T.C.
ERCİYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

JSON WEB TOKEN KULLANARAK TOKEN TABANLI KİMLİK DOĞRULAMA GİNE VATANDAŞI BİLGİ SİSTEMİ RESTFUL WEB SERVİSİ

Hazırlayan
Saikou Ibrahima DIALLO

Danışman
Dr. Öğr. Üyesi Özkan Ufuk NALBANTOĞLU

Yüksek Lisans Tezi

Temmuz 2019
KAYSERİ

BİLİMSEL ETİĞE UYGUNLUK

Bu çalışmadaki tüm bilgilerin, akademik ve etik kurallara uygun bir şekilde elde edildiğini beyan ederim. Aynı zamanda bu kural ve davranışların gerektirdiği gibi, bu çalışmanın özünde olmayan tüm materyal ve sonuçları tam olarak aktardığımı ve referans gösterdiğimi belirtirim.

Saikou Ibrahima DIALLO

İmza 

"JSON Web Token Kullanarak Token Tabanlı Kimlik Doğrulama Gine Vatandaşı Bilgi Sistemi RESTful Web Servisi" adlı Yüksek Lisans Tezi, Erciyes Üniversitesi Lisansüstü Tez Önerisi ve Tez Yazma Yönergesi' ne uygun olarak hazırlanmıştır.

Hazırlayan

Saikou Ibrahima DIALLO

İmza

Danışman

Dr.Öğr. Üyesi Özkan Ufuk NALBANTOĞLU

Bilgisayar Mühendisliği ABD Başkanı

Prof. Dr. Veysel ASLANTAŞ

İmza

Dr.Öğr. Üyesi Özkan Ufuk NALBANTOĞLU danışmanlığında **Saikou Ibrahima DIALLO** tarafından hazırlanan **"JSON Web Token Kullanarak Token Tabanlı Kimlik Doğrulama Gine Vatandaşı Bilgi Sistemi RESTful Web Servisi"** adlı bu çalışma jürimiz tarafından Erciyes Üniversitesi Fen Bilimleri Enstitüsü **Bilgisayar Mühendisliği** Anabilim Dalında **Yüksek Lisans** tezi olarak kabul edilmiştir.

03 / Temmuz / 2019

JÜRİ:

Danışman : Dr.Öğr. Üyesi Özkan Ufuk NALBANTOĞLU .....

Üye : Dr.Öğr. Üyesi Fehim KÖYLÜ .....

Üye : Dr. Öğr. Üyesi Burcu GÜNGÖR .....

ONAY:

Bu tezin kabulü Enstitü Yönetim Kurulunun 16/07/2019 tarih ve 2019/14-18 sayılı kararı ile onaylanmıştır.

16/07/2019

Prof. Dr. Mehmet AKKURT

Enstitü Müdürü

TEŞEKKÜR

Öncelikle, bu Çalışmalarım boyunca farklı bakış açıları ve bilimsel katkılarıyla beni aydınlatan ve yardım eden sayın hocam Dr. Öğr. Üyesi Özkan Ufuk NALBANTOĞLU'ya teşekkür etmek istiyorum. Danışmanımın yanı sıra, tezim sırasında verdiği destek ve yardımlarını için arkadaşım Maliki MOUSTAPHA'ya teşekkür ediyorum.

Bu Yüksek Lisans eğitimim için burs imkânı sunan ve gereken desteği veren Yurtdışı Türkler ve Akraba Topluluklar Başkanlığı'na teşekkür ederim.

Ayrıca sınıf arkadaşı arkadaşlarım Ahmed USMAN ALİ, Yousouf Djelal İBRAHİM , Fatih BURADARDA ve Ousainou NYASSİ ile fikir alışverişinde bulundukları için teşekkür ederim.

Aileme özel bir teşekkürlerimi sunarım, benim için yaptıkları fedakarlıklar için aileme minnettar olduğumu ifade edemez. Duaları hayatımım bu aşamasına gelmeme yardımcı oldu. Sonunda, bu çalışmalarda beni destekleyen tüm arkadaşlarım ve kişilere teşekkür etmek istiyorum.

Saikou Ibrahima DIALLO

Haziran 2019, KAYSERİ

JSON WEB TOKEN KULLANARAK TOKEN TABANLI KİMLİK DOĞRULAMA GİNE VATANDAŞI BİLGİ SİSTEMİ RESTFUL WEB SERVİSİ

Saikou Ibrahima DIALLO
Erciyes Üniversitesi, Fen Bilimleri Enstitüsü
Yüksek Lisans Tezi, Haziran 2019
Danışman: Dr.Öğr. Üyesi Özkan Ufuk NALBANTOĞLU

ÖZET

Son on yılda, dünyadaki Afrika ülkelerinin ve diğer ülkelerin çoğu elektronik yönetimi geliştirmektedir. Özellikle Gine hükümeti, elektronik yönetim uygulama, süreçleri otomatikleştirme, çeşitli sistemleri kullanarak bilgi ve hizmetleri dijitalleştirme, bazı sorunları çözme, ülke ile ilgili tüm süreçleri kolaylaştırmayı amaçlayan bir plan yapmıştır. Ancak, çoğu uygulamanın karşılaştığı birçok sorun vardır ve genellikle en büyük sorun güvenliktir. Bu bağlamda ve Gine'deki elektronik idarenin gelişmesine katkıda bulunmak için, vatandaşlara, iş dünyasına ve diğer devlet kurumlarına sağlanan e-devlet uygulamaları ve hizmetleri için entegre bir platform sağlayan bir API önerilmektedir. Bu tezde JSON Web Token'i kullanan bir sistem önerilmektedir. Yeni bir endüstri standardı tabanlı (RFC 7519) JSON Web Token, durumsuzdur, bu nedenle bir RESTful Web Hizmeti uygulamasında uygulanması uygundur. Bu tez projesi, bir prototip olarak geliştirdiğimiz Gine vatandaşı uygulamasına yüksek düzeyde güvenli bir kimlik doğrulaması sağlayacak olan JSON Web Token ile token tabanlı kimlik doğrulama yöntemini kullanarak kimlik doğrulama tekniklerinin nasıl uygulanacağını tartışmaktadır.

Anahtar Kelimeler: Web Servis,Web servis Güvenliği,JSON Web Token,Spring Boot,RESTful

APPLİCATİON OF TOKEN-BASED AUTHENTİCATİON İN THE RESTFUL WEB SERVİCE USİNG JSON WEB TOKEN FOR GUİNEAN CİTİZEN INFORMATİON SYSTEM

Saikou Ibrahima DIALLO

Erciyes University, Graduate School of Natural and Applied Sciences
Master Thesis, June 2019
Supervisor: Dr.Özkan Ufuk NALBANTOĞLU

ABSTRACT

During the last decade, most of African countries and other countries around the world have been developing the electronic administration. Especially in Guinea the government engaged itself in a plan aiming to implement an electronic administration, to automate processes, to digitalize information and services using various systems, to solve some issues, to facilitate all the processes about the country. However, there are a lot of problems faced by most of the applications and generally the biggest problem is the security. In this context, and to contribute to the development of the electronic administration in Guinea, we propose an API that ensures an integrated platform for the e-government applications and services provided to citizens, business and other government agencies. This thesis project uses JSON Web Token. JSON Web Token is based on the new industry standard (RFC 7519). JSON Web Token is stateless, so it is suitable to be implemented on a RESTful Web Service application. This thesis project discusses how to implement authentication techniques using token-based authentication method with a JSON Web Token, which will produce a high level secure authentication to the Guinean citizen application we developed as a prototype.

Keywords: Web Service, Web Service Security, JSON Web Token, Spring Boot. RESTful

İÇİNDEKİLER

JSON WEB TOKEN KULLANARAK TOKEN TABANLI KİMLİK DOĞRULAMA GİNE VATANDAŞI BİLGİ SİSTEMİ RESTFUL WEB SERVİSİ

KISALTMALAR

WS : Web Service (Web Servis)

JSON: JavaScript Object Notation

JWT: JSON Web Tokens

REST: Representational State Transfer

SOAP: Simple Object Access Protocol

UDDI: Universal Description Discovery and Integration

WSDL: Web Services Description Language

API: Application Programming Interface

JAX-RS: Java API for RESTful Web Services

HTTP: Hypertext Transfer Protocol

URL: Uniform Resource Locator

XML: Extensible Markup Language

W3C: World Wide Web Consortium

RBAC: Role-Based Access Control

CORBA: Common Object Request Broker Architecture

DAO: Data Access Object

JDBC: Java DataBase Connectivity

LDAP: Lightweight Directiory Access Protocol

TABLOLAR LİSTESİ

ŞEKİLLER LİSTES

GİRİŞ

Son on yılda, Afrika ülkeleri ve dünyadaki diğer gelişmekte olan ülkelerin çoğu elektronik yönetim sistemleri geliştirmektedir. Özellikle Gine hükümeti, elektronik bir yönetim kurmayı, süreçleri otomatikleştirmeyi, çeşitli sistemleri kullanarak bilgi ve hizmetleri dijitalleştirmeyi amaçlayan bir plan yapmıştır.

Vatandaşların bilgisayar başında hizmet alması temel amaç edinilmektedir. Aslında, hükümetler yeni nesil bilgi teknolojilerinin kullanımını ve internet tabanlı kullanıcıları teşvik etmektedir. Ayrıca, tüm sosyal kategorileri bilgisayar tekniklerine dahil etmeyi amaçlayan aile bilgisayar programları başlatılmıştır. Günümüzde, Gine'nin teknolojik altyapısı e-devlet eylemleri için tercih edilmştir ve bu yönde projeler üretilmiştir. Ancak şu anda bu hizmetler yalnızca bilgilendirici amacı gütmekte, yani gerçekleştirilen e-devlet uygulamaları devlet hizmetleri hakkında bilgi sunmakta, ancak gerekli eylemleri sunmamaktadır. Gine'nin e-devlete olan ilgisini gösteren bir diğer girişim, ekonomi bakanlığı içindeki elektronik idare için bir dijital birimin oluşturulmasıdır. Gine, ikinci aşamada çevrimiçi hizmetler sunan e-devlet uygulamaları sunmayı planlamaktadır.

Genellikle, e-devlet uygulamaları hem yüksek düzeyde birlikte çalışabilirlik hem de siber güvenlik gerektirir. Birlikte çalışabilirlik gereksinimi, bu tür uygulamalardaki çok sayıda heterojen katılımcıdan kaynaklanmaktadır. Devlet kurumları ve vatandaşlar arasında paylaşılan özel ve gizli bilgiler ışığında, güvenlik sürekliliğinin de dikkate alınması gerekmektedir. Servis odaklı bir yaklaşım bu gereksinimler için bir çözüm sunmaktadır. İyi tanımlanmış standartlar, çeşitli heterojen uygulamalar arasındaki karşılıklı bağlantıya izin vermektedir. Web servisleri [4] yaygın olarak servis odaklı yaklaşımlar için bir uygulama olarak kullanılmaktadır. Bu yapılar, birlikte çalışabilir varlıklar sunar ve onları güvenceye almak için standartlar önerir. Bu gerçek, çeşitli ülkeleri kendilerini Web hizmetleri tabanlı e-devlet uygulamalarına dahil etmeye teşvik etmektedir.

Bu bağlamda, bu tezde, Gine'deki elektronik yönetimin gelişmesine katkıda bulunmak için, vatandaşlara, iş dünyasına ve diğer devlet kurumlarına sunulan e-devlet uygulamaları ve hizmetleri için entegre bir platform sağlayan bir API önerilmektedir

1. BÖLÜM
GENEL BİLGİLER VE LİTERATÜR ÇALIŞMASI

I. Web Servisleri Hakkında Genel Bakış

1.1 Web Servislerinin Tanımı

Bir web servisi, web üzerinden iki yazılım varlığı arasındaki iletişim için standart bir yöntemdir (web API) [1]. Web Servisleri, önder ağını http protokolü üzerinden ileten ve mesajları ya da verileri xml olarak ve json gibi çeşitli formatlarda veri alışverişi yapan istemci sunucular içermektedir. Farklı ortamlarda inşa edilen yapılar birbirlerine tebliğde bulunabilir ve bu çerçevede farklı yazılım uygulamaları arasında çalışabilirliği sağlarlar. Bu sayede, uygulamaların, gevşek bir şekilde birbirine bağlı bir şekilde geliştirilmesine de izin verirler.

World Wide Web Consortium, bir ağ üzerinden çalışabilir makinelerden etkileşimi desteklemek için tasarlanmış bir yazılım sistemi olarak Web Servisi tanımlar.

Bu tanımda üç önemli anahtar noktadan bahsedebiliriz:

- Makineden makineye (veya uygulamadan uygulamaya) etkileşim için tasarlanmıştır.
- Birlikte çalışılabilir olması gereken herhangi bir başka uygulama ile konuşulmalıdır. Ve bu konuşmalar platforma bağlı değildir.
- Ağ üzerinden iletişime izin vermelidir.

Diğer şekilde, Web Servisleri iki bölüme odaklanmıştır: Servis Sağlayıcı olarak adlandırılan bir ana bilgisayar web uygulaması ve Servis İsteği Yapan bir Talep yazılımı uygulaması.

Servis Sağlayıcı, herhangi bir programlama dilinde geliştirilebilir, web servislerini kullanmanın en önemli esneklik özelliği şudur: Servis İstemci, Servis Sağlayıcı'yı geliştirmek için kullanılacak herhangi bir programlama dilini seçebilir.

W3C (World Wide Web Consortium) Tanımı web servislerini şu şekilde tanımlar: "Bir web servisi, bir URL tarafından tanımlanmış, ortak arayüzlerin ve ciltlemelerin XML'de tanımlandığı bir yazılım sistemidir. Tanımı, diğer yazılım sistemleri tarafından dinamik olarak keşfedilebilir. Bu diğer sistemler daha sonra, İnternet servisiyle, internet protokolleri tarafından taşınan XML mesajlarını kullanarak tanımında tarif edildiği şekilde etkileşime girebilir.[1]

Bir web servisi, uygulamalar veya sistemler arasında veri alışverişi için kullanılan açık protokoller ve standartlar topluluğudur. Çeşitli programlama dillerinde yazılmış ve çeşitli platformlarda çalışan yazılım uygulamaları, internet gibi bilgisayar ağları üzerinden tek bir bilgisayardaki işlem içi iletişime benzer şekilde veri alışverişinde bulunmak için web hizmetlerini kullanabilir. Bu birlikte çalışabilirlik (örneğin, Java ve Python veya Windows ve Linux uygulamaları arasında) açık standartların kullanımından kaynaklanmaktadır.[2]

1.2 Evrensel Tanım Keşif Ve Entegrasyon (UDDI)

UDDI web servislerini tanımlamak, yayınlamak ve bulmak için kullanılan XML tabanlı bir standarttır.

UDDI, HTTP, XML, XML Şeması ve SOAP dahil olmak üzere ortak bir dizi endüstri standardını temel alır. Yalnızca bir kuruluş içinde dahili olarak açığa çıkan veya halka açık hizmetler için Web Hizmetleri tabanlı bir yazılım ortamı için bir altyapı sağlar.

UDDI, web servislerinin dağıtılmış bir kaydı için tanımlanmış bir standarttır. Bu standart platformdan bağımsız, açık bir çerçeveye sahiptir.

UDDI, SOAP, CORBA, Java RMI Protokolü aracılığıyla iletişim kurabilir. Ayrıca web servislerine arayüzleri tanımlamak için Web Servis Tanımlama Dilini (WSDL) kullanır.

Bu standart, SOAP ve WSDL ile birlikte web servislerinin üç temel standartından biri olarak kabul görmektedir.

UDDI, işletmelerin birbirlerini keşfetmelerini ve İnternet üzerinden nasıl etkileşime girdiklerini belirlemelerini sağlayan açık bir endüstri girişimidir.

1.3 Web Hizmetleri Açıklama Dili (WSDL)

WSDL, bilgiyi merkezi olmayan ve dağıtılmış ortamlar aracılığıyla göndermek ve almak için kullanılan XML tabanlı bir protokole karşılık gelir. Web hizmetinde hangi servislerin mevcut olduğunu ve ayrıca Web hizmeti için yöntemleri, parametre adlarını, parametre veri türlerini ve dönüş veri türlerini tanımlar. WSDL belgesi oldukça güvenilirdir ve web servislerini kullanan uygulamalarda sıklıkla tercih edilmektedir.

1.4 Basit Nesne Erişim Protokolü (SOAP)

SOAP, dağıtılmış bir ortamda bilgi alışverişinde kullanılan XML tabanlı bir standart birlikte çalışabilirlik protokolüdür. İstemciler ve hizmetler arasında veri alışverişi yapmak için ortak bir mesaj biçimi sağlar.

SOAP, merkezi olmayan, dağıtılmış bir ortamda yapılandırılmış bilgilerin değiş tokuş edilmesini amaçlayan hafif bir protokoldür. Genişleyen bir mesajlaşma çerçevesini tanımlamak için, çeşitli temel protokoller üzerinden değiştirilebilen bir mesaj yapısı sağlayan XML teknolojilerini kullanır. Çerçeve, belirli bir programlama modelinden ve diğer uygulamaya özel sözdiziminden bağımsız olacak şekilde tasarlanmıştır. [1]

1.5 Web Servisinin Özellikleri

Web servisleri, servis sağlayıcı ile servis tüketicisi arasında mesaj alışverişinde bulunmak için XML kullanır. XML kullanmak, işletim sistemi veya platform bağlama gereksinimini ortadan kaldırır.

Web servisleri tüketicileri bu Web servisine doğrudan bağlı değildir. Web hizmeti arayüzü, müşterinin servisle etkileşim yeteneğinden ödün vermeden zaman içinde değişebilir.

Web servisleri senkron veya asenkron olabilir. Eşzamanlı çağrılarda, kullanıcı devam etmeden önce hizmetin çalışmasının tamamlamasını engeller ve bekler. Eşzamansız istemciler ise sonuçlarını daha sonraki bir zamanda alabilirler.

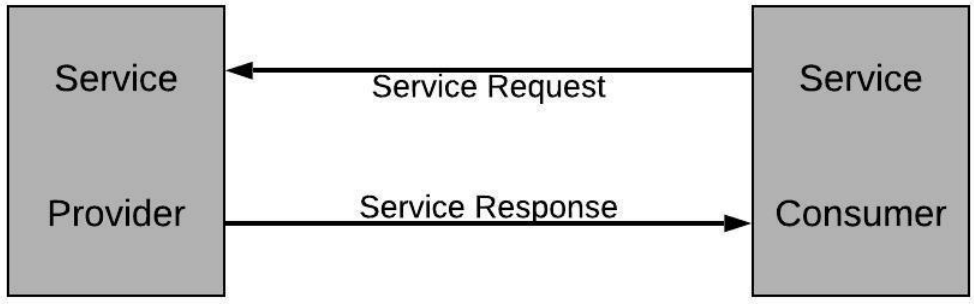
Web servisleri, istemcilerin XML tabanlı bir protokol kullanarak uzak nesnelerdeki prosedürlerin, işlevlerin ve yöntemlerin çağırmasını sağlar. Uzaktan prosedürler, bir web servisinin desteklemesi gereken giriş ve çıkış parametrelerini gösterir.

Web hizmetleri, iş entegrasyonunu kolaylaştırmak için şeffaf belge alışverişini destekler.

