

UDK 528.9:681.6:004.4

Stručni članak / Professional paper

Open source webGIS tehnologija: kako prikazati prostorne podatke pomoću web karte?

Grgo DŽELALIJA – Zagreb¹, Ivana RACETIN – Split²,
Robert ŽUPAN – Zagreb¹

SAŽETAK. U ovom radu pregledana je tehnologija za razvoj interaktivnih web karata te mogućnosti OpenLayers biblioteke za izradu interaktivnih web karti. OpenLayers je uz Leaflet najpopularnija biblioteka za izradu interaktivnih web karti i naprednih webGIS sustava. Ova biblioteka omogućuje razvoj složenih webGIS aplikacija pružajući opširan set funkcionalnosti poput upravljanja slojevima karte, crtanja i uređivanja geometrijskih objekata, mjerenja udaljenosti ili površina te podršku pri radu u različitim kartografskim sustavima i projekcijama. OpenLayers biblioteka omogućuje jednostavnu prilagodbu ili proširivanje postojećih funkcionalnosti s obzirom na specifične zahtjeve pojedinih projekata, što je čini popularnom pri razvoju prilagođenih i kompleksnih webGIS rješenja.

Cljučne riječi: webGIS, prostorni podaci, OpenLayers.

1. Uvod

Prostorni podaci ključni su za razumijevanje i upravljanje resursima povezanim s prostorom. Prostorni podaci tako omogućuju donošenje odluka u urbanizmu, upravljanju zemljištem, ekologiji, poljoprivredi, transportu, upravljanju i učinkovitom reagiranju u hitnim situacijama te mnogim drugim područjima. Važnost vizualizacije prostornih podataka seže sve od prapovijesti kada su nastajali prvi oblici karata, poput babilonske karte svijeta izrađene na glinenoj pločici (Jacobs 1902).

¹ dr. sc. Grgo Dželalija, Ericsson Nikola Tesla d. d., Krapinska 45, HR-10000 Zagreb, Hrvatska, e-mail: grgo.dzelalija@ericssonnikolatesla.com, gdzelalija@gradst.hr
prof. dr. sc. Robert Župan, Geodetski fakultet Sveučilišta u Zagrebu, Kačićeva 26, HR-10000 Zagreb, Hrvatska, e-mail: robert.zupan@geof.unizg.hr

² prof. dr. sc. Ivana Racetin, Fakultet građevinarstva, arhitekture i geodezije, Sveučilište u Splitu, Matice hrvatske 15, HR-21000 Split, Hrvatska, e-mail: iracetin@gradst.hr

Tehnologija mjerenja i prikaza prostornih podataka razvijala se kroz povijest, te su karte postajale sve preciznije i dostupnije. Danas, u procesu digitalizacije omogućena je pohrana, obrada i diseminacija prostornih podataka putem digitalnih tehnologija. Time digitalizacija omogućuje bržu i učinkovitiju razmjenu prostornih podataka, smanjuje troškove, te povećava dostupnost prostornih podataka svim zainteresiranim strankama.

Razvoj tehnologije tako je omogućio razvoj web karata i vizualizaciju geoprostornih podataka putem interneta. WebGIS danas ima primjenu u različitim područjima, olakšavajući kolaboraciju i upravljanje prostornim podacima. Zajednički informacijski sustav katastra i zemljišne knjige jedno je od ključnih područja u kojem se webGIS tehnologije koriste za pohranu, upravljanje i diseminaciju katastarskim podacima. WebGIS tehnologija koristi se i za evidenciju i upravljanje objektima i vodovima elektroenergetske, elektroničko komunikacije, vodovodne, toplovodne, odvodne, plinovoda i naftovodne infrastrukture koje se vode u hrvatskom katastru infrastrukture (URL 1). Primjena webGIS-a sve je popularnija i u sektoru pametnih gradova za učinkovitije upravljanje prometom, komunalnom infrastrukturom i gradskim resursima. Također, koncept digitalnih blizanaca često koristi webGIS tehnologiju za stvaranje virtualnih modela gradova ili infrastrukturnih sustava koji omogućuju simulaciju i optimizaciju različitih scenarija (Tiwari i Jain 2017).

Open source tehnologije ključne su u razvoju posebno javnih projekata zbog svoje pristupačnosti i fleksibilnosti. Open source softver pruža slobodu prilagodbe i proširivosti, ima velike baze korisnika te je pristupačan što omogućuje jednostavnije financiranje i održavanje. Open source tehnologije tako se često koriste i pri razvoju webGIS sustava.

