

Korištenje baza podataka pomoću rest servisa u oblaku

Škrtić, Ivana

Master's thesis / Diplomski rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Economics and Business / Sveučilište u Zagrebu, Ekonomski fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:148:383689>

Rights / Prava: [Attribution-NonCommercial-ShareAlike 4.0 International](#)/[Imenovanje-Nekomercijalno-Dijeli pod istim uvjetima 4.0 međunarodna](#)

Download date / Datum preuzimanja: **2024-12-27**



Repository / Repozitorij:

[REPEFZG - Digital Repository - Faculty of Economics & Business Zagreb](#)



Sveučilište u Zagrebu
Ekonomski fakultet
Integrirani preddiplomski i diplomski sveučilišni studij
Poslovna ekonomija-Menadžerska informatika

**KORIŠTENJE BAZA PODATAKA POMOĆU REST SERVISIA
U OBLAKU**
**USE OF DATA BASES WITH REST SERVICES IN THE
CLOUD**

Diplomski rad

Student: Ivana Škrtić
JMBAG: 0067537375
Mentor: Prof. dr. sc. Katarina Ćurko

Zagreb, rujan 2020

IVANA ŠKRTIĆ
Ime i prezime studenta/ice

IZJAVA O AKADEMSKOJ ČESTITOSTI

Izjavljujem i svojim potpisom potvrđujem da je DIPLOMSKI RAD
(vrsta rada)
isključivo rezultat mog vlastitog rada koji se temelji na mojim istraživanjima i oslanja se na objavljenu literaturu, a što pokazuju korištene bilješke i bibliografija. Izjavljujem da nijedan dio rada nije napisan na nedozvoljen način, odnosno da je prepisan iz necitiranog rada, te da nijedan dio rada ne krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za bilo koji drugi rad u bilo kojoj drugoj visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.

Student/ica:

U Zagrebu, _____

(potpis)

SAŽETAK

Ovaj rad sastoji se od teorijskog i praktičnog dijela. U teorijskom dijelu opisani su RESTful i Big web servisi. Detaljno je objašnjena REST arhitektura i ograničenja, HTTP protokol i JSON format reprezentacije podataka. Opisano je računarstvo u oblaku, Oracle Cloud usluge i njihove specifičnosti.

U praktičnom dijelu prikazana je primjena RESTful servisa na B2B aplikaciji prodaje. Objašnjena je tro-slojna arhitektura aplikacije, opisani su alati za razvoj i testiranje. Razvoj aplikacije je prikazan kroz faze: konceptualnog modeliranja, logičkog modeliranja i fizičkog modeliranja u relacijskoj bazi podataka. Aplikacija je implementirana u Oracle Cloud-u koristeći Oracle Cloud Free Tier (Always Free Services). Objašnjeno je postavljanje REST servisa na objektima baze podataka tablicama i pogledima. Prikazani su slučajevi korištenja: registracija kupaca, pregled kategorije proizvoda, grupa proizvoda, proizvoda, zadavanje narudžbi, praćenje narudžbi.

Ključne riječi: web servis, REST, HTTP, JSON, računalni oblak, Oracle

SUMMARY

This paper consists of a theoretical and a practical part. The theoretical part describes RESTful web services and Big web services. The REST architecture and constraints, the HTTP protocol, and the JSON format of data representation are explained in detail. Cloud computing, Oracle Cloud services and their specifics are described.

The practical part shows the application of the RESTful service on the B2B sales application. The three-layer architecture of the application is explained, the tools for development and testing are described. Application development is presented through the phases of conceptual modeling, logical modeling, physical modeling in a relational database. The application is implemented in Oracle Cloud using Oracle Cloud Free Tier (Always Free Services). The setting up of REST services on database objects with tables and views is explained. The use cases are shown: customer registration, product category overview, product group, products, placing orders, order tracking.

Key words: web service, REST, HTTP, JSON, cloud computing, Oracle

Sadržaj

1. UVOD	1
1.1. Cilj rada	1
1.2. Metodologija istraživanja	1
1.3. Sadržaj i struktura rada	1
2. BAZE PODATAKA	3
2.1. Općenito o bazama podataka	3
2.2. Oracle baza podataka	4
3. WEB SERVISI	5
3.1. Općenito o web servisima	5
3.2. Big web servisi	5
3.3. Razvoj RESTful web servisa	6
4. RAČUNARSTVO U OBLAKU	8
4.1. Općenito o računarstvu u oblaku	8
4.2. Oracle Cloud	11
5. RESTFUL WEB SERVISI	15
5.1. Elementi REST arhitekture	15
5.1.1 Elementi podataka	15
5.1.2 Konektori	16
5.1.3. Komponente	17
5.2. Ograničenja REST arhitekture	17
5.3. HTTP protokol	20
5.3.1. HTTP zahtjev	20
5.3.2. HTTP odgovor	22
5.3.3. Statusni kodovi	23
5.4. Sigurnost	24
5.5. JSON format	26
5.6. Metode	27
5.7. Primjena	28
5.8. Prednosti i nedostaci	29
6. ARHITEKTURA REST SERVISA	31
6.1. Općenito	31
6.2. Komponente REST servisa	33
6.3. Razvojni alati	35

7. PRIMJER REST SERVISA - B2B APLIKACIJA PRODAJE.....	36
7.1. Modeliranje podataka	36
7.2. Implementacija modela u relacijsku bazu podataka	38
7.3. Oracle Rest Data Services	44
7.4. Slučajevi korištenja.....	51
7.4.1. Registracija kupaca	51
7.4.2. Pregled proizvoda	54
7.4.3. Zadavanje narudžbi	55
7.4.4. Praćenje narudžbe	59
8. ZAKLJUČAK	61
LITERATURA.....	63
POPIS TABLICA.....	67
POPIS SLIKA	67
ŽIVOTOPIS	69

1. UVOD

Internet je omogućio jednostavno povezivanje računala i računalnih sustava na fizičkom nivou. Veliki problem i izazov današnjih informacijskih sustava u poslovnom svijetu je komunikacija među različitim tehnologijama odnosno interoperabilnost između različitih platformi. Jedan od problema je korištenje baza podataka iz različitih aplikacija i platformi. Izlaganje baza podataka odnosno korištenje baza podataka pomoću web servisa sve se više koristi. Web servisi jedan su od novih trendova projektiranja informacijskih sustava. REST (engl. Representational State Transfer) stil arhitekture danas je jedan od najvažnijih i najviše korištenih stilova arhitekture za kreiranje web servisa.

1.1. Cilj rada

Istraživanjem se želi potvrditi da se korištenjem REST servisa na bazi podataka može brzo i efikasno razviti REST sučelje koje će omogućiti pristup i ažuriranje podataka iz različitih platformi i programskih jezika. Želi se dokazati tehnološki napredak u razvoju REST servisa, tako da više nije potrebno ulagati u infrastrukturu, angažirati programere i ostale IT stručnjake sa specifičnim znanjima i vještinama. Razvoj REST sučelja sada mogu raditi informatičari koji imaju znanja o bazama podataka i SQL-u. Istraživanjem se također želi potvrditi zrelost tehnologije računarstva u oblaku u segmentima baze podataka, web servisa u srednjem sloju i razvojnih alata. Ostvarenje ciljeva bit će prikazano na primjeru razvoja sučelja B2B aplikacije prodaje koja će biti kompletno razvijena i implementirana u Oracle Cloud-u.

1.2. Metodologija istraživanja

Metodologija istraživanja je razvoj softvera (eksperimentiranje/prototipiranje), kvalitativna metodologija. Teorijski dio rada temelji se na proučavanju stručne i znanstvene literature, podataka iz raznih izvora. U radu se koriste sljedeće metode: deskripcija, analiza, sinteza, dedukcija/indukcija, klasifikacija.

1.3. Sadržaj i struktura rada

Rad je podijeljen u dvije cjeline. Prva cjelina je općenito o web servisima i računarstvu u oblaku. Opisani su REST i Big (SOAP) web servisi. Detaljnije je objašnjena REST arhitektura, HTTP protokol, JSON format za prezentaciju podataka. Objašnjene su osnovne metode REST servisa: GET, POST, PUT, DELETE. Opisano je računarstvo u oblaku, Oracle Cloud usluge i njegove specifičnosti.

U drugoj cjelini prikazana je primjena REST servisa na B2B aplikaciji prodaje. Objasnjena je tro-slojna arhitektura i alati za razvoj. Prikazan je razvoj aplikacije kroz faze konceptualnog modeliranja, logičkog modeliranja, fizičkog modeliranja u relacijskoj bazi podataka. Objasnjeno je postavljanje REST servisa, odnosno stvaranje resursa na objektima baze podataka, tablicama i pogledima. Prikazani su slučajevi korištenja: registracija kupaca, pregled kategorije proizvoda, grupa proizvoda, proizvoda, zadavanje narudžbi, praćenje narudžbi.

2. BAZE PODATAKA

2.1. Općenito o bazama podataka

Baza podataka sadržajem svojih podataka predstavlja model podataka poslovnog sustava odnosno segmenata stvarnog svijeta. Baza podataka je organizirana kolekcija međusobno povezanih podataka, pohranjenu na sredstvima informacijsko-komunikacijske tehnologije (poslužitelju, mreži, oblaku).¹

Struktura podataka u bazama podataka može biti raznolika, razlikujemo: baze strukturiranih podataka, baze nestrukturiranih ili polustrukturiranih podataka (baze dokumenata), baze znanja.

Baze nestrukturiranih podataka sadrže nestrukturirane ili polustrukturirane jednomedijske (tekst, audio, video) ili multimedijske podatke. Objekti čije podatke pohranjuju su novinski, stručni, znanstveni članci odnosno dokumenti, bibliografske jedinice, slike, videozapisi, zvukovni zapisi.

Baze znanja sadrže znanje prikazano u različitim oblicima: pravilima, semantičkim mrežama, okvirima, scenarijima. Sustavi temeljeni na znanju (npr. ekspertni sustavi) mogu rješavati različite probleme: dijagnoza uzroka grešaka u složenim sustavima, financijska predviđanja, planiranje akcija.

Baze strukturiranih podataka imaju široku primjenu. Najčešće se koriste se u poslovnim sustavima.

Po građi razlikuju se starije hijerarhijske i mrežne te novije relacijske, objektne i dimenzijske baze podataka. Danas se najviše koriste relacijske baze podataka.

Sustav za upravljanje bazom podataka (SUBP) (engl. DBMS Database Management System) je programski sustav koji omogućuje rad s bazom podataka. SUBP ima sljedeće funkcije:²

- Definiranje baze podataka – koristi se zaseban jezik koji opisuje shemu baze podataka (engl. DDL Data Definition Language)
- Manipulaciju podacima u bazi podataka – ostvaruje se naredbama za manipulaciju podacima (engl. DML Data Manipulation Language)
- Upravljačke funkcije – funkcije sigurnosti baze podataka, zaštita od neovlaštenog korištenja

¹ Varga, M. (1994) Baze podataka Konceptualno, logičko i fizičko modeliranje podataka. Zagreb:DRIP. str. 10

² Ibid. str. 11

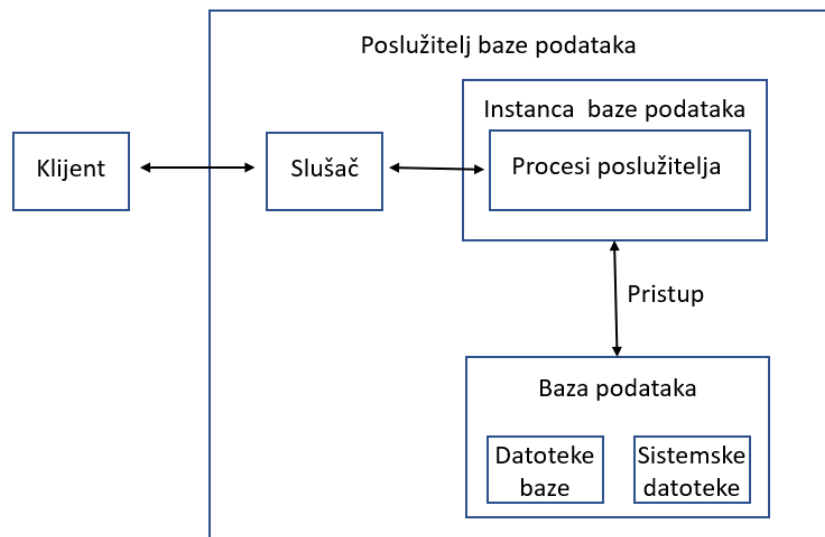
očuvanja integriteta baze podataka, zaštita od mogućih oštećenja i oporavak, funkcije statističkog praćenja rada baze podataka

2.2. Oracle baza podataka

Oracle baza podataka je sustav upravljanja bazama podataka koji proizvodi i prodaje tvrtka Oracle Corporation. Tvrtka Oracle izdala je prvu komercijalnu verziju baze podataka 1979. godine. Oracle baza podataka najnaprednija je baza podataka koja se danas može naći na tržištu.

Poslužitelj Oracle baze podataka sastoji se od baze podataka i barem jedne instance. Instanca baze podataka je sučelje između klijentskih aplikacija (korisnika) i baze podataka. Instanca je kombinacija memorije i procesa poslužitelja. Baza podataka skup je datoteka u koje se pohranjuju podaci kao i sistemskih datoteka koje služe za rad SUBP-a. Klijent i instanca baze podataka komuniciraju pomoću slušača (engl. Listener). Klijenti su aplikacije kojima korisnici baze podataka zadaju SQL naredbe (DDL i DML) poslužitelju baze podataka. Klijenti mogu biti korisničke aplikacije ili alati za razvoj aplikacija i upravljanje bazom podataka. U ovom radu klijent za razvoj aplikacija korišten je klijent Oracle SQL Developer.

Slika 1: Arhitektura poslužitelja baze podataka Oracle



Izvor: Izrada autora prema Oracle Tutorial (2020) Oracle Database Architecture [online].

Dostupno na: <https://www.oracletutorial.com/oracle-administration/oracle-database-architecture/> [26.kolovoza.2020]

3. WEB SERVISI

3.1. Općenito o web servisima

Web servis je općeniti termin za interoperabilnost softvera između računala koja imaju mrežnu adresu i nalazi se na mrežnoj lokaciji.³ Web servis ima sučelje koje skriva detalje implementacije kako bi se moglo koristiti neovisno o hardverskoj ili softverskoj platformi na kojoj je implementiran i neovisno o programskom jeziku na kojem je napisan. Neovisnost omogućuje aplikacijama da se jednostavno povezuju, omogućuju implementacije na različitim tehnologijama. Web servisi mogu se koristiti samostalno ili s drugim web servisima za obavljanje složenih poslovnih transakcija.

Web servisi su aplikacije za klijente i poslužitelje koje komuniciraju uglavnom putem protokola HTTP (engl. HyperText Transfer protocol) na WWW (engl. World Wide Web).⁴ Kao što je opisao konzorcij World Wide Web Consortium (W3C), web servisi pružaju standardni način interakcije između softverskih aplikacija koje se izvode na različitim platformama. Web servise karakteriziraju velika interoperabilnost i proširivost. Web servisi se mogu kombinirati i jednostavno povezati da bi se mogle obavljati složene operacije. Programi koji pružaju jednostavne usluge mogu međusobno komunicirati da bi pružili sofisticirane usluge.

3.2. Big web servisi

Big web servisi su softverske komponente koje se mogu pozivati putem Interneta koristeći XML poruke koje slijede SOAP standard. SOAP (engl. Simple Object Access Protocol) je način na koji program koji se izvodi na određenom operativnom sustavu može komunicirati s drugim programom koji se može izvoditi na istom ili različitom operativnom sustavu, pritom koristeći standardni HTTP protokol i XML način zapisivanja informacija.⁵ Ovi web servisi su najpopularnija realizacija SOA servisno orijentirane arhitekture (engl. Service Oriented Architecture). SOA je stil arhitekture koji podržava integraciju poslovnih procesa kao

³ IBM (2020) IBM Knowledge Center What is a web service [online]. Dostupno na: https://www.ibm.com/support/knowledgecenter/SSGMCP_5.2.0/com.ibm.cics.ts.webservices.doc/concepts/dfhws_definition.html [20.svibnja.2020]

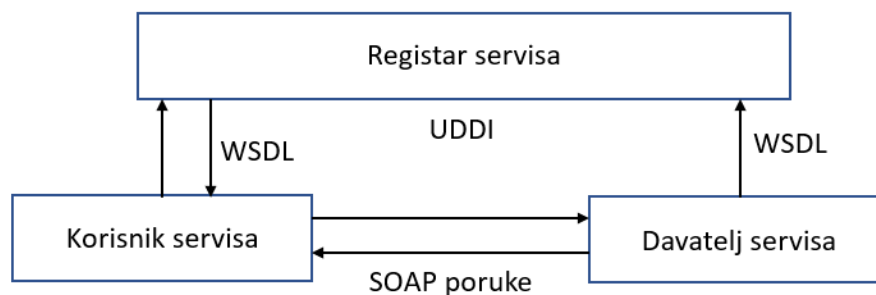
⁴ Oracle (2013) The Java EE 6 Tutorial, What Are Web Services [online]. Dostupno na: <https://docs.oracle.com/javase/6/tutorial/doc/gijvh.html> [20.svibnja.2020]

⁵ Trstenjak, B. (2010) SOAP komunikacijski protokol [online]. Dostupno na: <https://hrcak.srce.hr/55559> [5.svibnja.2020]

povezanih servisa kojima se može pristupiti preko mreže. SOA predstavlja programski model za izradu Web servisa.⁶

Kod Big web servisa postoje tri uloge odnosno tri tipa sudionika: davatelj servisa, korisnik servisa i registar servisa.⁷ Davatelj stvara web servis i njegovu definiciju, a zatim ga objavljuje u registar servisa. Registar servisa je temeljen na standardu koji se zove UDDI (engl. Universal Description, Discovery, and Integration). Jednom kada je web servis objavljen, korisnici ga mogu pronaći usluge preko UDDI sučelja. WSDL (engl. Web Services Definition Language) je specifikacija koja opisuje web servise klijentima. WSDL je XML dokument koji ima specifikacije za programsko sučelje korisnika. Nakon što implementiraju programsko sučelje prema WSDL specifikaciji, korisnici mogu koristiti web servis.

Slika 2. Komunikacija sa web servisom



Izvor: Izrada autora prema Centar Informacijske Sigurnosti (2012) Ispitivanje sigurnosti web servisa [online]. Dostupno na: <https://www.cis.hr/files/dokumenti/CIS-DOC-2012-02-040.pdf> [21.ožujka.2020]

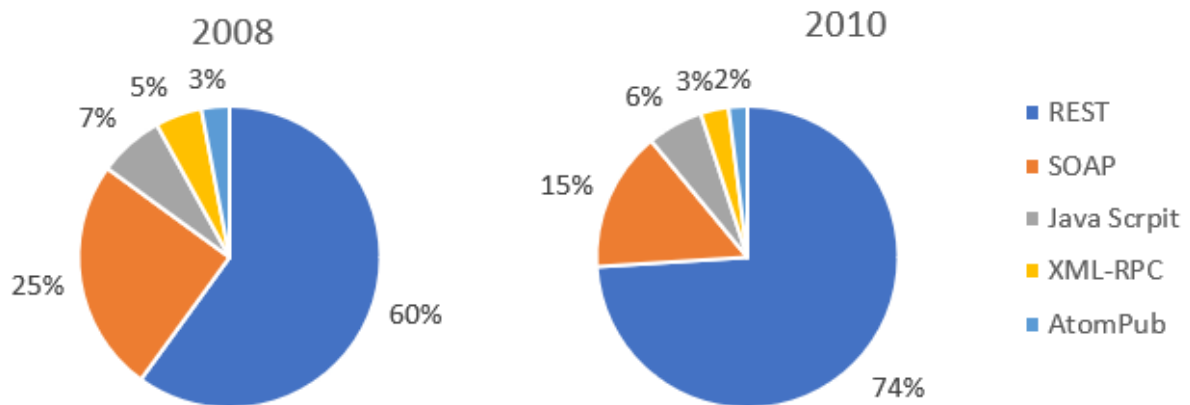
3.3. Razvoj RESTful web servisa

RESTful web servisi pojavili su se nakon Big web servisa. RESTful web servisi su jednostavniji za izradu i korištenje, dobro su integrirani sa HTTP-om, ne zahtjevaju XML u porukama te trebaju manje resursa za obradu. Širenjem mobilnih aplikacija, a zbog ograničenja resursa na mobilnim platformama RESTful web servisi postali su nezamjenjivi u mobilnim aplikacijama. Pružatelji usluga u oblaku također ih masovno koriste. Većina projekata koji se danas razvijaju bazirani su na RESTful web servisima.

