



OAuth autorizacijski sustav

Izradili: Ivan Močilac, David Škrnički, Ivana
Mikulić, Marko Galavić, Amar Ademi

Predmet: Šifre i kodovi



Uvod

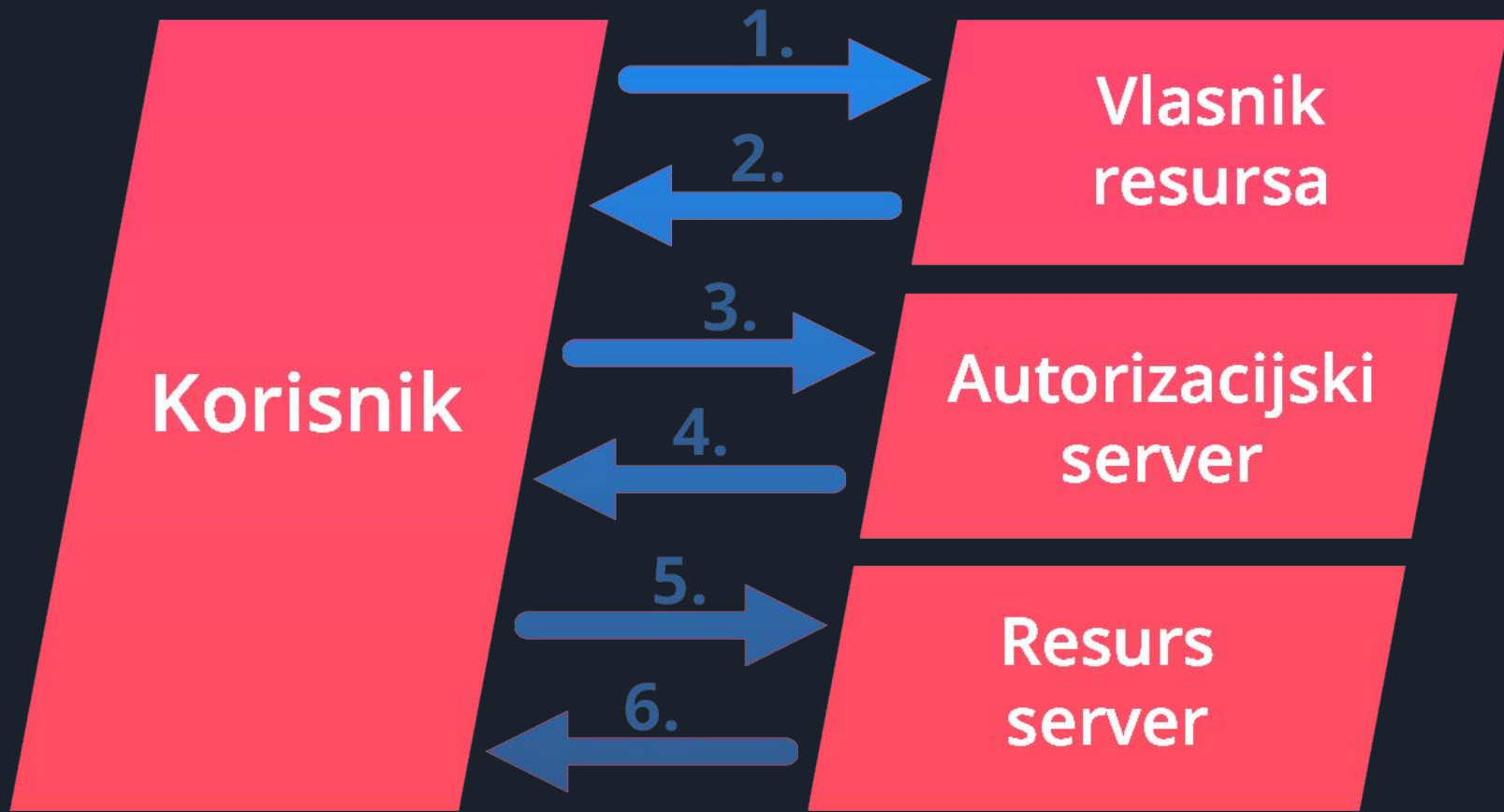


- OAuth je otvoreni standard za autorizaciju koji omogućuje sigurnu razmjenu podataka između aplikacija.
- Kratak povijesni pregled: OAuth je nastao 2010. godine kao odgovor na potrebu za sigurnom razmjenom podataka bez dijeljenja lozinki.
- Svrha i važnost: Omogućava sigurnu i standardiziranu autorizaciju korisnika i aplikacija.