2. Open source tehnologija i interaktivne web karate

U Svijetu uporaba open source tehnologija za razvoj interaktivnih web karata i GIS aplikacija raste. Biblioteke poput Leafleta i OpenLayersa najpopularnije su JavaScript biblioteke za prikaz i manipulaciju geoprostornih podataka. Leaflet je objavljen 2011. godine te je brzo postao jedan od najpopularnijih alata za izradu interaktivnih web karata, ponajviše zbog svoje jednostavnosti i malog obujma koda. Najčešće se koristi za izradu brzih i jednostavnih web karata. OpenLayers je objavljen 2006. godine, a razvijen je za naprednije funkcionalnosti web karata poput rada s kartografskim projekcijama (URL 2). Najčešće se koristi za razvoj složenijih webGIS aplikacija, kao što su katastar infrastrukture ili Zajednički informacijski sustav katastra i zemljišne knjige (URL 3).

Za prikaz trodimenzionalnih prostornih podataka najčešće se koriste open source biblioteke poput CesiumJS-a i Three.js-a koje nude napredne mogućnosti vizualizacije 3D prostornih podataka. CesiumJS razvijen je 2011. godine te je prvi open source alat koji omogućuje vizualizaciju globusa u web preglednicima renderiranjem pomoću WebGL-a. S druge strane, three.js također koristi WebGL tehnologiju ali se koristi za vizualizaciju 3D scena i geometrija.

Kada su u pitanju 3D podaci često se koriste i biblioteke D3.js i Deck.gl. Bib-

lioteka D3.js razvijena je 2011. godine te omogućuje izradu interaktivnih vizualizacija manipulacijom DOM (engl. Document Object Model) elemenata. Koristi se najčešće za izradu jednostavnijih i specifičnih vizualizacija prostornih podataka. Deck.gl razvila je kompanija Uber Engineering, također temeljen na WebGL-u zbog potrebe za analizom i praćenjem kretanja vozila u stvarnom vremenu.

Kod razvoja webGIS aplikacija koristi se još i biblioteka Turf.js koja omogućuje prostorne analize i izvršavanje geometrijskih operacija izravno u web pregledniku, poput računanja udaljenosti, unija ili presjeka geometrija i slično. Ovim tehnologijama moguće je pokriti većinu zahtjeva koji se traže od modernih webGIS sustava.

3. Izrada interaktivnih web karata

Moderne web aplikacije trebaju biti responzivnog dizajna, brze i interaktivne. Danas takve aplikacije najčešće koriste SPA arhitekturu (engl. Single Page Application). Takva arhitektura omogućuje da se podaci i sučelje učitavaju dinamički bez osvježavanja cijele web stranice. Među najpopularnijim alatima za izradu takvih web aplikacija su React i Angular.

React je JavaScript biblioteka koju je razvila kompanija Meta fokusirana na izradu korisničkih sučelja idealnih za izradu dinamičnih aplikacija poput društvenih mreža ili aplikacija nadzornih ploča. Angular, koji je razvila kompanija Google, je razvojni okvir temeljen na TypeScriptu te nudi robustan sustav za izradu velikih poslovnih aplikacija. Obje tehnologije omogućuju razvoj modernih web aplikacija (URL 4). U ovom radu dat ćemo primjer izrade interaktivne web karte pomoću Angular razvojnog okvira i OpenLayers biblioteke.

3.1. Preglednik i slojevi karte

Prvi korak u kreiranju web karte pomoću biblioteke OpenLayers je instanciranje klase *Map*. Kako bi se instancirao ovaj objekt potrebno je definirati parametre *target*, *layers* i *view*. Parametar *target* definira HTML element u koji će biti ugrađena OpenLayers karta, dok pomoću *view* objekta definiramo svojstva karte poput projekcije, centra prikaza karte ili početnog mjerila karte. Parametar *target* možemo definirati pomoću identifikatora HTML elementa, dok parametar *view* definiramo instanciranjem klase *View* (slika 1).

```
1. const map = new Map({  
2.   target: 'map',  
3.   view: new View({  
4.     center: [EASTING, NORTHING],  
5.     zoom: ZOOM_LEVEL  
6.   })  
7. });  
8.
```

Slika 1. Instanciranje karte.

