



جمهوری اسلامی ایران
سازمان برنامه و بودجه کشور
سازمان نقشه برداری کشور

دستورالعمل پیاده سازی تحلیل های مکانی تحت وب (WPS)

دستورالعمل شماره ———

آخرین ویرایش: ۱۴۰۴-۰۴-۰۴

<http://www.ncc.gov.ir>

سازمان نقشه برداری کشور

اداره تحلیل های مکان مبنا

اداره کل سامانه های ملی و

زیرساخت های اطلاعات مکانی

تیر ۱۴۰۴

پیشگفتار

در سالهای اخیر، اهمیت تحلیل‌های مکان‌مبنا، به‌ویژه تحلیل‌های مکانی تحت وب و برخط، بیش از گذشته آشکار شده است. با این حال، این تحلیل‌ها به‌دلیل مفاهیم پیچیده ریاضی، دشواری در پیاده‌سازی و کمبود منابع آموزشی جامع، برای بسیاری از کاربران غیرقابل دسترس بوده‌اند. از این‌رو، تدوین سندی که هم به مبانی نظری بپردازد و هم جنبه‌های عملی پیاده‌سازی را پوشش دهد، ضروری به نظر می‌رسد.

مستند حاضر با رویکردی اجرایی و با بهره‌گیری از تجربیات کارشناسان سازمان نقشه‌برداری کشور تهیه شده است. هدف آن تسهیل درک و استفاده از سرویس‌های پردازشی تحت وب است تا زمینه را برای توسعه و به‌کارگیری این فناوری در پروژه‌های مکانی فراهم آورد. در پایان، از همکاری تمامی افرادی که در تهیه این سند مشارکت داشته‌اند، قدردانی می‌شود.

همچنین سپاسگزاری ویژه‌ای از جناب آقای مهندس رضا احمدیه به عمل می‌آید که در سال‌های گذشته با نظارت دقیق بر نسخه اولیه این سند، ارائه پیشنهادات اصلاحی و همراهی ارزشمند خود، نقش مؤثری در ارتقای کیفیت آن ایفا نمودند.

تهیه و کنترل « دستورالعمل پیاده‌سازی تحلیل‌های مکانی تحت وب (WPS) »

اعضای گروه تهیه‌کننده:

کارشناسی ارشد GIS	سازمان نقشه‌برداری کشور	میلاذ رفیعی طرقي
دانشجوی دکتری GIS	سازمان نقشه‌برداری کشور	محسن یوسف زاده سورکی
دکتری RS & GIS	سازمان نقشه‌برداری کشور	نیما قاسملو
دکتری RS & GIS	سازمان نقشه‌برداری کشور	زهرا رضائی

اعضای گروه نظارت :

کارشناسی ارشد IT	سازمان نقشه‌برداری کشور	مهدی سعیدی انجیله
------------------	-------------------------	-------------------

اعضای گروه هدایت و راهبری:

کارشناسی ارشد GIS	سازمان نقشه‌برداری کشور	پیمان بکتاش
کارشناسی ارشد RS & GIS	سازمان نقشه‌برداری کشور	مریم صارمی

فهرست مطالب

فصل اول	۱
۱-۱- مقدمه	۳
۲-۱- تحلیل مکانی	۳
۳-۱- چالش‌های موجود در تحلیل‌های مکانی	۵
۱-۳-۱- حجم زیاد داده‌ها (Big Data)	۵
۲-۳-۱- تنوع داده‌ها و فرمت‌های مختلف	۶
۳-۳-۱- کارایی و مقیاس‌پذیری پردازش‌ها	۶
۴-۳-۱- یکپارچگی و ارتباط بین داده‌ها	۶
۵-۳-۱- دقت و صحت داده‌ها	۶
۶-۳-۱- قابلیت تعامل در سیستم‌های توزیع‌شده	۷
۷-۳-۱- پردازش و ذخیره‌سازی داده‌های سه‌بعدی و پیچیده	۷
۴-۱- مقدمه‌ای بر تحلیل‌های مکانی دسکتاپ و برخط	۷
۵-۱- مزایا و معایب تحلیل‌های مکانی دسکتاپ و برخط	۸
۱-۵-۱- تحلیل‌های مکانی دسکتاپ	۸
۲-۵-۱- تحلیل‌های مکانی برخط	۹
۶-۱- بررسی چالش‌های فنی و زیرساختی در پیاده‌سازی WPS	۹
۱-۶-۱- چالش‌های فنی در پیاده‌سازی WPS	۱۰
۲-۶-۱- چالش‌های زیرساختی در پیاده‌سازی WPS	۱۰
۷-۱- مجموعه استاندارد OGC در داده‌های مکانی	۱۱
۱-۷-۱- Web Map Service (WMS)	۱۲
۲-۷-۱- Web Feature Service (WFS)	۱۲
۳-۷-۱- Web Coverage Service (WCS)	۱۲
۴-۷-۱- Web Processing Service (WPS)	۱۲
۵-۷-۱- Catalogue Service for the Web (CSW)	۱۳
۶-۷-۱- Sensor Observation Service (SOS)	۱۳

۱۳	 Web Map Tile Service (WMTS) -۷-۷-۱
۱۳	 پیاده‌سازی‌های مختلف WPS و ارزیابی آنها -۸-۱
۱۵	 نیازمندی‌ها و مراحل پیاده‌سازی استاندارد WPS در محیط ژئوپورتال -۹-۱
۱۵	 WPS -۱۰-۱ در ژئوپورتال سازمان نقشه برداری کشور

فصل دوم ۱۸

۲۰	 WPS -۱-۲ استاندارد
۲۱	 WPS -۲-۲ پیکربندی
۲۱	 Service Metadata -۱-۲-۲
۲۳	 Execution and resource settings -۲-۲-۲
۲۴	 Process status page -۳-۲-۲
۲۶	 WPS security page -۴-۲-۲
۲۹	 دامنه سرویس -۳-۲
۲۹	 WPS -۴-۲ نمای کلی
۲۹	 WPS -۵-۲ توابع
۳۰	 GetCapabilities -۱-۵-۲
۳۰	 DescribeProcess -۲-۵-۲
۳۲	 Execute -۳-۵-۲
۳۳	 WPS -۶-۲ تحلیل
۳۳	 Process chaining -۱-۶-۲
۳۳	 WPS Request Builder -۲-۶-۲
۳۵	 WPS -۷-۲ ماهیت عمومی و میان افزار

فصل سوم ۳۷

۳۹	 WPS -۱-۳ راه اندازی سرویس در ژئوسرور
۴۱	 WPS -۲-۳ نحوه پیاده سازی سرویس در ژئوسرور

فصل چهارم ۴۳

۴۵	 Buffer -۱-۴ تحلیل
۴۹	 Intersection -۲-۴ تحلیل

۵۳	۳-۴ تحلیل Clip
۵۵	۴-۴ تحلیل Nearest
۵۷	۵-۴ تحلیل Within
۵۹	۶-۴ تحلیل Contains
۶۱	۷-۴ تحلیل Contour
۶۳	۸-۴ تحلیل Query

۶۵ فصل پنجم

۶۷	۱-۵ اجرای یک پروژه موردی
----	--------------------------

۷۵ فصل ششم

۷۷	۱-۶ فرآیند (process) در استاندارد OGC
۷۷	۲-۶ ورودی و خروجی داده در استاندارد OGC
۷۷	۱-۲-۶ LiteralData
۷۸	۲-۲-۶ ComplexData
۷۸	۳-۲-۶ BoundingBoxData
۷۸	۳-۶ انتقال داده به فرآیند (passing data to process)
۷۸	۴-۶ درخواست فرآیند به صورت همزمان و غیرهمزمان
۷۸	۱-۴-۶ حالت همزمان (Synchronous mode)
۷۹	۲-۴-۶ حالت غیرهمزمان (Asynchronous mode)
۷۹	۵-۶ وضعیت فرایند (process Status)
۷۹	۶-۶ کدگذاری درخواست HTTP GET و POST
۷۹	۱-۶-۶ جفت کلید-مقدار (Key-value pairs)
۸۰	۲-۶-۶ محتوای XML (XML payload)
۸۰	۷-۶ نصب و راه اندازی PyWPS
۸۰	۸-۶ فرآیندها
۸۰	۱-۸-۶ نوشتن یک فرایند (writing a process)
۸۱	۹-۶ پیاده‌سازی نمونه اجرایی بر اساس PyWPS

فهرست اشکال

شکل ۱- برهم نهی لایه های مختلف اطلاعات مکانی و استخراج نقشه موضوعی از نتایج آن.....	۳
شکل ۲- ساختار موجود در ژئوپرتال سازمان نقشه برداری کشور.....	۱۶
شکل ۳- ساختار پیشنهادی.....	۱۷
شکل ۴- استاندارد WPS.....	۲۰
شکل ۵- دیاگرام سرویس های WFS ، CSW و WPS.....	۲۱
شکل ۶- SERVICE METADATA.....	۲۲
شکل ۷- EXECUTION & RESOURCE SETTINGS.....	۲۳
شکل ۸- PROCESS STATUS.....	۲۴
شکل ۹- WPS SECURITY.....	۲۶
شکل ۱۰- ROLES.....	۲۷
شکل ۱۱- WPS SECURITY : MANAGE.....	۲۷
شکل ۱۲- نمونه ای از محدودیت ها.....	۲۸
شکل ۱۳- پاسخ درخواست GETCAPABILITY.....	۳۰
شکل ۱۴- درخواست DESCRIBEPROCESS.....	۳۱
شکل ۱۵- نمودار UML رابط WPS.....	۳۲
شکل ۱۶- PROCESS CHAINING.....	۳۳
شکل ۱۷- GEOSERVER DEMOS.....	۳۳
شکل ۱۸- WPS REQUEST BUILDER.....	۳۴
شکل ۱۹- EXECUTE PROCESS IN NEW PAGE.....	۳۴
شکل ۲۰- صفحه دانلود GEOSERVER.....	۳۹
شکل ۲۱- دانلود نسخه های GEOSERVER.....	۳۹
شکل ۲۲- صفحه دانلود افزونه های WPS.....	۴۰
شکل ۲۳- قرار دادن افزونه WPS در مسیر LIB در نرم افزار GEOSERVER.....	۴۰
شکل ۲۴- استفاده از ماژول DEMOS برای ساخت درخواست های GEOSERVER.....	۴۱
شکل ۲۵- انتخاب گزینه WPS REQUEST BUILDER از ماژول DEMOS.....	۴۱
شکل ۲۶- انتخاب تحلیل مورد نظر منوی CHOOSE PROCESS.....	۴۲
شکل ۲۷- انتخاب تحلیل BUFFER از منوی CHOOSE PROCESS.....	۴۵
شکل ۲۸- انتخاب نوع ورودی از قسمت INPUT FEATURE COLLECTION.....	۴۵
شکل ۲۹- انتخاب لایه مورد نظر در حالت VECTOR_LAYER به عنوان لایه ورودی.....	۴۶
شکل ۳۰- انتخاب لایه مورد نظر در حالت TEXT به عنوان لایه ورودی.....	۴۶
شکل ۳۱- وارد کردن آدرس URL لایه ورودی مورد نظر در حالت REFERENCE.....	۴۷
شکل ۳۲- وارد کردن مقدار فاصله مورد نظر مربوط به تحلیل BUFFER.....	۴۷
شکل ۳۳- انتخاب فرمت استاندارد مورد نظر به عنوان خروجی تحلیل BUFFER.....	۴۷

شکل ۳۴- دریافت خروجی تحلیل BUFFER در زبان XML.....	۴۸
شکل ۳۵- نمایش انواع فرمت خروجی تحلیل‌های WPS در نرم‌افزار GEOSERVER.....	۴۸
شکل ۳۶- انتخاب تحلیل INTERSECTION از منوی CHOOSE PROCESS بخش WPS REQUEST BUILDER.....	۴۹
شکل ۳۷- انتخاب حالت SUBPROCESS به عنوان ورودی حالت اول در تحلیل INTERSECTION.....	۵۰
شکل ۳۸- انتخاب گزینه DEFINE/EDIT جهت معرفی پارامترهای تحلیل SUBPROCESS.....	۵۰
شکل ۳۹- انتخاب تحلیل مورد نظر به عنوان تحلیل دوم.....	۵۰
شکل ۴۰- انتخاب لایه ورودی از تحلیل BUFFER و تعیین مقدار فاصله.....	۵۱
شکل ۴۱- مشاهده کد XML حاصل از تحلیل BUFFER و انتخاب لایه دوم.....	۵۱
شکل ۴۲- انتخاب گزینه مربوطه به INTERSECTIONMODE.....	۵۲
شکل ۴۳- انتخاب فرمت استاندارد مورد نظر به عنوان خروجی تحلیل INTERSECTION.....	۵۲
شکل ۴۴- دریافت خروجی تحلیل INTERSECTION در زبان XML.....	۵۲
شکل ۴۵- انتخاب تحلیل CLIP از منوی CHOOSE PROCESS.....	۵۳
شکل ۴۶- انتخاب لایه ورودی و حالت CLIP GEOMETRY از تحلیل CLIP.....	۵۳
شکل ۴۷- انجام تنظیمات مورد نیاز CLIP GEOMETRY مربوط به تحلیل CLIP.....	۵۴
شکل ۴۸- انتخاب فرمت استاندارد مورد نظر به عنوان خروجی تحلیل CLIP.....	۵۴
شکل ۴۹- ایجاد فایل XML به عنوان خروجی تحلیل CLIP.....	۵۵
شکل ۵۰- انتخاب تحلیل NEAREST و لایه مورد نظر.....	۵۶
شکل ۵۱- انجام تنظیمات مورد نظر جهت وارد کردن مختصات نقطه‌ای و سیستم مختصات لایه مربوطه.....	۵۶
شکل ۵۲- معرفی کد EPSG سیستم مختصات و سیستم تصویر لایه مورد نظر.....	۵۶
شکل ۵۳- خروجی حاصل از تحلیل NEAREST.....	۵۷
شکل ۵۴- انتخاب تحلیل JTS:WITHIN.....	۵۷
شکل ۵۵- معرفی لایه اول به صورت TEXT در قالب GML3.1.....	۵۸
شکل ۵۶- تعیین نوع ورودی و معرفی آدرس URL لایه دوم.....	۵۸
شکل ۵۷- فایل XML حاصل از تحلیل WITHIN.....	۵۹
شکل ۵۸- انتخاب تحلیل CONTAINS و انجام تنظیمات مورد نظر.....	۶۰
شکل ۵۹- فایل XML حاصل از تحلیل CONTAINS.....	۶۰
شکل ۶۰- انتخاب تحلیل CONTOUR و انجام تنظیمات مورد نظر.....	۶۱
شکل ۶۱- انتخاب دکمه EXECUTE PROCESS اجرای تحلیل CONTOUR و نمایش خروجی حاصل از آن.....	۶۲
شکل ۶۲- فایل XML حاصل از تحلیل CONTAINS.....	۶۲
شکل ۶۳- انتخاب تحلیل GS:QUERY و تعیین لایه ورودی.....	۶۳
شکل ۶۴- وارد کردن کوئری مورد نظر.....	۶۴
شکل ۶۵- خروجی حاصل از تحلیل QUERY.....	۶۴
شکل ۶۶- سرویس WMS لایه رودخانه‌ها.....	۶۷
شکل ۶۷- سرویس WMS لایه روستاها.....	۶۸

شکل ۶۸- انتخاب تحلیل QUERY و وارد نمودن کوئری مورد نظر.....	۶۹
شکل ۶۹- نمایش رودخانه جاجرود.....	۶۹
شکل ۷۰- انتخاب تحلیل QUERY به عنوان ورودی تحلیل BUFFER.....	۷۰
شکل ۷۱- فایل XML حاصل از تحلیل QUERY به عنوان ورودی تحلیل BUFFER.....	۷۰
شکل ۷۲- سرویس نمایشی حاصل از تحلیل BUFFER بر روی رودخانه جاجرود.....	۷۱
شکل ۷۳- انتخاب تحلیل BUFFER به عنوان ورودی لایه دوم.....	۷۲
شکل ۷۴- وارد کردن کوئری مورد نظر به منظور انتخاب رودخانه جاجرود.....	۷۲
شکل ۷۵- خروجی حاصل از تحلیل فوق و نمایش روستاهای در معرض خطر سیل.....	۷۳
شکل ۷۶- خروجی XML تحلیل CHAIN.....	۷۴
شکل ۷۷- ایجاد محیط مجازی MYENV.....	۸۲
شکل ۷۸- فعالسازی محیط مجازی MYENV.....	۸۲
شکل ۷۹- نصب آفلاین کتابخانه‌ها.....	۸۳
شکل ۸۰- ایجاد پروژه DJANGO با نام WPS01.....	۸۴
شکل ۸۱- ساختار پروژه ی DJANGO.....	۸۴
شکل ۸۲- اجرای سرور توسعه ای DJANGO.....	۸۵
شکل ۸۳- مشاهده ی پروژه ی DJANGO در مرور گر.....	۸۵
شکل ۸۴- ایجاد یک APP در درون پروژه ی DJANGO.....	۸۵
شکل ۸۵- اضافه کردن نام APP01 به تنظیمات پروژه.....	۸۶
شکل ۸۶- اضافه کردن فایل پیکربندی PYWPS.CFG به پروژه.....	۸۶
شکل ۸۷- بخش هایی از پیکربندی شامل فراداده.....	۸۷
شکل ۸۸- اضافه کردن فایل LOG به پروژه.....	۸۷
شکل ۸۹- فراخوانی کلاسهای مورد نیاز برای فرایند SAYHELLO.....	۸۸
شکل ۹۰- تعریف ورودی و خروجی ها در فرایند SAYHELLO.....	۸۸
شکل ۹۱- مقداردهی اولیه فرآیند SAYHELLO.....	۸۸
شکل ۹۲- تعریف متد HANDLER به منظور انجام پردازش مورد نیاز.....	۸۹
شکل ۹۳- افزودن فرایند SAYHELLO به لیست فرایندهای فعال.....	۸۹
شکل ۹۴- اجرای سرور توسعه ای DJANGO.....	۹۰
شکل ۹۵- پاسخ سرور به درخواست GETCAPABILITIES.....	۹۱
شکل ۹۶- پاسخ سرور به درخواست DESCRIBEPROCESS.....	۹۲
شکل ۹۷- پاسخ سرور به درخواست EXECUTE.....	۹۲
شکل ۹۸- نصب GDAL با استفاده از WHEEL فایل.....	۹۳
شکل ۹۹- فراخوانی کلاسها و ماژول های مورد نیاز.....	۹۴
شکل ۱۰۰- تعریف ورودی و خروجی ها در فرایند بافر.....	۹۴
شکل ۱۰۱- مقداردهی اولیه فرآیند بافر.....	۹۵

شکل ۱۰۲- تعریف متد HANDLER به منظور انجام پردازش مورد نیاز.....	۹۵
شکل ۱۰۳- ذخیره سازی خروجی تحلیل بافر به صورت GML.....	۹۵
شکل ۱۰۴- ساختار کلاس OGR.....	۹۶
شکل ۱۰۵- افزودن فرایند SIMPLE_BUFFER به لیست فرایندهای فعال.....	۹۶
شکل ۱۰۶- پاسخ سرور به درخواست GETCAPABILITIES.....	۹۷
شکل ۱۰۷- پاسخ سرور به درخواست DESCRIBEPROCESS.....	۹۸
شکل ۱۰۸- پاسخ سرور به درخواست EXECUTE.....	۹۹
شکل ۱۰۹- تبدیل خروجی به متر با تغییر WGS84 به سیستم تصویر UTM.....	۹۹