⁶ Mehta, M., Lee S. (2006) Service-Oriented Architecture: Concepts and Implementation [online]. Dostupno na: https://www.researchgate.net/publication/237776180_Service-Oriented_Architecture_Concepts_and_Implementation [3.svibnja.2020]

⁷ Pennington, C. et al. (2006) Introduction to Web Services [online]. Dostupno na: https://www.researchgate.net/profile/Jorge_Cardoso4/publication/236860265_Introduction_to_Web_Services/links/02e7e5273ff5a8c4f9000000/Introduction-to-Web-Services.pdf [20.veljače.2020]

Slika 3. Prelazak servisa na REST protokol



Izvor: Izrada autora prema Lozić, D., Šimec A. (2014) Pametna komunikacija na Internetu preko REST protokola [online]. Dostupno na: https://bib.irb.hr/datoteka/713376.2014_MIPRO_Pametna_komunikacija_na_Internetu_preko_REST_protokola.pdf [10.travnja.2020]

4. RAČUNARSTVO U OBLAKU

4.1. Općenito o računarstvu u oblaku

Računarstvo u oblaku je mogućnost krajnjih korisnika da sukladno svojim potrebama koriste masovne računalne resurse i da do njih mogu doći brzo i jednostavno.⁸

Računalni oblak (engl. Cloud computing) gledan sa stajališta krajnjeg korisnika je model korištenja računalnih resursa (servera, diskova, operacijskog sustava i aplikacija) na način da se ti resursi unajmljuju a ne kupuju. Posljedica takvoga pristupa jest da krajnji korisnik plaća samo onoliko računalnih resursa koliko doista i koristi (engl. Pay as you go) te da više ne mora voditi brigu oko nabave hardvera te instalacije i održavanja softvera (OS-a i aplikacija) na tom hardveru. Karakteristike računarstva u oblaku:⁹

1. Agilnost
 - Oblak djeluje u distribuiranom računalnom okruženju
 - Dijeli resurse među korisnicima i radi vrlo brzo
2. Visoka dostupnost i pouzdanost
 - Dostupnost poslužitelja je visoka i pouzdana jer su šanse za pogreške infrastrukture minimalne
3. Visoka skalabilnost
 - Masovni resursi koji se nude u oblaku su vrlo skalabilni tj. mogu se zakupljivati prema potrebama i otporni su na vršna opterećenja
4. Višestruko dijeljenje
 - Uz pomoć računarstva u oblaku, više korisnika i aplikacija može učinkovitije raditi sa smanjenjem troškova dijeljenjem zajedničke infrastrukture
5. Neovisnost o uređaju i lokaciji
 - Računarstvo u oblaku omogućuje korisnicima pristup sustavima putem web klijenata bez obzira na njihovu lokaciju ili uređaj koji koriste, npr. računalo, mobilni telefon
 - Budući da je infrastruktura izvan web mjesta (obično je dostupna od treće strane) i pristupa se putem Interneta, korisnici se mogu povezati s bilo kojeg mjesta

⁸ Spremić, M. (2017). Digitalna transformacija poslovanja. Zagreb:Ekonomski fakultet Zagreb. str.24

⁹ Sanneboina, A. (2020) Cloud computing [online]. Dostupno na:

https://www.researchgate.net/publication/340332092_cloud_computing [10.svibnja.2020]

6. Održavanje

- Održavanje aplikacija kod računarstva u oblaku je jednostavnije jer ih nije potrebno instalirati i održavati na računalu svakog korisnika

7. Niski troškovi

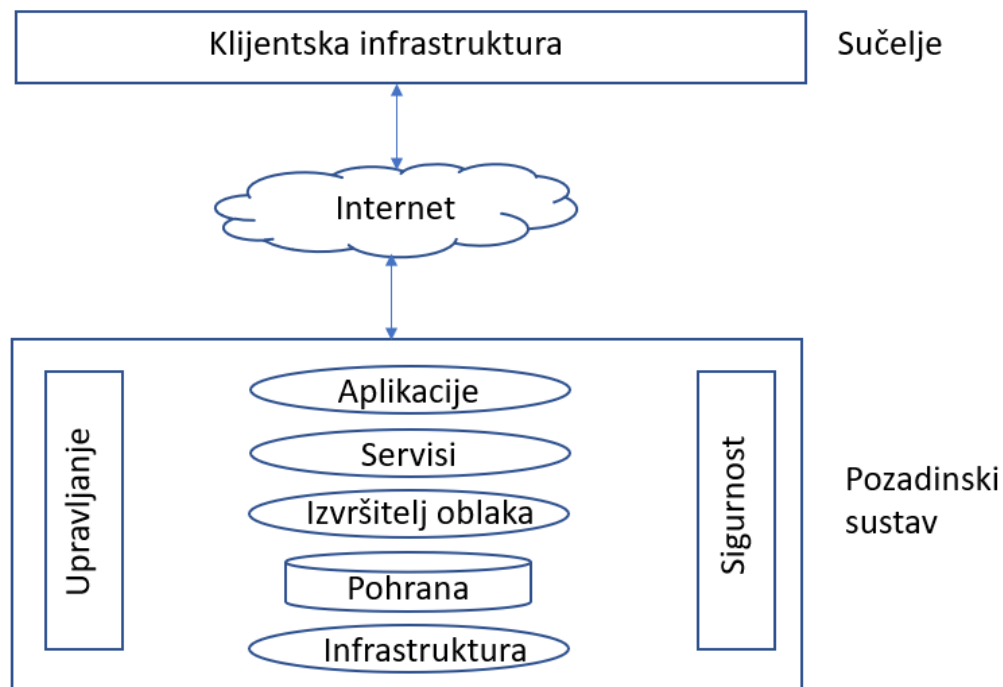
- Korištenjem računarstva u oblaku smanjit će se troškovi jer nisu potrebne investicije u vlastitu infrastrukturu

8. Usluge plaćanja po upotrebi

- Korištenje usluga računarstva u oblaku plaća se prema upotrebi

Arhitektura računarstva u oblaku podijeljena je u dva dijela: sučelje (engl. Front end), pozadinski sustav (engl. Back end).

Slika 4. Arhitektura računarstva u oblaku



Izvor: Izrada autora prema Sanneboina A. (2020) Cloud computing [online]. Dostupno na: https://www.researchgate.net/publication/340332092_cloud_computing [10.svibnja.2020]

U sučelju se nalaze klijenti odnosno korisnici oblaka. Sučelje sadrži aplikacije na strani klijenta koje su potrebne za pristup računalnim platformama u oblaku. To su uglavnom web preglednici (Chrome, Firefox, Internet Explorer, itd.) za računala, tablete i mobilne uređaje. U pozadinskom sustavu su davatelji usluga. Oni upravljaju svim resursima koji su potrebni za pružanje usluga

računarstva u oblaku. To uključuje velike količine prostora za pohranu podataka, sigurnosne mehanizme, virtualne strojeve, poslužitelje, mehanizme za kontrolu prometa.

Osnovni modeli računalnih oblaka su :¹⁰

1. Javni oblak

- Sličan bilo kojoj javnoj usluzi npr. usluge električne energije
- Davatelj usluge naplaćuje uslugu onoliko koliko je korisnik to zaista koristio

2. Privatni oblak (interni oblak)

- Poduzeće svoje podatke, aplikacije i poslovne procese odlučilo je premjestiti u oblak

3. Hibridni oblak

- Kombinacija javnog i privatnog oblaka
- Na privatnom oblaku se odvijaju poslovni procesi , a javni oblak osigurava zaštitu i obnovu podataka

4. Zajednički oblak

- Samostalno korištenje oblaka je skuplje pa korisnici odluče koristiti zajednički oblak, tako će osigurati veću razinu sigurnosti i privatnosti nego u javnom oblaku

Računarstvo u oblaku se u praksi primjenjuje na tri različita načina:¹¹

1. Infrastruktura kao usluga (engl. IaaS, Infrastructure-as-a-Service)

- IaaS je poznata i kao Hardver kao usluga (engl. HaaS, Hardware-as-a-Service).
- To je računalna infrastruktura kojom se upravlja putem interneta.
- Glavna prednost korištenja je u tome što on pomaže korisnicima da izbjegnju troškove i složenost kupnje i upravljanja fizičkim poslužiteljima

2. Platforma kao usluga (engl. PaaS, Platform-as-a-Service)

- Model koji nudi cjelokupnu platformu hardversku i softversku na kojoj tvrtke mogu pokretati vlastite poslovne aplikacije koje se nalaze na njihovoj infrastrukturi

¹⁰ Pejić Bach, M. et al. (2016) Informacijski sustavi u poslovanju. Zagreb:Ekonomski fakultet Zagreb. str. 212-213

¹¹ Pejić Bach, M. et al. (2016) Informacijski sustavi u poslovanju. Zagreb:Ekonomski fakultet Zagreb. str.211-212

3. Softver kao usluga (engl.SaaS, Software-as-a-Service)

- Naziva se još i kao softver na zahtjev
- Postojeće aplikacije koje se nalaze na internetu nude se privatnim i poslovnim korisnicima

Prednosti:

- Plaća se usluga ovisno o tome koliko se koristi
- Raspoloživost najnovije verzije programske podrške
- Programskoj podršci i podacima moguće je pristupiti s različitih lokacija
- Stalna dostupnost i raspoloživost usluge u oblaku uz pristup internetu 24x7
- Podaci su centralizirani i pohranjeni na jednom mjestu te stalno dostupni
- Smanjeni troškovi nadogradnje i održavanja softvera
- Nema izravnih troškova ulaganja u hardver te njegovog održavanja
- Oblak omogućava stalne usluge antivirusne zaštite i arhiviranja (engl. Back up) podataka.

Nedostaci:

- Ukoliko je internetska veza slaba dolazi do problema dostupnosti te nije moguće koristiti uslugu računarstva u oblaku
- Problem sigurnosti - razvitkom i intenzivnom uporabom novih digitalnih tehnologija postoji veća mogućnost izloženosti raznim napadima i prijetnjama
- Kompanije ne žele svoje povjerljive poslovne informacije stavljati u oblak
- Mogućnost krađe, prodaje i zlouporabe podataka (podaci o korisnicima, dokumenti, baze podataka, podaci o ponašanju korisnika i korištenju aplikacija)
- Mogućnost prisluškivanja komunikacije između korisnika i pružatelja usluge oblaka
- Problem zakonske regulative u području računarstva u oblaku

4.2. Oracle Cloud

Oracle Cloud je usluga računarstva u oblaku tvrtke Oracle Corporation koja nudi kompletan set usluga u oblaku. Platforma podržava mnoge otvorene standarde, otvorena rješenja, kao i mnoštvo baza podataka, programskih jezika, alata i okvira, uključujući Oracle softver, softver

otvorenog koda i sustave drugih proizvođača. Prema načinu primjene računarstva u oblaku Oracle Cloud nudi:¹²

Infrastruktura kao usluga (IaaS)

Infrastruktura kao usluga nudi računalne resurse visokih performansi kao i kapacitet za pohranu podataka u fleksibilnoj virtualnoj mreži koja je sigurna i uvijek dostupna korisniku usluga. Oracle Cloud Infrastructure uključuje sljedeće usluge:

Compute: Virtualne mašine (enlg. Virtual Machine) za različite vrste opterećenja i karakteristika performansi s mogućnošću skaliranja prema na više i na manje.

Storage: Usluga pruža pohranu podataka i arhivsku pohranu s mogućnošću baze podataka, analitike, sadržaja i drugih aplikacija putem uobičajenih protokola i API-ja.

Networking: Pruža mrežu s potpuno podesivim IP adresama, podmrežama, usmjeravanjem i vatrozidima za podršku novih ili postojećih privatnih mreža.

Governance: Pruža upravljanje identitetom i pristupom, ima mogućnosti provjere integriteta podataka, mogućnosti sljedivosti i upravljanja pristupom.

Database: Usluga omogućuje stvaranje baza podataka Oracle na zahtjev u okruženju oblaka pomoću stvarnih aplikacijskih klastera, sigurnost podataka i detaljne kontrole.

Load Balancing: Usluga nudi mogućnost uravnoteženja opterećenja za automatsko usmjeravanje prometa kroz domene dostupnosti.

FastConnect: Usluga pruža privatnu povezanost u lokalnim mrežama i mrežama u oblaku.

Platforma kao usluga (PaaS)

Platforma kao usluga pruža platformu za razvoj, implementaciju i upravljanje aplikacijama bez potrebe izgradnje i održavanja infrastrukture uključujući poslužitelje, pohranu, mrežu ili operativne sustave. Iako ne instaliraju i ne konfiguriraju komponente infrastrukture korisnici imaju kontrolu nad aplikacijama i konfiguracijama sustava. Oracle Cloud Platform uključuje sljedeće usluge:

Data Management: Platformu za upravljanje bazama podataka, uključujući OLTP, skladištenje, veliki podaci (engl. Big Data), strojno učenje, pretraživanje teksta, analitiku slika i dubinsko učenje.

Application Development: za razvoj aplikacija oblak nudi otvorenu platformu za razvoj aplikacija utemeljenu na standardima za izgradnju, implementaciju i upravljanje aplikacijama.

¹² Spec India (2019) Oracle Cloud Services: Detailed Overview And Benefits [online]. Dostupno na: <https://www.spec-india.com/blog/oracle-cloud-services-guide> [16.travnja.2020]

Usluge uključuju Javu, mobilne uređaje, razmjenu poruka, oblak kontejnera aplikacija, oblak za programere, vizualni graditelj, API katalog, platformu za umjetnu inteligenciju, Blockchain.

Integration: Usluga koja nudi adaptere za integriranje aplikacija instaliranih kod korisnika i aplikacija u oblaku. Mogućnosti uključuju integraciju i replikaciju podataka, upravljanje API-jem, analitiku integracije, zajedno s migracijom i integracijom podataka. Nude usluge kao što su Cloud Platform Integration Data, Cloud Integrator Data Cloud, GoldenGate Cloud Service, Integration Cloud, Process Cloud Service, API Cloud Platform Cloud Service, Apiary Cloud Service i SOA Cloud Service.

Business Analytics: Usluga Business Analytics koja može analizirati i generirati preglede iz podataka u različitim aplikacijama, skladištima podataka. Usluge koje se nude uključuju analitički oblak, poslovnu inteligenciju, otkrivanje velikih podataka, pripremu velikih podataka, vizualizaciju podataka.

Security: Osigurava sigurnosne mehanizme za siguran pristup hibridnom okruženju i nadzor nad njim te ispunjavanje zahtjeva za informatičkim upravljanjem i usklađivanjem.

Management: Pruža integriranu platformu za praćenje, upravljanje i analitiku. Ova platforma također koristi strojno učenje i velike podatke na skupu operativnih podataka. Platforma se koristi za poboljšanje IT stabilnosti, sprečavanje grešaka kod aplikacija i jačanje sigurnosti. Ponuđene usluge uključuju nadzor performansi aplikacija, nadzor infrastrukture, analitiku sistemskih zapisa, orkestraciju, konfiguraciju i usklađenost.

Softver kao usluga (SaaS)

Oracle nudi SaaS aplikacije poznate i kao Oracle Cloud Applications. Te se aplikacije nude u raznim proizvodima, industrijskim sektorima. Sada su u ponudi sljedeće aplikacije:

Customer Experience (CX), Human Capital Management (HCM), Enterprise Resource Planning (ERP), Supply Chain Management (SCM), Enterprise Performance Management (EPM), Internet of Things Applications (IoT), SaaS Analytics, Industry Solutions (Communications, Financial Services, Consumer Goods, High Tech and Manufacturing, Higher Education, Hospitality, Utilities), Block-Chain Cloud Service u suradnji sa SAP, IBM and Microsoft, Blockchain Applications.

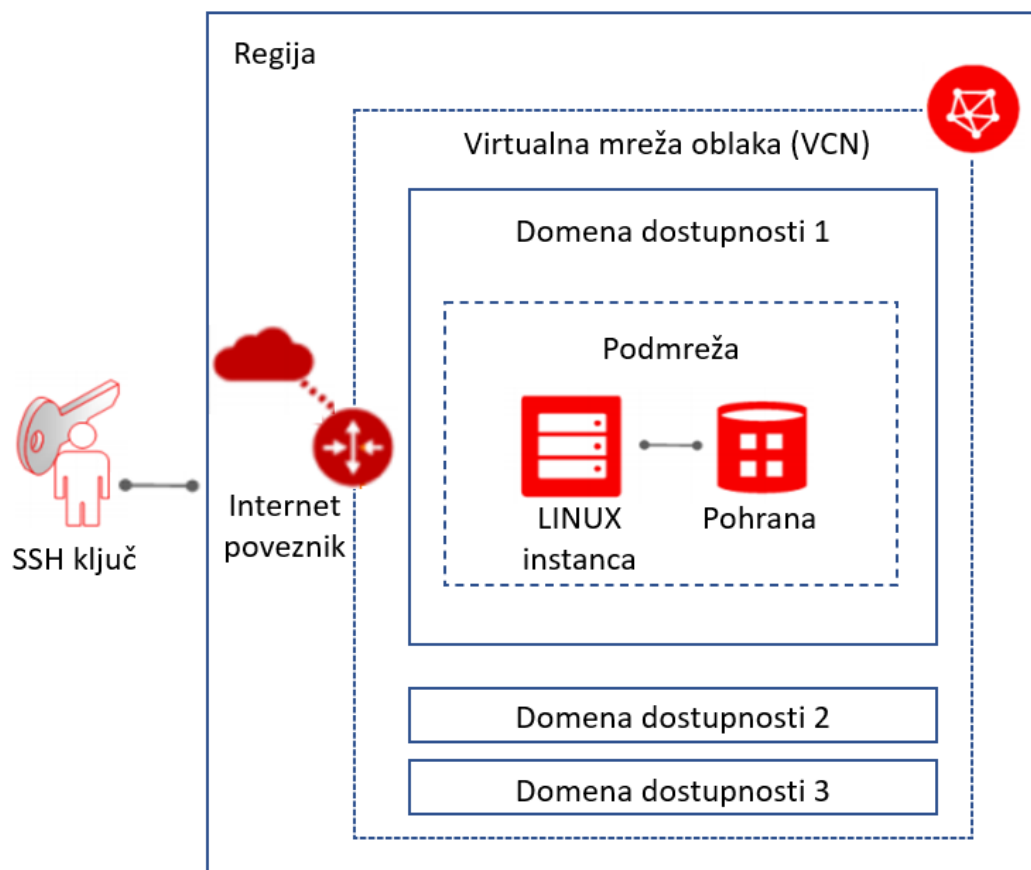
Podaci kao usluga (DaaS)

Ova usluga je poznata i kao Oracle Data Cloud (ODC). Platforma analizira i objedinjuje podatke o tvrtkama i potrošačima. Oracleovi podaci kao usluga pružaju pristup više od 300

milijuna zapisa o tvrtkama i 100 milijuna podataka o kontaktima. Usluga omogućava korisnicima da mogu imati najnovije ažurne podatke o poslovnim kontaktima diljem cijelog svijeta.

Oracle Cloud fizički je smješten u više regija u svijetu. Svaka regija ima više domena dostupnosti (engl. Availability Domain) koje čine jedan ili više podatkovnih centara koji se nalaze u nekoj regiji. Resursi Oracle Cloud Infrastrukture su specifični za regiju, poput virtualnog oblaka mreže ili domene. Domene dostupnosti su konfigurirane tako da su međusobno izolirane da bi bile otporne na interne greške kao i vanjske utjecaje. Kada korisnici konfiguriraju svoje usluge u oblaku, mogu koristiti više domena dostupnosti kako bi osigurali visoku dostupnost i raspoloživost svojih resursa.