Kartu je često potrebno prikazati u nekoj određenoj kartografskoj projekciji, a ako ne definiramo kartografsku projekciju, zadana će biti Web Mercator, odnosno EPSG:3857. Kartografsku projekciju možemo definirati pomoću paketa proj4 (slika 2).

```

1. proj4.defs(
2.   'EPSG:3765',
3.   '+proj=tmerc +lat_0=0 +lon_0=16.5 +k=0.9999 +x_0=500000 +y_0=0 +ellps=GRS80
+towgs84=0,0,0,0,0,0,0 +units=m +no_defs'
4. );
5. register(proj4);
6. const projection = getProjection('EPSG:3765');
7. if (!projection) {
8.   throw new Error('Projection EPSG:3765 is not registered.');
```

Slika 2. Dodavanje HTRS96/TM kartografske projekcije na kartu.

Na kartu možemo dodati slojeve karte. OpenLayers podržava više različitih vrsti slojeva. Podržani su vektorski i rasterski slojevi. Za rasterske slojeve postoje tri klase:

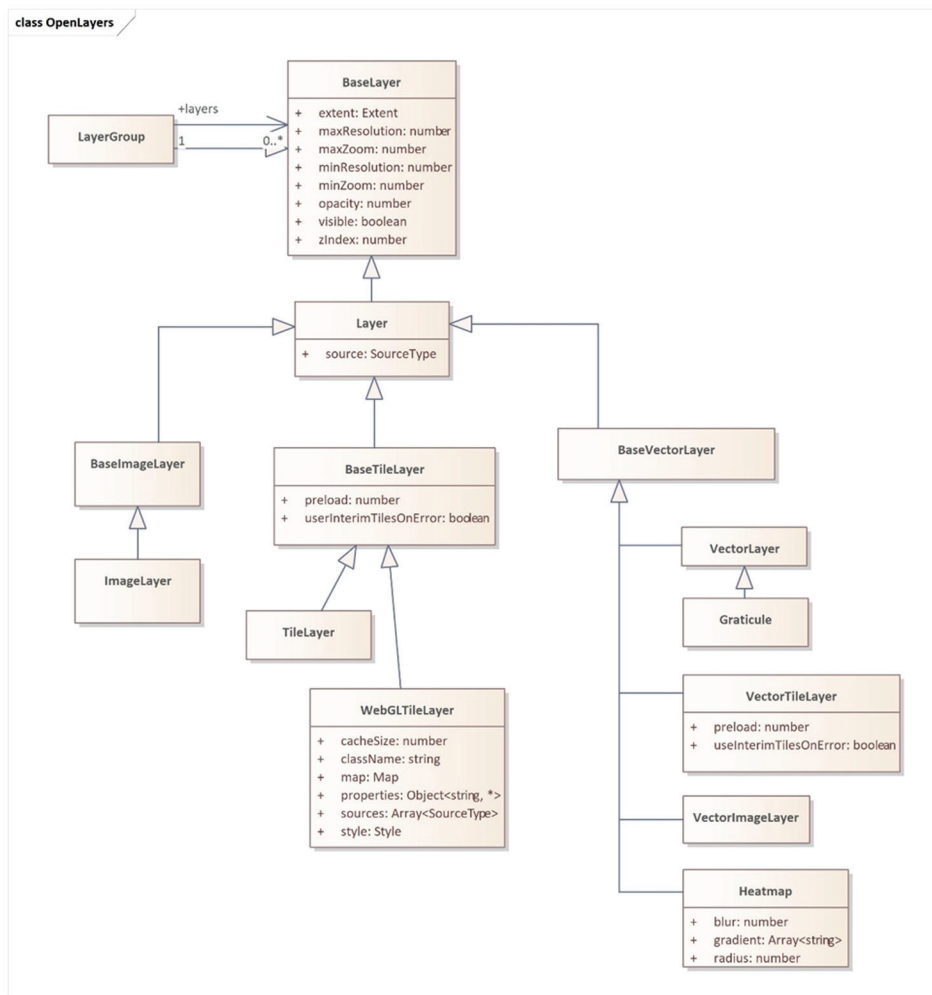
- Klasu `ol/layer/Tile~TileLayer` se koristi za definiranje sloja čiji izvor podataka su serverski renderirane slike organizirane u grupe po mjerilima za specifične rezolucije karte.
- Klasu `ol/layer/Image~ImageLayer` se koristi za definiranje sloja čiji izvor podataka je jedna serverski renderirana slika za bilo koju zatraženu rezoluciju i obim.
- Klasu `ol/layer/WebGLTile~WebGLTileLayer` se koristi za napredni način prikaza rasterskog sloja izravno koristeći mogućnosti WebGL-a za bolje performanse.

