



Web Servisi

Osnove · SOAP · WSDL · REST · GraphQL · gRPC · Najnoviji trendovi

Interoperabilnost informacijskih sustava

doc.dr.sc. Aleksander Radovan, prof. struč. stud.
Algebra Bernays University · Akademski godina 2025./26.



Sadržaj predavanja

01

Osnove web servisa

Definicija, značajke, arhitektura

02

SOAP - dublja analiza

Protokol, struktura poruke, Java primjeri

03

WSDL

Jezik opisa servisa, struktura, alati

04

Prednosti, nedostaci i primjene

Kada SOAP ima smisla (a kada ne)

05

Alternativne tehnologije

REST, GraphQL, gRPC – usporedba

06

WS-* i moderni standardi

Druga i treća generacija WS-a

07

Trendovi i statistike 2025.

Podaci o stvarnoj primjeni

08

Resursi za učenje

Videozapisi, dokumentacija, alati

01 · Osnove web servisa

Definicija · Značajke · Arhitektura · Model procesa

▶ [Web Services Introduction](#)



Što su web servisi?

W3C Definicija

"Softverski sustavi dizajnirani za podršku međusobno interoperabilne (stroj-prema-stroju) interakcije putem mreže, s opisom sučelja u obliku koji računalo može obraditi (primarno WSDL), komuniciranjem putem SOAP poruka kroz HTTP s XML serijalizacijom."



Interoperabilni

Neovisni o platformi i jeziku – Java komunicira s .NET-om i Pythonom



Mrežno temeljeni

Rade putem standardnih internetskih protokola (HTTP/HTTPS)



Samoodescriptivni

WSDL klijentima točno govori kako koristiti servis



Labavo povezani

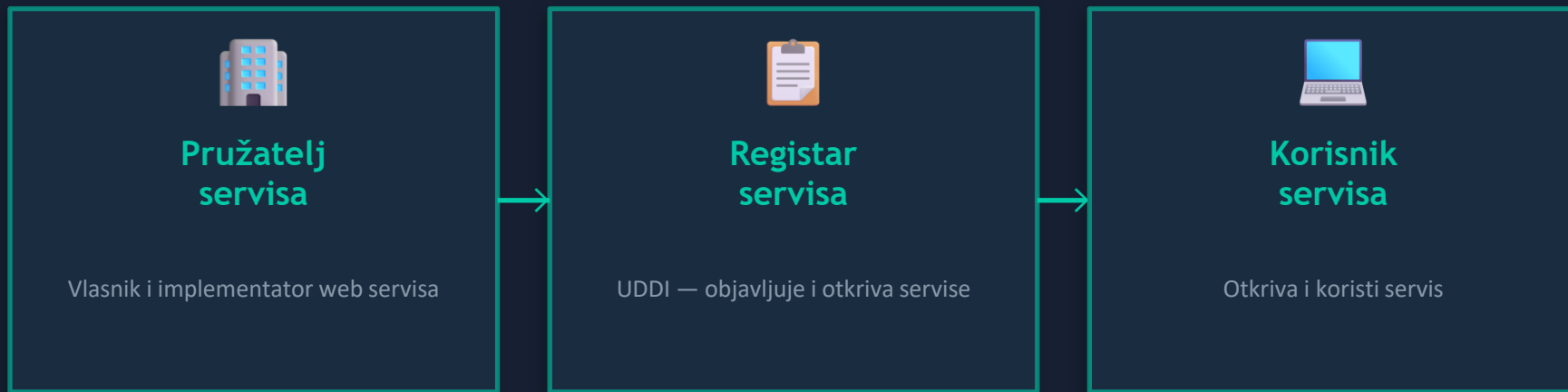
Pružatelj i potrošač mogu se razvijati neovisno



Ključno: Web servisi NISU namijenjeni krajnjim korisnicima — omogućuju komunikaciju stroj-prema-stroju. Razmišljajte API-prema-API, ne preglednik-prema-poslužitelju.



Model procesa web servisa



- 1 Pružatelj objavljuje opis servisa u registar
- 2 Korisnik otkriva servis u registru
- 3 Korisnik se poveže s pružateljem i počinje razmjenu poruka
- 4 Neki koraci su automatizirani; implementacija može zahtijevati ljudsku intervenciju

02 · SOAP - Simple Object Access Protocol

Povijest · Struktura poruke · Omotnica · Java primjeri



SOAP - Uvod i povijest

1998.

