Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.9
DOI: https://doi.org/10.24867/10BE26Krivokapic

UPOREDNA ANALIZA ALATA ZA UPRAVLJANJE *API*-JIMA

COMPARATIVE ANALYSIS OF API MANAGEMENT TOOLS

Marija Krivokapić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – RAČUNARSTVO I AUTOMATIKA

Kratak sadržaj – *Ovaj rad se bavi uporednom analizom sledeća tri alata: Akana, Apigee i 3Scale. Osnovnih 6 koraka koji su sastavni deo životnog ciklusa jednog API-ja je odrađeno u svakom od navedena 3 alata, ne bi li se utvrdilo koji je najlakši za upotrebu i koji pruža najviše mogućnosti.*

Ključne reči: *API, alati, API Menadžment, Akana, Apigee, 3Scale*

Abstract – *Main goal of this paper is to perform comaprative analysis of following 3 tools: Akana, Apigee and 3Scale. Basic 6 steps, that are part of API lifecycle, have been performed in each of 3 mentioned tools, in order to realize which of them is easiest in reallife use, and as well which of them is offering the biggest number of possibilities.*

Keywords: *API, tools, API Management, Akana, Apigee, 3Scale*

1. UVOD

U današnje vreme, eri računara, napredak tehnologije je sve veći. Budući da je život nemoguće zamisliti bez elektronskih uređaja, interneta, društvenih mreža, pojavljuje se velika ekspanzija novih tehnologija koje olakšavaju kreiranja svega što je potrebno korisnicima. Jedna od tih tehnologija je predstavljanje servisa preko *API*-ja radi lakšeg i sigurnijeg korišćenja veb i mobilnih aplikacija. Inicijalno, *API* je kreiran u svrsi definisanja i opisivanja bilo kog programskog interfejsa za biblioteku, ili modula koji predstavljaju deo nekog većeg softverskog sistema. Tokom poslednjih par godina, najčešće se koristi kao arhitekturalni stil u klijent-server komunikaciji, sa oslanjanjem na HTTP protokole [1].

Sa pojavom *API*-ja, nastaje i nova oblast rada pod nazivom *API Management*. Ta oblast obuhvata procese za distribuciju, kontrolu i analizu *API*-ja. Glavni cilj *API Management*-a je što veće olakšavanje potreba programera, prilikom kreiranja *API*-ja, kao i tokom nadgledanja aktivnosti prilikom upotrebe već postojećih *API*-ja. Jedna od najbitnijih prednosti *API*-ja je mogućnost brzih izmena samog servisa radi ispunjavanja korisničkih zahteva. Upotreba mikroservisne arhitekture umnogome pomaže ubrzavanju razvijanja softvera. HTTP orijentisani API-ji predstavnjaju sponu između mikroservisa u sinhronoj interakciji mikroservisne arhitekture. Još jedna korisna strana *API*-ja je ta što upotrebom različitih polisa obezbeđuju sigurnost [2].

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Branko Milosavljević, red. prof.

2. KORIŠĆENI ALATI

2.1. Akana

Ovaj alat omogućava potpuni životni ciklus *API Management*-a sa kraja na kraj. Taj životni ciklus obuhvata dizajniranje, implementiranje, obezbeđivanje sigurnosti, nadgledanje (*monitoring*) i na kraju publikovanje *API*-ja. U okviru svega prethodno navedenog se podrazumeva razmena podataka preko *API*-ja, kao i povezivanje i integrisanje različitih aplikacija [3].

2.2. Apigee

Isto kao prethodno opisan alat, i Apigee omogućava potpuni životni ciklus *API Management*-a sa kraja na kraj. Pored svega pomenutog, Apigee nudi mogućnost monetizacije [4]. Takođe, još neke od pogodnosti koje ovaj alat nudi su automatsko razvijanje dokumentacije, podržavanje cloud sistema, upotreba principa mašinskog učenja zarad što boljih analiza [5].