فهرست جداول

۱۴	جدول ۱- پیاده سازی های WPS
۲۲	جدول ۲- المان های SERVICE METADATA
۲۳	جدول ۳- EXECUTION & RESOURCE SETTINGS
۲۵	جدول ۴- فازهای اجرایی در ژئوسرور
۲۹	جدول ۵- عملگرهای استاندارد WPS
۳۱	جدول ۶- اطلاعات موجود در پاسخ DESCRIBEPROCESS
۳۲	جدول ۷- PPIO
۳۶	جدول ۸- دسته بندی پردازش ها در ژئوسرور
۴۵	جدول ۹- تحلیل BUFFER
۴۹	جدول ۱۰- تحلیل INTERSECTION
۵۳	جدول ۱۱- تحلیل CLIP
۵۵	جدول ۱۲- تحلیل NEAREST
۵۷	جدول ۱۳- تحلیل WITHIN
۵۹	جدول ۱۴- تحلیل CONTAINS
۶۱	جدول ۱۵- تحلیل CONTOUR
۶۳	جدول ۱۶- تحلیل QUERY

فصل اول

مفاهیم پایه تحلیل های مکانی

۱-۱- مقدمه

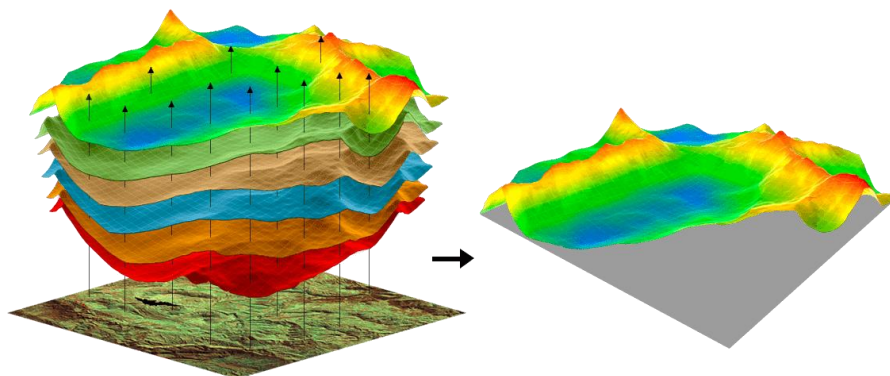
در دنیای امروز، دانش به عنوان بنیان اصلی پیشرفت شناخته می شود و توسعه پایدار مبتنی بر دانش، به عنوان هدف کلیدی بسیاری از کشورها از جمله ایران در نظر گرفته شده است. چنین دانشی با پردازش و تجزیه و تحلیل داده های با کیفیت و قابل اعتماد حاصل می گردد. تحقیق های علمی حاکی از آن است که بیش از ۷۰ درصد از داده هایی که سازمان ها برای تصمیم سازی و تصمیم گیری، مدیریت، برنامه ریزی و اجرا نیاز دارند، به طور مستقیم با مکان و ویژگی های مکانی مرتبط هستند. با توجه به پیچیدگی های موجود در پردازش این داده ها، نیاز به ابزارهایی برای تجزیه و تحلیل داده های مکانی در سطوح مختلف ضروری به نظر می رسد.

از زمان تأسیس OGC در سال ۱۹۹۴، استانداردسازی برای کشف، نمایش، تبادل و پردازش داده های مکانی در قالب سرویس های تحت وب به طور جدی پیگیری شده است. در این زمینه، سرویس های مبتنی بر Web Processing Service (WPS) یکی از ابزارهای اصلی برای پردازش داده های مکانی به صورت آنلاین و از راه دور هستند که می توانند در سیستم های اطلاعات مکانی (GIS) مورد استفاده قرار گیرند. این سرویس ها در راستای تسهیل تحلیل های مکانی و اشتراک گذاری پردازش های پیچیده در بستر وب، اهمیت زیادی دارند.

استاندارد WPS، راه حلی برای انجام پردازش های مکانی به طور آنلاین و در بستر وب فراهم می کند. این سرویس امکان اشتراک گذاری پردازش های پیچیده مکانی را فراهم آورده و به ویژه در پیاده سازی های سامانه های مرتبط با GIS در محیط های دیجیتال، اهمیت ویژه ای پیدا می کند.

۲-۱- تحلیل مکانی

تحلیل مکانی یکی از اجزای اساسی سامانه های اطلاعات مکانی محسوب می شود که داده های خام مکانی را به الگوهای معنادار تبدیل می کند. متخصصان این حوزه با بررسی موقعیت ها، ویژگی ها و روابط مکانی میان عوارض، قادر به شناسایی الگوها و بهبود فرآیند تصمیم گیری بوده و همچنین چالش های پیچیده دنیای واقعی را حل نمایند.



شکل ۱- برهم نهی لایه های مختلف اطلاعات مکانی و استخراج نقشه موضوعی از نتایج آن

تحلیل مکانی شامل پردازش و تفسیر داده های مکانی به منظور استخراج اطلاعات کاربردی است. این تحلیل به پرسش هایی از جمله موارد زیر پاسخ می دهد:

- موقعیت مکانی عوارض خاص (مانند بیمارستان ها یا مناطق مستعد سیل)
- الگوهای موجود در داده های مکانی (مانند پراکندگی بیماری ها یا تراکم ترافیک)
- روابط متقابل میان عوارض (مانند تأثیر جنگل زدایی بر زیستگاه های حیات وحش)

اجزای اصلی تحلیل مکانی عبارتند از پرس و جوهای توصیفی، پرس و جوهای مکانی، تحلیل مجاورت، تحلیل همپوشانی و غیره که در ادامه به توضیح هر یک خواهیم پرداخت.

- **پرس و جوهای توصیفی:** فیلتر نمودن داده ها بر اساس ویژگی های ذاتی مانند شناسایی شهرهای با جمعیت بیش از یک میلیون نفر
- **پرس و جوهای مکانی:** انتخاب عوارض بر مبنای موقعیت جغرافیایی مانند تعیین جاده های متأثر از سیلاب
- **تحلیل مجاورت:** ارزیابی عوارض در محدوده های فاصله ای مشخص مانند مدارس واقع در شعاع یک کیلومتری ایستگاه های پلیس
- **تحلیل همپوشانی:** تلفیق لایه های اطلاعاتی مختلف مانند ترکیب داده های خاک و بارش برای بهینه سازی کشت
- **تحلیل شبکه:** مدلسازی ارتباطات و مسیریابی مانند محاسبه کوتاه ترین مسیر برای توزیع کالا
- **زمین آمار:** شناسایی روندها و الگوهای مکانی مانند تعیین کانون های جرم خیز یا پیش بینی انتشار آلاینده ها

تحلیل مکانی در طیف وسیعی از صنایع، از برنامه ریزی شهری تا کشاورزی پیشرفته، به ابزاری ضروری برای تصمیم گیری های مبتنی بر شواهد تبدیل شده است. از جمله این موارد می توان به موارد زیر اشاره نمود:

- **برنامه ریزی شهری و توسعه شهرهای هوشمند**
 - مکان یابی بهینه خدمات شهری: انتخاب بهترین موقعیت برای احداث مدارس، بیمارستان ها و ایستگاه های حمل و نقل عمومی بر اساس معیارهای دسترسی و پراکندگی جمعیت
 - مدیریت هوشمند ترافیک: تحلیل الگوهای ترافیکی و بهینه سازی زمان بندی چراغ های راهنمایی
 - پهنه بندی کاربری اراضی: تعادل بخشی بین مناطق مسکونی، تجاری و فضای سبز با ملاحظه معیارهای توسعه پایدار
- **پایش و حفاظت محیط زیست**
 - پایش تغییرات پوشش جنگلی: ردیابی نرخ جنگل زدایی با پردازش تصاویر ماهواره ای چندزمانه
 - مدلسازی مخاطرات طبیعی: پیش بینی مناطق سیل گیر با تلفیق داده های توپوگرافی، هیدرولوژیکی و اقلیمی
 - مدیریت زیستگاه های حیات وحش: طراحی کریدورهای امن مهاجرت گونه ها با تحلیل الگوهای حرکتی جانوران
- **تحلیل های بازار و کسب و کار**
 - مکان یابی هوشمند مراکز فروش: تعیین موقعیت بهینه فروشگاه ها بر اساس تحلیل ترکیبی
 - بهینه سازی زنجیره تأمین: طراحی کارآمدترین مسیرهای توزیع برای کاهش هزینه های عملیاتی

- بازاریابی جغرافیایی: هدف گیری تبلیغاتی بر اساس الگوهای رفتاری مصرف کنندگان در مناطق مختلف

• کشاورزی دقیق و هوشمند

- مدیریت نهاده های کشاورزی: تهیه نقشه های توصیه کود و آبیاری بر اساس تغییر پذیری مکانی ویژگی های خاک
- پایش محصولات زراعی: پیش بینی عملکرد با استفاده از شاخص های گیاهی استخراج شده از تصاویر سنجش از دور
- ارزیابی مخاطرات اقلیمی: تحلیل تأثیرات خشکسالی بر تولیدات کشاورزی و برنامه ریزی سازگاری

برخی از رایج ترین توابع تحلیلی مورد استفاده در GIS عبارتند از:

• توابع همپوشانی (Overlay)

- تابع Identity در تابع همپوشانی برداری
- تابع Intersect در تابع همپوشانی برداری
- تابع Union در تابع رویهم قرار گیری برداری

• توابع مجاورت

- تابع Buffer را در توابع مجاورت برداری
- تابع Near را در توابع مجاورت برداری
- تابع Point Distance را در توابع مجاورت برداری

• توابع فاصله

- تابع Euclidean در توابع مجاورت رستری
- تابع Cost در توابع مجاورت رستری
- تابع Corridor در توابع مجاورت رستری

۱-۳- چالش های موجود در تحلیل های مکانی

تحلیل های مکانی به طور معمول شامل پردازش داده هایی هستند که از منابع مختلف، با مقیاس های زمانی و مکانی متفاوت، جمع آوری می شوند. این داده ها معمولاً شامل اطلاعات مکانی پیچیده ای مانند تصاویر ماهواره ای، مدل های ارتفاعی و داده های سنسور حاصل از محیط های واقعی هستند. پردازش این داده ها به ویژه در مقیاس های بزرگ می تواند چالش های متعددی به دنبال داشته باشد که در اینجا به برخی از مهم ترین آن ها اشاره می کنیم.

۱-۳-۱- حجم زیاد داده ها (Big Data)

یکی از بزرگ ترین چالش ها در تحلیل های مکانی، مدیریت و پردازش حجم بالای داده ها است. داده های مکانی معمولاً شامل اطلاعات با ابعاد زمانی و مکانی مختلف بوده که می تواند بسیار بزرگ و پیچیده باشند. به عنوان مثال، تصاویر ماهواره ای با وضوح بالا یا داده های حسگرهای محیطی می توانند حجم بسیار زیادی داشته باشند که این امر نیاز به منابع محاسباتی

و ذخیره‌سازی زیادی دارد. علاوه بر این، پردازش این داده‌ها نیازمند الگوریتم‌ها و روش‌های پیچیده است که به‌طور موثر، این حجم عظیم داده‌ها را پردازش کنند.

۱-۳-۲- تنوع داده‌ها و فرمت‌های مختلف

داده‌های مکانی از منابع مختلفی مانند نقشه‌ها، تصاویر ماهواره‌ای، حسگرهای محیطی و پایگاه‌های داده مکانی به‌دست می‌آیند. این داده‌ها ممکن است در فرمت‌های مختلفی از جمله فایل‌های رستری (مانند GeoTIFF)، برداری (شامل Shapefile و GeoJSON)، یا مدل‌های سه‌بعدی (مانند CityGML) باشند. این تنوع در فرمت‌ها و منابع مختلف، چالش‌هایی را در پردازش و تحلیل یکپارچه داده‌ها به‌وجود می‌آورد؛ به‌ویژه زمانی که جهت انجام برخی تحلیل‌های مکانی، داده‌ها باید از منابع مختلف با یکدیگر ترکیب شوند.

۱-۳-۳- کارایی و مقیاس‌پذیری پردازش‌ها

پردازش داده‌های بزرگ و پیچیده نیاز به محاسبات بسیار زیاد و زمان‌بر دارد. به‌ویژه در تحلیل‌هایی که به‌صورت زمانی یا مکانی مقیاس‌پذیر هستند، مانند پیش‌بینی سیلاب یا تحلیل ریسک‌های زیست‌محیطی. استفاده از پردازش‌های توزیع‌شده، رایانش ابری، یا تکنیک‌های بهینه‌سازی الگوریتم‌ها می‌تواند به کاهش زمان پردازش کمک کند، اما این کار نیازمند زیرساخت‌های پیچیده و منابع محاسباتی پیشرفته است.

۱-۳-۴- یکپارچگی و ارتباط بین داده‌ها

یکی از چالش‌های اساسی در پردازش داده‌های مکانی، یکپارچگی داده‌ها است. داده‌های مکانی معمولاً از منابع مختلفی به‌دست می‌آیند و ممکن است با یکدیگر ناسازگار باشند. به‌عنوان مثال، داده‌های موجود در دو سامانه مختلف ممکن است از نظر سیستم مختصات جغرافیایی، دقت، یا مقیاس متفاوت باشند. این اختلاف‌ها می‌توانند بر دقت تحلیل‌ها تأثیرگذار بوده و جهت همسان‌سازی و تطبیق اطلاعات نیازمند پیش‌پردازش داده‌ها باشد.

۱-۳-۵- دقت و صحت داده‌ها

داده‌های مکانی ممکن است به‌طور دقیق و مستقیم از منابع اولیه به‌دست نیامده باشند و نیاز به تصحیح یا بهینه‌سازی داشته باشند. به‌عنوان مثال، داده‌های حاصل از حسگرهای محیطی ممکن است دچار خطاهای مرتبط با سنسور یا نویز باشند که بر دقت تحلیل‌ها تأثیرگذار است. لذا یکی از چالش‌های عمده در تحلیل‌های مکانی، پردازش این نوع داده‌ها می‌باشد.

۱-۳-۶- قابلیت تعامل در سیستم های توزیع شده

با توجه به اینکه داده های مکانی معمولاً از منابع مختلف و در زمان های مختلف جمع آوری می شوند، نیاز به سیستم های توزیع شده برای پردازش داده ها وجود دارد. این سیستم ها باید توانایی پردازش داده ها را به صورت موازی و با استفاده از چندین سرور یا رایانه فراهم کنند. این چالش ها نیازمند معماری های پیچیده برای مدیریت و هماهنگی پردازش ها و ذخیره سازی داده ها هستند.

۱-۳-۷- پردازش و ذخیره سازی داده های سه بعدی و پیچیده

داده های سه بعدی، مانند مدل های شهری یا داده های مرتبط با ارتفاع، نیاز به پردازش هایی پیچیده دارند. این داده ها معمولاً حجم بالایی داشته و نیازمند الگوریتم هایی برای تجزیه و تحلیل دقیق و بهینه می باشند. علاوه بر این، جهت ذخیره سازی داده های سه بعدی، به زیرساخت هایی خاص برای حفظ و پردازش اطلاعات به صورت مؤثر و با کمترین هزینه نیاز است.

۱-۴- مقدمه ای بر تحلیل های مکانی دسکتاپ و برخط

تحلیل های مکانی همواره بخشی اساسی از سیستم های اطلاعات مکانی (GIS) بوده اند. در گذشته، این تحلیل ها عمدتاً در محیط های دسکتاپ و با استفاده از نرم افزارهایی مانند ArcGIS، QGIS و یا GRASS GIS انجام می شدند. در این روش، کاربران ملزم به نصب نرم افزارهای تخصصی بر روی سیستم های محلی خود بودند و انجام پردازش های پیچیده نیازمند سخت افزارهای قدرتمند و دانش فنی بالا بود. این رویکرد با محدودیت هایی همچون وابستگی به منابع محلی، دشواری در به روزرسانی الگوریتم ها و عدم امکان اشتراک گذاری ساده پردازش ها بین کاربران مختلف همراه بود. با گسترش فناوری های مبتنی بر وب و نیاز روزافزون به دسترسی پذیری و تعامل پذیری در سرویس های مکانی، استانداردهای جدیدی توسط کنسرسیوم Open Geospatial Consortium (OGC) معرفی شدند. این تحول باعث شده که تحلیل های مکانی به راحتی از طریق اینترنت قابل دسترس بوده و امکان استفاده از سرورهای قدرتمند برای پردازش داده ها فراهم گردد. این روند نه تنها کارایی و دقت تحلیل ها را افزایش داده بلکه به تسهیل همکاری های بین سازمانی و اشتراک گذاری داده های مکانی کمک کرده است.

یکی از این استانداردهای کلیدی، Web Processing Service (WPS) است که امکان اجرای پردازش های مکانی را به صورت برخط فراهم می کند. این استاندارد، با تعریف یک چارچوب مناسب برای توصیف، اجرا و مدیریت پردازش های مکانی، تحولی اساسی در نحوه انجام تحلیل های مکانی ایجاد کرده است.

۱-۵- مزایا و معایب تحلیل‌های مکانی دسکتاپ و برخط

استفاده از سیستم‌های دسکتاپ و برخط برای تحلیل‌های مکانی هر کدام مزایا و معایب خاص خود را دارند که در اینجا تا حدی به بررسی هر یک خواهیم پرداخت.

۱-۵-۱- تحلیل‌های مکانی دسکتاپ

مزایا:

- سیستم‌های دسکتاپ معمولاً از منابع سخت‌افزاری خاصی مانند پردازنده‌های قدرتمند، حافظه بیشتر و کارت‌های گرافیکی ویژه استفاده می‌کنند و قادر به پردازش داده‌های پیچیده و حجیم با سرعت بالا می‌باشند. این امر به‌ویژه در تحلیل‌های مکانی که نیازمند پردازش‌های محاسباتی سنگین هستند، مفید است.
- داده‌ها معمولاً به‌طور محلی ذخیره می‌شوند و این امکان را به کاربران می‌دهد که کنترل کاملی بر امنیت داده‌ها و منابع خود داشته باشند. همچنین، سیستم‌های دسکتاپ معمولاً از نظر دسترسی به داده‌ها و منابع از انعطاف‌پذیری بیشتری برخوردارند.
- سیستم‌های دسکتاپ می‌توانند از طیف وسیعی از فرمت‌های داده‌های مکانی و غیرمکانی پشتیبانی کنند و این امر به کاربران این امکان را می‌دهد که داده‌های مختلف را در قالب‌های متنوعی پردازش کنند.
- کاربران سیستم‌های دسکتاپ کنترل بیشتری بر روی فرآیندهای پردازش داده‌ها دارند. به‌عنوان مثال، کاربران می‌توانند پس از تنظیم مناسب پارامترهای مورد نیاز، پردازش‌ها را به‌طور دقیق‌تری انجام دهند. این امر به‌ویژه در پروژه‌های پیچیده که نیازمند تنظیمات خاص هستند، مفید است.

معایب:

- سیستم‌های دسکتاپ برای پردازش داده‌های حجیم و پیچیده نیاز به سخت‌افزار قدرتمند دارند. این موضوع باعث افزایش هزینه‌ها برای خرید و نگهداری سیستم‌های دسکتاپ می‌شود. علاوه بر این، در صورتی که حجم داده‌ها افزایش یابد، سیستم‌های دسکتاپ ممکن است به‌طور کامل نتوانند این داده‌ها را پردازش کنند.
- داده‌ها در سیستم‌های دسکتاپ معمولاً به‌طور محلی ذخیره می‌شوند و برای دسترسی به آن‌ها از هر نقطه‌ای نیاز به اتصال فیزیکی به سیستم است. این امر دسترسی به داده‌ها و تحلیل‌ها را محدود می‌کند و در صورت نیاز به همکاری با دیگر افراد، مشکلاتی ایجاد می‌کند.
- سیستم‌های دسکتاپ به‌طور معمول قابلیت اشتراک‌گذاری داده‌ها و تحلیل‌ها را به‌صورت آنلاین یا از طریق اینترنت ندارند. این محدودیت می‌تواند در محیط‌هایی که به صورت گروهی فعالیت می‌کنند، مشکل‌ساز باشد.