Slika 5. Arhitektura regije Oracle Cloud-a



Izvor: Izrada autora prema Oracle (2020) Oracle Cloud Infrastructure Getting Started Guide [online]. Dostupno na: https://docs.cloud.oracle.com/en-us/iaas/pdf/gsg/OCI_Getting_Started.pdf [12.lipnja.2020]

5. RESTFUL WEB SERVISI

U ovom poglavlju bit će opisani elementi REST sustava i karakteristike odnosno ograničenja REST arhitekture koje sustav mora zadovoljavati da bi se smatrao RESTful sustavom. Elemente koji čine REST arhitekturu i ograničenja arhitekture u svom doktorskom radu *Architectural Styles and the Design of Network-based Software Architectures* opisao je R. T. Fielding.

5.1. Elementi REST arhitekture

REST stil je apstrakcija arhitektonskih elemenata unutar distribuiranog hipermedijskog sustava. REST ignorira detalje implementacije komponenata, sintaksa protokola kako bi se usredotočili na uloge komponenti i ograničenja u njihovoj interakciji s drugim komponentama. REST uključuje temeljna ograničenja na komponente, konektore i podatke koji definiraju osnovu Web arhitekture, a time i suštinu njihovog ponašanja u mrežnim aplikacijama.¹³

5.1.1 Elementi podataka

Komponente REST-a komuniciraju prijenosom reprezentacije resursa u formatu koji odgovara nekom od postojećih setova standardnih tipova podataka, odabranih dinamički na temelju mogućnosti ili potreba primatelja i prirode resursa.

Tablica 1. REST elementi podataka

Element podataka	Web primjeri
Resurs	Meta hipertekst referenca
Identifikator resursa	URL, URN, URI
Reprezentacija	HTML, JSON, XML dokumenti, JPEG slike
Metapodaci reprezentacije	Vrsta medija (engl. media type)
Metapodaci resursa	Izvorni link (enlg.source link), zamjenski link (enlg. alternates link)
Kontrolni podaci	Kontrola predmemorije (engl. cache-control)

Izvor: Izrada autora prema Fielding, R. T. (2000) *Architectural Styles and the Design of Network-based Software Architectures* [online]. Dostupno na: https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf [20.veljače.2020]

¹³ Fielding, R. T. (2000) *Architectural Styles and the Design of Network-based Software Architectures* [online]. Dostupno na: https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf [20.veljače.2020] str. 86-97

Za korisnika nije bitno je li reprezentacija resursa u istom obliku kao izvor ili je izvedena iz izvora. To ostaje skriveno iza sučelja.

5.1.2 Konektori

REST koristi razne vrste konektora (priključaka) kako bi objedinio aktivnosti pristupa resursima i prijenosa reprezentacija resursa. Konektori predstavljaju apstraktno sučelje za komunikaciju komponenata, poboljšavajući jednostavnost, osiguravajući razdvajanje zadataka i skrivanje temeljne implementacije resursa i komunikacijskih mehanizama.

Tablica 2. REST konektori

Konektor	Web primjer
Klijent	Šalje zahtjeve poslužitelju i prima odgovore (libwww)
Poslužitelj	Odgovara na zahtjeve klijenta (Apache API, NS API)
Predmemorija	Predmemorija preglednika (engl. browser cahce)
Rješavač	DNS
Tunel	SOCKS, SSL

Izvor: Izrada autora prema Fielding, R. T. (2000) Architectural Styles and the Design of Network-based Software Architectures [online]. Dostupno na: https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf [20.veljače.2020]

Osnovni tipovi konektora su konektor klijent i konektor poslužitelja. Osnovne razlike između njih su što klijent počinje komunikaciju slanjem zahtjeva. Poslužitelj osluškuje konekcije i odgovara na zahtjev. Sljedeći tip konektora je konektor predmemorije. Predmemorija se može nalaziti na sučelju konektora klijenta ili konektora poslužitelja. Klijent može koristiti predmemoriju za da bi smanjio potrebu za slanjem zahtjeva poslužitelju. Poslužitelj može koristiti predmemoriju kako bi izbjegao ponavljanje postupka generiranja odgovora ako već ima odgovore pohranjene u predmemoriji. Rješavač (engl. Resolver) prevodi djelomične ili potpune identifikatore resursa u informacije o mrežnoj adresi potrebne za uspostavljanje veze. Primjer rješavača je DNS (engl .Domanin Name Service) koji nazive poslužitelja pretvara u IP adrese na Internetu. Tunel (engl.Tunell) je tip konektora koji prenosi komunikaciju preko prepreka na mreži kao što je vatrozid.

5.1.3. Komponente

Korisnički agent (engl. User agent) koristi konektor klijenta kako bi poslao zahtjev i postao primatelj odgovora. Primjer korisničkog agenta je Web preglednik. U ovom radu primjer korisničkog agenta je Postman. Izvorni poslužitelj (engl. Origin server) radi reprezentaciju svojih resursa i mora biti krajnji primatelj svakog zahtjeva koji namjerava mijenjati vrijednost tih resursa. Posrednik (engl. Proxy) je komponenta odabrana od strane klijenta za pružanje sučelja za prijevod podataka, poboljšanje performansi ili sigurnosnu zaštitu. Poveznik (engl. Gateway) je komponenta odabrana od strane mreže ili poslužitelja za pružanje sučelja za prijevod podataka, poboljšanje performansi ili sigurnosnu zaštitu.

Tablica 3. REST komponente

Komponenta	Web primjer
Izvorni poslužitelj	Apache HTTPD, Microsoft IIS
Poveznik	Squid, CGI
Posrednik	CERN Proxy, Netscape Proxy, Gaulnet
Korisnički agent	Internet Explorer, FireFox, Postman

Izvor: Izrada autora prema Fielding, R. T. (2000) Architectural Styles and the Design of Network-based Software Architectures [online]. Dostupno na: https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf [20.veljače.2020]

5.2. Ograničenja REST arhitekture

REST je skup arhitektonskih ograničenja kojim se pokušava svesti na najmanju moguću mjeru kašnjenje i mrežna komunikacija uz istodobno maksimiziranje neovisnosti i skalabilnost implementacije komponenata.¹⁴ Postoji šest ograničenja REST-a koje sustav mora zadovoljiti da bi se smatrao RESTful sustavom:¹⁵

1. Klijent-poslužitelj

Prvo ograničenje je da sustavi moraju biti sastavljeni od klijenata i poslužitelja. Poslužitelji imaju resurse koje klijenti žele koristiti. Ovo ograničenje znači da može postojati više različitih

¹⁴ Fielding, R. T. (2000) Architectural Styles and the Design of Network-based Software Architectures [online]. Dostupno na: https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf [20.veljače.2020] str. 148

¹⁵ Ibid. str. 78-85

vrsta klijenata (web portali, mobilne aplikacije, itd.) koji pristupaju istom poslužitelju, a svaki se klijent može razvijati neovisno o ostalim klijentima i poslužitelju. Odvojenost klijenata i poslužitelja smanjuje složenost poslužitelja, jer se ne treba baviti s korisničkim sučeljem. Ovo je vjerojatno najmanje kontroverzno ograničenje REST-a, jer je model klijent-poslužitelj danas sveprisutan. Iako se HTTP protokol uglavnom koristi za komunikaciju klijent-poslužitelj važno je napomenuti da nema ograničenja koja obvezuju korištenje samo HTTP protokola. Mogu se koristiti i drugi protokoli kao na primjer FTP.

2. Bez stanja

Da bi se dodatno pojednostavila interakcija između klijenata i poslužitelja, drugo ograničenje je komunikacija bez stanja (engl.Stateless). To znači da se svi podaci o sesiji čuvaju na klijentu. Za održavanje sesije ne trebaju postojati kolačići, varijable sesije ili drugi mehanizmi za održavanje sesija. Svaki zahtjev mora sadržavati sve informacije potrebne da bi se mogao izvršiti, jer na poslužitelju ne postoje informacije o sesiji odnosno ne postoji kontekst koji se stvara prijavom na poslužitelja. Poslužitelj ne mora pratiti sesije, što pojednostavljuje i olakšava rad poslužitelja. Resursi na poslužitelju mogu se osloboditi kada je zahtjev završen. Korištenjem komunikacije bez stanja u REST aplikacijama postoji manja mogućnost grešaka nego kod web aplikacija gdje je praćenje sesija česti uzrok problema.

3. Predmemorija

Ograničenje u komunikaciji klijent-poslužitelj propisuje da odgovori s poslužitelja moraju biti označeni da se spremaju u predmemoriju (engl. Cacheable) ili da se ne spremaju u predmemoriju (engl. Non-cacheable). Dobro upravljanje predmemorijom može optimizirati komunikaciju klijent-poslužitelj odnosno smanjiti broj interakcija klijent-poslužitelj, što pozitivno doprinosi performansama sustava.

4. Uniformno sučelje

Ono što REST stvarno odvaja od ostalih arhitektura jest uniformno sučelje. Ovo ograničenje sučelja sastoji se od četiri pod ograničenja:

Identifikacija resursa

REST je usredotočen na resurse. Pod pojmom resursa smatra se bilo koji izvor informacija ili usluge dostupne na mreži. U poslovnom softveru resursi su obično subjekti iz poslovne domene

(npr. proizvodi, kupci, narudžbe). Na razini implementacije, često su tablice baza podataka s dodanom poslovnom logikom izložene kao resursi. Svaki resurs u RESTful dizajnu mora moći biti jedinstveno identificiran putem URI-a. URI (engl. Uniform Resource Identifier) je niz znakova koji jednoznačno identificira stvarni ili apstraktni resurs na mreži i zapravo predstavlja njegovu adresu. Svaki resurs koji se želi izložiti putem REST API-ja mora imati vlastiti URI.¹⁶ Primjer URI-ja koji se često koristi je: `schema://računalo:vrata/staza/datoteka#sidro`. Shema određuje protokol za pristup i transport sadržaja, računalo određuje ime računala na kojem se nalazi resurs, vrata je port za komunikaciju (najčešće se koristi 80 ili 443), slijedi naziv datoteke na poslužitelju. Sidro je posebno definirano mjesto unutar datoteke.

Manipulacija resursima putem reprezentacija

Drugo ograničenje uniformnog sučelja jest to što se resursima manipulira putem reprezentacija. To znači da klijent neizravno komunicira s resursom poslužitelja. Poslužitelj izlaže prikaz sadržaj resursa i prikazuje ga u neutralnom formatu. Najčešći format za REST API je JSON, koji se koristi u tijelu HTTP zahtjeva i odgovora. Prednost ovakvog pristupa resursima je što se izbjegava zavisnost klijenta od poslužitelja. To olakšava klijentima jer ne trebaju razumjeti osnovnu tehnologiju koju koristi svaki poslužitelj s kojim komuniciraju.

Poruke koje opisuju samo sebe

Treće ograničenje uniformnog sučelja je da svaki zahtjev ili odgovor mora sadržavati dovoljno informacija da budu razumljivi klijentu i poslužitelju. Svaka poruka mora imati vrstu medija (na primjer, "application/json" ili "application/xml") koja govori kako poruku treba pročitati odnosno obraditi. Iako HTTP nije obavezan kod RESTful servisa to je uobičajeni način prijenosa poruka. Ako se koristi HTTP treba koristiti GET, POST, PUT, DELETE metode. Ove četiri HTTP metode jasno su definirane, pa to pridonosi intuitivnosti i lakšem snalaženju korisnika REST API-ja. HTTP protokol i metode bit će detaljnije opisane u sljedećim poglavljima.

Veze među resursima

Ograničenje veze među resursima HATEOS (engl. Hypermedia as the engine of application state) kaže da prikaz resursa može sadržavati veze prema drugim resursima poput onih u HTML

¹⁶ Bažant, A. et al. (2004) Osnovne arhitekture mreža. Zagreb:Element. str. 283

stranicama. To olakšava upotrebu resursa jer ne treba gledati u dokumentaciju API-ja i kodirati URI-ove koje se želi pozvati.

5. Slojni sustav

Peto ograničenje kaže da klijent treba znati samo neposredni sloj s kojim komunicira, a ne treba znati ostale slojeve koji koristi poslužitelj. To znači da klijent ne zna da li komunicira s posrednikom ili stvarnim poslužiteljem. Na taj način poslužitelj može zvati više drugih poslužitelja kako bi obradio zahtjev i odgovorio klijentu. To omogućuje odvajanje poslovne logike od baze podataka, dodavanje sigurnosnog sloja i drugih slojeva potrebnih za obradu zahtjeva klijenta.

6. Kod na zahtjev

Ovo je jedino neobavezno ograničenje u REST-u. Kod na zahtjev, znači da poslužitelj može proširiti funkcionalnost klijenta tijekom izvršavanja, slanjem koda koji treba izvršiti (poput JavaScript-a). Ovo nije uobičajeni način korištenja REST API-ja ali bi mogao biti dobar način poboljšanja funkcionalnosti klijenata. Kod na zahtjev se vrlo često koristi kod web stranica gdje se sa servera preuzima JavaScript i izvodi u web pregledniku klijenta.

5.3. HTTP protokol

HTTP (engl. Hypertext Transfer Protocol) je oznaka za mrežni protokol koji se koristi za prijenos datoteka i različitih tipova podataka zajedničkim imenom nazvanih resursi na World Wide Web-u bilo da se radi o HTML datotekama, slikama, rezultatima različitih upita ili nečemu drugom¹⁷. HTTP protokol se koristi u računalnom modelu klijent – poslužitelj. Funkcionira po principu zahtjev–odgovor. U WWW okruženju primjer klijenta je web preglednik. U ovom radu za testiranje aplikacije za klijenta će se koristiti Postman. Klijent šalje HTTP zahtjev u obliku poruke poslužitelju. Poslužitelj vraća poruku odgovora klijentu.

5.3.1. HTTP zahtjev

U HTTP zahtjevu osim same poruke šalju se i druge informacije o zahtjevu. Slika 6 prikazuje elemente HTTP zahtjeva.

¹⁷ Fakultet elektrotehnike i računarstva Sveučilište u Zagrebu (2006) HTTP protokol [online]. Dostupno na: http://docbook.rasip.fer.hr/or_smil/10b_HTTP/Example/HTTP%20protokol.pdf [10.ožujka.2020], str. 3

Slika 6. HTTP zahtjev

Metoda	URI	HTTP verzija
Zaglavlje zahtjeva		
Tijelo zahtjeva		

Izvor: Izrada autora prema Vaqqas, M. (2014) RESTful Web Services: A Tutorial [online].
Dostupno na: <https://www.drdoobs.com/web-development/restful-web-services-a-tutorial/240169069> [18.veljače.2020]

Metoda je jedna od HTTP metoda GET, PUT, POST, DELETE. URI je adresa resursa na koji se šalje zahtjev. HTTP verzija je inačica HTTP-a, uglavnom je "HTTP v1.1". Zaglavlje zahtjeva sadrži metapodatke u formatu ključ, vrijednost. To su informacije o poruci i njenom pošiljatelju poput vrste klijenta, formatima koje klijent podržava, informacije o formatu tijela poruke, postavke predmemorije i druge. U zaglavlje zahtjeva klijent može staviti identifikacijske podatke za korištenje resursa (npr. korisnički identifikator, zaporku). Tijelo zahtjeva je stvarni sadržaj poruke. U metodama GET i DELETE tijelo zahtjeva je prazno jer se ne prenosi sadržaj poruke.

Slika 7. HTTP POST zahtjev

```
▼ POST https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/kupac/
POST /ords/prodaja/kupac/ HTTP/1.1

▼ Request Headers
Content-Type: "application/json"
User-Agent: "PostmanRuntime/7.25.0"
Accept: "*/*"
Postman-Token: "3b4f8e72-1454-48d9-8214-5a65d6a3b5fb"
Host: "qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com"
Accept-Encoding: "gzip, deflate, br"
Connection: "keep-alive"
Content-Length: 210

▼ Request Body 
{
  "oib": "02345678912",
  "naziv": "Tvrtka četiri d.o.o.",
  "telefon": "01/439612",
  "mobitel": "095/452186",
  "e-mail": "tvrtka.cetiri@gmail.com"
}
```

Izvor : Izrada autora iz alata Postman

5.3.2. HTTP odgovor

U HTTP odgovoru poslužitelj vraća kod odgovora koji sadrži status zahtjeva. Ovaj kod odgovora općenito je troznamenkasti HTTP statusni kod. Zaglavlje odgovora sadrži metapodatke o poruci odgovora. Tijelo za odgovor sadrži odgovor poslužitelja uz pretpostavku ako je zahtjev bio uspješan.

Slika 8. HTTP odgovor

HTTP verzija	Kod odgovora
Zaglavlje odgovora	
Tijelo odgovora	

Izvor: Izrada autora prema Vaqqas, M. (2014) RESTful Web Services: A Tutorial [online]. Dostupno na: <https://www.drdoobs.com/web-development/restful-web-services-a-tutorial/240169069> [18.veljače.2020]

Slika 9. Odgovor na POST zahtjev

```
▼ Response Headers
  Date: "Tue, 09 Jun 2020 19:05:42 GMT"
  Content-Type: "application/json"
  Transfer-Encoding: "chunked"
  Connection: "keep-alive"
  Content-Location: "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/kupac/9"
  ETag: "\"Ix/gJOYNYMmqBqmMoCg0qlpa1Zxd9GBsa6Zbxbgt9zdG0KZAGXmWwKooKcQc0tXEozqJy3KQgv7LieIX09RfZW==\""
  Location: "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/kupac/9"
▼ Response Body 
  {"id":9,"oib":"02345678912","ime":null,"prezime":null,"naziv":"Tvrtka četiri d.o.o.,"telefon":"01/439612","mobitel":"095/452
  ttps://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/kupac/9"},{"rel":"edit","href":"htt
  ef":"https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/metadata-catalog/kupac/item"}
  pac/"}}]
```

Izvor: Izrada autora iz alata Postman

5.3.3. Statusni kodovi

Statusni kod je troznamenasti cijeli broj u kojem prva znamenka određuje općenitu kategoriju odgovora:¹⁸

- 1xx označava samo informacijsku poruku
- 2xx označava neku vrstu uspjeha
- 3xx preusmjerava klijenta na drugi URL
- 4xx označava pogrešku na strani klijenta
- 5xx označava pogrešku na strani poslužitelja

Slijedi detaljniji opis najčešćih statusnih kodova:¹⁹

200 - U redu (engl. OK) . Zahtjev je uspješno obrađen. Ako postoji dokument u tijelu odgovora on predstavlja reprezentaciju nekog resursa.

201 – Stvoreno (engl. Created). Poslužitelj šalje ovaj statusni kod kada stvara novi resurs na zahtjev klijenta. Zaglavlja odgovora treba sadržavati lokaciju odnosno URI resursa koji je kreiran. Tijelo odgovora može sadržavati reprezentaciju resursa koji je kreiran i URI na taj resurs.

400 - Loš zahtjev (engl. Bad Request) Postoji problem na strani klijenta. Ako postoji dokument u tijelu odgovora on predstavlja poruku o grešci. Poruka o grešci može biti korisna za rješavanje problema.

¹⁸ Fakultet elektrotehnike i računarstva Sveučilište u Zagrebu (2006) HTTP protokol [online]. Dostupno na: http://docbook.rasip.fer.hr/or_smil/10b_HTTP/Example/HTTP%20protokol.pdf [10.ožujka.2020], str. 6

¹⁹ Richardson, L., Ruby, S. (2007) RESTful Web Services. Sebastopol, California: O'Reilly Media Inc. str. 372-374

404 - Nije pronađeno (engl. Not Found). Poslužitelj vraća ovu poruku kada klijent zahtjeva URI za koji ne postoji resurs.

500 - Interna pogreška poslužitelja (engl. Internal server error). Postoji problem na strani poslužitelja. Dokument u tijelu odgovora, ako ga ima, je poruka o grešci na strani servera.

