**BiSU Teknik Test Dokümantasyonu**

**Sorular**

1. Callback cehennemi nedir, bundan kurtulmak için kullanmayı tercihe ettiğin yöntemi kısaca anlatır mısın?
2. NodeJs ile yazılmış bir sunucu, single thread çalıştığına göre sisteme aynı anda yoğun istek gelen durumlarla nasıl başa çıkıyor?
3. NodeJs ve Express framework kullanılırken global değişken kullanmamız gerektiğinde nasıl ilerlenir?
4. Aşağıdaki kod şu anda çalışmıyor, nasıl çalışır hale getirebiliriz?

#!/usr/lib/node/v8/bin/node

let parent = {

func2: function() {

return new Promise(function (resolve, reject) {

return resolve(1);

});

},

func1: function() {

return new Promise(function (resolve, reject) {

this.func2().then( (count) => {

console.log(count + 1);

});

});

}

}

parent.func1();

**Ödev**

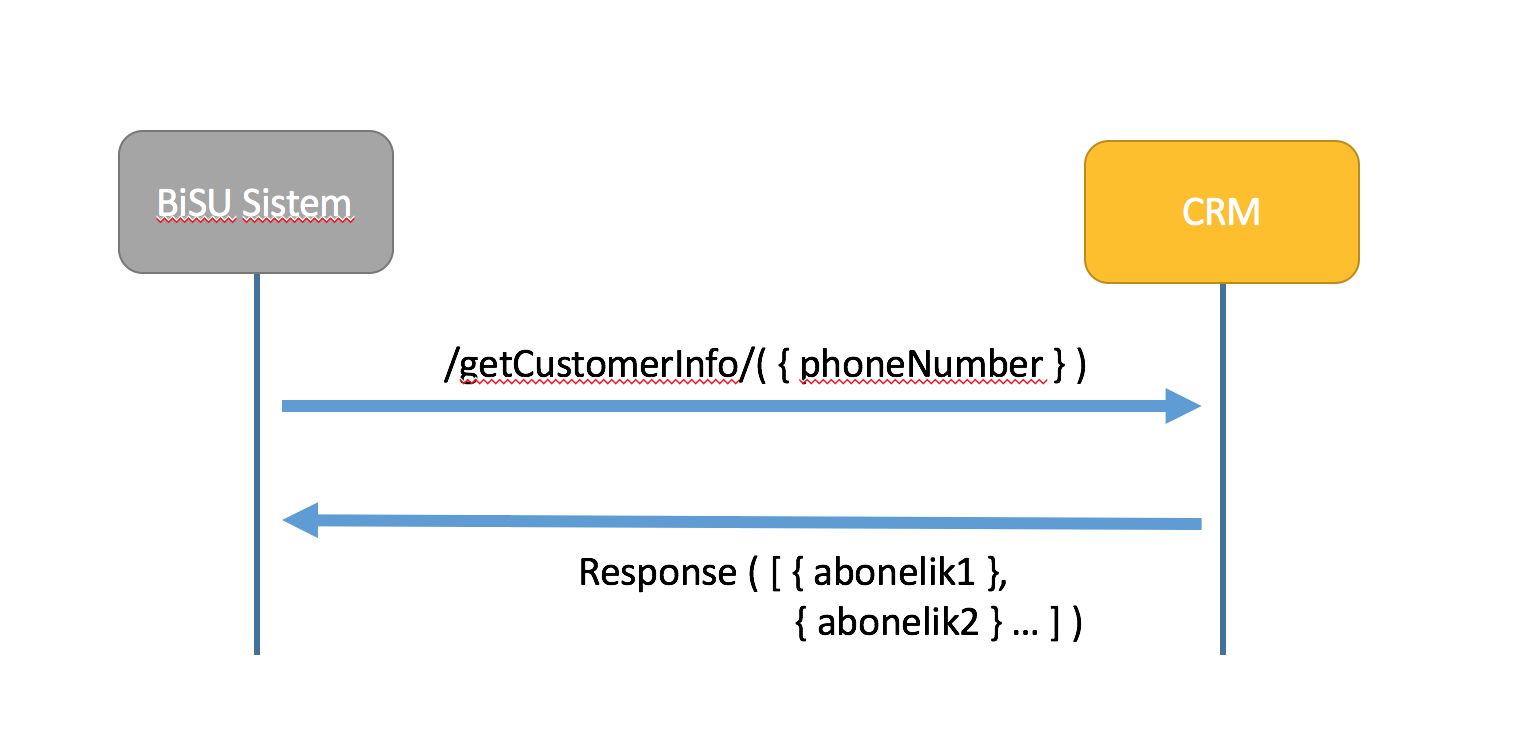
BiSU’da kaydı olan bir aboneliğin müşteri bilgilerinin ve geçmiş sipariş bilgilerinin çekilip, bir Web uygulaması üzerinden bunların görüntülenebileceği bir proje yapmanı istiyoruz. Bunun için lokal bir MySQL database oluşturup, datayı çekebilecek bir Node.js servisi ve Angular.js tabanlı bir Web Uygulaması geliştirmen gerekmekte.