Osnovna ideja
(Winer, Box,
Atkinson)

2000.

SOAP 1.1
objavljeno

2003.

SOAP 1.2
W3C standard

2007.

SOAP 1.2
2. izdanje



Što je SOAP?

- Simple Object Access Protocol
- Lagani protokol za razmjenu poruka temeljen na XML-u
- Omogućuje komunikaciju između distribuiranih sustava
- Neovisan o transportu (HTTP, SMTP, JMS...)
- SOAP 1.2: više nije kratica!
- W3C standard: <http://www.w3.org/TR/soap12>



Dijelovi specifikacije SOAP 1.2

- Dio 0: SOAP Primer (neinformativni tutorial)
- Dio 1: SOAP Messaging Framework
 - Model obrade poruka
 - Model proširivosti
 - Okvir za vezivanje protokola
- Dio 2: SOAP Adjuncts (enkodiranje, RPC)

Struktura SOAP poruke

Envelope (Omotnica)

Korijenski element — označava XML dokument kao SOAP poruku

Header (Zaglavlje) -
neobavezno

Metapodaci poruke: tokeni autentikacije, ID transakcije, usmjeravanje

Body (Tijelo)

Stvarni sadržaj — zahtjev ili odgovor (obavezno)

Fault (Greška) -
neobavezno

Informacije o grešci: VersionMismatch, MustUnderstand, Server, Client



Prostori imena:

SOAP 1.1: `http://schemas.xmlsoap.org/soap/envelope/`

SOAP 1.2: `http://www.w3.org/2003/05/soap-envelope`



SOAP Omotnica - XML primjer

SOAP 1.2 XML

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
  xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
  soap:encodingStyle="http://www.w3.org/2003/05/soap-encoding">

  <soap:Header>
    <auth:Security xmlns:auth="http://security.example.org">
      <auth:Token>Bearer eyJhbGciOiJIUzI1NiJ9...</auth:Token>
    </auth:Security>
  </soap:Header>

  <soap:Body>
    <m:GetStudentInfo xmlns:m="urn:StudentService">
      <m:studentId>1234567</m:studentId>
    </m:GetStudentInfo>
  </soap:Body>

</soap:Envelope>
```



Java SOAP primjer - Poslužiteljska strana (1/2)

HelloWorldSOAPService – implementacijska klasa web servisa:

```
import javax.jws.WebService;
import javax.jws.WebMethod;

@WebService(name = "HelloWorldSOAP",
            targetNamespace = "urn:HelloWorld")
public class HelloWorldSOAPService {

    @WebMethod
    public String sayHello(String firstName,
                          String lastName) {
        return firstName + " " + lastName
            + ", dobrodošli! 🎉";
    }

    @WebMethod
    public String addString(String symbol,
                          String dataType) {
        return symbol + dataType;
    }
}
```

Java (JAX-WS)

Ključne anotacije

- @WebService — označava klasu kao WS krajnju točku
- targetNamespace — SOAP namespace
- name — naziv servisa u WSDL-u
- @WebMethod — izlaže metodu u WSDL ugovoru
- Povratni tipovi automatski se mapiraju na XSD

Java SOAP primjer - Klijentska strana (2/2)

TestClient – programatski poziv SOAP servisa putem Apache Axis:

```
Service service = new Service();
Call call = (Call) service.createCall();
call.setTargetEndpointAddress(
    new URL("http://localhost:8080/HelloWorld"));

// Odabir operacije
call.setOperationName(
    new QName("urn:HelloWorld", "sayHello"));

// Postavljanje parametara
call.addParameter("p1", XSD_STRING, IN);
call.addParameter("p2", XSD_STRING, IN);
call.setReturnType(XSD_STRING);

// Pozivanje i ispis rezultata
String result = (String) call.invoke(
    new Object[]{"Ana", "Kovač"});
System.out.println("Rezultat: " + result);
```

Java (Apache Axis)

Koraci na klijentskoj strani

- 1. Kreiraj Service i Call objekte
- 2. Postavi URL krajnje točke
- 3. Odredi naziv operacije (QName)
- 4. Registriraj ulazne/izlazne tipove
- 5. invoke() šalje SOAP zahtjev
- 6. Obradi tipiziran odgovor
- ⚠ Moderno: koristi wsimport za autogeneriranje stub klasa



SOAP Zahtjev i odgovor (XML)

ZAHTJEV →

