

**SVEUČILIŠTE U ZAGREBU
GEODETSKI FAKULTET**

Sergiy Kharchenko

SERVIS OGC-a ZA TRANSFORMACIJU KOORDINATA

Diplomski rad

Zagreb, 2012

I. Autor

Ime i prezime: Sergiy Kharchenko

II. Diplomski rad

Naslov: Servis OGC-a za transformaciju koordinata

Mentor i voditelj: doc. dr. sc. Dražen Tutić

III. Ocjena i obrana

Datum zadavanja zadatka: 23. 01. 2012.

Datum obrane: 29. 06. 2012.

Sastav povjerenstva pred kojim je branjen diplomski rad:

1. Doc. dr. sc. Dražen Tutić
2. Prof. dr. sc. Nada Vučetić
3. Doc. dr. sc. Ivka Kljajić

ZAHVALA:

Zahvaljujem mentoru doc. dr. sc. Draženu Tutiću na predloženoj temi, stručnoj pomoći i savjetima tijekom izrade diplomskog rada.

Zahvaljujem i svim djelatnicima Geodetskog fakulteta Sveučilišta u Zagrebu koji su svojim radom pomogli mi proći ovaj put.

Hvala svim kolegama i prijateljima koji su mi uljepšali studentske dane.

Na kraju, najveće hvala mojoj obitelji na potpori i razumijevanju..

SERVIS OGC-a ZA TRANSFORMACIJU KOORDINATA

Sergiy Kharchenko

Sažetak.

U ovom radu se opisuje upravljanje geoprocima na web-poslužitelju u okviru implementacije standarda Open GIS konzorcija za web-servise za obradu podataka. Praktični projekt opisuje izradu web-servisa za transformaciju koordinata koji je pisan u programskom jeziku Python.

Ključne riječi. Open GIS konzorcij (OGC), web-servis za obradu podataka (WPS), pyWPS, transformacija koordinata

Abstract.

This paper describes the management of geospatial web server in the implementation of Web Processing Service specified by Open GIS Consortium.. Practical project describes the development of service for the transformation of coordinates written in the programming language Python.

Keywords. Open GIS Consortium (OGC), Web Processing Service (WPS), pyWPS, coordinate transformation

SADRŽAJ

1. UVOD	7
2. TEORIJSKE POSTAVKE I TEHNOLOGIJE	8
2.1. MREŽNI SERVISI	8
2.2. XML (EXTENDED MARKUP LANGUAGE)	8
2.3. APACHE	9
2.4. PYTHON	9
3. STANDARD OGC-A ZA WPS	11
3.1. OPĆENITO O OGC-U	11
3.2. ŠTO JE WPS?	11
3.2.1. <i>GetCapabilities</i>	12
3.2.2. <i>DescribeProcess</i>	13
3.2.3. <i>Tipovi ulaznih i izlaznih podataka</i>	14
3.2.3.1. <i>LiteralData</i>	14
3.2.3.2. <i>ComplexData</i>	16
3.2.3.3. <i>BoundingBoxData</i>	17
3.2.4. <i>Execute</i>	18
3.2.5. <i>Zahtjev Execute u obliku KVP</i>	19
3.2.6. <i>Zapis složenog podatka u XML-u</i>	20
3.2.7. <i>XML Execute</i>	25
3.2.8. <i>Dokument odgovora</i>	27
3.2.9. <i>Lineage</i>	28
3.2.10. <i>Reference</i>	29
4. IMPLEMENTACIJA WPS-A U WINDOWS-U	31
4.1. INSTALACIJA APACHEA	31
4.2. INSTALACIJA PYTHON-A	34

4.2.1. Instalacija interpretera	34
4.2.2 Instalacija biblioteka Pythona	38
4.3. INSTALACIJA PYWPS-A	39
4.4. KONFIGURACIJA PYWPS-A	42
4.5. PRIMJER INSTALACIJE PYWPS-A	43
5. PROCES ZA TRANSFORMACIJU KOORDINATA	44
5.1. REFERENTNI KOORDINATNI SUSTAV	44
5.2. GEODETIC PARAMETER DATASET	47
5.3 CRS ZA HRVATSKU	47
5.4. KREIRANJE WPS-PROCESA ZA TRANSFORMACIJU KOORDINATA.....	50
5.5. JEDNOSTAVNI KLIJENT ZA WPS-SERVIS ZA TRANSFORMACIJU KOORDINATA	53
6. ZAKLJUČAK.....	59
LITERATURA	60
POPIS SLIKA.....	62
POPIS TABLICA	63
SADRŽAJ PRILOŽENOG OPTIČKOG MEDIJA.....	63

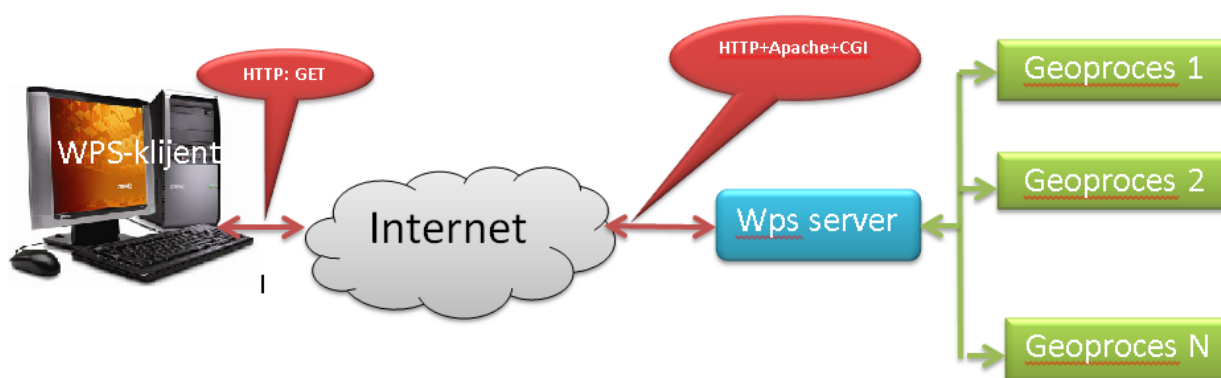
1. UVOD

Klasični geoinformacijski sustavi koji omogućuju pregledavanje i izmjenu prostornih podataka često su skupi, te ih koriste tvrtke i ustanove koje si to mogu priuštiti. Ponekad su takvi sustavi nedovoljno fleksibilni i nisu primjereni malim i srednjim tvrtkama (Lonsley i dr. 2011).

Kako bi se GIS približio malim tvrtkama, on mora biti jednostavan i jeftin. Danas je to moguće putem *open-source* alata (alata otvorenog koda) i besplatnih servisa za prikaz digitalnih karata (npr. Google Maps). Budući da takvi alati i servisi već postoje, nisu potrebna dodatna financijska i vremenska ulaganja u njihov razvoj te je konačni proizvod (GIS) jeftiniji. Takvim se sustavima pristupa preko interneta, što je dodatna prednost jer je za upotrebu potreban jedino internetski preglednik i pristup internetu (URL 1).

Veliku ulogu u takvim sustavima imaju standardi i specifikacije o načinu spremanja, uređivanja i razmjeni prostornih podataka koji olakšavaju nadogradnje sustava i povezivanje s postojećim. Ti standardi omogućavaju razvoj različitih geo-servisa preko interneta (URL 2).

U ovom je radu opisana mogućnost implementacije standarda WPS (Web Processing Service) (URL 3), koji omogućava kreiranje više servisa za obradu podataka na jednom poslužitelju. Implementacija je prikazana na primjeru realizacije servisa za transformaciju koordinata (slika 1).



Slika 1. Implementacija WPS-a (Web Processing Service), servisa za obradu prostornih podataka

2. TEORIJSKE POSTAVKE I TEHNOLOGIJE

2.1. Mrežni servisi

Mrežni servisi omogućavaju korisniku razmjenu podataka i izvršavanje određenih zahtjeva na poslužitelju. Korisnik poslužitelju šalje upit, a poslužitelj zatim izvrši potrebne radnje i vraća korisniku rezultate tog upita. Rezultati su obično formatirani u formatu XML (eXtended Markup Language). Najčešće se za komunikaciju koristi protokol HTTP (HyperText Transfer Protocol) (URL 4). Konzorcij W3C (World Wide Web Consortium) definira mrežni servis kao: “Programski sustav dizajniran tako da podržava međusobnu stroj-stroj interakciju preko mreže“ (URL 5). Mrežni servis može imati sučelje opisano u formatu koji je moguće strojno obrađivati. Drugi sustavi komuniciraju s mrežnim servisom putem poruka SOAP (Simple Object Access Protocol) (URL 6), najčešće koristeći protokol HTTP i datoteke u formatu XML, uz eventualno korištenje i drugih standarda vezanih uz internet. Osim komunikacije putem SOAP-a moguće je komunicirati i primjenom principa REST (Representation State Transfer) (URL 7) koji je jednostavniji za korištenje. WPS omogućuje korisniku znati kakve usluge nudi servis kako bi ih mogao koristiti. U nastavku će biti opisani protokoli i formati datoteka koje koriste mrežni servisi WPS-a.

2.2. XML (eXtended Markup Language)

XML (Morrison, 2010) je kratica koja označava eXtensible Markup Language, odnosno jezik za opis podataka. Ideja koja stoji iza XML-a je stvaranje jednostavnoga opisnog jezika koji bi bio čitljiv i ljudima i strojevima, za razliku od npr. binarnih datoteka koje su teško čitljive ljudima i standardnih tekstualnih datoteka kojima manjka struktura podataka. Oznake koje se koriste u jeziku XML-a su dosta slične onima koje koristi jezik HTML (Deitel i dr. 2002)). Razlika je u tome što HTML-ima strogo definirane oznake (tagove) koji se smiju koristiti, dok je kod XML-a to puno fleksibilnije, odnosno ostavljeno na volju korisniku. XML je našao brojne primjene, tako se koristi za odvajanje podataka od prezentacije (npr. HTML je prezentacijski jezik), razmjenu podataka (npr. kod mrežnih servisa), spremanje podataka (npr. program može svoje postavke čuvati u XML datoteci koja pomalo zamjenjuje standardnu konfiguracijsku datoteku s ekstenzijom .ini), itd.

XML je zamišljen tako da podatke čuva u tekstualnom obliku, ali da ih uokviruje oznakama koje govore o kakvom se podatku radi. Na taj način je XML jednostavno čitljiv čovjeku (tekst), a računalo jednostavno nalazi potrebne podatke koristeći oznake (tagove).

Računalo to radi putem programa koji pretražuje datoteku i izdvaja i grupira podatke ovisno o oznakama koje ih okružuju. Ti se program nazivaju *parseri*. Zahtjevi koje je W3C postavio pred XML početkom devedesetih godina i koji su i karakteristike XML jezika (URL 8) su :

1. XML mora biti izravno primjenjiv na internetu.
2. XML mora podržavati širok spektar primjena.
3. XML mora biti kompatibilan sa SGML-om (Standard Generalized Markup Language) (URL 9).
4. Mora biti lako pisati programe koji obrađuju (parsiraju) dokumente u XML-u.
5. Broj dodatnih opcija u XML-u mora biti apsolutno minimalan, u idealnom slučaju jednak nuli.
6. Dokumenti u XML-u moraju biti čitljivi ljudima, te u razumnoj mjeri jednostavni.
7. Standard mora biti specificiran što prije.
8. Dizajn XML-a mora biti formalan i precizan.
9. Kreiranje dokumenta u XML-u mora biti jednostavno.
10. Sažetost kod označavanja dokumenta XML-om je od minimalnog značaja.

2.3. Apache

Programski paket Apache je HTTP poslužitelj, tj. web aplikacija otvorenoga kôda koju je razvila organizacija Apache Software Foundation (URL 10). Taj se programski paket može izvršavati na više platformi: Linux, Windows, Unix i drugim. Programski paket Apache smatra se vrlo stabilnim web poslužiteljem koji se upotrebljava za najzahtjevnije projekte. Sva se konfiguracija web poslužitelja Apache obavlja u datotekama koje se nalaze u direktoriju "conf", unutar samog programa. Konfiguracijska datoteka je "httpd.conf".

2.4. Python

Python je dinamički i objektno orijentirani programski jezik koji može biti primijenjen u različite svrhe (URL 11). Može se usporediti s programskim jezicima Tcl, Perl, Ruby, Scheme ili Javom. Odlikuje se vrlo čistom, čitljivom sintaksom i bogatom standardnom bibliotekom koja pokriva područja kao što su:

- upravljanje tekstem (regularni izrazi, podrška za *unicode*, datoteke),

- internetski protokoli (HTTP, FTP, SMTP, XML-RPC, POP, IMAP) i
- operacijski sustavi (sistemski pozivi, datotečni sustavi, TCP/IP).

Programski kôd u Pythonu se može izvoditi na svim važnijim operacijskim sustavima (Windows, Linux/Unix, OS/2, OS X), a uključen je u većinu GNU/Linux distribucija i OS X gdje nije potrebna posebna instalacija.

Za pokretanje programa u Pythonu potreban je interpreter Pythona. Program napisan u Pythonu interpreter prevodi u međukôd (bytecode) koji se izvršava u virtualnom okruženju.

Standardna biblioteka Pythona dobro je prilagođena pisanju web aplikacija, ali ne uključuje module za direktnu obradu prostornih podataka već pruža API (Application Programming Interface) (URL 12) s pomoću kojeg se mogu napraviti takve obrade. Web-aplikacije mogu koristiti module izvan standardnih biblioteka koje su već dostupne za obradu prostornih podataka (npr. pyPROJ, GDAL/ORG, Numpy i sl.).