5.4. Sigurnost

REST je nastao kao teorijska podloga za poboljšanje učinkovitosti i jednostavniji razvoj servisa, dok sigurnosna ograničenja nisu ugrađena u osnovnom modelu²⁰. To predstavlja problem jer se moraju dodavati vanjski zaštitni mehanizmi. Primarne funkcionalnosti obrade i prezentacije zahtjeva uglavnom su u prvom planu kod implementacije REST servisa. Sigurnosne funkcije servisa su često u drugom planu i one se dodaju naknadno. Ovakvi uvjeti vode povećanju ranjivosti REST servisa. Jedna od najčešćih ranjivosti REST servisa je autentifikacija korisnika uporabom jedinstvenog ključa. Davanje jedinstvenog identifikatora svakom resursu je osnovno načelo REST modela koje ima značajni nedostatak da ne koristi tajni podatak kojim se potvrđuje identitet korisnika koji je poslao zahtjev. Drugi česti izvor ranjivosti je korištenje HTTP protokola za razmjenu korisničkih podataka za autentifikaciju. Time se lozinka prenosi u izvornom obliku kroz nepouzdanu mrežu. REST servisi imaju slične ranjivosti kao i web aplikacije. Iako REST modelom nisu propisane konkretne metode za zaštitu servisa moguće je upotrijebiti neke osnovne mehanizme koji se koriste u web aplikacijama. Jedan od glavnih ograničenja REST-a u odnosu na web aplikacije je komunikacija bez stanja. Između klijenta i poslužitelja se ne stvaraju sesije pa poslužitelj ne može spremati stanje klijenta. Prilikom svakog zahtjeva klijent mora slati svoje podatke za autentifikaciju poslužitelju. Provjera autentičnosti klijenta kod svakog zahtjeva troši resurse i može značajno opterećivati sustav. Za provjeru autentičnosti klijenta često se upotrebljavaju mehanizmi HTTP osnovne autentifikacije, HTTP autentifikacije na temelju reprezentacije, autentifikacije na temelju tokena.²¹

HTTP osnovna autentifikacija

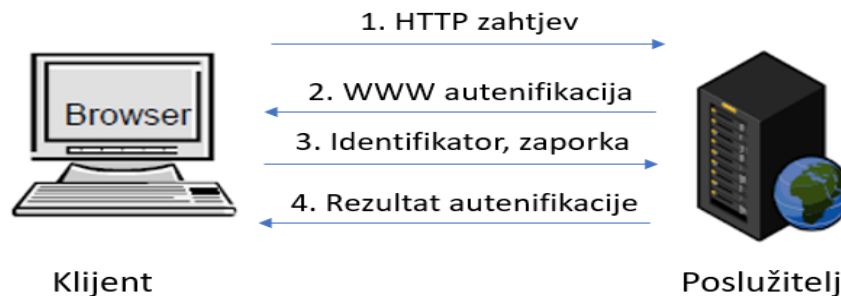
HTTP osnovna autentifikacija ili provjera identiteta (engl. HTTP Basic Authentication) je mehanizam provjere identiteta koji koristi identifikator i lozinku klijenta za provjeru

²⁰ Centar Informacijske Sigurnosti (2012) Ispitivanje sigurnosti web servisa [online]. Dostupno na: <https://www.cis.hr/files/dokumenti/CIS-DOC-2012-02-040.pdf> [21.ožujka.2020], str 14

²¹ Sungchul, L. et al. (2015) A Method for Secure RESTful Web Service A Method for Secure RESTful Web Service [online]. Dostupno na: https://www.researchgate.net/publication/298262824_A_Method_for_Secure_RESTful_Web_Service_A_Method_for_Secure_RESTful_Web_Service [19.ožujka.2020]

autentičnosti zahtjeva klijenta. Identifikator i lozinka stavljaju se u zaglavlje HTTP zahtjeva. Slika 10 prikazuje faze provjere autentičnosti zahtjeva. Kada primi zahtjev poslužitelj provjerava identifikator i lozinku klijenta. U slučaju da nije prošla provjera autentičnosti server šalje odgovora sa HTTP kodom 401.

Slika 10. HTTP osnovna provjera identiteta



Izvor: Izrada autora prema Sungchul, L. et al. (2015) A Method for Secure RESTful Web Service A Method for Secure RESTful Web Service [online]. Dostupno na: https://www.researchgate.net/publication/298262824_A_Method_for_Secure_RESTful_Web_Service_A_Method_for_Secure_RESTful_Web_Service [20.veljače.2020]

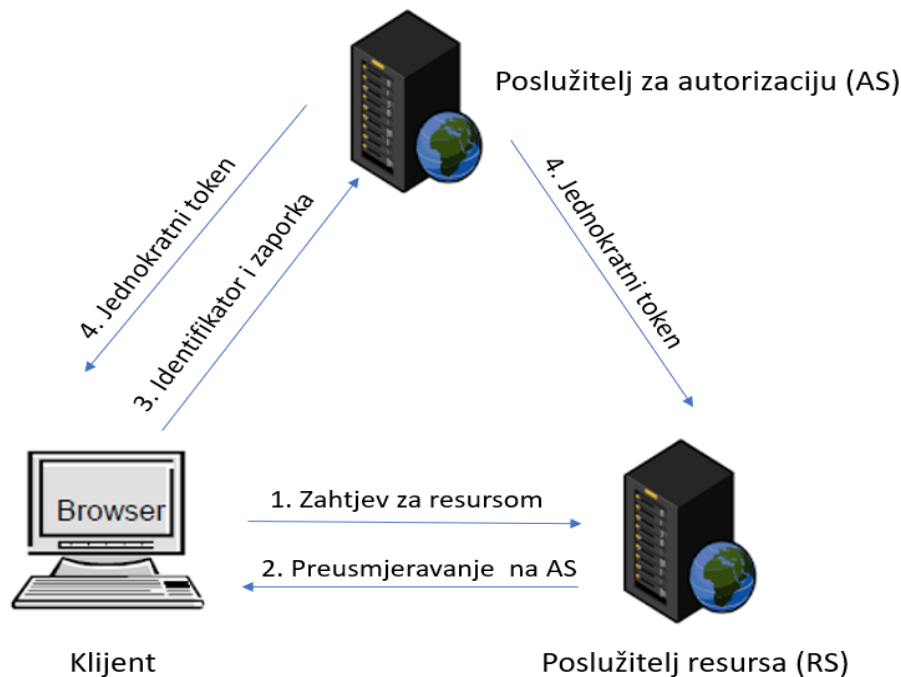
HTTP autentifikacija na temelju reprezentacije

HTTP autentifikacija na temelju reprezentacije (engl. HTTP Digest Authentication) je napredna verzija osnovne provjere identiteta. Identifikator i lozinka klijenta šalju se zaštićene pomoću nekog od algoritma sažetka ili reprezentacije (npr. HASH algoritmi MD5, SHA-1, SHA-2). U zahtjev se također dodaje i vremenski pečat koji se provjerava na poslužitelju i može zaštititi poruku klijenta od ponovne reprodukcije.

Autentifikacija na temelju tokena

Autentifikacija na temelju tokena (OAuth) koristi jednokratni token umjesto korisničkog identifikatora i zaporkke. Slika 11 prikazuje način provjere autentičnosti na temelju tokena. Prilikom upita poslužitelj resursa (RS) preusmjerava klijenta na autorizacijski poslužitelj (AS) gdje se klijent prijavljuje sa svojim identifikatorom i zaporkom. Nakon uspješne autentifikacije autorizacijski poslužitelj daje klijentu token kojim će se predstaviti poslužitelju resursa. Isti taj token dobiva poslužitelj resursa kako bi mogao provjeriti poruke koje mu šalje klijent. OAuth metoda je često implementirana od velikih tvrtki za Web servise poput Twittera, Yahooa, Googlea, Facebooka, Microsofta.

Slika 11. Autentifikacija na temelju tokena



Izvor: Izrada autora prema Sungchul, L. et al. (2015) A Method for Secure RESTful Web Service A Method for Secure RESTful Web Service [online]. Dostupno na: https://www.researchgate.net/publication/298262824_A_Method_for_Secure_RESTful_Web_Service_A_Method_for_Secure_RESTful_Web_Service [20. veljače. 2020]

5.5. JSON format

Format podataka JSON (engl. Java Script Object Notation) omogućava aplikacijama da komuniciraju preko mreže, obično putem RESTful API-ja. To znači da JSON označava objekte po uzoru na JavaScript programski jezik. JSON je tehnološki neutralan i portabilan. Svi moderni jezici (npr. Java, JavaScript, Ruby, C#, PHP, Python i Groovy) i platforme pružaju izvrsnu podršku za generiranje (serializiranje) i korištenje (deserializiranje) JSON podataka. JSON ima jednostavnu sintaksu tako da je razumljiv i lako čitljiv.²² Prema specifikaciji ekstenzija .json je standardna vrsta datoteke kod pohranjivanja JSON podataka u datotečne sustave. Kod HTTP protokola tijelo JSON poruke se obilježava pomoću tipa medija application/json. JSON tipovi podataka su: par ključ/vrijednost, objekt, polje. Par ključ/vrijednost (engl. Name or Key/value pair) sastoji se od para naziva atributa i vrijednosti atributa. Ključ i vrijednost odvojeni su dvotočkom. Objekt je skup parova ključ/vrijednost. Objekt počinje lijevom vitičastom zagradom '{' i završava desnom vitičastom zagradom '}'.

²² Marrs, T. (2017) JSON at Work. Sebastopol California: O'Reilly Media Inc. str. 3

Elementi objekta odvojeni su zarezom. Polje je kolekcija vrijednosti. Vrijednosti mogu biti parovi ključ/vrijednosti, objekti ili polja. Polje počinje lijevom uglatom zagradom '[' i završava desnom uglatom zagradom ']'. Elementi polja odvojeni su zarezom. JSON tipovi vrijednosti mogu biti: objekt, polje, niz (engl. String), broj, logička vrijednost (engl. Boolean), null.²³

Slika 12. Primjer poruke u JSON formatu

```
{
  "items": [
    {
      "proizvod_id": 2,
      "naziv_proizvoda": "banana",
      "cijena": 9.9
    },
    {
      "proizvod_id": 3,
      "naziv_proizvoda": "limun",
      "cijena": 12.9
    }
  ]
}
```

Izvor: Izrada autora

5.6. Metode

Za upravljanje resursima u RESTful sustavima koriste se RESTful API metode koje obavljaju operacije dohvata sadržaja resursa, stvaranje novih sadržaja, promjenu i brisanje sadržaja resursa. HTTP metode koje se koriste u RESTful API-ju su: GET, POST, PUT, DELETE.

Karakteristike HTTP metoda su sigurnost (engl. Safety) i idempotentnost (engl. Idempotency)²⁴. Sigurne metode su one metode koje ne mijenjaju sadržaj resursa. Idempotentne metode su one za koje vrijedi da će uvijek kreirati isti rezultat, neovisno o tome koliko puta budu pozvane.

Prikaz karakteristika HTTP metoda dan je u tablici 4.

Tablica 4. Karakteristike HTTP metoda

Metoda	Sigurnost	Idempotentnost
GET	DA	DA
HEAD	DA	DA
OPTIONS	DA	DA
PUT	NE	DA
DELETE	NE	DA
POST	NE	NE

Izvor: Izrada autora prema Allamaraju, S. (2010) RESTful Web Services Cookbook. Sebastopol, California: O'Reilly Media Inc.

²³ Marrs, T. (2017) JSON at Work. Sebastopol California: O'Reilly Media Inc. str. 8-11

²⁴ Allamaraju, S. (2010) RESTful Web Services Cookbook. Sebastopol, California: O'Reilly Media Inc. str. 9-12

RESTful API metode mogu se usporediti sa SQL naredbama za manipulaciju sa podacima u bazi podataka. SQL CRUD (engl. Create Read Update Delete) sličan je RESTful API metodama²⁵.

Usporedba RESTful API metoda i CRUD SQL naredbi dana je u tablici 5.

Tablica 5. Usporedba API metoda

CRUD	RESTful API	SQL
Create	POST	insert
Read	GET	select
Update	PUT	update
Delete	DELETE	delete

Izvor: Izrada autora prema Allamaraju, S. (2010) RESTful Web Services Cookbook. Sebastopol, California: O'Reilly Media Inc.

5.7. Primjena

Jedan od najvećih zagovornika REST protokola je Google. Servisi kao što su Google Translate API, Google Earth API, Google Maps API itd. Microsoft također koristi REST servise kao što Locations API, Traffic API i slični te se tako ti servisi mogu koristiti i u privatnim aplikacijama.²⁶

U Hrvatskoj također postoje RESTful sustavi u javnoj upotrebi.

Sustav Dabar²⁷ je digitalni akademski arhiv i repozitorij. To je jedno od središnjih mjesta za pouzdanu i trajnu pohranu digitalne imovine i sadržaja važnih za ustanove iz sustava znanosti i visokog obrazovanja u Republici Hrvatskoj. Sveučilišni računski centar Sveučilišta u Zagrebu (Srce) je javna ustanova koja kao dio redovne djelatnosti planira i osigurava računalno-komunikacijsku i programsku infrastrukturu potrebnu za funkcioniranje Dabra. Pri tome se Srce oslanja na suradnju i podršku šire zajednice. Fakultet organizacije i informatike (FOI) je koristeći Dabar REST API programsko sučelje pohranio više od 2.500 završnih radova u institucijski repozitorij. U Dabru približno 1.200 disertacija pohranjeno je putem Dabar REST API-ja u Repozitorij Sveučilišne knjižnice Rijeka.

²⁵ Allamaraju, S. (2010) RESTful Web Services Cookbook. Sebastopol, California: O'Reilly Media Inc., str. 30

²⁶ Lozić, D., Šimec, A. (2014) Pametna komunikacija na Internetu preko REST protokola [online]. Dostupno na: https://bib.irb.hr/datoteka/713376.2014_MIPRO_Pametna_komunikacija_na_Internetu_preko_REST_protokola.pdf [10.travnja.2020]

²⁷ Digitalni akademski arhivi i repozitoriji (2017) [online]. Dostupno na: <https://dabar.srce.hr/2017-03-07/pohranjen-znacajan-broj-radova-koristeci-dabar-rest-api-programsko-sucelje> [21.svibnja.2020]

ISVU (Informacijski Sustav Visokih Učilišta)²⁸ je projekt Ministarstva znanosti, obrazovanja i sporta pokrenut sredinom 2000. godine u sklopu programa informatizacije visokih učilišta u Republici Hrvatskoj. ISVU REST API je namijenjen programskoj interakciji s ISVU-om, to je programsko sučelje za pristup podacima i procedurama (resursima i metodama) pojedinog visokog učilišta u ISVU. API je dizajniran na način kako bi mogao obrađivati veliki broj zahtjeva raznih informacijskih sustava koji će ga koristiti.

eVisitor²⁹ je jedan od primjera primjene . Kroz eVisitor Web API sučelje moguće je izvršiti sve operacije koje su dostupne kroz web sučelje same eVisitor aplikacije. Između ostalog to su: prijava turista, odjava turista, izmjena prijave turista, poništenje prijave turista, otvaranje novog obveznika, pregled postojećih obveznika, otvaranje novog objekta, deaktivacija objekta, pokretanje okvirnog obračuna.

5.8. Prednosti i nedostaci

Po broju podržanih platformi kao i po broju korisnika danas su RESTful servisi dominantni prema ostalim web servisima. Bez obzira na dominaciju kod odabira web servisa odnosno arhitekture u projektima treba sagledati prednosti i nedostatke web servisa i odabrati web servis koji će najbolje riješiti postavljeni zadatak. Prednosti i nedostaci RESTful web servisa su:³⁰

Prednosti:

- REST koristi manje poruke, efikasan je u komunikaciji osobito kada se koriste poruke u JSON formatu
- Osim JSON-a može koristiti i druge formate poruka
- Efikasno koristi HTTP naredbe (GET, PUT, POST, DELETE)
- Za sigurnost koristi HTTP standarde
- Jednostavan je za implementaciju
- REST mogu koristiti standardni REST klijenti
- Podržava komunikaciju bez stanja što pojednostavljuje stranu poslužitelja

²⁸ ISVU [online]. Dostupno na:<https://www.isvu.hr/apipropa/dokumentacija/index.html> [21.svibnja.2020]

²⁹ eVisitor (2019) Web API [online]. Dostupno na: <https://www.evisitor.hr/eVisitorWiki/Javno.Web-API.ashx> [19.svibnja.2020]

³⁰ Halili, F., Ramadani E. (2018) Web Services: A Comparison of Soap and Rest Services [online]. Dostupno na:https://www.researchgate.net/publication/323456206_Web_Services_A_Comparison_of_Soap_and_Rest_Services [2.svibnja.2020]

Nedostaci:

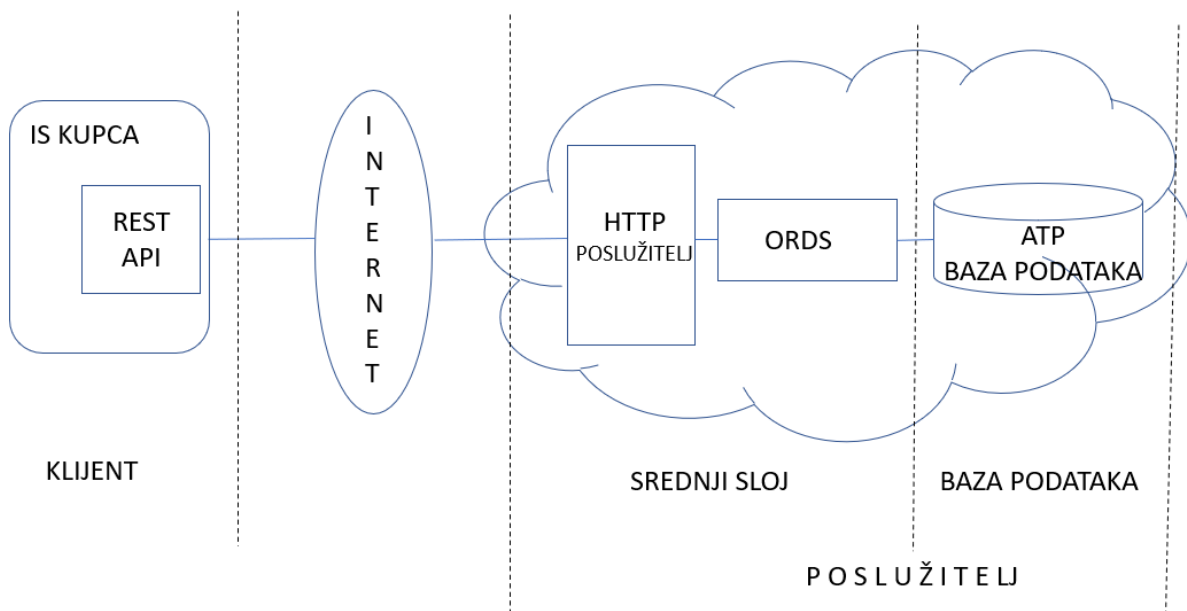
- REST zahtjevi osobito GET metoda nisu prikladni za prijenos velike količine podataka
- REST nije pouzdan kao SOAP protokol
- REST nema propisane standarde web servisa kao što su standardi za sigurnost i transakcije
- Latencija – moguća su kašnjenja u obradi zahtjeva

6. ARHITEKTURA REST SERVISIA

6.1. Općenito

B2B aplikacija prodaje u ovom radu koristi tro-slojnu arhitekturu. Zahtjeve šalje klijent putem REST API-ja. HTTP zahtjevi dolaze na HTTP poslužitelj. HTTP poslužitelj prosljeđuje zahtjeve ORDS poslužitelju koji ih obrađuje i pretvara u upite baze podataka. Sva obrada podataka i poslovna logika izvršavaju se u bazi podataka. Kada završi obrada zahtjeva u bazi podataka odgovor se vraća ORDS poslužitelju koji pretvara odgovor u JSON format i prosljeđuje ga HTTP poslužitelju koji vraća odgovor klijentu. Ova tro-slojna arhitektura koju je implementirao Oracle odstupa od tro-slojne arhitekture opisane u literaturi³¹. Oracle poslovnu logiku stavlja u bazu podataka umjesto u srednji sloj kako predviđa teorija tro-slojne arhitekture. Arhitektura produkcijskog okruženja prikazana je na slici 13. Srednji sloj i baza podataka nalaze se u Oracle Cloudu.