2.3. 3Scale

3Scale je alat koji omogućava potpuni životni ciklus *API Management*-a sa kraja na kraj, i kao i Apigee pruža mogućnost monetizacije. Neki od benefita ovog alata su: kontrola saobraćaja, integracija sa OpenShift-om i RedHat Fuse-om, podržavanje hybrid cloud sistema... [6].

3. METOD ISTRAŽIVANJA

3.1. Kreiranje *API*-ja

Prilikom njegovog kreiranja najbitnije je definisati 3 stvari:

- Pristupni (*access*) URL – to je URL na koji će se slati zahtevi sa aplikacije (neki *frontend*)
- Metode – resursi koji će biti omogućeni na *API*-ju. U okviru njihovog definisanja specificira se koji HTTP glagol će se odnositi na taj resurs kao i sam naziv resursa.
- Backend URL – predstavlja pritupnu tačku krajnjeg sistema. Na ovaj URL se prosleđuju primljeni zahtevi.

3.2. Konfiguracija autorizacije

U idealnom slučaju, zarad što veće sigurnosti, trebalo bi imati 2 autorizacije, jednu postavljenu između klijenta i *API*-ja, a drugu između *API*-ja i *backend*-a. Ove konfiguracije se razlikuju od platforme od platforme, ali u najčešćem slučaju se to podešava uz pomoć ugrađenih polisa.

3.3. Dodavanje restrikcija

U cilju sprečavanja preopterećenja servera, DDoS napada, kao i zbog oslobođanja resursa u što kraćem vremenskom roku, uvode se različite restrikcije. Tri najvažnije restrikcije su: Concurrency, Quota i Timeout. Pored ovih restrikcija, zarad ograničavanja dolaznih zahteva, kreiraju se CORS polise.

3.4. Izmena zahteva

Postoje slučajevi kada *backend* očekuje zahtev drugačije formatiran u odnosu na zahtev koji je prvobitno došao do *API*-ja. Kod svakog zahteve se mogu modifikovati i zaglavlja i samo telo zahteva.

Ukoliko se radi o slučaju izmene tela zahteva, kao što je npr. pretvaranje iz XML-a u JSON, najčešće se koriste ugrađene polise.

3.5. Testiranje *API*-ja

Radi provere same funkacionalnosti, kao i verifikacije da li su ispunjeni svi klijentski zahtevi, nakon kreiranja *API*-ja sledi njegovo testiranje. Pored validnih, često se svesno šalju neispravni zahtevi ne bi li se proverilo da li je pokriven svaki slučaj koji kasnije može dovesti do velikih problema i incidenata. Još jedna od vrsta testiranja su stres ili performans testovi. Prilikom ovakvog testiranja šalje se velik broj zahteva, i cilj je simuliranje realnog sistema.

3.6. Publikovanje i monitoring

Od momenta publikovanja tj. postavljanja *API*-ja na produkciju, on postaje javno dostupan što znači da pored programera, i ljudi iz spoljnih sistema mogu slati zahteve na taj *API*. Nakon toga, kad *API* počne da se upotrebljava, programeri imaju mogućnost monitoringa. To podrazumeva pregled svih primljenih zahteva, bilo da su oni uspešni ili ne.

4. REZULTATI

4.1. Akana

Prilikom kreiranja *API*-ja programeri koriste dva portala:

- Community Manager (CM) – za kreiranje *API*-ja, konfigurisanje metoda, URL-ova, kreiranje aplikacija, testiranje i nadgledanje *API*-ja
- Policy Manager (PM) – takođe nudi mogućnost kreiranja i nadgledanja *API*-ja, ali je manje *user friendly*, te se ne koristi u te svrhe. Za razliku od CM-a, programeri samo ovde mogu kreirati polise za restrikcije. S druge strane, PM ne nudi mogućnost testiranja *API*-ja.

4.1.1. Kreiranje *API*-ja

U slučaju kreiranja REST *API*-ja, bira se prva opcija prilikom čega se unose naziv *API*-ja i *backend* URL. Ukoliko se pak radi o SOAP *API*-ju, bira se druga opcija, gde se uveze fajl koji se prethodno mora nalaziti lokalno na samoj mašini.