Osnovne komponente OAuth-a

- Korisnik (Resource Owner): Osoba koja posjeduje podatke ili resurse.
- Klijent (Client): Aplikacija koja želi pristupiti resursima u ime korisnika.
- Poslužitelj resursa (Resource Server): Server koji čuva resurse kojima se želi pristupiti.
- Poslužitelj autorizacije (Authorization Server): Server koji autorizira zahtjeve za pristup.





Razlike između OAuth 1.0 i OAuth 2.0

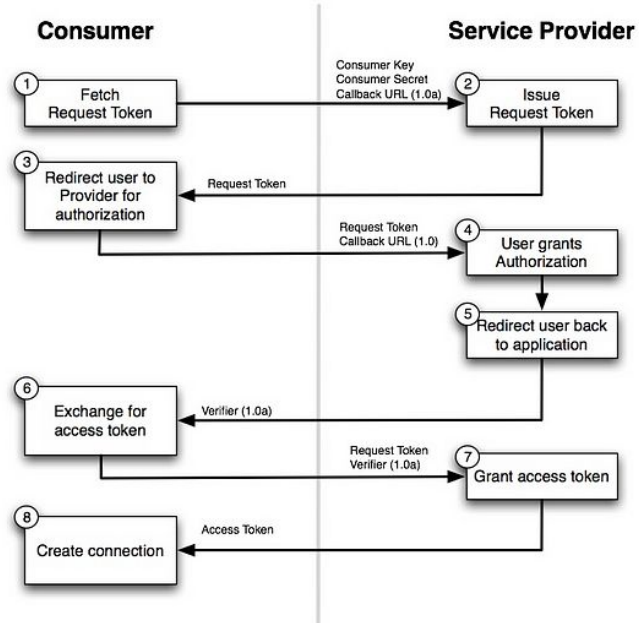
OAuth 1.0

1. **Složena autentifikacija:** Koristi kriptografski potpis za svaki zahtjev, što zahtjeva složenu obradu i potpisivanje.
2. **Trajni tokeni:** Tokeni su trajni i zahtijevaju kriptografski potpis za svaki zahtjev, čime se osigurava visoka razina sigurnosti.
3. **Ograničena fleksibilnost:** Teža implementacija i manje fleksibilnosti za proširenja i modifikacije, čime je manje prilagođen modernim aplikacijama.

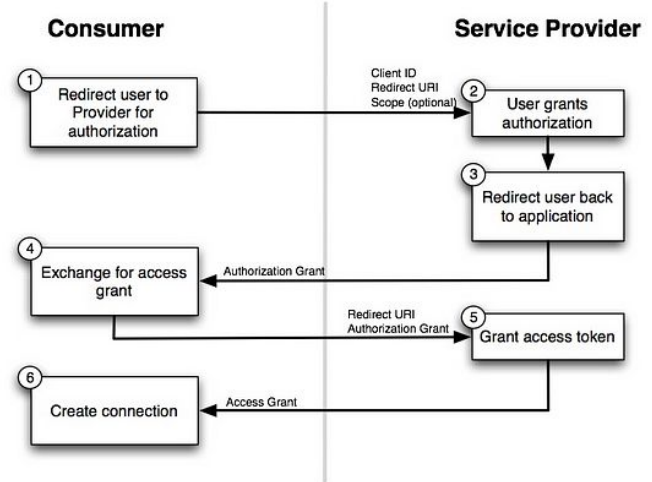
OAuth 2.0

1. **Jednostavnija implementacija:** Koristi HTTPS za sigurnost i pruža različite tokove autorizacije bez potrebe za kriptografskim potpisom svakog zahtjeva.
2. **Kratkotrajni tokeni:** Uvodi kratkotrajne pristupne tokene i mogućnost osvježavanja tokena za dugoročan pristup.
3. **Veća fleksibilnost i proširivost:** Dizajniran s fleksibilnošću za prilagodbu i proširenje, podržava moderne web i mobilne aplikacije.