Slika 13. B2B aplikacija prodaje - produkcijsko okruženje

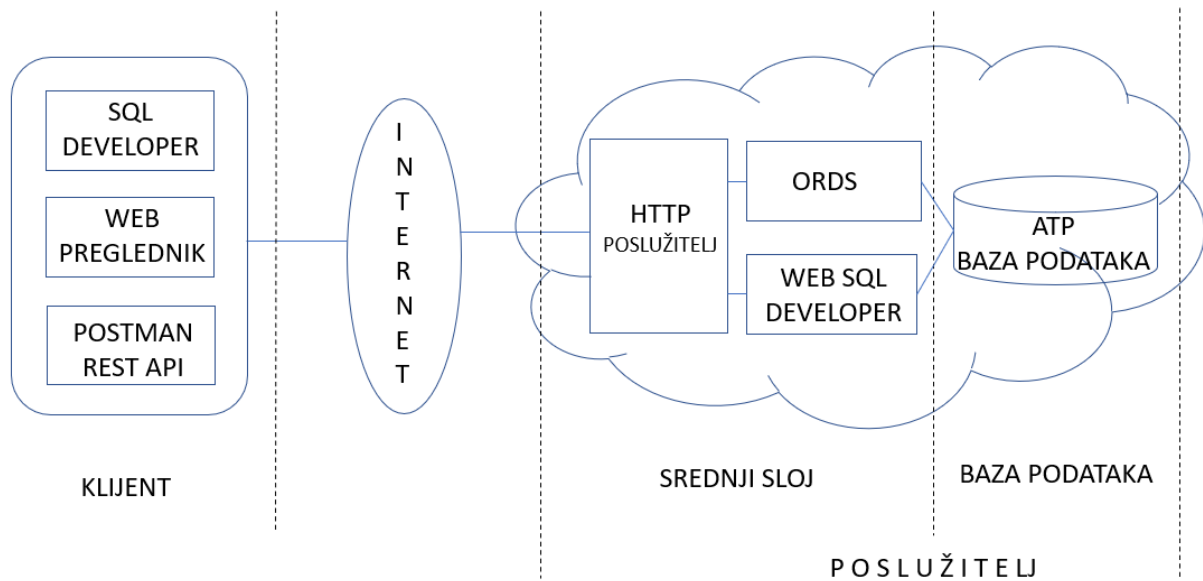


Izvor: Izrada autora

³¹ Kukhnavets P. (2019) How to Organize Application Code With 3-Tier Architecture [online]. Dostupno na: <https://cuspy.io/blog/how-organize-application-code-with-3-tier-architecture/> [2.ožujka.2020]

Razvojno okruženje koje je korišteno u ovom radu prikazano je na slici 14. U razvojnom okruženju nalaze se i komponente potrebne za razvoj i testiranje aplikacije. To su Oracle SQL Developer i Postman API klijent.

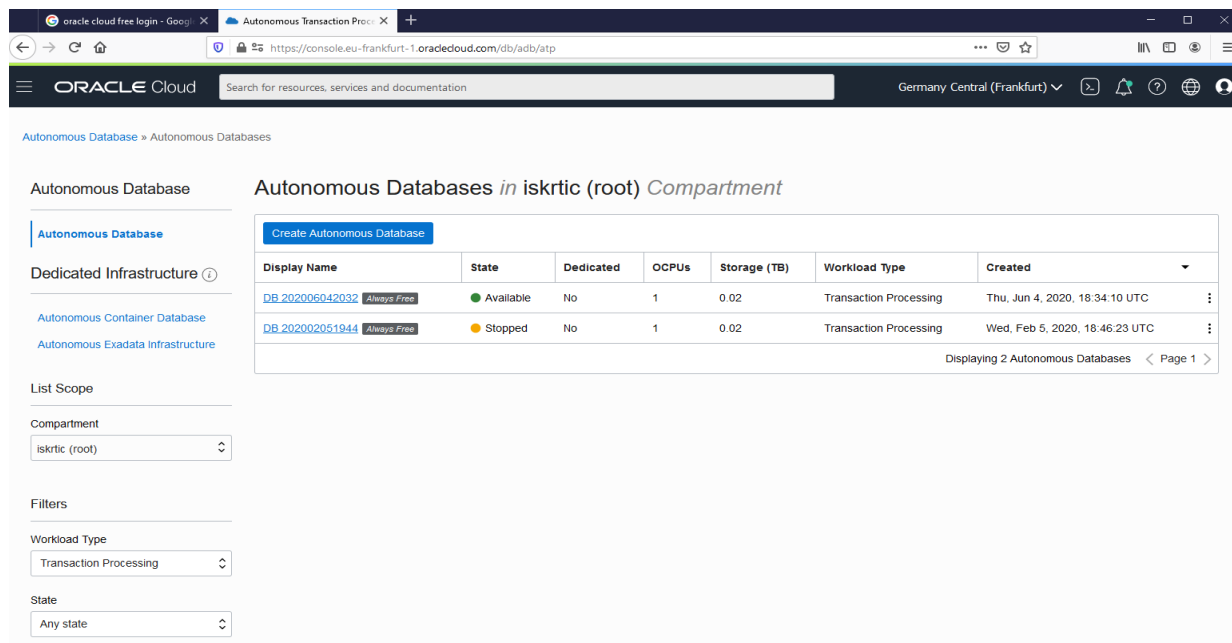
Slika 14. B2B aplikacija prodaje - razvojno okruženje



Izvor: Izrada autora

U ovom radu korišten je Oracle Cloud Free Tier (Always Free Services). Besplatnom registracijom na servis omogućene su usluge: Compute, Storage, Networking, Storage, Networkong, Autonomus Data Warehouse, Autonomus Transaction Processing, Application Integration, Identity, Security, Governance. Pristup oblaku je na regionalnom centru Germany Central (Frankfurt). Za pristup uslugama, upravljanje oblakom koristi se konzola u web pregledniku.

Slika 15. Konzola Oracle oblaka



Izvor: Oracle Cloud konzola

6.2. Komponente REST servisa

Baza podataka

U radu je korištena Oracle baza podataka 18c Autonomous Transaction Processing, autonomna baza podataka koja se može besplatno koristiti kao dio programa Oracle Cloud Infrastructure (Always Free). Besplatne autonomne baze podataka imaju 8 GB memorije, 20 GB prostora za pohranu, 1 CPU, maksimalni broj sesije prema bazi je 20.³² Mogu se konfigurirati bilo za autonomnu obradu transakcija ili za rad autonomnog skladišta podataka. Kreiranje instance baze podataka je jednostavno. Izvodi se u konzoli oblaka sa nekoliko klikova mišem. Autonomna baza podataka samostalno odrađuje zadatke održavanja.³³

- Izrada sigurnosne kopije
- Instaliranje zakrpa
- Instaliranje novih verzija
- Optimizaciju rada baze podataka

³² Oracle (2020) Oracle Cloud Infrastructure User Guide [online]. Dostupno na: https://docs.cloud.oracle.com/en-us/iaas/pdf/ug/OCI_User_Guide.pdf [12.lipnja.2020], str. 31

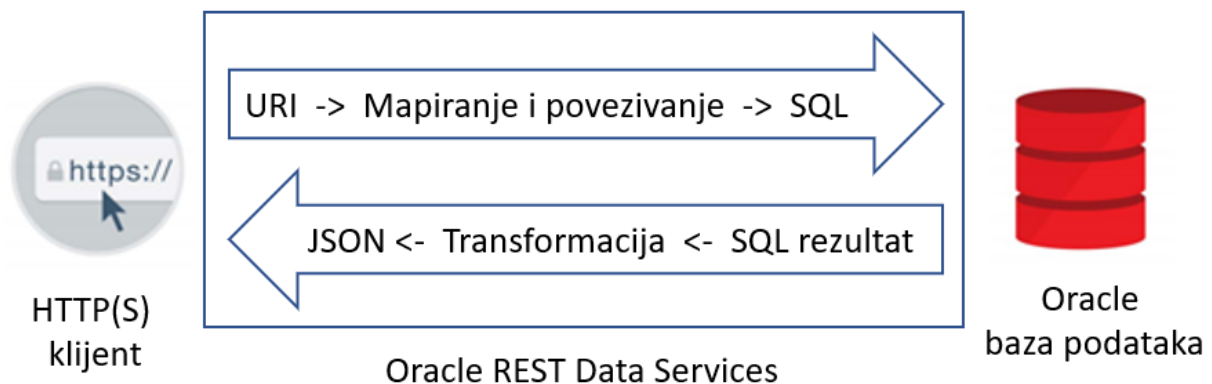
³³ Oracle Cloud Infrastructure Documentation, <https://docs.cloud.oracle.com/en-us/iaas/Content/Database/Concepts/adboverview.htm> [16.svibnja.2020]

Za ove zadatke prije je bilo potrebno angažirati administratore baze podataka.

Srednji sloj

U srednjem sloju nalazi se HTTP poslužitelj koji prihvaća zahtjeve klijenta i prosljeđuje ih ORDS poslužitelju. ORDS je java aplikacija koja temeljem URI-ja, HTTP metode, zaglavlja zahtjeva, tijela zahtjeva prevodi REST zahtjeve u pozive baze podataka. Odgovore iz baze podataka, odnosno rezultate obrade pretvara u odgovore klijentu. U tijelo odgovora stavlja podatke u JSON formatu, dodaje zaglavlje odgovora prije nego što će odgovor biti poslani natrag klijentu. ORDS omogućuju programerima stvaranje REST sučelja za Oracle bazu podataka.³⁴ Programeri mogu koristiti funkcionalnosti AutoREST za tablice i pogleda odnosno mogu koristiti Oracle programske module koje koriste SQL i PL/SQL rutine za implementaciju složenije poslove logike. AutoREST funkcionalnost korištena je u ovom radu.

Slika 16. Oracle REST Data Service



Izvor: Izrada autora prema Oracle Spatial Summit at BIWA (2017) RESTful Spatial services with ORDS [online]. Dostupno na https://download.oracle.com/otndocs/products/spatial/pdf/biwa2017/Biwa2017_RESTful_Spatial_Services_Lapp_842619.pdf [20.veljače.2020]

Klijent

Klijentsku stranu čini aplikacija koja je dio informacijskog sustava kupca. U toj aplikaciji implementiran je REST klijent. REST klijenta pomoću HTTP(S) protokola komunicira sa REST API-jem na strani poslužitelja. Za razvoj REST klijenta postoji puno biblioteka za različite programske jezike i platforme koje implementiraju metode za komunikaciju sa poslužiteljem. Primjer takve biblioteke za java programski jezik je JAX-RS Client API. Za testiranje REST API-ja u ovom radu korišten je alat Postman.

³⁴ Oracle (2019) Oracle REST Data Services Installation, Configuration, and Development Guide [online]. Dostupno na: <https://docs.oracle.com/en/database/oracle/oracle-rest-data-services/19.2/aelig/oracle-rest-data-services-installation-configuration-and-development-guide.pdf> [20.veljače.2020], str. 3-52

6.3. Razvojni alati

Oracle SQL Developer

U ovom radu osnovni razvojni alat je Oracle SQL Developer. To je grafički alat koji razvojnim programerima i administratorima baza podataka omogućava obavljanje osnovnih zadataka.³⁵ Pomoću SQL Developera mogu se pregledavati, stvarati, uređivati i brisati podaci, pokretati SQL naredbe i skripte, pisati PL/SQL kod i ispravljati greške u programskom kodu. Može se raditi uvoz i izvoz podataka, te pregledavati i stvarati izvješća. SQL Developer može se povezati s Oracle bazom podataka kao i sa drugom bazama kao što su: MySQL, Microsoft SQL Server, Sybase Adaptive Server, IBM DB2. Pomoću SQL Developer-a mogu se migrirati podaci iz drugih baza u Oracle bazu. U SQL Developer integriran je alat SQL Developer Data Modeler. Oracle SQL Developer Data Modeler također je besplatan grafički alat koji podiže produktivnost i pojednostavljuje zadatke za modeliranje podataka. Koristeći Data Modeler korisnici mogu kreirati, pregledavati i uređivati, logičke, relacijske, fizičke, višedimenzionalne modele i vrste podataka. Data Modeler pruža mogućnost reverznog inženjeringa, podržava kolaborativni razvoj putem integrirane kontrole izvornog koda. Oracle SQL Developer može se koristiti u tradicionalnom okruženju (engl. on-premise) kao i u okruženju u oblaku.

Postman

Postman je HTTP klijent za testiranje web servisa. Osim RESTful servisa Postman podržava testiranje i SOAP servisa. To je proizvod tvrtke Postdot Technologies (Postman, Inc.). Besplatan je za pojedince i male timove. Postoje i komercijalne verzije Postman Pro i Postman Enterprise. Postman omogućuje formiranje složenih HTTP zahtjeva. Moguć je odabir HTTP metoda, postavljanje parametara zaglavlja upita, odabir protokola (HTTP ili HTTPS) i drugih parametara. Postman omogućuje stvaranje kolekcija upita koje ubrzavaju i olakšavaju testiranje. Dio Postmana je i konzola koja pomaže pri rješavanju problema u testiranju. Pomoću konzole moguće je detaljno pratiti sve parametre HTTP zahtjeva i odgovora.

³⁵ Oracle (2019) Oracle SQL Developer User's Guide [online]. Dostupno na: <https://docs.oracle.com/en/database/oracle/sql-developer/19.1/rptug/oracle-sql-developer-users-guide.pdf> [15. veljače.2020], str.1-2

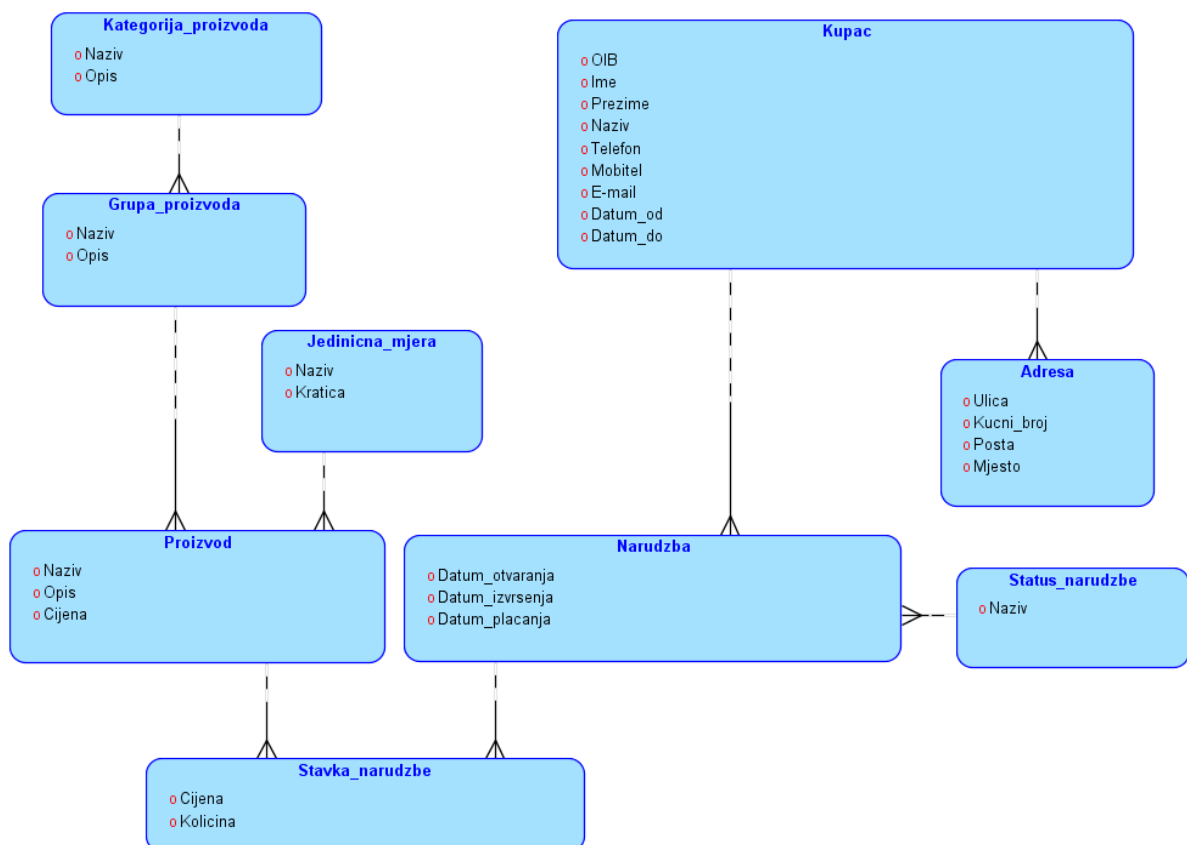
7. PRIMJER REST SERVISIA - B2B APLIKACIJA PRODAJE

B2B aplikacije namijenjene su povezivanju poslovnih partnera odnosno poslovnih sustava između kojih postoji intenzivna poslovna aktivnost. Napravljena je poslužiteljska strana aplikacije kao RESTful servis koji implementira REST API. U bazi podataka implementiran je model podataka i minimalna poslovna logika. Objekti baze podataka tablice i pogledi osnove su za resurse aplikacije. REST API izveden je pomoću Oracle Rest Data Services. Klijentsku stranu aplikacije izrađivat će poslovni partneri odnosno kupci ugrađivanjem REST API poziva i analizom odgovora odnosno analizom reprezentacija resursa.

7.1. Modeliranje podataka

Alat SQL Developer Data Modeler korišten je za konceptualno i logičko oblikovanje. Iz logičkog modela generirane su SQL DDL skripte za implementaciju u bazi podataka. Rezultat konceptualnog modeliranja je ER dijagram prikazan na slici 17.

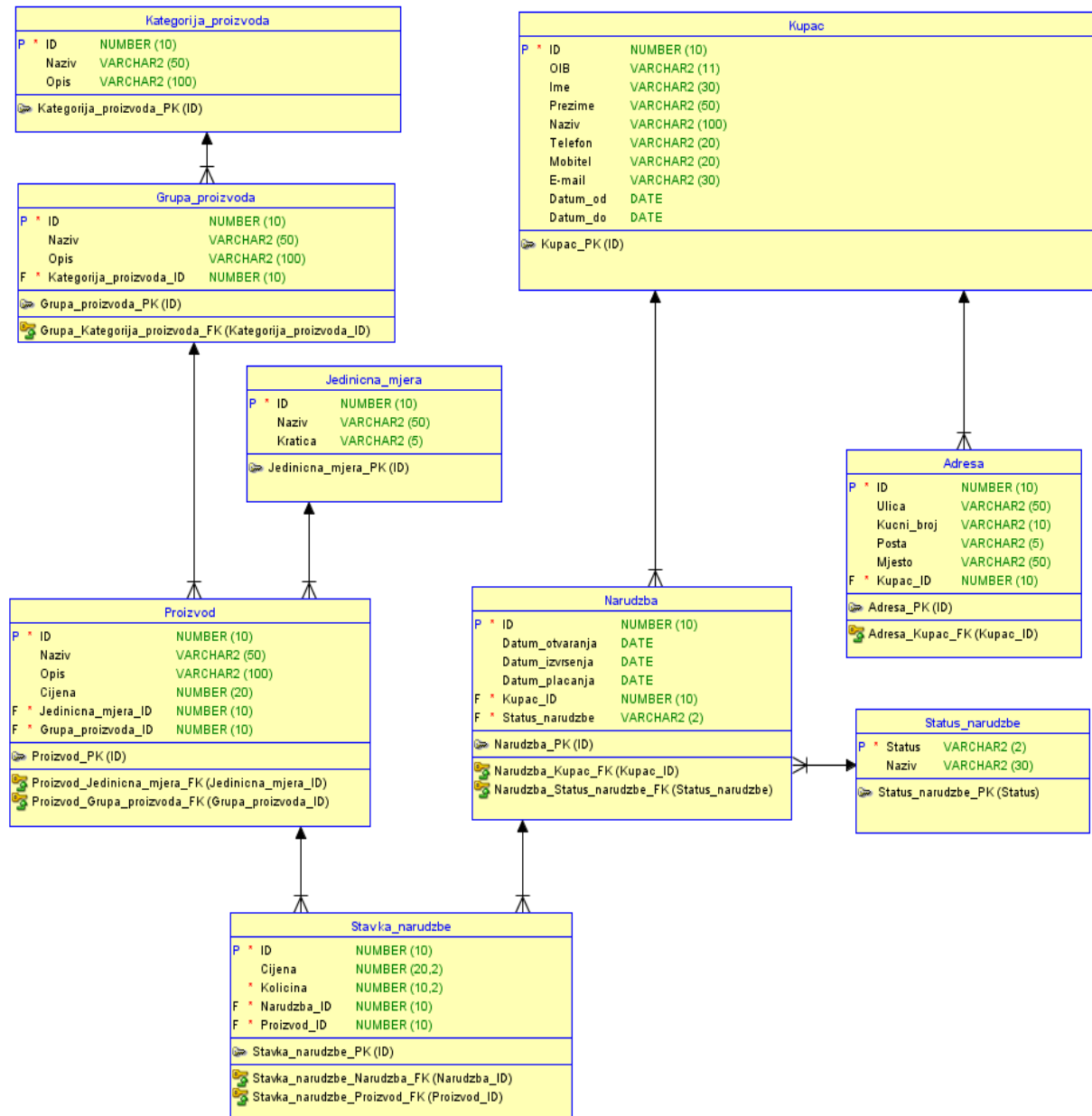
Slika 17. ER dijagram B2B aplikacije



Izvor: Izrada autora

Logičko ili implementacijsko modeliranje polazi od ER modela , a rezultira izrađenim logičkim ili implementacijskim modelom.³⁶

Slika 18. Relacijska shema B2B aplikacije



Izvor: Izrada autora

Na slici 18. Prikazana je relacijska shema , a u nastavku slijedi prikaz tablica sa poljima, primarnim i stranim ključevima.