1.6 Web Servisinin Mimarisi

Servis odaklı mimari, temel olarak bir hizmetler koleksiyonudur. Bu hizmetler birbirleriyle iletişim kurarlar. İletişim, basit veri aktarımı içerebilir ya da bazı aktiviteleri koordine eden iki ya da daha fazla servisi içerebilir. Servisleri birbirine bağlamak için bazı araçlara ihtiyaç vardır.

Web Servisleri teknolojisi, servis odaklı mimarilerin en muhtemel bağlantı teknolojisidir. Aşağıdaki şekilde temel hizmet odaklı bir mimari gösterilmektedir. Sağdaki servis tüketicisinden soldaki bir servis sağlayıcısına bir servis talebi mesajı gönderdiği gösterilmektedir. Servis sağlayıcı servis tüketicisine bir cevap mesajı döndürür. İstek ve müteakip cevap bağlantıları, hem servis tüketicisi hem de servis sağlayıcı için anlaşılabilir bir şekilde tanımlanır. Bu bağlantıların nasıl tanımlandığı web servisleri açıklamasında açıklanmıştır. Bir servis sağlayıcı, bir servis tüketicisi de olabilir.



Şekil 1. 1: Web Service Mimarisi

1.7 Web Servisleri Güvenliği

Genel bir web güvenliği tanımı Garfinkel ve Spafford 1997 [3] tarafından sağlanmıştır. Web güvenliği, web sunucularını, web kullanıcılarını ve çevreleyen organizasyonları korumaya yönelik bir dizi prosedür, uygulama ve teknolojidir. Güvenlik sizi (kullanıcıyı) beklenmeyen davranışlara karşı korur. Kullanıcılar ve web servis sağlayıcıları, web

konusunda güvenlikle ilgili beklenen davranışlarla ilgili bir takım varsayımlara sahiptir. Kullanıcıların bakış açısına göre beklentileri, sunulan hizmetin meşru, güvenli ve özel olmasıdır; Sağlanan hizmetler veya bilgilerin bilgisayar virüsleri veya kullanıcının bilgisayar sisteminin kötü amaçlı amaçlar için kullanılmasına izin verecek içerik içermemesi anlamında güvenlidir. Sunucunun bakış açısına göre beklentileri bilgi istemidir veya bir servis meşru ve sorumludur; Hem sunucu hem de kullanıcı açısından bakıldığında, iletimlerinin üçüncü bir tarafça değiştirilmemesi açısından, iletişimlerinin gizlice dinlenmekten özgür ve güvenilir olacağı beklentisine sahiptirler.

1.7.1 Web Güvenliğinin Hedefleri Arasında:

1.7.1.1 Kimlik Doğrulama:

Güvenlik sistemlerinde, kimlik doğrulama, kişinin kendi kimliğine dayalı sistem nesnelerine erişimin doğrulama sonucunda verilmesi sürecidir. Kimlik doğrulama, bireyin iddia ettiği kişi olduğunun teyit edilmesini sağlar.

1.7.1.2 Yetkilendirme:

Birine bir şeyler yapmasına izin verme sürecidir. Çok kullanıcılı bilgisayar sisteminde, bir sistem yöneticisi kullanıcı için sisteme açık bir erişim olan ve kullanım ayrıcalıklarına sahip olan sistemi tanımlar.

1.7.1.3 Gizlilik:

Bilgilendirme, yalnızca erişmeye yetkili olan kullanıcıların erişimine açık bulumakta, ve böylece belli bir seviyede gizliliğe erişmektedir. Gizlilik, bilgisayarla ilgili varlıklara yalnızca yetkili taraflarca erişilmesini sağlar.

1.7.1.4 Mesaj Bütünlüğü:

Bir mesajın içeriğinde tahrif ve değişikliğe gidilmemesini sağlayacak yöntemler bulundurulmaktadır. Bu, varlıkları sadece izin veren taraflarca veya sadece yetkili şekillerde değiştirilebileceği anlamına gelir.

1.7.1.5 Hesap verebilirlik:

Bir saldırıdan kurtulmanın yollarından biri, sistemdeki kullanıcıların ne yaptığını ve de ne zaman yaptıklarının bilinmesidir. Hesap verebilirlik için bir sistem olduğunu bilmek, potansiyel saldırganları caydırmaktadır.

1.7.1.6 Uygunluk:

Kullanılabilirlik, varlıkların yetkili kişiler tarafından uygun zamanlarda erişilebilir olduğu anlamına gelir. Kullanılabilirlik, sistemin çalışır durumda olduğu sürenin oranıdır.

1.7.1.7 İtiraz Edilmeme:

Reddetmeme, itirazda bulunan bir tarafın bir beyan veya sözleşmenin geçerliliğini reddetmemesini veya reddetmesini sağlama kavramıdır. Bu kavramın en yaygın uygulaması imzaların doğrulanması ve güvende olmasıdır.

1.7.1.8 Doğrulama

Doğrulama, tüm uygulamalar için en önemli kısımlardan biridir. Birkaç adım içerisinde doğrulamayı sağlamak açısından, sunucu tabanlı çerezler ve kimlik doğrulama, web tabanı uygulamasında kimlik doğrulama için en kolay çözümdür.

1.8 REST Hizmetlerinin Özellikleri

1.8.1.1 İstemci-sunucu tabanlı mimari:

İstemci-sunucu mimarisi, sunucunun, müşteri tarafından tüketilecek kaynakların ve hizmetlerin çoğunu barındırdığı, sağladığı ve yönettiği bir bilgi işlem modelidir. Bu tür bir mimaride, bir ağ veya internet bağlantısı üzerinden merkezi bir sunucuya bağlı bir veya daha fazla istemci bilgisayar bulunur. Bu sistem bilgi işlem kaynaklarını paylaşır.

Durumsuzluk bir REST servisinin en önemli özelliğidir. Bir REST HTTP isteği, sunucunun yanıtı anlaması ve geri dönüş yapması için gereken tüm verilerden oluşur. Bir istek yapıldığında, sunucu isteğin bir süre sonra gelip gelmediğini hatırlayamamaktadır. Bu sebeple operasyonun durumsuz bir operasyon olduğu söylenebilir.

1.8.1.2 Önbelleklenebilirlik

Birçok geliştiriciye göre bir teknoloji yığınının web uygulamalarını veya API'larını engelliyor gibi bir görüş bulunsa da gerçekte öne sürülen mimarilerin amacı budur. Veritabanı bir web uygulamasında potansiyel bir dönüm parçası olabilir. Bir uygulamanın etkili ölçeklendirilebilmesi için içeriğin önbelleğe alınıp yanıt olarak gönderilmesi gerekmektedir. Önbellek geçerli değilse, durumu kotarmak bizim sorumluluğumuzdur. REST hizmetleri, ölçeklendirme için uygun şekilde önbelleğe alınmalıdır.

1.8.1.3 Çok katmanlı sistem

REST API'si birden fazla sunucudan servis edilebilir. Bir sunucu bir diğerinden talepte bulunabilir. Böylece bir talep müşteriden geldiğinde, nihayet müşteriye bir cevap vermek için birçok sunucu arasında istek ve cevap iletilebilir. Bu kolayca uygulanabilir çok katmanlı sistem her zaman web uygulamasını gevşek bir şekilde bir arada tutmak için iyi bir stratejidir.

1.8.1.4 Kaynakların temsili

REST API, konuşmak için tek tip bir arayüz sağlar. Tekdüzen Kaynak Tanımlayıcı (URI) kaynakları (veri) eşlemek için kullanır. Ayrıca, cevap olarak belirli bir veri formatı isteme avantajına da sahiptir. İnternet Ortam Türü (MIME türü), sunucuya istenen kaynağın o türden olduğunu söyleyebilir.

1.8.1.5 Uygulama özgürlüğü

REST, web servislerinizi tanımlamak için sadece bir mekanizmadır. Birden çok şekilde uygulanabilen mimari bir stildir. Bu esneklik nedeniyle, istenilen şekilde REST hizmetlerinin oluşturulması mümkündür. REST ilkelerini izleyene kadar sunucu, platformu veya teknolojiyi seçme özgürlüğüne sahiptir. REST hizmetlerinin ölçeklendirilmesi için akıllı önbellekleme şarttır.

Web Hizmetleri güvenlik politikası genellikle üç boyuta ele alınır: kimlik doğrulama politikası, erişim kontrol politikası ve gizlilik politikası.

Kimlik Doğrulama Politikası: Kimlik doğrulama politikası, temel olarak kaynaklara erişen öznenin sistemde var olan ve şu anda yaygın olarak kullanılan öznitelik kimlik belgesi olduğunu belirlemek için bazı belgelendirme bilgilerini tanımlamayı amaçlar. Öte

yandan, kimlik doğrulama politikasında genellikle erişim kontrolü veya gizlilik koruma sorunlarını çözmek için erişim kontrolü politikası ve gizlilik politikası ile birlikte kullanılır.

1.8.1.6 Erişim Kontrolü Politikası

Web Servislerinde erişim kontrolü, bir kullanıcının veya uygulamanın, belirli bir koşulu yerine getirme sözü verilen uygun Web Servislerine erişebileceği anlamına gelir. Erişim kontrolü, Web Hizmetleri güvenlik politikası için çok önemlidir. Rol tabanlı erişim kontrolü (RBAC) sayesinde rol kavramı tanımlanmıştır; böylece rollere izin verilerek kullanıcıların ve izinlerin mantıksal olarak ayrılması sağlanmıştır. Kullanıcılar, yöneticinin atadığı rollere göre uygun izinleri alır ve bu da yönetim maliyetini önemli ölçüde azaltır. RBAC, bir çok avantajı nedeniyle erişim kontrolünde sıcak bir araştırma haline gelmiştir ve daha yaygın olarak çalışılmakta ve uygulanmaktadır.

1.8.1.7 Gizlilik Politikası

Gizlilik politikası genellikle iki yönlü güvenli erişim politikası olarak kabul edilirken, erişim kontrolü politikası genellikle tek yönlüdür, yani Web Hizmetleri sağlayıcıları, kullanıcıların erişimini sınırlandırmak için erişim kontrolü kurallarının bazılarını tek taraflı olarak formüle eder. Ancak, hem Web Servisleri sağlayıcısı hem de talep eden, uygun gizlilik politikasına sahip olabilir. Servis sağlayıcılar için gizlilik politikası, Web Servisleri istemi tarafından sağlanan gerekli kullanıcı bilgilerini ve kullanıcı bilgilerinin nasıl kullanılacağını belirler. Web Servisleri talep edenlerin bakış açısına göre, kullanıcılar, kişisel bilgilerin ne olduğunu açıklamak istediğini belirten kişisel bir gizlilik politikasına sahip olabilir.

Web Hizmetleri güvenlik politikasının bu üç yönü bağımsız değil, birleşiktir. Her biri belirli bir güvenliğe odaklanır ve gerektiğinde sorunları çözmek için birleştirir. Erişim kontrolü politikası, Web Hizmetleri güvenlik politikası için önemlidir ve erişim kontrolü modeli, Web Hizmetleri iletişimindeki uygulamaları için hızla geliştirilmiştir ve kaynakları yetkisiz erişime karşı korumada önemli bir rol oynamıştır.

REST mimarisinde, REST mimarisini izleyen sunucu, veri kaynağına ve veri alan istemcilere erişim sağlar. Her veri kaynağı URL bağlantısı kullanılarak tanımlanır.

REST, metin, JSON ve XML gibi verileri sunmak için çeşitli formatlar kullanır. Aşağıda kullanılan HTTP yöntemleri verilmiştir: REST mimarisi:

GET - veri kaynağını okumak için erişim sağlar

PUT - mevcut verilerin güncellenmesini sağlar

DELETE - veri silinmesini sağlar

POST - yeni veri oluşturmayı sağlar

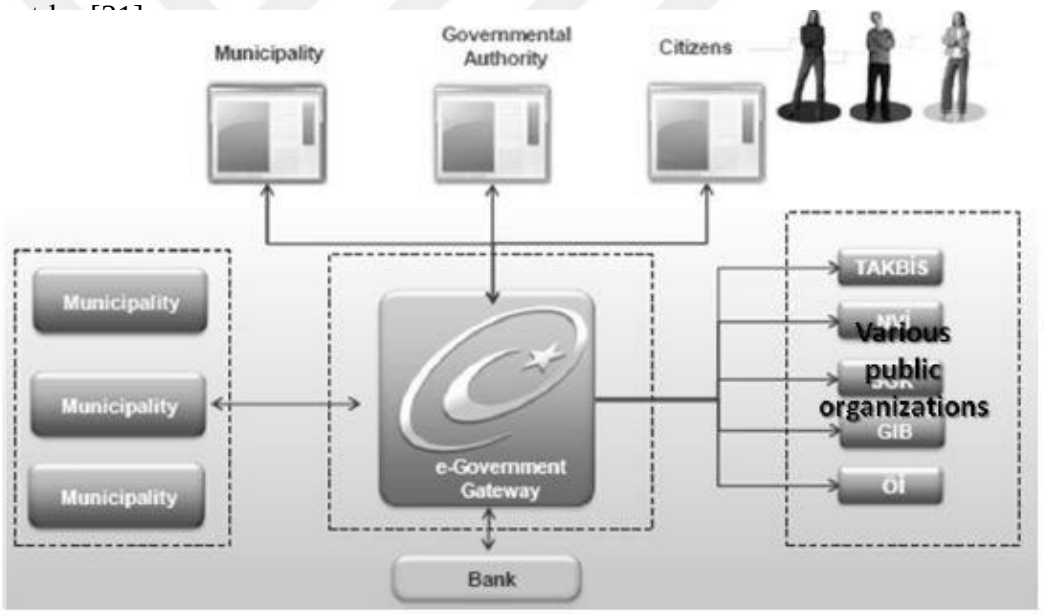
1.9 Türkiye'de E-Devlet Sistemi

Bununla birlikte, çeşitli problemler ve eksikliklerle bile olsa, ISS, bu tür e-dönüşüm perspektifi içinde, diğerlerinin yanı sıra, "vatandaş odaklı hizmet dönüşümü" ve "kamu yönetiminde modernleşme" için Türkiye'deki çeşitli girişimlerin temeli olmuştur. Örneğin, e-hizmetlerin yaygınlaştırılması ve yaygınlaştırılması için fizibilite çalışmaları yapılmıştır. Vatandaşlara faydaları en üst seviyeye çıkaran hizmetleri öncelik sırasına koymayı ve "olduğu gibi" ve "olmak" durumları arasında "boşluk analizi" gerçekleştirmeyi amaçlayan model tabanlı bir yaklaşım metodolojisinin ardından, bu fizibilite çalışmaları ile 100'den fazla uygulanabilir kamu hizmeti belirlenmiş, 40 devlet organı tarafından elektronik ortamda hizmet verilebileceği görülmüştür. (Acar ve Kumas, 2009) Bu fizibilite çalışmalarına, ilgili kamu kurumları arasında farkındalık, bilgilendirme ve güncelleme amaçlı çeşitli sunumlar ve toplantılar eşlik etmektedir.

Bu fizibilite çalışmaları ve çeşitli devlet kurumları ve kamu kurumları arasındaki yoğun etkileşimlere dayanarak, bir hükümet ağ geçidi (EGG) (ve portal) geliştirilmiştir. Resmi olarak Aralık 2008'de başlatılan EGG, Bilgi Toplumu'na dönüşüm ve Türkiye tarafından belirlenen ilgili Stratejik Hedeflere ulaşmak için önemli bir dönüm noktasıdır. Türkiye'nin 70 milyon vatandaşına ve iş dünyasına, e-Devlet hizmetlerine tek noktadan erişime sahip, tek elden hizmet vermeyi hedeflemektedir. EGG'nin uygulama süreci hizmet dönüşümüne dayanmaktadır. Temel ilke, kamu kurum ve kuruluşlarına vatandaşlara ve işletmelere sunulan hizmetlerin iş süreçlerini yeniden tasarlarken kullanıcı memnuniyeti konularının dikkate alınması olmuştur. Hizmet dönüşümündeki öncelik, yalnızca herhangi bir iyileştirme yapmadan mevcut iş süreçlerini elektronik kanallara aktarmak değildir; Aksine, amaç bu elektronik devlet hizmetlerini,

gerektiğinde, etkili, kesintisiz, hızlı, şeffaf, güvenilir bir şekilde entegrasyon veya basitleştirme için kullanıcı ihtiyaçlarına göre yeniden tasarlanan süreçler olarak sunmak olmuştur. Halen EGG'ye entegre edilmiş 215'ten fazla e-hizmet vardır ve yeni hizmetler geliştirilme aşamasındadır. Bu hizmetlerin yanı sıra, EGG bireysel ajansların kendi web sitelerinden sağladığı yüzlerce elektronik hizmete bağlantı sağlar.

EGG, tüm hizmetlere tek bir yerden güvenli bir şekilde erişebilmek için tek bir kayıt mekanizmasıyla güvenilir, kullanıcı dostu bir seçenek sunar. Kişisel şifre ve e-imza kullanarak kimlik yönetimi entegre edilmiştir Türkiye, şu anda 3.00.000 kayıtlı / normal kullanıcıya 215'ten fazla entegre hizmet sunan EGG ile beraber rehberlik eden binlerce içerik sayfasının yanı sıra vatandaşları, işletmeler ve devlet kurumları için tek duraktır. noktası olmayı hedeflemektedir. daha fazla bilgi veya hizmet için ilgili kurumlara



Şekil 1. 2 Türkiye'de e-devlet sistemi

II. LİTERATÜR ÇALIŞMASI

Daha önce yapılan çalışmalara göre ve bir yaklaşım çözüm önermek için Web Servisleri Güvenliği alanında hala önemli gelişmeler görülmektedir . Önerilen sistem bu çalışmalara dayanarak yapılandırılmıştır. Bazı önceki çalışmalar aşağıdaki gibidir.

[2] Bu çalışmada JSON Web Token kullanılmıştır. JSON Web Tokenin durumsuz ve bu nedenle bir RESTful Web Hizmeti uygulamasında uygulanması uygun olmasından faydalanılmıştır. Bu araştırmada, bir JSON Web Token ile token tabanlı kimlik doğrulama yöntemini kullanarak kimlik doğrulama tekniğinin nasıl uygulanacağı, nasıl güvenli bir kimlik doğrulama üretileceği ve web servisinin nasıl SIKASIR'ın çok platformlu hale getireceğini tartışılmıştır.

[3] Bu çalışmada, Biyometrik kimlik doğrulama teknolojisini tanıtarak güvenlik ve doğrulama sürecini arttırmak ve böylece yasadışı göçmenleri ve teröristleri tespit etmek için tüm vatandaşlara benzersiz bir numara sağlamak için bir Vatandaş Kart Sistemi geliştirmesi amaçlanmıştır.

Adnaan Masood: [4] araştırmalarında, mevcut standartların zorluklarını tartışmış ve güvenlik açıklarını tespit ederek hizmet güvenliğini artıran ve OWASP (Açık Web Uygulaması Güvenlik Projesi) web hizmetleri için ilk 10 güvenlik açığına genel bir bakış sunan yeni teknikler ve araçları gözden geçirmiştir ve bu güvenlik açıklarını keşfetmek için potansiyel statik kod analiz teknikleri önermiştir. Web servislerini, yazılım / program doğrulamasını ve güvenlik geliştirme yaşam döngüsünü hedef alan güvenlik sorunlarını gözden geçirmektedir. Ayrıca hem SOAP hem de REST tabanlı web servislerinin uyarlayıcı ve uygulayıcılarını önermektedir.