Za vektorske podatke prepoznamo pet različitih klasa:

- Klasu `ol/layer/Vector~VectorLayer` se koristi za definiranje sloja vektorskih podataka. Ovaj sloj se renderira na klijentu iz vektorskih podataka te tako daje najprecizniji prikaz.
- Klasu `ol/layer/VectorTile~VectorTileLayer` se koristi u slučaju velikog seta podataka. Vektorski podaci dijele se u manje skupine te renderiraju u manje slike.
- Klasu `ol/layer/VectorImage~VectorImageLayer` također se koristi kod prikazivanja velikih vektorskih slojeva radi boljih performansi, međutim nudi lošiju preciznost od `VectorLayer`-a.
- Klasu `ol/layer/Heatmap~Heatmap` koristi se za renderiranje vektorskih

podataka u obliku heat-mape.

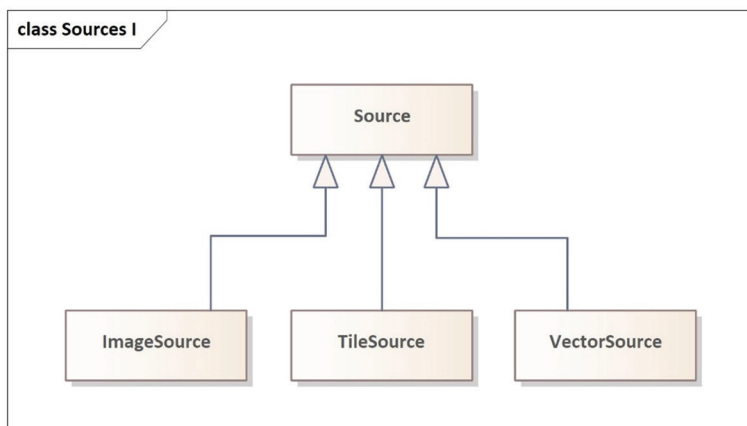
- Klasu `ol/layer/Graticule~Graticule` koristi se za prikaz kartografske mreže.



Slika 3. OpenLayers klase za definiranje slojeva karte.

Za svaki sloj potrebno je definirati njegov izvor, tj. izvor podataka koji će se prikazati na sloju karte. Izvori mogu biti datoteke različitih formata ili web servisi poput WMS (engl. Web Map Service) ili WFS (engl. Web Feature Service) servisa. OpenLayers klase za definiranje izvora podataka dijele se na tri vrste izvora: rasterski izvori podataka s jednom slikom, rasterski izvori podataka s matricom slika i vektorski izvori podataka. Prikaz sloja može biti jedna slika, pri čemu se koriste koriste podklase klase `ImageSource`, ili skup manjih slika, pri čemu će se koristiti podklase klase `TileSource`, dok se za prikaz vektorskih

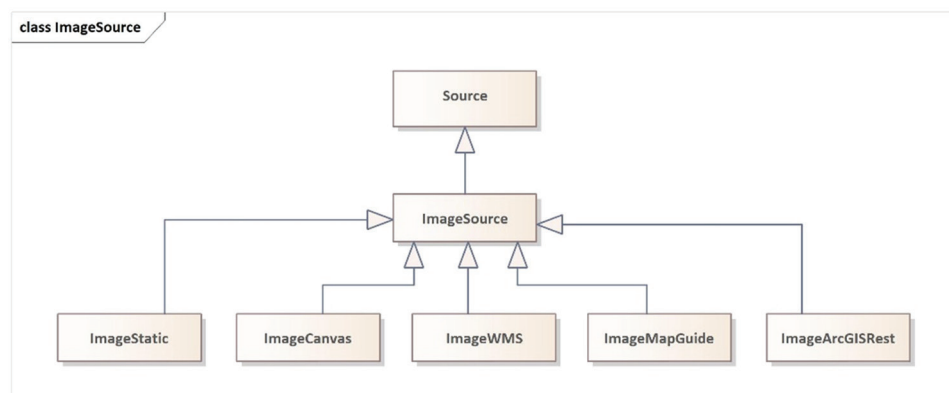
podataka koji se renderiraju na klijentu koriste podklase klase VectorSource (slika 4).



Slika 4. Temeljne klase izvora podataka.