KATEGORIJA_PROIZVODA (**ID**, Naziv, Opis)

³⁶ Varga, M. (1994) Baze podataka Konceptualno, logičko i fizičko modeliranje podataka. Zagreb:DRIP. str. 91

GRUPA_PROIZVODA (**ID**, Naziv, Opis, Kategorija_proizvoda_ID)

JEDINICNA_MJERA (**ID**, Naziv, Kratica)

PROIZVOD (**ID**, Naziv, Opis, Cijena, Jedinicna_mjera_ID, Grupa_proizvoda_ID)

KUPAC (**ID**, OIB, Ime, Prezime, Naziv, Telefon, Mobitel, Email, Datum_od, Datum_do)

ADRESA (**ID**, Ulica, Kucni_broj, Posta, Mjesto, Kupac_ID)

STATUS_NARUDZBE (**Status**, Naziv)

NARUDZBA (**ID**, Datum_otvaranja, Datum_izvršenja, Datum_plaćanja, Kupac_ID,
Status_narudzbe)

STAVKA_NARUDZBE (**ID**, Cijena, Kolicina, Narudzba_ID, Proizvod_ID)

7.2. Implementacija modela u relacijsku bazu podataka

U bazi podataka kreirane su tablice (kategorija_proizvoda, grupa_proizvoda, jedinicna_mjera_proizvod, kupac, adresa, narudzba, status_narudzbe, stavke_narudzbe). Svaku tablicu identificira primarni ključ, a povezane su stranim ključevima.

kategorija_proizvoda

Implementacija tablice kategorija_proizvoda odnosno fizičko oblikovanje u bazi podataka provodi se naredbom CREATE TABLE. Primarni ključ tablice je kolona ID. Za popunjavanje vrijednosti kolone ID kreira se sekvenca kategorija_proizvoda_id_seq. Okidač kategorija_proizvoda_bri_trg se izvodi prije umetanja sloga u tablicu i služi za automatsko popunjavanje kolone ID sekvencom. DDL naredbe prikazane su na slici 19.

Slika 19. DDL kategorija proizvoda

```
CREATE TABLE kategorija_proizvoda (
  id      NUMBER(10) NOT NULL,
  naziv   VARCHAR2(50),
  opis    VARCHAR2(100)
);

ALTER TABLE kategorija_proizvoda ADD CONSTRAINT kategorija_proizvoda_pk PRIMARY KEY
( id );

CREATE SEQUENCE kategorija_proizvoda_id_seq START WITH 1 NOCACHE ORDER;

create or replace TRIGGER kategorija_proizvoda_bri_trg BEFORE
  INSERT ON kategorija_proizvoda
  FOR EACH ROW
  WHEN ( new.id IS NULL )
BEGIN
  :new.id := kategorija_proizvoda_id_seq.nextval;
END;
```

Izvor: Izrada autora

grupa_proizvoda

Implementacija tablice grupa_proizvoda odnosno fizičko oblikovanje u bazi podataka provodi se naredbom CREATE TABLE. Primarni ključ tablice je kolona ID. Za popunjavanje vrijednosti kolone ID kreira se sekvenca grupa_proizvoda_id_seq. Strani ključ tablice je kategorija_proizvoda_id. Okidač grupa_proizvoda_bri_trg se izvodi prije umetanja sloga u tablicu i služi za automatsko popunjavanje kolone ID sekvencom. DDL naredbe prikazane su na slici 20.

Slika 20. DDL grupa proizvoda

```
CREATE TABLE grupa_proizvoda (
  id          NUMBER(10) NOT NULL,
  naziv       VARCHAR2(50),
  opis        VARCHAR2(100),
  kategorija_proizvoda_id  NUMBER(10) NOT NULL
);

ALTER TABLE grupa_proizvoda ADD CONSTRAINT grupa_proizvoda_pk PRIMARY KEY ( id );

ALTER TABLE grupa_proizvoda
  ADD CONSTRAINT grupa_kategorija_proizvoda_fk
  FOREIGN KEY ( kategorija_proizvoda_id )
  REFERENCES kategorija_proizvoda ( id );

CREATE SEQUENCE grupa_proizvoda_id_seq START WITH 1 NOCACHE ORDER;

create or replace TRIGGER grupa_proizvoda_bri_trg BEFORE
  INSERT ON grupa_proizvoda
  FOR EACH ROW
  WHEN ( new.id IS NULL )
BEGIN
  :new.id := grupa_proizvoda_id_seq.nextval;
END;
```

Izvor: Izrada autora

jedinicna_mjera

Implementacija tablice jedinicna_mjera odnosno fizičko oblikovanje u bazi podataka provodi se naredbom CREATE TABLE. Primarni ključ tablice je kolona ID. Za popunjavanje vrijednosti kolone ID kreira se sekvenca jedinicna_mjera_id_seq. Okidač jedinicna_mjera_bri_trg se izvodi prije umetanja sloga u tablicu i služi za automatsko popunjavanje kolone ID sekvencom. DDL naredbe prikazane su na slici 21.

Slika 21. DDL jedinična mjera

```
CREATE TABLE jedinicna_mjera (
  id          NUMBER(10) NOT NULL,
  naziv       VARCHAR2(50),
  kratica     VARCHAR2(5)
);

ALTER TABLE jedinicna_mjera ADD CONSTRAINT jedinicna_mjera_pk PRIMARY KEY ( id );

CREATE SEQUENCE jedinicna_mjera_id_seq START WITH 1 NOCACHE ORDER;
```

```

create or replace TRIGGER "JEDINICNA_MJERA_BRI_TRG" BEFORE
  INSERT ON jedinicna_mjera
  FOR EACH ROW
  WHEN( new.id IS NULL ) BEGIN
    :new.id := jedinicna_mjera_id_seq.nextval;
  END;

```

Izvor: Izrada autora

proizvod

Implementacija tablice proizvod odnosno fizičko oblikovanje u bazi podataka provodi se naredbom CREATE TABLE. Primarni ključ tablice je kolona ID. Za popunjavanje vrijednosti kolone ID kreira se sekvenca proizvod_id_seq. Strani ključevi tablice su. grupa_proizvoda_id, jedinicna_mjera. Okidač proizvod_bri_trg se izvodi prije umetanja sloga u tablicu i služi za automatsko popunjavanje kolone ID sekvencom. DDL naredbe prikazane su na slici 22.

Slika 22. DDL proizvod

```

CREATE TABLE proizvod (
  id          NUMBER(10) NOT NULL,
  naziv       VARCHAR2(50),
  opis        VARCHAR2(100),
  cijena      NUMBER(20),
  jedinicna_mjera_id NUMBER(10) NOT NULL,
  grupa_proizvoda_id NUMBER(10) NOT NULL
);

ALTER TABLE proizvod ADD CONSTRAINT proizvod_pk PRIMARY KEY ( id );

ALTER TABLE proizvod
  ADD CONSTRAINT proizvod_grupa_proizvoda_fk
  FOREIGN KEY ( grupa_proizvoda_id )
  REFERENCES grupa_proizvoda ( id );

ALTER TABLE proizvod
  ADD CONSTRAINT proizvod_jedinicna_mjera_fk
  FOREIGN KEY ( jedinicna_mjera_id )
  REFERENCES jedinicna_mjera ( id );

CREATE SEQUENCE proizvod_id_seq START WITH 1 NOCACHE ORDER;

create or replace TRIGGER proizvod_bri_trg BEFORE
  INSERT ON proizvod
  FOR EACH ROW
  WHEN ( new.id IS NULL )
  BEGIN
    :new.id := proizvod_id_seq.nextval;
  END;

```

Izvor: Izrada autora

kupac

Implementacija tablice kupac odnosno fizičko oblikovanje u bazi podataka provodi se naredbom CREATE TABLE. Primarni ključ tablice je kolona ID. Za popunjavanje vrijednosti kolone ID kreira se sekvenca kupac_id_seq. Okidač kupac_bri_trg se izvodi prije umetanja sloga u tablicu i služi za automatsko popunjavanje kolone ID sekvencom. DDL naredbe prikazane su na slici 23.

Slika 23. DDL kupac

```
CREATE TABLE kupac (
  id          NUMBER(10) NOT NULL,
  oib         VARCHAR2(11),
  ime         VARCHAR2(30),
  prezime     VARCHAR2(50),
  naziv       VARCHAR2(100),
  telefon     VARCHAR2(20),
  mobitel     VARCHAR2(20),
  "E-mail"    VARCHAR2(30),
  datum_od   DATE,
  datum_do   DATE
);

ALTER TABLE kupac ADD CONSTRAINT kupac_pk PRIMARY KEY ( id );

CREATE SEQUENCE kupac_id_seq START WITH 1 NOCACHE ORDER;

create or replace TRIGGER kupac_bri_trg BEFORE
  INSERT ON kupac
  FOR EACH ROW
  WHEN ( new.id IS NULL )
BEGIN
  :new.id := kupac_id_seq.nextval;
  :new.datum_od := trunc(sysdate);
END;
```

Izvor: Izrada autora

adresa

Implementacija tablice *adresa* odnosno fizičko oblikovanje u bazi podataka provodi se naredbom `CREATE TABLE`. Primarni ključ tablice je kolona `ID`. Za popunjavanje vrijednosti kolone `ID` kreira se sekvenca `adresa_id_seq`. Strani ključ tablice je `kupac_id`. Okidač `adresa_bri_trg` se izvodi prije umetanja sloga u tablicu i služi za automatsko popunjavanje kolone `ID` sekvencom. DDL naredbe prikazane su na slici 24.

Slika 24. DDL adresa

```
CREATE TABLE adresa (
  id          NUMBER(10) NOT NULL,
  ulica       VARCHAR2(50),
  kucni_broj  VARCHAR2(10),
  posta       VARCHAR2(5),
  mjesto      VARCHAR2(50),
  kupac_id    NUMBER(10) NOT NULL
);

ALTER TABLE adresa ADD CONSTRAINT adresa_pk PRIMARY KEY ( id );

ALTER TABLE adresa
  ADD CONSTRAINT adresa_kupac_fk FOREIGN KEY ( kupac_id )
  REFERENCES kupac ( id );

CREATE SEQUENCE adresa_id_seq START WITH 1 NOCACHE ORDER;
```



```

create or replace TRIGGER
adresa_bri_trg BEFORE
  INSERT ON adresa
  FOR EACH ROW
  WHEN ( new.id IS NULL )
BEGIN
  :new.id := adresa_id_seq.nextval;
END;

```

Izvor: izrada autora

status_narudzbe

Implementacija tablice status_narudzbe odnosno fizičko oblikovanje u bazi podataka provodi se naredbom CREATE TABLE. Primarni ključ tablice je kolona status. DDL naredbe prikazane su na slici 25.

Slika 25. DDL status narudžbe

```

CREATE TABLE status_narudzbe (
  status VARCHAR2(2) NOT NULL,
  naziv VARCHAR2(30)
);

ALTER TABLE status_narudzbe ADD CONSTRAINT status_narudzbe_pk PRIMARY KEY ( status );

```

Izvor: Izrada autora

narudzba

Implementacija tablice narudzba odnosno fizičko oblikovanje u bazi podataka provodi se naredbom CREATE TABLE. Primarni ključ tablice je kolona ID. Za popunjavanje vrijednosti kolone ID kreira se sekvenca narudzba_id_seq. Strani ključevi tablice su: kupac_id, status_narudzbe. Okidač narudzba_bri_trg se izvodi prije umetanja sloga u tablicu i služi za automatsko popunjavanje kolone ID sekvencom. DDL naredbe prikazane su na slici 26.

Slika 26. DDL narudžba

```

CREATE TABLE narudzba (
  id NUMBER(10) NOT NULL,
  datum_otvaranja DATE,
  datum_izvršenja DATE,
  datum_plaćanja DATE,
  kupac_id NUMBER(10) NOT NULL,
  status_narudzbe VARCHAR2(2) NOT NULL
);

ALTER TABLE narudzba ADD CONSTRAINT narudzba_pk PRIMARY KEY ( id );

ALTER TABLE narudzba
  ADD CONSTRAINT narudzba_kupac_fk FOREIGN KEY ( kupac_id )
  REFERENCES kupac ( id );

ALTER TABLE narudzba
  ADD CONSTRAINT narudzba_status_narudzbe_fk
  FOREIGN KEY ( status_narudzbe )
  REFERENCES status_narudzbe ( status );

```

```

CREATE SEQUENCE narudzba_id_seq START WITH 1 NOCACHE ORDER;

create or replace TRIGGER narudzba_bri_trg BEFORE
  INSERT ON narudzba
  FOR EACH ROW
  WHEN ( new.id IS NULL )
BEGIN
  :new.id := narudzba_id_seq.nextval;
  :new.datum_otvaranja := trunc(sysdate);
  :new.status_narudzbe := '10';
END;

```

Izvor: Izrada autora

stavka_narudzbe

Implementacija tablice stavka_narudzbe odnosno fizičko oblikovanje u bazi podataka provodi se naredbom CREATE TABLE. Primarni ključ tablice je kolona ID. Za popunjavanje vrijednosti kolone ID kreira se sekvenca stavka_narudzbe_id_seq. Strani ključevi tablice su: narudzba_id, proizvod_id. Okidač stavke_narudzbe_bri_trg se izvodi prije umetanja sloga u tablicu i služi za automatsko popunjavanje kolone ID sekvencom. DDL naredbe prikazane su na slici 27. U okidaču osim popunjavanja kolone sekvencom dodan je i dio poslovne logike koja računa cijenu u stavci narudžbe.

Slika 27. DDL stavka narudžbe

```

CREATE TABLE stavka_narudzbe (
  id          NUMBER(10) NOT NULL,
  cijena      NUMBER(20, 2),
  kolicina    NUMBER(10, 2) NOT NULL,
  narudzba_id NUMBER(10) NOT NULL,
  proizvod_id NUMBER(10) NOT NULL
);

ALTER TABLE stavka_narudzbe ADD CONSTRAINT stavka_narudzbe_pk PRIMARY KEY ( id );

ALTER TABLE stavka_narudzbe
  ADD CONSTRAINT stavka_narudzbe_narudzba_fk FOREIGN KEY ( narudzba_id )
  REFERENCES narudzba ( id );

ALTER TABLE stavka_narudzbe
  ADD CONSTRAINT stavka_narudzbe_proizvod_fk FOREIGN KEY ( proizvod_id )
  REFERENCES proizvod ( id );

CREATE SEQUENCE stavka_narudzbe_id_seq START WITH 1 NOCACHE ORDER;

create or replace TRIGGER stavka_narudzbe_bri_trg BEFORE
  INSERT ON stavka_narudzbe
  FOR EACH ROW
  WHEN ( new.id IS NULL )
declare
v_jed_cijena number(20,2);
BEGIN
  :new.id := stavka_narudzbe_id_seq.nextval;
  select cijena into v_jed_cijena from proizvod where id=:new.proizvod_id;
  :new.cijena := :new.kolicina * v_jed_cijena;
END;

```

```

create or replace TRIGGER STAVKE_NARUDZBE_BRU_TRG
BEFORE UPDATE ON stavka_narudzbe
FOR EACH ROW
declare
v_jed_cijena number(20,2);
BEGIN
select cijena into v_jed_cijena from proizvod where id=:new.proizvod_id;
:new.cijena := :new.kolicina * v_jed_cijena;
END;

```

Izvor: izrada autora

7.3. Oracle Rest Data Services

U ovom radu resursi Oracle Rest Data Services (ORDS) su kreirani pomoću AutoREST-a. AutoREST je funkcionalnost ORDS-a koja omogućuje automatsko kreiranje resursa na tablicama i pogledima. PL/SQL paket ORDS sadrži funkcije i procedure za razvoj RESTful usluga pomoću ORDS-a. Procedura ENABLE_OBJECT omogućuje ili onemogućuje ORDS na pojedinoj tablici ili pogledu. Sintaksa procedure prikazana je na slici 28.

Slika 28. Procedura ORDS.ENABLE_OBJECT

```

ORDS.ENABLE_OBJECT (
    p_enabled IN boolean DEFAULT TRUE,
    p_schema IN ords_schemas.parsing_schema%type DEFAULT NULL,
    p_object IN ords_objects.parsing_object%type,
    p_object_type IN ords_objects.type%type DEFAULT 'TABLE',
    p_object_alias IN ords_objects.object_alias%type DEFAULT NULL,
    p_auto_rest_auth IN boolean DEFAULT NULL
);

```

Izvor: Preuzeto iz Oracle (2019) Oracle REST Data Services Installation, Configuration, and DevelopmentGuide [online]. Dostupno na: <https://docs.oracle.com/en/database/oracle/oracle-rest-data-services/19.2/aelig/oracle-rest-data-services-installation-configuration-and-development-guide.pdf> [20.veljače.2020]

Parametri:

- p_enabled - TRUE dozvoli pristup objektu, FALSE zabrani pristup objektu
- p_schema – naziv sheme baze podataka gdje se nalazi tablica ili pogled
- p_object – naziv objekta (tablice ili pogleda)
- p_object_type – (tip objekta TABLE / VIEW)
- p_object_alias – alias za objekt, odnosno naziv resursa
- p_auto_rest_auth – parametar koji određuje da li je potrebna autorizacija korisnika kod korištenja resursa

Slika 29 prikazuje poziv procedure ORDS.enable_object kojom tablica KUPAC postaje REST resurs sa URI <https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/kupci/>

Slika 29. Kreiranje resursa kupci

```
BEGIN
  ORDS.enable_object(
    p_enabled      => TRUE,
    p_schema       => 'PRODAJA',
    p_object       => 'KUPAC',
    p_object_type  => 'TABLE',
    p_object_alias => 'kupci'
  );
  COMMIT;
END;
```

Izvor: Izrada autora

Slika 30 prikazuje poziv procedure ORDS.enable_object kojom tablica ADRESA postaje REST resurs sa URI <https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/adrese/>

Slika 30. Kreiranje resursa adrese

```
BEGIN
  ORDS.enable_object(
    p_enabled      => TRUE,
    p_schema       => 'PRODAJA',
    p_object       => 'ADRESA',
    p_object_type  => 'TABLE',
    p_object_alias => 'adrese'
  );
  COMMIT;
END;
```

Izvor: Izrada autora

Slika 31 prikazuje poziv procedure ORDS.enable_object kojom tablica NARUDZBA postaje REST resurs sa URI <https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/narudzbe/>

Slika 31. Kreiranje resursa narudzbe

```
BEGIN
  ORDS.enable_object(
    p_enabled      => TRUE,
    p_schema       => 'PRODAJA',
    p_object       => 'NARUDZBA',
    p_object_type  => 'TABLE',
    p_object_alias => 'narudzbe'
  );
  COMMIT;
END;
```

Izvor: Izrada autora

Slika 32 prikazuje poziv procedure ORDS.enable_object kojom tablica STAVKA_NARUDZBE postaje REST resurs sa URI <https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/stavkenarudzbe/>

Slika 32. Kreiranje resursa stavkenarudzbe

```
BEGIN
  ORDS.enable_object(
    p_enabled      => TRUE,
    p_schema       => 'PRODAJA',
    p_object       => 'STAVKA_NARUDZBE',
    p_object_type  => 'TABLE',
    p_object_alias => 'stavkenarudzbe'
  );
  COMMIT;
END;
```

Izvor: Izrada autora

Slika 33 prikazuje poziv procedure ORDS.enable_object kojom pogled V_PROIZVOD postaje REST resurs sa URI <https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/vproizvodi/>

Slika 33. Kreiranje resursa vproizvodi

```
BEGIN
  ORDS.enable_object(
    p_enabled      => TRUE,
    p_schema       => 'PRODAJA',
    p_object       => 'V_PROIZVOD',
    p_object_type  => 'VIEW',
    p_object_alias => 'vproizvodi'
  );
  COMMIT;
END;
```

Izvor: Izrada autora

Slika 34 prikazuje poziv procedure ORDS.enable_object kojom pogled V_STAVKA_NARUDZBE postaje REST resurs sa URI <https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/vstavkenarudzbe/>

Slika 34. Kreiranje resursa vstavkenarudzbe

```
BEGIN
  ORDS.enable_object(
    p_enabled      => TRUE,
    p_schema       => 'PRODAJA',
    p_object       => 'V_STAVKA_NARUDZBE',
    p_object_type  => 'VIEW',
    p_object_alias => 'vstavkenarudzbe'
  );
  COMMIT;
END;
```

Izvor: Izrada autora

Katalog metapodataka (engl. Metadata catalog) predstavlja podatke o podacima, odnosno podatke koji opisuju model podataka. Nakon što su kreirani svi resursi za aplikaciju prodaja pozivom <https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/metadata-catalog/> možemo prikazati odnosno katalog metapodataka odnosno opis svih resursa kojima se može upravljati pomoću REST API-ja. Rezultat je prikazan na slici 35.