Chi Po Cheong, Chris Chatwin ve Rupert Young: [5] makalelerinde , mesaj bütünlüğü ve mesaj gizliliği sağlayan mevcut Web Servis Güvenliği standartlarını iyileştirmek için yeni bir güvenlik belirteci önermekte ve uzak bir müşteri konumunun doğrulanması için kullanmaktadırlar. Ayrıca, XML ve Web Hizmetleri güvenlik standartlarına ve bunların ilişkilerine kısa bir giriş sunmaktadırlar.

[6] Anton V. Kolomeets Sonlu otomata modelini kullanarak web servis güvenliğini kontrol etmeyi düşünmüşlerdir. Saldırı düzenlerinin nasıl sunulacağını (dizeler, normal ifadeler veya otomatlar olarak) ve bu gösterimin avantaj ve dezavantajlarını ve ayrıca geri sembolik analizlerden gelen otomat denklemlerini kısaca ele aldıklarını tartışmışlardır.

[7] : SOA gibi herhangi bir Web servisine bağlı ortamın güvenlik seviyesini artırmak için yeni bir güvenlik çerçevesi önerilmiştir. Bu güvenlik modelinin Web hizmetleri için sunulan diğer güvenlik çözümlerinden temel farkı , Web hizmetinin yeni bir yönünü düşünmektir . XML şifreleme ve XKMS içeren bir çerçeve önerilmiş ve WSDL dosyasının güvenliğine odaklanılmıştır.

[8] : Genel olarak, semantik Web servisleri modeline genişletilmiş bir güvenlik kontrol merkezine dayanan etkili bir semantik Web servislerine erişim kontrol modeli ve metodu önerilmiştir . Bu modelde, erişim kontrol fonksiyonu güvenlik kontrol merkezi tarafından yürütülür. Bu yöntem sayesinde, erişim kontrolü, doğrulama ve anlamlandırma belirsizliği, araştırma çalışmasının anlamsal Web servislerinin güvenlik seviyesini oldukça artıracaktır.

[9] Bu araştırmada REST mimarisi stili ve JSON Schema / JSON Meta Şeması gibi iki güçlü teknolojinin bir araya getirilmesinin, veri bilim adamlarının bilgi grafikleri oluşturulmasına izin veren bir RESTful web servisini kullanmalarını nasıl sağladığını ve büyük ve anlamsal olarak tercih edilen temsillerin bir sunumunu nasıl yaptıkları incelenmiştir.

[10] Andrea Arcuri:makalerinde,yazdıkları koda tam erişime sahip olan geliştiriciler açısından test etmeyi düşünmektedir. Bu nedenle, test vakalarının evrimsel bir algoritma kullanarak otomatik olarak üretildiği tam otomatik bir beyaz kutu test yaklaşımı önermektedir. Testler kod kapsamı ve hata bulma ölçütlerine göre ödüllendirilir. Ayrıca tekniklerini açık kaynaklı olan EVOMASTER adlı bir araçta da uygulamışlardır. İki açık kaynaklı, ancak önemsiz olmayan RESTful servis ve endüstriyel bir servis üzerinde yapılan deneyler, yeni tekniği otomatik olarak bu uygulamalarda 38 gerçek hata bulduğunu göstermektedir. Bununla birlikte, elde edilen kod kapsamı, bu hizmetlerde zaten mevcut olan el ile yazılmış test takımlarının başardıklarından daha düşüktür.

[11]: Bu araştırmada, web servis kompozisyonu için çeşitli otomasyon ve teknikleri araştırmakta ve her yöntemin avantajlarını, dezavantajlarını ve zorluklarını ayrıntılarıyla açıklayan 18 yöntemi karşılaştırmaktadır. Araştırma, bu alanı ve sorunlarını inceleme, daha çeşitli veriler sağlama ve trafik ağlarının sorunlarını inceleme imkanı sunar. Bu, araştırmacıların yenilikçi olmalarını ve REST web hizmeti kompozisyonunun otomasyonu için çeşitli zorluklar ortaya koymalarını sağlamaktadır.

[12] : Bu araştırmada, düşük işlem süresi ve düşük mesajlaşma ek yükü açısından hafif bir güvenlik yaklaşımı önerilmiştir. Uygulamadan, önerilen güvenlik sisteminin hem işlem süresi hem de genel mesajlaşma adı verilen parametreler açısından mevcut SSL / TLS'den daha iyi performans gösterdiğini belirlenmiştir. JSON ve Plain text gibi hafif veri formlarını kullandılar, bu da XML'den daha az bant genişliği tüketerek iletişim performansını daha da geliştirmiştir. Bu nedenle, önerdikleri güvenlik sistemi, şifreleme tekniğini kullanarak veri gizliliğini, karma ve kullanıcıyı kullanan veri bütünlüğünü ve ayrıca imza mekanizmasını kullanan veri doğruluğunu korumaya yardımcı olmaktadır.

[13] : Bu çalışmada, bir REST web servis API'si sağladığını iddia eden tam olarak 500 site ve diğer teknolojilere dayalı 22 servis (ör. SOAP) bulmak için alexa.com'daki en iyi 4000 siteyi ziyaret etmişlerdir. REST hizmetlerinden geçerek, literatürde toplanan toplam 26 farklı özelliği analiz ederek, REST ilkelerine uyumluluk düzeyini, ortak geliştirici kararlarını ve REST hizmet geliştirme en iyi uygulamalarına uymalar tartışılmıştır.

[14] **:** Bu araştırmada , Sosyal Güvenlik'in özel alanı için kalite özelliklerini göz önünde bulundurarak , mobil uygulamalar için hizmet odaklı bir yazılım mimarisinin tasarımı ve geliştirilmesine odaklanılmıştır **.**

[15] : Makalelerinde Özel şifreleme algoritması tüm XML belgesini tarar. <Encrypt> etiketi altına herhangi bir öğe yerleştirildiğinde, öğe verileri şifrelenir. Şifreleme hem elementte hem de içerik seviyesinde olur**.** Ham şifreleme tüm belgeyi şifreler.

[16] **:** Bu çalışmada, bir android uygulaması ile uygulanmış bir araç takip sistemi arasındaki etkileşimi ve aralarında konumlandırma verisi alışverişini destekleyen bir metodolojiyi desteklemek için bir Web hizmetleri geliştirilmişlerdir. Bu yazıda, zamandan, sistem kaynaklarından tasarruf etmek ve sistem güvenliğini arttırmak amacıyla konumlandırma verilerini optimize etmek için algoritmalar ve çözümler yayınlamaya odaklanmaktadırlar. Çok sayıda tipik modelden bahsederek ve bazı önemli faktörleri değerlendirerek, REST tabanlı modelin uygulamaları için en uygun çözüm olduğunu da belirlenmiştir. Konumlandırma verilerini optimize etmeye yönelik algoritmaları kapsamlı bir şekilde sunmuşlar ve Android akıllı telefonlara dayalı araç izleme sistemi için REST tabanlı ve SOAP tabanlı iki tipik Web servisi modeli arasında karşılaştırma yapmış ve karşılaştırma yapmışlardır.

[17] : Bu makalede, Sağlık Hizmet Birimlerinin en popüler alanlarındaki güvenlik sorunlarını, e-ticaret işlemlerini kapsamaktadır. Sayfa imzası ve güven sıralamasının popüler algoritmalarının karşılaştırılması ve web hizmetlerinde XML imzası ve XML imzası ve bunun doğrulanması ve büyük güvenlik saldırılarının ortaya çıkması üzerine yapılan araştırmayı araştırmak suretiyle web hizmetlerinde XML aracılığıyla daha fazla güvenlik elde etmektedir.

[18]**:** SELS'te şifreleme uygulamasının veri hırsızlığı, yasadışı erişim, veri değişikliği veya uyumu önleyebileceğini ve testte kanıtlanmış olduklarını göstermişlerdir.

[19]: Bu araştırmada, güvenli bir RESTful web hizmeti elde etmek için kimlik tabanlı bir kimlik doğrulama algoritması geliştirilmiştir. Kimlik tabanlı kimlik doğrulama, Boneh-Franklin Kimlik tabanlı şifreleme ve REST URL'sine dayanmaktadır. Bu işlem,

istemcinin tüm durumunu durumsuz REST için saklamak yerine, müşterinin URI'sını kabul ederek müşterinin isteğini yerine getirmesine olanak sağlamıştır. Ayrıca, böyle bir işlem, REST kimlik doğrulaması uygulanarak ortaya çıkan mevcut sorunları çözmektedir.

[20] Bu makalede, Otomatik kompozisyon yöntemlerini değerlendirmek için kullanılabilecek hafif ve ölçeklenebilir bir test platformu için bir mimari önerilmiştir. Bu test platformunun kullanılması, anlamsal web servis kompozisyonunun gelişimini önemli ölçüde geliştirmektedir. Bu test platformunun hafif ve genişletilebilir altyapısı, kompozisyon oluşturma ve eşleştirme algoritması geliştirmeyi kolaylaştırmakta ve test platformunu test etmek için test platformunda e-Mağaza ve seyahat kompozit hizmetleri gibi birçok gerçek dünya örneği geliştirmektedir. Ayrıca, kompozisyon çerçevesindeki hizmetlerin sayısının artmasının performansını önemli ölçüde azaltacağını da bulmuşlardır.

[21] Bu çalışmada, insanların zamanında bir bilgisayar ya da bir ağ sistemine karşı çeşitli reddi hizmet saldırıları tespitini sağlayan bir yöntem sunulmuştur. Ağ paketlerinden çıkarılan istatistiklere anomali tespit tabanlı bir yaklaşım uygulayarak şifreli protokolleri kullanan uygulama katmanı DoS saldırılarının tespitine genel olarak odaklanmakyadır. Ağ trafiğinin şifre çözme ile ilgili etik normları ve gizlilikle ilgili düzenlemeleri ihlal edebileceğinden, önerilen tespit şeması şifre çözme olmadan ağ trafiğini analiz etmektedir. Bu şema, bir web sunucusu ile müşterileri arasındaki konuşmaları analiz etmeyi, bu konuşmaları kümelere bölerek normal bir kullanıcı davranışı modelinin oluşturulmasını ve bu konuşmaları, kümelenen otomatik kodlayıcı yardımı ile sonuçlanan kümeler arasında dağılımının incelenmesini içerir. Bu amaçla derin öğrenme algoritmaları kullanılmıştır. Bu normal örüntülerden sapan müşterilerin konuşmaları anormal olarak sınıflandırılır. Önerilen teknik, gerçekçi bir siber ortamın yardımıyla elde edilen veriler üzerinde test edilmiştir.

[22] Bu çalışmada, IoT'ye RESTful web servisinin dahil edilmesinin faydalarını açıklamakta, ayrıca gerçek dünyadaki kimlik doğrulama problemlerini çözmenin çeşitli yöntemlerini de araştırılmaktadır. Dahası, RESTful web hizmetinde durumsuzluğu tam anlamıyla uygulamak için kimliğe dayalı şifreleme önerilmiştir.

[23] : Bu makalede önerdikleri çözüm, otomatik web hizmeti kompozisyon algoritmalarını uygulamak, test etmek ve değerlendirmek için hafif ve ölçeklenebilir bir test platformudur ve test platformunun test edilmesi ve değerlendirilmesi için bazı senaryolar verilmiştir. Ayrıca,kompozisyon yönteminin performansının, web servislerinin sayısı arttıkça çarpıcı biçimde azalacağını da bulunmuştur.

Fielding'de [24] REST tabanlı bir web hizmetinin sistematik ve anlaşılır bir şekilde nasıl geliştirilebileceği henüz belli değilken, REST mimari tarzını tasarlamak için yapılandırılmış bir yaklaşım sunulmaktadır.Ayrıca, hipermedia'nın uygulama durumunun motoru olarak nasıl kullanılabileceğine dair somut örnekler de yoktur; bunun, REST'in farklı şekilde anlaşılmasının ve uygulanmasının bir nedeni olabileceği düşünülmüştür.

Mulloy [25], "pragmatik REST" konusuna odaklanırken Web API'leri için farklı tasarım ilkeleri ve en iyi uygulamaları sunmaktadır. "Pragmatik REST", sonuçta ortaya çıkan Web API'sinin kullanılabilirliğinin herhangi bir tasarım ilkesinden veya kılavuzundan daha önemli olduğu anlamına gelir. Ancak, bu karar hipermedia gibi REST'in arkasındaki temel kavramların ihmal edilmesine yol açabilir.

Burke [26], RESTful Web Servisleri (JAX-RS) spesifikasyonu için Java API'sine dayanarak web servislerinin nasıl geliştirileceğine dair teknik bir rehber sunmaktadır. Ancak bu çalışma, gerekli geliştirme kararlarının alınması gereken bir web servisinin tasarım aşamasından ziyade uygulama aşamasına odaklanmaktadır.

Vinoski [27], RESTful web servislerini geliştirmek için en iyi uygulamaların bir kontrol listesi sunmuş, yazar açıkça REST'in dağıtılmış hesaplama alanındaki tek cevap olmadığını açıklığa kavuşturmuştur. En iyi uygulamaları, kaynakların temsili gibi RESTful bir web servisinin farklı alanlarını ele alan dört bölümde yapılandırılmıştır.

Tüm açıklamalarına rağmen, çalışmada belirsizliği azaltmak için somut örnekler bulunmamaktadır.

[28] çalışmasında, bibliyografik inceleme yoluyla analiz edilmektedir. Web servislerinin performansı yönlendirilenlerden etkilenir. XML güvenlik özellikleri ve bu bir çözüm öneren bu amaçla bir araya gelerek bu sorunu giderir çeşitli şifreleme algoritmalarının SOAP mesajı oluşturan verileri, verinin gizlilik derecesi,

[29] Bu araştırmada, simetrik ve asimetrik algoritma standartlarının karma ile kombinasyonunu bir çözüm olarak önermişlerdir. Kombinasyon, veri bütünlüğünü sağlamak için SHA256 karma ile hem RSA hem de AES'in özünü içeren bir karma algoritma ile sonuçlanır. Gönderici ve alıcı arasındaki karma şifreleme ve şifre çözme, aynı anda yalnızca meşru verilerin değiş tokuş edilmemesini sağlar. Gönderen tarafı veriler değiştirilmez.

[30] araştırmasında, Web servislerine yönelik saldırıları tespit etmek ve incelemek için petekler kullanılarak bir çözüm önerilmiştir. Honeypots, üretim sistemlerini saldırılara karşı daha iyi korumak için bilgisayar korsanlarının yeni tekniklerini, araçlarını ve motivasyonlarını öğrenmek için kullanılır. Çözüm olarak kullanılan honeypot (WS Honeypot) bir web servis uygulaması olarak dağıtmak içindir. Bu honeypot tüm istek mesajlarını toplar ve analiz eder. Makine öğrenme tekniklerini kullanarak saldırılar tespit edilmektedir.

[31] çalışmasında temel olarak Türkiye'deki bilgi toplumu stratejisini ve e-devlet gelişimini analiz edilmiştir. Değerlendirme raporları ve vatandaş geribildirimlerindeki mevcut ve analiz edilen sayısal verilere dayanarak, sonuçta ortaya çıkan makale, farklı devlet hizmetlerinin entegrasyonu ve düzenlenmesi için kurumlar arası koordinasyonun giderek daha fazla gerekli olduğunu tartışmaktadır. Bu makale ile ayrıca, vatandaş merkezli hizmetlerin gelecekteki gelişimi için daha da kişiselleştirilmesi önerilmektedir.

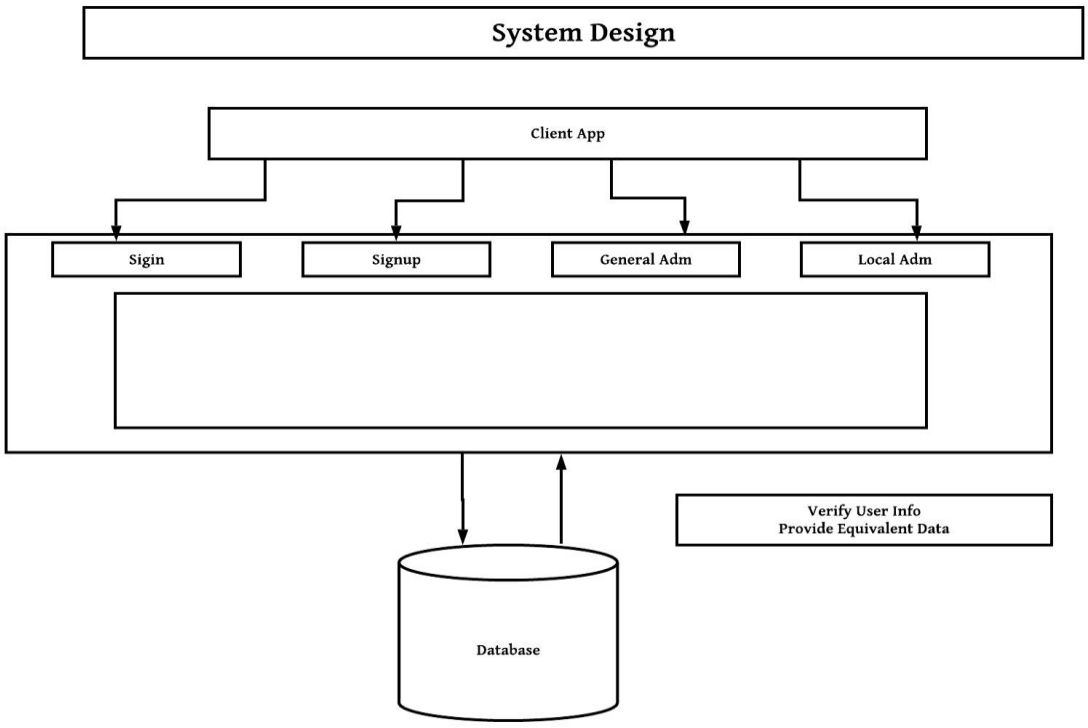
[32] çalışmasına göre e-devlet hizmetlerini değerlendirmek ve optimize etmek devletler için zorunludur. Özellikle e-hizmetlerin kamu idarelerini dönüştürme ve hükümetlerin vatandaşlar, işletmeler ve diğer devlet kurumları ile etkileşmektedir. Yaygın olarak mevcut uygulamalı değerlendirme yaklaşımları vatandaşların memnuniyet tedbirlerini dahil etmeyi ihmal eder. Bu makaleye göre süreçler iki yönlüdür: vatandaş merkezli e-devlet değerlendirme anlayışına katkıda bulunmak ve mevcut kilit performans göstergelerini (KPI'lar) birleştirmek ve bir romanın referans süreç modelini önermek. Bir “know-how” deposu oluşturulmasını kolaylaştırmak için birleşik KPI'ları kullanan değerlendirme yaklaşımları önerilmektedir..

2. BÖLÜM

YÖNTEM VE MATERYAL

Bu bölümde, tez çalışmasında geliştirilen sistemin yönteminde kullanılan malzemeler literatür taramasının önceki bölümünde belirtildiği gibi tartışılacaktır. Bu çalışma esas öncelikle Gine'deki özerk bölgelerin yöneticilerine yardımcı olmak için tasarlanmıştır.