۱-۵-۲- تحلیل های مکانی برخط

مزایا:

- سیستم های برخط امکان دسترسی به داده ها و پردازش ها را از هر نقطه ای فراهم می آورند. کاربران می توانند بدون نیاز به نصب نرم افزارهای تخصصی، از خدمات پردازش مکانی برخط استفاده نموده که این امر موجب افزایش انعطاف پذیری می شود.
- سیستم های برخط به ویژه با استفاده از رایانش ابری و پردازش توزیع شده، قادرند داده های بزرگ و پیچیده را به صورت مقیاس پذیر پردازش کنند. این امر کاربران را قادر می سازد بدون نیاز به منابع سخت افزاری خاص، پردازش های پیچیده را با سرعت و کارایی بالا انجام دهند.
- یکی از مزایای مهم سیستم های برخط این است که امکان اشتراک گذاری داده ها و نتایج تحلیل ها را به صورت آنلاین فراهم می کنند. این امر به تیم ها و سازمان ها اجازه می دهد تا به طور همزمان و به صورت موثر با یکدیگر همکاری نموده و به داده ها دسترسی داشته باشند.
- در سیستم های برخط، به روزرسانی نرم افزارها و داده ها به صورت متمرکز و از طریق سرورهای مرکزی انجام می شود. این امر موجب می شود فرآیند نگهداری سیستم ساده تر شده و به روزرسانی ها به طور خودکار در دسترس کاربران قرار گیرد.

معایب:

- یکی از معایب اصلی سیستم های برخط این است که برای دسترسی به خدمات و انجام پردازش ها، نیاز به اتصال پایدار به اینترنت وجود دارد. این امر ممکن است در مناطقی با اینترنت ضعیف یا در شرایط اضطراری که دسترسی به اینترنت محدود است، به یک محدودیت تبدیل شود.
- داده ها و پردازش ها در سیستم های برخط معمولاً در سرورهای از راه دور ذخیره می شوند. این موضوع می تواند مشکلات امنیتی و حریم خصوصی ایجاد کند، به ویژه در صورتی که داده های حساس و شخصی در میان باشد. اطمینان از حفاظت از داده ها در برابر تهدیدات امنیتی در سیستم های برخط می تواند چالش برانگیز باشد.
- در سیستم های برخط، کنترل داده ها به طور عمده در دست ارائه دهندگان سرویس قرار دارد. این موضوع ممکن است برای سازمان هایی که نیاز به کنترل دقیق تر و امنیت بیشتر دارند، مشکل آفرین باشد.

۱-۶- بررسی چالش های فنی و زیرساختی در پیاده سازی WPS

پیاده سازی استاندارد WPS می تواند مزایای زیادی از جمله دسترسی آسان، مقیاس پذیر، و امکان اشتراک گذاری پردازش ها را به صورت آنلاین داشته باشد. با این حال، چالش های فنی و زیرساختی مختلفی در این فرآیند وجود دارد که

نیازمند توجه و مدیریت دقیق هستند. در این بخش، برخی از چالش‌های موجود در پیاده‌سازی WPS بر بستر وب را بررسی می‌کنیم.

۱-۶-۱- چالش‌های فنی در پیاده‌سازی WPS

- یکی از چالش‌های اصلی در پیاده‌سازی WPS، پیچیدگی طراحی و توسعه فرآیندها و سرویس‌های مبتنی بر آن است. برای پیاده‌سازی یک سرویس WPS کارآمد، نیاز است که فرآیندهای پیچیده مکانی به صورت استاندارد تعریف شده و ورودی‌ها و خروجی‌ها به طور دقیق مشخص شوند. علاوه بر این، باید اطمینان حاصل شود که این سرویس‌ها قابلیت تعامل با دیگر سرویس‌ها و سیستم‌های موجود را دارا بوده که این امر می‌تواند نیازمند زمان و منابع زیادی باشد.
- با توجه به اینکه سرویس WPS به طور طبیعی برای پردازش داده‌های مکانی تحت وب طراحی شده است، این سرویس‌ها در مقیاس‌های بزرگ، ممکن است با مشکلات مقیاس‌پذیری مواجه شوند به ویژه زمانی که داده‌ها حجیم یا پیچیده باشند. برای پردازش‌های پیچیده و داده‌های بزرگ، نیاز به زیرساخت‌های محاسباتی قدرتمند و توزیع شده است که می‌تواند هزینه‌ها و پیچیدگی‌های بیشتری را به همراه داشته باشد.
- از آنجا که داده‌های ورودی سرویس WPS ممکن است از منابع مختلف با فرمت‌های متفاوت (شامل داده‌های برداری، داده‌های رستری و داده‌های سه بعدی) باشند، یکپارچگی این داده‌ها در فرآیندهای مختلف می‌تواند چالشی بزرگ باشد. ناسازگاری در سیستم‌های مختصات، فرمت‌ها و مدل‌های داده می‌تواند به مشکلاتی در انجام پردازش‌ها و تحلیل‌های مکانی منجر شده که نیازمند پیاده‌سازی الگوریتم‌های پیچیده برای همسان‌سازی و هماهنگی داده‌ها است.
- یکی دیگر از چالش‌های فنی در پیاده‌سازی WPS، پشتیبانی از عملیات همزمان (Synchronous) و ناهمزمان (Asynchronous) است. در برخی موارد، پردازش‌های مکانی ممکن است زمان‌بر باشند، به ویژه زمانی که داده‌ها حجیم یا تحلیل‌ها پیچیده هستند. این امر نیاز به مدیریت صحیح درخواست‌ها و نتایج پردازش‌ها دارد تا مطمئن شویم که درخواست‌های کاربران به طور مؤثر مدیریت شده و سیستم به درستی به درخواست‌ها پاسخ می‌دهد.

۱-۶-۲- چالش‌های زیرساختی در پیاده‌سازی WPS

- پیاده‌سازی WPS در بستر وب نیازمند منابع محاسباتی کافی است تا بتواند پردازش‌های پیچیده مکانی را به صورت مؤثر انجام دهد. به ویژه هنگامی که پردازش‌ها باید در مقیاس بزرگ انجام شوند، نیاز به سرورها و منابع ابری با

توان محاسباتی بالا است. علاوه بر این، نیاز به ذخیره سازی داده های بزرگ (مانند تصاویر ماهواره ای و مدل های سه بعدی) و مدیریت کارآمد آنها نیز چالشی زیرساختی است.

- استاندارد WPS به عنوان یکی از استانداردهای OGC نیاز به پیاده سازی دقیق پروتکل ها و فرمت های باز دارد و ارتباطات میان سیستم های مختلف را تسهیل می کند. جهت پیاده سازی موفق این استاندارد و نیز تبادل داده ها و عملیات در بستر وب، لازم است زیرساخت های فناوری اطلاعات از استانداردهایی مانند SOAP، WSDL و XML پشتیبانی کنند. هرگونه ناسازگاری با این استانداردها می تواند منجر به بروز مشکلاتی در تعاملات بین سرویس ها و پردازش ها شود.
- یکی از چالش های اساسی در پیاده سازی WPS بر بستر وب، مسائل امنیتی و حفظ حریم خصوصی است. داده های با ماهیت مکانی ممکن است حساسیت زیادی داشته باشند، به ویژه زمانی که شامل اطلاعات خصوصی مانند مالکیت املاک یا داده های محیطی خاص باشد. اطمینان از امنیت داده ها، به ویژه در زمان انتقال آنها از طریق اینترنت، و حفاظت از اطلاعات حساس در برابر تهدیدات امنیتی مانند حملات سایبری از اهمیت زیادی برخوردار است.
- یکی از ویژگی های مهم سیستم های برخط، مقیاس پذیری است تا در شرایطی که حجم داده ها افزایش می یابد، عملکرد خود را حفظ کنند. این امر به ویژه در پردازش های پیچیده مکانی که نیازمند منابع بالای پردازشی هستند، اهمیت پیدا می کند. برای این کار، ممکن است نیاز به زیرساخت های توزیع شده ای مانند رایانش ابری یا سرورهای مجازی باشد که بتوانند منابع محاسباتی را به طور مقیاس پذیر تخصیص داده و از هرگونه خرابی یا کاهش عملکرد جلوگیری کنند.
- یکی دیگر از چالش های زیرساختی در پیاده سازی WPS، مدیریت تعداد درخواست های همزمان از کاربران مختلف است. در مواردی که تعداد درخواست های همزمان به طور قابل توجهی افزایش یابد، ممکن است بار زیادی به شبکه وارد شده و منجر به کاهش کارایی یا حتی از کار افتادن سیستم شود. برای مدیریت این بار، نیاز به استفاده از فناوری هایی از قبیل کش (caching) یا توزیع بار (load balancing) است تا پردازش ها به طور مؤثر تقسیم شده و سیستم بتواند به صورت پایدار به درخواست ها پاسخ دهد.

۷-۱- مجموعه استانداردهای OGC در داده های مکانی

OGC (Open Geospatial Consortium) یک سازمان بین المللی است که استانداردهای مختلفی را برای تبادل، پردازش و نمایش داده های مکانی تعریف کرده است. این استانداردها در زمینه هایی مانند سیستم های اطلاعات مکانی (GIS)، پردازش های مکانی تحت وب، و وب سرویس ها مورد استفاده قرار می گیرند. در ادامه، به توضیح برخی از مهم ترین استانداردهای OGC و ارتباط آنها با WPS می پردازیم.

Web Map Service (WMS) - ۱-۷-۱

استاندارد WMS یکی از شناخته‌شده‌ترین استانداردهای OGC است که از عکس‌ها و تصاویر رستر جهت ارائه داده‌های مکانی استفاده می‌کند. این سرویس کاربران را قادر می‌سازد تا لایه‌های مختلف مکانی، نقشه‌های شهری و یا تصاویر ماهواره‌ای را از طریق وب درخواست نموده و آن‌ها را به صورت تصاویر در فرمت‌های مختلف مانند PNG یا JPEG نمایش دهند.

نتایج حاصل از پردازش‌های صورت گرفته توسط استاندارد WPS، می‌تواند به عنوان ورودی سرویس WMS مورد استفاده قرار گیرد. به طور مثال، پس از انجام تحلیل‌های مکانی (مانند شناسایی زمین‌های کشاورزی واقع در محدوده‌های سیل گیر) از طریق سرویس WPS، نتایج می‌توانند به صورت یک لایه جدید توسط سرویس WMS نمایش داده شوند.

Web Feature Service (WFS) - ۲-۷-۱

یکی دیگر از استانداردهای پر کاربرد OGC استاندارد WFS نام دارد که به کاربران این امکان را می‌دهد که داده‌های مکانی را به همراه ویژگی‌های مکانی عوارض (مانند اطلاعات هندسی، اطلاعات توصیفی و ...) از طریق وب درخواست کنند. برخلاف استاندارد WMS که سرویس لایه‌های مکانی را در قالب تصاویر رستر ارائه می‌دهد، استاندارد WFS این امکان را به کاربران می‌دهد تا از خروجی‌های حاصل از این سرویس، جهت بازیابی، ویرایش و انجام تحلیل‌های مکانی استفاده شود. به طور مثال، پس از دریافت داده‌های حاصل از شبکه آبریز یک منطقه با استفاده از استاندارد WFS، می‌توان به کمک استاندارد WPS تحلیل‌هایی مانند شبیه‌سازی سیلاب را پیاده‌سازی نمود.

Web Coverage Service (WCS) - ۳-۷-۱

استاندارد WCS برای دسترسی و بازیابی داده‌های رستری، مانند تصاویر ماهواره‌ای، مدل‌های ارتفاعی یا سایر مجموعه داده‌های مکانی مبتنی بر رستر طراحی شده است. برخلاف WMS که وب سرویس‌های ایجاد شده را به صورت یه تصویر شماتیک برمی‌گرداند، استاندارد WCS بر ارائه داده‌های رستری جهت مشاهده، بازیابی و انجام تحلیل‌های مکانی تمرکز دارد.

Wep Processing Service (WPS) - ۴-۷-۱

از دیگر استانداردهای پر کاربرد OGC، استاندارد WPS است که جهت اجرای پردازش‌های مکانی (مانند پیاده‌سازی تحلیل مکانی، مدل‌سازی یا انجام تبدیلات) به صورت تحت وب مورد استفاده قرار می‌گیرد. برخلاف سرویس‌هایی که داده‌های خام (مانند WCS یا WFS) یا نمایش شماتیک (مانند WMS) را برمی‌گردانند، WPS بر انجام پردازش‌ها بر اساس تقاضا و اجرای الگوریتم در یک محیط توزیع‌یافته متمرکز است.

۱-۷-۵- Catalogue Service for the Web (CSW)

استاندارد CSW یکی از استانداردهای OGC است که به صورت تحت وب برای جستجوی فراداده های مکانی، خدمات و منابع مرتبط استفاده می شود. این سرویس به عنوان یک کاتالوگ قابل جستجو (مانند یک "کتابخانه مکانی") عمل می کند و به کاربران و برنامه ها کمک می کند تا مجموعه داده های مرتبط (مانند تصاویر ماهواره ای، نقشه ها یا خدمات WMS/WFS) را بر اساس معیارهای فراداده به صورت تحت وب جستجو نمایند.

۱-۷-۶- Sensor Observation Service (SOS)

استاندارد SOS برای جمع آوری، مدیریت و ارائه داده های سنسوری محیطی مانند دما، رطوبت و فشار هوا طراحی شده است. این سرویس که زیرمجموعه استانداردهای OGC است به کاربران اجازه می دهد تا انواع داده های محیطی مختلف را درخواست کنند. استاندارد WPS می تواند داده های SOS را برای انجام پردازش های پیچیده مانند پیش بینی وضعیت هوا یا شبیه سازی تغییرات اقلیمی استفاده کند و تاثیرات این شرایط را بر اکوسیستم ها تحلیل کند.

۱-۷-۷- Web Map Tile Service (WMTS)

استاندارد WMTS یکی از استانداردهای OGC برای ارائه نقشه های بخش بندی شده تحت وب است. این سرویس با تقسیم نقشه به قطعات کوچک و قابل بارگذاری سریع، نمایش روان داده های مکانی را در وب و موبایل ممکن می سازد. WMTS به ویژه برای نمایش نقشه های پایه با عملکرد بالا مانند سامانه های ناوبری، نقشه های تعاملی و داشبوردهای مکانی مناسب است و امکان مشاهده نقشه در مقیاس های مختلف را بدون نیاز به بارگذاری کامل هر بار فراهم می کند.

۸-۱- پیاده سازی های مختلف WPS و ارزیابی آنها

برای راه اندازی یک فرآیند پردازشی تحت وب مطابق با استانداردهای OGC، استفاده از استاندارد WPS ضروری است. این استاندارد امکان اجرای پردازش های مکانی-زمانی را به صورت تحت وب فراهم می کند. پیاده سازی های مختلفی از WPS وجود دارند که هر یک دارای ویژگی ها، مزایا و محدودیت های خاص خود هستند. انتخاب مناسب ترین گزینه از میان آن ها به عوامل کلیدی زیر بستگی دارد:

- نسخه ی استاندارد WPS مورد پشتیبانی
- ویژگی های فنی (مانند پردازش موازی، پشتیبانی از فرمت های مختلف داده)
- زیرساخت اجرایی (سیستم عامل، زبان برنامه نویسی، کتابخانه های وابسته)
- مقیاس پذیری و انعطاف پذیری

در جدول ۱ به بررسی مختصری از این پیاده سازی ها می پردازیم:

جدول ۱- پیاده‌سازی‌های WPS

معایب	مزایا	ویژگی‌ها	زبان برنامه‌نویسی	پیاده‌سازی WPS
- نیاز به منابع محاسباتی بالا - پیچیدگی پیکربندی در پردازش‌های بسیار پیچیده	- مقیاس‌پذیری بالا - پشتیبانی کامل از استانداردهای OGC - سادگی در پیاده‌سازی برای توسعه‌دهندگان	- استفاده از Java Framework - پشتیبانی کامل از پروتکل‌های OGC (WPS, WMS, WFS) - مناسب پردازش‌های پیچیده و حجیم	Java	
- پیچیدگی در پیکربندی و نگهداری - نیاز به زیرساخت‌های قدرتمند برای پردازش‌های توزیع‌شده - نیاز به دانش فنی زیاد	- مدولار و انعطاف‌پذیر - پشتیبانی از زبان‌های مختلف - مناسب برای سیستم‌های توزیع‌شده و مقیاس‌پذیر	- معماری مدولار - پشتیبانی از چندین زبان برنامه‌نویسی - پشتیبانی از پردازش‌های موازی و توزیع‌شده - پیاده‌سازی استاندارد WPS 2.0.0	Python, C, JavaScript	
- محدودیت در سفارشی‌سازی خارج از ArcGIS - هزینه‌های لایسنس	- یکپارچگی عالی با اکوسیستم ArcGIS - پشتیبانی قوی از تحلیل‌های مکانی	- رابط گرافیکی کاربرپسند - بهینه‌شده برای پردازش‌های ArcGIS - پشتیبانی از فرمت‌های Esri, GeoJSON - پشتیبانی از ArcGIS ModelBuilder, REST API و سرویس‌های پردازش مکانی	Python, JavaScript	
- پیچیدگی در پیاده‌سازی - نیاز به منابع محاسباتی زیاد - نامناسب برای پروژه‌های کوچک	- پشتیبانی کامل از استانداردهای OGC - مقیاس‌پذیری بالا - مناسب برای سازمان‌ها و پروژه‌های بزرگ	- پشتیبانی کامل از پروتکل‌های OGC (WPS, WMS, WFS) - یکپارچگی داده‌های مکانی از منابع مختلف - مناسب برای پروژه‌های مقیاس بزرگ	Java	
- کارایی محدود در پردازش‌های بسیار پیچیده - نیاز به پیکربندی اضافی برای مقیاس‌های بزرگ	- پشتیبانی از فرمت‌های مختلف داده - رابط کاربری ساده - پشتیبانی از پردازش‌های پیچیده	- پشتیبانی کامل از پروتکل‌های OGC (WPS, WMS, WFS) - مناسب برای پروژه‌های کوچک تا متوسط - رابط گرافیکی کاربرپسند	Java	
- کارایی کمتر نسبت به دیگر پیاده‌سازی‌ها در پردازش‌های بسیار بزرگ - نیاز به دانش در برنامه‌نویسی Python	- سادگی در توسعه - مناسب برای پروژه‌های مقیاس کوچک و متوسط - پشتیبانی از پردازش‌های پیچیده در Python	- پیاده‌سازی مبتنی بر Python - قابلیت گسترش و سفارشی‌سازی بالا - پشتیبانی از پردازش‌های پیچیده - ساده و سریع در توسعه فرایندهای WPS	Python	

۹-۱- نیازمندی ها و مراحل پیاده سازی استاندارد WPS در محیط ژئوپورتال

پیاده سازی پردازش های مکانی آنلاین در محیط ژئوپورتال با استفاده از استاندارد WPS، نیازمند توجه به مجموعه ای از نیازمندی ها و مراحل خاص است. در ادامه به معرفی برخی از این موارد می پردازیم.