Slika 35. Katalog metapodataka

```
{
  "items": [
    {
      "name": "ADRESA",
      "links": [
        {
          "rel": "describes",
          "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/adrese/"
        },
        {
          "rel": "canonical",
          "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/metadata-catalog/adrese/",
          "mediaType": "application/json"
        },
        {
          "rel": "alternate",
          "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/open-api-catalog/adrese/",
          "mediaType": "application/openapi+json"
        }
      ]
    },
    {
      "name": "KUPAC",
      "links": [
        {
          "rel": "describes",
          "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/kupci/"
        },
        {
          "rel": "canonical",
          "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/metadata-catalog/kupci/",
          "mediaType": "application/json"
        },
        {
          "rel": "alternate",
          "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/open-api-catalog/kupci/",
          "mediaType": "application/openapi+json"
        }
      ]
    }
  ],
}
```

```

{
  "name": "NARUDZBA",
  "links": [
    {
      "rel": "describes",
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/narudzbe/"
    },
    {
      "rel": "canonical",
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/metadata-catalog/narudzbe/",
      "mediaType": "application/json"
    },
    {
      "rel": "alternate",
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/open-api-catalog/narudzbe/",
      "mediaType": "application/openapi+json"
    }
  ]
},
{
  "name": "STAVKA_NARUDZBE",
  "links": [
    {
      "rel": "describes",
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/stavkenarudzbe/"
    },
    {
      "rel": "canonical",
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/metadata-catalog/stavkenarudzbe/",
      "mediaType": "application/json"
    },
    {
      "rel": "alternate",
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/open-api-catalog/stavkenarudzbe/",
      "mediaType": "application/openapi+json"
    }
  ]
},
{
  "name": "V_PROIZVOD",
  "links": [
    {
      "rel": "describes",
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/vproizvodi/"
    },
    {
      "rel": "canonical",
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/metadata-catalog/vproizvodi/",
      "mediaType": "application/json"
    }
  ]
},

```

```

        {
          "rel": "alternate",
          "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/open-api-catalog/vproizvodi/",
          "mediaType": "application/openapi+json"
        }
      ]
    },
    {
      "name": "V_STAVKA_NARUDZBE",
      "links": [
        {
          "rel": "describes",
          "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/vstavkenarudzbe/"
        },
        {
          "rel": "canonical",
          "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/metadata-catalog/vstavkenarudzbe/",
          "mediaType": "application/json"
        },
        {
          "rel": "alternate",
          "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/open-api-catalog/vstavkenarudzbe/",
          "mediaType": "application/openapi+json"
        }
      ]
    }
  ],
  "hasMore": false,
  "limit": 25,
  "offset": 0,
  "count": 6,
  "links": [
    {
      "rel": "self",
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/metadata-catalog/"
    },
    {
      "rel": "first",
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/metadata-catalog/"
    }
  ]
}

```

Izvor: Izlaz B2B aplikacije prodaje

Za svaki resurs mogu se vidjeti metapodaci. Metapodaci daju specifikacije resursa koje su potrebne programerima na klijentskoj strani. Pozivom na URI <https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/metadata-catalog/stavkenarudzbe> prikazuju su metapodaci za resurs stavke narudžbe na slici 36.

Slika 36. Metapodaci resursa stavkenarudzbe

```
{
  "name": "STAVKA_NARUDZBE",
  "primarykey": [
    "id"
  ],
  "members": [
    {
      "name": "id",
      "type": "NUMBER"
    },
    {
      "name": "cijena",
      "type": "NUMBER"
    },
    {
      "name": "kolicina",
      "type": "NUMBER"
    },
    {
      "name": "narudzba_id",
      "type": "NUMBER"
    },
    {
      "name": "proizvod_id",
      "type": "NUMBER"
    }
  ],
  "links": [
    {
      "rel": "collection",
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.
        oraclecloudapps.com/ords/prodaja/metadata-catalog/",
      "mediaType": "application/json"
    },
    {
      "rel": "canonical",
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.
        oraclecloudapps.com/ords/prodaja/metadata-catalog/stavkenarudzbe/",
      "mediaType": "application/json"
    },
    {
      "rel": "alternate",
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.
        oraclecloudapps.com/ords/prodaja/open-api-catalog/stavkenarudzbe/",
      "mediaType": "application/openapi+json"
    },
    {
      "rel": "describes",
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.
        oraclecloudapps.com/ords/prodaja/stavkenarudzbe/"
    }
  ]
}
```

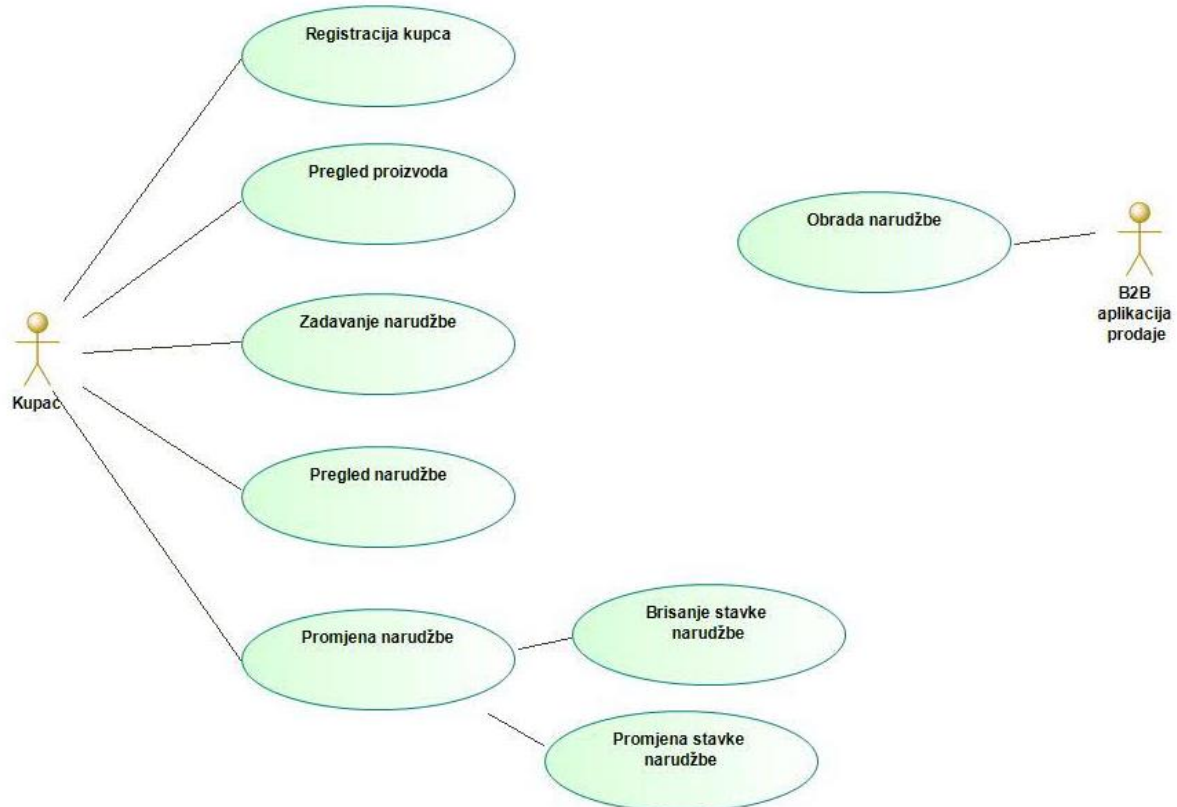
Izvor: Izlaz B2B aplikacije prodaje

7.4. Slučajevi korištenja

Slučajevi korištenja predstavljaju funkcionalnost sustava odnosno zahtjeve koji na sustav postavljaju korisnici. To je pogled na funkcionalnosti sustava iz perspektive korisnika.

Slika 37. Dijagram slučajeve korištenja

Izvor: Izrada autora



7.4.1. Registracija kupaca

Registracija kupaca dodaje podatke u tablice kupac i adresa. Dodavanje resursa kupci radi se pozivom REST API-ja odnosno slanjem zahtjeva na URI <https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/kupci/>. Metoda zahtjeva je POST. U tijelo zahtjeva stavljaju se podaci o kupcu u JSON formatu slika 38.

Slika 38. Dodavanje kupca (tijelo zahtjeva)

```
{
  "oib": "02345678911",
  "naziv": "Tvrtka dva d.o.o.",
  "telefon": "01/539612",
  "mobitel": "095/452186",
  "e-mail": "tvrtka.two@gmail.com",
}
```

Izvor: Izrada autora

Kada poslužitelj obradi zahtjev odgovara sa HTTP kodom 201 (engl. Created). Tijelo odgovora prikazano je slici 39.

Slika 39. Dodavanje kupca (tijelo odgovora)

```
{
  "id": 10,
  "oib": "02345678911",
  "ime": null,
  "prezime": null,
  "naziv": "Tvrtka dva d.o.o.",
  "telefon": "01/539612",
  "mobitel": "095/452186",
  "e-mail": "tvrtka.two@gmail.com",
  "datum_od": "2020-06-11T00:00:00Z",
  "datum_do": null,
  "links": [
    {
      "rel": "self",
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.
        oraclecloudapps.com/ords/prodaja/kupci/10"
    },
    {
      "rel": "edit",
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.
        oraclecloudapps.com/ords/prodaja/kupci/10"
    },
    {
      "rel": "describedby",
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.
        oraclecloudapps.com/ords/prodaja/metadata-catalog/kupci/item"
    },
    {
      "rel": "collection",
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.
        oraclecloudapps.com/ords/prodaja/kupci/"
    }
  ]
}
```

Izvor: Izlaz B2B aplikacije prodaje

SQL naredba kojom se dodaju podaci u tablicu kupac prikazana je na slici 40.

Slika 40. SQL naredba za dodavanje kupca

```
INSERT INTO kupac (
  oib,
  naziv,
  telefon,
  mobitel,
  "E-mail",
) VALUES (
  '02345678911',
  'Tvrtka dva d.o.o.',
  '01/539612',
  '095/452186',
  'tvrtka.two@gmail.com',
);
```

Izvor: Izrada autora

Adresa

Dodavanje resursa adrese radi se pozivom REST API-ja odnosno slanjem zahtjeva na URI <https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/adrese/>. Metoda zahtjeva je POST. U tijelo zahtjeva stavljaju se podaci o adresi u JSON formatu slika 41.

Slika 41. Dodavanje adrese (tijelo zahtjeva)

```
{
  "ulica": "Savska",
  "kucni_broj": "10",
  "posta": "10000",
  "mjesto": "Zagreb",
  "kupac_id": 10
}
```

Izvor: Izrada autora

Kada poslužitelj obradi zahtjev odgovara sa HTTP kodom 201 (engl. Created). Tijelo odgovora prikazano je slici 42.

Slika 42. Dodavanje adrese (tijelo odgovora)

```
{
  "id": 3,
  "ulica": "Savska",
  "kucni_broj": "10",
  "posta": "10000",
  "mjesto": "Zagreb",
  "kupac_id": 10,
  "links": [
    {
      "rel": "self",
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/adrese/3"
    },
    {
      "rel": "edit",
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/adrese/3"
    },
    {
      "rel": "describedby",
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/metadata-catalog/adrese/item"
    },
    {
      "rel": "collection",
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/adrese/"
    }
  ]
}
```

Izvor: Izlaz B2B aplikacije prodaje

SQL naredba kojom se dodaju podaci u tablicu adresa prikazana je na slici 43.

Slika 43. SQL naredba za dodavanje adrese

```
INSERT INTO adresa (  
    ulica,  
    kucni_broj,  
    posta,  
    mjesto,  
    kupac_id  
) VALUES (  
    'Savska',  
    '10',  
    '10000',  
    'Zagreb',  
    10  
);
```

Izvor: Izrada autora

7.4.2. Pregled proizvoda

Pregled resursa vproizvodi radi se pozivom REST API-ja odnosno slanjem zahtjeva na URI [https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/vproizvodi/?q={"grupa_proizvod":"voće"}](https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/vproizvodi/?q={). Ovo je zahtjev sa uvjetom. Zahtjev ima argument q (engl. Query) u pozivu. Metoda zahtjeva je GET. Kod GET metode tijelo zahtjeva ostaje prazno. Ako je zahtjev uspješno obrađen poslužitelj odgovara sa HTTP kodom 200 (engl. OK). U tijelo odgovora poslužitelj stavlja reprezentaciju resursa u JSON formatu slika 44.

Slika 44. Pregled proizvoda uz uvjet (tijelo odgovora)

```
{  
  "items": [  
    {  
      "proizvod_id": 2,  
      "kategorija_proizvod": "voće i povrće",  
      "grupa_proizvod": "voće",  
      "naziv_proizvoda": "banana",  
      "cijena": 9.9,  
      "jed_mjera": "kilogram"  
    },  
    {  
      "proizvod_id": 3,  
      "kategorija_proizvod": "voće i povrće",  
      "grupa_proizvod": "voće",  
      "naziv_proizvoda": "limun",  
      "cijena": 12.9,  
      "jed_mjera": "kilogram"  
    }  
  ],  
  "hasMore": false,  
  "limit": 25,  
  "offset": 0,  
  "count": 2,  
  "links": [  
    {  
      "rel": "self",  
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/vproizvodi/?q=%7B+%22grupa_proizvod%22:%22vo%C4%87e%22%7D"  
    }  
  ],  
}
```

```

{
  "rel": "edit",
  "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/vproizvodi/?q=%7B+%22grupa_proizvod%22:%22vo%C4%87e%22%7D"
},
{
  "rel": "describedby",
  "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/metadata-catalog/vproizvodi/"
},
{
  "rel": "first",
  "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/vproizvodi/?q=%7B+%22grupa_proizvod%22:%22vo%C4%87e%22%7D"
}
]
}

```

Izvor: Izlaz B2B aplikacije prodaje

SQL naredba kojom se dohvaćaju podaci u iz pogleda V_PROIZVOD sa uvjetom prikazana je na slici 45.

Slika 45. SQL naredba za pregled proizvoda

```

select * from V_PROIZVOD where grupa_proizvod='voće';

```

Izvor: Izrada autora

7.4.3. Zadavanje narudžbi

Dodavanje resursa narudžbe radi se pozivom REST API-ja odnosno slanjem zahtjeva na URI <https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/narudzbe/>. Metoda zahtjeva je POST. U tijelo zahtjeva stavljaju se podaci o narudžbi u JSON formatu slika 46.

Slika 46. Dodavanje narudžbe (tijelo zahtjeva)

```

{
  "kupac_id": 10
}

```

Izvor: Izrada autora

Kada poslužitelj obradi zahtjev odgovara sa HTTP kodom 201 (engl. Created). Tijelo odgovora prikazano je slici 47.

Slika 47. Dodavanje narudžbe (tijelo odgovora)

```
{
  "id": 4,
  "datum_otvaranja": "2020-06-11T00:00:00Z",
  "datum_izvršenja": null,
  "datum_plaćanja": null,
  "kupac_id": 10,
  "status_narudzbe": "10",
  "links": [
    {
      "rel": "self",
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/narudzbe/4"
    },
    {
      "rel": "edit",
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/narudzbe/4"
    },
    {
      "rel": "describedby",
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/metadata-catalog/narudzbe/item"
    },
    {
      "rel": "collection",
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/narudzbe/"
    }
  ]
}
```

Izvor: Izlaz B2B aplikacije prodaje

SQL naredba kojom se dodaju podaci u tablicu narudžba prikazana je na slici 48.

Slika 48. SQL naredba za dodavanje narudžbe

```
INSERT INTO narudzba (
  kupac_id
) VALUES (10);
```

Izvor: Izrada autora

Dodavanje stavke narudžbe

Dodavanje resursa stavkenarudzbe radi se pozivom REST API-ja odnosno slanjem zahtjeva na URI <https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/stavkenarudzbe/>. Metoda zahtjeva je POST. U tijelo zahtjeva stavljaju se podaci o stavkama narudžbe u JSON formatu slika 49.

Slika 49. Dodavanje stavke narudžbe (tijelo zahtjeva)

```
{
  "kolicina": 3,
  "narudzba_id": 4,
  "proizvod_id": 2
}
```

Izvor: Izrada autora

Kada poslužitelj obradi zahtjev odgovara sa HTTP kodom 201 (engl. Created). Tijelo odgovora prikazano je slici 50.

Slika 50. Dodavanje stavke narudžbe (tijelo odgovora)

```
{
  "id": 6,
  "cijena": 29.7,
  "kolicina": 3,
  "narudzba_id": 4,
  "proizvod_id": 2,
  "links": [
    {
      "rel": "self",
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.
        oraclecloudapps.com/ords/prodaja/stavkenarudzbe/6"
    },
    {
      "rel": "edit",
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.
        oraclecloudapps.com/ords/prodaja/stavkenarudzbe/6"
    },
    {
      "rel": "describedby",
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.
        oraclecloudapps.com/ords/prodaja/metadata-catalog/stavkenarudzbe/item"
    },
    {
      "rel": "collection",
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.
        oraclecloudapps.com/ords/prodaja/stavkenarudzbe/"
    }
  ]
}
```

Izvor: Izlaz B2B aplikacije prodaje

SQL DML naredba kojom se dodaju podaci u tablicu stavka_narudzbe prikazana je na slici 51.

Slika 51. SQL naredba za dodavanje stavke narudžbe

```
INSERT INTO stavka_narudzbe (
  kolicina,
  narudzba_id,
  proizvod_id
) VALUES (
  3,
  4,
  2
);
```

Izvor: Izrada autora

Promjena stavke narudžbe

Promjena resursa stavkenarudzbe radi se pozivom REST API-ja PUT metodom. U ovom primjeru prikazana je promjena količine stavke narudžbe id=6, slanjem zahtjeva na URI <https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/>

ords/prodaja/stavkenarudzbe/6. U tijelo zahtjeva stavljaju se podaci o stavci narudžbe u JSON formatu slika 52.

Slika 52. Promjena stavke narudžbe (tijelo zahtjeva)

```
{
  "kolicina": 2,
  "narudzba_id": 4,
  "proizvod_id": 2
}
```

Izvor: Izrada autora

Kada poslužitelj obradi zahtjev odgovara sa HTTP kodom 200 (engl. OK). Tijelo odgovora prikazano je slici 53.