Bu projede kullanılan yazılım içerikleri iki farklı bölüme ayrılır: İlk kısım tamamen Springboot arka ucunu tasarlamak için gereken yazılımla, ikinci kısım ise Frontend ile ön ucu geliştirmek için gereken tüm yazılımlarla ilgilidir.



Şekil 2. 1: Sistemin mimarisi

2 SİSTEMİN MİMARİSİ

Sistem iki bölüme ayrılmıştır:

Ön uç: Angular 6 kullanılarak geliştirilmiştir.

Arka uç: springboot tarafından oluşturulan sunucu tarafını temsil eder.

JSON Web Simgesini (JWT) tanımlandığı kabul edilmektedir.

JSON Web Token (JWT), taraflar arasında bir JSON nesnesi olarak güvenli bir şekilde bilgi aktarımı için kompakt ve kendi kendine yeten bir yol tanımlar.

2.1 JSON Web Jetonları kullanışlıdır:

2.1.1 Yetkilendirme:

JWT kullanımı için en yaygın senaryoyu oluşturur. Tekli Oturum Açma, JWT'yi yaygın olarak kullanan bir özelliktir.

2.1.2 Bilgi Değişimi:

JWT'ler imzalanabildiğinden, JSON Web Belirteçleri taraflar arasında güvenli bir şekilde bilgi iletmenin iyi bir yoludur.

JSON Web Token (JWT), HTTP Yetkilendirme başlıkları ve URL sorgu parametreleri gibi sınırlı çevre alanı için tasarlanan talep formatının bir temsilidir. JWT'ler, Signature JSON Web'den (JWS) veri yükü yapısı olarak kullanılan JSON nesneleri veya JSON Web Şifrelemesi'nden (JWE) düz metin yapısı olarak kullanılan, Mesaj Kimlik Doğrulama Kodu (MAC ile dijital olarak imzalanmasını / korunmasını sağlar) ile oluşturulur [2].

JWT, tam bir noktalama (.) ile ayrılmış üç yapıdan oluşur:

• **Başlık:** Başlıkta iki bölüm vardır, bunlardan biri belirteç türü, yani JWT, diğeri ise HMAC SHA256 veya RSA gibi kullanılan bir karma algoritmadır.

• **Yük taşıma kapasitesi:** İddiadan oluşan belirtecin ikinci kısmıdır. Talep, bir varlık (genellikle kullanıcılar) ve ek meta veriler hakkında yapılan açıklamadır.

• **İmza:** Bir imza oluşturmak için JWT, JWS şartnamesine uymalıdır. JWS, dijital veri veya JSON'u temel veri yapısı olarak kullanan MAC'ler tarafından güvence altına alınan içeriktir [2].

Başlık: HMAC SHA256 algoritması kullanılarak imza oluşturmak için kullanılan çeşitli algoritmalar hakkında bilgilerden oluşur. JSON başlık formatı aşağıdaki gibidir:

{

“Alg”:”HS256” ,

”Type”: ”jwt”

}

Yukarıdaki JSON, base64 algoritmasıyla kodlanacaktır ve yük, talepleri içerir. Talepler işletme ve ek bilgiler hakkında beyanlardır.

3 tür talep vardır:

Kayıtlı hak talepleri,

Genel iddialar,

Özel talepler.

JSON Yükünün formatı aşağıdaki gibidir:

{

"sub": "Saikou",

"iat": 1537603195,

"exp": 1537689595

}

İmza: base64 algoritması kullanılarak birleştirilen ve kodlanan başlık ve yükten oluşturulur, ardından HMAC-256 şifreleme algoritması kullanılarak şifrelenir:

HMACSHA512 (

base64UrlEncode (header) + "." +

base64UrlEncode (payload),

your-256-bit sırrı

)

Hepsini bir araya getirirsek, noktalarla ayrılmış 3 Base64-URL dizgisi elde ederiz.

Bu üç işlemin sonucu birleştirilecek ve JWT ile sonuçlanacaktır. Başlık, bilgi yükü ve imzaya katılmak için sözde kod örneği şöyledir:

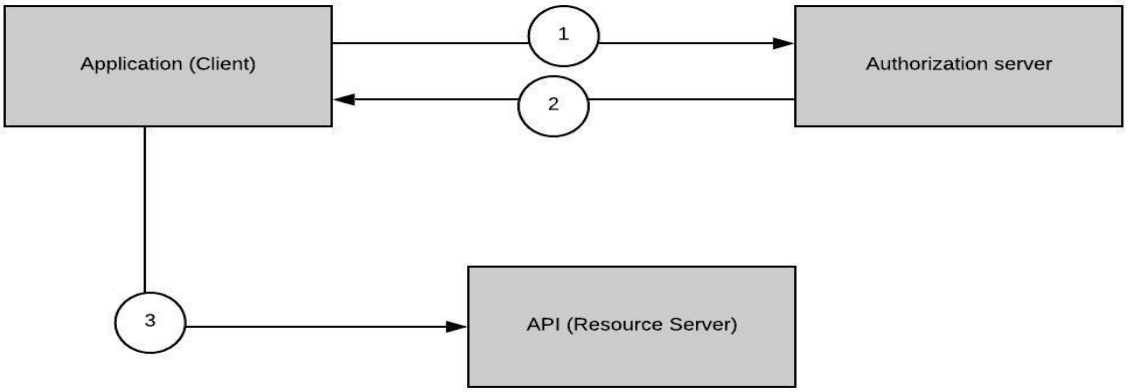
$ token = $ başlık + '.' + $ payload + '.

Buna örnek olarak aşağıdaki gibi bir kod alabiliriz:

"eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJ0aG9tYXNna3oiLCJpYXQiOjE1Mzc2MDMxOTUsImV4cCI6MTUzNzY4OTU5NX0.m2YMjTYmOnfR7nnVNxqCzWbQ2FhKRe1eiizxnC2TF4eAoEzKlwo7PheVkKcxj08ST3vB-ZOIhiORvYVfSgzcog"

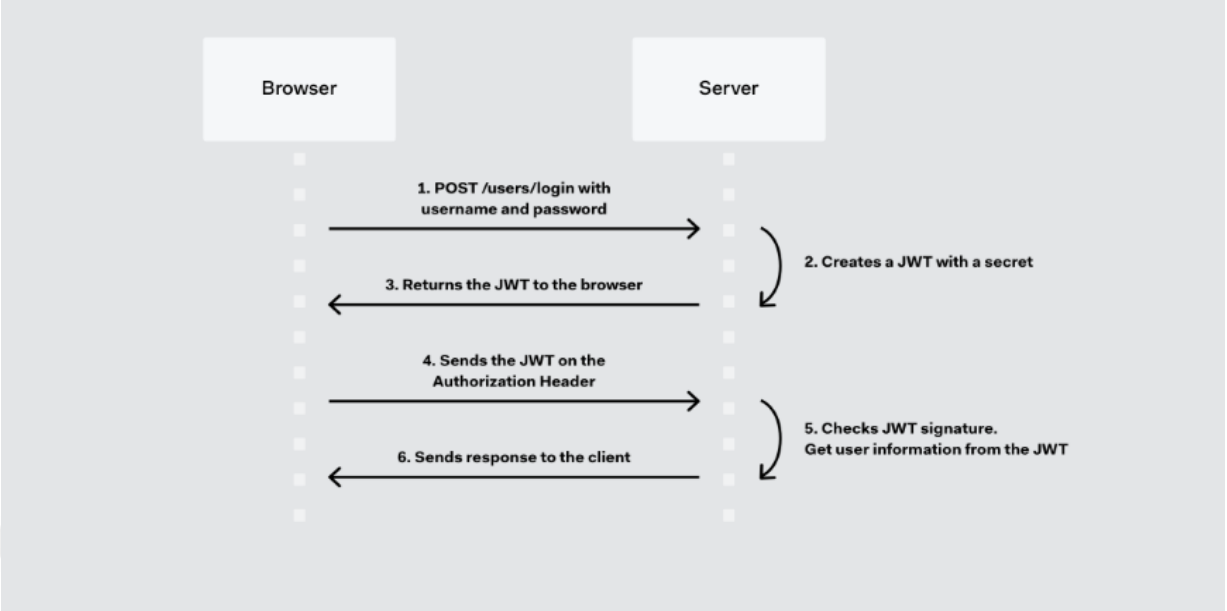
Korunan bir yola veya kaynağa erişirken, kullanıcı aracısı JWT'yi, genellikle taşıyıcı şemasını kullanarak yetkilendirme başlığında göndermelidir. Takip olarak şu kod örnek verilebilir:

Authorization:Bearer
eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJ0aG9tYXNna3oiLCJpYXQiOjE1Mzc2MDMxO
TUsImV4cCI6MTUzNzY4OTU5NX0.m2YMjTYmOnfR7nnVNxqCzWbQ2FhKRe1eii
zxnC2TF4eAoEzKlwo7PheVkKcxj08ST3vB-ZOIhiORvYVfSgzcog



Şekil 2. 2: JWT'nin API'lere veya kaynaklara erişmek için nasıl elde edildiği ve kullanıldığı

Belirteç kullanarak, oturum depolamaya gerek yoktur; belirteç, tüm kullanıcı bilgilerini ileten kendi kendine yeten bir varlıktır. Ayrıca, belirteç durumsuz olduğundan sunucu tarafında ölçeklenebilirlikle uğraşmak kolaydır. Belirteç, tarayıcı ve mobil gibi farklı istemcilere uyarlanmıştır. Bu arada, CORS ve CSRF saldırısından etkili bir şekilde kaçınabilir.



Şekil 2. 3: kullanıcı kaydı ve kullanacı gırısı senarıosu

Tez projesinin amacı, Springboot Security kullanarak arka uçtan açılıya kadar ön uçtan tam yığın uygulama oluşturmaktır. Sistem, genel veya yerel yönetici gibi farklı kullanıcıların sistemde oturum açmasına ve kullanıcı türüne göre vatandaşın bilgilerine erişmesine olanak sağlar.

Genel yönetici tüm vatandaş bilgilerine erişebilir ve yeni vatandaş oluşturma, güncelleme ve silme gibi tüm işlemleri gerçekleştirebilir.

Yerel yönetici, yalnızca o bölge veya bölgeye uygun vatandaşlara erişebilir ve bazı işlemleri genel yönetici olarak yapabilir.

Bu uygulama JWT kimlik doğrulaması ve Spring Security ile güvence altına alınır, daha sonra geçerli kullanıcının rolüne göre (genel veya yerel yönetim sağlanır. Sistem neye erişebileceğine karar verir.

2.2 Teknoloji

Bu çalışmada aşağıdakiler gibi farklı teknolojiler kullanılmaktadır

- Spring boot

- jjwt

- Spring Güvenliği

- Spring JPA

- MySQL

-Angular 6

2.2.1 Spring boot

Spring Boot, bir Mikro Hizmet oluşturmak için kullanılan açık kaynaklı bir Java tabanlı çerçevedir. Pivotal Team tarafından geliştirilmiştir. Spring Boot kullanarak tek başına ve üretime hazır Spring uygulamaları oluşturmak kolaydır. Spring Boot, bir mikro hizmet geliştirmek için kapsamlı bir altyapı desteği içerir ve "yalnızca çalıştırabileceğiniz" kurumsal kullanıma hazır uygulamalar geliştirmenize olanak sağlar.

Bu çerçevede Micro Service (mikro hizmet), Spring Boot ve Spring Security gibi temel kavramları tanımlanmıştır.

Mikro Hizmet, geliştiricilerin hizmetleri bağımsız olarak geliştirmelerine ve dağıtmalarına izin veren bir mimaridir. Çalışan her hizmetin kendi süreci vardır ve bu, iş uygulamalarını desteklemek için hafif bir model oluşturur.

Mikro hizmetler, geliştiricilere aşağıdaki avantajları sunar :

-Kolay dağıtım

-Basit ölçeklenebilirlik

-Konteynerler ile uyumluluk

-Minimum konfigürasyon

-Kısa geliştirme süreleri

Spring Boot, Java geliştiricilerinin, izole olarak çalıştırabileceğiniz tek başına ve üretim düzeyinde bir spring uygulaması geliştirmeleri için iyi bir platform sağlar. Tüm Spring konfigürasyon kurulumuna gerek olmadan minimum konfigürasyonlarla yapabilir.

Spring Boot geliştiricilere aşağıdaki avantajları sunar:

- Spring uygulamalarını anlamak ve geliştirmek kolaydır.

- Verimliliği arttırır.

- Geliştirme süresini kısaltır.

Spring Boot aşağıdaki hedeflerle tasarlanmıştır:

-Üretime hazır bir spring uygulamasını daha kolay bir şekilde geliştirmek,

-Geliştirme süresini kısaltmak ve uygulamayı bağımsız olarak çalıştırmak,

-Uygulamaya başlamanın daha kolay bir yolunun sağlanması.

Spring Boot'u seçmek için bir kaç özellikler ve avantajlar sunmaktadır:

-Java Fasulyelerini, XML yapılandırmalarını ve Veritabanı İşlemlerini yapılandırmak için esnek bir yol sağlar.

-Güçlü bir toplu işlem sağlar ve REST bitiş noktalarını yönetir.

-Spring Boot'ta her şey otomatik olarak yapılandırılır; manuel yapılandırma gerekmez.

-Ek açıklama tabanlı spring uygulaması sunar.

-Katıştırılmış Servlet Kabı içerdiğinden bağımlılık yönetimini kolaylaştırır.

Spring Boot, ek açıklamasını kullanarak projeye eklediğiniz bağımlılıkları temel alarak uygulamanızı otomatik olarak yapılandırır. Örneğin, MySQL veritabanı sınıf yolunuzdaysa ancak herhangi bir veritabanı bağlantısı yapılandırmadıysanız, Spring Boot bir bellek içi veritabanını otomatik olarak yapılandırır.

Spring önyükleme uygulamasının giriş noktası, sınıfta ek açıklamasını ve ana yöntemi içerir.

Spring Boot, ek açıklamasını kullanarak projeye dahil olan tüm bileşenleri otomatik olarak tarar.

Spring boot Başlatıcıları

Bağımlılık yönetimini idare etmek büyük projeler için zahmetli bir iştir. Spring Boot, geliştiricilerin rahatlığı için bir takım bağımlılıklar sağlayan bir çözüm sunmaktadır.

Örneğin, veritabanı erişimi için Spring ve JPA kullanmak istiyorsanız, projenize spring-boot-starter-data-jpa bağımlılığını dahil etmeniz yeterlidir.

Tüm Spring Boot başlatıcılarının aynı isimlendirme modeline sahip olduğunu unutmayın; burada uygulamanın bir tür olduğunu belirtir.

Spring güvenliği, güçlü ve son derece özelleştirilebilir bir kimlik doğrulama ve erişim kontrolü çerçevesidir. Spring tabanlı uygulamaları korumak için fiili bir standarttır.

Spring güvenliği, Java uygulamalarına hem kimlik doğrulama hem de yetkilendirme sağlamaya odaklanan bir çerçevedir. Tüm Spring projeleri gibi, spring güvenliğinin gerçek gücü de özel gereksinimleri karşılamanın ne kadar kolay genişletilebileceğidir

2.2.1.1 Spring Security kullanmanın yararlarından bazıları şunlardır:

Bu yaklaşım kanıtlanmış teknolojidir, dolayısıyla bunu kullanmak tekerleği yeniden icat etmekten daha uygundur. Güvenlik, fazladan özen göstermemiz gereken bir başlıktır, aksi takdirde uygulamamız saldırganlar için savunmasız olacaktır.

CSRF, oturum sabitleme saldırıları gibi bazı genel saldırıları önler.

Herhangi bir web uygulamasına entegrasyonu kolaydır. Web uygulaması yapılandırmalarını değiştirmemize gerek yoktur, Spring otomatik olarak web uygulamasına güvenlik filtreleri enjekte etmektedir.

Kimlik doğrulama için farklı yöntemlerle (bellekteki, DAO, JDBC, LDAP ve daha fazlası) destek sağlar.

Statik HTML, resim dosyaları sunmak için iyi olan belirli URL modellerini yoksayma seçeneği sunar.

Spring Verileri JPA

Spring Verileri ailesinin bir parçası olan Spring Verileri JPA, JPA tabanlı depoları kolayca uygulamayı kolaylaştırır. Bu modül, JPA tabanlı veri erişim katmanları için geliştirilmiş destek ile ilgilidir. Veri erişim teknolojilerini kullanan Spring destekli uygulamalar oluşturmayı kolaylaştırır.

Bir uygulamanın veri erişim katmanını uygulamak bir süredir zahmetli olmuştur. Basit sorgular yapmak, sayfalama yapmak ve denetim yapmak için çok fazla kazan plakası kodu yazılmalıdır. Spring Verileri JPA, gerçekte ihtiyaç duyulan miktara gösterilen çabayı azaltarak veri erişim katmanlarının uygulanmasını önemli ölçüde iyileştirmeyi amaçlamaktadır. Bir geliştirici olarak, özel bulucu yöntemleri de dahil olmak üzere depo arabirimlerinizi yazarsınız ve Spring, uygulamayı otomatik olarak sağlar.

-Spring ve JPA'ya dayanan depolar oluşturmak için gelişmiş destek verir.

-Querydsl desteği, yazı tipi güvenli JPA sorgularını tahmin eder.

-Etki alanı sınıfının şeffaf denetimi sağlanır.

-Sayfalandırma desteği, dinamik sorgu yürütme, özel veri erişim kodunu entegre etme yeteneği sağlanır.

-Sorgu açıklamalı sorguların önyükleme zamanında doğrulanması mümkündğr.

-XML tabanlı varlık eşlemesi desteği tanıtımıyla JavaConfig tabanlı depo yapılandırması mevcuttur.