OAuth 1.0



OAuth 2.0





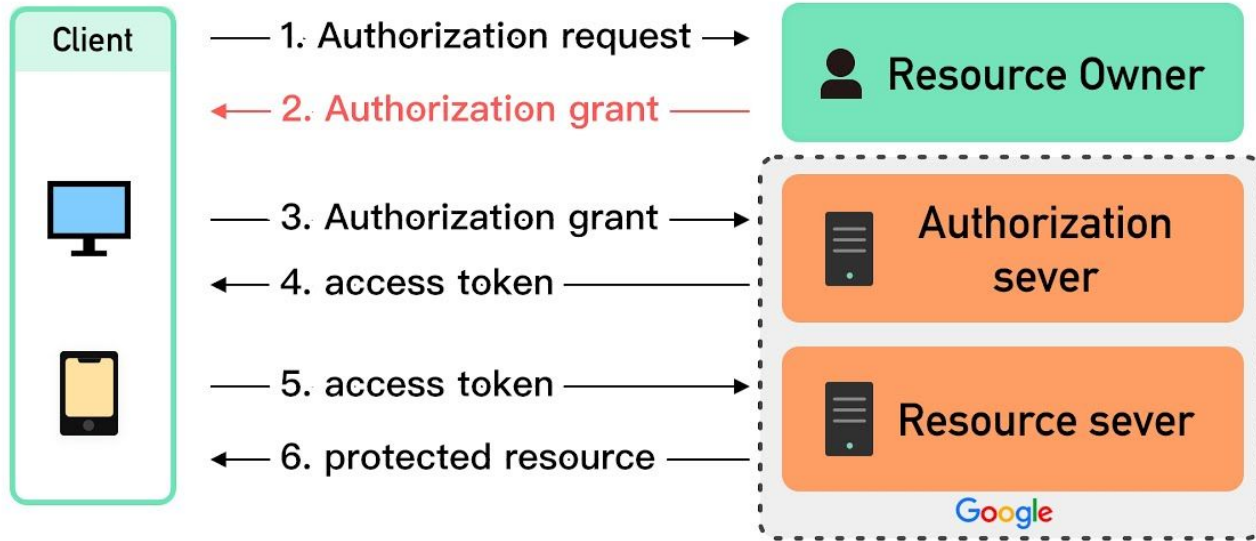
OAuth tokovi autorizacije

- Implicitni tok: Koristi se za javne klijente, bez zahtjeva za osvježavanjem tokena.
- Tok autorizacijskog koda: Koristi kratkotrajne autorizacijske kodove za sigurno dobivanje pristupnih tokena.
- Tok vlasnika resursa: Korisnik daje svoje vjerodajnice za dobivanje pristupa.
- Tok vjerodajnica klijenta: Koristi se za server-server komunikaciju bez sudjelovanja korisnika.



Sigurnosni koncepti OAuth-a

- Tokeni za pristup: Kratkoročni tokeni koji omogućuju pristup resursima bez slanja lozinki.
- Osvježavanje tokena: Omogućuje dulji pristup bez ponovnog autentificiranja korisnika.
- Enkripcija podataka: Osigurava zaštitu podataka tijekom prijenosa i pohrane koristeći moderne enkripcijske protokole.





Implementacija OAuth-a

- Kako implementirati OAuth: Koraci uključuju registraciju aplikacije, konfiguraciju klijenata i autorizacijskog servera, te upravljanje tokenima.
- Koraci implementacije: Definiranje opsega pristupa, postavljanje redirect URI-a, i integracija s postojećim sustavom.
- Najbolje prakse: Korištenje sigurnih pohrana za tokene, redovito reviziranje pristupa i ažuriranje sigurnosnih postavki.



Prednosti OAuth-a

- Povećana sigurnost: Omogućava siguran pristup resursima bez dijeljenja lozinki.
- Fleksibilnost: Podržava različite načine autorizacije prilagodljive različitim scenarijima.
- Smanjena potreba za dijeljenjem lozinki: Korisnici ne trebaju dijeliti svoje lozinke sa aplikacijama.