Sledeći korak je definisanje metoda. Kod SOAP *API*-ja je ovaj korak nepotreban budući da se sve te informacije već prethodno trebaju nalaziti u Swagger/WSDL fajlovima. Prilikom kreiranja metoda definiše se njen naziv i bira se vrsta metode tj. bira se HTTP glagol.

Zatim sledi definisanje opisa samog *API*-ja, koje uključuje unos detalja o *API*-ju, podataka o verzijama, dodavanje ikonice i tagova.

4.1.2. Konfiguracija autorizacije

Za potrebe autorizacije, kod Akane se kreiraju aplikacije. Svaka aplikacija ima svoj jedinstven identifikator i šifru, koji imaju ulogu *username*-a i *password*-a prilikom kreiranja validnih tokena. Prilikom njenog kreiranja programer definiše naziv, identifikator (App ID) i šifru (Shared Secret).

Zatim se treba vratiti na *API*, i sa toolbar-a koji se nalazi sa leve strane treba odabrati deo „Implementations“. Na sredini se nalazi deo sa polisama koji je takođe bitan deo podešavanja autorizacije. Tu se mogu naći *default*-ne ugrađene polise, ali takođe i polise kreirane od strane programera. Svaka od polisa se može primeniti na bilo koji *API*.

4.1.3. Dodavanje restrikcija

Restrikcije se definišu na nivou koji je između *API*-ja i aplikacije. Kreiranje restrikcija se vrši u PM-u. U zavisnosti od vrste polise se bira podfolder, „Operational Policies“ za CORS polise ili „QoS Policies“ za kreiranje polisa koje ograničavaju broj zahteva ili otvorenih konekcija u određenom vremenskom periodu i sl.

Nakon što su sve neophodne polise kreirane, aplikacija i API se povezuju. Njihovo povezivanje se vrši u CM-u. Klikom na „Access“ dugme na stranici *API*-ja, programerima se nudi lista postojećih aplikacija i oni biraju sa kojom aplikacijom iz liste žele da povežu svoj *API*. Na kraju ostaje da odabere koje sve restrikcije želi da važe za njegov *API*.

4.1.4. Izmena zahteva

Bilo kakva izmena zahteva u Akani podrazumeva izmenu procesa samog resursa. „Receive“ i „Reply“ se smatraju početnom i krajnjom aktivnošću i automatski se kreiraju prilikom kreiranja svake metode. Svaka željena izmena zahteva se izvršava definisanjem odgovarajućih skripti koje moraju biti smeštene između „Receive“ i „Reply“ aktivnosti.

Ukoliko je potrebno izvršiti transformaciju formata tela zahteva, koristi se aktivnost pod nazivom „FreeMarker“.

Za dodavanje i uklanjanje zaglavlja, kao i za modifikacije izgleda resursa koristi se aktivnost pod nazivom „Script“.

4.1.5. Testiranje *API*-ja

Potrebno je iz toolbar-a odabrati opciju „Test Client“. Zatim se programer preusmerava na stranicu za terstiranje gde može da odabere koju medotu želi da testira. Dugme „Setup“ služi za odabir aplikacije, tj. zahteve kog klijenta programer želi da simulira, ukoliko je više aplikacija povezano sa jednim *API*-jem. Klikom na dugme „Invoke“ inicira se zahtev, koji simulira klijentski zahtev.

4.1.6. Publikovanje i monitoring

Akana je konfigurisana tako, da nisu potrebni posebni koraci za publikovanje. Onog momenta kada se definišu zona i *context path*, *API* postaje javno dostupan. Do logova i pregleda prispelih zahteva se može doći odlaskom na deo pod nazivom „Analytics“. Pored vremena, može se filtrirati i po aplikaciji sa koje je došao zahtev. Uspešni zahtevi su označeni zelenom bojom, a pali crvenom. Prikazane su osnovne informacije kao što su status kod, vreme iniciranja zahteva, utrošeno vreme na odgovor... Klikom na bilo koji od tih zahteva, dobija se detaljniji pregled za taj konkretan zahtev.