Bootstrap

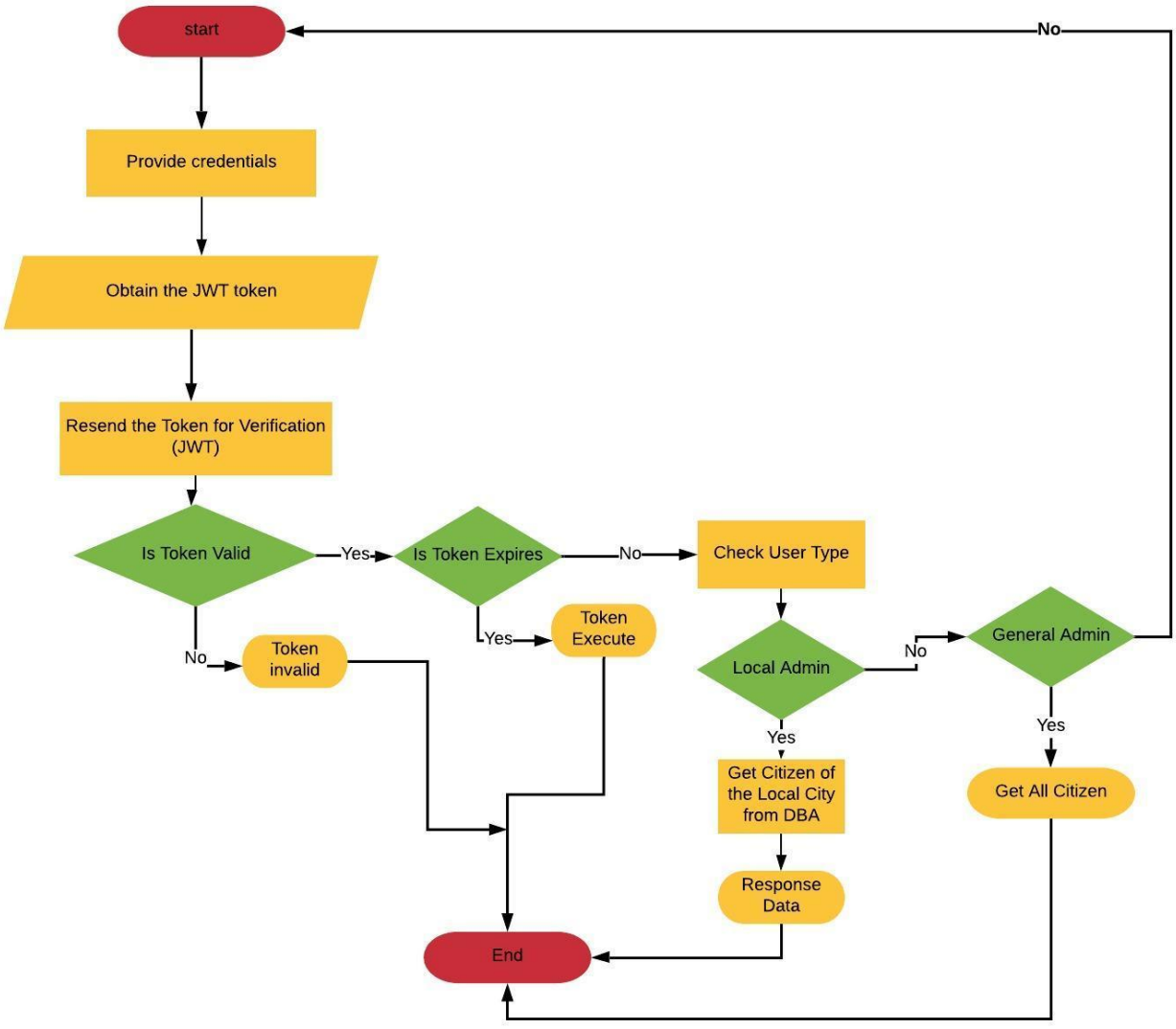
Bootstrap, ücretsiz ve açık kaynaklı bir ön uç Web çerçevesidir. İsteğe bağlı JavaScript uzantıları yanı sıra, tipografi, formlar, düğmeler, gezinme ve diğer arabirim bileşenleri için HTML ve CSS tabanlı tasarım şablonları içerir. Daha önceki birçok web çerçevesinin aksine, yalnızca ön uç geliştirme ile ilgilidir.

2.2.2 MySQL

Birçok sistemin büyük miktarda veri üretmesi, işlenmesi ve analiz edilmesi için bir veritabanı sistemi gerekir. MySQL, en popüler açık kaynaklı SQL veritabanı yönetim sistemlerinden biridir. MySQL, tabloların belirli kriterlere göre birbirleriyle ilişkili olduğu ilişkisel bir veritabanını destekler.

2.2.3 Angular 6

Angular 6, JavaScript'in bir üst kümesi olan JavaScript, Html ve TypeScript'te web uygulamaları ve uygulamaları oluşturmak için bir JavaScript çerçevesidir. Angular, animasyon, Http hizmeti ve sırayla otomatik tamamlama, gezinme, araç çubuğu, menüler vb. gibi özelliklere sahip malzemeler için yerleşik özellikler sağlar.



Şekil 2. 4: Sistemin akış diyagramı

İşlem, kimlik bilgisi JWT'yi aldıktan sonra JWT belirteci almak doğru ise, kimlik bilgisi kullanıcı adı ve şifresini sağlamakla başlar, belirteç doğrulama için yeniden gönderilir. Her şeyden önce, sunucu sağlanan bilgilerin veritabanında olup olmadığını kontrol eder. Bilgiler veritabanında yanlışsa veya kullanılamıyorsa (kullanıcının veya kayıtlı olması durumunda), sunucu yetkisiz bir erişim mesajı gönderir, çünkü kayıtlı olmayan kullanıcı kaynaklara erişemez, ardından token geçerliyse, sunucu doğrulanmışsa sunucu doğrulanır. Geçerli bir işlem otomatik olarak bitmez. Belirteç geçerliyse, sunucu belirtecin süresi dolmadığında ve belirtecin süresi dolduğunda doğrulanmış olan sunucu doğrulanır ve işlem doğrudan sona erer. Sunucu yoksa, genel yönetici veya yerel yönetici olarak tercih edilen kullanıcı türünü kontrol edin, kaynaklara rollerine göre erişebilir.

Başarılı bir giriş işleminden sonra, kullanıcı vatandaş işlemlerini farklı işlemlerle yönetme olanağına sahiptir. Bu işlemler sunucu tarafından Restfull servisleri olarak verilmektedir.

3. BÖLÜM

UYGULAMA DETAYLARI

İkinci bölümde, sistemi geliştirmek için kullanılan bileşenler ve metodoloji tartışıldıktan sonra, bu bölümün son bölümünde sistemle etkileşimin uygulama ve farklı senaryolarından bahsedilmiştir. Bu bölüm, uygulamanın prototipinin detaylarına odaklanmaktadır.

3 PROTOTİPİN TASARIM VE UYGULAMASI

Bu, 3 katmana ayrılan Spring Security / JWT sınıfları için şemadır:

3.1.1 HTTP

Köprü Metni Aktarım Protokolü (HTTP), HTML gibi hiper medya belgelerini aktarmak için bir uygulama katmanı protokolüdür. Web tarayıcıları ve web sunucuları arasındaki iletişim için tasarlanmıştır, ancak başka amaçlar için de kullanılabilir. HTTP, klasik bir istemci-sunucu modelini izler, müşteri bir istek yapmak için bir bağlantı açar ve ardından bir cevap alana kadar bekler. HTTP, durumsuz bir protokoldür, yani sunucunun iki istek arasında herhangi bir veri (durum) tutmadığı anlamına gelir. Genellikle bir TCP / IP katmanını temel almasına rağmen, herhangi bir güvenilir taşıma katmanında kullanılabilir; yani, UDP gibi mesajları sessizce kaybetmeyen bir protokoldür.

3.1.2 Spring Güvenliği

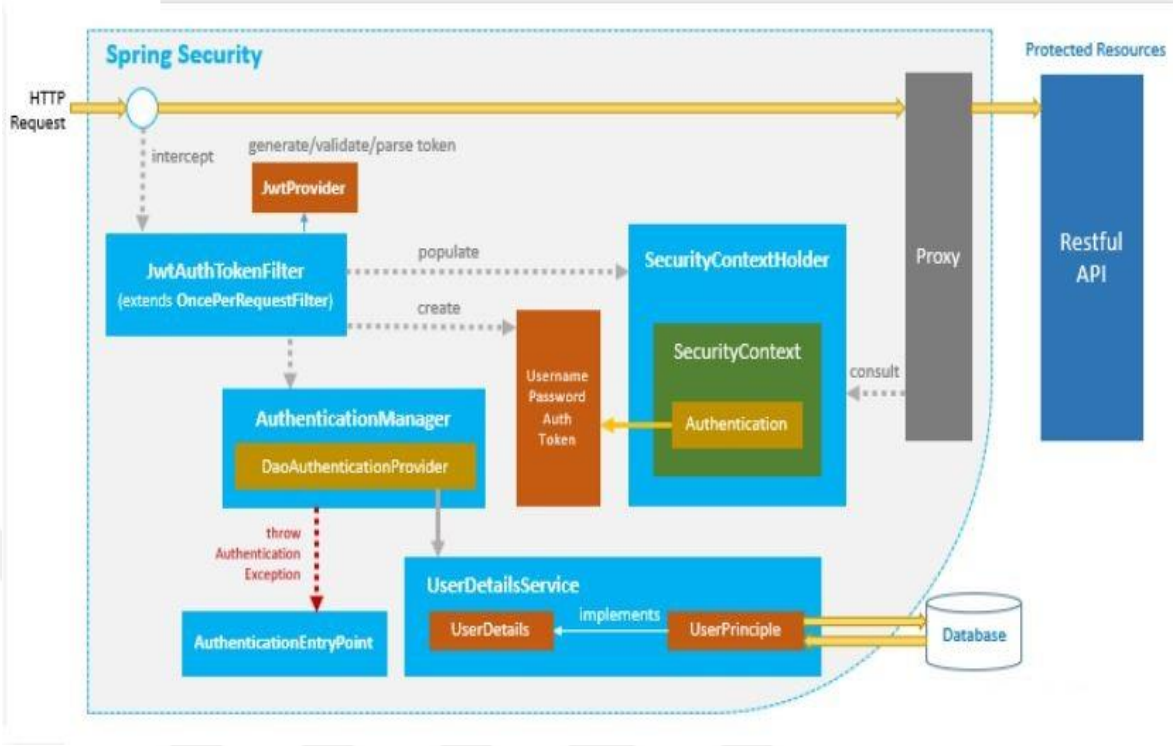
Spring Security, güçlü ve son derece özelleştirilebilir bir kimlik doğrulama ve erişim kontrolü çerçevesidir. Spring tabanlı uygulamaları korumak için fiili bir standarttır. Spring Security, Java uygulamalarına hem kimlik doğrulama hem de yetkilendirme sağlamaya odaklanan bir çerçevedir. Tüm Spring projeleri gibi, Spring güvenliğinin gerçek gücü de özel gereksinimleri karşılamanın ne kadar kolay genişletilebileceğidir.

3.1.3 REST API

Bir API, bir program veya web sitesinin diğerleriyle konuştuğu bir arayüzdür. Veri ve hizmetleri paylaşmak için kullanılırlar ve birçok farklı format ve türde gelirler.

Basitlik için HTTP kullanmaya ek olarak, REST çeşitli avantajlar sunar.

REST, daha geniş çeşitlilikte veri formatlarına izin verir.
JSON ile birleştiğinde (genellikle verilerle daha iyi çalışır ve daha hızlı ayrıştırma sunar), REST genellikle birlikte çalışmanın daha kolay olduğu kabul edilir.
REST, tarayıcı müşterileri için daha iyi destek sunar.
REST, özellikle değiştirilmemiş ve dinamik olmayan bilgileri önbelleğe alarak üstün performans sağlar.
REST, Yahoo, Ebay, Amazon ve hatta Google gibi büyük hizmetler için en sık kullanılan protokoldür.
REST genellikle daha hızlıdır ve daha az bant genişliği kullanır. Site altyapısını yeniden yapılandırmaya gerek kalmadan mevcut web sitelerine entegrasyon da kolaydır. Bu, geliştiricilerin bir siteyi sıfırdan yeniden yazmak için zaman harcamak yerine daha hızlı çalışmasını sağlar. Bunun yerine, ek işlevsellik ekleyebilirler.
RESTful API, programların, sunucuların ve web sitelerinin veri ve hizmetleri paylaşabilmesinin olası yollarından biridir. REST (Temsili Durum Aktarımı), veri ve hizmetlerin API aracılığıyla nasıl temsil edildiğine ilişkin genel kuralları açıklar; böylece diğer programlar bir API'nin sağladığı veri ve hizmetleri doğru şekilde talep edebilir ve alabilir.



Şekil 3. 1: Spring boot Güvenliği

- ✓ SecurityContextHolder, SecurityContext'e erişim sağlar.
- ✓ SecurityContext, Kimlik Doğrulama ve muhtemelen talebe özgü güvenlik bilgilerini tutar.
- ✓ Kimlik doğrulama, bir yöneticiye verilen uygulama genelindeki izinleri yansıtan, GrantedAuthority'yi içeren müdürü temsil eder.
- ✓ UserDetails, DAO'lardan veya diğer güvenlik verileri kaynağından bir Kimlik Doğrulama nesnesi oluşturmak için gerekli bilgileri içerir.
- ✓ UserDetailsService, String tabanlı bir kullanıcı adından bir UserDetails oluşturmaya yardımcı olur ve genellikle AuthenticationProvider tarafından kullanılır.
- ✓ JwtAuthTokenFilter (OncePerRequestFilter öğesini genişletir) Token'den HTTP isteğini işler, Token'den Kimlik Doğrulama oluşturur ve SecurityContext'e doldurur.

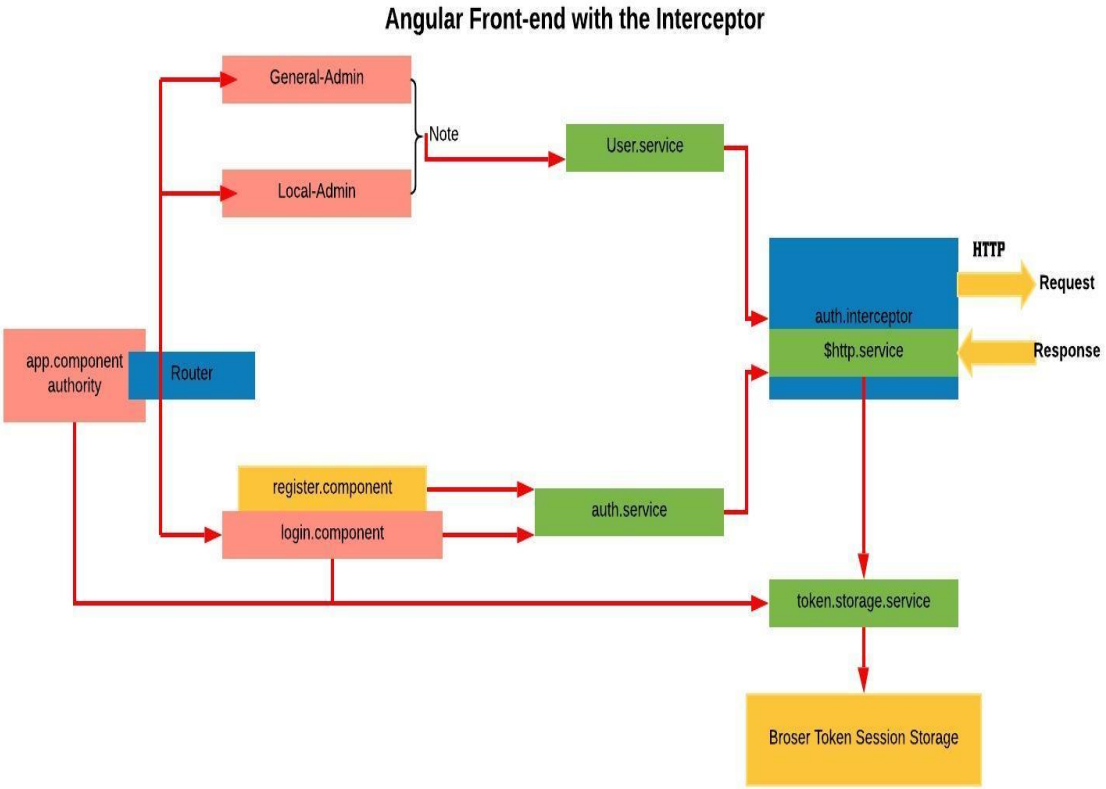
- ✓ JwtProvider, token String'i doğrular, ayrıştırır veya UserDetails öğesinden token String'i oluşturur.
- ✓ UsernamePasswordAuthenticationToken, oturum açma isteğinden kullanıcı adı / parola alır ve bir Authentication arabirimi örneği oluşturur.
- ✓ AuthenticationManager, UsernamePasswordAuthenticationToken örneğini doğrulamak için DaoAuthenticationProvider (UserDetailsService & PasswordEncoder yardımıyla) kullanır, ardından başarılı kimlik doğrulaması için tam doldurulmuş bir Kimlik Doğrulama örneği verir.
- ✓ SecurityContext, yukarıda belirtilen kimlik doğrulama nesnesini içeren SecurityContextHolder.getContext (). SetAuthentication (…) öğesini arayarak oluşturulur.
- ✓ AuthenticationEntryPoint, AuthenticationException'ı işler.
- ✓ Restful API'ye erişim HTTPSecurity tarafından korunur ve Method Security Expressions ile yetkilendirilir.

Bir Taşıyıcı Token, kullanan istemciler için herhangi bir anlamı olmayan bir opak dizedir. Bazı sunucular kısa onaltılık karakter dizisi olan belirteçleri verirken, diğerleri JSON Web Belirteçleri gibi yapılandırılmış belirteçleri kullanabilir. Taşıyıcı kimlik doğrulaması (belirteç doğrulaması olarak da bilinir), taşıyıcı belirteçleri adı verilen güvenlik belirteçlerini içeren bir HTTP kimlik doğrulama şemasıdır. "Taşıyıcı kimlik doğrulaması" adı "bu belirtecin taşıyıcısına erişim izni ver" olarak anlaşılabilir. Taşıyıcı belirteç, genellikle bir oturum açma isteğine yanıt olarak sunucu tarafından oluşturulan şifreli bir dizedir. Korunan kaynaklara istekte bulunurken, istemci bu kodu Yetkilendirme başlığında göndermelidir.

Bearer kimlik doğrulama şeması başlangıçta RFC 6750'de OAuth 2.0'ın bir parçası olarak oluşturuldu, ancak bazen kendi başına da kullanılmaktadır. Temel kimlik doğrulamaya benzer şekilde, taşıyıcı kimlik doğrulaması yalnızca HTTPS (SSL) üzerinden kullanılmalıdır.

Taşıyıcı kimlik doğrulaması, type: http and scheme: bearer türündeki bir güvenlik şemasıdır. Önce güvenlik şemasını bileşenler / securitySchemes altında tanımlamanız gerekmektedir. İsteğe bağlı bearerFormat, taşıyıcı belirtecinin nasıl biçimlendirileceğini belirten isteğe bağlı bir dizedir. Taşıyıcı belirteçleri genellikle sunucu tarafından üretildiğinden, bearerFormat, temel olarak istemcilere ipucu olarak belgeleme amacıyla kullanılır. Yukarıdaki örnekte, JSON Web Token anlamına gelen "JWT" dir. [ ] BearerAuth öğesindeki köşeli parantez: [ ], API çağrıları için gereken güvenlik kapsamlarının bir listesini içerir. Liste boştur, çünkü kapsamlar yalnızca OAuth 2 ve OpenID Connect ile birlikte kullanılır. Yukarıdaki örnekte, Bearer kimlik doğrulaması genel olarak tüm API'ye uygulanır. Yalnızca birkaç işleme uygulamanız gerekirse, bunu genel olarak yapmak yerine işlem düzeyinde güvenlik eklemektedir.

Sistemin uygulanması



Şekil 3. 2: Angular ön ucu

- ✓ app.component yönlendirici için yönlendiriciLink ve yönlendirici-çıkışını içeren ana bileşendir. Ayrıca gezinti çubuğunda öğeleri görüntülemek için koşul olarak bir yetki değişkenine sahiptir.

- ✓ local.admin.component, general.admin.component, Kullanıcı Kurulu, Yönetici Kurulu için Açısal Bileşenlere karşılık gelir. Her Kurul yetkili otorite verilerine erişmek için user.service kullanır.

- ✓ register.component Kullanıcı Kayıt formunu içerir, formun teslimi auth.service'i arayacaktır.

- ✓ login.component Kullanıcı Girişi formunu içerir, formun teslimi auth.service ve token-storage.service'i arayacaktır.

- ✓ user.service, Angular HttpClient ($ http service) kullanarak Sunucu'dan yetki verilerine erişir.