```
<soapenv:Envelope
  xmlns:soapenv=
    "http://www.w3.org/2003/05/soap-envelope"
  xmlns:ns1="urn:HelloWorld">
  <soapenv:Body>
    <ns1:sayHello>
      <p1 xsi:type="xsd:string">
        Ana
      </p1>
      <p2 xsi:type="xsd:string">
        Kovač
      </p2>
    </ns1:sayHello>
  </soapenv:Body>
</soapenv:Envelope>
```

SOAP Zahtjev



← ODGOVOR

```
<soapenv:Envelope
  xmlns:soapenv=
    "http://www.w3.org/2003/05/soap-envelope"
  xmlns:ns1="urn:HelloWorld">
  <soapenv:Body>
    <ns1:sayHelloResponse>
      <return xsi:type="xsd:string">
        Ana Kovač,
        dobrodošli! 🇭🇷
      </return>
    </ns1:sayHelloResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

SOAP Odgovor



SOAP - Prednosti i nedostaci

✓ Prednosti

- ✓ Strogi WSDL ugovor — jasna API dokumentacija
- ✓ Ugrađena sigurnost (WS-Security)
- ✓ ACID transakcije (WS-AtomicTransaction)
- ✓ Neovisnost o protokolu (HTTP, SMTP, JMS...)
- ✓ Podrška enterprise alata (SAP, Oracle, IBM)
- ✓ Pouzdana dostava poruka (WS-ReliableMessaging)

✗ Nedostaci

- ✗ Opširan XML — velika potrošnja propusnosti
- ✗ Složena implementacija i otklanjanje grešaka
- ✗ Sporiji od REST/gRPC za jednostavne operacije
- ✗ Loša podrška u preglednicima (nema nativnog JS-a)
- ✗ Rigidan — promjene WSDL-a lome klijente
- ✗ Pretjerano za jednostavne API-je



SOAP - Idealni slučajevi primjene



Financijski sektor

Bankarski prijenosi, SWIFT poruke,
ISO 20022 platni API-ji



Zdravstvo

HL7 FHIR SOAP vezivanja, bolnički
informacijski sustavi



Putovanje i avijacija

Rezervacije letova (Amadeus,
Sabre GDS), elektronske karte



Državne e-usluge

Porezni sustavi, socijalno
osiguranje, digitalna identifikacija



ERP integracija

SAP, Oracle ERP API-ji — interne
poslovne procesne integracije



Telekom naplata

Sustavi naplate i pružanja telekom
usluga (OSS/BSS)



Visoko sigurni API-ji

Zahtjevi WS-Security, WS-Policy —
gdje REST zaostaje



Nasljedna integracija

Održavanje kompatibilnosti
unatrag sa starijim SOAP klijentima



Pravilo palca: Odaberi SOAP kada ti trebaju garantirana dostava, formalni ugovori, enterprise sigurnost ili složene transakcije. REST/gRPC za većinu modernih API-ja.

HR SOAP u Hrvatskoj - Tko ga koristi?

FINA

Financijska agencija



e-Račun, SOKNO, RGFI financijski API-ji

Porezna uprava

Ministarstvo financija – Porezna uprava



ePDV, eDohodak, eKnjiga SOAP API-ji

HZZO

Hrv. zavod za zdravstveno osiguranje



Razmjena podataka o zdravstvenom osiguranju, eRecepti

e-Građani

Platforma digitalnih državnih usluga



Centralna autentikacija + integracija usluga za građane

HZMO

Hrv. zavod za mirovinsko osiguranje



Mirovinski zapisi, API za doprinose poslodavaca

HZZ

Hrv. zavod za zapošljavanje



Podaci o zapošljavanju, API-ji za posredovanje rada



SOAP u svijetu - Veliki korisnici



Amazon AWS

EC2, S3 nasljedni API-ji; SOAP krajnje točke za starije SDK-ove



SAP

SAP NetWeaver, SAP S/4HANA: SOAP za ERP-prema-ERP integraciju



Healthcare (HL7)

FHIR SOAP vezivanje, bolnički sustavi diljem svijeta



Oracle EBS

E-Business Suite WebServices — nabava, HR, financije



Salesforce

SOAP API — puni pristup CRM podacima, još uvijek primarni enterprise API



PayPal (nasljeđe)