- زیرساخت سخت افزاری و محاسباتی:
 - ✓ تأمین سرورهای با قدرت پردازشی مناسب جهت انجام تحلیل های مکانی پیچیده به صورت آنلاین
 - ✓ پشتیبانی از پردازش موازی برای عملیات سنگین محاسباتی
 - ✓ امکان مدیریت داده های حجیم مکانی با کارایی مطلوب
 - استانداردهای داده ای و فرمت های پشتیبانی:
 - ✓ استفاده از استاندارد WMS برای نمایش لایه ها
 - ✓ استفاده از استاندارد WFS برای انتقال داده های برداری
 - ✓ استفاده از استاندارد WCS برای داده های رستری
 - ✓ استفاده از استاندارد GML به عنوان فرمت استاندارد تبادل داده های مکانی
- پیکربندی سرور پیشنهادی به منظور راه اندازی یک سامانه تحلیل مکانی تحت وب به صورت زیر می باشد:
- RAM ۱۶ گیگابایت
 - CPU ۸ هسته ای
 - SSD ۱۰۰ گیگابایت

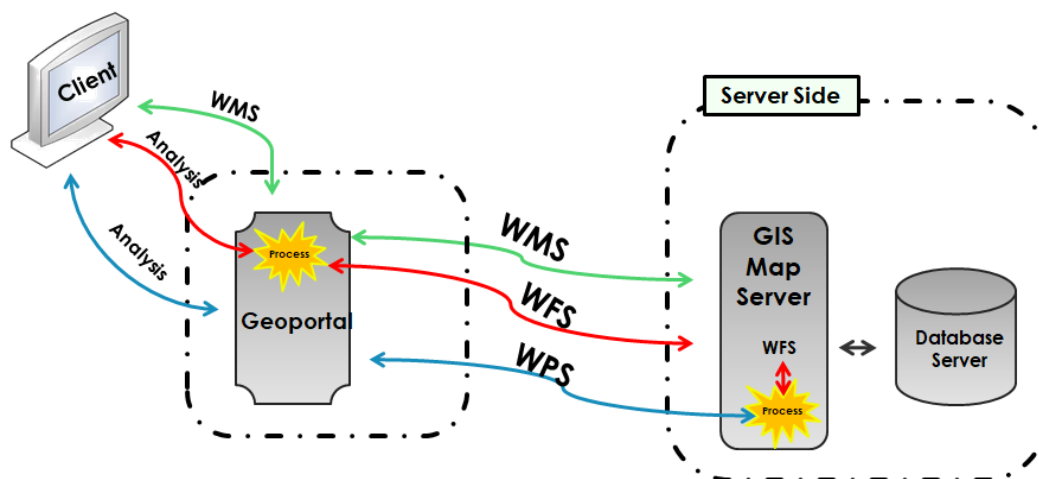
۱۰-۱- WPS در ژئوپورتال سازمان نقشه برداری کشور

وضعیت موجود شامل ساخت و نمایش سرویس های WMS از لایه های موجود در سازمان های مدیریت و برنامه ریزی در سطح استانی بر روی ژئوپورتال ملی سازمان نقشه برداری می باشد. بدین صورت که داده ها از طریق سرویس های WFS و یا WCS توسط ژئوپورتال سازمان نقشه برداری کشور مورد دسترسی و در این سطح توسط سرویس WPS مورد پردازش قرار می گیرد.

چنین ساختار معایبی را به همراه دارد، که از آن جمله می توان به موارد زیر اشاره کرد:

- سرویس WFS به دلیل دارا بودن مختصات مکانی و اطلاعات توصیفی عوارض، عملکرد نسبتاً کندی دارد. از جمله مهم ترین عوامل موثر در این زمینه، می توان به موارد ذیل اشاره کرد:
 - حجم داده: هر چه داده های مکانی و توصیفی بیشتر باشد، کوئری گرفتن و انتقال داده نیز کندتر خواهد بود.
 - بار سرور^۱: تقاضای زیاد به خصوص برای مجموعه داده ای بزرگ می تواند منجر به زمان پاسخ کندتری گردد.

- تاخیر شبکه^۱: سرعت اتصال شبکه بین سرور متولیان داده و سرور ملی همچنین می‌تواند بر سرعت انتقال داده تاثیرگذار بوده باشد.
- در دسترس قرار دادن سرویس WFS سبب ایجاد یکسری دغدغه‌ها در رابطه با امنیت داده و حقوق ذینفعان می‌شود که به موجب آن، منجر به عدم رغبت متولیان داده در به اشتراک‌گذاری داده‌های مکانی می‌گردد.

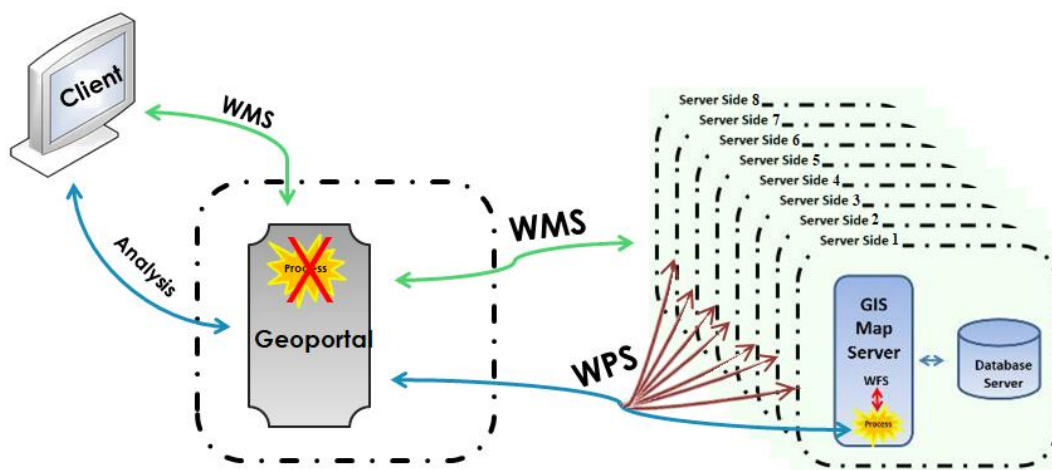


شکل ۲- ساختار موجود در ژئوپرتال سازمان نقشه برداری کشور

از گزینه‌های جایگزین به منظور رفع و کاستن از مشکلات مطرح شده در وضع موجود، می‌توان به استفاده از سرویس WPS و پردازش داده در سمت سرور متولیان داده و سپس ارسال نتایج به ژئوپرتال ملی اشاره نمود (شکل ۳). به طور مثال در فرایند انتخاب مکان بهینه برای ایجاد یک بیمارستان در شهر تهران، متولیان داده دیگر داده‌های خام را از طریق سرویس WFS در اختیار کاربران قرار نخواهند داد و در عوض سرویس حاصل از نتایج پردازش را به صورت فایل رستر ارائه می‌دهند. در نهایت سرور ملی با اعمال عملیات پردازشی مورد نظر بر مبنای یکسری معیارها، خروجی نهایی را ایجاد می‌کند.

چنین اقدامی، نیازمند انتقال داده از سمت سرور متولیان داده به سرور سازمان نقشه برداری نبوده و فقط نتایج به سمت سرور ملی ارسال می‌شود، که سبب حذف مشکلات مطرح شده در روش پیشین می‌گردد.

^۱ Network Latency



شکل ۳ - ساختار پیشنهادی

در بین گزینه‌های موجود در پیاده‌سازی WPS، استفاده از زبان برنامه‌نویسی جاوا به دلیل قابلیت‌های محدود در اسکرپت‌نویسی و مصرف بالای حافظه، اجتناب و از طرف دیگر، زبان برنامه‌نویسی پایتون به دلیل سادگی، قابلیت گسترش و امکان اسکرپت‌نویسی در SDI سازمان نقشه‌برداری کشور انتخاب شده است.

از جمله مباحث مهم دیگر، وجود پیاده‌سازی پایتون برای کتابخانه‌هایی نظیر OGR/GDAL (انجام تحلیل‌های مکانی) و PROJ (تبدیل سیستم‌های مختصات) می‌باشد. به همین دلایل، پیاده‌سازی PyWPS از این بین انتخاب شده است که در ادامه به طور مبسوط به توضیح و تشریح و مثال‌هایی در این رابطه پرداخته خواهد شد. در فصول آتی این مستند، مباحث تئوریک استاندارد WPS و جزئیات فنی آن بر اساس دو پیاده‌سازی GeoServer و PyWPS به تفصیل شرح داده خواهد شد با این توضیح که GeoServer با فراهم آوردن GUI امکان راحت‌تر استفاده از تحلیل‌ها را فراهم می‌آورد. از طرف دیگر PyWPS به دلیل کدنویسی امکان شخصی‌سازی بالاتری را برای توسعه‌دهندگانی که می‌خواهند فرآیندهای پیچیده و خاص را پیاده‌سازی کنند، در اختیار قرار می‌دهد.

فصل دوم

ویژگی های سرویس تحلیل مکانی

۲-۱- استاندارد WPS

استاندارد WPS یکی از استانداردهای سازمان OGC است که چارچوبی کلیدی برای ارائه، کشف و استفاده از سرویس‌های پردازش مکانی در محیط‌های تحت وب است. این استاندارد با تعریف روشی یکپارچه، دسترسی ساده و ساختاریافته به طیف گسترده‌ای از پردازش‌های مکانی - از محاسبات ساده تا مدل‌سازی‌های پیچیده - را ممکن می‌سازد.



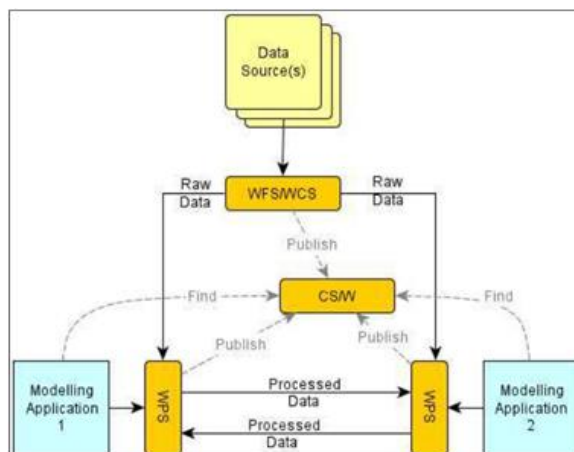
شکل ۴- استاندارد WPS

یکی از مزایای اصلی WPS، استقلال آن از فناوری‌های سمت سرور است. این استاندارد با پنهان کردن جزئیات پیاده‌سازی، امکان فراخوانی سرویس‌های پردازشی را روی هر پلتفرمی فراهم می‌کند. به‌علاوه، انعطاف‌پذیری بالایی در پشتیبانی از انواع داده‌های مکانی (بردار و رستری) و پردازش‌های مختلف - از تعیین حریم رودخانه تا شبیه‌سازی تغییرات آب‌وهوایی - دارد.

داده‌های مورد نیاز پردازش‌ها می‌توانند هم از طریق شبکه (مانند سرویس‌های WMS و WFS) و هم به‌صورت مستقیم از سرورها تأمین شوند. همچنین، استاندارد WPS با ارائه اطلاعات قابل خواندن توسط ماشین (مانند XML/GML) و فراداده‌های قابل خواندن توسط انسان، جستجو و استفاده از سرویس‌ها را تسهیل می‌کند. از نظر فنی، این استاندارد از زبان‌های XML و GML برای تبادل داده‌ها و دستورات پردازشی استفاده می‌کند و هر فرآیند را به‌صورت یک الگوریتم یا مدل محاسباتی تعریف می‌نماید. سازگاری WPS با دیگر استانداردهای OGC مانند WMS و WFS، امکان توسعه سرویس‌های مکانی یکپارچه و پیشرفته را فراهم می‌آورد.

به‌طور کلی، WPS با ارائه یک چارچوب استاندارد و مستقل از فناوری، بستری ایده‌آل برای پیاده‌سازی سرویس‌های پردازش مکانی در سیستم‌های اطلاعات مکانی و محیط‌های ابری ایجاد کرده است. این ویژگی‌ها آن را به ابزاری ارزشمند برای تحلیل‌های مکانی در مقیاس‌های مختلف تبدیل کرده است.

شکل ۵ یک دیاگرام رابط WPS را به‌طور خلاصه نمایش می‌دهد. در این دیاگرام، داده‌ها از طریق سرویس‌های استاندارد WFS و WCS و بر اساس فراداده‌هایی که توسط کاتالوگ سرویس‌های تحت وب ایجاد شده‌اند در اختیار سرویس استاندارد WPS قرار گرفته و این سرویس، خروجی را به صورت استاندارد در اختیار نرم‌افزارهای کاربردی به منظور انجام مدل‌سازی‌های مختلف قرار می‌دهند.



شکل ۵- دیاگرام سرویس‌های WFS، CSW و WPS

۲-۲-۲- پیکربندی WPS

یک سرویس WPS به گونه‌ای می‌تواند پیکربندی گردد که هر نوع قابلیت تحلیلی GIS را به کاربران در سراسر شبکه ارائه دهد؛ از جمله دسترسی به محاسبات از پیش برنامه‌ریزی شده و یا مدل‌های محاسباتی که بر روی داده‌های مکان مرجع مورد استفاده قرار می‌گیرند. فرآیندهای WPS مشمول همان محدودیت feature limit سرویس WFS می‌باشد. این محدودیت برای پردازش ورودی‌ها اعمال می‌شود، بنابراین حتی پردازش‌هایی که سبب خلاصه سازی داده می‌شوند و نتایج کمی را برمیگردانند در صورت اعمال بر داده‌های زیاد، تحت تاثیر قرار خواهند گرفت. این محدودیت در تنظیمات WFS انجام می‌شود. در صورت اعمال آن خطایی داده نمی‌شود اما نتایج ممکن است ناکامل باشد. پیکربندی سرویس WPS شامل سه قسمت کلی Service Metadata، Execution Settings و Resource Settings می‌باشد که در بخش Services با انتخاب WPS در دسترس خواهد بود. در ادامه این قسمت‌ها با جزئیات بیشتر بررسی می‌گردند.

Service Metadata -1-2-2

بخش Service Metadata در بین تمامی سرویس ها (WPS, WMS, WFS, WCS) مشترک می باشد. این بخش از یکسری تعاریف فراداده ای رایج برای توصیف سرویس مد نظر و نشان دادن محدودیت ها استفاده میکند.

Service Metadata

☒ Enable WPS

☐ Strict CITE compliance

Maintainer
Ross Blake

Online resource
<http://www.geoserver.org>

Title
GeoServer WPS

Abstract
Exposing spatial analysis functions to the web

Fees
NONE

Access Constraints
NONE

Current Keywords
wps
process
asynchronous

New Keyword
asynchronous

Vocabulary

Add Keyword

شکل ۶- Service Metadata

المان‌های شکل ۶ در جدول ۲ شرح مختصری داده می‌شوند. اگرچه فیلدهای فوق در سرویس‌ها یکی می‌باشند، مقادیر آنها به اشتراک گذاشته نمی‌شوند.

جدول ۲- المان‌های Service Metadata

فیلد	شرح مختصر
Enabled WPS	فعال بودن یا نبودن سرویس WPS را مشخص می‌کند. در صورت غیرفعال بودن درخواست‌های سرویس مذکور پردازش نخواهند شد.
Strict CITE compliance	در صورت انتخاب، انطباق کامل با CITE ^۱ را تضمین می‌کند. در زمان اجرای تست‌های مطابقت اعمال آن پیشنهاد می‌گردد.
Maintainer	نام طرف مسئول (شخص، نهاد و ...) که از سرویس مد نظر نگهداری می‌کند.
Online Resource	HTTP URL سرویس را تعریف می‌کند. به طور معمول این فیلد با URL صفحه‌ی نخست سرویس تکمیل می‌گردد. تکمیل این فیلد الزامی می‌باشد.
Title	یک عنوان خواندنی به منظور شناسایی مختصر سرویس به کلاینت‌ها. تکمیل این فیلد الزامی می‌باشد.

^۱ Compliance Interoperability Test Engine (CITE)

Abstract	یک توضیحی با اطلاعات بیشتر در مورد سرویس فراهم میکند.
Fees	نشانگر هزینه ی اعمال شده توسط فراهم کننده ی سرویس به منظور استفاده از سرویس می باشد. کلمه ی کلیدی NONE کلمه ی رزرو شده به معنی بدون هزینه بوده که در بیشتر حالات استفاده می گردد.
Access Constraints	قیودی که توسط فراهم کننده ی سرویس اعمال می شود را توصیف میکند. کلمه ی کلیدی NONE کلمه ی رزرو شده می باشد که نشانگر عدم اعمال هیچگونه قیود دسترسی بوده و در بیشتر حالات استفاده می گردد.
Keywords	لیستی از لغاتی که مرتبط با سرویس بوده و به منظور بهبود جستجو و فهرست نویسی استفاده میشود.

۲-۲-۲ Execution and resource settings

Execution Settings

Connection Timeout (seconds, -1 for infinite timeout)

Maximum synchronous executions run parallel

Maximum execution time for synchronous requests (seconds, -1 for no limit)

Maximum queue and execution time for synchronous requests (seconds, -1 for no limit)

Maximum asynchronous executions run parallel

Maximum execution time for asynchronous requests (seconds, -1 for no limit)

Maximum queue and execution time for asynchronous requests (seconds, -1 for no limit)

Resource Settings

Resource Expiration Timeout (seconds) (0 for no expiration)

Resource storage directory

External output directory (empty to disable)

شکل ۷- Execution & Resource Settings

جدول ۳- Execution & Resource Settings

شرح مختصر	فیلد
Execution settings	
ثانیه هایی که سرویس WPS منتظر می ماند قبل از اینکه بخواهد ارتباط مبتنی بر HTTP از راه دور را متوقف کند.	Connection timeout
حداکثر تعداد پردازش های همزمان که به طور موازی در یک زمان مشخص اجرا می شوند.	Maximum synchronous executions run parallel
حداکثر زمانی که یک پردازش همزمان مجاز به اجرا می باشد. پردازش هایی که در حالت همزمان اجرا می شوند باید در زمان مجاز تمام گردند. در غیر این صورت به طور اتوماتیک کنار گذاشته می شوند. این درخواست	Maximum execution time for synchronous requests

	ها بدلیل اینکه منجر به انتظار کلاینت برای پاسخ از طریق اتصال HTTP می باشند، باید زمان های نسبتا کوتاهی انتخاب گردد (به طور مثال ۶۰ ثانیه).
Maximum queue and execution time for synchronous requests	حداکثر زمانی که یک پردازش مجاز به ماندن در نوبت و اجرایش می باشد. پردازش هایی که در حالت همزمان اجرا می شوند باید در زمان مجاز تمام گردند. در غیر این صورت به طور اتوماتیک کنار گذاشته می شوند. این درخواست ها بدلیل اینکه منجر به انتظار کلاینت برای پاسخ از طریق اتصال HTTP می باشند، باید زمان های نسبتا کوتاهی انتخاب گردد (به طور مثال ۶۰ ثانیه).
Maximum asynchronous executions run parallel	حداکثر تعداد پردازش های غیرهمزمان که به طور موازی در زمان تعیین شده اجرا می شوند. بقیه در نوبت قرار می گیرند.
Maximum execution time for asynchronous requests	حداکثر زمانی که یک پردازش غیرهمزمان مجاز به اجرا می باشد. پردازش هایی که در حالت غیرهمزمان اجرا می شوند باید در زمان محدود تعیین شده به اتمام برسند در غیر این صورت به طور اتوماتیک کنار گذاشته می شوند.
Maximum queue and execution time for asynchronous requests	حداکثر زمانی که یک پردازش غیرهمزمان مجاز به ماندن در نوبت و اجرایش می باشد. پردازش هایی که در حالت غیر همزمان اجرا می شوند باید در زمان محدود مجاز تمام گردند. در غیر این صورت به طور اتوماتیک کنار گذاشته می شوند.
Resource settings	
Resource expiration timeout	تعداد ثانیه هایی که نتیجه ی اجرای غیرهمزمان برای کاربر به منظور بازیابی در دسترس خواهد بود. به محض اتمام این زمان، این منابع مستعد پاکسازی می گردند (که در بازه های زمانی مشخص انجام میگردد).
Resource storage directory	مکانی روی دیسک که ورودی ها، منابع خروجی و موقت مرتبط با یک پردازش مشخص نگهداری می شوند. مکان پیش فرض temp/wps در data directory خود ژئوسرور می باشد.
External output directory	برخی پردازش ها اجازه ی ذخیره سازی خروجی ها را در یک دایرکتوری خارجی نیز میدهند.