3. STANDARD OGC-a ZA WPS

3.1. Općenito o OGC-u

Open Geospatial Consortium (Udruženje za otvorenost prostornih podataka) međunarodno je udruženje koje okuplja 401 tvrtku, vladine organizacije i sveučilišta radi stvaranja javno dostupnih standarda vezanih uz prostorne podatke (URL 13). Ti su standardi zamišljeni kao rješenje za razmjenu prostornih podataka na internetu, bežičnim i položajno orijentiranim servisima i općenito u računalnim informacijskim sustavima. Standardi i specifikacije OGC-a su tehnički dokumenti koji opisuju sučelja prema kojima razvojni programeri stvaraju podršku u svojim programskim paketima ili servisima. U idealnom slučaju, kada dva programera u svom softveru nezavisno implementiraju mogućnosti prema specifikacijama, rezultat bi trebao biti ispravan (bez pogrešaka) razmjenom podataka između ta dva nezavisno razvijana softverska paketa ili servisa. Standardi i specifikacije OGC-a nose naziv OpenGIS Standards, a u nastavku je detaljnije opisan WPS (Web Processing Servis) (URL 14).

3.2. Što je WPS?

WPS je određen za obradu podataka (usluge) na webu i to je standard OGC-a (Open Geospatial Consortium) za protokol kojim se mogu ostvariti GIS-usluge preko interneta. U teoriji, WPS bi trebao biti u mogućnosti opisivati bilo koji proces, te ga pokrenuti pomoću unaprijed definiranih ulaznih i izlaznih podataka..

OGC definira:

“WPS daje pravila za standardizaciju ulaza i izlaza (zahtjeva i odgovora) za geoprostorne usluge koje obrađuju prostorne podatke, primjerice poligone.”

Standard također definira kako klijent može zatražiti izvršenje procesa, te kako rezultat izlazi iz procesa obrade. Definira sučelje koje olakšava objavljivanje prostornih procesa.

Podaci potrebni za WPS se mogu isporučiti preko mreže, ili mogu biti dostupni na poslužitelju. WPS podržava istodobnu izloženost procesa putem protokola HTTP GET, HTTP POST i SOAP, te na taj način omogućava klijentu da odabere najprikladniji mehanizam za sučelje. Metapodaci se prikupljaju korištenjem protokola HTTP GET dok se izvršenje obavlja s pomoću protokola HTTP POST.

WPS definira tri osnovne operacije:

1. *GetCapabilities*, vraća metapodatke o dostupnim procesima

2. *DescribeProcess*, vraća opis procesa te ulaznih i izlaznih podataka
3. *Execute*, vraća rezultat izvršavanja procesa

3.2.1. *GetCapabilities*

GetCapabilities zahtjev vraća metapodatke o svim dostupnim uslugama, odnosno procesima, a strukturiran je u sljedeće cjeline:

1. *Service Identification*, identifikator usluge
2. *Service Provider*, informacija o tome tko pruža usluge i kako se obratiti osobi zaduženoj za njih (ime i prezime, telefon i sl.
3. *Operation Metadata*, opis za protokole HTTP GET i POST
4. *Processes Offered*, popis procesa, sažetak, identifikator, metapodaci i sl.
5. *Languages Supported*, jezici podržani od strane službe; pretpostavljena vrijednost je engleski

GetCapabilities zahtjev formuliran u protokolu HTTP GET izgleda otprilike ovako:

`http://apps.esdi-humboldt.cz/pywps?service=WPS&request=GetCapabilities`

`http://apps.esdi-humboldt.cz/pywps` je URI (Uniform Resource Identifier) servisa. Nakon znaka “?” dolaze parametri u formatu *KVP* (*Key-Value-Pair*). U principu, parametri se navode u obliku “...&key1=value1&key2=value2&..”.

KVP-zahtjev WPS-u putem protokola HTTP GET može biti definiran i kao XML-zahtjev putem protokola HTTP POST.

```
<?xml version="1.0" encoding="UTF-8"?>
<ows:GetCapabilities xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/ows/1.1 ..\wpsGetCapabilities_request.xsd"
language="en" service="WPS">
<ows:AcceptVersions>
<ows:Version>1.0.0</ows:Version>
```

```
</ows:AcceptVersions>
</ows:GetCapabilities>
```

3.2.2. DescribeProcess

Ovaj zahtjev osigurava klijentu podatke s pomoću kojih može utvrditi obvezne, opcionalne i zadane parametre za određeni proces, kao i format ulaznih i izlaznih podataka.

Zahtjev *DescribeProcess* putem protokola HTTP GET može izgledati ovako:

```
http://apps.esdi-humboldt.cz/pywps/?service=WPS&version=1.0.0
&request=DescribeProcess&identifier=all
```

Parametri *service=WPS* i *version=1.0.0* su obavezni. Parametar *identifier* može biti ili identifikator procesa kako je naveden u *GetCapabilities* odgovoru ili ključna riječ "all" za sve dostupne procese.

Ovakav zahtjev će vratiti sljedeće informacije:

1. *Service identification*, naslov i sažetak
2. *Identifier*, posebni identifikator usluge
3. *Data Inputs*, ulazni podaci
4. *Data Outputs*, izlazni podaci

Gore navedeni zahtjev preveden u XML-zahtjev koji se izvršava putem protokola HTTP POST izgleda ovako:

```
<DescribeProcess xmlns="http://www.opengis.net/wps/1.0.0"
  xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
  ../wpsDescribeProcess_request.xsd"
  service="WPS" version="1.0.0" language="en-CA">
```

```
<ows:Identifier>all</ows:Identifier>
</DescribeProcess>
```

3.2.3. Tipovi ulaznih i izlaznih podataka

Tri su oblika ulaznih i izlaznih podataka definirani u standardu OGC-a. To su *LiteralData*, *ComplexData* i *BoundingBoxData*.

3.2.3.1. LiteralData

U tipu *LiteralData* može biti bilo koji niz znakova, realni broj, datum, itd. Obično opisuje primitivne tipove u shemi W3C XML koji su opisani na <http://www.w3.org/TR/xmlschema-2/t#built-in-primitive-datatypes>.

WPS također dopušta korištenje *UOM (Units of Measurement)*, mjerne jedinice za zadane vrijednosti i *AllowedValues* odnosno zadavanje domene dopuštenih vrijednosti, što znači da usluga može biti svjesna mjernih jedinica. U slučaju nedostatka ulazne vrijednosti servis će koristiti pretpostavljene vrijednosti, a ulazna vrijednost bi trebala biti u zadanoj domeni.

Na primjer, *DescribeProcess* za određeni proces može sadržavati sljedeće opise ulaznih podataka:

```
<Input minOccurs="0" maxOccurs="1">
<ows:Identifier>method</ows:Identifier>
<ows:Title>Interpolation method</ows:Title>
<ows:Abstract>Interpolation method to be used in a dataset of points</ows:Abstract>
<LiteralData>
<ows:DataType ows:reference="xs:string">string</ows:DataType>
<ows:AllowedValues>
<ows:Value>idw</ows:Value>
<ows:Value>kriging</ows:Value>
<ows:Value>thysson</ows:Value>
```

```

</ows:AllowedValues>
<DefaultValue>kriging</DefaultValue>
</LiteralData>

```

Navedenim XML-om je opisano sljedeće: ulazna vrijednost koja se može pojaviti jednom s imenom "method", tipa je *string*, a vrijednost može biti jednaka "idw", "kriging" ili "thysso", ali ako nema vrijednosti, koristi se pretpostavljena vrijednost "kriging".

Sljedeći primjer daje opis korištenja UOM-a, mjernih jedinica.

```

<Input minOccurs="0" maxOccurs="1">
  <ows:Identifier>BufferDistance</ows:Identifier>
  <ows:Title>Buffer Distance</ows:Title>
  <ows:Abstract>Distance to be used to calculate buffer.</ows:Abstract>
  <LiteralData>
    <ows:DataType ows:reference="http://www.w3.org/TR/xmlschema-2/#float">float</ows:DataType>
    <UOMs>
      <Default>
        <ows:UOM>meters</ows:UOM>
      </Default>
      <Supported>
        <ows:UOM>meters</ows:UOM>
        <ows:UOM>feet</ows:UOM>
      </Supported>
    </UOMs>
    <ows:AnyValue/>
    <DefaultValue>100</DefaultValue>
  </LiteralData>
</Input>

```

U tom slučaju imamo ulaz koji je realni broj i čije jedinice mogu biti u metrima ili stopama, ako ulazna vrijednost nije definirana pretpostavlja da se radi 100.0 metara. Obično se *UOM*, *DefaultValue* i *AllowedValue* ne koriste, ali WPS može prijaviti pogreške ako su ulazne vrijednosti netočne ili loše izražene.

3.2.3.2. ComplexData

Tip podataka *ComplexData* koristi se za predaju vektorskih ili rasterskih podataka na server ili za dobivanje rezultata procesa. Postoje dva načina kako se ovi složeni podaci obrađuju. Može ih se poslati izravno kao dio zahtjeva na poslužitelj ili ih se može dobiti kao dio odgovora poslužitelja u XML-u. Za vektorske podatke uglavnom se koristi GML (Geography Markup Language) ili neki drugi tekstualni format. Za rasterske podatke može biti korišteno binarno kodiranje *base64*. Može se poslati ili dobiti referencu na podatke (URL datoteke ili usluge), gdje ti podaci mogu biti preuzeti.

Uobičajeno složeni podaci slijede logiku: *format / kodiranje / shema*. Također za ulazne datoteke može biti definirana njihova maksimalna veličina.

Primjer opisa složenog podataka:

```
<Input minOccurs="1" maxOccurs="1">
  <ows:Identifier>InputPolygon</ows:Identifier>
  <ows:Title>Polygon to be buffered</ows:Title>
  <ows:Abstract>URI to a set of GML that describes the polygon.</ows:Abstract>
  <ComplexData maximumMegabytes="5">
    <Default>
      <Format>
        <MimeType>text/xml</MimeType>
        <Encoding>base64</Encoding>
        <Schema>http://foo.bar/gml/3.1.0/polygon.xsd</Schema>
      </Format>
    </Default>
  </Supported>
  <Format>
```



```

<MimeType>text/xml</MimeType>
<Encoding>UTF-8</Encoding>
<Schema>http://foo.bar/gml/3.1.0/polygon.xsd</Schema>
</Format>
</Supported>
</ComplexData>
</Input>

```

U navedenom slučaju je definiran složeni podatak, koji ima identifikator *InputPolygon*. Njegova veličina ne smije prelaziti 5 Mb i opisan je XML-om s kodiranjem *base64*. Ako postoji potreba za provjeru, njegova shema se nalazi na: <http://foo.bar/gml/3.1.0/polygon.xsd>.

3.2.3.3. BoundingBoxData

Granični okvir odnosno pravokutnik, ili samo *BBOX* je treći tip podataka te se koristi za opisivanje graničnog okvira nekakvog područja. Ulazni podatak mora navesti referentni koordinatni sustav (CRS) koji se koristi, obično to je URI na sustav s kôdom EPSG-a (European Petroleum Survey Group).

Primjer:

```

<Input>
<ows:Identifier>bboxInput</ows:Identifier>
<ows:Title>bounding box of dummy polygon</ows:Title>
<ows:Abstract>Bounding box of dummy polygon to be used for
fast polygon interception calculation</ows:Abstract>
<BoundingBoxData>
<Default>
<CRS>urn:ogc:def:crs:EPSG:6.6:4326</CRS>
</Default>
<Supported>
<CRSsType>

```

```

<CRS>urn:ogc:def:crs:EPSG:6.6:4326</CRS>
<CRS>urn:ogc:def:crs:EPSG:6.6:4979</CRS>
</CRSsType>
</Supported>
</BoundingBoxData>
</Input>

```

U gornjem primjeru se očekuje da ulazni podatak *BBOX* bude u EPSG:4326 i formatu verzije 6.6, što znači geografska širina/dužina u datumu WGS84. Servis također podržava ulaz u referentnom koordinatnom sustavu EPSG:4979 koji je također geografska širina/dužina u datumu WGS84.

KVP-zahjev bi izgledao ovako:

```
... & bboxInput = 71.63,41.75, -70.78,42.90, urn: ogc: def: crc: EPSG: 6.6:4326
```

gdje *bboxInput* = *LowerCorner* (geografska dužina), *LowerCorner* (geografska širina), *UpperCorner* (geografska dužina), *UpperCorner* (geografska širina).

3.2.4. Execute

Execute zahtjev je najvažniji zahtjev, budući da će pokrenuti određeni proces usluge. Za izvršavanje zahtjeva klijenta potrebno je odrediti:

1. Identifikator procesa
2. Ulazne vrijednosti definirane kao u *DescribeProcess*
3. Verziju i jezik
4. Način preuzimanja izlaznih podataka:
 1. Spremljen na poslužitelju
 2. Sadržan unutar odgovora u XML-u
 3. Neobrađeni odgovor u jednoj vrijednosti
5. Status dokumenta (sinkroni ili asinkroni poziv)
6. Trebaju li ulazni podaci biti vraćeni u dokumentu odgovora (*lineage*)

3.2.5. Zahtjev *Execute* u obliku *KVP*

Iako je izvršavanje upita bolje učiniti s pomoću XML-a jer neki zahtjevi mogu biti kompleksni, moguće je upit izvršiti s pomoću protokola HTTP GET i KVP-parametrima. Kao primjer dan je KVP-zahtjev za proces koji ima 4 ulazna podatka tipa *LiteralData* (*int*, *float*, *zeroset* i *string*):

```
http://apps.esdi-humboldt.cz/pywps/?
service=WPS&version=1.0.0&
request=Execute&
identifier=literalprocess&
datainputs=int=1;float=3.2;zeroset=0;string=spam&
storeExecuteResponse=false
lineage=true
status=false
```

Service i *version* su obvezni parametri kao i u zahtjevu *DescribeProcess*, a parametar *datainputs* sadrži ulazne podatke. U ovom slučaju definirani su sljedeći podaci: *int*, *float*, *zeroset* i *spam* s pridruženim vrijednostima 1, 3.2, 0 i *spam*. *StoreExecuteResponse = false* definira da poslužitelj neće spremiti dokument s rezultatom obrade. *Lineage* parametar definira da vraćeni dokument također sadrži ulazne podatke. *Status = false* ukazuje na to da odgovor nema status dokumenta, i mora biti uključen u rezultat.