OpenLayers biblioteka omogućuje prikaz ne-georeferenciranih i georeferenciranih slika, te jednostavni prikaz najčešće korištenih servisa za dohvat slika poput WMS-a primjenom različitih klasa (slika 5). Za definiranje izvora podataka rasterskih slojeva s jednom slikom za cijeli prikaz tako prepoznamo sljedeće klase:

- Klasu `ol/source/ImageStatic~Static` koristi se za definiranje za prikaz jedne statične slike. Ovu klasu koristiti se za dodavanje ne-georeferenciranih slika na kartu. Potrebno je definirati prostorni obuhvat i željenu kartografsku projekciju.
- Klasa `ol/source/ImageCanvas~ImageCanvasSource` je temeljna klasa za izvore podataka kojima je element HTML platna slika.
- Klasa `ol/source/ImageWMS~ImageWMS` koristi se za definiranje rasterskih izvora podataka WMS servisa koji servira jednu georeferenciranu sliku s proizvoljnim prostornim obuhvatom. Ovim pristupom renderiranje zatražene slike na serveru je sporije, ali s browsera se upućuje samo jedan upit.
- Klasu `ol/source/ImageMapGuide` koristi se za rasterski izvor podataka s MapGuide servera koji servira jednu georeferenciranu sliku.
- Klasu `ol/source/ImageArcGISRest` koristi se za rasterski izvor podataka s ArcGIS Rest servisa za jednu georeferenciranu sliku.



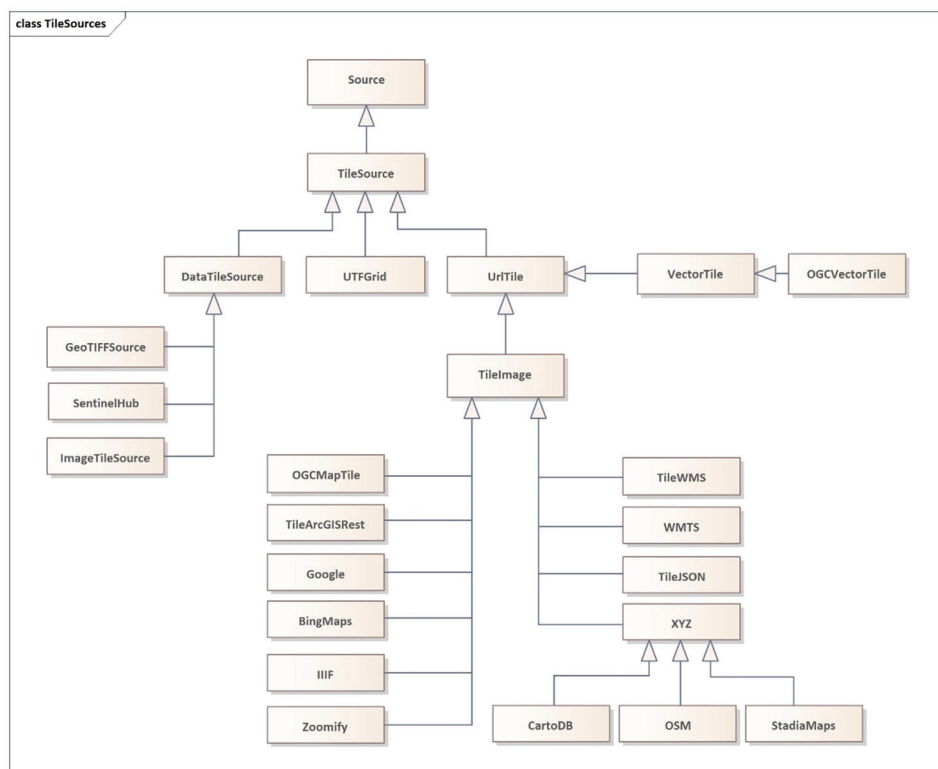
Slika 5. Dodavanje sloja na kartu.

Na kartama često treba prikazati velike količine podataka visoke rezolucije zbog čega renderiranje i prijenos cijelog prikaza karte u jednu sliku može biti zahtjevno. Podaci se zbog toga često dijele u skupine manjih slika podijeljenih s obzirom na rezoluciju i mjerilo prikaza. Vrste izvora podataka, te načini podjele i dohvaćanja takvih slika, se razlikuju pa tako OpenLayers ima više klasa za različite primjene (slika 6):