Merchant API v1 — SOAP prije migracije na REST



SWIFT Banking

ISO 20022 XML putem SOAP-a — globalna međubankovna plaćanja



Vlada SAD-a

IRS, SSA, VA: nasljedni vladini API-ji i dalje koriste SOAP

03 · WSDL - Web Service Description Language

Struktura · Primjeri · Semantika · Prednosti i nedostaci



WSDL - Uvod i struktura

<types>

XSD definicije tipova podataka za poruke
(ulazne/izlazne)

<binding>

Protokol i format podataka za portType
(SOAP, HTTP, MIME)

<message>

Apstraktna definicija sadržaja poruke
(parametri)

<service>

Kolekcija portova — gdje servis živi (URL-ovi)

<portType>

Apstraktni skup operacija — sučelje servisa (WSDL 1.1)

<port>

Jedna krajnja točka: binding + mrežna adresa (URL)



WSDL 1.1 - Primjer (1/2) Types i Messages

WSDL 1.1 XML

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="HelloWorldService"
  targetNamespace="urn:HelloWorld"
  xmlns:tns="urn:HelloWorld"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns="http://schemas.xmlsoap.org/wsdl/">

  <!-- TIPOVI PODATAKA -->
  <types>
    <xsd:schema targetNamespace="urn:HelloWorld">
      <xsd:element name="sayHelloRequest">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="firstName"
              type="xsd:string"/>
            <xsd:element name="lastName"
              type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:schema>
  </types>

  <!-- PORUKE -->
  <message name="sayHelloRequest">
    <part name="parameters" element="tns:sayHelloRequest"/>
  </message>
  <message name="sayHelloResponse">
    <part name="return" type="xsd:string"/>
  </message>
```



WSDL 1.1 - Primjer (2/2) PortType, Binding i Service

WSDL 1.1 XML

```
<!-- PORT TYPE (apstraktno sučelje) -->
<portType name="HelloWorldPortType">
  <operation name="sayHello">
    <input message="tns:sayHelloRequest"/>
    <output message="tns:sayHelloResponse"/>
  </operation>
</portType>

<!-- BINDING (detalji SOAP protokola) -->
<binding name="HelloWorldBinding"
  type="tns:HelloWorldPortType">
  <soap:binding style="document"
    transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="sayHello">
    <soap:operation soapAction="urn:HelloWorld#sayHello"/>
    <input> <soap:body use="literal"/> </input>
    <o> <soap:body use="literal"/> </o>
  </operation>
</binding>

<!-- SERVICE (URL krajnje točke) -->
<service name="HelloWorldService">
  <port name="HelloWorldPort"
    binding="tns:HelloWorldBinding">
    <soap:address
      location="http://localhost:8080/HelloWorld"/>
    </port>
  </service>
</definitions>
```

WSDL - Prednosti, nedostaci i alternative

✓ Prednosti

✓ Čitljivo strojevima — automatski generira klijentske stub-ove

✓ Precizan ugovor — nema dvosmislenosti u tipovima podataka

✓ Podrška alata: wsimport, SoapUI, Visual Studio

✓ Neovisan o jeziku — Java, .NET, Python

✓ Verzioniranje putem targetNamespace-a

✗ Ograničenja

✗ Izuzetno opširan XML format

✗ Samo strukturalan — nema semantičkog značenja

✗ Teško čitati i pisati ručno

✗ Promjene lome postojeće klijente

✗ Nema ugrađene dokumentacije za čovjeka

Moderne alternative za WSDL

OpenAPI (Swagger) za REST · GraphQL SDL · Protobuf (.proto) za gRPC · AsyncAPI za event-driven arhitekture · JSON Schema za validaciju

04 · Alternativne tehnologije

REST · GraphQL · gRPC · WebSockets · Usporedba



SOAP vs REST - Izravna usporedba

Značajka	SOAP	REST
Vrsta	Protokol (pravila)	Arhitekturni stil
Format podataka	Samo XML	JSON, XML, HTML, plain text
Transport	HTTP, SMTP, JMS, TCP	Samo HTTP/HTTPS
Ugovor sučelja	WSDL (strogi)	OpenAPI (neobavezno)
Sigurnost	WS-Security (ugrađena)	OAuth 2.0, HTTPS (vanjska)
Performanse	Teški (XML overhead)	Lagani, brži
Transakcije	WS-AtomicTransaction 	Nije nativno podržano 
Krivulja učenja	Strma	Niska
Idealan za	Enterprise, bankarstvo, zdravstvo	Web/mobilne API-je, mikroservise



GraphQL - Fleksibilni upitni jezik za API-je

GraphQL upit (traži samo što ti treba)

GraphQL