4.2. Apigee

Ceo proizvod se sastoji iz sledeće 3 komponente:

- Apigee Edge – glavna komponenta, koja omogućava izvršavanje svih koraka potrebnih za nastanak jednog *API*-ja, počevši od kreiranja pa sve do njegovog publikovanja i monitoringa.

- Apigee Sense – komponenta kojoj je glavna uloga zaštita *API*-ja od neželjenih zahteva uključujući i napade od zlonamernih klijenata. Ova komponetna konstantno analizira saobraćaj zahteva, pri čemu identifikuje šablone koji mogu predstavljati nepoželjne zahteve.
- Apigee Monetization – podkomponenta koja programerima pruža mogućnost pristupa planovima stopa pretplata budućih klijenata, omogućava automatizaciju procesa isplaćivanja i naplate,...

4.2.1. Kreiranje *API*-ja

Prvobitno se kreira specifikacija koja je u stvari fajl tipa YAML i u sebi sadrži opise samih *API*-ja, metode tog *API*-ja zajedno sa HTTP glagolima, parametrima i kodovima odgovora.

Sledeći korak je kreiranje proksija. Odabirom opcije za kreiranje novog proksija programeru se nudi izbor vrste proksija. Najčešću upotrebu ima „Reverse Proxy“, tj. proksi kod kog se konfiguriše krajnji URL *backend* servisa.

Poslednji korak prilikom kreiranja proksija je izbor da li taj proksi treba da postoji na testnom ili produkcionom okruženju, ili pak na oba.

4.2.2. Konfiguracija autorizacije

Iz sigurnosnih razloga, Apigee uvodi proizvode koji predstavljaju neki konkretan *API*. Nakon uspešno kreiranog proizvoda, sledi kreiranje aplikacije. Prilikom kreiranja aplikacije osim osnovnih detalja kao što je naziv, treba upisati listu programera, ili bar jednog, kojima će biti dozvoljen pristup aplikaciji.

Po završetku kreiranja proizvoda i aplikacije, sledi dodavanje polise za autentifikaciju. Prvo se treba vratiti na detalje proksija i otići na *Develop* tab. Postoje predefinisane polise za različite vrste autentifikacije. Posle kreiranja polisa, one će postati vidljive u odeljku sa polisima, te ih onda treba prevući na dijagram koji simulira izgled toka.

4.2.3. Dodavanje restrikcija

Dodavanje restrikcija je poprilično slično dodavalju polisa za autentifikaciju, s tim što u ovom slučaju treba posmatrati početni deo liste vrsta polisa označen kao „Traffic Management“. Restrikcija koja se najčešće koristi kao zaštita od napada je ograničavanje broja zahteva poslatih u određenom vremenskom intervalu. Ona se dodaje kreiranjem polise tipa „Spike Arrest“. Ova polisa je tipa XML i ograničenje, tj. broj zahteva u intervalu se specificira pod tagom „<Rate>“.

4.2.4. Izmena zahteva

Ukoliko je potrebno dodavanje nekih zaglavlja sa predefinisanim vrednostima ili preuzetim iz tela inicjalnog zahteva, preporučuje se upotreba polisa koje su u suštini JavaScript fajlovi.

U slučaju transformacija zahteva iz jednog tipa u drugi kao npr. iz JSON-a u XML i obrnuto, Apigee nudi ugrađene polise. Nakon kreiranja neke od ovih polisa neophodno je jedino prevući ih na deo toka u dijagramu.

4.2.5. Testiranje *API*-ja

Proces testiranja počinje klikom na dugme pod nazivom „Start Trace Session“. Jedna sesija traje 10 minuta. Za slanje zahteva preporučuje se upotreba nekog drugog alata kao što je npr. Postman. Programer se može kretati kroz svaki momenat zahteva, izvršenje svake skripte, te pred kraj, kada se dođe do dela za odgovor tačno se ispisuje odgovor koji je *backend* poslao.