- Klasu `ol/source/VectorTile~VectorTile` koristi se za vektorske izvore podataka podijeljenih u matricu. U slučajevima kada je potrebno prikazati veliku količinu vektorskih podataka radi boljih performansi, podatke je moguće podijeliti u skupine podijeljene po prostornom obuhvatu te ih pojedinačno dohvaćati i renderirati na klijentu. Ovaj pristup rezultira manjom preciznosti od uobičajenog vektorskog prikaza te nije zamišljen za uređivanje geometrije prikazanih objekata.
- Klasa `ol/source/WMTS~WMTS` koristi se za definiranje izvora podataka s WMTS servisa. WMTS servis sastoji se od unaprijed serverski renderiranih slika podijeljenih u predefinirane skupine po prostornom obuhvatu, koordinatnim sustavima, rezoluciji i mjerilima. Zbog toga prilikom definiranja WMTS izvora podataka, potrebno je pročitati `GetCapabilities` dokument WMTS servisa te odabrati odgovarajući set slika.
- Klasa `ol/source/TileJSON~TileJSON` koristi se za servise poput WMTS-a, uz jednostavniju verziju zapisa informacija o podjeli georeferenciranih slika od WMTS `GetCapabilities` dokumenta.
- Klasu `ol/source/XYZ~XYZ` koristi se za definiranje rasterskih izvora podataka u XYZ formatu koji prati Googleov rasterski format slika.
- Klasu `ol/source/TileWMS~TileWMS` koristi se za izvor podataka WMS servisa. WMS servisi vraćaju georeferencirane slike manjih dimenzija renderirane na serveru. Za razliku od WMTS servisa prostorni obuhvat slika je proizvoljan.
- Klasu `ol/source/ImageTile~ImageTileSource` koristi se za dohvaćanje slika grupiranih po rezoluciji, mjerilu i koordinatnom sustavu. Ovu klasu upotre-

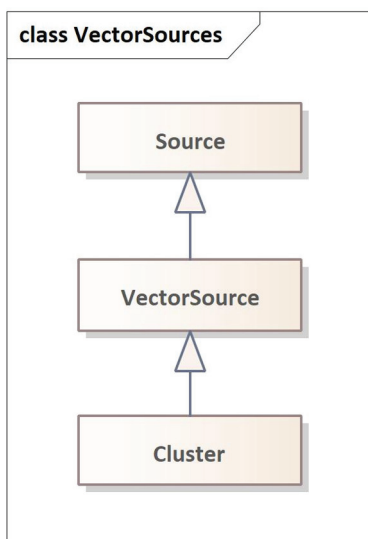
bljava se ako ne želimo koristiti neki od uvrješjenijih pristupa poput XYZ ili TMS formata URL-a za dohvaćanje pojedinih slika iz matrice.

- Klasu `ol/source/Zoomify~Zoomify` koristi se za definiranje izvora podataka u Zoomify formatu. Zoomify format koristi se za serviranje ne-georeferenciranih slika visokih rezolucija.
- Klasu `ol/source/GeoTIFF~GeoTIFFSource` koristi se za definiranje raster-skih izvora podataka GeoTIFF datoteka.
- Klasu `ol/source/Google~Google` koristi se za definiranje izvora podataka s Googleovog Map Tiles API servisa.
- Klasu `ol/source/CartoDB~CartoDB` koristi se za definiranje izvora podataka s CartoDB Maps API servisa.
- Klasu `ol/source/BingMaps~BingMaps` koristi se za definiranje izvora podataka s Bing Maps servisa.
- Klasu `ol/source/StadiaMaps~StadiaMaps` koristi se za definiranje izvora podataka s Stadia Maps tile servisa.
- Klasu `ol/source/SentinelHub~SentinelHub` koristi se za definiranje izvora podataka korištenjem Sentinel Hub API-a.
- Klasu `ol/source/TileArcGISRest~TileArcGISRest` koristi se za definiranje izvora podataka s ArcGIS Rest servisa.
- Klasu `ol/source/OGCMapTile~OGCMapTile` koristi se za definiranje izvora podataka s OGC API Tiles servisa.
- Klasu `ol/source/OGCVectorTile~OGCVectorTile` koristi se za definiranje izvora podataka s OGC API Tiles servisa. Za razliku od prethodne, ova klasa koristi se kada su izvorišni podaci u vektorskog formatu, te se prije dodavanja na kartu renderiranju u slike.
- Klasu `ol/source/OSM~OSM` koristi se za definiranje izvora podataka s OSM servisa.
- Klasu `ol/source/IIIF~IIIF` koristi se za definiranje izvora podataka koji koriste IIIF (engl. International Image Interoperability Format) API servis. Ovi servisi koriste se za posluživanje ne-georeferenciranih slika visokih rezolucija.



Slika 6. Klase izvora podataka za skupine slike.