```
query GetStudent($id: ID!) {  
  student(id: $id) {  
    firstName  
    lastName  
    courses {  
      name  
      grade  
    }  
  }  
}
```

SOAP ekvivalent: dohvatiti CIJELI

objekt studenta + sve relacije!

GraphQL: ti definiraš oblik odgovora



Ključni koncepti GraphQL-a

- Jedna krajnja točka (POST /graphql)
- Klijent kontrolira oblik odgovora
- Nema over-fetching / under-fetching
- Introspection — samoodescriptivan
- Subscriptions za podatke u stvarnom vremenu
- SDL (Schema Definition Language)
- Koriste: Meta, GitHub, Shopify, Twitter

vs SOAP:

- GraphQL: fleksibilni upiti, jednostavnost REST-a + ugovori poput SOAP-a | SOAP: strogi ugovori, bolji za transakcije

⚡ gRPC - Visokoučinkoviti RPC okvir

```
// 1. Definiraj servis u .proto datoteci  
syntax = "proto3";
```

Protocol Buffers
(.proto)

```
service StudentService {  
  rpc GetStudent(StudentRequest)  
    returns (StudentResponse);  
  rpc ListStudents(empty)  
    returns (stream StudentResponse);  
}
```

```
message StudentRequest {  
  string student_id = 1;  
}  
message StudentResponse {  
  string first_name = 1;  
  string last_name = 2;  
  int32 grade = 3;  
}
```

⚡ Zašto gRPC?

- Binarni protokol (Protobuf) — 5–10× manji
- HTTP/2 — multipleksiranje, streaming
- Strogo tipiziran ugovor (.proto)
- Generiranje koda za 10+ jezika
- 4 načina streaminga: unary, server, client, bidirectionalni
- Koriste: Google, Netflix, Square, Dropbox

vs SOAP: gRPC pobjeđuje u performansama i streamingu · SOAP pobjeđuje u nasljednoj podršci i WS-* enterprise značajkama · Oba imaju stroge ugovore

Usporedba: SOAP · REST · GraphQL · gRPC

Aspekt	SOAP	REST	GraphQL	gRPC
Protokol	SOAP/XML	HTTP	HTTP	HTTP/2
Format	XML	JSON/XML	JSON	Protobuf (binarni)
Ugovor	WSDL	OpenAPI	SDL	.proto
Performanse	☆☆	☆☆☆	☆☆☆	☆☆☆☆☆
Sigurnost	WS-Security	OAuth/JWT	OAuth/JWT	TLS/mTLS
Streaming	✗	Ograničen	Subscriptions	✓ Potpun
Nativno u preglednicima	✗	✓	✓	Ograničen
Idealan za	Enterprise nasljeđe	Web API-ji mobilne	Složeno dohvaćanje	Mikroservisi real-time

05 · WS-* standardi i moderni web servisi

2. generacija · 3. generacija · Mikroservisi · Event-driven arhitektura



WS-* - Standardi druge generacije



WS-Security

