**1. Promise Nedir?**

Promise, JavaScript’te bir işlemin sonucunu gelecekte temsil eden bir nesnedir. Asenkron işlemler için kullanılır ve üç farklı durumu vardır:

* **Pending (Beklemede)**: İşlem hala devam ediyor.
* **Fulfilled (Tamamlandı)**: İşlem başarıyla sonuçlandı.
* **Rejected (Reddedildi)**: İşlem başarısız oldu.

Kullanımı şu şekildedir:

A computer screen shot of text

Description automatically generated

### 2. ****Async/Await Nedir?****

Async/Await, JavaScript'teki asenkron işlemleri senkron kod gibi yazmamızı sağlayan bir yapıdır. async bir fonksiyon, her zaman bir Promise döner. await ise bir promise'in sonucunu beklemek için kullanılır. Bu, kodun okunabilirliğini artırır ve "callback hell" sorununu çözmeye yardımcı olur.

A computer screen shot of a program code

Description automatically generated

**Avantajları:**

* Kod daha okunabilir ve takip edilmesi kolay olur.
* Promise zincirlemeye kıyasla hata yakalama (try-catch) daha basit olur.
* Asenkron işlemler senkron görünümlü hale getirilir, böylece karmaşıklık azalır.

**3. Callback Hell Nedir?**

Callback hell, iç içe geçmiş callback fonksiyonlarının çok fazla olması durumudur ve bu durum kodun okunabilirliğini zorlaştırır. Genellikle asenkron işlemler gerçekleştirilirken, işlemlerin sırasıyla yapılması gerektiğinde bu sorun ortaya çıkar.

**Callback Hell Örneği:**

A screenshot of a computer program

Description automatically generated

**Çözüm:** Promise ve async/await yapıları kullanılarak callback hell problemi çözülebilir.

### 4. ****Fetch API Nedir?****

fetch API, modern tarayıcılar tarafından sağlanan bir yöntemdir ve ağ istekleri yapmak için kullanılır. Bu API, asenkron HTTP istekleri yapar ve genellikle veri almak veya göndermek için kullanılır.

A blue screen with green text

Description automatically generated

**Eksik Yanları:**

* fetch sadece başarısız HTTP isteklerini catch bloğuna atar, ancak 404 veya 500 gibi hatalı HTTP durum kodlarında bile Promise resolve olur.
* Gelişmiş konfigürasyon ve hata yönetimi sınırlıdır.
* Zaman aşımı (timeout) desteği yerleşik olarak yoktur.

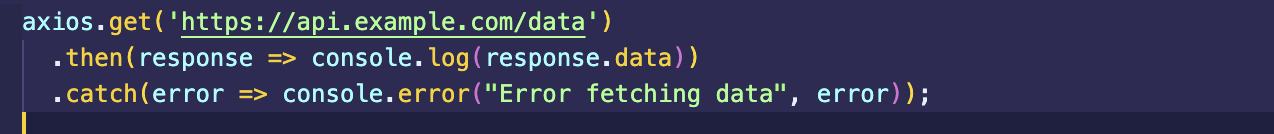
**5. Axios Nedir?**

Axios, HTTP istekleri yapmak için kullanılan popüler bir kütüphanedir. Promise tabanlıdır ve fetch API'ye göre daha zengin özelliklere sahiptir.

**Fetch ve Axios Arasındaki Farklar:**

* Axios, istek ve yanıtları otomatik olarak JSON formatına çevirir.
* Zaman aşımı, otomatik hata işleme ve istek iptali gibi gelişmiş özellikler sunar.
* HTTP hatalarını daha iyi yönetir (örneğin, 404 ve 500 gibi durum kodlarını yakalar).

**Axios Kullanımı:**



### 6. ****Singleton Axios Instance Nasıl Oluşturulur?****

Tek bir Axios örneği oluşturmak, özellikle aynı ayarları tekrar tekrar uygulamamak için yararlıdır.

**Örnek Singleton Axios Instance:**

A screenshot of a computer program

Description automatically generated

Bu instance'ı projedeki diğer dosyalarda kullanarak aynı yapılandırmayı tekrar tekrar tanımlamak zorunda kalmazsınız.

**7. Axios Interceptor’ları Ne İşe Yarar?**

Axios interceptors, her HTTP isteğini veya yanıtını işlemden önce veya sonra yakalamak için kullanılır. Örneğin, global hata işleme veya isteğe özel bir header eklemek için interceptors kullanılabilir.

**İki Tür Interceptor Vardır:**

1. **Request Interceptor:** İstek yapılmadan önce bir işlem yapmak için kullanılır.
2. **Response Interceptor:** Yanıt alınmadan önce bir işlem yapmak için kullanılır.

**Kullanımı:**

A computer screen shot of a program code

Description automatically generated

Bu interceptors, tüm uygulama genelinde HTTP isteklerini ve yanıtlarını işlemek için güçlü bir yol sağlar.

**React Native'de Axios Kullanımı**

React Native projelerinde Axios, backend API'leriyle iletişim kurmak için yaygın olarak kullanılır. Yukarıdaki örneklerde olduğu gibi kullanabilir ve Redux ile entegre edebilirsiniz.

Özetle:

* Promise yapısı asenkron işlemleri yönetir.
* async/await, kodu daha okunabilir hale getirir ve callback hell sorununu çözer.
* fetch ve Axios asenkron HTTP istekleri için kullanılır, ancak Axios daha fazla özellik sunar.
* Axios interceptors global istek ve yanıt yönetimini sağlar, isteklere dinamik olarak müdahale edilebilir.