

RENKLİ REÇETE PROJESİ

HBYS ve AHBS Entegrasyon Kılavuzu

İçindekiler

.....	1
Amaç.....	3
Yetkilendirme	3
Entegrasyon Adımları	3
Token Oluşturma Servisi.....	3
Harici Giriş Yönlendirmesi	5
Reçete Liste Servisi	6
Örnek İstemci Kodu	7

Amaç

Elektronik Renkli Reçete Uygulamasının amacı, hali hazırda matbu olarak basılan tüm renkli reçetelerin (kırmızı, yeşil, turuncu ve mor) elektronik ortama aktarılmasını sağlamaktır. Bu dokümanın amacı ise geliştirilen Renkli Reçete Uygulamasına Hbys ve Ahbs'lerin entegre olmasını sağlayacak web api'lerin kullanımının anlatılmasıdır.

Yetkilendirme

Renkli Reçete Uygulamasına entegre olacak Hbys ve Ahbs firmalarının ilk olarak kuruma başvurarak akredite olmaları gerekmektedir. Kurum tarafından firmalara iletilecek kullanıcı adı ve şifre bilgileri web api'lere erişim için zorunludur.

Servislerin test ortamında kullanılabilmesi için aşağıdaki bilgiler tanımlanmıştır. Test sürecinde tüm firmalar bu veriler üstünden işlem yapabilirler.

Parametre Adı	Değeri
Firma Kullanıcı Adı	11900012016
Firma Şifre	123456
Tesis Kodu	11000001 (TEST HASTANESİ)
Doktor Tc	11111111111

Entegrasyon Adımları

Renkli Reçete Uygulaması devreye alındıktan sonra Hbys ve Ahbs ekranlarından Kırmızı veya Yeşil Reçeteli ilaçların yazılması engellenecektir. Bu ekranlarda renkli reçeteli ilaç yazılmak istendiğinde hekimler otomatik olarak Renkli Reçete uygulamasına yönlendirilecektir. Renkli Reçete Uygulaması web tabanlı bir uygulama olması nedeniyle yönlendirme sırasında kullanıcının tarayıcı olarak Google Chrome kullanması sağlanmalıdır.

Yönlendirme işlemi için ilk olarak kullanıcı adı, şifre, tesis kodu ve doktor TC Kimlik numarası parametreleri ile token oluşturma servisi çağrılmalıdır. Bu servisten alınacak token, normal şartlarda elektronik imza ile sisteme girmesi gereken doktorların elektronik imzaya gerek kalmadan sisteme girmesine olanak sağlayacaktır.

Entegrasyon için kullanılacak servisler ve bunlara ait açıklamalar aşağıdadır.

Token Oluşturma Servisi

İlk adım olarak sisteme istekte bulunacak firma bilgilerinin, tesis ve doktor bilgileri ile beraber doğrulanması gerekmektedir. Bu doğrulama sırasında firmaya iletilmiş kullanıcı adı ve şifre bilgisinin doğruluğu, firma ile tesis bilgisinin eşleşip eşleşmediği, gönderilen tc kimlik numarasına sahip doktor bilgilerinin doktor bilgi bankasında olup olmadığı kontrol edilmektedir.

Eğer tüm adımlar başarılı şekilde doğrulanırsa sistem bir token oluşturur, response header kısmında bu token döndürülür. Aksi durumda servisten alınacak hata mesajlarına ve açıklamalarına tablodan ulaşabilirsiniz.

SONUÇ KODU	SONUÇ AÇIKLAMA
0	İşlem başarılı
1001	HBYS kullanıcı adı/şifre hatalı

1002	HBYS - Tesis bilgileri uyuşmuyor
1003	Doktor - Tesis bilgileri doktor bilgi bankasından alınan bilgilerle uyuşmuyor
1004	Doktor kaydı yok

Sistem tarafından üretilen tokenların geçerlilik süresi 1 saattir. Ancak üretilen tokenlar tek kullanımlıktır. Bu token kullanılarak sisteme giriş yapıldıktan sonra aynı token ile işlem yapılamayacaktır.

Aşağıda örnek bir request yer almaktadır. Header bilgisine ve method'a dikkat edilmesi gerekmektedir. Ayrıca firmaya tanımlanan şifrenin base64 hali gönderilmelidir.

Api adresi : <https://renklirecete.saglik.gov.tr/api/receteapi/tokenolustur>

Header Bilgisi : Content-Type:application/json

Method: POST

Örnek Data :

{DoktorTc: "11111111111", TesisKodu: "11000001", Sifre:" MTİzNDU2", KullaniciAdi: "11900012016"}

URL:

Headers:
Name: Value:
value '##' to delete

name	value
Content-Type	application/json

Content Body:

Response:
status: 200 OK

- Pragma: no-cache
- Date: Thu, 17 Nov 2016 11:47:51 GMT
- Server: Microsoft-IIS/8.5
- X-AspNet-Version: 4.0.30319
- X-Powered-By: ASP.NET
- Content-Type: application/json; charset=utf-8
- Access-Control-Expose-Headers: Token,TokenExpiry
- Cache-Control: no-cache
- TokenExpiry: 18.11.2016 14:47:51
- Content-Length: 55
- Token: 7yx5j98O1iiUskImPoBtRIDHn7mAQKFyMTE5MDAwMTlwMDcxMTE5MTE5MTQ2MDk=
- Expires: -1

{"SonucKodu": "0", "SonucAciklama": "İşlem başarılı"}

Harici Giriş Yönlendirmesi

Token oluşturma işlemi başarıyla tamamlandıktan sonra hekimlerin Renkli Reçete Uygulamasına yönlendirilmesi için bu sayfa kullanılacaktır. Normal şartlarda elektronik imzaları ile sisteme giriş yapabilen hekimler, bu yönlendirme sayesinde tekrar bir giriş ekranı ile karşılaşmadan reçete yazabileceklerdir.