۲-۲-۳- Process status page

علاوه بر موارد مذکور، صفحه ی وضعیت پردازش در بخش About and Status موجود می باشد که در رابطه با پردازش های در حال اجرا و آنهایی که اخیرا به پایان رسیده اند گزارشی می دهد (شکل ۸).

Process status								
Lists all running and recently completed processes								
Dismiss selected processes								
<< < 1 > >> Results 1 to 2 (out of 2 items)			Search					
S/A	Node	User	Process name	Created	Phase	Progress	Task	
☐	A	192.168.2.42	anonymous	gs:BufferFeatureCollection	26/11/14	RUNNING	66,333	Writing outputs
☐	A	192.168.2.42	anonymous	gs:BufferFeatureCollection	26/11/14	RUNNING	0	Retrieving/parsing process input: features
<< < 1 > >> Results 1 to 2 (out of 2 items)								

شکل ۸- Process Status

جدول حاصله حاوی یکسری اطلاعات می باشد، که ستونهای آن به شرح زیر می باشند:

- S/A: اجرای به صورت همزمان یا غیرهمزمان (synchronous or asynchronous)
 - Node: نام ماشین یا سیستمی که پردازش را اجرا می کند.
 - User: کاربری که فرایند اجرا را آغاز کرده است.
 - Process name: پردازش اصلی که در حال اجرا می باشد. (chain process در این جدول نشان داده نخواهد شد).
 - Created: زمانی که پردازش ایجاد شده است.
 - Phase: وضعیت موجود در چرخه ی عمر پروژه
 - Progress: پیشرفت فعلی
 - Task: در طی فرآیند پردازش، در حال حاضر مشغول انجام چه کاری میباشد.
- فازهای اجرایی در ژئوسرور می تواند حالات زیر را داشته باشد (در حقیقت مقادیری است که ستون phase می تواند به خود اختصاص دهد):

جدول ۴- فازهای اجرایی در ژئوسرور

QUEUED	پردازش منتظر اجرا شدن می باشد.
RUNNING	پردازش یا در حال بازیابی و تجزیه کردن ^۱ ورودی ها برای محاسبه و یا نوشتن نتایج می باشد.
FAILED	اجرای پردازش با شکست به پایان رسیده است.
SUCCESS	اجرای پردازش با موفقیت به پایان رسیده است.
DISMISSING	اجرای پردازش نادیده گرفته شده است. وابسته به ماهیت پردازش ممکن است آنی باشد یا مدتی طول بکشد.

تمامی اجراهای نمایش داده شده در جدول ۴ قابلیت انتخاب و منفصل کردن را با استفاده از لینک Dismiss selected processes در بالای جدول دارند. برخلاف شبه عملگر Dismiss (در قسمت توابع توضیح داده شده) این رابط کاربری اجازه منفصل کردن پردازش های همزمان را نیز می دهد. زمانی که نادیده گرفتن پردازش به اتمام رسید، اجرای پردازش از جدول فوق ناپدید می شود.

لازم به ذکر است پردازش های اتمام یافته نیز می توانند نادیده گرفته شوند. این کار منجر به حذف آنی تمام منابع مرتبط با پردازش بر روی دیسک می گردد.

WPS security page - ۴-۲-۲

امنیت سرویس ژئوسرویس به طور معمول مبتنی بر تنظیمات کلی امنیت OGC می باشد. اما در مورد WPS نیاز به محدود کردن دسترسی به تک تک پردازش ها نیز احساس می شود. به همان صورتی که امنیت داده، دسترسی به لایه ها را محدود می کند.

Enabled	Group prefixes	Group title	Summary	Roles	Child processes
☒	JTS, gs, gt	Deprecated processes	All processes active		Manage
☒	gs	GeoServer specific processes	All processes active		Manage
☒	geo	Geometry processes	All processes active	ADMIN;	Manage
☒	polygonlabelprocess	PolygonLabelProcess	All processes active		Manage
☒	ras	Raster processes	All processes active		Manage
☒	vec	Vector processes	All processes active		Manage

Input limit defaults

Maximum size for complex inputs, MB (0 for no limit)

10

Complex Inputs

☐ Disable loading complex inputs from remote references

Process Access Mode

☒ HIDE

☐ MIXED

☐ CHALLENGE

شکل ۹- WPS Security

تنظیمات امنیتی WPS می تواند در بخش Security و سپس WPS security انجام شود (شکل ۹). در حقیقت، لیست role هایی که به یک پروسه یا مجموعه ای از پروسه ها ضمیمه می شود مشخص کننده ی آن است که کدام کاربر به می تواند به کدام پردازش / مجموعه پردازش ها دسترسی داشته باشد. در صورتی که لیست role ها خالی باشد نشان دهنده آن است که پردازش (ها) توسط تمام کاربران قابلیت دسترسی خواهد داشت، مگر آنکه از حالت enabled به disabled تغییر داده شوند که در آن صورت توسط هیچ کس در دسترس نخواهد بود. رشته‌هایی که در ستون Roles نوشته می شود باید لیستی از role ها باشد که با نقطه ویرگول (;) از یکدیگر جدا شده اند. اطلاعات وارد شده در این ستون همچنین قابلیت تکمیل خودکار و کپی و پیست لیست role ها را از یک پردازش به دیگری فراهم میکند (شکل ۱۰).

Group title	Summary	Roles	Ch
Deprecated processes	All processes active	G	Ma
GeoServer specific processes	All processes active	GROUP_ADMIN;	Ma

شکل ۱۰- Roles

در ادامه، سه بخش انتهایی Input limit defaults، Complex Inputs و Process Access Mode در صفحه WPS security توضیح داده می شوند.

❖ Input limit defaults

میزان منابع مورد استفاده توسط یک پردازش معمولاً به طور مستقیم وابسته به پارامترهای ورودی می باشد. با در نظر گرفتن این مساله، مدیران میتوانند سه نوع محدودیت بر روی ورودی های هر پردازش تعیین کنند:

○ حداکثر حجم ورودی های Complex

○ محدوده ی مقادیر قابل قبول برای مقادیر عددی

○ حداکثر تعدد ورودی های تکرار پذیر

حداکثر حجم می تواند در صفحه ی WPS Security page یک مقدار کلی پیش فرض داده شود. همچنین تعریف محدودیت به ازای هر پردازش در قسمت Limits ممکن می باشد: در هر گروه وارد manage شده و به ازای پردازش مد نظر در ستون Limits و اعمال محدودیت مد نظر می کنیم.

Enabled	Name	Description	Roles	Limits
☒	JTS:area	Returns the area of a geometry, in the units of the geometry. Assumes a Cartesian plane, so this process is only recommended for non-geographic CRSes.		Edit... *
☒	JTS:boundary	Returns a geometry boundary. For polygons, returns a linear ring or multi-linestring equal to the boundary of the polygon(s). For linestrings, returns a multipoint equal to the endpoints of the linestring. For points, returns an empty geometry collection.		Edit...
☒	JTS:buffer	Returns a polygonal geometry representing the input geometry enlarged by a given distance around its exterior.		Edit... *
☒	JTS:centroid	Returns the geometric centroid of a geometry. Output is a single point. The centroid point may be located outside the geometry.		Edit...

شکل ۱۱- WPS Security : manage

همانطور که در شکل ۱۱ مشاهده می‌شود پردازش‌هایی که در کنار Edit دارای علامت * می‌باشند محدودیت برایشان تعریف شده می‌باشد.

Process limits for ras:Contour

Edit process input limits

Input name	Limit Type	Editor
data	Max input size MB	Max 20
band	Min/Max values	Min Max
levels	Max input multiplicity	Max 200
levels	Min/Max values	Min 0 Max 10000
interval	Min/Max values	Min Max
roi	Max input size MB	Max 20

شکل ۱۲- نمونه ای از محدودیت‌ها

این محدودیت می‌تواند بر تمامی input های یک پردازش مشخص که قابلیت اش را داشته باشند اعمال گردد.

❖ Complex Inputs

به صورت پیش فرض، درخواست‌های Execute، بارگذاری ورودی‌های Complex از نوع refrence به local files و یا سرورهای خارجی پشتیبانی می‌کند. این رفتار قابلیت محدود شدن را در این بخش دارد. در صورت تیک خوردن گزینه Disable loading complex inputs from remote references درخواست Execute با ورودی از نوع refrence که از نوع درخواست WCS/WFS/WPS داخلی نباشد منجر به ایجاد خطا می‌گردد.

❖ Process Access Mode

قسمت تنظیمات process access mode نحوه ی انتشار پردازش‌های ایمن و نوع رفتار در زمان دسترسی به یک پردازش ایمن بدون امتیاز لازم برای دسترسی را در ژئوسرور مشخص می‌کند. سه حالت برای آن تعریف شده است:

- **HIDE (default):** پردازش‌هایی که در دسترس کاربر فعلی نیست، از دید او مخفی می‌شوند (در سندهای مربوط به capabilities برای او لیست نمی‌شوند).
- **CHALLENGE:** تمامی پردازش‌ها در سند capabilities نمایش داده می‌شود. اما در صورت درخواست یکی از پردازش‌های ایمن شده توسط کاربری که امتیاز دسترسی ندارد، باید احراز هویت انجام شود.
- **MIXED:** پردازش‌های ایمن شده برای کاربرانی با عدم امتیاز دسترسی کافی در سند capabilities نمایش داده نخواهد شد، اما درخواست احراز هویت همچنان در هنگام درخواست یک پردازش ایمن شده صورت می‌گیرد.

۲-۳- دامنه سرویس

در اینجا یک سازوکار عمومی، به منظور توصیف و فعال سازی فرآیندهای مکانی تحت وب، توصیف داده های مورد نیاز جهت اطلاعات ورودی و خروجی یک فرآیند، مشخص می شود. این داده ها می توانند از طریق شبکه و سرور قابل دسترس باشند. داده ها می تواند شامل فرمت های داده رستری مانند GeoTIFF یا استانداردهای تبادل داده مانند GML باشد. ورودی های داده می توانند به صورت استاندارد با سرویس های استاندارد OGC در ارتباط باشند. به عنوان مثال، یک ورودی داده برای عملیات تقاطع می تواند یک پلیگون باشد که در پاسخ به درخواست WFS به دست می آید، در این صورت ورودی داده WPS از پرسش (کوئری) حاصل از WFS خواهد بود.

۲-۴- نمای کلی WPS

فعال کردن پردازش مکانی در اینترنت مستلزم توسعه طیف گسترده ای از خدمات وب برای پشتیبانی از عملیات مکانی تجزیه ناپذیر (اتمیک) و همچنین قابلیت های مدلسازی پیچیده است. استاندارد کردن روش فراخوانی این فرآیندها به منظور کاهش میزان برنامه نویسی مورد نیاز، و تسهیل در اجرا و استفاده از سرویس های جدید اهمیت زیادی دارد. سرویس WPS برای دستیابی به این اهداف در نظر گرفته شده است.

۲-۵- توابع WPS

استاندارد WPS نسخه ی ۱,۰,۰ سه عملگر را برای بررسی و اجرای پردازش های مکانی تعریف میکند، به نحوی که ژئوسرور قابلیت های این سه عملگر را با دو شبه عملگر^۱ گسترش می دهد. اجرای سه عملیات اصلی توسط همه سرورها الزامی است به نحوی که توسط یک کاربر، درخواست شده و توسط یک سرور WPS انجام گیرد. این عملیات ها عبارتند از:

جدول ۵- عملگرهای استاندارد WPS

عملیات	توضیحات
GetCapabilities	این عملگر یک سند XML برمیگرداند که قابلیت های WPS، تحلیل ها و نحوه فراخوانی آنها را بیان می کند.
DescribeProcess	این عملگر نیز یک سند XML برمیگرداند که تحلیل مورد نظر را با جزییات توصیف میکند و شامل جزییاتی به منظور فراهم آوردن ورودی های مناسب و فهم پاسخ می باشد.
Execute	عملگری است که کلاینت در زمان نیاز به فراخوانی یک تحلیل خاص صدا می زند.
Dismiss	به منظور لغو اجرای پردازش
GetExecutions	بازیابی لیستی از وضعیت اجراهای جاری

لازم به ذکر است دو عملگر Dismiss و GetExecutions مختص ژئوسرور می باشند.

^۱ pseudo-operator

۲-۵-۱- GetCapabilities

این عملیات به کاربر اجازه می‌دهد تا اسناد فراداده (یا قابلیت‌ها) سرویس‌ها را درخواست و دریافت کند. عملیات GetCapabilities نام و توضیحات کلی هر یک از فرآیندهای ارائه شده توسط یک نمونه WPS را ارائه می‌دهد. یک سرور WPS که به درخواست GetCapabilities پاسخ می‌دهد، فراداده‌های مربوط به سرویس، از جمله عملیات و فرآیندهای پشتیبانی شده را باز می‌گرداند.

نمونه‌ای از درخواست GetCapabilities که از طریق روش HTTP Get ارسال می‌شود، به صورت ذیل می‌باشد.

<http://localhost:8080/geoserver/ows?service=WPS&version=2.0.0&request=GetCapabilities>

پاسخ دریافتی به صورت شکل ۱۳ قابل مشاهده می‌باشد.

```
<ows:DCP>
  <ows:HTTP>
    <ows:Get xlink:href="http://iransdi.ncc.gov.ir/VB/fars/wps"/>
    <ows:Post xlink:href="http://iransdi.ncc.gov.ir/VB/fars/wps"/>
  </ows:HTTP>
</ows:DCP>
<ows:Operation>
  <ows:Operation name="Execute">
    <ows:DCP>
      <ows:HTTP>
        <ows:Get xlink:href="http://iransdi.ncc.gov.ir/VB/fars/wps"/>
        <ows:Post xlink:href="http://iransdi.ncc.gov.ir/VB/fars/wps"/>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
</ows:OperationsMetadata>
<ows:ProcessOfferings>
  <ows:Process wps:processVersion="1.0.0">
    <ows:Identifier>JTS:area</ows:Identifier>
    <ows:Title>Area</ows:Title>
    <ows:Abstract>Returns the area of a geometry, in the units of the geometry. Assumes a Cartesian plane, so this process is only recommended for non-geographic CRSes.</ows:Abstract>
  </ows:Process>
  <ows:Process wps:processVersion="1.0.0">
    <ows:Identifier>JTS:boundary</ows:Identifier>
    <ows:Title>Boundary</ows:Title>
    <ows:Abstract>Returns a geometry boundary. For polygons, returns a linear ring or multi-linestring equal to the boundary of the polygon(s). For linestrings, returns a multipoint equal endpoints of the linestring. For points, returns an empty geometry collection.</ows:Abstract>
  </ows:Process>
  <ows:Process wps:processVersion="1.0.0">
    <ows:Identifier>JTS:buffer</ows:Identifier>
    <ows:Title>Buffer</ows:Title>
    <ows:Abstract>Returns a polygonal geometry representing the input geometry enlarged by a given distance around its exterior.</ows:Abstract>
  </ows:Process>
  <ows:Process wps:processVersion="1.0.0">
    <ows:Identifier>JTS:centroid</ows:Identifier>
    <ows:Title>Centroid</ows:Title>
    <ows:Abstract>Returns the geometric centroid of a geometry. Output is a single point. The centroid point may be located outside the geometry.</ows:Abstract>
  </ows:Process>
  <ows:Process wps:processVersion="1.0.0">
```

شکل ۱۳- پاسخ درخواست GetCapability

همانطور که در شکل نشان داده شده است خلاصه اطلاعاتی در مورد سرویس‌های تحلیلی پیاده‌سازی شده در یک سرور را نمایش می‌دهد.

۲-۵-۲- DescribeProcess

این عملیات به کاربر اجازه می‌دهد تا اطلاعات دقیقی درباره فرآیندهایی که می‌توانند در سرویس اجرا شوند، از جمله ورودی‌های مورد نیاز، فرمت‌های مجاز آن‌ها و خروجی‌هایی که می‌توان تولید، درخواست و دریافت کرد را ارائه دهد. پارامتر identifier تحلیلی که قرار است توصیف شود و توضیحاتی در رابطه با آن ارائه شود را مشخص می‌کند. چندین تحلیل می‌

تواند مورد درخواست قرار گیرد بدین صورت که باید توسط کاما (:) از یکدیگر جدا شوند (به طور مثال `identifier=JTS:buffer,gs:Clip`). طی این درخواست حداقل یک تحلیل باید انتخاب شود. قابل ذکر است همانند تمامی پارامترهای OGC، کلمات کلیدی نظیر `request`، `version` و ... غیرحساس به حروف بزرگ و کوچک بوده و مقادیری نظیر `GetCapabilities`، `JTS:buffer` و ... به حروف بزرگ و کوچک حساس می باشند. به عنوان نمونه درخواست تحلیل `JTS:buffer` به صورت شکل ۱۴ می باشد.

```
http://localhost:8080/geoserver/ows?
service=WPS&
version=1.0.0&
request=DescribeProcess&
identifier=JTS:buffer
```

شکل ۱۴- درخواست `DescribeProcess`

پاسخ آن در قالب xml شامل اطلاعات موجود در جدول ۶ می باشد.

جدول ۶- اطلاعات موجود در پاسخ `DescribeProcess`

Title	"Buffers a geometry using a certain distance"
Inputs	geom: "The geometry to be buffered" (<i>geometry, mandatory</i>)
	distance: "The distance (same unit of measure as the geometry)" (<i>double, mandatory</i>)
	quadrant segments: "Number of quadrant segments. Use > 0 for round joins, 0 for flat joins, < 0 for mitred joins" (<i>integer, optional</i>)
	capstyle: "The buffer cap style, round, flat, square" (<i>literal value, optional</i>)
Out	One of the GeoServer supported formats for Geometry encoding

به طور کلی، تحلیل های GeoServer از ورودی های پیچیده ای نظیر `geometry`، `feature collection` و `raster` استفاده می کند. فرمت های موجود برای ورودی و خروجی هر کدام از پردازش ها / تحلیل ها معمولاً توسط کلاس های ^۱ PPIO و رمزگشا / رمزگذارهای عمومی^۲ مشخص می گردند که توسط سازوکار WPS به هر کدام از تحلیل ها منتصب می شود. لازم به ذکر است مجموعه PPIO ها با نصب یکسری افزونه هایی قابلیت گسترش دارند. در هسته مرکزی WPS، PPIO های زیر موجود می باشند.