**Müşteri Kaydı**

BiSU’da müşteri bilgisi Subscription tablosunda tutulacak. Örnek Subscription datası aşağıdaki gibi olmalı.

**BACKEND**

**Müşteri Bilgisi Sorgulama**



Telefon numarası ile müşteri bilgilerini öğrenmek için kullanılacak çağrıdır. Servis, telefon numarasına tanımlı tüm abonelikler/adresleri **liste(array)** halinde dönmelidir.

*Örnek İstek*

|  |  |
| --- | --- |
| **Method** | **URL** |
| **GET** | api/getCustomerInfo |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET | phoneNumber | string |

**phoneNumber**

Müşterinin abonelik oluşturma sırasında kullandığı telefon numarası.

*Örnek Cevap*

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | [  // 1. abonelik  {     “subscriptionId”: [string],  “fullname”: [string],     “address”: [string],     “locationName”: [string],     “subCityName”: [string],     “cityName”: [string],     “brand”: [string],     “phoneNumber”: [string],     “distributorNumber”: [string]  },  // 2. Abonelik  ...  ] |
| 403 | {"message":"Belirtilen telefona ait kayıt bulunamamıştır."} |
| 500 | {"message":"Sunucu hatası. Lütfen tekrar deneyin."} |

**fullName:** Müşteri ismi

**subscriptionid**: Müşteri abonelik unique id’si (diğer servis isteklerinde kullanılacak)

**address**: Abonelik adresi

**locationName**: Abonelik semti

**subCityName**: Abonelik ilçesi

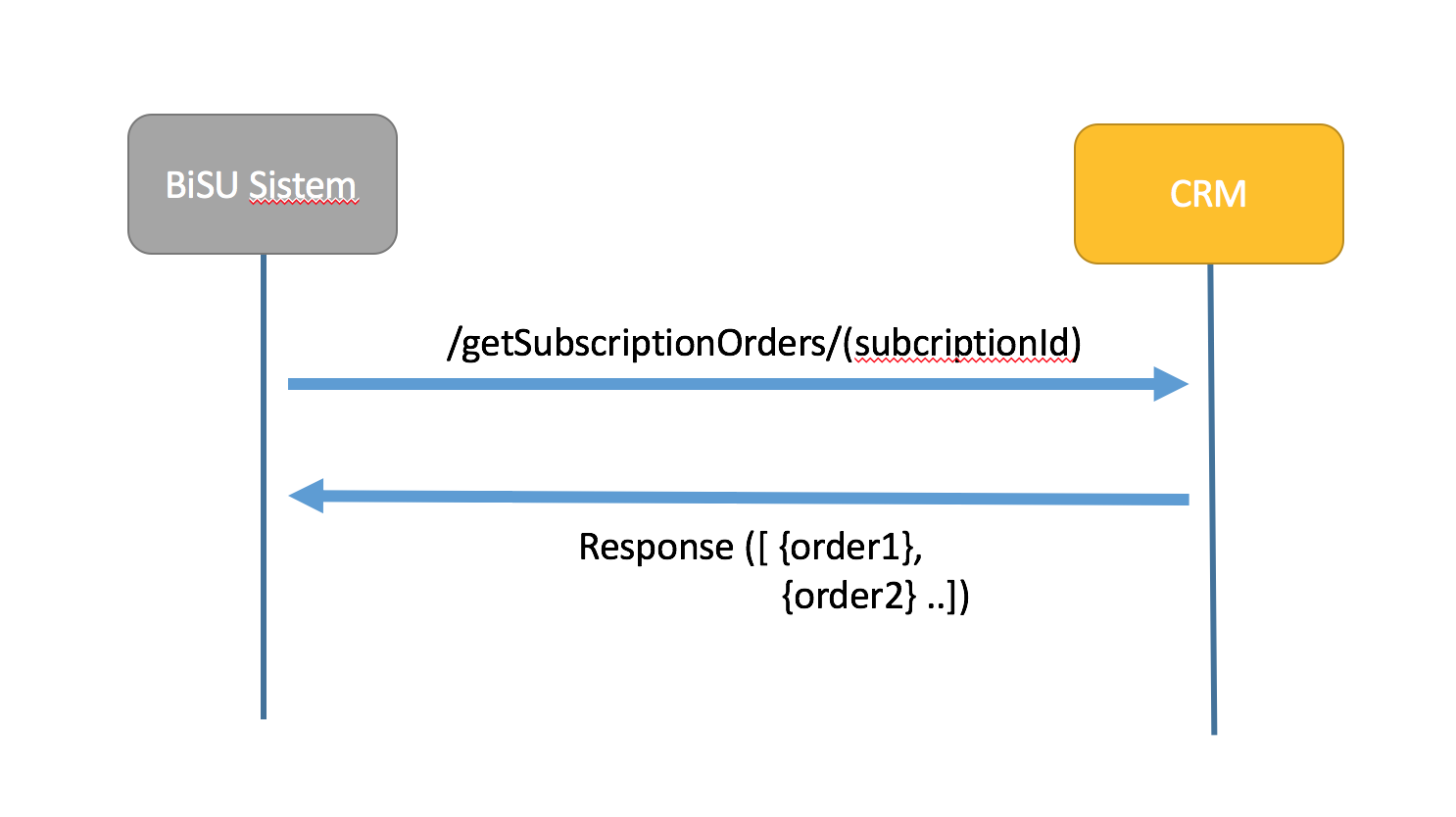
**cityName**: Abonelik ili

**brand**: Abonelik marka ismi

**phoneNumber**: Aboneliğin kaydolduğu telefon numarası

**distributorNumber**: Aboneliğin ilişkilendirildiği bayi numarası

**Sipariş Bilgisi Sorgulama**

****

Abonelik numarası ile müşterinin o aboneliğinin geçmiş siparişlerinin alınacağı çağrıdır. Servis, tüm siparişleri **liste(array)** halinde dönmelidir.

*Örnek İstek*

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/getSubscriptionOrders |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| POST | subscriptionId | string |

**subscriptionid**: Müşteri abonelik unique id’si

*Örnek Cevap*

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | [  // 1. order  {     “subscriptionId”: [string],  “orderId”: [string],     “deliveryDate”: [string],     “paymentMethod”: [string],     “products”: [  { “productName”:[string],  “quantity”:[number]  }  …  ],     “totalAmount”: [string],     “status”: [string]  },  // 2. order  ...  ] |
| 403 | {"message":"Belirtilen telefona ait kayıt bulunamamıştır."} |