Hbys ve Ahbs'ler renkli reçete uygulamasından aldıkları tokenı ve aşağıda detayı anlatılan biçimde oluşturdukları parametreyi harici giriş sayfası linkine ekleyerek aşağıdaki gibi bir link oluşturacaktır. Oluşturdukları bu linki Google Chrome tarayıcı ile otomatik olarak açacaklar, hekimler ekstra bir işlem yapmadan reçete yazma ekranına ulaşacaktır.

<https://renlirecete.saglik.gov.tr/Auth/ExternalLogin?accessToken=cNj4OMgO1E.....M5Mg%3D%3D¶m=MjA1ODgwODA.....2NTlwMCwx>

Burada dikkat edilmesi gereken en önemli husus parametrelerin url'e eklenirken **encode** edilerek verilmesidir. Aksi takdirde linkte meydana gelecek bozulma işleminin başarısız olmasına neden olacaktır.

Hbys'den alınması gereken hasta tc kimlik numarası, yatan hasta bilgisi, takip numarası, protokol numarası, provizyon tipi ve doktor medula branş kodu bilgileri aşağıdaki formatta aralarına virgül konularak birleştirilmeli, tek bir string oluşturulmalıdır.

HASTA_TC,YATAN_HASTA,TAKİP_NO,PROTOKOL_NO,PROVİZYON_TİPİ,DOKTOR_BRANŞ_KODU

ÖR: 2147....356,H,24P..QX,A1613..986,1,1500

Yukarıdaki formatta oluşturulan string base64 e çevrilip encode edildikten sonra url'e eklenmelidir. Belirtilen şekilde oluşturulan url üzerinden giriş yapıldığında doktor aşağıdaki ekranla karşılaşacaktır.

The screenshot shows the 'Renkli Reçete' application interface. The main content area is titled 'REÇETE OLUŞTUR'. It contains several sections:

- Hasta TC Kimlik Numarası:** A text input field containing '21477085256' and a 'Değiştir' button.
- Hasta Bilgileri:** A section with buttons for 'Hastanın Elindeki İlaçlar' and 'Hastanın Raporları'. It contains fields for 'Ad Soyad' (ÇENGİZ TUĞRA ERER), 'Cinsiyet' (Erkek), and 'Doğum Tarihi' (22.11.2000).
- Tanı ve Reçete Türü Bilgisi:** A section with a 'Tanı Kodu / Tanı' field and a 'Reçete Türü' dropdown menu set to 'Ayaktan'.
- Reçete Bilgisi:** A section with an 'İlaç Adı' field and a 'Majistrel Reç oluştur' button.

At the bottom right, there is a 'Reçeteyi Kaydet' button with a checkmark icon.

Örnek İstemci Kodu

Token oluşturma ve Reçete listesi alma işlevlerini kullanan örnek bir istemci uygulaması hazırlanmıştır. Uygulamaya ait örnek C# kodları aşağıdadır. Konsol uygulamasının tamamına geliştiriciler için hazırlanan portalden ulaşabilirsiniz.

```
namespace ApiTestUygulaması
{
    class Program
    {
        static void Main(string[] args)
        {
            using (var client = new HttpClient())
            {
                client.BaseAddress = new Uri("https://renklirecete.saglik.gov.tr/");
                client.DefaultRequestHeaders.Accept.Clear();
                client.DefaultRequestHeaders.Accept.Add(new
MediaTypeWithQualityHeaderValue("application/json"));

                TokenOlustur(client).Wait();
                ReceteListeGetir(client).Wait();
            }
        }

        private static async Task TokenOlustur(HttpClient client)
        {
            var receteToken = new ReceteToken()
            {
                DoktorTc = "58588134440", //11 hane olmak zorunda
                TesisKodu = "11000001", //belirli bir karakter sınır yok
                Sifre = "MTIzNDU2",
                KullaniciAdi = "11900012016" //11 hane olmak zorunda
            };
            HttpResponseMessage res = await
client.PostAsJsonAsync("api/receteapi/tokenolustur", receteToken);
            IEnumerable<string> tokens;
            res.Headers.TryGetValues("Token", out tokens);
            Console.WriteLine(tokens.FirstOrDefault());
            Console.WriteLine("Token başarıyla oluşturuldu.");
        }

        private static async Task ReceteListeGetir(HttpClient client)
        {
            var values = new Dictionary<string, string>();
            values.Add("kullaniciAdi", "11900012016");
            values.Add("sifre", "MTIzNDU2");
            values.Add("tesisKodu", "11000001");
            values.Add("baslangic", "01/12/2016");
            values.Add("bitis", "01/01/2017");
            HttpResponseMessage response = await
client.GetAsync("api/receteapi/recetelistegetir" +
Utilities.BuildUrlParametersString(values));
            response.EnsureSuccessStatusCode();
            string responseBody = await response.Content.ReadAsStringAsync();

            //istenirse direk modele cast edilebilir
            var result = JsonConvert.DeserializeObject<ReceteListe>(responseBody);
        }
    }
}
```

```
        Console.WriteLine("Reçete listesi başarıyla alındı.");  
        Console.WriteLine(responseBody);  
        Console.ReadLine();  
    }  
}
```