End-to-end sigurnost SOAP poruka — XML enkripcija, digitalni potpisi



WS-ReliableMessaging

Garantirana dostava s potvrdom i osiguravanjem redoslijeda



WS-Coordination

Čuvanje konteksta distribuiranih transakcija između servisa



WS-Transaction

ACID transakcije (WS-AtomicTransaction) i dugoročne (WS-BusinessActivity)



WS-Policy

Deklariranje zahtjeva servisa: sigurnost, QoS, politike pouzdanosti



BPEL4WS

Business Process Execution Language — orkestrira WS u radne tokove



WS-Attachments

Prijenosi binarnih datoteka putem SOAP-a (MIME/DIME enkapsulacija)



WS-BasicProfile

Smjernice za interoperabilnost — SOAP 1.2 + UDDI 3 + WS-Addressing



Treća generacija web servisa i moderni koncepti



Mikroservisi

Aplikacije podijeljene u neovisne, male servise (Docker, Kubernetes)



Serverless

Functions as a Service — AWS Lambda, Azure Functions, GCP Functions



Event-driven

Apache Kafka, RabbitMQ — asinkrono porukovanje, eventualna konzistentnost



WebSockets

Dvosmjerna komunikacija u stvarnom vremenu: chat, dashboardi, igre



GraphQL

Facebook 2015 — upiti koje pokreće klijent, jedna krajnja točka



gRPC

Google 2016 — Protobuf binarni format, HTTP/2, streaming



OAuth 2.0 + OIDC

Moderni standard autentikacije — tokenski sustav, zamjena WS-Security



OpenAPI 3.x

REST API standard — zamjena za WSDL kod HTTP servisa



Ključni pomak: Od 'integriraj sustave' (SOAP era) → 'komponuj servise' (mikroservisi) → 'streami događaje' (moderne arhitekture)

▶ [Microservices Architecture Explained](#)

▶ [Event-Driven Architecture \(EDA\)](#)

06 · Trendovi i statistike 2025.

Korištenost API-ja · Stope usvajanja · Industrijski podaci

[State of APIs 2025 \(Postman Report\)](#)



Statistike korištenosti API-ja

80%+

API-ja koristi
REST arhitekturu

~8%

API-ja i dalje
koristi SOAP

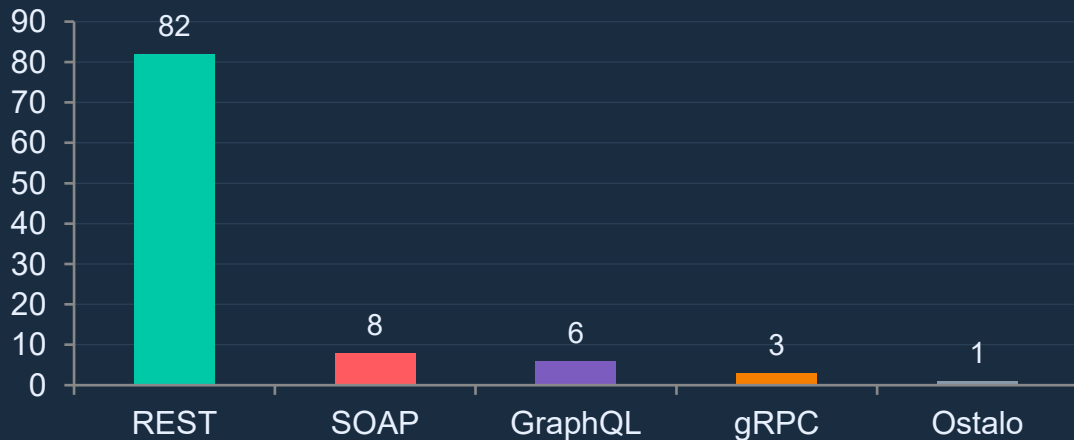
30%+

poduzeća sada
koristi GraphQL

14,2
mlrd \$

tržište upravljanja
API-jima

Usvajanje API protokola (2024.)



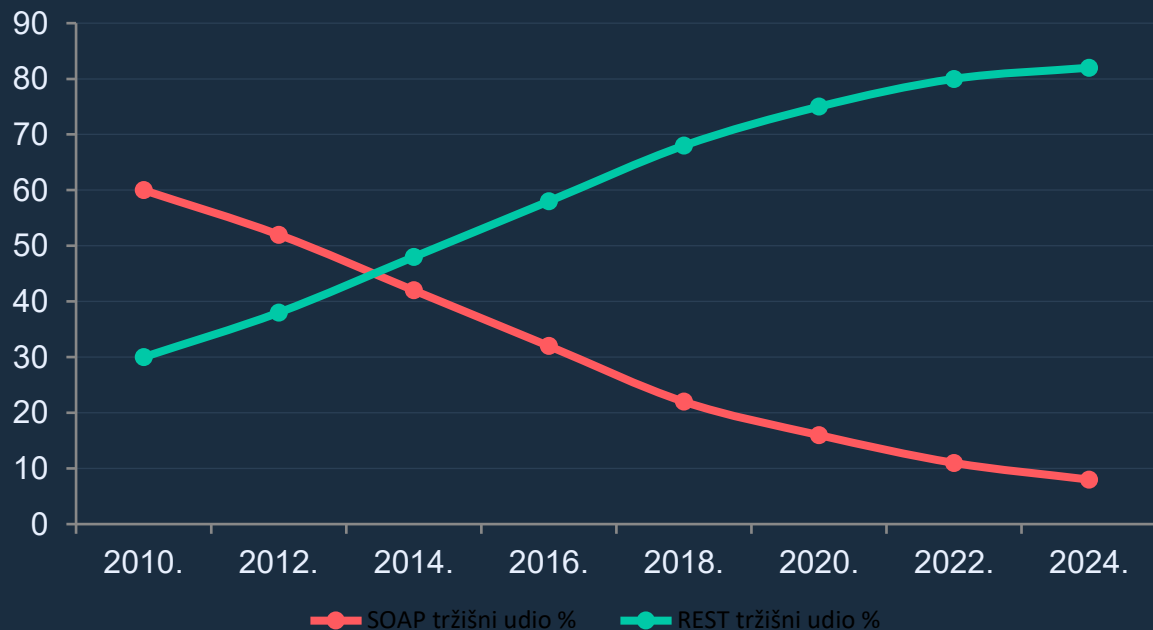
Ključne spoznaje

- REST dominira s 82% (Postman)
- SOAP opada, ali ~8% još aktivni
- GraphQL raste 30% godišnje
- gRPC dobiva na popularnosti u mikroservisima
- 90% aplikacija dnevno koristi API-je