- ✓ auth.service, Angular HttpClient ($ httpservice) kullanarak Sunucu ile kimlik doğrulama ve kaydolma işlemlerini işler.

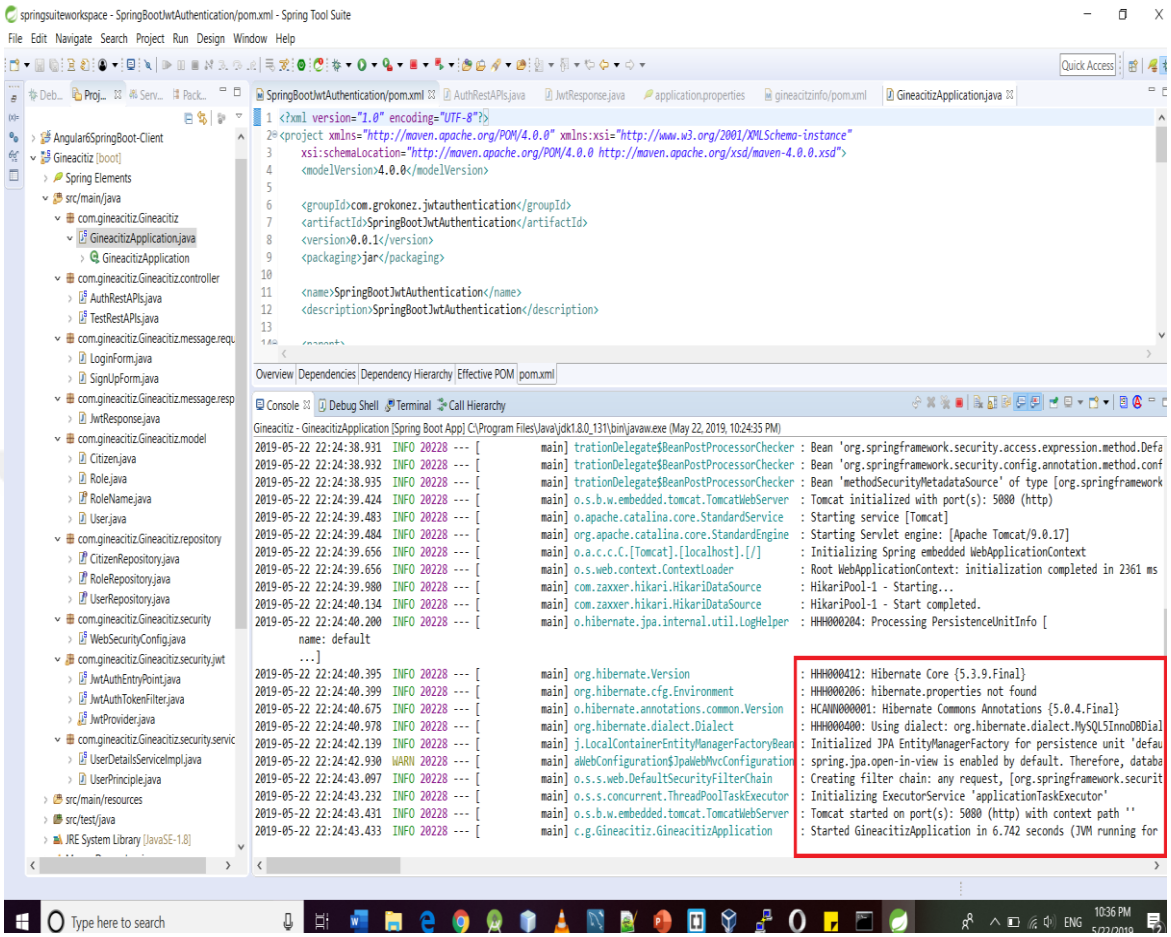
- ✓ $ http hizmeti tarafından yapılan her HTTP isteği, Auth-Interceptor (Sunucuya uygular) tarafından Sunucuya gönderilmeden önce denetlenecek ve dönüştürülecektir.

- ✓ auth-interceptor, Token'i HTTP İsteklerinin Yetkilendirme Başlığına eklemek için token-storage.service'den Token'i kontrol eder..

- ✓ token-storage.service, Tarayıcı'nın sessionStorage içindeki Token'i yönetir.

3.2 Sistemin arka uç bileşeni

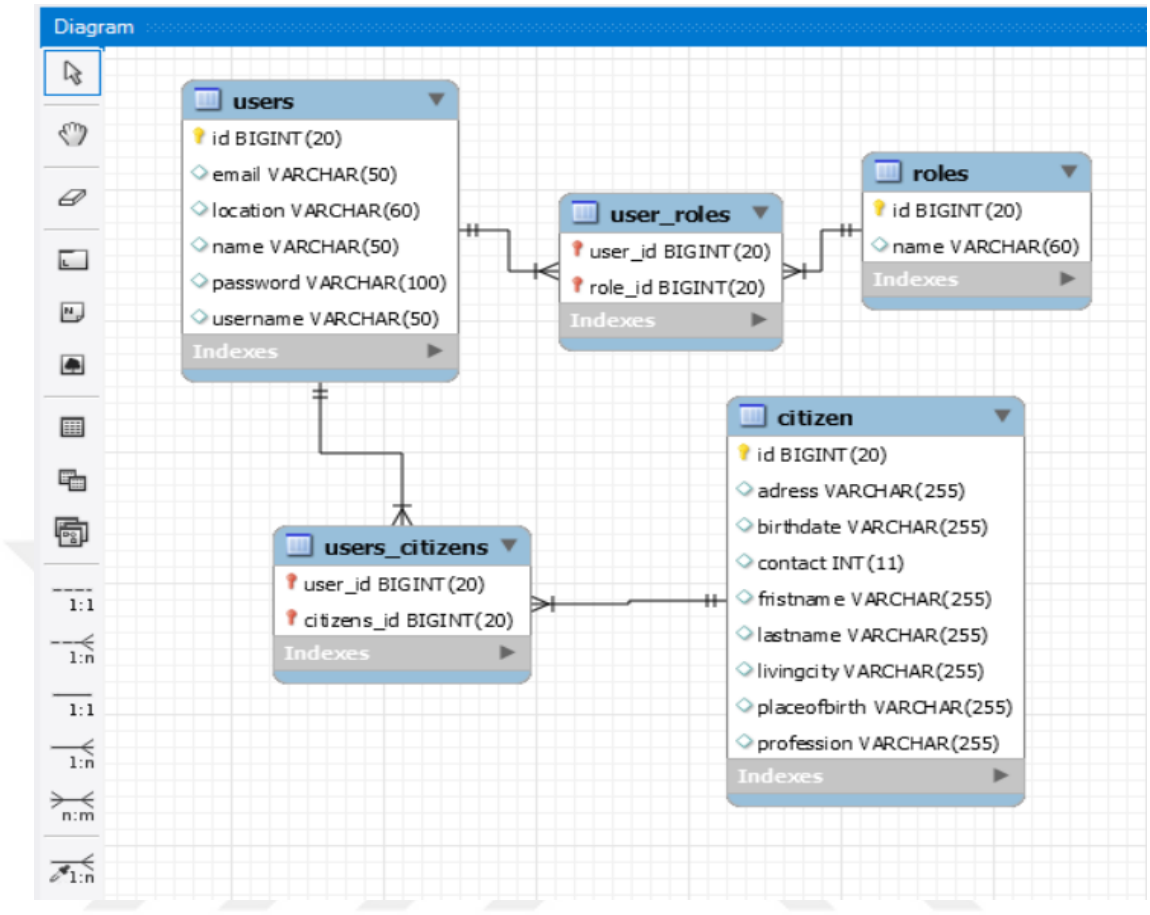
Sistem hakkındaki teorik detaylar bir önceki bölümde hazırlandıktan sonra, bu kısım sistemin prototip işlevselliği ile ilgili tüm pratik detayları ortaya koyacaktır. Şekil 3.3'te gösterildiği gibi, sistemin arka uç sunucusu Spring boot yazılımı kullanılarak gerçekleştirilmiştir. Görüntü arka planda çalışan sunucuyu gösterir. Ayrıca, uygulamanın sol tarafında, sistemin arka ucunda yer alan tüm sınıfları gösterir. Gerçekleştirildiği üzere, kırmızı dikdörtgen kullanılan sunucu türü (Tomcat), sunucunun kullandığı bağlantı noktası, başlatma zamanı ve GinieacitizApplication vb.



Şekil 3. 3: Arka uç sunucusu (Tomcat) çalışıyor

3.3 Sistemin veri yapısı

Sistem ile etkileşim verilerini depolamak ve yönetmek için MySQL tezgah kullanarak bir veri tabanı geliştirilmiştir. Şekil 3.4, kullanıcıların ve vatandaşların verilerinin yönetimi için tasarlanmış veritabanının mimari diyagramını göstermektedir.



Şekil 3. 4: veritabanı diyagramı

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id	email	location	name	password	username
1	Malik@gmail.com	Konakrv	Malik	$2a$10$P5V92RD2oi/X3LvCTaCwk.AaE0tGfxni5ob8//.BMni24ImuiX5JG	Malik
2	avinde@gmail.com	istanbul	MALIKI MOUSTAPHA	$2a$10$.k6eK9uVwNZatititzExLSOBC1fPwZ1OiEauAnSwmokaBBviNn9G7O	avindemalik
3	saiko@gmail.com	Konacrv	Saiko Diallo	$2a$10$2ZbrVwIzAtRCaa6oCfSGcualkxTi.nr4OixwibRH1aHDSCWSYTeYC	saiko
4	mvemail@gmail.com	kavseri	mvname	$2a$10$7MG3luLAEuh876bKZSYzo.7MvmEFxtRtdGJm1Dta9z6TEEcoEWNOS	mvusername
5	saiko2@gmail.com	Konacrv	Saiko Diallo	$2a$10$w25P7S.adTTFt68Yu5r9Oev55/u6zBFOkTRiNdarue5BhoL6oiHnu	saiko2
6	memail@gmail	location	mvname	$2a$10$uahVYTCOma2EVvoOdFRSkuvY0WGdC4BwPPUZYt9/V.9ftaIIcfasC	mvusename
7	memail2@gmail	location2	mvname2	$2a$10$6SB7/O7Rt7i/hnFXfN.4HOGV40KchPv/oktoH32AiX7CasOIR8KoW	mvusename2
8	theemail@gmail.com	thelocation	thename	$2a$10$cO8h5i.1uaCCHkfZmVtx/eOWnIkSbroLIIfOKOiOuZimUB24AiRHO	theusername
9	saikodiallo@gmail.com	Konakrv	Saiko DIALLO	$2a$10$dVeCMfL12h6SeXaT8ICDuuAaifzaoGzhUXPPrfvOhES.V0VlUwmNW	saiko-diallo
10	avindemalik@gmail.com	Konakrv	MALIKI MOUSTAPHA	$2a$10$4AAWWJNTUBmVSf91eOP/7.sGLa9HsTVv9r59nEvw6oAivfCdDmctO	avinde-malik
NULL	NULL	NULL	NULL	NULL	NULL

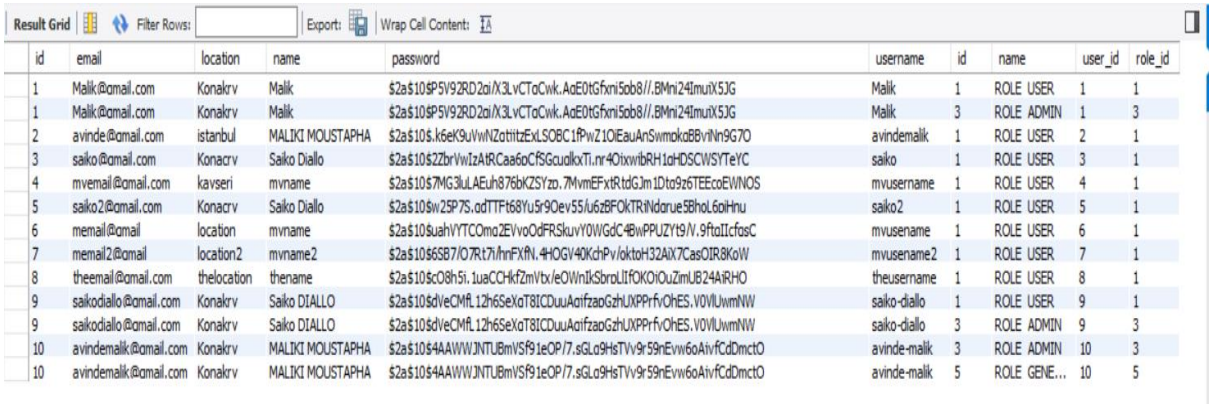
Şekil 3. 5: tüm kullanıcılar tablosu

Result Grid | Filter Rows: | Edit:

id	name
3	ROLE_ADMIN
5	ROLE_GENERAL_ADMIN
4	ROLE_LOCAL_ADMIN
2	ROLE_PM
1	ROLE_USER
NULL	NULL

Şekil 3. 6:Roller tablosu

Önceki bölümde tarif edildiği gibi, şekil 3.6, tüm kullanım bilgilerini ve ilişkili rollerini gösterir.

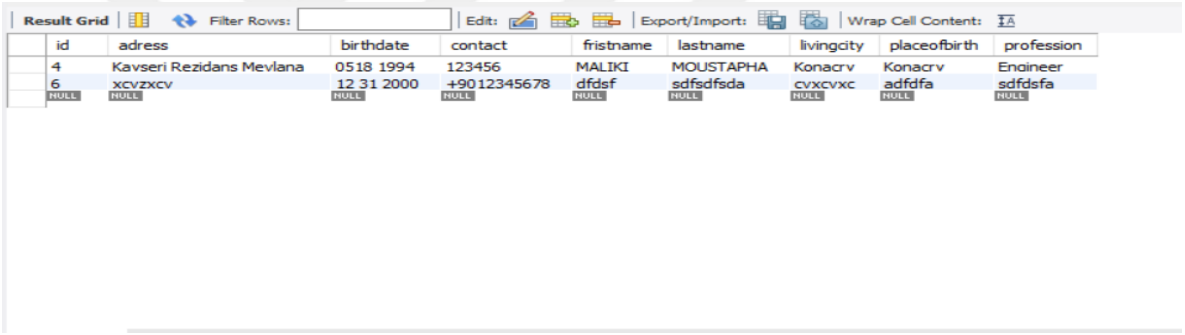


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

id	email	location	name	password	username	id	name	user_id	role_id
1	Malik@gmail.com	Konakrv	Malik	$2a$10$P5V92RD2ai/X3LvCTaCwk.AaE0tGfxni5ob8//.BMni24ImuiX5JG	Malik	1	ROLE USER	1	1
1	Malik@gmail.com	Konakrv	Malik	$2a$10$P5V92RD2ai/X3LvCTaCwk.AaE0tGfxni5ob8//.BMni24ImuiX5JG	Malik	3	ROLE ADMIN	1	3
2	avinde@gmail.com	istanbul	MALIKI MOUSTAPHA	$2a$10$.k6eK9uVwNZatitzExLSOBC1fPwZ1OiEauAnSwmokaBBviNn9G7O	avindemalik	1	ROLE USER	2	1
3	saiko@gmail.com	Konacrv	Saiko Diallo	$2a$10$2ZbrVwIzAtRCaa6oCfSGcudkxTi.nr4OtxwibRH1aHDSCWSYTeYC	saiko	1	ROLE USER	3	1
4	mvemail@gmail.com	kavseri	mvname	$2a$10$7MG3luLAEuh876bKZSYzo.7MvmEFxtRtdGJm1Dta9z6TEEcoEWNOS	mvusername	1	ROLE USER	4	1
5	saiko2@gmail.com	Konacrv	Saiko Diallo	$2a$10$w25P7S.adTTFt68Yu5r9Oev55/u6zBFOkTRtNdarue5BhoL6oiHnu	saiko2	1	ROLE USER	5	1
6	memail@gmail	location	mvname	$2a$10$uahVYTCOma2EVvoOdFRSkuvY0WGdC4BwPPUZYt9/V.9ftaIIcfasC	mvusename	1	ROLE USER	6	1
7	memail2@gmail	location2	mvname2	$2a$10$6SB7/O7Rt7i/hnFXfN.4HOGV40KchPv/oktoH32AiX7CasOIR8KoW	mvusename2	1	ROLE USER	7	1
8	theemail@gmail.com	thelocation	thename	$2a$10$cO8h5i.1uaCCHkfZmVtx/eOWnIkSbroLIfOKOiOuZmUB24AiRHO	theusername	1	ROLE USER	8	1
9	saikodiallo@gmail.com	Konakrv	Saiko DIALLO	$2a$10$dVeCMfL12h6SeXaT8ICDuuAaifzapGzhUXPPrfvOhES.V0VlUwmNW	saiko-diallo	1	ROLE USER	9	1
9	saikodiallo@gmail.com	Konakrv	Saiko DIALLO	$2a$10$dVeCMfL12h6SeXaT8ICDuuAaifzapGzhUXPPrfvOhES.V0VlUwmNW	saiko-diallo	3	ROLE ADMIN	9	3
10	avindemalik@gmail.com	Konakrv	MALIKI MOUSTAPHA	$2a$10$4AAWWJNTUBmVSf91eOP/7.sGLa9HsTVv9r59nEvw6oAivfCdDmctO	avinde-malik	3	ROLE ADMIN	10	3
10	avindemalik@gmail.com	Konakrv	MALIKI MOUSTAPHA	$2a$10$4AAWWJNTUBmVSf91eOP/7.sGLa9HsTVv9r59nEvw6oAivfCdDmctO	avinde-malik	5	ROLE GENE...	10	5

Şekil 3. 7:Kullanıcı rolleri tablosu

Şekil 3. 8: sisteme kayıtlı tüm vatandaşlarla ilgili bilgileri sunmaktadır.