Slika 53. Promjena stavke narudžbe (tijelo odgovora)

```
{
  "id": 6,
  "cijena": 19.8,
  "kolicina": 2,
  "narudzba_id": 4,
  "proizvod_id": 2,
  "links": [
    {
      "rel": "self",
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/stavkenarudzbe/6"
    },
    {
      "rel": "edit",
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/stavkenarudzbe/6"
    },
    {
      "rel": "describedby",
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/metadata-catalog/stavkenarudzbe/item"
    },
    {
      "rel": "collection",
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/stavkenarudzbe/"
    }
  ]
}
```

Izvor: Izlaz B2B aplikacije prodaje

Brisanje stavke narudžbe

Brisanje stavke narudžbe radi se pozivom REST API-ja odnosno slanje zahtjeva na URI <https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/stavkenarudzbe/7>. Metoda zahtjeva je DELETE. Ova metoda nema tijelo zahtjeva.

Slika 54. Brisanje stavke narudžbe (tijelo odgovora)

```
{
  "rowsDeleted": 1
}
```

Izvor: Izlaz B2B aplikacije prodaje

Slika 55. SQL naredba za brisanje stavke narudžbe

```
delete from stavka_narudzbe where id=7;
```

Izvor: Izrada autora

7.4.4. Praćenje narudžbe

Izvršenje narudžbe može se pratiti upitom u bazu podataka. U primjeru je prikazan upit u narudžbu id=4. Pomoću GET metode radit će se upiti na resurs `vstavkenarudzbe` slanjem zahtjeva na URI `https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/vstavkenarudzbe/?q={"narudzba_id":"4"}`.

Ovo je zahtjev sa uvjetom. Zahtjev ima argument `q` (engl. Query) kojim se zadaje id narudžbe za upit. Kada poslužitelj obradi zahtjev odgovara sa HTTP kodom 200 (engl. OK). U tijelo odgovora poslužitelj stavlja reprezentaciju resursa u JSON formatu slika 56.

Slika 56. Upit u narudžbe (tijelo odgovora)

```
{
  "items": [
    {
      "stavka_id": 6,
      "narudzba_id": 4,
      "kolicina": 2,
      "jed_mjera": "kilogram",
      "cijena": 19.8,
      "jedinicna_cijena": 9.9,
      "naziv_proizvoda": "banana"
    }
  ],
  "hasMore": false,
  "limit": 25,
  "offset": 0,
  "count": 1,
  "links": [
    {
      "rel": "self",
      "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.oraclecloudapps.com/ords/prodaja/vstavkenarudzbe/?q=%7B+%22narudzba_id%22:%224%22%7D"
    }
  ],
}
```

```
{
  "rel": "edit",
  "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.
oraclecloudapps.com/ords/prodaja/vstavkenarudzbe/
?q=%7B+%22narudzba_id%22:%224%22%7D"
},
{
  "rel": "describedby",
  "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.
oraclecloudapps.com/ords/prodaja/metadata-catalog/vstavkenarudzbe/"
},
{
  "rel": "first",
  "href": "https://qcq4zhemr7pykjg-db202006042032.adb.eu-frankfurt-1.
oraclecloudapps.com/ords/prodaja/vstavkenarudzbe/
?q=%7B+%22narudzba_id%22:%224%22%7D"
}
]
}
```

Izvor: Izlaz B2B aplikacije prodaje

8. ZAKLJUČAK

Ovaj rad obrađuje komunikaciju među različitim tehnologijama odnosno interoperabilnost različitih platformi. U radu je pokazana mogućnost korištenja baza podataka iz različitih aplikacija i platformi odnosno izlaganje baza podataka pomoću web servisa.

U teorijskom dijelu rada prikazani su osnovni elementi rada: baze podataka, web servisi i računarstvo u oblaku. Objasnjena je arhitektura poslužitelja baze podataka Oracle. Opisana je Oracle Cloud usluga računarstva u oblaku tvrtke Oracle Corporation, korištena u ovom radu koja nudi kompletan set usluga u oblaku kao i visoku dostupnost i raspoloživost svojih resursa. Opisane su danas dominantne tehnologije i standardi HTTP, JSON koje se koriste u radu.

Web servis je ključni termin za interoperabilnost različitih softvera. Općenito su opisani web servisi, dok su detaljno su obrađeni RESTful web servisi. Objasnjeni su elementi REST sustava, karakteristike odnosno ograničenja REST arhitekture koje sustav mora zadovoljavati da bi se smatrao RESTful sustavom. Većina projekata koji se danas razvija bazirani su na RESTful web servisima. Po broju podržanih platformi kao i po broju korisnika ovi servisi danas su dominantni prema ostalim web servisima. RESTful web servisi postali su nezamjenjivi u mobilnim aplikacijama, pružatelji usluga u oblaku također ih masovno koriste. Jedan od najvećih zagovornika REST-a su velike softverske kompanije Google, Microsoft, IBM. U Hrvatskoj također postoje REST sustavi u javnoj upotrebi. Iako se masovno koriste jedan od većih problema je sigurnost REST sustava. U radu su obrađene su najčešće ranjivosti sustava, objašnjeni su najčešći sigurnosni mehanizmi za provjeru autentičnosti korisnika.

U praktičnom dijelu rada prikazana je B2B aplikacije prodaje. B2B aplikacija prodaje namijenjena je povezivanju poslovnih partnera odnosno poslovnih sustava između kojih postoji intenzivna poslovna aktivnost. Napravljena je poslužiteljska strana aplikacije kao RESTful servis koji implementira REST API. Klijentsku stranu aplikacije izrađivat će poslovni partneri odnosno kupci ugrađivanjem REST API poziva u svoje aplikacije. Razvoj i implementacija napravljeni su u računalnom oblaku koristeći Oracle Cloud Free Tier (Always Free Services). Razvoj aplikacije proveden je kroz faze konceptualnog modeliranja, logičkog modeliranja, fizičkog modeliranja. Za razvoj je korišten alat Oracle SQL Developer Web. Model podataka implementiran je u relacijskoj bazi Oracle 18c u oblaku. Pomoću ORDS (engl. Oracle REST Data Services) u srednjem sloju aplikacije implementirano je REST sučelje na modelu podataka aplikacije. Upotrebom funkcionalnosti AutoREST tablice i pogledi baze podataka postali su

resursi web servisa. Omogućen je pregled, unos, ažuriranje i brisanje u bazi podataka iz različitih platformi i programskih jezika pomoću REST API-ja.

Ovim radom pokazan je tehnološki napredak u razvoju REST servisa. Danas više nije potrebno ulagati u infrastrukturu, angažirati programere i ostale IT stručnjake sa specifičnim znanjima i vještinama. Razvoj REST servisa sada mogu raditi informatičari koji imaju znanja o bazama podataka i SQL-u. Potvrđena je zrelost tehnologije računarstva u oblaku u segmentima baze podataka, web servisa u srednjem sloju i razvojnih alata. U samo nekoliko klikova mišem moguće je kreirati bazu podataka, web mjesto i ostale potrebne resurse. Pokazano je da se uz minimalna početna ulaganja i resurse mogu brzo razviti web servisi koji su dostupni na Internetu.

LITERATURA

1. Allamaraju S. (2010) RESTful Web Services Cookbook. Sebastopol, California: O'Reilly Media Inc.
2. Bažant A. et al. (2004) Osnovne arhitekture mreža. Zagreb:Element
3. Centar Informacijske Sigurnosti (2012) Ispitivanje sigurnosti web servisa [online]. Dostupno na: <https://www.cis.hr/files/dokumenti/CIS-DOC-2012-02-040.pdf> [21.ožujka.2020]
4. Codecademy (2020) What is REST? [online]. Dostupno na: <https://www.codecademy.com/articles/what-is-rest> [2.ožujka.2020]
5. Digitalni akademski arhivi i repozitoriji (2017) [online]. Dostupno na: <https://dabar.srce.hr/2017-03-07/pohranjen-znacajan-broj-radova-koristeci-dabar-rest-api-programsko-sucelje> [21.svibnja.2020]
6. eVisitor (2019) Web API [online]. Dostupno na: <https://www.evisitor.hr/eVisitorWiki/Javno.Web-API.ashx> [19.svibnja.2020]
7. Fakultet elektrotehnike i računarstva Sveučilište u Zagrebu (2006) HTTP protokol [online]. Dostupno na: http://docbook.rasip.fer.hr/or_smil/10b_HTTP/Example/HTTP%20protokol.pdf [10.ožujka.2020]
8. Fielding R. T. (2000) Architectural Styles and the Design of Network-based Software Architectures [online]. Dostupno na: https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf [20.veljače.2020]
9. Gottschalk K. et al. (2002) Introduction to Webservices architecture [online]. Dostupno na: https://www.researchgate.net/profile/Heather_Kreger/publication/224101831_Introduction_to_Web_services_architecture/links/563a67e008ae337ef2984608/Introduction-to-Web-services-architecture.pdf [22.veljače.2020]
10. Halili F., Ramadani E. (2018) Web Services: A Comparison of Soap and Rest Services [online]. Dostupno na: https://www.researchgate.net/publication/323456206_Web_Services_A_Comparison_of_Soap_and_Rest_Services [2.svibnja.2020]

11. IBM (2020) IBM Knowledge Center What is a web service [online]. Dostupno na: https://www.ibm.com/support/knowledgecenter/SSGMCP_5.2.0/com.ibm.cics.ts.webservices.doc/concepts/dfhws_definition.html [20.svibnja.2020]
12. ISVU [online]. Dostupno na: <https://www.isvu.hr/apiproba/dokumentacija/index.html>
13. Kukhnavets P. (2019) How to Organize Application Code With 3-Tier Architecture [online]. Dostupno na: <https://cuspy.io/blog/how-organize-application-code-with-3-tier-architecture/> [2.ožujka.2020]
14. Karunakaran K. (2020) Web Services, JSON, and XML [online]. Dostupno na: https://www.researchgate.net/publication/339754784_Web_Services_JSON_and_XML [22.travnja.2020]
15. Lange K. (2016) The Little Book on REST Services [online]. Dostupno na: <https://www.kennethlange.com/books/The-Little-Book-on-REST-Services.pdf> [4.svibnja.2020]
16. Loney K., Koch G. (2000). Oracle 8i : The complete reference. Berkeley, California: Osborne/McGraw-Hill.
17. Lozić D., Šimec A. (2014) Pametna komunikacija na Internetu preko REST protokola [online]. Dostupno na: https://bib.irb.hr/datoteka/713376.2014_MIPRO_Pametna_komunikacija_na_Internetu_preko_REST_protokola.pdf [10.travnja.2020]
18. Marrs T. (2017) JSON at Work. Sebastopol California:O'Reilly Media Inc.
19. Mehta M., Lee S. (2006) Service-Oriented Architecture: Concepts and Implementation [online]. Dostupno na: https://www.researchgate.net/publication/237776180_Service-Oriented_Architecture_Concepts_and_Implementation [3.svibnja.2020]
20. Oracle (2020) Oracle Cloud Infrastructure Getting Started Guide [online]. Dostupno na: https://docs.cloud.oracle.com/en-us/iaas/pdf/gsg/OCI_Getting_Started.pdf [12.lipnja.2020]
21. Oracle (2020) Oracle Cloud Infrastructure User Guide [online]. Dostupno na: https://docs.cloud.oracle.com/en-us/iaas/pdf/ug/OCI_User_Guide.pdf [12.lipnja.2020]
22. Oracle Cloud Infrastructure Documentation, <https://docs.cloud.oracle.com/en-us/iaas/Content/Database/Concepts/adboverview.htm> [16.svibnja.2020]
23. Oracle (2019) Oracle REST Data Services Installation, Configuration, and DevelopmentGuide [online]. Dostupno na:

- <https://docs.oracle.com/en/database/oracle/oracle-rest-data-services/19.2/aelig/oracle-rest-data-services-installation-configuration-and-development-guide.pdf> [20.veljače.2020]
24. Oracle Spatial Summit at BIWA (2017) RESTful Spatial services with ORDS [online]. Dostupno na: https://download.oracle.com/otndocs/products/spatial/pdf/biwa2017/Biwa2017_RESTful_Spatial_Services_Lapp_842619.pdf [21.veljače.2020]
 25. Oracle (2019) Oracle SQL Developer User's Guide [online]. Dostupno na: <https://docs.oracle.com/en/database/oracle/sql-developer/19.1/rptug/oracle-sql-developer-users-guide.pdf> [15.veljače.2020]
 26. Oracle (2013) The Java EE 6 Tutorial, What Are Web Services [online]. Dostupno na: <https://docs.oracle.com/javae/6/tutorial/doc/gijvh.html> [20.svibnja.2020]
 27. Oracle Tutorial (2020) Oracle Database Architecture [online]. Dostupno na: <https://www.oracletutorial.com/oracle-administration/oracle-database-architecture/> [26.kolovoza.2020]
 28. Oracle Using JAX-RS Client API to consume RESTful Web Services [online]. Dostupno na: <https://www.oracle.com/webfolder/technetwork/tutorials/obe/java/JAXRS2/jaxrs-clients.html> [20.svibnja.2020]
 29. Pejić Bach, M. et al. (2016) Informacijski sustavi u poslovanju. Zagreb:Ekonomski fakultet Zagreb
 30. Postman Learning Center [online]. Dostupno na: <https://learning.postman.com/> [5.ožujka.2020]
 31. Pennington C. et al. (2006) Introduction to Web Services [online]. Dostupno na: https://www.researchgate.net/profile/Jorge_Cardoso4/publication/236860265_Introduction_to_Web_Services/links/02e7e5273ff5a8c4f9000000/Introduction-to-Web-Services.pdf [20.veljače.2020]
 32. Richardson L., Ruby S. (2007) RESTful Web Services. Sebastopol, California: O'Reilly Media Inc.
 33. Spec India (2019) Oracle Cloud Services: Detailed Overview And Benefits [online]. Dostupno na: <https://www.spec-india.com/blog/oracle-cloud-services-guide> [16.travnja.2020]
 34. Sungchul, L. et al. (2015) A Method for Secure RESTful Web Service A Method for Secure RESTful Web Service [online]. Dostupno na: https://www.researchgate.net/publication/298262824_A_Method_for_Secure_RESTful_Web_Service_A_Method_for_Secure_RESTful_Web_Service [19.veljače.2020]

35. Sanneboina A. (2020) Cloud computing [online]. Dostupno na:
https://www.researchgate.net/publication/340332092_cloud_computing [10.svibnja.2020]
36. Spremić, M. (2017). Digitalna transformacija poslovanja. Zagreb:Ekonomski fakultet Zagreb.
37. Tilkov, S. (2010) A Brief Introduction to REST [online]. Dostupno na:
https://www.fer.unizg.hr/_download/repository/A_Brief_Introduction_to_REST.pdf
[20.veljače.2020]
38. Trstenjak B. (2010) SOAP komunikacijski protokol [online]. Dostupno na:
<https://hrcak.srce.hr/55559> [5.svibnja.2020]
39. Vaqqas M. (2014) RESTful Web Services: A Tutorial [online]. Dostupno na:
<https://www.drdoobs.com/web-development/restful-web-services-a-tutorial/240169069>
[18.veljače.2020]
40. Varga M. (1994) Baze podataka Konceptualno, logičko i fizičko modeliranje podataka. Zagreb:DRIP.

POPIS TABLICA

Tablica 1. REST elementi podataka	15
Tablica 2. REST konektori.....	16
Tablica 3. REST komponente	17
Tablica 4. Karakteristike HTTP metoda.....	27
Tablica 5. Usporedba API metoda	28

POPIS SLIKA

Slika 1: Arhitektura poslužitelja baze podataka Oracle	4
Slika 2. Komunikacija sa web servisom.....	6
Slika 3. Prelazak servisa na REST protokol.....	7
Slika 4. Arhitektura računarstva u oblaku	9
Slika 5. Arhitektura regije Oracle Cloud-a.....	14
Slika 6. HTTP zahtjev	21
Slika 7. HTTP POST zahtjev	22
Slika 8. HTTP odgovor	22
Slika 9. Odgovor na POST zahtjev	23
Slika 10. HTTP osnovna provjera identiteta	25
Slika 11. Autentifikacija na temelju tokena	26
Slika 12. Primjer poruke u JSON formatu	27
Slika 13. B2B aplikacija prodaje - produkcijsko okruženje.....	31
Slika 14. B2B aplikacija prodaje - razvojno okruženje.....	32
Slika 15. Konzola Oracle oblaka.....	33
Slika 16. Oracle REST Data Service.....	34
Slika 17. ER dijagram B2B aplikacije	36
Slika 18. Relacijska shema B2B aplikacije.....	37
Slika 19. DDL kategorija proizvoda.....	38
Slika 20. DDL grupa proizvoda	39
Slika 21. DDL jedinična mjera.....	39
Slika 22. DDL proizvod	40
Slika 23. DDL kupac	41
Slika 24. DDL adresa	41

Slika 25. DDL status narudžbe.....	42
Slika 26. DDL narudžba.....	42
Slika 27. DDL stavka narudžbe.....	43
Slika 28. Procedura ORDS.ENABLE_OBJECT	44
Slika 29. Kreiranje resursa kupci	45
Slika 30. Kreiranje resursa adrese	45
Slika 31. Kreiranje resursa narudzbe.....	45
Slika 32. Kreiranje resursa stavkenarudzbe	46
Slika 33. Kreiranje resursa vproizvodi	46
Slika 34. Kreiranje resursa vstavkenarudzbe	46
Slika 35. Katalog metapodataka	47
Slika 36. Metapodaci resursa stavkenarudzbe.....	50
Slika 37. Dijagram slučajeve korištenja	51
Slika 38. Dodavanje kupca (tijelo zahtjeva).....	51
Slika 39. Dodavanje kupca (tijelo odgovora).....	52
Slika 40. SQL naredba za dodavanje kupca	52
Slika 41. Dodavanje adrese (tijelo zahtjeva)	53
Slika 42. Dodavanje adrese (tijelo odgovora)	53
Slika 43. SQL naredba za dodavanje adrese	54
Slika 44. Pregled proizvoda uz uvjet (tijelo odgovora).....	54
Slika 45. SQL naredba za pregled proizvoda	55
Slika 46. Dodavanje narudžbe (tijelo zahtjeva).....	55
Slika 47. Dodavanje narudžbe (tijelo odgovora).....	56
Slika 48. SQL naredba za dodavanje narudžbe.....	56
Slika 49. Dodavanje stavke narudžbe (tijelo zahtjeva)	56
Slika 50. Dodavanje stavke narudžbe (tijelo odgovora)	57
Slika 51. SQL naredba za dodavanje stavke narudžbe.....	57
Slika 52. Promjena stavke narudžbe (tijelo zahtjeva)	58
Slika 53. Promjena stavke narudžbe (tijelo odgovora)	58
Slika 54. Brisanje stavke narudžbe (tijelo odgovora)	59
Slika 55. SQL naredba za brisanje stavke narudžbe	59
Slika 56. Upit u narudžbe (tijelo odgovora).....	59

ŽIVOTOPIS



Životopis

OSOBNJE INFORMACIJE

Škrtić Ivana

📍 Stara Cesta 37, 47250 Duga Resa (Hrvatska)

☎ +385955336867

✉ ivana.skrkich@gmail.com

OSOBNJI PROFIL

Student Ekonomskog fakulteta u Zagrebu smijer Menadžerska informatika u potrazi za karijerom

OBRAZOVANJE I OSPOSOBLJAVANJE

01.10.2015.–danas

Ekonomski fakultet Zagreb, Zagreb (Hrvatska)

01.9.2011.–01.6.2015.

Ekonomist

Ekonomsko turistička škola, Karlovac (Hrvatska)

01.9.2003.–15.6.2011.

Osnovna škola Dubovac, Karlovac (Hrvatska)

OSOBNJE VJEŠTINE

Materinski jezik hrvatski

Strani jezici

	RAZUMIJEVANJE		GOVOR		PISANJE
	Slušanje	Čitanje	Govorna interakcija	Govorna produkcija	
engleski	C1	C1	B1	C1	B2

Stupnjevi: A1 i A2: Početnik - B1 i B2: Samostalni korisnik - C1 i C2: Iskusni korisnik
Zajednički europski referentni okvir za jezike - Ljestvica za samoprocjenu

Komunikacijske vještine

- Spremna za timski rad
- Kooperacija i razumijevanje

Digitalne vještine

SAMOPROCJENA				
Obrada informacija	Komunikacija	Stvaranje sadržaja	Sigurnost	Rješavanje problema
Iskusni korisnik	Iskusni korisnik	Iskusni korisnik	Samostalni korisnik	Samostalni korisnik

Digitalne vještine - Tablica za samoprocjenu