Za definiranje izvora podataka vektorskih slojeva prepoznavamo klase VectorSource i Cluster (slika 7). Klasa VectorSource podržava gotovo sve korištene formate vektorskih podataka, pa se pomoću nje može definirati izvor podataka u različitim JSON (engl. JavaScript Object Notation) formatima poput EsriJSON, GeoJSON i TopoJson formata. Podržani su i drugi najčešće korišteni formati bazirani na XML-u (engl. Extensible Markup Language), poput GML (engl. Geography Markup Language), GPX (engl. GPS Exchange Format) i KML (engl. Keyhole Markup Language) formata. Uz XML i JSON formate, podržani su i tekstualni formati poput WKT-a (engl. Well-Known Text), binarni formati WKB (engl. Well-Known Binary) i EWKB (engl. Extended WKB) te Mapboxov MVT format. Klasa Cluster koristi se za klasteriranje vektorskih podataka.



Slika 7. *OpenLayers* klase za definiranje vektorskih izvora podataka.

Slojeve se može dodati na kartu instanciranjem neke od klasa za definiranje slojeva i izvora podataka te korištenjem metode *addLayer* klase *Map*. Na slici 8 dodan je rasterski *OpenStreetMap* sloj na kartu. Na ovaj način sloj je dodan na kartu te pohranjen u listu slojeva gdje je njima moguće upravljati, na primjer, mijenjajući redoslijed tako dodanih slojeva.

```
1. map.addLayer(new TileLayer({
2.   source: new OSM()
3. }));
4.
```

Slika 8. Dodavanje sloja na kartu.

Ukoliko na kartu želimo dodati samo privremeni sloj, preporučeno je koristiti metodu *addMap* neke od klasa za definiranje slojeva. Na ovaj način sloj se prikazuje kao posljednji sloj karte te karta njime ne upravlja.

3.2. Dohvaćanje informacija o featureu

Interaktivne web karte često trebaju imati mogućnost prikaza dodatnih podataka o pojedinom prikazanom objektu na karti. U slučaju rasterskih slojeva, osim same vrijednosti piksela ostale podatke, ako postoje, moguće je dohvatiti samo dodatnim pozivom na izvorni servis, a ukoliko se radi o servisu koji nema unaprijed renderirane i slike pohranjene u predmemoriju. Najčešće korišten takav izvor podataka je WMS servis. Dodatni atributni podaci o objektu na određenom pikselu kod WMS servisa dohvaćaju se konstruiranjem *GetFea-*

tureInfo requesta. TileWMS i ImageWMS klase sadrže metodu `getFeatureInfoUrl` koja za pojedini izvor konstruira URL za dohvat atributnih podataka objekta na traženoj lokaciji (slika 9).

```
1. wmsLayer
2.   .getSource()
3.   .getFeatureInfoUrl(evt.coordinate, viewResolution, COORDINATE_SYSTEM, {
4.     INFO_FORMAT: 'application/json',
5.   });
6.
```

Slika 9. Konstruiranje URL-a za dohvaćanje atributnih podataka objekta na karti.

Kod vektorskih slojeva atributni podaci već su dostupni na klijentu. U tom slučaju koristi se interakcija karte koja omogućuje označavanje objekata na karti. Interakcija se kreira instanciranjem klase `ol/interaction/Select~Select` te njenim dodavanjem na kartu (slika 10).

```
1. let selectInteraction = new Select({
2.   layers: [wmsLayer],
3.   style: searchLayerStyleFunction,
4. });
5. map.addInteraction(this.selectInteraction);
6.
```

Slika 10. Dodavanje interakcije označavanja na kartu.

Interakciju označavanja moguće je podesiti ovisno o potrebama postavljanjem atributa pri inicijalizaciji klase `Select`. Atributima *condition*, *toggleCondition*, *addCondition* i *removeCondition* moguće je definirati uvjete kada će se objekti dodati ili izbaciti iz liste označenih objekata. Atributom *features* moguće je definirati listu u koju će se pohranjivati označeni objekti, ukoliko se ovaj atribut ne popuni kreirat će se interna lista interakcije. Atributom *layers* može se definirati slojeve koje je moguće označavati, a ako se vrijednost ne definira označavat će se objekti iz svih slojeva dodanih na kartu. Atributom *style* može se definirati stil označenih objekata kako bi ih se na karti moglo vizualno razlikovati od ostalih. Postavljanjem vrijednosti *true* ili *false* na atribut *multi* definira se hoće li interakcija omogućavati označavanje više objekata ili samo jednog. Atributom *filter* moguće je filtrirati objekte za koje ćemo dozvoliti označavanje. Konačno, atributom *hitTolerance* definiramo radijus tolerancije za označavanje objekata.