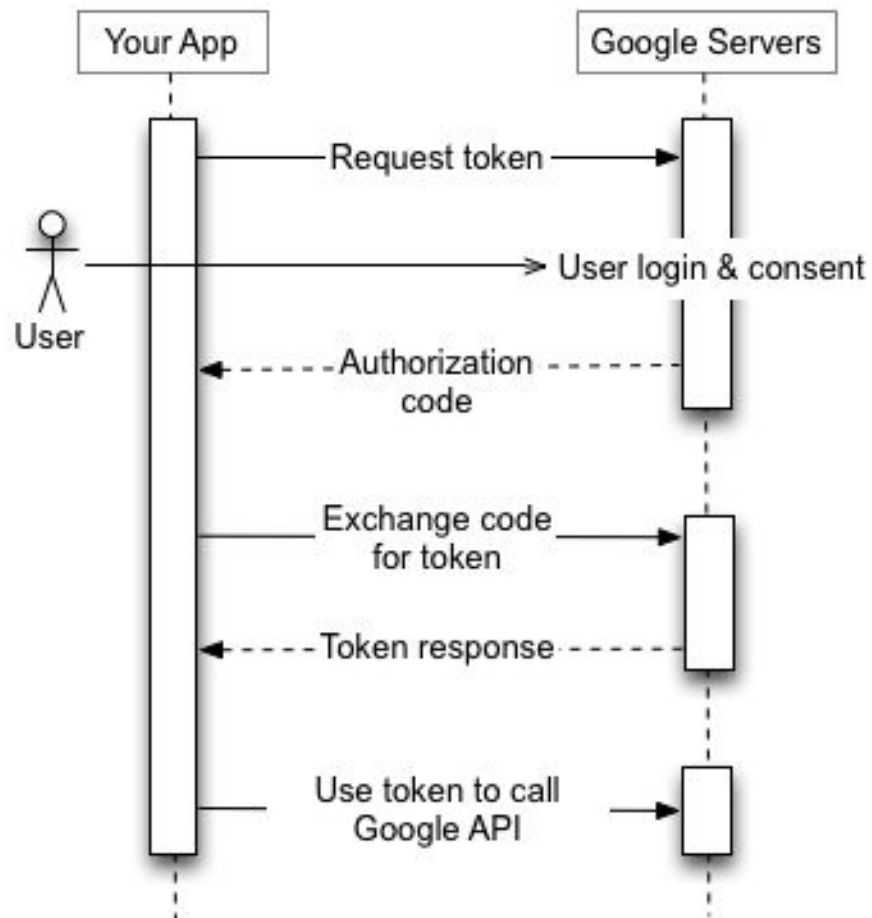
Nedostaci OAuth-a

- Kompleksnost implementacije: Zahtijeva značajno tehničko znanje i resurse.
- Sigurnosni izazovi: Potreba za redovitim ažuriranjima i provjerama sigurnosnih protokola.
- Održavanje: Kontinuirana potreba za održavanjem i ažuriranjem kako bi se osigurala sigurnost sustava.



Primjeri upotrebe OAuth-a

- Google i Facebook integracija: Koriste OAuth za omogućavanje trećih aplikacija pristup korisničkim podacima.
- GitHub OAuth: Omogućava developerima da aplikacije pristupaju njihovim GitHub podacima.
- Drugi primjeri: Salesforce, LinkedIn, i mnogi drugi koriste OAuth za sigurnu autorizaciju.





Napredne značajke i prilagodbe

- Scope: Definicija granularne kontrole pristupa za različite resurse.
- JWT tokeni: Koriste se za prijenos sigurnih informacija između stranaka.
- Korisnička autentifikacija nasuprot autorizaciji: Razlika između provjere identiteta korisnika i autorizacije pristupa resursima.

Zaključak



- **OAuth pruža sigurnu i standardiziranu autorizaciju:** Uspostavlja povjerenje između aplikacija i korisnika, omogućujući siguran pristup podacima bez potrebe za dijeljenjem lozinki.
- **Fleksibilan i prilagodljiv protokol:** S podrškom za različite tokove autorizacije, OAuth je prilagodljiv za razne aplikacije, od jednostavnih mobilnih aplikacija do složenih web servisa.
- **Neophodan za moderne aplikacije:** Zbog jednostavne implementacije i održavanja sigurnosti, OAuth je ključan za razvoj i integraciju modernih aplikacija koje zahtijevaju siguran pristup korisničkim podacima i resursima





Literatura

- Boyd, Ryan. Getting started with OAuth 2.0. " O'Reilly Media, Inc.", 2012.
- Gibbons, Kevin, John O. Raw, and Kevin Curran. "Security evaluation of the OAuth 2.0 framework." Information Management and Computer Security 22.3 (2014): 01-23.

Zahvaljujemo na
pozornosti!