^۱ Process Parameter IO classes

^۲ generic encoders/decoders

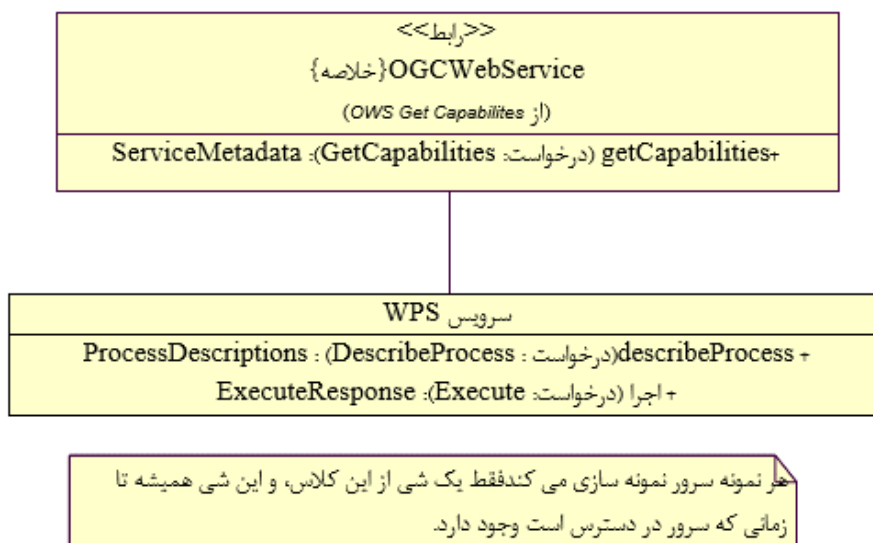
جدول ۷- PPIO

geometry	GML2, GML3, GeoJSON, WKT
feature collection	GML2, GML3, GeoJSON, Zipped shapefile, KML
raster	GeoTIFF, PNG, JPEG

۲-۵-۳- Execute

این عملیات به کاربر اجازه می دهد تا با استفاده از مقادیر پارامتر ورودی ارائه شده و برگرداندن خروجی های تولید شده، یک فرآیند مشخص را اجرا کند که توسط WPS پایاده سازی شده است. این عملیات شباهت های زیادی به سایر سرویس های تحت وب استاندارد OGC از جمله WFS، WMS و WCS دارند.

شکل ۱۵ یک نمودار UML ساده است که رابط WPS را خلاصه می کند. این نمودار کلاس نشان می دهد که کلاس واسط WPS عملیات GetCapabilities را از کلاس رابط OGCWebService به ارث می برد و عملیات DescribeProcess و Execute را اضافه می کند. تست این عملیات به طور کامل در قسمت مراحل راه اندازی سرویس WPS در ژئوسرور توضیح داده خواهد شد.



برای مثال، حالت ساده فرآیندی را در نظر بگیرید که می تواند تقاطع دو پلیگون را نشان دهد. پاسخ به درخواست GetCapabilities ممکن است نشان دهد که WPS از عملیاتی به نام "تقاطع" پشتیبانی می کند و این عملیات محدود به تقاطع یک پلیگون با پلیگون دوم است. پاسخ به یک درخواست DescribeProcess برای فرآیند «تقاطع» ممکن است نشان دهد که به دو ورودی نیاز دارد، یعنی: «FirstPolygon» و «SecondPolygon»، و این ورودی ها باید در GML ۲٫۲ ارائه شوند. علاوه بر این، این فرآیند یک خروجی را در GML ۲٫۲ یا GML ۳٫۱ تولید می کند و می تواند به عنوان یک منبع در دسترس وب تحویل داده شود. کلاینت فرآیند را با فراخوانی عملیات Execute اجرا می کند و ممکن است انتخاب

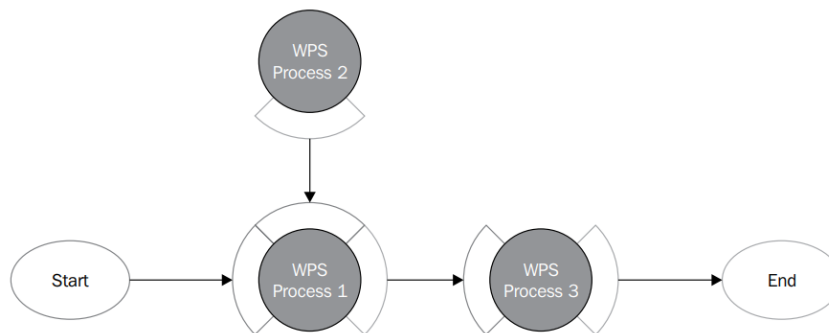
کند که دو پلیگون ورودی را که مستقیماً در داخل درخواست تعبیه شده اند ارائه دهد و تشخیص دهد که خروجی باید به عنوان یک منبع قابل دسترسی تحت وب ذخیره شود. پس از تکمیل، فرآیند یک سند XML ExecuteResponse را برمی گرداند که ورودی ها و خروجی ها را شناسایی می کند، نشان می دهد که آیا فرآیند با موفقیت اجرا شده است یا خیر.

۲-۶- تحلیل WPS

یک تحلیل WPS یک واحد اتمی سرویس WPS می باشد. این تحلیل در حقیقت کد پردازش مکانی می باشد که توسط درخواست Execute در سمت سرور اجرا می شود. یک تحلیل WPS می تواند شامل چندین ورودی دارای انواع مختلف باشد که به صورت value یا reference می تواند مشخص گردد. همچنین تحلیل WPS می تواند خروجی های متعددی را برگرداند، به همین دلیل یک تحلیل صرفاً به یک پارامتر خروجی وابسته نمی باشد. استفاده از خروجی یک تحلیل به عنوان ورودی یک تحلیل دیگر نیز امکان پذیر می باشد که از آن با WPS process chaining یاد می شود.

۲-۶-۱- Process chaining

انجام یک تحلیل WPS به منظور اجرای یکسری تحلیل ها صرفاً امکان استفاده محدودی از تحلیل های مکانی را به ما می دهد. در صورتی که بخواهیم یکسری آنالیزهای پیچیده تر را انجام بدهیم، نیازمند اجرای تعداد زیادی تحلیل های WPS می باشیم. به طوری که هر بار خروجی را ذخیره و و در تحلیل بعدی استفاده کنیم.



شکل ۱۶- Process Chaining

۲-۶-۲- WPS Request Builder

افزونه WPS ژئوسرور شامل یک Request Builder به منظور تست پردازش های WPS نیز می باشد. به منظور دسترسی به آن ابتدا Demos انتخاب و سپس WPS Request Builder از لیست انتخاب می شود (شکل ۱۷).

GeoServer Demos

Collection of GeoServer demo applications

- [Demo requests](#) Example requests for GeoServer (using the TestServlet).
- [SRS List](#) List of all SRS known to GeoServer
- [WCS request builder](#) Step by step WCS GetCoverage request builder
- [WPS request builder](#) Step by step WPS request builder

شکل ۱۷- GeoServer Demos

WPS Request Builder اساساً شامل یک کادر به منظور لیست نمودن پردازش‌های موجود و چند دکمه برای ارسال درخواست WPS و نمایش درخواست POST می‌باشد. وابسته به نوع پردازش و نوع ورودی انتخاب شده صفحه‌ی نمایش تغییر میکند. به طور مثال، پردازش‌های در گروه JTS می‌توانند feature collection های مبتنی بر GML/WKT و یا URL refrence و یا subprocess را به عنوان ورودی بگیرند. در حالی که پردازش‌های در گروه gs علاوه بر ورودی‌های ذکر شده قابلیت انتخاب لایه‌های ژئوسرور را نیز به عنوان ورودی دارد.

WPS request builder

Step by step WPS request builder.

Choose process

gs:Bounds

Computes the bounding box of the input features. (WPS DescribeProcess)

Process inputs

features* - FeatureCollection

Input feature collection

VECTOR_LAYER ▾ topp:states ▾

Process outputs

bounds* - ReferencedEnvelope

Bounding box of input features

☒ Generate

Authentication

☐ Authenticate (will run the request as anonymous otherwise)

Execute in Demo Requests

Execute process in New Page

Generate XML from process inputs/outputs

شکل ۱۸- WPS request builder

با کلیک بر روی دکمه‌ی Generate XML from process inputs/outputs می‌توان پردازش را در قالب درخواست POST مشاهده نمود. و با Execute Process in New Page نتایج در پنجره‌ای نمایش داده می‌شود (شکل ۱۹).

```
<ows:BoundingBox xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:ows="http://www.opengis.net/ows/1.1" crs="EPSG:4326">
  <ows:LowerCorner>-124.73142200000001 24.955967</ows:LowerCorner>
  <ows:UpperCorner>-66.969849 49.371735</ows:UpperCorner>
</ows:BoundingBox>
```

شکل ۱۹- Execute Process in New page

مشاهده و اجرای درخواست در Demo Requests نیز با استفاده از دکمه‌ی Execute Process in Demo Requests

محقق می‌شود.

۷-۲- ماهیت عمومی و میان افزار WPS

WPS امکان ارائه داده های ورودی را در دو روش مختلف فراهم می کند. داده ها را می توان در درخواست Execute جاسازی کرد یا به عنوان یک منبع قابل دسترسی تحت وب اشاره کرد. در رویکرد قبلی، WPS به عنوان یک سرویس مستقل عمل می کند. در حالت دوم، WPS به عنوان سرویس میان افزار برای داده ها، با به دست آوردن داده ها از یک منبع خارجی به منظور اجرای یک فرآیند عمل می کند.

WPS به رابط های نرم افزاری موجود اجازه می دهد تا به عنوان خدمات وب به شبکه ارائه شوند. بنابراین پیاده سازی WPS را می توان میان افزار برای نرم افزار در نظر گرفت. در حقیقت، WPS یک رابط عمومی است که فرایندهای اجرایی آن ها را با ورودی و خروجی های مربوطه پشتیبانی می کند. WPS را می توان به عنوان یک مدل انتزاعی از یک وب سرویس در نظر گرفت، که برای آن باید پروفایل هایی برای پشتیبانی از موارد مورد استفاده توسعه داده شود و برای پشتیبانی از قابلیت همکاری استاندارد شود. همانند سایر مشخصات OGC GML، این توسعه، انتشار و پذیرش پروفایل ها است که کاربردهای خاص این مشخصات را تعریف می کند.

مکانیسم های کشف و اتصال WPS، از استانداردهای OGC که مشابه آن در WMS و WFS تنظیم شده است، پیروی می کند. به این صورت که WPS یک عملیات GetCapabilities را تعریف نموده و درخواست ها بر اساس GET و POST در پروتکل HTTP صورت می گیرند. سرویس WPS یک رابط درخواست/پاسخ را مشخص می کند که چگونه:

- درخواست ها را برای اجرای فرآیند رمزگذاری کنید.
- پاسخ های حاصل از اجرای فرآیند را رمزگذاری کنید.
- داده ها و فراداده ها را در ورودی/خروجی های اجرای فرآیند جاسازی کنید.
- ورودی/خروجی داده های قابل دسترسی تحت وب را مرجع کنید.
- از فرآیندهای طولانی مدت پشتیبانی کنید.
- اطلاعات وضعیت فرآیند را بازگردانید.
- خطاهای پردازش را برگردانید.
- ذخیره سازی خروجی های فرآیند را درخواست کنید.

در حقیقت، سرویس WPS یک متدی به منظور انتشار پردازش های مکانمند توصیف و ارایه می دهد. اما به طور خاص مشخص نمی کند که آن پردازش ها چه باید باشند. بنابراین سرورهایی که سرویس WPS را پیاده میکنند در رابطه با اینکه چه پردازش هایی را پیاده و به چه صورت پیاده کنند، آزادی عمل کاملی دارند. این بدین معنی است که انتظار نمی رود یک درخواست پردازشی که برای یک نوع از WPS طراحی شده بر روی نوع دیگر این سرویس کار بکند. ژئوسرور پردازش ها را بر اساس موضوع و کتابخانه های مورد استفاده، دسته بندی می کند. این دسته ها به واسطه یک پیشوند مشخص می شوند که در جدول ۸ نمایش داده شده است.

جدول ۸- دسته بندی پردازش ها در ژئوسرور

۱	geo	geometry processes
۲	ras	raster processes
۳	vec	Vector processes
۴	gs	GeoServer-specific processes

دسته geometry processes با استفاده از ^۱ JTS Topology Suite ساخته می شوند. JTS شامل توابعی از کتابخانه جاوا به منظور پردازش هندسه در دو بعد می باشد. JTS شامل توابع رایج مکانی نظیر area, buffer و intersection می باشد. سرویس WPS ژئوسرور یکسری از این توابع را در گروه پردازش های geo پیاده میکند. اسم و تعریف توابع میتواند تغییر کنند که برای دیدن مجموعه ی کاملی از آن می توان به سند Capabilities سرویس WPS مراجعه کرد.

سرویس WPS ژئوسرور شامل یکسری پردازش هاست که صرفا برای کار با ژئوسرور ایجاد شده اند. این پردازش ها معمولا توابع خاص ژئوسرور نظیر bounds و reprojection می باشند (GeoServer-specific). این پردازش ها از یکسری اتصال داخلی به سرویس های WFS/WCS ژئوسرور برای خواندن و نوشتن داده استفاده میکنند.

فصل سوم

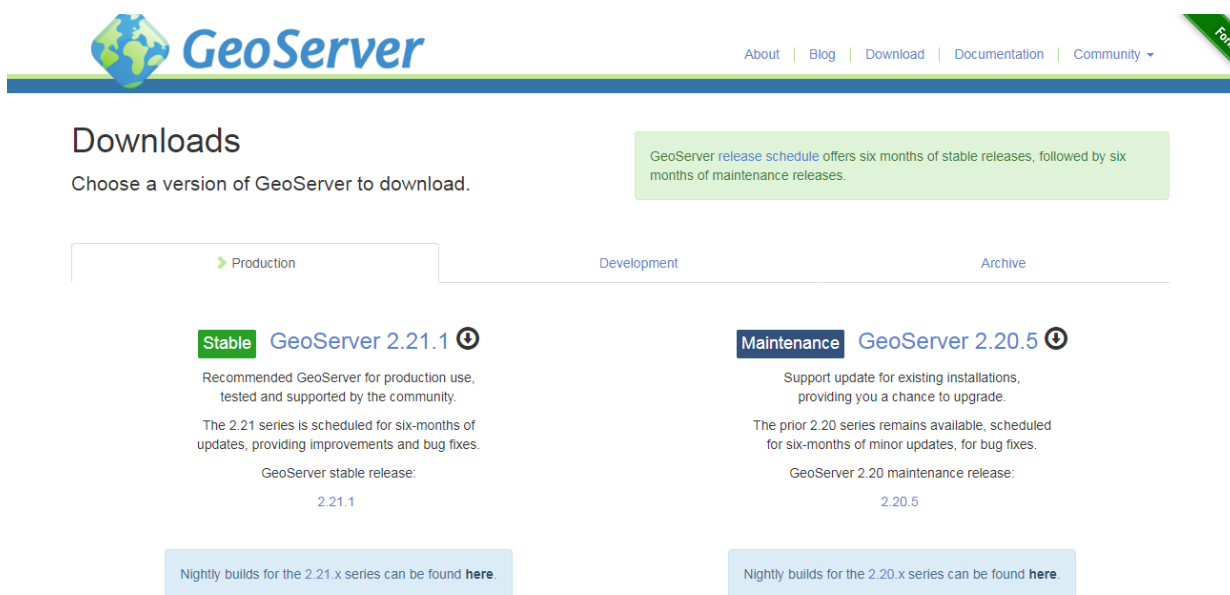
نصب و راه اندازی WPS در

ژئوسرور

۳-۱- مراحل راه اندازی سرویس WPS در ژئوسروور

سرویس WPS به طور پیش فرض بخشی از ژئوسروور نمی‌باشد، اما به عنوان یک افزونه در دسترس و قابل نصب است. جهت دانلود و نصب آن مراحل زیر را دنبال کنید:

۱. وارد سایت ژئوسروور به مسیر <https://geoserver.org> شده و لینک download را انتخاب کنید.



شکل ۲۰- صفحه دانلود GeoServer

۲. از تب Archive صفحه‌ای را که با نسخه GeoServer شما دقیقاً مطابقت دارد انتخاب نمایید.



شکل ۲۱- دانلود نسخه‌های GeoServer

۳. از قسمت Extensions بخش Services گزینه WPS را انتخاب نمایید. بعد از گذشت چند ثانیه فایل به طور خودکار دانلود خواهد شد. توجه داشته باشید که حتما نسخه افزونه با نسخه نرم‌افزار ژئوسرور مطابقت داشته باشد، در غیر این صورت با خطا مواجه خواهید شد.

Extensions



Extensions

GeoServer Extension downloads.

Vector Formats

- App Schema
- ArcSDE
- DB2
- H2
- MySQL
- Oracle
- Pregeneralized Features
- SQL Server
- Teradata
- MongoDB

Cartography

- Chart Symbolizer
- CSS Styling
- MBStyle Styling
- Printing
- YSLD Styling

Miscellaneous

- Control Flow
- Cross Layer Filtering
- Monitor (Core)
- Importer (Core, BDB Backend)
- INSPIRE
- Resource Browser Tool

Coverage Formats

- GDAL
- GRIB
- Image Pyramid
- JDBC Image Mosaic
- JPEG2K
- NetCDF

Output Formats

- DXF
- Excel
- Image Map
- JPEG Turbo
- NetCDF
- OGR (WFS, WPS)
- Vector Tiles
- XSLT

Security

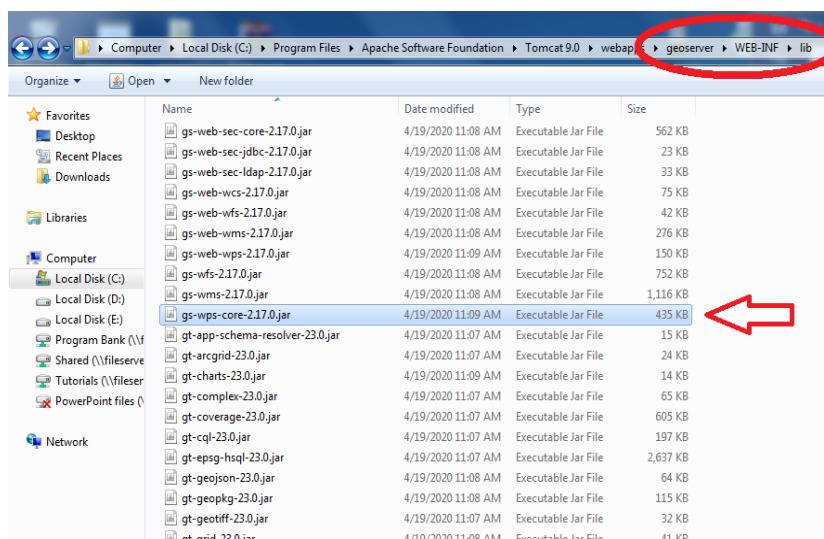
- Key authentication
- CAS
- GeoFence (Client, Server)

Services

- CSW
- WCS 2.0 EO
- **WPS**
- WPS Hazelcast
- SLDService

شکل ۲۲- صفحه دانلود افزونه‌های WPS

۴. فایل را از حالت فشرده خارج نموده و در مسیر نصب ژئوسرور، فولدر WEB-INF و در نهایت فولدر lib کپی نمایید.

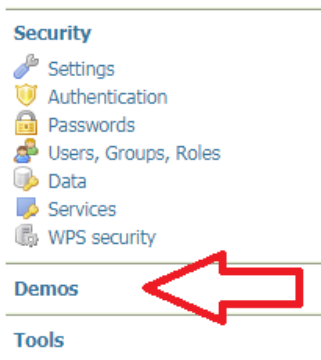


شکل ۲۳- قرار دادن افزونه WPS در مسیر lib در نرم‌افزار GeoServer

۵. نرم افزار را ریستارت نمایید.

۳-۲- نحوه پیاده سازی سرویس WPS در ژئوسروور

بعد از نصب افزونه WPS و به منظور پیاده سازی تحلیل های مکانی موجود در ژئوسروور، وارد نرم افزار ژئوسروور شده و روی بخش Demos کلیک می کنیم.



شکل ۲۴- استفاده از ماژول Demos برای ساخت درخواست های GeoServer

از صفحه GeoServer Demos گزینه WPS request Builder را انتخاب می نمایم.