4.2.6. Publikovanje i monitoring

Nakon svakog kreiranja nove verzije, programer ima opciju da odluči da li će ta verzija ići na testno ili produkcino okruženje, birajući iz padajućeg menija dela „Deployment“.

Što se tiče monitoringa, u „Analyze“ delu, programeru se pružaju različite mogućnosti pregleda izveštaja: može da bira sa okruženja želi da vidi podatke; može da bira vremenski period za koji ga interesuje saobraćaj, kao i da li želi izveštaje za sve proksije, ili ga pak interesuje samo jedan konkretan proksi.

4.3. 3Scale

Poslednji alat koji će biti obrađen, slično kao i prethodni, sastoji se iz više komponenti:

- Admin portal – glavni portal gde se kreiraju i definišu sve bitne stvari vezane za jedan *API*. Ono što je specifično je činjenica da svaki *API*, ima svoj poseban admin portal kom se pristupa preko jedinstvenog URL-a. Onog momenta kada se definiše „company name“, automatski se kreiraju linkovi za developer i admin portal sa istim nazivom.
- Developer portal – portal čiji je glavni cilj pružanje što boljeg iskustva i olakšavanje programerima koji će kasnije manipulisati sa već kreiranim *API*-jima. To se ostvaruje omogućavanjem sve neophodne dokumentacije zajedno sa primerima kodova, slučajevima upotrebe, cenovnim planovima itd.

4.3.1. Kreiranje *API*-ja

Za razliku od alata koji su prethodno obrađeni, u slučaju ovog alata, nema potrebe za ručnim kreiranjem *API*-ja budući da se on automatski kreira, sa predefinisanim vrednostima. Svaki od ovih podataka se kasnije treba izmeniti spram klijentskih zahteva.

Sledeći korak koji je dodavanje metoda. Iznad liste svih metoda, koja je inicijalno prazna se nudi opcija pod nazivom „+ New method“.

Poslednja stvar koja se treba izmeniti jeste *backend* URL.

4.3.2. Konfiguracija autorizacije

Prvi korak je kreiranje aplikacionog plana. Za jedan *API* može da se definiše više aplikacionih planova, gde uvek postoje dva predefinisana. U okviru plana se definiše lista korisnika i kreira se aplikacija.

Nakon što su sve potrebne komponente kreirane, treba definisati vrstu autorizacije. Naime, 3Scale, nudi 3 izbora: Api key, Uređeni par app_id i api_key, OAuth. Pored izbora vrste autentifikacije, programer ima mogućnost odabira lokacije kredencijala.

4.3.3. Dodavanje restrikcija

Restrikcije se definišu u okviru aplikacionog plana, s tim što ovaj alat nudi isključivo restrikciju koja se odnosi na ograničavanje broja zahteva u određenom vremenskom intervalu. Odabirom opcije „+ New usage limit“ kreira se nova restrikcija tj. limit, i tom prilikom se definišu maksimalan broj zahteva i vremenski period.

4.3.4. Izmena zahteva

Odabirom dela „Policies“ iz „Integration“ odeljka, programeru se prikazuje lanac svih dodataih polisa i redosled njihovog izvršavanja.

Što se tiče polisa koje se odnose na izmene tela zahteva, to je moguće dodavanjem ekstenzija za polise, koje je moguće izvršiti samo uz pomoć Red Hat Fuse-a i OpenShift klastera, za šta su neophodne mnoge dodatne instalacije.

4.3.5. Testiranje *API*-ja

Da bi započela testiranja *API*-ja, on se prvo mora prebaciti u fazu za testiranje.

Potom treba kreirati novu specifikaciju odlaskom na opciju „ActiveDocs“. Nakon što programer sva polja popuni popuni, klikom na dugme „Try it out!“, okida se zahtev.

4.3.6. Publikovanje i monitoring

Postupak publikovanja je identičan stavljanju *API*-ja na testno okruženje, osim sto treba odabrati opciju „Promoting v.1 to Production APIcast“.