| 500 | {"message":"Sunucu hatası. Lütfen tekrar deneyin."} |

**orderId:** sipariş referans numarası.

**subscriptionId:** Müşteri abonelik IDsi

**products:** Sipariş sepetindeki ürün bilgilerini içeren array

* **productId:** Database’de tanımlı ürün IDsi.
* **quantity:** Adet

**paymentMethod:** Ödeme tipi. [Kapıda, Kredi Kartlı, BKM, Hopi]

**deliveryDate:** teslimat tarihi, örnek tarih formatı **‘YYYY-MM-DD HH:mm:ss’**

**totalAmount:** Toplam sipariş ücreti.

**status:** Sipariş durumu [yeni, iletildi, teslim edildi, iptal]

**DATA**

Database için aşağıdaki örnek data kullanılabilir. MySQL ile bu tabloların oluşturup, Backend servislerinde SQL kullanılarak dataya erişim sağlanmalıdır.

**Örnek data**

Subscription

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| subscriptionId | fullname | address | locationName | subCityName | cityName | brand | phoneNumber | distributorNumber |
| abc123 | Utku | sair nefi sokak | caferaga | kadiköy | istanbul | hayat | 5332858530 | 2161000000 |
| abc124 | Utku | sair nefi sokak | caferaga | kadiköy | istanbul | sirma | 5332858530 | 2161000004 |
| abc125 | Ozan | bahariye sokak | caddebostan | kadiköy | istanbul | erikli | 5331533630 | 2161000001 |
| abc126 | Ergin | moda caddesi | göztepe | kadiköy | istanbul | sirma | 5332858530 | 2161000002 |

Orders

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| orderId | subscriptionId | deliveryDate | paymentMethod | products | totalAmount | status |
| 1 | abc123 | 2017-05-02 00:13 | BKM | [ {"product":"19 lt damanaca", "quantity":1 }] | 10.00 | NEW |
| 2 | abc123 | 2017-04-23 20:05 | MASTERPASS | [ {"product":"19 lt damanaca", "quantity":1 }, {"product":"10 lt pet", "quantity":2 }] | 24.00 | CONFIRMED |
| 3 | abc123 | 2017-03-31 16:35 | PAYATDOOR | [ {"product":"19 lt damanaca", "quantity":2 }] | 12.00 | CANCELED |
| 4 | abc125 | 2017-03-29 11:19 | PAYATDOOR | [ {"product":"19 lt damanaca", "quantity":3 } ] | 36.00 | COMPLETED |
| 5 | abc125 | 2017-03-23 17:59 | BKM | [ {"product":"5 lt pet", "quantity":2 }] | 15.00 | CONFIRMED |
| 6 | abc126 | 2017-03-09 19:02 | MASTERPASS | [ {"product":"19 lt damanaca", "quantity":1 }, {"product":"5 lt pet", "quantity":3 }] | 32.50 | CANCELED |

**Web Uygulaması**

Servisler aracılığıyla ulaşılan datanın Angular.js tabanlı bir uygulama üzerinden görüntülenmeli. Web arabirimi UX/UI tasarımı çok önemli olmamakla beraber, responsive olmalı.