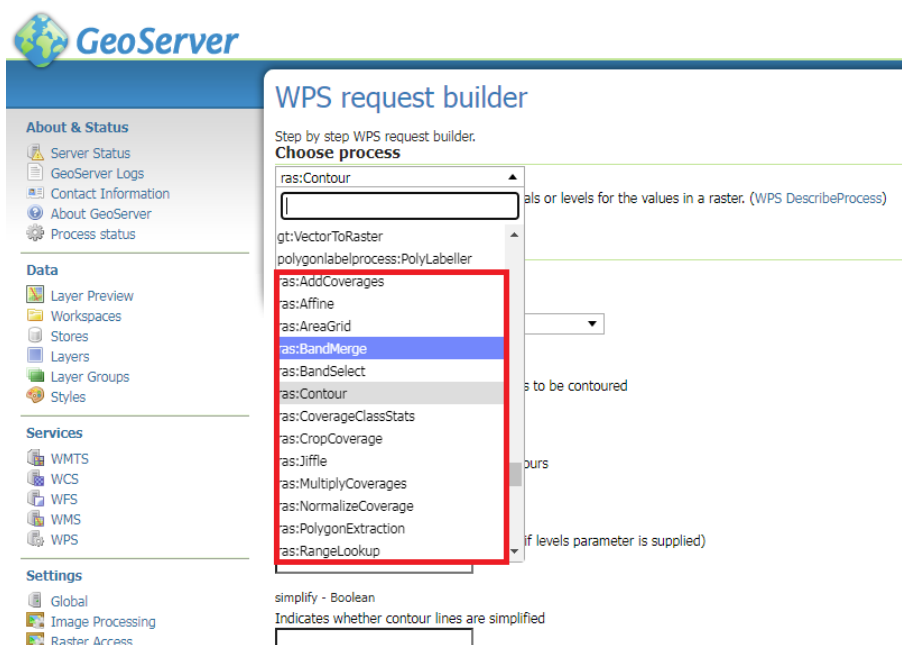
شکل ۲۵- انتخاب گزینه WPS request builder از ماژول Demos

صفحه WPS Request Builder در درجه اول شامل یک کادر انتخابی است که تمام فرآیندهای موجود را فهرست می کند. WPS از انواع مختلفی از فرآیندهای برداری/هندسی پشتیبانی می کند. ما عمدتاً می توانیم آنها را به سه نوع بزرگ طبقه بندی کنیم. برای توابع برداری سه حالت زیر در نظر گرفته می شوند که عبارتند از:

- **فرآیندهای JTS:** یک نگاشت مستقیم از توابع هندسه پایه هستند. مجموعه توپولوژی JTS یک API از مقدمات و توابع پایه مکانی برای پردازش هندسه است. JTS یک کتابخانه جاوا از توابع برای پردازش هندسه در دو بعد است. JTS مطابق با مشخصات ویژگی های ساده برای SQL است که توسط کنسرسیوم فضایی باز

(OGC) ، مشابه PostGIS منتشر شده است. JTS شامل توابع فضایی رایج مانند مساحت، بافر، تقاطع و ساده سازی است.

- **فرآیندهای پرس و جو گر:** این نوع فرآیندها را می توان به عنوان یک توسعه WFS در نظر گرفت. از طریق آنها می توانیم پرس و جوها و توابع پیچیده را برای هر نوع هندسه ای، خارجی نیز صادر کنیم.
 - **فرآیندهای هندسی:** توابع هندسی مثل بافر و تقاطع که بر روی منابع داخلی و خارجی کار می کنند.
- برای استفاده از تحلیل‌های داده‌های رستری می‌توان در طبقه‌بندی مشاهده شده، تحلیل‌های با پیشنهاد Ras را انتخاب نمود.



شکل ۲۶- انتخاب تحلیل مورد نظر منوی Choose process

فصل چهارم

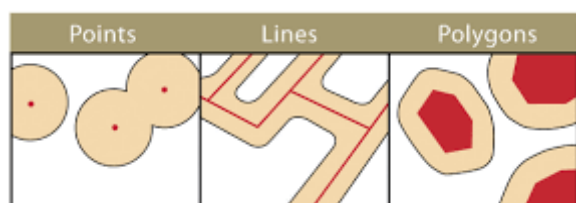
انواع تحلیل های مکانی پایه در

ژئوسرور

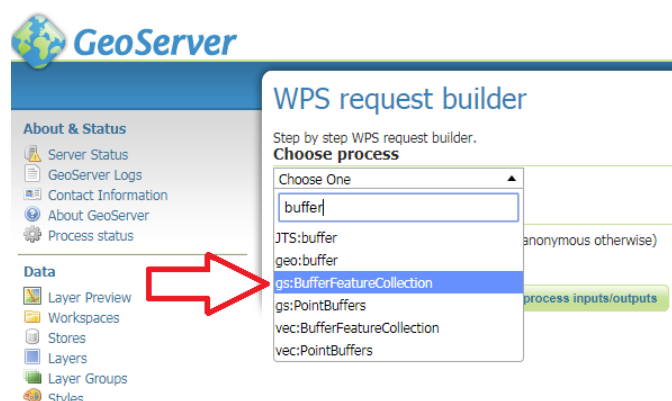
۴-۱- تحلیل Buffer

جدول ۹- تحلیل Buffer

نام تحلیل	نام تابع	توضیحات
Buffer	gs:BufferFeatureCollection	ایجاد محدوده‌ای در اطراف عوارض مثال: محدوده ۵۰۰ متری اطراف میدان آزادی

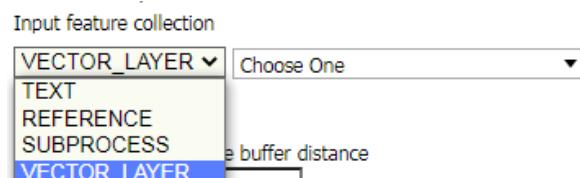


به منظور پیاده‌سازی تحلیل Buffer در نرم‌افزار ژئوسرور، از صفحه WPS request builder و از قسمت Choose process، کلمه buffer را جستجو و گزینه gs:BufferFeatureCollection را انتخاب می‌کنیم.



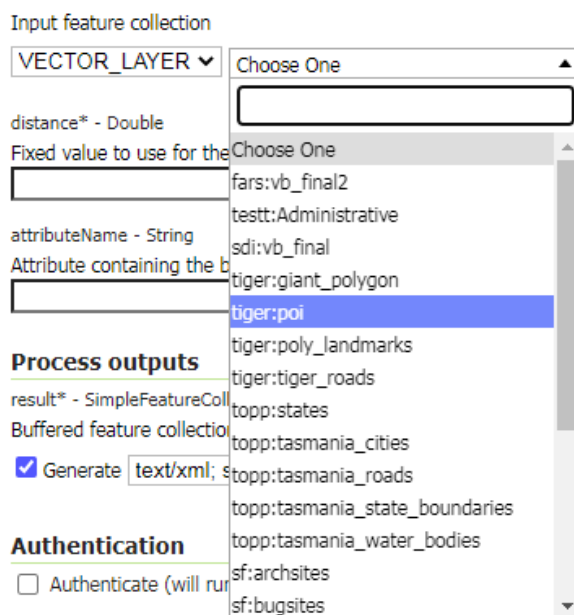
شکل ۲۷- انتخاب تحلیل Buffer از منوی Choose process

از قسمت Input feature collection به صورت پیش فرض گزینه VECTOR_LAYER انتخاب شده است. ابتدا در این قسمت ورودی‌های مختلف ارائه خواهد شد.



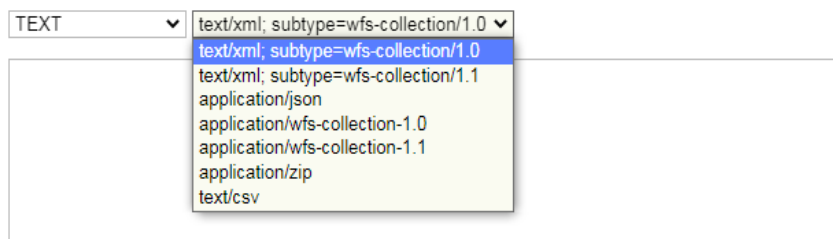
شکل ۲۸- انتخاب نوع ورودی از قسمت Input feature collection

همانطور که در شکل نشان داده است ورودی‌ها در ۴ دسته قرار خواهند گرفت. حالت پیش فرض Vector_Layer است که به طور پیش‌فرض یکی از لایه‌های موجود در یک GeoServer قابل انتخاب خواهد بود.



شکل ۲۹- انتخاب لایه مورد نظر در حالت Vector_Layer به عنوان لایه ورودی

در حالت Text اطلاعات مکانی در قالب یکی از فرمت‌های استاندارد می‌توانند تعریف شوند. این فرمت‌ها در شکل ۳۰ نشان داده شده‌اند.



شکل ۳۰- انتخاب لایه مورد نظر در حالت TEXT به عنوان لایه ورودی

به عنوان مثال شکل زیر حالت استاندارد برای استفاده از فرمت JSON را نشان می‌دهد.

```
{ "type": "FeatureCollection", "features": [ { "type": "Feature", "id": "bugsites.1", "geometry": { "type": "Point", "coordinates": [ 590232, 4915039 ] }, "geometry_name": "the_geom", "properties": { "cat": 1, "str1": "Beetle site" } }, { "type": "Feature", "id": "bugsites.2", "geometry": { "type": "Point", "coordinates": [ 590430, 4915204 ] }, "geometry_name": "the_geom", "properties": { "cat": 2, "str1": "Beetle site" } }, { "type": "Feature", "id": "bugsites.3", "geometry": { "type": "Point", "coordinates": [ 590529, 4914625 ] }, "geometry_name": "the_geom", "properties": { "cat": 3, "str1": "Beetle site" } }, { "type": "Feature", "id": "bugsites.4", "geometry": { "type": "Point", "coordinates": [ 590546, 4915353 ] }, "geometry_name": "the_geom", "properties": { "cat": 4, "str1": "Beetle site" } }, { "type": "Feature", "id": "bugsites.5", "geometry": { "type": "Point", "coordinates": [ 590612, 4915320 ] }, "geometry_name": "the_geom", "properties": { "cat": 5, "str1": "Beetle site" } }, { "type": "Feature", "id": "bugsites.6", "geometry": { "type": "Point", "coordinates": [ 590744, 4915535 ] }, "geometry_name": "the_geom", "properties": { "cat": 6, "str1": "Beetle site" } } ] }
```

برای انتخاب ورودی ها در حالت Reference می توان اطلاعات ورودی سرویس WPS را از یک سرور خارجی با استفاده از استاندارد WFS دریافت نمود.

شکل ۳۱- وارد کردن آدرس URL لایه ورودی مورد نظر در حالت Reference

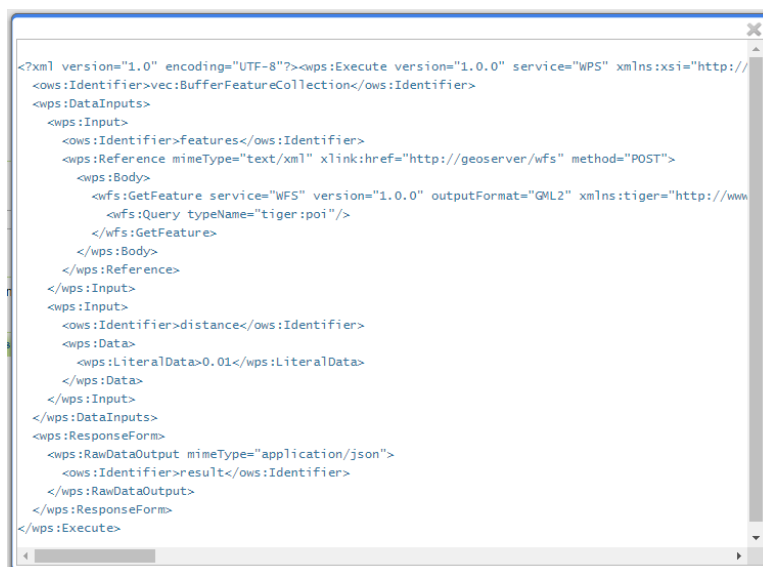
حالت Subprocess یک انتخاب مهم برای ساخت مدل های تحلیلی می باشد. در این حالت ورودی یک تحلیل از یک تحلیل دیگر به دست می آید. پس از تعریف تحلیل بافر، نمونه استفاده از این حالت را در قسمت تحلیل تقاطع به طور کامل توضیح خواهیم داد. در این حالت به عنوان مثال، ورودی تقاطع حریم یک لایه دیگر خواهد بود. لایه مدنظر را انتخاب و سپس در قسمت distance مقدار فاصله مورد نظر را با توجه به سیستم مختصات و سیستم تصویر لایه وارد می نماییم.

شکل ۳۲- وارد کردن مقدار فاصله مورد نظر مربوط به تحلیل Buffer

در قسمت Process outputs و از Generate نوع خروجی فایل بر اساس یک فایل متنی قابل انتخاب است. با انتخاب گزینه Execute process تحلیل فوق اجرا و خروجی در صفحه نمایش قابل مشاهده است (شکل ۳۳).

شکل ۳۳- انتخاب فرمت استاندارد مورد نظر به عنوان خروجی تحلیل Buffer

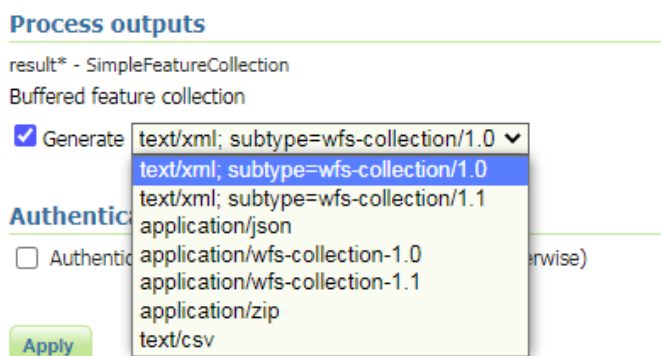
همچنین برای ایجاد درخواست بر اساس یک فایل XML می‌توان از گزینه Generate XML from process inputs/outputs استفاده کرد.



شکل ۳۴- دریافت خروجی تحلیل Buffer در زبان XML

تعریف خروجی‌ها:

پس از اجرای تحلیل می‌توان نوع خروجی تحلیل را مشخص نمود.

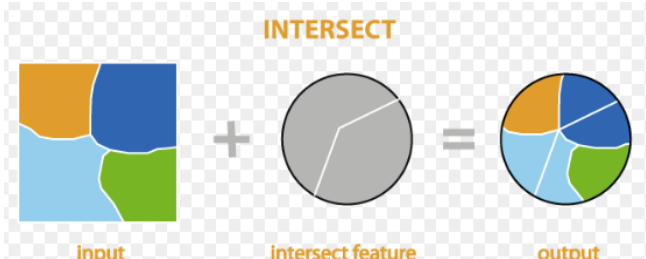


شکل ۳۵- نمایش انواع فرمت خروجی تحلیل‌های WPS در نرم‌افزار GeoServer

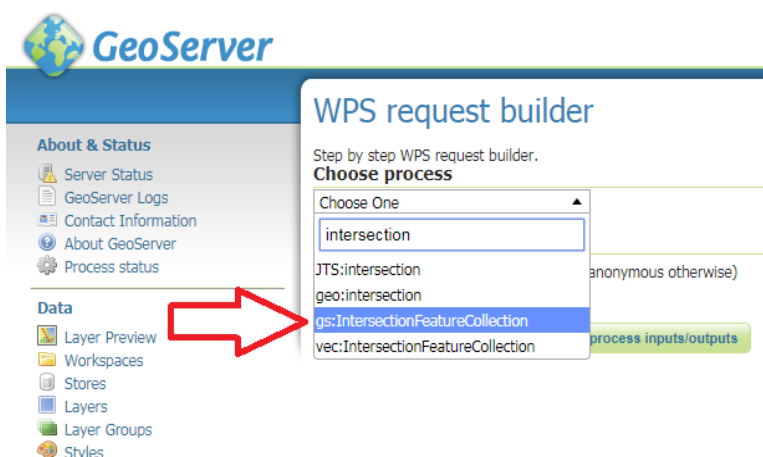
خروجی‌ها می‌تواند به صورت استاندارد WFS، GeoJSON و یا به صورت شیپ فایل زیپ شده تعریف گردد. در هنگام استفاده از شیپ فایل زیپ شده، خروجی پس از اجرای تحلیل دانلود می‌شود.

۲-۴- تحلیل Intersection

جدول ۱۰- تحلیل Intersection

نام تحلیل	نام تابع	توضیحات
Intersection	gs:IntersectionFeatureCollection	خروجی حاصل از دو مجموعه عوارض که دارای همپوشانی مکانی اند مثال: بزرگراه هایی که با محدوده طرح ترافیک تقاطع دارند 

جهت اجرای تحلیل Intersection در نرم افزار ژئوسرور، مجدداً از بخش Demos وارد صفحه WPS request builder شده و از قسمت Choose process ، کلمه Intersection را جستجو و گزینه gs:IntersectionFeatureCollection را انتخاب می کنیم.



شکل ۳۶- انتخاب تحلیل Intersection از منوی Choose process بخش WPS request builder

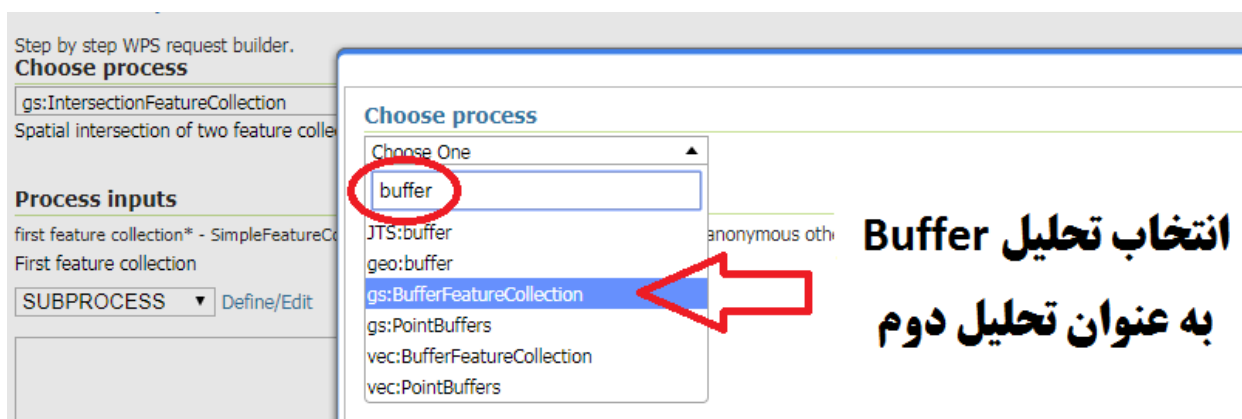
از قسمت First feature collection به صورت پیش فرض گزینه VECTOR_LAYER انتخاب شده است که با توجه به توضیحات قسمت قبل می توان از چهار حالت موجود انتخاب شود. در این قسمت حالت SubProcess را توضیح خواهیم داد. در این حالت کاربر می تواند به عنوان ورودی تحلیل جدید، خروجی یک تحلیل دیگر را قرار دهد. به عنوان مثال برای ورودی تقاطع، ابتدا می توان بر روی یک لایه حریم ایجاد نموده و خروجی را با یک لایه دیگر تقاطع دهد.



