapatmaz ise doğal olarak değişkenimiz $\_SERVER içinde olmaya devam edecek ve sormayacak tekrar.

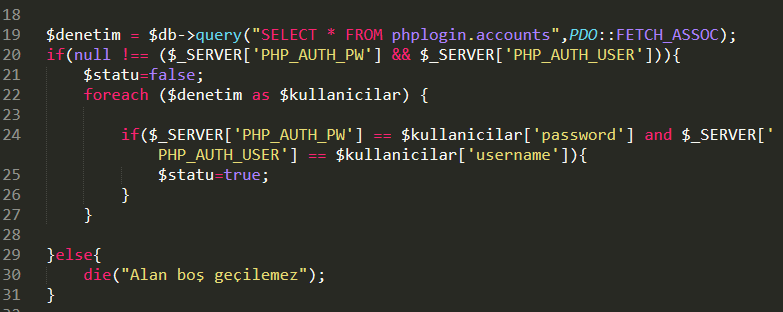


Headerımız çalıştığına göre şimdi mysql bağlantısı oluşturalım. PDO kullanacağımızdan bahsetmiştik. Driver olarak mysql’i seçtik daha sonra host charset kullanıcı adı ve şifreyi girdik.



Bağlantıda bir hata oluşması ihtimaline karşı try catch kullandık. Bağlantı sırasında değişik hatalar olabilir biz PDOException dedik ve bunu $Hata değişkenine adres gösterdik yani herhangi bir sıkıntıda bu $Hata değişkeninde tutulacak. Hatayı die() fonksiyonu içinde gösterip yine die() ile sonraki kodların çalışmasını engelledik.

Daha önceden oluşturduğumuz veritabanına query() ile sorgu yapıp $denetim olarak depoladık. Bu kullanıcı bilgilerini doğrulamak için kullanılacaktır. O yüzden benim yaptığım gibi yapmayın 😊 Kullanıcıdan aldığınız verileri filtreleyin.

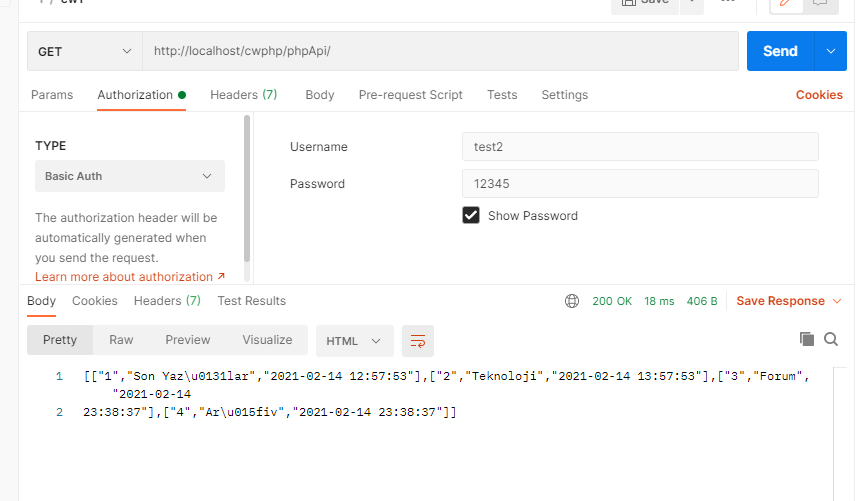


PHP\_AUTH\_PW kullanıcının girdiği şifre değerini taşıyor PHP\_AUTH\_USER da kullanıcı adını tutuyor. Eğer bu iki değerde boş değilse veritabanında değerleri eşleştirmeye başlıyor herhangi bir durumda sağlıyorsa $statu değişkenimizi true ya çeviriyoruz.Bu değer aşağıda işimize yarayacak.



Kullanıcının artık sistemde kayıtlı olduğunu öğrendiğimize göre ona gerekli bilgileri hazırlayıp verelim. Eğer statu true ise yani kullanıcı veritabanında eşlendiyse yeni bir veritabanı sorgusu çalışacak bu sefer blog.category üzerinden blog bilgilerini çekecek bunları liste adındaki arraye atayıp daha sonra json\_encode() ile json formatı olmaya zorlucaz.

Son olarak bir deneme yapıp konuyu kapatalım.



Görüldüğü gibi veritabanına eklediğimiz blog bilgileri test 12345 için çalıştı. Burdan sonrası kullanıcının get methodu ile telefon uygulamasından istek yollaması ve gelen verileri işlemesi. Eğer bilgilerinizin güvende kalmasını istiyorsanız kod içerisinde filtreleme validation için farklı yollar izlenebilir. Kullanıcıya özel Token üretilip onun üzerinden session yapılabilir bunlar araştırırken gördüğüm bazı yöntemler.