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id	adress	birthdate	contact	fristname	lastname	livingcity	placeofbirth	profession
4	Kavseri Rezidans Mevlana	0518 1994	123456	MALIKI	MOUSTAPHA	Konacrv	Konacrv	Engineer
6	xcvzxcv	12 31 2000	+9012345678	dfdsf	sdfsdfsda	cvxcvxc	adfdfa	sdfdsfa
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

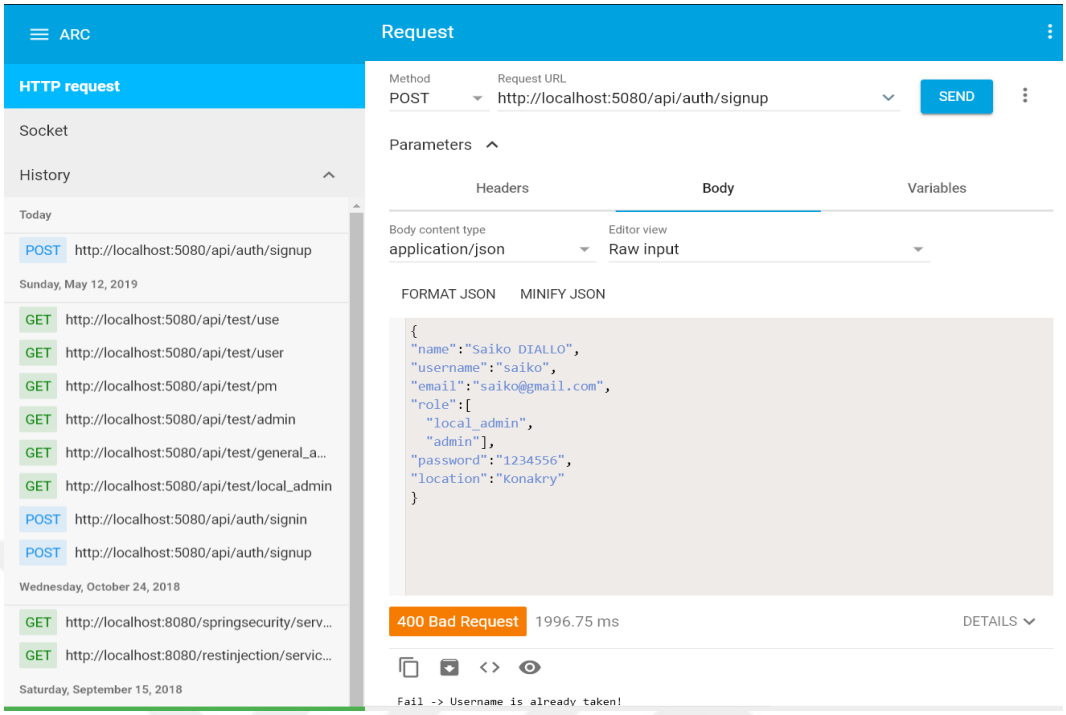
Tablo 3. 1: Bütün vatandaşlar tablosu

3.4 Arka uç testi işlem testi

Sunucuyla etkileşime geçmek için google chrome Advanced REST istemcisi, önerilen Api'nin kaydolma, oturum açma vb. Gibi özelliklerini test etmek için kullanılmıştır.

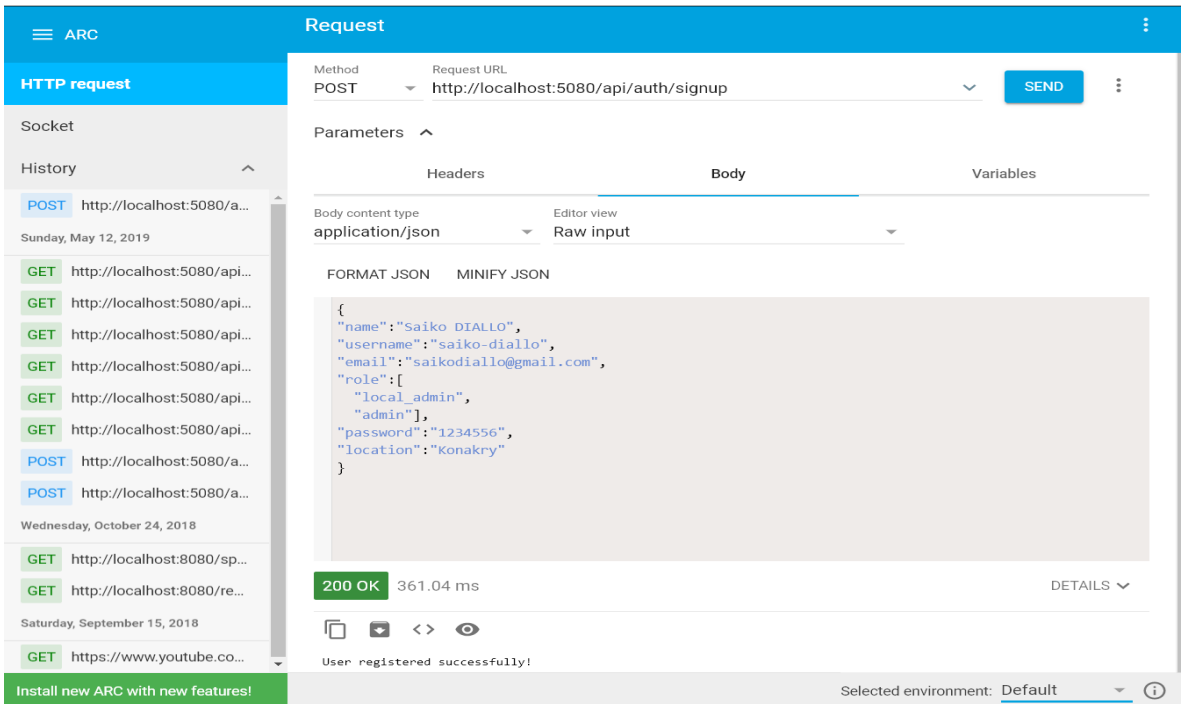
3.4.1 Giriş senaryosu

Herhangi bir kullanıcı türü (yerel yönetici veya genel yönetici) sisteme giriş yapabilmek için önce kaydolmalıdır. Şekil 3.10 ve şekil 3.10, kullanıcının tam adı, kullanıcı adı, şifre, kullanıcı türü ve konum gibi kayıt için gerekli tüm bilgileri gösterir. Gerçekleştirilebileceği gibi, şekil 3.10, kullanıcı bir kullanıcı adı veya başka bir kullanıcı tarafından zaten alınmış bir e-posta göndermeye çalıştığında başarısız bir kayıt görüntüler.



Şekil 3. 9: başarısız kayıt

Ancak, diğer yandan şekil 3.11, gerekli tüm ve doğru bilgilerin verilmesi durumunda başarılı bir kullanıcı kaydı görüntüler.

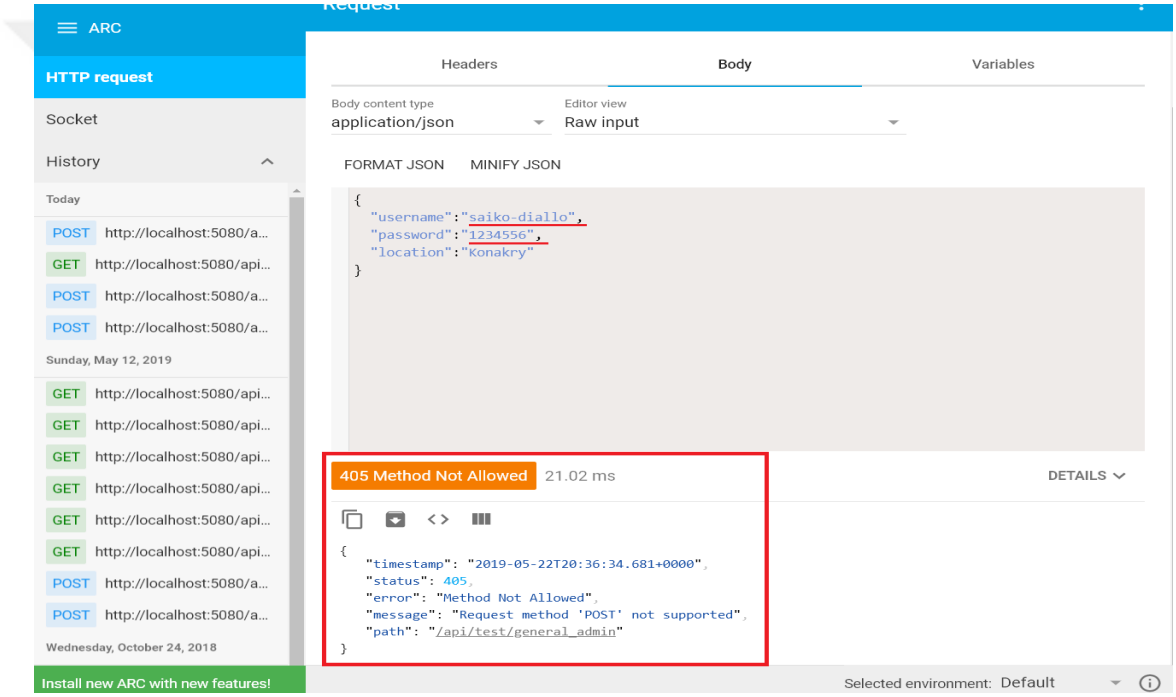


Şekil 3. 10:başarılı kayıt

3.4.2 Giriş yap senaryosu

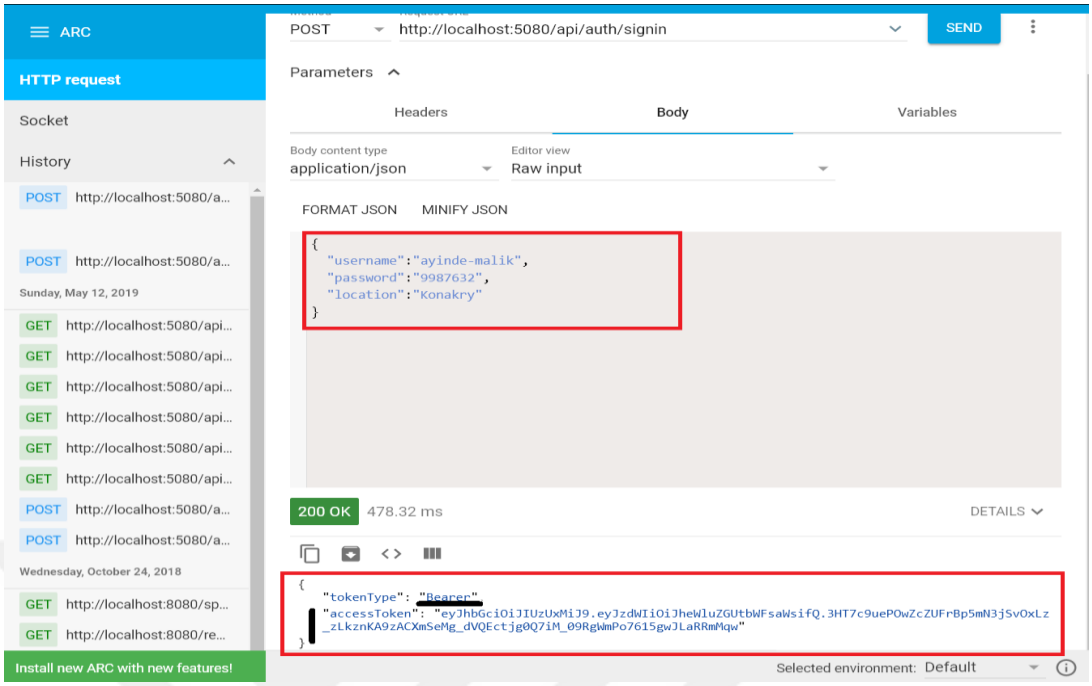
Bu işlem, metodoloji bölümünde açıklandığı gibi JWT belirteci alma senaryosuyla başlar. Şekil 3.12, gerekli kullanıcı bilgilerini (kullanıcı adı ve şifre) gönderdiğimiz giriş işlemini göstermektedir.

Her şeyden önce, sunucu sağlanan bilgilerin veri tabanında olup olmadığını kontrol eder. Bilgiler veritabanında yanlışsa veya kullanılamıyorsa (kullanıcının veya kayıtlı olması durumunda), sunucu kayıtlı olmayan bir kullanıcının kaynaklarına erişemediğinden yetkisiz bir erişim mesajı gönderir.

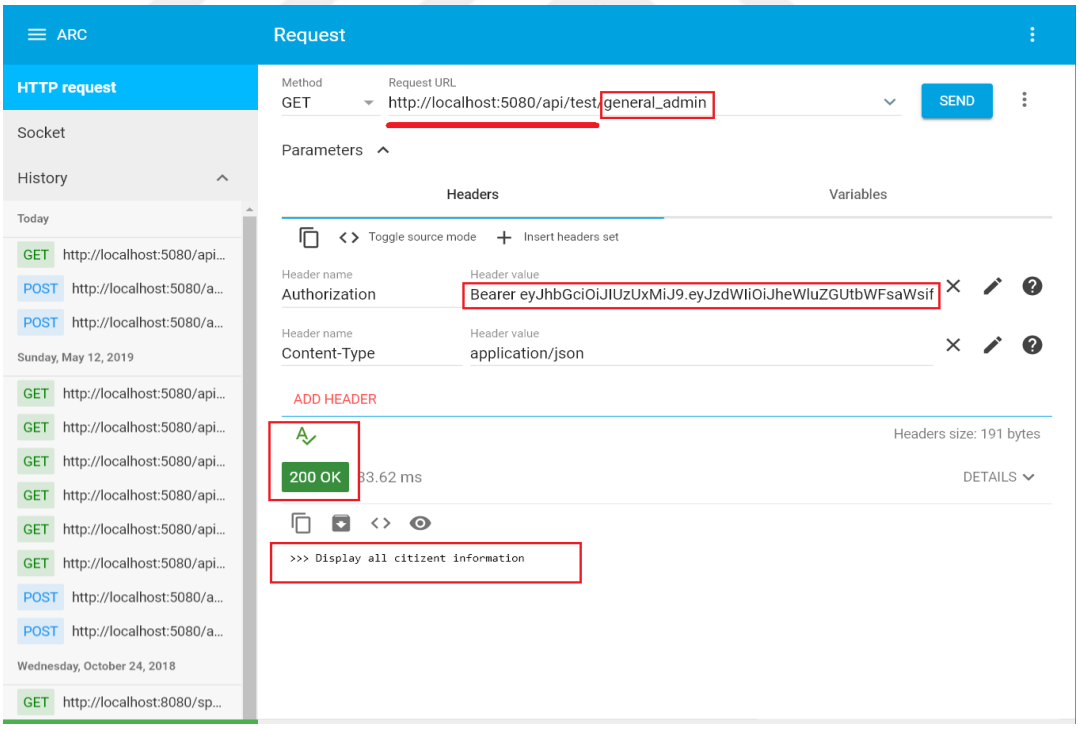


Şekil 3. 11: Yetkisiz erişim yanıtı

Öte yandan, sağlanan bilgiler doğrulanırsa, sunucu jwt belirtecini bir bearer anahtar kelimesiyle geri alır. Ardından, istemci kimlik doğrulaması için bu simgeyi sunucuya geri göndermelidir, şekil 3.13. Sunucu, simgeyi kontrol etme sorumluluğunu üstlenir ve simgenin doğru olması durumunda, kullanıcının kaynaklara erişimini sağlar.



Şekil 3. 12: JWT belirtecini sunucudan alma



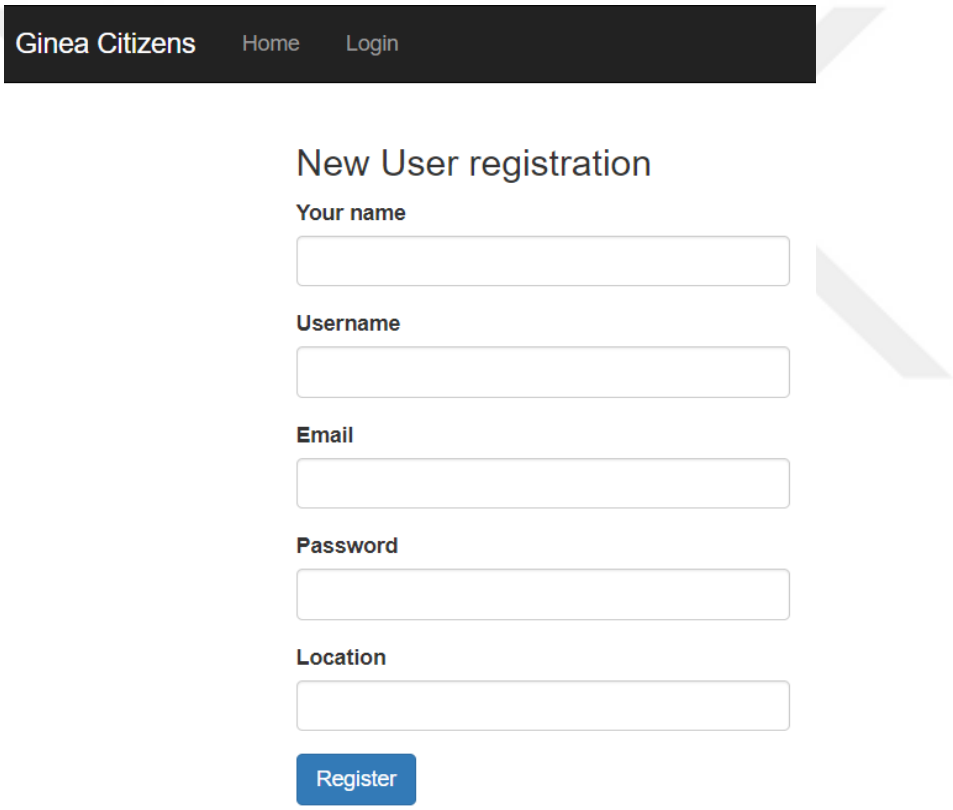
Şekil 3. 13: JWT Token doğrulama

3.5 Ön uç testleri

Yazı tipi sonu testi Angular 6 kullanılarak tasarlanmıştır. Bir kullanıcı arayüzü prototipi geliştirmek ve tasarlamak için kullanılmıştır. Ön uç, kullanıcı kaydı, kullanıcı girişi, yeni vatandaş oluşturma veya ekleme, tüm vatandaş bilgilerini arama ve görüntüleme gibi farklı senaryolardan oluşur.

3.5.1 Kullanıcı kaydı

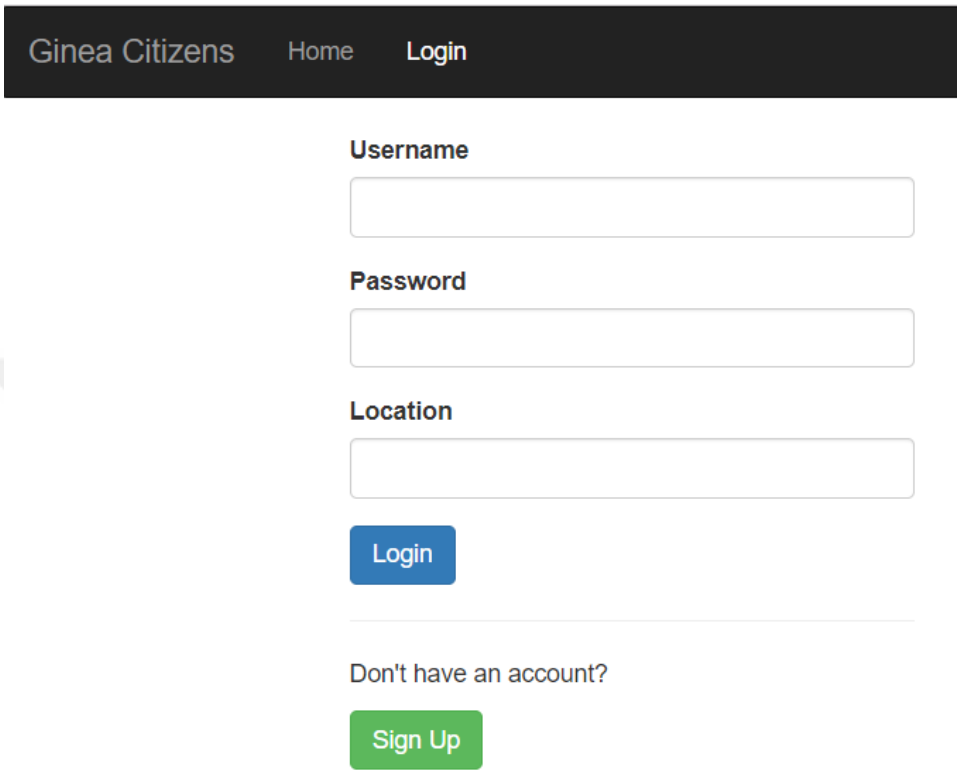
Bu senaryo arayüzü, kayıt işlemini otomatikleştirmek için kullanılır. Arka uç işleminde açıklandığı gibi, kullanıcı şekil 3.15'te gösterilen gerekli bilgileri sağlar.