- Sigurnost API-ja — izazov br. 1



Trend usvajanja SOAP-a - Je li SOAP mrtav?

SOAP vs REST usvajanje (2010.–2024.)



🤖 Je li SOAP mrtav?

- NIJE mrtav — samo je specifična niša
- Naslijeđeni sustavi: 20+ god. SOAP API-ji
- Bankarstvo / telekom: neće migrirati brzo
- Zdravstvo: SOAP propisan standardima
- Novi API-ji: 99% bira REST/gRPC
- Hrvatski državni API-ji: još aktivni
- Zaključak: uči SOAP za nasljednu integraciju



Budućnost web servisa - Trendovi 2025.+



AI-pokretani API-ji

LLM API-ji (OpenAI, Anthropic) — streaming REST krajnje točke postaju novi standard



Zero-Trust API sigurnost

mTLS, SPIFFE/SPIRE — svaki servis verificira svaki drugi servis



API-First razvoj

Dizajniraj ugovor (OpenAPI/Proto) prije pisanja koda — fokus na iskustvo razvojnih inženjera



AsyncAPI i događaji

Event-driven arhitektura zamjenjuje req/res — Kafka, NATS, MQTT protokoli



API-ji na rubu mreže

Cloudflare Workers, Deno Deploy — API-ji rade na CDN rubu, latencija ispod 10ms



Web3 i decentralizacija

API-ji pametnih ugovora, IPFS, otkrivanje servisa temeljeno na blockchainu



Resursi za učenje za Gen Z developere



Videozapisi koje morate pogledati

▶ [SOAP Web Services in Java \(in28minutes\)](#)

▶ [REST API Crash Course \(Traversy\)](#)

▶ [GraphQL Full Course \(freeCodeCamp\)](#)

▶ [gRPC & Protobuf \(Hussein Nasser\)](#)



Alati za isprobati

SoapUI

Testiranje SOAP i REST API-ja |
soapui.org

Postman

Dizajn i testiranje API-ja |
postman.com

Insomnia

REST/GraphQL klijent | insomnia.rest

wsimport

Generira Java SOAP stub klase iz
WSDL-a



Službena dokumentacija i reference

- W3C SOAP 1.2 specifikacija — [w3.org/TR/soap12](https://www.w3.org/TR/soap12)
- W3C WSDL 2.0 — [w3.org/TR/wsdl20](https://www.w3.org/TR/wsdl20)
- GraphQL specifikacija — spec.graphql.org
- gRPC dokumentacija — grpc.io/docs
- OpenAPI 3.1 — spec.openapis.org
- Postman State of API 2025. — postman.com/state-of-api



Pitanja

Web servisi · SOAP · WSDL · REST · GraphQL · gRPC

doc.dr.sc. Aleksander Radovan, prof. struč. stud.

Algebra Bernays University · Interoperabilnost informacijskih sustava · 2025./26.

► Web 1.0 vs Web 2.0 and Web
3.0: What's The Difference