Na samom kraju ostaje još samo monitoring. Odlaskom na odeljak „Analytics“, nude se različite mogućnosti nadgledanja *API*-ja na produkciji. Pored ukupnog saobraćaja, nude se mogućnosti pregleda proseka na dnevnom nivou, na nivou sata, alerti, kao i status kodovi zahteva.

5. ZAKLJUČAK

API Management, iako relativno nova tehnologija, umnogome pomaže i olakšava posao programerima omogućavajući dosta funkcionalnsti, unapred predefinisanih. Što se tiče ključnih osobina alata za upravljanje *API*-jima, svaki od ovih alata ih podržava. Lista ključnih osobina je sledeća:

- Access Control – svaki od alata podržava kontrolu pristupa na taj način što administratori definišu listu programera koji imaju pravo pristupa.
- Analytics – kao i prethodnu osobinu, svaki od 3 alata koja su obrađena u ovom radu, podržava analizu javno dostupnih servisa, pružajući informacije o saobraćaju, broju uspešnih i neuspešnih zahteva, status kodovima itd. Ipak, kod ove osobine, najbolje se pokazala Akana, budući da ona, osim informacija o uspešnosti zahteva, čuva podatke i o tome kako su izgledali zahtevi.
- API Design – osobina prisutna kod svih alata. U svakom od njih dizajniranje tj. kreiranje API-ja je izuzetno jednostavan korak.
- Testing Management – svaki alat pruža mogućnost testiranja API-ja pre nego što postanu javno dostupni. U slučaju testiranja, najbolje se pokazao Apigee, zato što kod svakog zahtev koji se pošalje u toku jedne sesije, moguće je ispratiti i pregledati svaki odrađen korak, počevši od izgleda svih zaglavlja i tela, do izvršenja svake polise koja je ubačena u tok..
- Threat & Traffic Protection – osobina koja se izvršava uz pomoć restrikcija, preko predefinisanih ili ručno kreiranih polisa. Slično prethodnoj osobini, i ovde se može reći da je Apigee najbolji alat, zato što pruža najveći broj ugrađenih polisa koje programeri mogu koristiti.
- Version Control – poslednja u nizu osobina koju pruža svaki do alata. Dodavanje nove verzije je poprilično slično realizovano u svakom od alata.

Na samom kraju, može se izvući zaključak da je Apigee alat koji je najbolje koristiti u cilju upravljanja *API*-jima. Ovo se odnosi na perspektivu programera, tj. Apigee je najjednostavniji i pruža najviše mogućnosti, ugrađenih komponenti, olakšavajući programerima ceo životni ciklus jednog *API*-ja.

6. LITERATURA

[1] https://medium.com/@robert.broeckelmann/what-are-apis-the-technology-perspective-ca7e33d383c1 (pristupljeno u aprilu 2020.)

[2] https://www.redhat.com/en/topics/api/what-is-api-management (pristupljeno u aprilu 2020.)

[3] https://www.akana.com/products/api-platform (pristupljeno u aprilu 2020.)

[4] https://docs.apigee.com/api-platform/get-started/what-apigee-edge (pristupljeno u aprilu 2020.)

[5] https://searchapparchitecture.techtarget.com/definition/Apigee (pristupljeno u aprilu 2020.)

[6] https://www.redhat.com/en/technologies/jboss-middleware/3scale (pristupljeno u maju 2020.)

Kratka biografija:

Marija Krivokapić rođena je 21.12.1995. godine u Novom Sadu. Završila je osnovnu školu „Petefi Šandor“ kao nosilac Vukove diplome. Gimnaziju „Jovan Jovanović Zmaj“ završava 2014. godine kao nosilac Vukove diplome, tokom čijeg pohađanja je bila i član školskog hora. Školske 2014/2015 upisuje Fakultet tehničkih nauka, smer Elektroenergetski softverski inženjering. Letnji semestar školske 2017/2018 bila je na razmeni u Univerzitetu u Groningenu, u Holandiji, gde je položila sve predmete. U septembru 2018. završava osnovne studije, nakon čega upisuje master studije takođe na Fakultetu Tehničkih nauka, smer Računarstvo i Automatika, podsmer – Elektronsko poslovanje.