Şekil 3. 14: Ön kullanıcı kayıt sayfası

3.5.2 Giriş senaryosu

Kullanıcı adı, şifre ve konum gibi gerekli bilgileri sağlayarak, kullanıcı yerel yönetici veya genel yönetici gibi rollerine göre kaynaklara erişebilir.

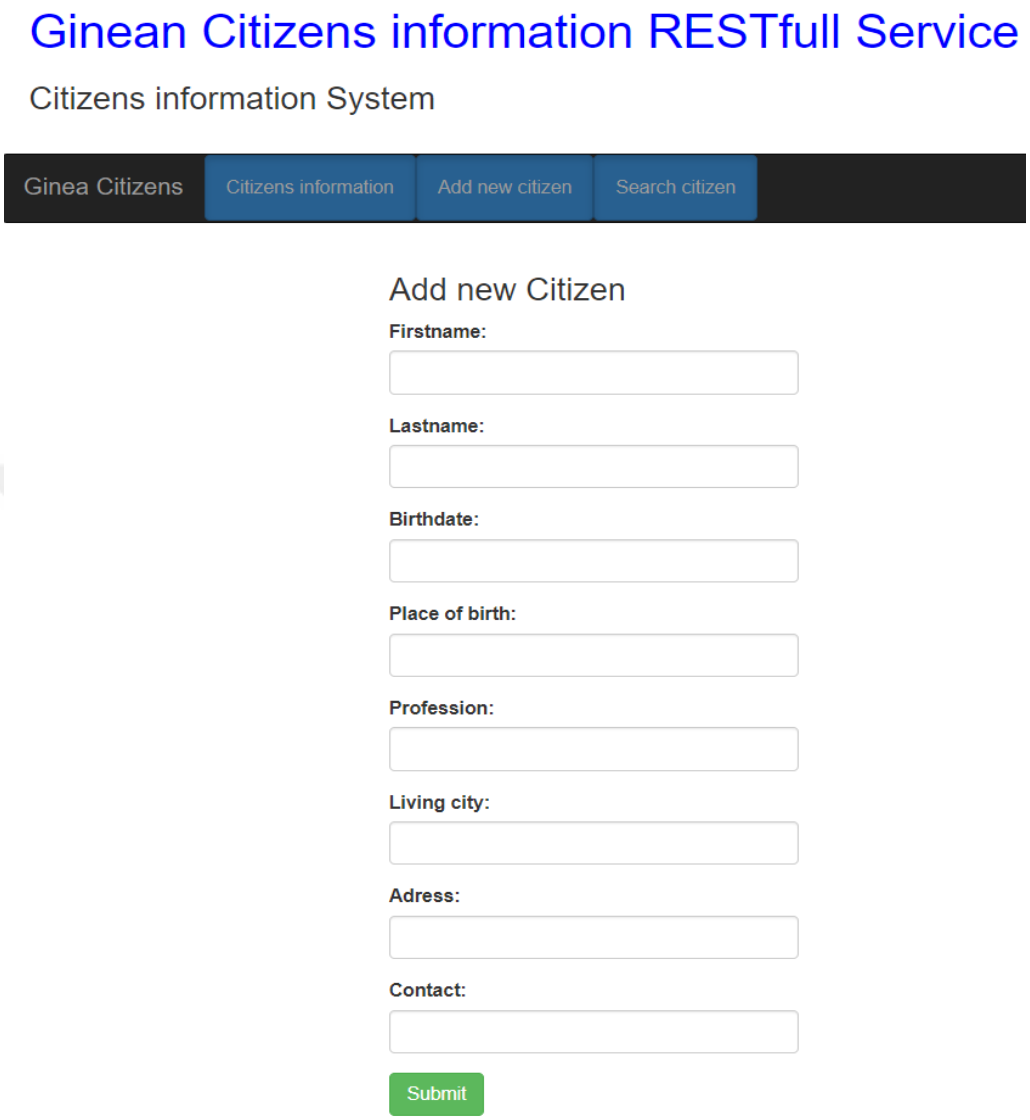


Şekil 3. 15: Ön uç giriş ölçüsü

3.5.3 Yeni vatandaş oluştur veya ekle

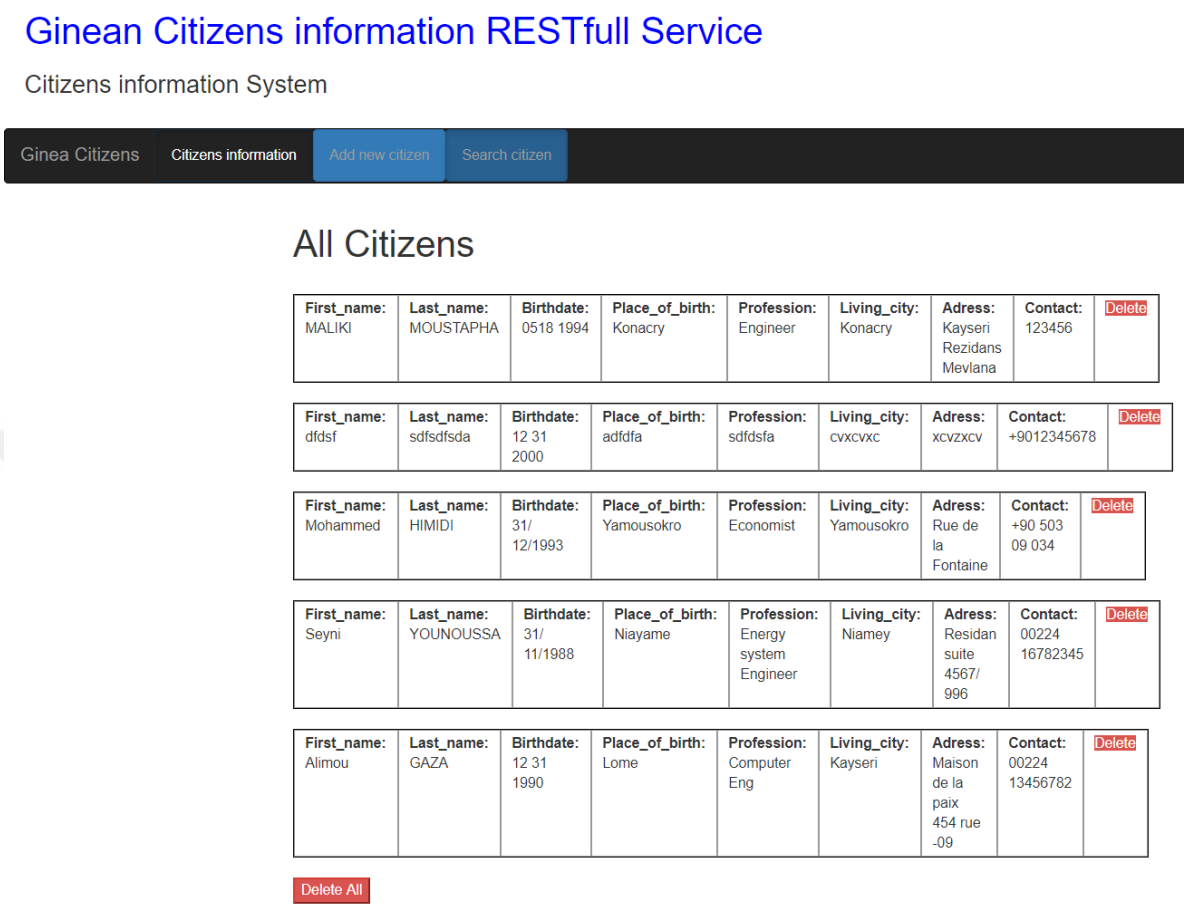
Başarılı bir giriş işleminden sonra, kullanıcı vatandaş işlemlerini farklı işlemlerle yönetme olanağına sahiptir. Bu işlemler sunucu tarafından Restfull servisleri olarak verilmektedir. Ve bu işlemler, bir gezinti çubuğu vasıtasıyla tüm kullanıcılar için uygun

hale getirilen şekiller 3.17'de görselleştirilebilir.



Şekil 3. 16: Yeni vatandaş ekleme

Bu sekmede, kullanıcı yeni vatandaş edinme kabiliyetine sahiptir, tüm vatandaş tablo 3.2'de görüntülenmektedir.



Tablo 3. 2: Tüm vatandaş bilgileri

4. BÖLÜM

TARTIŞMA-SONUÇLAR VE ÖNERİLER

4 Tartışma-Sonuçlar Ve Öneriler

4.1 Tartışma

Tezin önceki bölümlerinde, Web Servisi hakkında genel bilgiler, Web Servislerinin ortak tanımları, Web Servislerinde Güvenlik ve tüm teknolojiler hakkında farklı bölümler verilmiştir. Daha sonra web servisini daha güvenli ve daha verimli hale getirmek için geliştirilen prototip uygulamasında ayrı bir bölümde özetlenmiştir. Literatür taraması bölümünde, yüksek güvenlik seviyesine getirmek için web servis güvenliğine yönelik çeşitli yaklaşım çözümleri tartışılmaktadır.

Bu projede örnek uygulama amacıyla talep sahibi ve sağlayıcı ile iletişimi kolaylaştırmak için Tomcat sunucusu gibi pek çok uygulama kullanılmış ve birçok teknoloji Spring önyükleme, Spring Güvenliği, arka uç geliştirme için Spring JPA kullanılmıştır. Daha önce de belirtildiği gibi, ön ucun geliştirilmesi için kullanılan platform ise Angular 6'dır.

Tez çalışmasında, web servislerinde kimlik doğrulama güvenliği ile ilgili bazı problemlerin çözülmesi amaçlanmıştır çünkü web servisi yeni ve yaygın bir teknolojidir ve uygulamada güvenliği sağlamak için JWT'yi kullanmaya karar verdiğimiz her yerde kullanılır.

4.2 Sonuçlar ve Öneriler

Testlerden tanımlanabilir ve prototip uygulamasının JSON Web belirteci kullanılarak belirteç tabanlı kimlik doğrulaması, herhangi bir Web Hizmetinde uygulanabileceği sonucuna varılabilir. Belirteçle, uygulamaya kimlerin erişebileceğini bilmekteyiz. Belirteci içeriğini değiştirmeye çalışan biri varsa, web servisi belirteci geçersiz olduğunu bildiren bir hata mesajı gönderir. Bu nedenle, JSON Web Token (JWT) kullanarak yapılan kimlik doğrulamanın güvenli olduğu kabul edilir.

Gelecekte yapılması planlanan gelişmelerden bazıları aşağıdaki noktalarda önerilebilir:

- ✓ Geliştirdiğimiz prototip, gelecekte çevrimiçi ortamda uygulamak istediğimiz yerel ortamda çalışmak içindir.
- ✓ Belirteci mobil, masaüstü veya web uygulamasında tutmak için daha fazla güvenlik tenikleri bulmak gerekli olacaktır.
- ✓ JWT'nin güvenliğini artırmak için daha güvenli ve yeni teknolojiler bulmak gerekli olacaktır.

KAYNAKÇA

1. Pravat Sahoo, Nilesh Kumar Janghel, 2017 **Debabrata Samanta Securing WEB API Based on Token Authentication** International Journal on Advanced Electrical and Computer Engineering (IJAECE) ISSN(Online): 2349-9338, ISSN(Print): 2349-932X Volume -4, Issue -2,
2. Muhamad Haekal, Eliyani 2016 **Token-Based Authentication Using JSON Web Token on SIKASIR RESTful Web Service** International Conference on Informatics and Computing (ICIC) 978-1-5090-1648-8
3. Vijeyta Devi & Swapna Priya 2012 **A Perspective Approach on Citizen Card System- Applicability And Opportunity** International Journal of Computer Science and Informatics IJCSI ISSN (PRINT): 2231 –5292, Vol-1, Iss-4,
4. Adnan Masood PhD., Jim Java 2015 **Static Analysis for Web Service Security – Tools &Techniques for a Secure Development Life Cycle** IEEE 978-1-4799-1737-2
5. Chi Po Cheong, Chris Chatwin, Rupert Young 2011 **A New Secure Token For Enhancing Web Service Security** IEEE 978-1-4244-8728-8
6. Hatma Suryotrisongko, Dedy Puji Jayanto, Aris Tjahyanto 2017 **Design and Development of Backend Application for Public Complaint Systems Using Microservice Spring Boot** 4th Information Systems International Conference ISICO 1877-0509
7. Narges Shahgholi., Mehran Mohsenzadeh., Mir Ali Seyyedi., Saleh Hafez Qorani., 2011 **A New SOA Security Framework Defending Web services Against WSDL Attacks IEEE International Conference on Privacy, Security, Risk, and Trust**, and IEEE International Conference on Social Computing 978-0-7695-4578-3
8. Yang Xin, Jianjing Shen, Zhigang Si 2010 **A New Access Control Method for Semantic Web Services Based on Security Control Center** Sixth International Conference on Natural Computation (ICNC 2010) 978-1-4244-5961-2
9. Adam Agocs., Jean-Marie Le Goff 2018 **A web service based on RESTful API and JSON Schema/JSON Meta Schema to construct knowledge graphs** IEEE 978–1–5386–4599–4

10. Krisada sangsanit., Werasak Kurutach., Suronapee Phoomvuthisarn 2018 **REST Web Service Composition: A Survey of Automation and Techniques** IEEE 978-1-5386-2290-2
11. Bhumi N. Nakhuva., Prof. Tushar A. Champaneria, 2017 **Security Provisioning for RESTful web services in Internet of Things** International Conference on Advanced Computing and Communication Systems (ICACCS -2017) 978-1-5090-4559-4
12. Andy Neumann, Nuno Laranjeiro, Jorge Bernardino 2018 **An Analysis of Public REST Web Service APIs** IEEE Transactions on Services Computing.2847344
13. Nowshen Manzoor ,Sheena Mushtaq, Mrs Yasmeen 2017 **Biometric Based Citizen Information** International Journal of Advanced Research in Computer Sciences ISSNNo.0976-5697ISSN No. 0976-5697
14. Martin Garriga, Karina Rozas t, Diego Anabalon, Andres Flores and Alejandra Cechich 2016 **RESTful Mobile Architecture for Social Security Services:** A Case Study IEEE 978-1-5090-1633-4
15. V. Sankar Dr. G. Zayaraz 2016 **Securıng Confıdentıal Data In Xml Usıng Custom Level Encryptıon** International Conference on Computation of Power,E nergy Information and Communication (ICCPEIC) 978-1-5090-0901-5
16. Pham Tien Hung, Tran Minh Hoat, Nguyen Thanh, Hoang Van Dung 2013 **Solutions to Data Optimization and Security for Web Services in GNSS Applications based on Android Smartphone** IEEE 978-1-4799-3230-6
17. Iqra Ilyas , Muhammad Tayyab, Aliza Basharat 2018 **Solution to Web Services Security and Threats** International Conference on Computing, Mathematics and Engineering Technologies – iCoMET 2018 978-1-5386-1370-2
18. Ahmad Baihaqi, Obrina Candra Briliyant 2017 **Implementation of RSA 2048-bit and AES 128-bit for Secure E-Learning Web-based Application** IEEE 978-1-5386-3546-9
19. Sungchul Lee, Ju-Yeon Jo, Yoohwan Kim 2015 **A Method for Secure RESTful Web Service** IEEE ICIS 2015, 978-1-4799-8679-8
20. Andrea 2017 **ArcuriRESTful API Automated Test Case Generation** IEEE International Conference on Software Quality, Reliability and Security 978-1-5386-0592-9

21. Mikhail Zolotukhin, Tero Kokkonen, Jarmo Siltanen 2016 **Increasing Web Service Availability by Detecting Application-Layer DDoS Attacks in Encrypted Traffic** 23rd International Conference on Telecommunications (ICT) 978-1-5090-1990-8
22. Allan Delon Barbosa Araújo, Paulo Caetano da Silva **2015 PERSEC Middleware for Multiple Encryption in Web Services** 12th International Conference on Information Technology - New Generations 978-1-4799-8828-0
23. Xiuyu He and Xudong Yang, 2017 **Authentication and Authorization of End User in Microservice Architecture** Journal of Physics: Conference Series 910 012060
24. Gaoyuan Pan , Yongbin Wang 2012 **Securing RESTful WCF Services With Xauth and Service Authorization Manager** Fifth International Joint Conference on Computational Sciences and Optimization 978-0-7695-4690-2
25. Nowsheen Manzoor, Sheena Mushtaq 2015 **BIOMETRIC BASED CITIZEN INFORMATION SYSTEM** International Journal of Advanced Research in Computer Science , 0976-5697
26. Mohsen Rouached, Hassen Sallay 2013 **RESTful Web Services for High Speed Intrusion Detection Systems** IEEE 20th International Conference on Web Services 978-0-7695-5025-1
27. Tristan Lavarack and Marijke Coetzee 2010 **Considering web services security policy compatibility** IEEE 978-1-4244-5495-2
28. Gurpreet Singh, Supriya 2013 **A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security** ,International Journal of Computer Applications (0975 – 8887)Volume 67– No.19,
29. Kalyani Ganesh Kadam, Prof. Vaishali Khairnar 2015 **An Implementatıon Of Hybrıd Encryptıon-Decryptıon (Rsa Wıth Aes And Sha256) For Use In Data Exchange Between Clıent Applıcatıons And Web Servıces** International Journal of Technical Research and Applications e-ISSN: 2320-8163 PP. 51-56
30. Xiuyu He and Xudong Yang 2017 **Authentication and Authorization of End User in Microservice Architecture** Journal of Physics J. Phys.: Conf. Ser. 910 012060
31. Tunç D. Medeni et al 2011 **Informatıon Socıety Strategy & E-Government Gateway Development In Turkey: Movıng Towards Integrated Processes**

And Personalızed Servıces tGov Workshop ‟11 Brunel University, West London, UB8 3PH

32. Aggeliki Tsohou et al 2013 **Proposing a reference process model for the citizen-centric evaluation of e-government services** Emerald Group Publishing Limited 1750-6166

ÖZGEÇMİŞ

Adı Soyadı: Saikou Ibrahima DIALLO

Uyruğu: Gine

Doğum Tarihi ve Yeri: 30.05.1989 – Dalaba

Medeni Durum: Bekar

e-mail: sidmitty@gmail.com

Yazışma Adresi: Kayseri rezidans, Mevlâna Mah. Şehit Ali Örnek Cad. Ermiş sok. No 12 TALAS - KAYSERİ

EĞİTİM

Derece	Kurum	Mezuniyet Tarihi
Yüksek Lisans	Erciyes Üniversitesi, Bilgisayar Mühendisliği Bölümü (Türkiye)	2019
Lisans	University Nongo Conakry(UNC), Bilgisayar Bilimler Bölümü (Gine)	Temmuz 2014
Lise	Hadja Habibata Tounkara (Gine)	Temmuz 2010

İŞ DENEYİMLERİ

Yıl	Kurum	Görev
2018	Everest Çelik Kapı	Stajyer
2013-2015	HeroTic.Sarl	Ağ Güvenliği Yonetecisi

YABANCI DİL

Fulaça (Ana dili)

Fransızca

İngilizce

Türkçe