|  |
| --- |
| ***Teme:***  *http protokol, metode http protokola*  *realizacija http strežnikov*  *REST Api in OAS3*  *Generiranje in raba certifikatov /https/* |

|  |
| --- |
| Delna specifikacija naloge (preliminarni test za izvedbo naloge in seznanjevanje z ekositemom/okoljem) je podano na : <https://gitlab.vegova.si/SerhioN/comptest05/tree/master/java>  Specifikacija API-ja: https://gitlab.vegova.si/SerhioN/comptest05/tree/master/openapi  Relevantni mapi sta torej:   * java (dva strežnika) * openapi (oas3 definicija)   REST – API (poenostavljen) definira:   * Strežnik, s katerim komuniciramo * EndPoint-e (URI-je, ki jih lahko naslovimo) * Parametre pri komunikaciji * Podatkovne modele podatkov (ki jih vrača, jih dajemo kot parametre) * Kdo/kdaj/do česa lahko dostopa (security) • ……   Postopki v zvezi s certifikati so spisani tudi v : <https://gitlab.vegova.si/SerhioN/comptest05/-/tree/master/certificate_infos>  Naloga sovpada z nalogo dijakov predmeta NSA, le da je njihova realizacija strežnika drugačna / apache/nginx/php/node, je pa v opisu njihove naloge dana specifikacija za rešitev nalog 5 in 6 te vaje. |

**Naloga 1**

Preskusi oba strežnika (http,https) op: oba bi morala delovati

a) http: ali se da nasloviti vse specificirane endpoint-e (path definirani v oas3)

b) enako kot http pod a), vendar je predhodno potrebno definirati certifikat. Ta certifikat generirajte z javanskimi orodji (glej datoteko postopek\_certifikat.txt). Pri njegovi generaciji uporabite pravo (vaše) ime in priimek, Organization Unit=rso2021, O=VegovaLjubljana, city=ljubljana, coutryCode=SI

**Naloga 2**

Oba strežnika naj se odzivata na vse definirane endPoint-e. Odziv naj bo demonstrativen (da se vidi, da URL-ji delujejo; glej primer http strežnika). Definirajte in usposobite manjkajoče endpointe na obeh strežnikih.

V pomoč: editor.swagger.io - lahko uvozite openapi.json/yaml -> v UI dobite izrisane vse endpoint-e in podatkovne modele

**Naloga 3**

Ugotovite, ali se da v javanski keystore uvoziti openssl generiran certifikat ali kako ? Podatki za generiranje tega so kot pri zgornjem, le za organizacijsko enoto uporabite OU=rac202xOpenSSL.

Postopek generiranja potrdila oz. zahtevka za potrdilo, njegovo podpisovanje in pridobivanje je spisan v dokumentu <https://gitlab.vegova.si/SerhioN/comptest05/tree/master> NSAvaja32new.doc v nalogi 2, alinejah a in b. V datoteko .cfg boste dodali v razdleke [alt\_names] poleg obstoječih še email kot :

DNS.1 = janez.kranjski.si

DNS.2 = 192.168.106.222

#DNS.3 = streznikX.org.si

IP.1 = 192.168.106.222

#IP.2 = 2001:dead:beef:ZZZ::666

RFC822Name=janez.kranjski@students.vegova.si

Polji DNS in IP omogočata identifikacijo stroja, polje RFC822Names pa omogoča identifikcijo uporabnikovega emaila in s tem omogoči identifikacijo in avtentifikacijo uporabnika.

V poročilo boste oddali rezultat ukaza keytool -v -list -keystore .keystore (uvožena potrdila). Ter informacije o certifikatu (zaslonske slike certifikatov kot je specificirano v dokumentu NSA.. v alineji Kaj oddate)

**Naloga 4\***

V predhodnih dveh vajah smo a) ustvarili poročilo na osnovi vsebine neke tabele podatkovne zbirke v PDF oblike in b)izdelali mehanizem za elektronsko podpisovanje tega PDF-a. Ideja naloge je preprosta:

Realizacija https strežnika REST strežnika, ki bi na enem izmed end-pointov generiral zahtevan PDF, na drugem pa bi ga npr. podpisal (da poenostavimo, slika vizuaanega podpisa bi bila fiksna na strežniku, podpis pa bi bil izveden na osnovi certifikata strežnika.

**Naloga 5\* (glej dokumente NSA)**

V okviru https strežnika implementirajte API specificiran openapi/openapi.json ali openapi/apenapi.yaml datoteki. Pri tem morate implementirati tabelo podatkovne baze, ki jo omenjena datoteka specificira v razdelku Schemas, kjer imate opisan model za Strel. Endpoint-i API ja specificrani razdelka path (gen, get, get/st\_serije; pri tem je serija opredeljena kot tisti streli, ki imajo isti timestamp, gen pa vedno generira naključnih 10 strelov z istim timestampom; get vedno vrne vse strele v tabeli; get/st\_serije pa zgolj strele izbrane serije).

**Naloga 6\* (glej dokumente NSA)**

Realizirajte celoten CRUD za tabelo Strel! (dodajanje posameznega strela, brisanje, ažuriranje, vračanje vrednosti). Pri tem si definirajte svoj API.

\* so v vsakem primeru neobvezne