Parametri kao što su *storeExecuteResponse* ili *status* ovisi o samom procesu, što znači da autor procesa definira da li će podržati *storeExecuteResponse* i ostale opcije.

KVP-zahtjev treba biti kodiran pomoću sljedećih pravila:

1. Točka-zarez (;) koristi se za odvajanje dva ulazna podatka
2. Znak jednakosti (=) se koristi za odvajanje imena ulaznog podatka od njegove vrijednosti i imena atributa ulaznih vrijednosti od njihovih vrijednosti
3. Simbol (@) se koristi za odvajanje ulazne vrijednosti od njezinih atributa i jednog atributa od drugog
4. Sve vrijednosti ulaznih podataka i vrijednosti atributa moraju biti kodirane pomoću standarda za kodiranje URL-ova

5. Upotrebljavaju se znakovi [] za označavanje početka i kraja *datainputs* kako slijedi:
datainputs=[int=1, float=3,2]
6. *LiteralData* vrijednost može biti kodirana na sljedeći način:
width=35 @ datatype=xs: integer@ uom=metar

3.2.6. Zapis složenog podatka u XML-u

Primjer:

```
....complexType=foo%2Ebar%2Fshapefile@Format=text/xml@Encoding=utf-8@Schema=gml.xsd.
```

U gornjem slučaju zapis složenog podatka u XML-u smješten je na *http://foo.bar/shapefile*. To je datoteka XML-a sa shemom *gml.xsd*.

Primjer za *Bounding box*: *bboxInput=46,102,47,103,urn:ogc:def:crs:EPSG:6.6:4326,2*

Zahtjev za *Execute* može rezultirati generiranjem odgovora u XML-u kao u sljedećem primjeru:

```
<?xml version="1.0" encoding="utf-8"?>
<wps:ExecuteResponse xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsGetCapabilities_response.xsd"
service="WPS" version="1.0.0" xml:lang="eng"
serviceInstance="http://appd.esdi-
humboldt.cz/pywps/?service=WPS&request=GetCapabilities&version=1.0.0"
statusLocation="http://apps.esdi-humboldt.cz/wps/wpsoutputs/pywps-
128222636556.xml">
<wps:Process wps:processVersion="None">
<ows:Identifier>literalprocess</ows:Identifier>
<ows:Title>Literal process</ows:Title>
</wps:Process>
```

```
<wps:Status creationTime="Thu Aug 19 15:59:25 2010">
<wps:ProcessSucceeded>PyWPS Process literalprocess successfully
calculated</wps:ProcessSucceeded>

</wps:Status>
<wps:ProcessOutputs>
<wps:Output>
<ows:Identifier>int</ows:Identifier>
<ows:Title>Integer data out</ows:Title>
<wps:Data>
<wps:LiteralData dataType="integer">1</wps:LiteralData>

</wps:Data>
</wps:Output>
<wps:Output>
<ows:Identifier>float</ows:Identifier>
<ows:Title>Float data out</ows:Title>
<wps:Data>
<wps:LiteralData dataType="float">3.2</wps:LiteralData>

</wps:Data>
</wps:Output>
<wps:Output>
<ows:Identifier>string</ows:Identifier>
<ows:Title>String data out</ows:Title>
<wps:Data>
<wps:LiteralData dataType="string">spam</wps:LiteralData>

</wps:Data>
</wps:Output>
</wps:ProcessOutputs>
</wps:ExecuteResponse>
```

Literalprocess je primjer procesa koji se koristi za *debugging*, odnosno testiranje programa koji ulazne podatke vrata kao izlazne podatke.

Obično izlaz s tipom *ComplexData* može biti prilično opsežan. Na primjer, sljedeći zahtjev *Execute* će pokrenuti proces koji ima ulazne vektorske podatke u XML-u te rasterske podatke, a nakon par sekundi rada proces će vratiti iste podatke.

```
http://apps.esdi-humboldt.cz/pywps/?
service=wps&version=1.0.0&
request=Execute&
identifier=complexprocess&
datainputs=vectorin=http://apps.esdi-
humboldt.cz/classification/traning_areas/training_areas_en.gml;
rasterin=http://rsg.pml.ac.uk/staff/jmdj/raster.tif;
pause=0&
```

Vraćeni dokument u XML-u je iznimno opsežan, zbog toga je prikazan samo početak XML-a:

```
<!-- Something something something -->
<wps:ProcessOutputs>
<wps:Output>
<ows:Identifier>vectorout</ows:Identifier>
<ows:Title>Vector file</ows:Title>
<wps>Data>
<wps:ComplexData mimeType="text/xml">
<ogr:FeatureCollection
  xmlns:ogr="http://ogr.maptools.org/"
  xmlns:gml="http://www.opengis.net/gml">

<gml:boundedBy>
<gml:Box>
```

```

<gml:coord><gml:X>-559044.5280103994</gml:X><gml:Y>-
1177026.734255324</gml:Y></gml:coord>
<gml:coord><gml:X>-554835.891394174</gml:X><gml:Y>-
1169621.932698363</gml:Y></gml:coord>
</gml:Box>
</gml:boundedBy>
<gml:featureMember>

<ogr:features fid="F0">
<ogr:geometryProperty><gml:Polygon><gml:outerBoundaryIs>
<gml:LinearRing><gml:coordinates>-555043.324615493183956,-
1174010.838661683257669 -554930.435787564259954,
-1174159.005248340079561 -555085.657925966545008,-1174293.060731505509466
-555276.157823096611537,-1174201.338558813324198 -555191.491202149889432,
-1174088.449730884516612 -555043.324615493183956,-
1174010.838661683257669</gml:coordinates>
</gml:LinearRing></gml:outerBoundaryIs></gml:Polygon></ogr:geometryProperty>
<ogr:areaClass>1</ogr:areaClass>
<ogr:classLabel>broad_leaved</ogr:classLabel>

<!-- more stuff-->
<ows:Identifier>rasterout</ows:Identifier>
<ows:Title>Raster file</ows:Title>
<wps:Data>
<wps:ComplexData mimeType="image/tiff">
SUkqAEZlAAAJIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyM
<!-- even more stuff-->

```

WPS pruža korištenje veza (linkova) unutar dokumenta s odgovorom koje pokazuju gdje su rezultati spremljeni. To će izgledati kao u primjeru:

```
...responsedocument=vectorout=@asreference=true;
```

```
rasterout=@asreference=true
```

Varijabla *responsedocument* ima atribut "asreference" postavljen na *true*, što znači da se za izlazne podatke šalje samo veza, što rezultira manjim dokumentom odgovora u XML-u.

```
<wps:ExecuteResponse xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsGetCapabilities_response.xsd"
service="WPS" version="1.0.0" xml:lang="eng"
serviceInstance="http://appd.esdi-
humboldt.cz/pywps/?service=WPS&request=GetCapabilities&version=1.0.0"
statusLocation="http://apps.esdi-humboldt.cz/wps/wpsoutputs/pywps-
128222940863.xml">
<wps:Process wps:processVersion="None">
<ows:Identifier>complexprocess</ows:Identifier>
<ows:Title>Complex process</ows:Title>
</wps:Process>
<wps:Status creationTime="Thu Aug 19 16:50:33 2010">
<wps:ProcessSucceeded>PyWPS Process complexprocess successfully
calculated</wps:ProcessSucceeded>

</wps:Status>
<wps:ProcessOutputs>
<wps:Output>
<ows:Identifier>vectorout</ows:Identifier>
<ows:Title>Vector file</ows:Title>
<wps:Reference xlink:href="http://apps.esdi-humboldt.cz/wps/wpsoutputs/vectorout-
22983" mimeType="text/xml"/>
</wps:Output>
```



```
<wps:Output>

<ows:Identifier>rasterout</ows:Identifier>
<ows:Title>Raster file</ows:Title>
<wps:Reference xlink:href="http://apps.esdi-humboldt.cz/wps/wpsoutputs/rasterout-
22983" mimeType="image/tiff"/>
</wps:Output>
</wps:ProcessOutputs>
</wps:ExecuteResponse>
```

3.2.7. XML Execute

90% zahtjeva za izvršavanje obavlja se s pomoću dokumenta u XML-u.

Kao što je već objašnjeno dokument u XML-u je podijeljen na 3 glavna odjeljka:

1. Identifikacija servisa, jezik i verzija zahtjeva
2. Uvod podataka
3. Izrada odgovora dokumenta

Sljedeći KVP-zahtjev:

```
http://apps.esdi-humboldt.cz/pywps/?
service=WPS&version=1.0.0&
request=Execute&
identifier=literalprocess&
datainputs=int=1;float=3.2;zeroset=0;string=spam&
storeExecuteResponse=false
lineage=true
status=false
```

definiran je u XML-zahtjevu:

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:Execute service="WPS" version="1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsExecute_request.xsd">
<ows:Identifier>literalprocess</ows:Identifier>
<wps>DataInputs>
<wps:Input xmlns:xlink="http://www.w3.org/1999/xlink">
<ows:Identifier>int</ows:Identifier>
<wps>Data>
<wps:LiteralData dataType="xs:integer">1</wps:LiteralData>
</wps>Data>
</wps:Input>
<wps:Input>
<ows:Identifier>float</ows:Identifier>
<wps>Data>
<wps:LiteralData dataType="xs:float">3.2</wps:LiteralData>
</wps>Data>
</wps:Input>
<wps:Input>
<ows:Identifier>zeroset</ows:Identifier>
<wps>Data>
<wps:LiteralData dataType="xs:integer">0</wps:LiteralData>
</wps>Data>
</wps:Input>
<wps:Input>
<ows:Identifier>string</ows:Identifier>
<wps>Data>
<wps:LiteralData dataType="xs:string">spam</wps:LiteralData>
</wps>Data>
```

```

</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
<wps:ResponseDocument lineage="true" storeExecuteResponse="false"
status="false">
<wps:Output asReference="false">
<ows:Identifier>int</ows:Identifier>
</wps:Output>
<wps:Output asReference="false">
<ows:Identifier>float</ows:Identifier>
</wps:Output>
<wps:Output asReference="false">
<ows:Identifier>zeroset</ows:Identifier>
</wps:Output>
<wps:Output asReference="false">
<ows:Identifier>string</ows:Identifier>
</wps:Output>
</wps:ResponseDocument>
</wps:ResponseForm>
</wps:Execute>

```

Lineage, *storeExecuteResponse* i *status* sada su atributi elementa *ResponseDocument* i *asReference* je zadan kao atribut izlaznog elementa.

3.2.8. Dokument odgovora

U prethodnom tekstu naveden je primjer gdje se izlazi podaci traže kao reference (veze). Parametar *responsedocument* se koristi za generiranje izlaznog dokumenta koji sadrži samo jedan (ili više) od svih mogućih izlaznih vrijednosti procesa. U sljedećem primjeru traži se dokument odgovora samo s podatkom *int*.

```
http://apps.esdi-humboldt.cz/pywps/?
```

```

service=WPS&version=1.0.0&
request=Execute&
identifier=literalprocess&
datainputs=[int=1;float=3.2;zeroset=0;string=spam]&
storeExecuteResponse=false&
lineage=true&
status=false&
responsedocument=int

```

U dokumentu odgovora nalazit će se i sljedeći podaci:

```

<wps:ProcessOutputs>
<wps:Output>
<ows:Identifier>int</ows:Identifier>
<ows:Title>Integer data out</ows:Title>
<wps>Data>
<wps:LiteralData dataType="integer">1</wps:LiteralData>
</wps>Data>
</wps:Output>
</wps:ProcessOutputs>

```

Nedostatak upotrebe parametra *responsedocument* je da će se vratiti svi izlazni podaci, i ako je prisutan parametar *responsedocument* korisnik će morati odrediti koji su izlazni podaci potrebni.

3.2.9. Lineage

Dokument odgovora može sadržavati vrijednosti *dataInputs* koje su se koristile za pokretanje WPS-a, i to se obično naziva kao praćenje porijekla (*lineage*). Da bi zatražili strukturu *lineage* u dokumentu odgovora KVP-zahtjev putem protokola HTTP GET treba sadržavati *lineage= true*, kako slijedi:

```

storeExecuteResponse=false&
lineage=true&

```

```
status=false&
```

U XML-zahjtvu putem protokola HTTP POST *lineage* je atribut u parametru *responsedocument*:

```
<wps:ResponseDocument lineage="true" storeExecuteResponse="false"
status="false">
```

3.2.10. Reference

WPS može ispisati određeni sadržaj kao referencu, što znači da se izlaz vraća kao URL, koji se može koristiti za preuzimanje sadržaja, omogućujući korisniku da prihvati kratki WPS odgovor, a kasnije dohvati rezultat obrade. To je vrlo praktično ako je izlaz rasterska slika ili opsežna XML-struktura.

AsReference treba navesti kao atribut izlaza u XML-zahjtvu putem protokola HTTP POST na sljedeći način:

```
:
<wps:ResponseForm>
<wps:ResponseDocument lineage="false" storeExecuteResponse="false"
status="false">
<wps:Output asReference="false">
<ows:Identifier>bigXML</ows:Identifier>
</wps:Output>
:
</wps:ResponseDocument>
</wps:ResponseForm>
```

U KVP-zahjtvu putem protokola HTTP GET to izgleda ovako:

```
&responsedocument=bigXML=@asreference=true
```

Dokument odgovora će sadržavati URL, koji može biti izravno "spojen" na drugi WPS ili može biti korišten od nekog drugog programa (npr. pretraživača ili GIS-a)

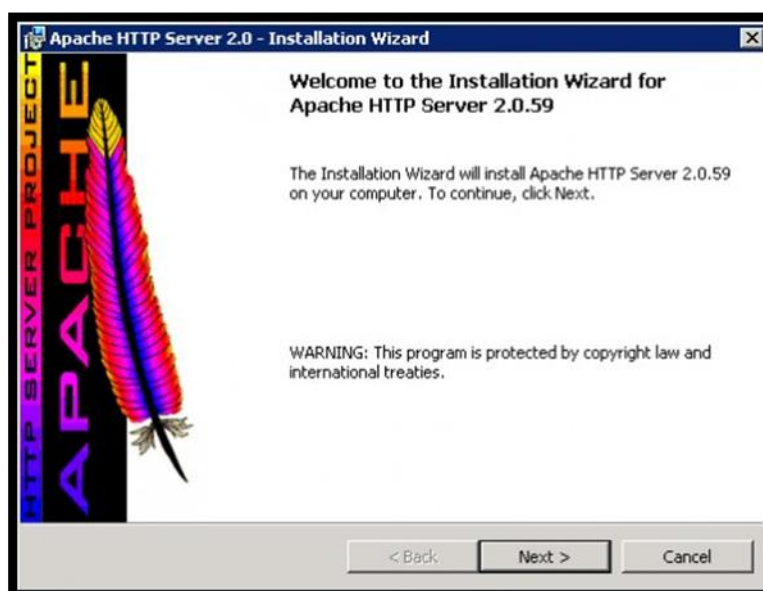
4. IMPLEMENTACIJA WPS-A U WINDOWS-U

4.1. Instalacija Apachea

Apache HTTP Server je besplatni (open source) web poslužitelj za Unix sustave, Microsoft Windows, Novell NetWare i druge platforme. Apache je najčešće korišteni web poslužitelj na internetu s udjelom korisnika više od 73%.

Kako bi mogli lokalno na svom računalu izvršavati web-skripte pisanu u Pythonu, potreban je web-poslužitelj i Python. Za početak treba preuzeti Apache (URL 15). Preporučam preuzeti samoinstalački paket *Win32 Binary without crypto (no mod_ssl)*, verziju za MSI Installer zbog najjednostavnije instalacije.

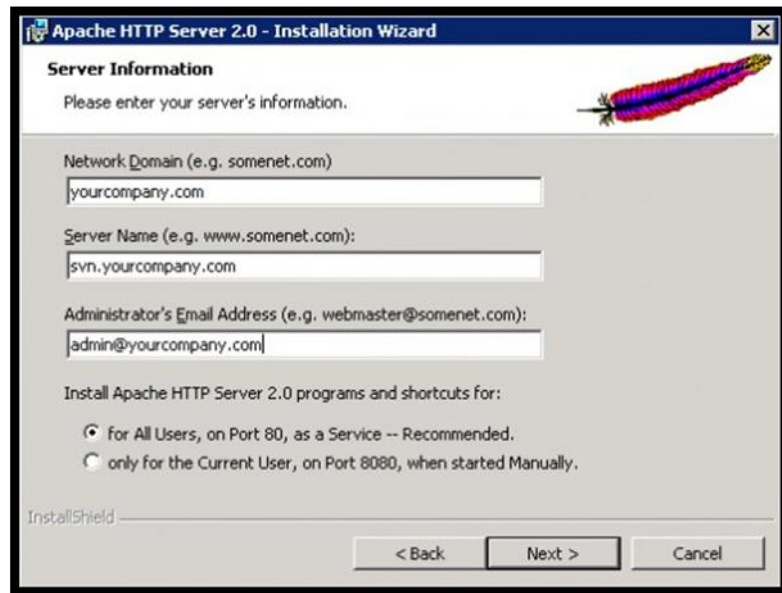
Nakon što je preuzet Apache, pokrenemo instalaciju (slika 2) koja je vrlo slična svim ostalim instalacijama u Windowsu.



Slika 2. Početak instalacije.

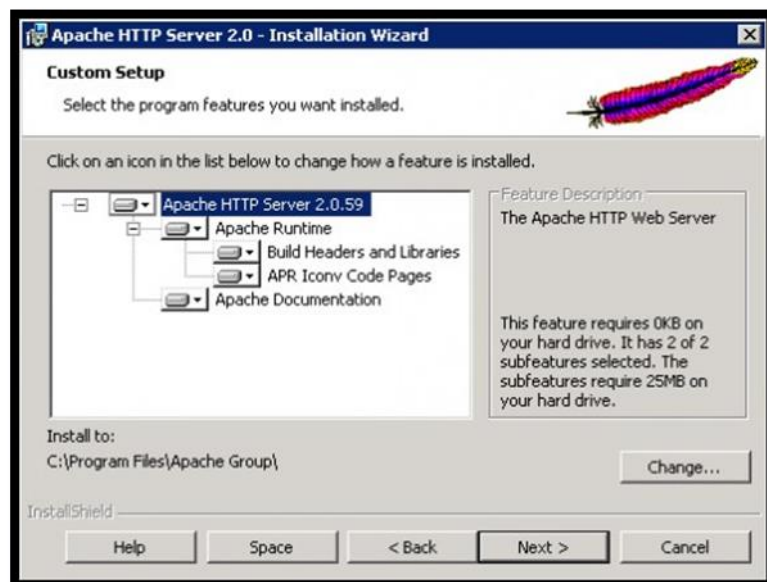
Sve do četvrtog prozora nije potrebno ništa unositi. U četvrtom prozoru morate unijeti podatke o poslužitelju (slika 3).

- Network Domain (localhost)
- Server Name (localhost)
- Administrators Email... (može se ostaviti kako je)
- Odaberimo opciju “for All Users, on port 80, as Service“

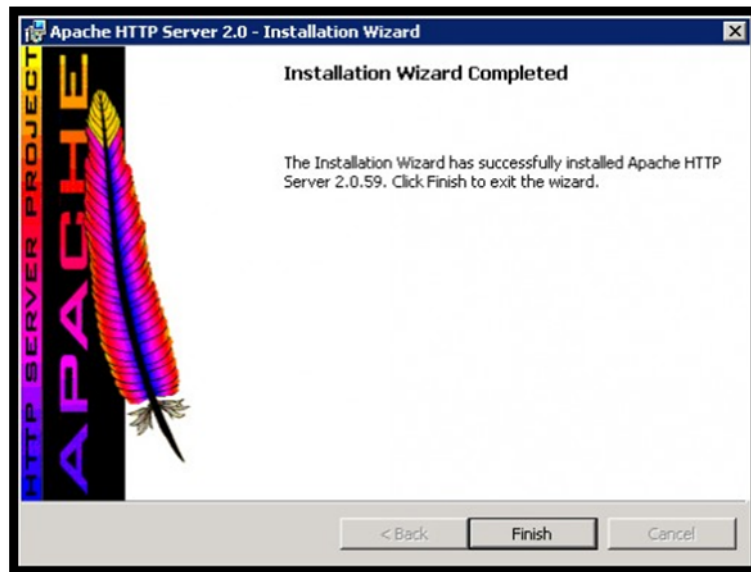


Slika 3. Podaci poslužitelja

Na sljedećem prozoru (slika 4) odaberemo *Custom* te potom *Next*, a potom samo promijenimo mjesto instalacije klikom na gumb *Change...* Preporuka je napraviti mapu u npr. *C:\Apache2*, ali to može biti bilo koja mapa na bilo kojem disku. Nakon što se odabere direktorij, potrebno je kliknuti na *Next* i zatim na *Install*.

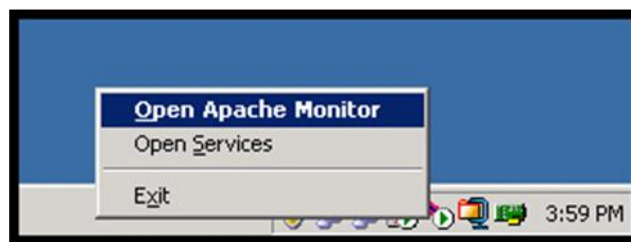


Slika 4. Odabir mjesta instalacije poslužitelja Apache.



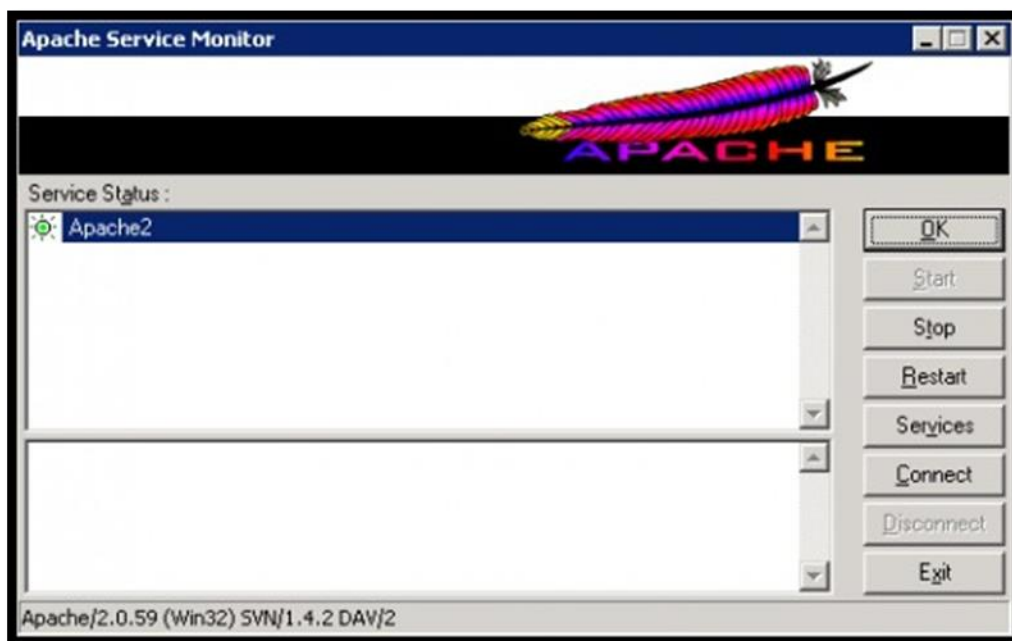
Slika 5. Završetak instalacije.

Po završetku instalacije, pojavit će se ikona u sistemskoj traci *TrayBar* u Windowsu. Desnim klikom na nju, te odabirom *Open Apache Monitor* (slika 6) otvorit će se prozor *Apache Agent* (slika 7).



Slika 6. Odabir Open Apache Monitor.

Kada je ikonica zelena, poslužitelj uredno radi. U slučaju da je poslužitelj ugašen, ikona će biti crvene boje. Ako odaberemo opciju *Services* može se vidjeti sve servise koji su registrirani na operacijskom sustavu.



Slika 7. Prozor *Apache Service Monitora*.

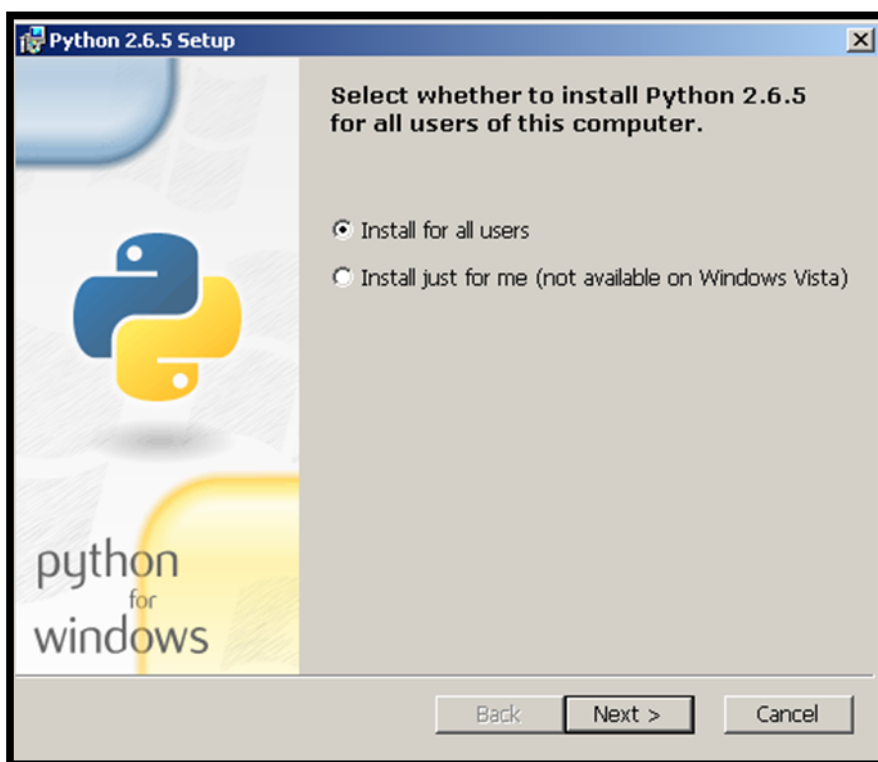
I za kraj potrebno je još obaviti testiranje poslužitelja otvaranjem web preglednika i unosom adrese *http://localhost* ili *http://127.0.0.1*. Ako je sve prošlo u redu trebala bi se pojaviti bijela stranica sa simboličnom rečenicom *It works!*

4.2. Instalacija Python-a

4.2.1. Instalacija interpretera

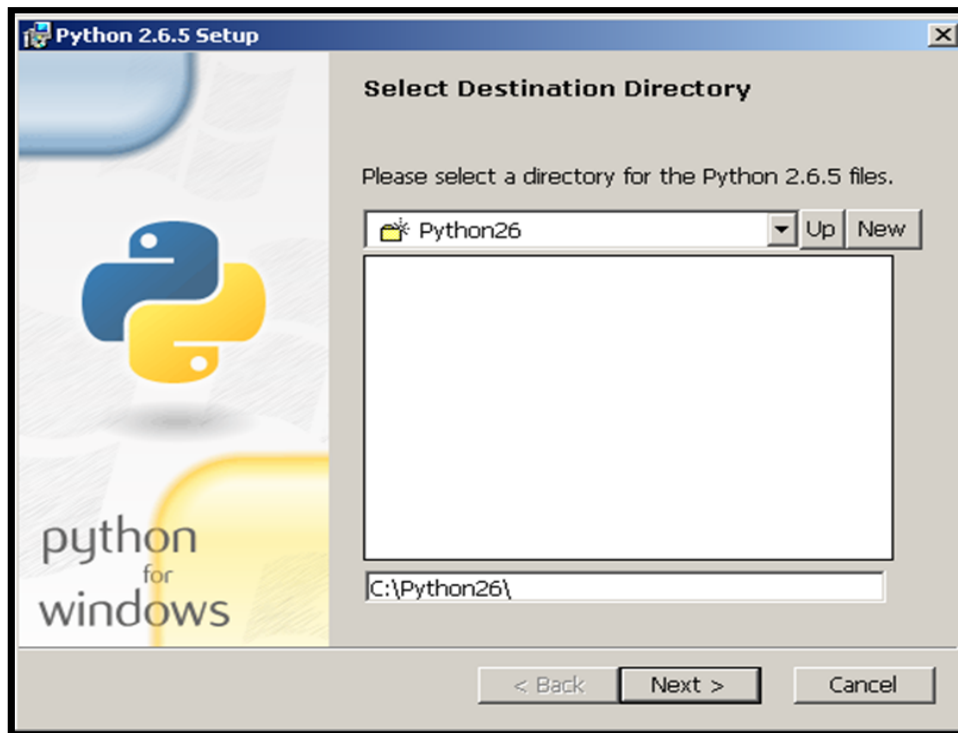
Kako je Python besplatan softver moguće ga je besplatno preuzeti s interneta na adresi *http://www.Python.org/download*.

Nakon preuzimanja istoga potrebno je pokrenuti čarobnjaka za instalaciju nakon čega se dolazi do izbornika (slika 8) koji nam nudi mogućnost instaliranja za trenutnog korisnika ili za sve korisnike koji koriste računalo. Ova odluka ovisi o vlastitom odabiru korisnika, ali preporuča se instalaciju provesti tako da Python bude dostupan svim korisnicima.



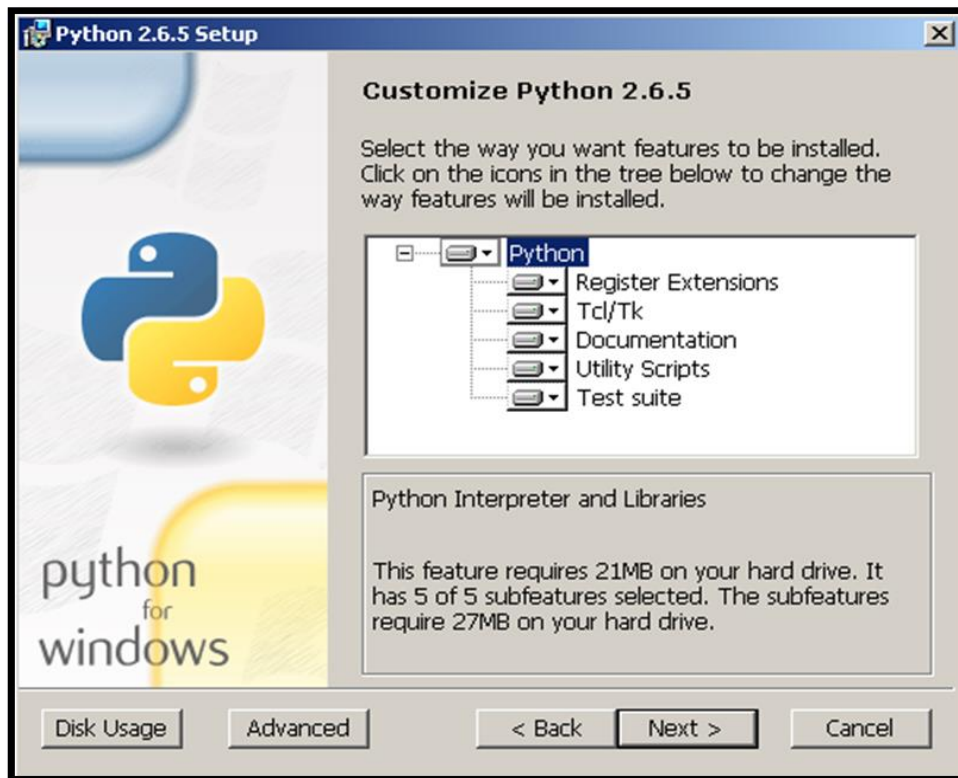
Slika 8. Početni izbornik instalacijskog čarobnjaka za *Python*

Nakon izvedenog odabira, isti potvrdimo klikom na *Next* te dolazimo do sljedećeg izbornika (slika 9) koji nam nudi mogućnost odabira lokacije na računalu i direktorija u koji ćemo instalirati *Python*. Preporuča se ostavljanje tvorničkih postavki kako u narednim koracima ne bi došlo do problema prilikom instaliranja biblioteka, jer će postupak biti opisan za tvorničke postavke softvera.



Slika 9. Izbornik instalacije s mogućnošću odabira mjesta i direktorija instalacije

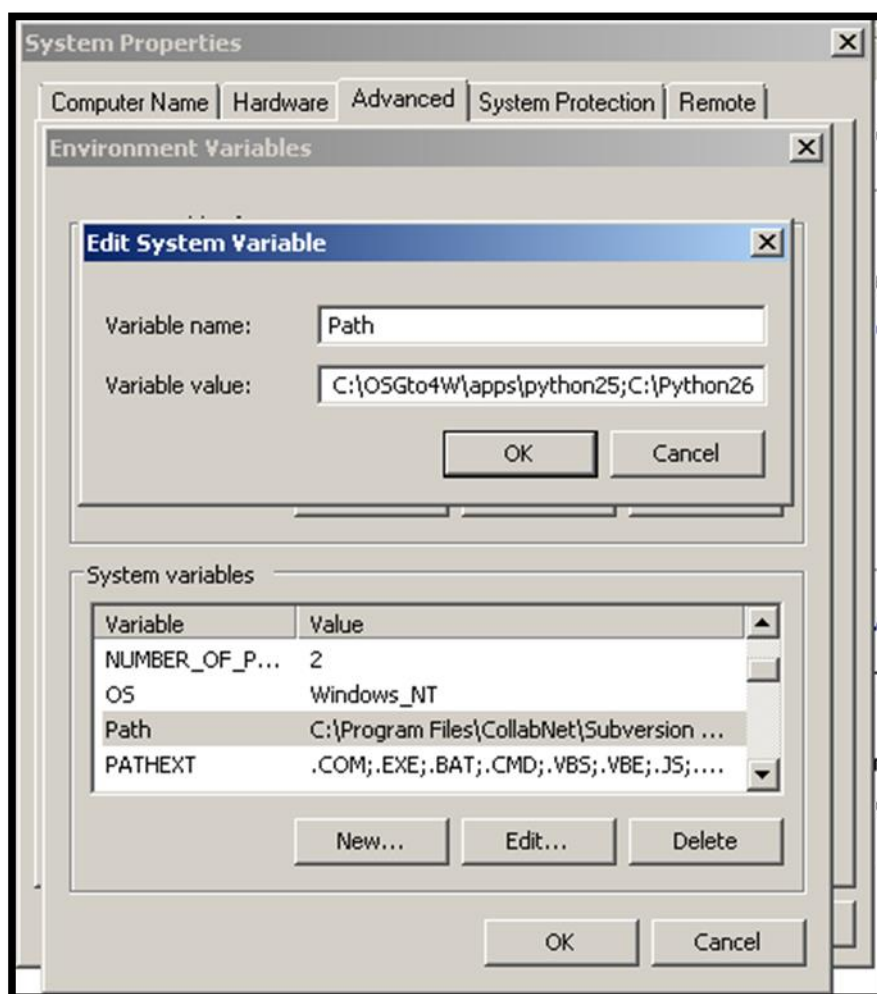
Nakon izvršenog odabira lokacije (particije i direktorija) za instalaciju programskog jezika Python, potvrdimo odabir klikom na *Next* i dolazimo do trećeg prozora izbornika (slika 10) koji nam nudi mogućnost odabira koje dijelove instalacijskog paketa želimo instalirati. Preporučuje se ostavljanje tvornički podešenog izbora instalacije svih modula koji dolaze s instalacijom Python jer cjelokupna instalacija Pythona sa svim modulima zauzima veoma malo memorijskog prostora za računala današnjeg vremena. S obzirom na sve navedeno bilo bi pogrešno odabrati samo pojedine module i time si uskratiti određene mogućnosti programskog jezika osim ukoliko ste iskusni korisnik koji poznaje sadržaj modula, njihovu funkcionalnost, moguće posljedice nedostatka pojedinog modula i sl.



Slika 10. Izbornik koji dozvoljava odabir pojedinih modula Pythona za instaliranje

Potvrđivanjem odabira počinje instalacija programskog jezika Python koja je relativno kratka i nakon što je gotova instalacija pojavit će se prozor koji se zahvaljuje autorima i zahtjeva od korisnika da potvrdi završetak instalacije klikom na gumb *Finish*.

Sada je potrebno promijeniti staze operacijskog sustava i varijable sukladno uputama. Iz izbornika *Start*, treba odabrati *Control Panel*, od ponuđenog odabrati *System* i otvori se prozor *SystemProperties*. U tome prozoru treba odabrati karticu *Advanced* i s navedene kartice odabrati *Environment Variables* (skup dinamički imenovanih vrijednosti koje utječu na način izvršavanja procesa u računalu). Tada se otvara novi prozor *Environment Variables* u kojem treba promijeniti staze (slika 11). Treba pronaći varijablu imena *Path* i pod vrijednost varijable dodati mape biblioteka Pythona odvojene znakom “;” (npr. ;C:\Python26\;C:\Python26\lib).



Slika 11. Dodavanje i uređivanje varijabli i staza operacijskog sustava

4.2.2 Instalacija biblioteka Pythona

Za implementaciju WPS-servisa potrebno je još instalirati dodatne biblioteke (tablica 1).

Tablica 1. Dodatne biblioteke Python-a potrebne za WPS.

Biblioteka	Opis
Pyton-gdal	Rad s vektorskim i rasterskim datotekama
pyProj	Rad s različitim datumama i projekcijama
python-lxml	Rad s XML datotekama
python-magic	Rad s klasama

Sve biblioteke treba preuzeti sa stranice <http://pypi.Python.org>. Cijeli postupak instalacije opisan je na istoj stranici.

4.3. Instalacija PyWPS-a

PyWPS je modul pisan u jeziku Python i implementacija je WPS-standarda. Modul pyWPS moguće je besplatno preuzeti s interneta na <http://pywps.valid.intervation.org/frs/download.php/900/pywps>.

Nakon preuzimanja, istoga treba raspakirati u bilo koju mapu, na primjer *C:\tmp*. Pokrećemo konzolu za naredbe: *Start>Run>CMD*. U konzoli izvršavamo naredbu:

```
C:>\tmp\python setup.py install
```

Nakon izvršavanja dobivamo instalirani modul pyWPS u datoteci:

```
C:\Python26\lib\site-package\pywps
```

U sljedećem koraku kreiramo mapu *C:\wps*. U toj mapi kreiramo dvije datoteke:

```
C:\wps\processes i
```

```
C:\wps\templates
```

Kopiramo mapu *C:\Python26\lib\site-packages\Templates* u mapu *C:\wps\Tempates*.

U mapi *C:\wps\processes* kreirat ćemo proces koji ćemo programirati.

U mapu *C:\cgi-bin\wps* kopiramo datoteku *wps.py*. Ovo je skripta koja će pokrenuti proces preko pyWPS-servisa. Skripta je napisana u Pythonu. Sadržaj skripte je sljedeći:

```
#!C:/Python26/python.exe

#-*- coding: utf-8 -*-

import os,sys,traceback

os.environ["PYWPS_CFG"]="C:/wps/pywps.cfg"

os.environ["PYWPS_PROCESSES"]="C:/wps/processes"

os.environ["PYWPS_TEMPLATES"]="C:/wps/Templates"

os.environ["HOME"]="C:/tmp/"

sys.path.append("c:/Python26/lib")
```

```
import pywps

from pywps.Exceptions import *

import urllib

method = os.getenv("REQUEST_METHOD")

if not method: # set standard method

    method = pywps.METHOD_GET

inputQuery = None

if method == pywps.METHOD_GET:

    try:

        inputQuery = os.environ["QUERY_STRING"]

    except KeyError:

        # if QUERY_STRING isn't found in env-dictionary, try to read

        # query from command line:

        if len(sys.argv)>1: # any arguments available?

            inputQuery = sys.argv[1]

            if not inputQuery:

                err = NoApplicableCode("No query string found.")

                pywps.response.response(err,sys.stdout)

                sys.exit(1)

        else:

            inputQuery = sys.stdin

inputQuery=urllib.unquote(inputQuery)
```



```
# create the WPS object

wps = None

try:

    wps = pywps.Pywps(method)

    if wps.parseRequest(inputQuery):

        pywps.debug(wps.inputs)

        response = wps.performRequest()

        # request performed, write the response back

        if response:

            # print only to standard out

            pywps.response.response(wps.response,

                sys.stdout,wps.parser.soapVersion,wps.parser.isSoap,wps.parser.isSoapExecute,

                wps.request.contentType)

        except WPSEException,e:

            traceback.print_exc(file=pywps.logFile)

            pywps.response.response(e, sys.stdout, wps.parser.soapVersion,

                wps.parser.isSoap,

                wps.parser.isSoapExecute)
```

U konzoli za naredbe (*Start>Run.>CMD*) treba izvršiti naredbu:

```
C:>\cgi-bin\wps.py
```

Trebamo dobiti rezultat sličan ovomu:

PyWPS NoApplicableCode: Locator: None; Value: No query string found.
Content-type: text/xml

```
<?xml version="1.0" encoding="utf-8"?>
<ExceptionReport version="1.0.0" xmlns="http://www.opengis.net/ows"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<Exception exceptionCode="NoApplicableCode">
<ExceptionText>
No query string found.
</ExceptionText>
</Exception>
</ExceptionReport>
```

Ako to je bilo tako, imamo korektno instalirani pyWPS.

Sad je potrebno konfigurirati pyWPS.

4.4. Konfiguracija PyWPS-a

Datoteka za konfiguraciju pyWPS-a je *pywps.cfg*. Sistemska varijabla definira mapu gdje se nalazi ova datoteka.