3.3. Uređivanje geometrije

Sve češće se atributni i prostorni podaci uređuju i putem web preglednika, pa tako današnji webGIS sustavi trebaju imati alate za uređivanje geometrije. OpenLayers omogućuje izradu takvih alata pomoću interakcija. Klasa `Draw` omogućuje definiranje interakcije za crtanje točaka, linije i poligona označavanjem lomnih točaka na karti (slika 11).

```
1. let drawInteraction = new Draw({  
2.   source: wfsLayer.getSource(),  
3.   type: 'LineString',  
4.   style: createStyle(),  
5. });  
6. map.addInteraction(drawInteraction);  
7.
```

Slika 11. Dodavanje interakcije crtanja linija na kartu.

Na ovaj način geometrijski se objekti na karti crtaju pozicioniranjem pokazivača i klikom na kartu. Ovaj pristup je zadovoljavajući za neke uporabe poput prijave nekog događaja, štete ili potrebne intervencije na gradskim aplikacijama. Neke primjene međutim zahtijevaju precizan unos lomnih točaka na određenim koordinatama. Za tu primjenu koristi se metoda *appendCoordinates* klase Draw pomoću koje se mogu unositi lomne točke definiranjem koordinata (slika 12).

```
1. this.interaction.appendCoordinates([  
2.   [eCoordinate, nCoordinate],  
3.   ]);  
4.
```

Slika 12. Precizno dodavanje lomnih točaka.

Za uređivanje geometrije već postojećih vektorskih objekata koristi se interakcija Modify. Uporabom ove interakcije korisnici mogu pomicati, dodavati ili brisati lomne točke geometrija. Klasa Modify inicijalizira se na sličan način kao i interakcija za crtanje geometrija. Ove dvije interakcije temelj su za aplikacije u kojima korisnici trebaju mogućnost izravnog uređivanja prostornih podataka na intuitivan način.

4. Zaključak

Digitalizacija prostornih podataka omogućuje brzo i precizno prikupljanje, analizu i vizualizaciju prostornih podataka, pa je stoga ključna za donošenje informiranih i pravovremenih odluka u različitim sektorima modernog života. Digitalizirani prostorni podaci olakšavaju njihovu primjenu i integraciju s modernim tehnologijama poput IoT-a i umjetne inteligencije.

Razvoj tehnologije donosi sve veće mogućnosti za upravljanje prostornim podacima i razvoj webGIS sustava. Tako je omogućena jednostavnija integracija prostornih podataka s različitim sustavima, te su oni dostupniji za analizu i vizualizaciju u stvarnom vremenu različitim korisnicima. Budući razvoj naprednih tehnologija poput umjetne inteligencije, strojnog učenja te cloud platformi obećava da će GIS sustavi postati sve moćniji i važniji u rješavanju različitih prostornih izazova.

Literatura

- Jacobs, J. (1902): The Story of Geographical Discovery, D. Appleton and Co, New York, SAD.
- Tiwari, A., Jain K. (2017): Concepts and Applications of Web GIS, Nova Scotia Publishers, New York, SAD.

Mrežne adrese

- URL 1: Jedinstvena informacijska točka – Sustav katastra infrastructure, <https://ski.dgu.hr/gis/startup>, (10. 1. 2025.).
- URL 2: OpenLayers API Docs, <https://openlayers.org/en/latest/apidoc/>, (15. 2. 2025.).
- URL 3: Zajednički informacijski sustav zemljišnih knjiga i katastra, <https://oss.uredjenazemlja.hr/>, (12. 1. 2025.).
- URL 4: AngularJS, <https://angularjs.org/>, (5. 1. 2025.).

Open Source webGIS Technology: How to Display Spatial Data Using a Web Map?

ABSTRACT. This paper reviews the technology for the development of interactive web maps and the capabilities of the OpenLayers library for creating interactive web maps. Along with Leaflet, OpenLayers is the most popular library for creating interactive web maps and advanced webGIS systems. This library enables the development of complex webGIS applications by providing an extensive set of functionalities such as managing map layers, drawing and editing geometric objects, measuring distances or areas, and supporting work in different map systems and projections. The OpenLayers library enables simple adaptation or expansion of existing functionalities with regards to the specific requirements of individual projects, which makes it popular for the development of customized and complex webGIS solutions.

Keywords: webGIS, spatial data, OpenLayers.

Primljeno / Received: 2025-03-07

Prihvaćeno / Accepted: 2025-03-28