```
PYWPS_CFG="C:\wps\pywps.cfg"
```

Sadržaj datoteke za moje potrebe je sljedeći:

```
[wps]
encoding=utf-8
title=PyWPS Server
version=1.0.0
abstract=See http://pywps.wald.intevation.org and
http://www.opengeospatial.org/standards/wps
fees=None
constraints=none
serveraddress=http://localhost/cgi-bin/wps.py
keywords=GRASS, GIS, WPS
lang=eng

[provider]
providerName=Sveučilište
individualName=Sergiy Kharchenko
positionName=student
role=diplom
deliveryPoint=Kačićeva 26
city=Zagreb
postalCode=10000
country=hr
electronicMailAddress=skarchenko@geof.hr
```

```

providerSite=http://localhost
phoneVoice=False
phoneFacsimile=False
administrativeArea=False

[server]
maxoperations=3
maxinputparamlength=1024
maxfilesize=3mb
tempPath=/tmp
processesPath=
outputUrl=http://localhost/wps/wpsoutputs
outputPath=C:/wps/wpsoutputs
debug=true
logfile=C:/wps/pywps.log

[grass]
path=C:/grass/bin/: C:/grass/scripts/
addonPath=
version=6.4.1
gui=text
gisbase=C:/grass/
ldLibraryPath=C:/grass/lib
gisdbase=C:/datagrass

```

4.5. Primjer instalacije PyWPS-a

PyWPS može biti instaliran na poslužitelju samo jednom. Moguće je konfigurirati da on pokrene nekoliko WPS-servisa. Svi procesi moraju biti pohranjeni kao programi u Pythonu u jednoj mapi.

Evo primjera:

1. Mapa za procese *C:\wps\processes*
2. Proces za transformaciju koordinata je u datoteci *transformkoordinata.py*
3. Spremit ćemo proces u mapu za procese *C:\wps\processes\transformkoordinata.py*
4. Svaki proces moramo prijaviti u datoteci *init.py* koja se nalazi u mapi za procese.

Datoteka *init.py* mora imati sljedeći redak:

```
__all__=["transformkoordinata"]
```

, gde *all* predstavlja popis svih procesa u mapi za procese.

5. PROCES ZA TRANSFORMACIJU KOORDINATA

5.1. Referentni koordinatni sustav

Prostorni podaci se uvijek odnose na neki koordinatni sustav. Svaki koordinatni sustav mora sadržavati datum, a može sadržavati i projekciju (ISO:International standard ISO19111, 2003). Primjerice, podaci koje snimamo s GPS-om se odnose na datum WGS84, ali nisu u projekciji. Podaci iz katastra se primjerice odnose na datum Bessel 1841 i poprečnu Mercatorovu projekciju (Transverse Mercator). Oblik Zemlje matematički se modelira rotacijskim elipsoidom. Matematička definicija tog elipsoida i njegova orijentacija se naziva datumom. On se definira skupom parametara. Na svijetu se koristi mnogo datuma (elipsoida), uglavnom svaka država ima neki svoj datum (elipsoid) koji najbolje paše obliku Zemlje na području te države, ali postoje i globalni datumi (elipsoidi) koje se odnose na cijelu Zemlju. Najpoznatiji globalni datum je WGS84, dok se za područje Hrvatske koristi Bessel 1841, a od 2010. g. elipsoid GRS80 (Husak i Lapaine, 2006). Koji parametri određuju te datume (elipsoide) je trenutno nebitno, glavno je znati da postoje ova tri koja se najčešće koriste na području Hrvatske.

Uzmimo jedan primjer, imamo neku točku u jednom datumu (elipsoidu), primjerice na WGS84. Ukoliko bi htjeli dobiti koordinate te iste točke u drugom datumu (elipsoidu), primjerice Bessel 1841, potrebni su nam transformacijski parametri. Ti parametri se u programu PROJ4 nazivaju *TOWGS*. Skoro svi datumi koji se koriste na svijetu imaju definiranu vezu preko datuma WGS84. Ukoliko njih ne bi bilo, vrlo teško bi uklopili podatke GPS-a koji se najviše koriste u praksi. Parametara ima 7, tri za translaciju, tri za rotaciju i jedan za mjerilo. Primjerice, zamislimo kada u CAD-u napravite *move* (translaciju), *rotate* (rotaciju) i *scale* (promjenu mjerila) poliliniije u 2D prostoru. Za to je potrebno dva parametra translacije (po x i y osi), jedan za rotaciju oko z osi i mjerilo. Ista je stvar, samo u 3D prostoru, tri parametra za translaciju (po x, y i z), tri za rotaciju (oko x, y i z) te jedan za mjerilo. Za Hrvatsku postoji nekoliko skupova parametara za vezu preko WGS84.

S obzirom da smo Zemlju aproksimirali elipsoidom, elipsoidne koordinate su još uvijek nepraktične. S takvim koordinatama teško je mjeriti, računati površine ili raditi bilo kakve druge matematičke operacije (moguće je, ali dosta teško). Primjerice, s točkama snimljenim GPS-om ne možemo raditi standardne operacije poput mjerenja udaljenosti ili površina jer se ti podaci nalaze u WGS84 datumu bez projekcije, tj. na elipsoidu. u kutnim jedinicama (φ i λ). Da bi se to olakšalo, potrebne su nam projekcije.

Projekcije nam omogućavaju da Zemlju koja je trenutno definirana elipsoidom, preslikamo u ravnu plohu. Kada imamo prikaz Zemlje na ravnoj plohi (ravnina) na njoj možemo lako mjeriti, računati itd. Projekcija nije ništa više od skupa matematičkih formula koje koordinate na elipsoidu pretvaraju u koordinate u ravnini. Opet kao i kod datuma, za svako područje na Zemlji postoje najpovoljnije projekcije. U Hrvatskoj se koristi poprečna Mercatorova projekcija ili još zvana Gauss–Krügerova projekcija. Ona je također definirana sa skupom parametra (URL 16). Koriste se dvije varijacije poprečne Mercatorove projekcije, prva koja rezultira koordinatnim sustavima 5. i 6. zone. Podaci u ovim koordinatnim sustavima se nalaze na datumu Bessel 1841.

Od 2010. g. u Hrvatskoj je stupio na snagu novi koordinatni sustav koji je definiran na elipsoidu GRS80 (HTRS96, Hrvatski terestički referentni sustav za epohu 1995.55) (Lapaine i Tutić 2007).

Za definiciju datuma i projekcije u PROJ4 koriste se sljedeći parametri:

+a	- velika poluos elipse
+axis	- orijentacija osi (novo u 4.8.0)
+b	- mala poluos elipse
+datum	- ime datuma (vidi `proj -ld`)
+ellps	- ime elipsoida (vidi `proj -le`)
+k	- linearno mjerilo (staro ime)
+k_0	- linearno mjerilo (novo ime)
+lat_0	- geografska širina ishodišta
+lat_1	- prva standardna paralela
+lat_2	- druga standardna paralela
+lat_ts	- paralela bez deformacija
+lon_0	- srednji meridijan
+lon_wrap	- geografska duljina prekida
+no_defs	- ne koristiti datoteku /usr/share/proj/proj_def.dat
+over	- dopušta geografske duljine izvan raspona -180 do 180, onemogućuje prekidanje
+pm	- alternativni početni meridijan (obično ime grada)

+proj	- ime projekcije (vidi `proj -l`)
+south	- južna polusfera UTM zone
+to_meter	- faktor za pretvaranje u metre
+towgs84	- tri ili sedam brojeva za transformaciju koordinata u datum WGS84 (vidi dolje)
+units	- metri, stope, itd.
+vto_meter	- faktor za pretvaranje visina u metre.
+vunits	- jedinice za visine.
+x_0	- pomak istočno
+y_0	- pomak sjeverno
+zone	- zona UTM-a
+e	- prvi ekscentricitet elipsoida
+es	- kvadrat prvog ekscentriciteta elipsoida
+f	- sploštenost elipsoida
+R_A	- računa polumjer sfere rako da je površina sfere jednaka površini elipsoida
+rf	- inverzna sploštenost elipsoida

Navedimo primjer korištenja parametara PROJ.4 za opis HDKS-a (Hrvatskog Državnog Koordinatnog Sustava), koji još uvijek najviše korišten u Hrvatskoj iako je novi stupio na snagu. Podaci u ovom koordinatnom sustavu se nalaze na datumu Bessel 1841 i poprečnoj Mercatorovoj projekciji.

5. zona:

```
+proj=tmerc +pm=greenwich +lat_0=0 +lon_0=15 +k=0.9999 +x_0=5500000 +y_0=0
+ellps=bessel +towgs84=550.499,164.116,475.142,5.80967,2.07902,-11.62386,0.99999445824
+units=m
```

6.zona:

```
+proj=tmerc +pm=greenwich +lat_0=0 +lon_0=18 +k=0.9999 +x_0=6500000 +y_0=0
+ellps=bessel +towgs84=550.499,164.116,475.142,5.80967,2.07902,-11.62386,0.99999445824
+units=m
```

Za svaki referentni koordinatni sustav postoji datoteka EPSG-a koja sprema parametre ovog sustava.

5.2. Geodetic Parameter Dataset

EPSG Geodetic Parameter Dataset (URL 18) je strukturirana datoteka koja se koristi za:

- identifikaciju referentnog koordinatnog sustava (CRS).
- definira transformacije koje omogućavaju konverziju koordinata iz jednog CRS-a u drugi CRS

Svaki CRS ima svoj broj, na primjer EPSG:4326 je za datum WSG84. Ova datoteka je slobodno dostupna na internetu (URL 19).

5.3 CRS za Hrvatsku

Hrvatska ima nacionalne prostorne podatke u različitim referentnim koordinatnim sustavima. U tablici 2. je naveden popis referentnih koordinatnih sustava za Hrvatsku i njihovi EPSG-kodovi.

Tablica 2 . Referentni koordinatni sustavi i njihovi EPSG-kodovi.

CRS	EPSG
HTRS96 / Croatia LCC	EPSG:3766
HTRS96 / Croatia TM	EPSG:3765
MGI / Balkans zone 6	EPSG:31276
MGI / Balkans zone 5	EPSG:31275
WGS 84/UTM zone 34N	EPSG:32634
WGS 84/UTM zone 33N	EPSG:32633
WGS 84/Geodetic	EPSG:4326.
HTRS96/Geodetic	EPSG:4761
HTRS96 /UTM zone 33N	EPSG:3767
HTRS96 /UTM zone 34N	EPSG:3768
MGI/Geodetic	EPSG:4805

Dolje su navedeni podaci za svaki navedeni EPSG-kod, koji su koriste za transformaciju koordinata u Hrvatskoj.

EPSG:3766**HTRS96 / Croatia LCC**

- **WGS84 Bounds:** 13.0200, 42.3000, 19.5000, 46.5700
 - **Projected Bounds:** -286997.6531, 5172974.1032, 247430.9705, 5648564.1259
 - **Scope:** Cadastral surveys, large scale topographic mapping, engineering survey.
 - **Last Revised:** 2007-09-29
 - **Area:** Croatia
- `+proj=lcc +lat_1=45.91666666666666 +lat_2=43.08333333333334 +lat_0=0 +lon_0=16.5 +x_0=0 +y_0=0 +ellps=GRS80 +towgs84=0,0,0,0,0,0 +units=m +no_defs`

EPSG:3765**HTRS96 / Croatia TM**

- **WGS84 Bounds:** 13.4700, 42.3900, 19.5000, 46.5700
 - **Projected Bounds:** 250515.0793, 4698849.3024, 747014.5638, 5163391.4419
 - **Scope:** Cadastral surveys, large scale topographic mapping, engineering survey.
 - **Last Revised:** 2007-09-29
 - **Area:** Croatia – onshore
- `+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 +units=m +no_defs`

EPSG:31276**MGI / Balkans zone 6**

- **WGS84 Bounds:** 16.5000, 41.8500, 19.5000, 46.5500
 - **Projected Bounds:** 6375522.8666, 4635097.0616, 6624637.4183, 5157307.2566
 - **Scope:** Large and medium scale topographic mapping and engineering survey.
 - **Last Revised:** 2005-05-27
 - **Area:** Europe - former Yugoslavia - 16.5°E to 19.5°E
- `+proj=tmerc +lat_0=0 +lon_0=18 +k=0.9999 +x_0=6500000 +y_0=0 +ellps=bessel +towgs84=577.326,90.129,463.919,5.137,1.474,5.297,2.4232 +units=m +no_defs`

EPSG:31275**MGI / Balkans zone 5**

- **WGS84 Bounds:** 13.3800, 43.0100, 16.5000, 46.8800
- **Projected Bounds:** 5367990.9196, 4763954.9999, 5622335.5699, 5194181.2805
- **Scope:** Large and medium scale topographic mapping and engineering survey.
- **Last Revised:** 2008-04-04
- **Area:** Europe - former Yugoslavia - west of 16.5°E
- `+proj=tmerc +lat_0=0 +lon_0=15 +k=0.9999 +x_0=5500000 +y_0=0 +ellps=bessel +towgs84=577.326,90.129,463.919,5.137,1.474,5.297,2.4232 +units=m +no_defs`

EPSG:32634**WGS 84 / UTM zone 34N**

- **WGS84 Bounds:** 18.0000, 0.0000, 24.0000, 84.0000
- **Projected Bounds:** 166021.4431, 0.0000, 833978.5569, 9329005.1825
- **Scope:** Large and medium scale topographic mapping and engineering survey.
- **Last Revised:** 1995-06-02
- **Area:** World - N hemisphere - 18°E to 24°E - by country
- `+proj=utm +zone=34 +ellps=WGS84 +datum=WGS84 +units=m +no_defs`

EPSG:32633**WGS 84 / UTM zone 33N**

- **WGS84 Bounds:** 12.0000, 0.0000, 18.0000, 84.0000
- **Projected Bounds:** 166021.4431, 0.0000, 833978.5569, 9329005.1825
- **Scope:** Large and medium scale topographic mapping and engineering survey.
- **Last Revised:** 1995-06-02
- **Area:** World - N hemisphere - 12°E to 18°E - by country
- `+proj=utm +zone=33 +ellps=WGS84 +datum=WGS84 +units=m +no_defs`

EPSG:4326**WGS 84 /Geodetic**

- **WGS84 Bounds:** -180.0000, -90.0000, 180.0000, 90.0000
- **Projected Bounds:** -180.0000, -90.0000, 180.0000, 90.0000

- **Scope:** Horizontal component of 3D system. Used by the GPS satellite navigation system and for NATO military geodetic surveying.
- **Last Revised:** 2007-08-27
- **Area:** World
- `+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs`

EPSG:4761

HTRS96 /Geodetic

- **WGS84 Bounds:** 13.0200, 42.3000, 19.5000, 46.5700
- **Projected Bounds:** 13.0200, 42.3000, 19.5000, 46.5700
- **Scope:** Horizontal component of 3D system.
- **Last Revised:** 2007-09-29
- **Area:** Croatia
- `+proj=longlat +ellps=GRS80 +towgs84=0,0,0,0,0,0 +no_defs`

EPSG:4805

MGI /Geodetic

- **WGS84 Bounds:** 9.5300, 40.8400, 23.0000, 49.0200
- **Projected Bounds:** 27.1967, 40.8400, 40.6667, 49.0200
- **Scope:** Geodetic survey.
- **Last Revised:** 2004-01-06
- **Area:** Europe - Austria and former Yugoslavia
- `+proj=longlat +ellps=bessel +pm=ferro +no_defs`

5.4. Kreiranje WPS-procesa za transformaciju koordinata

Proces je napisan u programskom jeziku Python. Spremljen je u datoteci *transformkoordinata.py*.

On dobiva četiri ulazna parametra (EPSG-kod ulaznog CRS-a, EPSG-kod izlaznog CRS-a i dvije koordinate (geografsku širinu i duljnu ili koordinate u ravnini). Transformacija koordinata je realizirana pomoću modula pyProj. Kao i svaki WPS-proces i ovaj ima dva dijela. Prvi dio kôda je deklarativni koji definira inicijalizaciju procesa odnosno količinu i tipove ulaznih parametra te

količinu i tip izlaznih parametara. Drugi dio koda je izvršni koji će biti pokretan kada pyWPS dobije zahtjev *Execute* za transformaciju koordinata. Izvorni kôd procesa je:

```
"""
Transformacija koordinata za Hrvatsku
Datoteka transformkoordinata.py je smještena u C:\wps\processes
"""
from pywps.Process import WPSProcess
import pyproj
class Process(WPSProcess):
    def __init__(self):
        # Prvi dio procesa
        WPSProcess.__init__(self,
            identifier = "transformkoordinata", # mora biti ista kao i naziv datoteke
            title="Transformation Process",
            version = "0.1",
            storeSupported = "true",
            statusSupported = "true",
            abstract="The process will accept 4 input numbers and will return the XML result with new
            coordinates",
            grassLocation =False)
        # Definirano je četiri ulazna parametra
        self.Input1 = self.addLiteralInput(identifier = "cordulaz",
            title = "Ulazni CRS",
            type=type(""),
            default="epsg:4326")
        self.Input2 = self.addLiteralInput(identifier = "cordizlaz",
            title = "Izlazni CRS",
            type=type(""),
            default="epsg:4764")
        self.Input3= self.addLiteralInput(identifier="Lat",
            title="Latitude",
            type=type(0.0),
            default=45.808497)
        self.Input4= self.addLiteralInput(identifier="Lon",
```

```
title="Longitude",
type=float,
default=15.962997)
# Defenicija izlaznih parametra
self.Output1=self.addLiteralOutput(identifier="output1",title="Koordinata X")
self.Output2=self.addLiteralOutput(identifier="output2",title="Koordinata Y" )
# Drugi dio procesa
def execute(self):
cordulaz=self.Input1.getValue()
cordizlaz=self.Input2.getValue()
lat=self.Input3.getValue()
lon=self.Input4.getValue()
p1=pyproj.Proj(init=cordulaz)
p2=pyproj.Proj(init=cordizlaz)
x2,y2 = pyproj.transform(p1,p2,lon,lat)
self.Output1.setValue(x2)
self.Output2.setValue(y2)
return
```

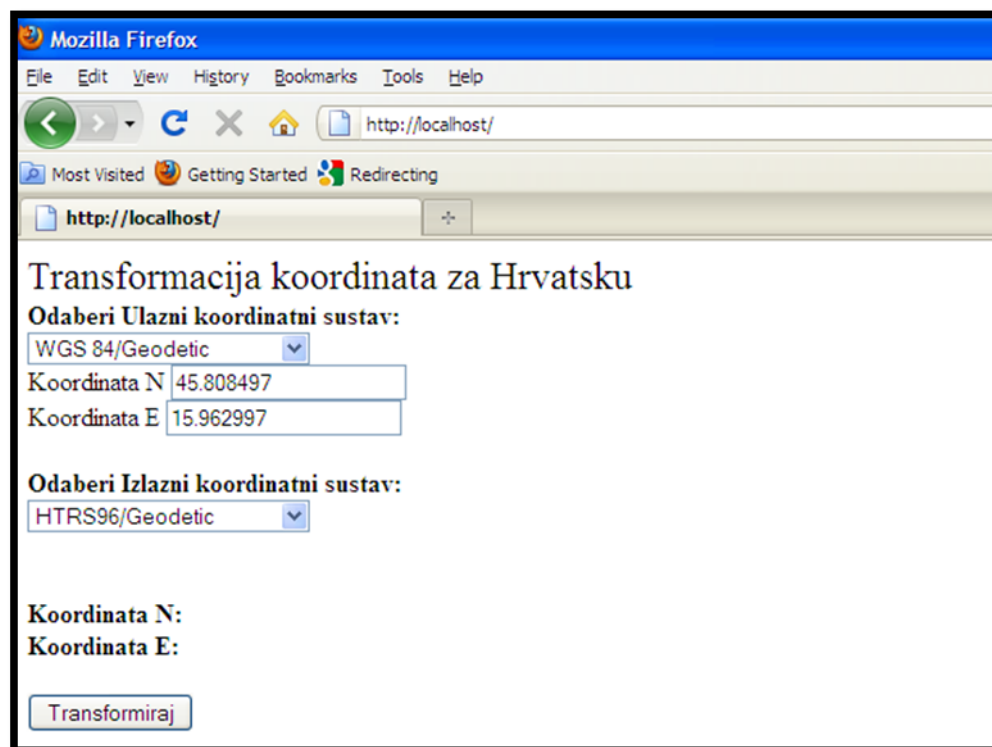
5.5. Jednostavni klijent za WPS-servis za transformaciju koordinata

Za testiranje procesa transformacije koordinata možemo koristiti bilo koji preglednik, na primjer Firefox. Pokrenemo Firefox i otvaramo stranicu `http://localhost` (slika 12).



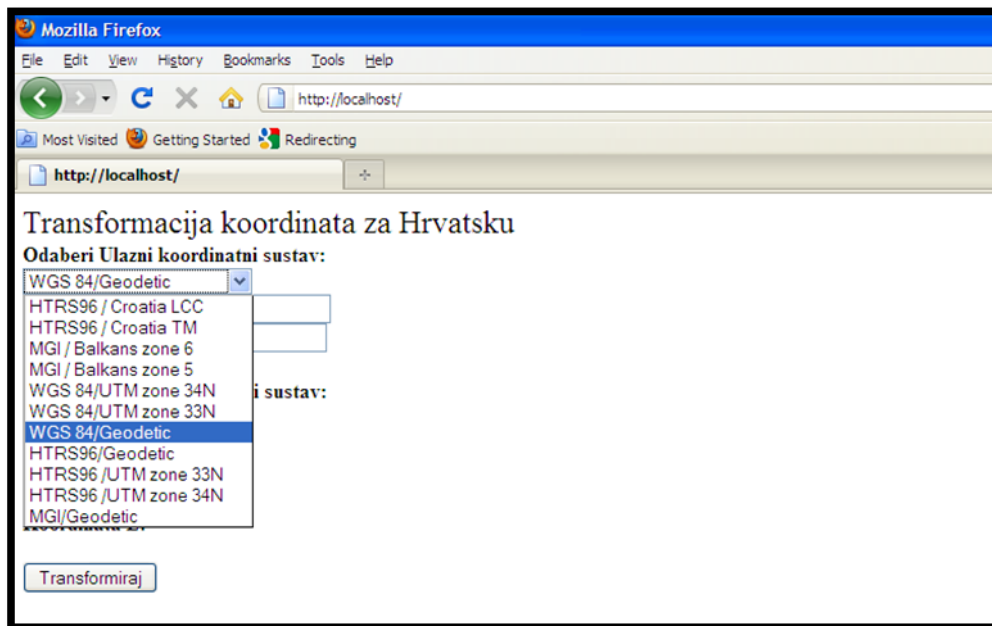
Slika 12. Adresa processa.

Server odgovara pokretanjem WPS-klijenta (`index.html`) za transformaciju (slika 13).

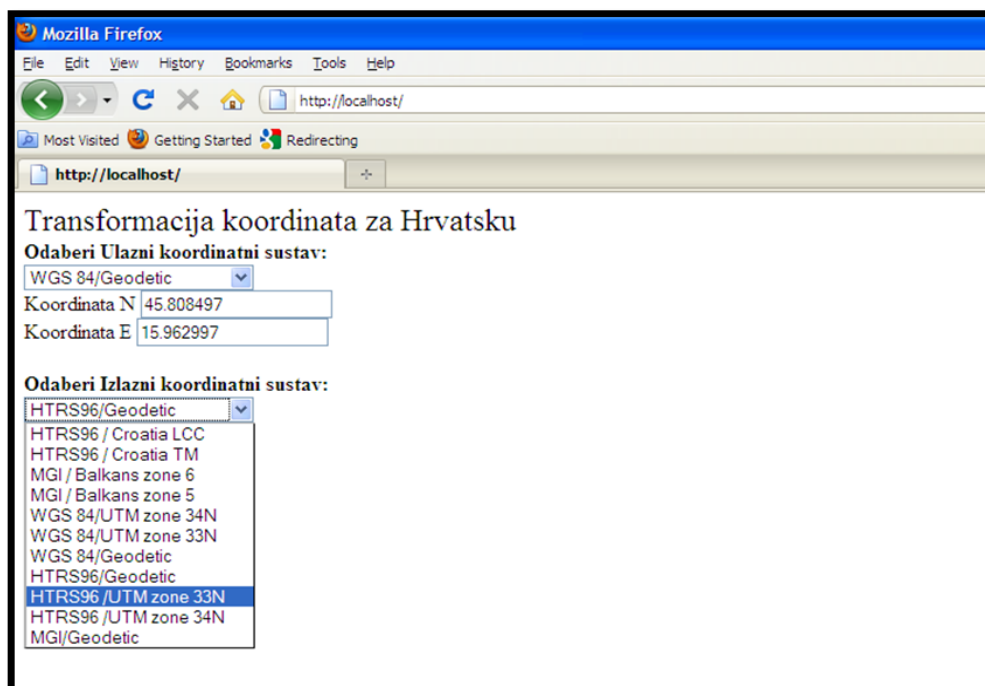


Slika13. WPS-klijent za transformaciju.

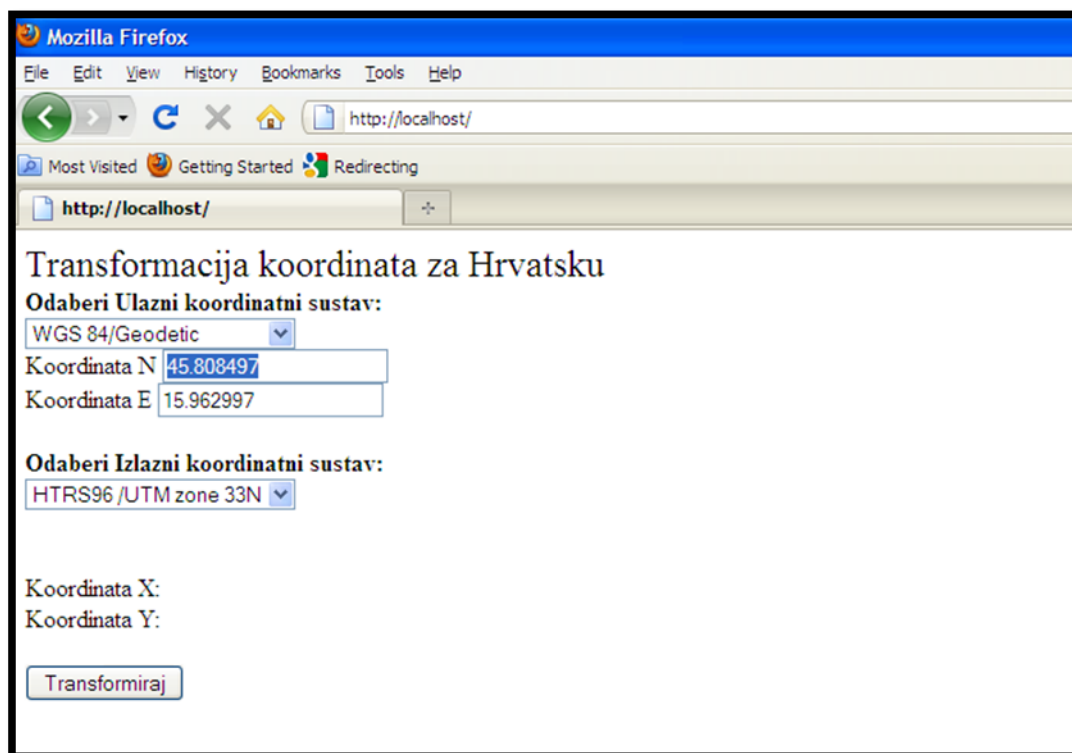
Sada možemo odabrati ulazni referentni koordinatni sustav (slika 14), izlazni koordinatni sustav (slika 15) i upisati koordinatu koju trebamo transformirati (slika 16).



Slika 14. Odabir ulaznog koordinatnog sustava.

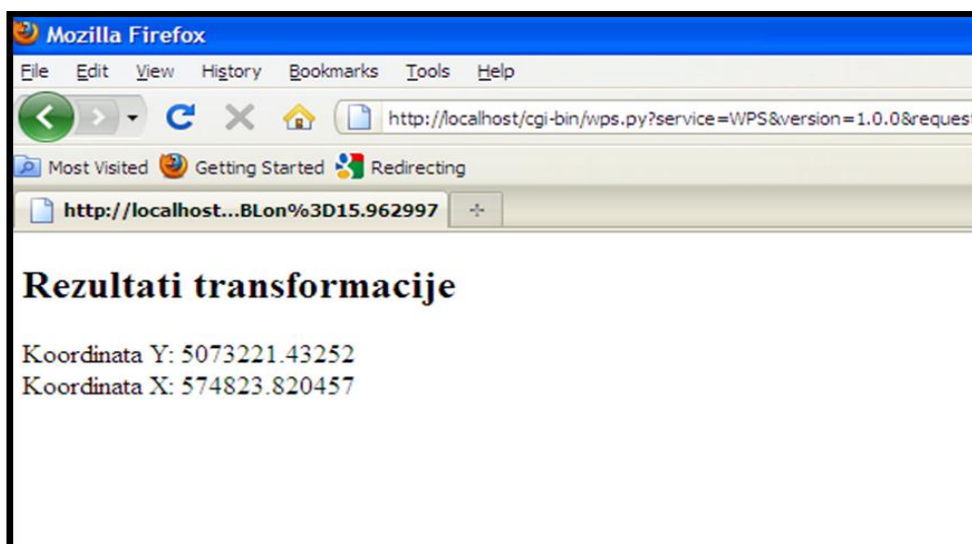


Slika 15. Odabir izlaznog koordinatnog sustava.



Slika 16. Ulazne koordinate za transformaciju.

Sada treba izabrati Transformiraj i dobit će se rezultat (slika 17).



Slika 17. Rezultat transformacije.

WPS-klijent za transformaciju koordinata je pisan u programskom jeziku JavaScript. Smješten je u datoteci `C:\Apache2\htdocs\index.html`.

Sadržaj te datoteke je sljedeći:

```
<!DOCTYPE HTML>
<html>
<head><font size=5>Transformacija koordinata za Hrvatsku </font>
<meta http-equiv="Content-Script-Type" content="text/javascript"/>
<script language="JavaScript">
function cordfun() {
var i1=document.ulaz.cordulaz.selectedIndex;
var s1=document.ulaz.cordulaz.options[i1].value;
var i2=document.ulaz.cordizlaz.selectedIndex;
var s2=document.ulaz.cordizlaz.options[i2].value;
var s4=document.getElementById("lat1").value;
var s5=document.getElementById("lon1").value;
var s3="cordulaz="+s1+";cordizlaz="+s2+";Lat="+s4+";Lon="+s5;
document.new.datainputs.value=s3;
return ;
}
function Make() {
var s7=document.ulaz.cordulaz.selectedIndex;
if (s7==6 || s7==7){
document.getElementById("lab1").innerHTML="Koordinata N";
document.getElementById("lab2").innerHTML="Koordinata E";
}
else {
document.getElementById("lab1").innerHTML="Koordinata X";
document.getElementById("lab2").innerHTML="Koordinata Y";
}
return;
}
function Make1() {
var s8=document.ulaz.cordizlaz.selectedIndex;
if (s8==6 || s8==7){
document.getElementById("lab3").innerHTML="<b>Koordinata N:</b>";
document.getElementById("lab4").innerHTML="<b>Koordinata E:</b>";
}
}
```



```
else {
document.getElementById("lab3").innerHTML="Koordinata X:";
document.getElementById("lab4").innerHTML="Koordinata Y:";
}
return;
}
</script>
</head>
<body>
<form name="ulaz">
<b>Odaberi Ulazni koordinatni sustav:</b><br>
<select name="cordulaz" size="1" onchange="Make()">
<option value="epsg:3766">HTRS96 / Croatia LCC </option>
<option value="epsg:3765">HTRS96 / Croatia TM</option>
<option value="epsg:31276">MGI / Balkans zone 6</option>
<option value="epsg:31275">MGI / Balkans zone 5</option>
<option value="epsg:32634">WGS 84/UTM zone 34N</option>
<option value="epsg:32633">WGS 84/UTM zone 33N</option>
<option value="epsg:4326" selected>WGS 84/Geodetic </option>
<option value="epsg:4761">HTRS96/Geodetic</option>
<option value="epsg:3767">HTRS96 /UTM zone 33N</option>
<option value="epsg:3768">HTRS96 /UTM zone 34N</option>
<option value="epsg:4805">MGI/Geodetic</option>
</select>
<br/>
<label for="lat1" id="lab1">Koordinata N</label>
<input type="text" name="lat" value="45.808497" id="lat1"/>
<br/>
<label for="lon1" id="lab2">Koordinata E</label>
<input type="text" name="lon" value="15.962997" id="lon1"/>
<br/>
<br/>
<b>Odaberi Izlazni koordinatni sustav:</b><br>
<select name="cordizlaz" size="1" onchange='Make1()'
```

```
<option value="epsg:3766">HTRS96 / Croatia LCC</option>
<option value="epsg:3765">HTRS96 / Croatia TM</option>
<option value="epsg:31276">MGI / Balkans zone 6</option>
<option value="epsg:31275">MGI / Balkans zone 5</option>
<option value="epsg:32634">WGS 84/UTM zone 34N</option>
<option value="epsg:32633">WGS 84/UTM zone 33N</option>
<option value="epsg:4326" >WGS 84/Geodetic </option>
<option value="epsg:4761" selected>HTRS96/Geodetic</option>
<option value="epsg:3767">HTRS96 /UTM zone 33N</option>
<option value="epsg:3768">HTRS96 /UTM zone 34N</option>
<option value="epsg:4805">MGI/Geodetic</option>
</select>
</form>
<form name="new" method="get" action="http://localhost/cgi-bin/wps.py">
<br/>
<br/>
<input type="hidden" name="service" value="WPS"/>
<input type="hidden" name="version" value="1.0.0"/>
<input type="hidden" name="request" value="Execute"/>
<input type="hidden" name="identifier" value="transformkoordinata"/>
<input type="hidden" name="datainputs" value=""/>
<label id="lab3"><b>Koordinata N:</b></label>
<br/>
<label id="lab4"><b>Koordinata E:</b></label>
<br/>
<br/>
<input type="submit" value="Transformiraj" onclick="cordfun()"/>
</form>
</body>
</html>
```

6. ZAKLJUČAK

Prostorni podaci mogu imati različite referentne koordinatne sustave (CRS), a ponekad ih treba transformirati u neki drugi koordinatni sustav kako bi se mogli integrirano analizirati i prikazivati.

INSPIRE propisuje pet mrežnih servisa za prostorne podatke. Jedan od njih je servis za transformacije. Poseban slučaj servisa za transformacije je servis za transformacije koordinata.

Države članice Europske unije odgovorne su za uspostavu i održavanje tih servisa.

Izrađeni servis za transformacijekoordinata u okviru ovog diplomskog rada je u skladu s standardom Open GIS konzorcija (OGC) koji se odnosi na mrežne servise za obradu (web processing services – WPS).

Upotreba takvog servisa omogućuje državama članicama da nacionalne podatke održavaju u nacionalnim koordinatnim sustavima, a kad je to potrebno, podaci se mogu transformirati u jedinstveni koordinatni sustav.

Hrvatska ima nacionalne prostorne podatke u različitim referentnim koordinatnim sustavima.

Ovaj rad je pokazao jedan od mogućih načina rješavanja takvog zadatka

Transformacije koordinata danas su na raspolaganju u većini GIS softvera. Međutim, transformacije u tim softverskim sustavima mogu raditi samo u vlastitoj okolini.

Ovaj rad pokazuje, na koji je način moguće napraviti transformacije koordinata s pomoću web-servisa koji je dostupan svim programima koji imaju klijent za WPS.

Za buduća istraživanja treba provesti usklađivanje sa specifikacijama WCTS-a (Web Coordinate Transformation Service). Nadalje potrebno je razviti proces koji može prihvatiti i transformirati podatke u različitim formatima. Također poželjno je razviti vlastiti web-klijent koji može prihvatiti i poslati i druge oblike datoteka s koordinatama.

LITERATURA

Deitel, P.J. Deitel, T.R. (2002): Internet and World Wide Web How to program, Prentice Hall

ISO : International standard ISO19111 (2003): Geographic information-Spatial referencing by coordinates, ISO copyright office, Geneva, 1-43

Husak, M., Lapaine, M. (2006): Odluka o utvrđivanju službenih geodetskih datuma i kartografskih projekcija Republike Hrvatske

Lapaine M., Tutić D., (2007): O novoj službenoj kartografskoj projekciji Hrvatske-HTRS96/TM, Kartografija i Geoinformacije, poseban broj. Hrvatsko kartografsko društvo, str.34-53

Longley, P.A., Goodchild, M.F., Maguire J. W., Rhind, D.J. (2011): Geographic Information System & Science, Wiley Publishing, Inc.

Michael Morrison (2010): Sams Teach Yourself XML in 24 Hours, Third Edition, Sams Publishing

URL 1. All Registered Product <http://www.opengeospatial.org/resource/products>, (05.01.2012)

URL 2. Open Geospatial consortium, http://en.wikipedia.org/wiki/Open_Geospatial_Consortium, (11.02.2012)

URL 3. Web Processing Service, <http://www.opengeospatial.org/standards/wps>, (11.02.2012)

URL 4. Hypertext Transfer Protocol, http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol, (12.03.2012)

URL 5. World Wide Web Consortium (W3C), <http://www.w3.org/>, (18.03.2012)

URL 6. Wikipedia: SOAP, <http://en.wikipedia.org/wiki/SOAP>, (30.02.2012)

URL 7. Wikipedia: REST, http://en.wikipedia.org/wiki/Representational_State_Transfer, (21.04.2012)

URL 8. Wikipedia: XML, <http://en.wikipedia.org/wiki/XML>, (30.03.2012)

URL 9. Wikipedia:

SGML, http://en.wikipedia.org/wiki/Standard_Generalized_Markup_Language, (27.04.2012)

URL 10. The Apache Software Foundation, <http://www.apache.org/> (20.04.2012)

-
- URL 11. Python Programming Language, <http://www.python.org/>, (20.04.2012)
- URL 12. The Python Standard Library, <http://docs.python.org/library/>, (20.04.2012)
- URL 13. OGC® Standards and Supporting Documents,
<http://www.opengeospatial.org/standards/>, (15.03.2012)
- URL 14. The OpenGIS® Web Processing Service (WPS) Interface Standard,
<http://www.opengeospatial.org/standards/wps>, (15.03.2012)
- URL 15. Apache HTTP Server 2.2.22 (httpd), <http://httpd.apache.org/download.cgi#apache24>,
(25.04.2012)
- URL 16. MGI / Balkans coordinate systems, http://spatial-analyst.net/wiki/index.php?title=MGI/_Balkans_coordinate_systems, (23.04.2012)
- URL 17. PROJ.4 - Cartographic Projections Library, <http://trac.osgeo.org/proj/>, (25.04.2012)
- URL 18. EPSG Geodetic Parameter Dataset, <http://www.epsg.org/geodetic.html>, (27.04.2012)
- URL 19. Epsg projection, <http://spatialreference.org>, (27.04.2012)

Popis slika

Slika 1. Implementacija WPS servera.....	7
Slika 2. Početak instaljacije.....	30
Slika 3. Podatke servera.....	30
Slika 4. Odabir mjesta instaljaciji Apache.....	31
Slika 5. . Završetak instaljaciji.....	31
Slika 6. . Odabir Open Apache Monitor.....	32
Slika 7. . Prozor Apache Monitora.....	32
Slika 8. Početni izbornik instalcijskog čarobnjaka za Python.....	33
Slika 9. Izbornik instalacije s mogućnošću odabira mjesta i direktorija instalacije.....	34
Slika 10. Izbornik koji dozvoljava odabir pojedinih modula Pythona za instaliranje.....	35
Slika 11. Dodavanje i uređivanje sistemskih varijabli i staza.....	36
Slika 12. Lansiranje geoprocessa.....	50
Slika 13. WPS-klijent za transformaciju.....	51
Slika 14. Odabir ulaznog koordinatnog sustava.....	51
Slika 15. Odabir izlaznog koordinatnog sustava.....	52
Slika 16. Uvod coordinate za transformaciju.....	52
Slika 17. Rezultat transformacije.....	53

Popis tablica

Tablica 1. Dodatne biblioteke.....	36
Tablica 2. Referentni koordinatni sustavi i njihovi EPSG-kodovi.....	46

Sadržaj priloženog optičkog medija

1. DiplomskiRad.doc: Tekst diplomskog rada
2. Aplikacija - index.html: WPS-klijent
3. Aplikacija - transformkoordinata.py: WPS-proces za transformaciju koordinata
- 4 .PyWPS modul: Python implementacija WPS-standarda