شکل ۳۷- انتخاب حالت SubProcess به عنوان ورودی حالت اول در تحلیل Intersection



شکل ۳۸- انتخاب گزینه Define/Edit جهت معرفی پارامترهای تحلیل SubProcess



شکل ۳۹- انتخاب تحلیل مورد نظر به عنوان تحلیل دوم



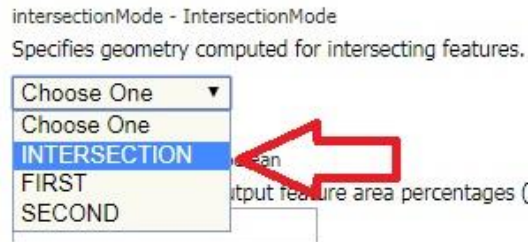
شکل ۴۰- انتخاب لایه ورودی از تحلیل Buffer و تعیین مقدار فاصله

پس از انجام تنظیمات فوق، بر روی دکمه Apply کلیک نموده و سپس از قسمت Second feature collection لایه دوم را انتخاب می‌نماییم.



شکل ۴۱- مشاهده کد XML حاصل از تحلیل Buffer و انتخاب لایه دوم

در مرحله بعد intersectionMode را روی گزینه INTERSECTION قرار می‌دهیم.



شکل ۴۲- انتخاب گزینه مربوطه به intersectionMode

در قسمت Process outputs و از Generate نوع خروجی فایل بر اساس یک فایل متنی قابل انتخاب است. با انتخاب گزینه Execute process تحلیل فوق اجرا و خروجی در صفحه نمایش قابل مشاهده است.



شکل ۴۳- انتخاب فرمت استاندارد مورد نظر به عنوان خروجی تحلیل Intersection

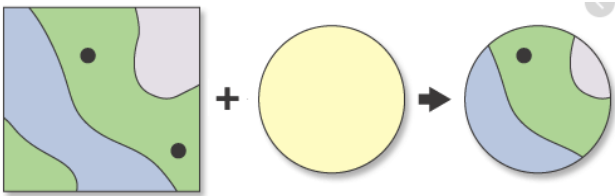
همچنین برای ایجاد درخواست بر اساس یک فایل XML می‌توان از گزینه Generate XML from process inputs/outputs استفاده کرد.



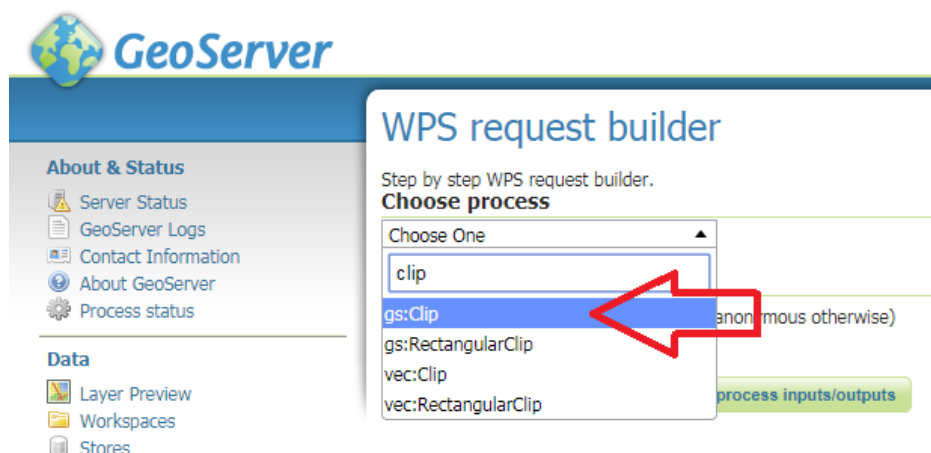
شکل ۴۴- دریافت خروجی تحلیل Intersection در زبان XML

۳-۴- تحلیل Clip

جدول ۱۱- تحلیل Clip

نام تحلیل	نام تابع	توضیحات
Clip	gs:Clip	برش زدن عوارض داخل محدوده موردنظر مثال: برش زدن محدوده ای خاص از شهر تهران  INPUT CLIP FEATURE OUTPUT

جهت اجرای تحلیل Clip در نرم افزار ژئوسرور، مجدداً از بخش Demos وارد صفحه WPS request builder شده و از قسمت Choose process، کلمه Clip را جستجو و گزینه gs:Clip را انتخاب می کنیم.



شکل ۴۵- انتخاب تحلیل Clip از منوی Choose process

از قسمت Input feature collection به صورت پیش فرض گزینه VECTOR_LAYER انتخاب شده است، لایه مورد نظر را معرفی و سپس از قسمت Clip Geometry گزینه Reference را انتخاب می نماییم.

Process inputs

features* - SimpleFeatureCollection

Input feature collection

VECTOR_LAYER sf:bugsites

انتخاب لایه مورد نظر

clip* - Geometry

Geometry to use for clipping (in same CRS as input features)

TEXT text/xml; subtype=gml/3.1.1

TEXT
REFERENCE
SUBPROCESS

انتخاب حالت Reference

شکل ۴۶- انتخاب لایه ورودی و حالت Clip Geometry از تحلیل Clip

در ادامه پس از انتخاب گزینه پیش فرض GET به عنوان روش ارسال درخواست URL و مشخص کردن Mime type مورد نظر، آدرس URL لایه دوم را به عنوان Clip feature وارد می نماییم.

clip* - Geometry
Geometry to use for clipping (in srs) REFERENCE
Method GET
Mime type application/json
URL http://localhost:8080/geo20/sf/ows?service=WFS&version=1.0.0&request=GetFeature&typeN

وارد کردن URL لایه Clip Feature

شکل ۴۷- انجام تنظیمات مورد نیاز Clip Geometry مربوط به تحلیل Clip

در قسمت Process outputs و از Generate نوع خروجی فایل بر اساس یک فایل متنی قابل انتخاب است. با انتخاب گزینه Execute process تحلیل فوق اجرا می گردد.

Process outputs
result* - SimpleFeatureCollection
Buffered feature collection
☒ Generate application/json
Authentication
☐ Authenticate (will run the request as anonymous otherwise)
Execute process Generate XML from process inputs/outputs

شکل ۴۸- انتخاب فرمت استاندارد مورد نظر به عنوان خروجی تحلیل Clip

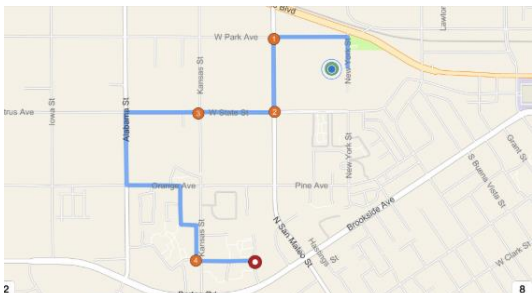
همچنین برای ایجاد درخواست بر اساس یک فایل XML می توان از گزینه Generate XML from process inputs/outputs استفاده کرد.



شکل ۴۹- ایجاد فایل XML به عنوان خروجی تحلیل Clip

۴-۴- تحلیل Nearest

جدول ۱۲- تحلیل Nearest

نام تحلیل	نام تابع	توضیحات
Nearest	gs:Nearest	نزدیک ترین عارضه به نقطه موردنظر مثال: نزدیک ترین مسیر به سازمان نقشه برداری 

جهت اجرای تحلیل Nearest در نرم افزار ژئوسرور، مجدداً از بخش Demos وارد صفحه WPS request builder شده و از قسمت Choose process، کلمه Nearest را جستجو و گزینه gs:Nearest را انتخاب می کنیم.

از قسمت Input feature collection که به صورت پیش فرض گزینه VECTOR_LAYER انتخاب شده است، لایه مورد نظر را انتخاب می نماییم.

WPS request builder

Step by step WPS request builder.

Choose process

gs:Nearest
Returns the feature in a given feature collection that is nearest to the specified point.

Process inputs

features* - FeatureCollection

Input feature collection

VECTOR_LAYER
tiger:poi

شکل ۵۰- انتخاب تحلیل Nearest و لایه مورد نظر

سپس از قسمت Point گزینه TEXT، سپس گزینه application/wkt را انتخاب نموده و مختصات نقطه مورد نظر را وارد می‌نماییم. به منظور معرفی سیستم مختصات و سیستم تصویر لایه ورودی رو گزینه Find کلیک می‌کنیم.

point* - Point

Point from which to compute distance

TEXT
application/wkt

POINT(-74.005 40.71)

وارد کردن مختصات نقطه‌ای مورد نظر

crs - CoordinateReferenceSystem

Coordinate reference system of the collection and point (default is the input collection CRS)

Find...

معرفی سیستم مختصات و سیستم تصویر مورد نظر

شکل ۵۱- انجام تنظیمات مورد نظر جهت وارد کردن مختصات نقطه‌ای و سیستم مختصات لایه مربوطه

در نهایت کد EPSG مربوط به سیستم مختصات نقطه وارد شده را نیز مشخص می‌کنیم.

Select a coordinate system. Use the search box to narrow the list

4326

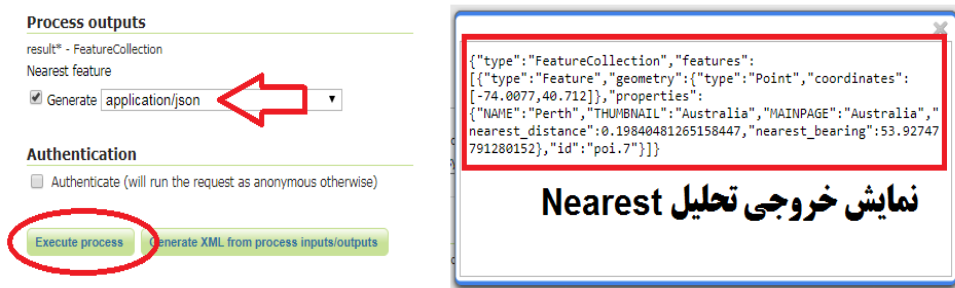
code	description
4326	WGS 84

Results 1 to 1 (out of 1 matches)

وارد کردن کد EPSG مرتبط با سیستم مختصات

شکل ۵۲- معرفی کد EPSG سیستم مختصات و سیستم تصویر لایه مورد نظر

در قسمت Generate از بخش Process outputs گزینه application/json را جهت نحوه نمایش خروجی انتخاب می‌نماییم و جهت اجرای تحلیل Nearest بر روی گزینه Execute process کلیک می‌کنیم.



شکل ۵۳- خروجی حاصل از تحلیل Nearest

۴-۵- تحلیل Within

جدول ۱۳- تحلیل Within

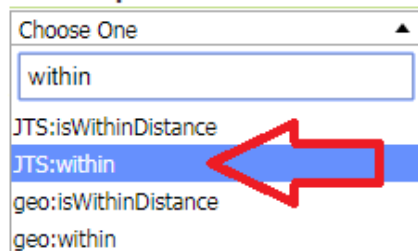
نام تحلیل	نام تابع	توضیحات
Within	JTS:within	عوارضی(نقطه، خط، سطح) که درون محدوده‌ای خاص قرار دارند مثال: چاه‌های آبی که درون مناطق مسکونی قرار گرفته‌اند

جهت اجرای تحلیل Within در نرم‌افزار ژئوسرور، مجدداً از بخش Demos وارد صفحه WPS request builder شده و از قسمت Choose process، کلمه Within را جستجو و گزینه JTS:Within را انتخاب می‌کنیم.

WPS request builder

Step by step WPS request builder.

Choose process



انتخاب تحلیل
within

شکل ۵۴- انتخاب تحلیل JTS:within

در این تحلیل برای معرفی لایه اول از قسمت Process inputs بخش First input geometry گزینه‌های پیش فرض TEXT و text/xml; subtype=gml/3.1.1 را انتخاب می‌نماییم.

Process inputs

a* - Geometry

First input geometry

TEXT text/xml; subtype=gml/3.1.1

```
<wfs:FeatureCollection xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:sf="http://www.openplans.org/spearfish"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:gml="http://www.opengis.net/gml"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" numberOfFeatures="0"
timeStamp="2022-10-30T06:34:56.754Z"
xsi:schemaLocation="http://www.opengis.net/wfs
http://localhost:8080/geoserver/schemas/wfs/1.1.0/wfs.xsd
http://www.openplans.org/spearfish http://localhost:8080/geoserver/sf/wfs?
service=WFS&version=1.0.0&request=DescribeFeatureType&typeName=sf%3Abugsites">
<gml:featureMembers>
<sf:bugsites gml:id="bugsites.1">
<sf:the_geom>
<gml:Point srsName="http://www.opengis.net/gml/srs/epsg.xml#26713"
srsDimension="2">
```

شکل ۵۵- معرفی لایه اول به صورت TEXT در قالب gml3.1

سپس از قسمت Second input geometry گزینه REFERENCE را انتخاب می‌نماییم و بعد از تعیین حالت ورودی یا Mime type بر اساس text/xml;subtype=gml/3.1.1، آدرس URL لایه دوم را وارد می‌نماییم.

b* - Geometry
Second input geometry

REFERENCE

Method GET

Mime type text/xml; subtype=gml/3.1.1

URL http://localhost:8080/geoserver/sf/ows?service=WFS&version=1.0.0&request=GetFeature&ty

تعیین ورودی اطلاعات

بر اساس GML3.1

**وارد کردن
URL لایه دوم**

شکل ۵۶- تعیین نوع ورودی و معرفی آدرس URL لایه دوم

در نهایت پس از انتخاب دکمه Generate XML from process inputs/outputs فایل XML تحلیل within به صورت

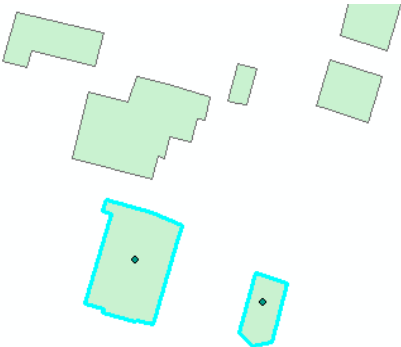
زیر قابل دسترسی خواهد بود.



شکل ۵۷- فایل XML حاصل از تحلیل within

۴-۶- تحلیل Contains

جدول ۱۴- تحلیل Contains

نام تحلیل	نام تابع	توضیحات
Contains	JTS:contains	محدوده‌هایی که شامل عوارضی خاص می‌باشند مثال: مناطقی که شامل چاه آب می‌باشند 

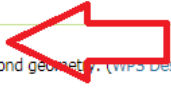
جهت اجرای تحلیل Contains در نرم‌افزار ژئوسرور، مجدداً از بخش Demos وارد صفحه WPS request builder شده و از قسمت Choose process، کلمه Contains را جستجو و گزینه JTS:Within را انتخاب می‌کنیم.

از قسمت First input geometry گزینه REFERENCE را انتخاب و URL لایه اول را وارد می‌نماییم. سپس از قسمت Second input geometry مجدداً گزینه REFERENCE را انتخاب و URL لایه دوم را وارد می‌نماییم.

WPS request builder

Step by step WPS request builder.

Choose process

JTS:contains  **انتخاب تحلیل**

Tests if the first geometry is contained in the second geometry. (WFS Descri...

Process inputs

a* - Geometry

First input geometry

REFERENCE  **وارد کردن URL لایه اول**

Method GET

Mime type application/json

URL <http://localhost:8080/geoserver/sf/ows?service=WFS&version=1.0.0&request=GetFeature&ty>

b* - Geometry

Second input geometry

REFERENCE  **وارد کردن URL لایه دوم**

Method GET

Mime type application/json

URL <http://localhost:8080/geoserver/sf/ows?service=WFS&version=1.0.0&request=GetFeature&ty>

Process outputs

result* - boolean

True if the first input is within the second input

☒ Generate

شکل ۵۸- انتخاب تحلیل Contains و انجام تنظیمات مورد نظر

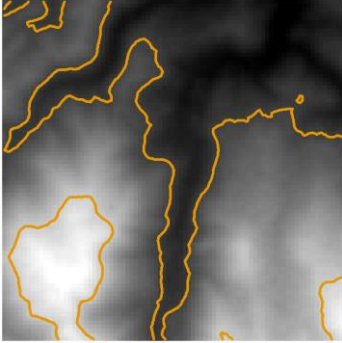
در نهایت پس از انتخاب دکمه Generate XML from process inputs/outputs فایل XML تحلیل contains به صورت زیر قابل دسترسی خواهد بود.

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WFS"
  <ows:Identifier>JTS:contains</ows:Identifier>
  <wps>DataInputs>
    <wps:Input>
      <ows:Identifier>a</ows:Identifier>
      <wps:Reference mimeType="application/json" xlink:href="http://localhost:
    </wps:Input>
    <wps:Input>
      <ows:Identifier>b</ows:Identifier>
      <wps:Reference mimeType="application/json" xlink:href="http://localhost:
    </wps:Input>
  </wps>DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput>
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>
```

شکل ۵۹- فایل XML حاصل از تحلیل Contains

۴-۷- تحلیل Contour

جدول ۱۵- تحلیل Contour

نام تحلیل	نام تابع	توضیحات
Contour	ras:contour	بدست آوردن منحنی های میزان با استفاده از فایل DEM 

با استفاده از این تحلیل می توان با معرفی فایل DEM به نرم افزار ژئوسرور و انجام تنظیمات مربوطه، منحنی میزان هایی با فاصله تعیین شده بدست آورد و از آن ها خروجی گرفت.

جهت اجرای تحلیل Contour در نرم افزار ژئوسرور، مجدداً از بخش Demos وارد صفحه WPS request builder شده و از قسمت Choose process، کلمه Contour را جستجو و گزینه ras:Contour را انتخاب می کنیم.

از قسمت Input raster لایه مورد نظر خود را انتخاب و سپس از قسمت Interval فاصله منحنی میزان را وارد می نماییم.

WPS request builder

Step by step WPS request builder.

Choose process

ras:Contour

Computes contour lines at specified intervals or levels from the values in a raster. (WPS DescribeProcess)

انتخاب تحلیل

Process inputs

data* - GridCoverage2D

Input raster

RASTER_LAYER test:sdem_clip

انتخاب لایه
مورد نظر

band - Integer

Band number (zero base) to use for values to be contoured

levels - double(0-2147483647)

Values of levels at which to generate contours

interval - Double

Interval between contour values (ft)

200

انتخاب فاصله
منحنی میزان

simplify - Boolean

Indicates whether contour lines are simplified

smooth - Boolean

Indicates whether contour lines are smoothed using Bezier smoothing

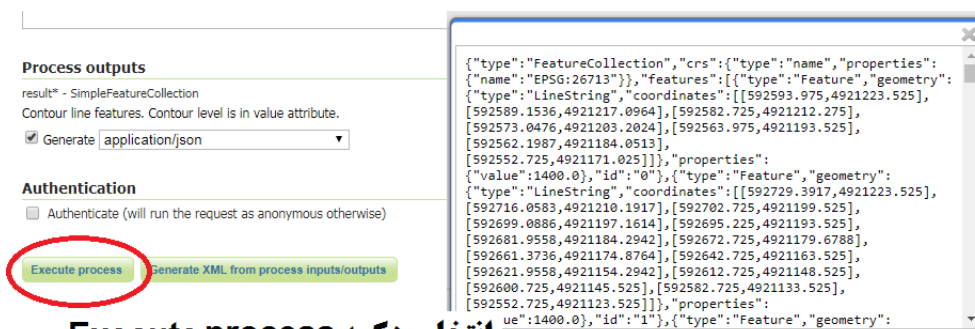
roi - Geometry

Geometry delineating the region of interest (in raster coordinate system)

TEXT text/xml; subtype=gml/3.1.1

شکل ۶۰- انتخاب تحلیل Contour و انجام تنظیمات مورد نظر

به منظور اجرای تحلیل Contour مطابق شکل ۶۱ عمل می کنیم.



انتخاب دکمه Execute process

جهت اجرای تحلیل

شکل ۶۱- انتخاب دکمه Execute process اجرای تحلیل Contour و نمایش خروجی حاصل از آن

در نهایت پس از انتخاب دکمه Generate XML from process inputs/outputs فایل XML تحلیل contour به صورت زیر

قابل دسترسی خواهد بود.



شکل ۶۲- فایل XML حاصل از تحلیل Contains

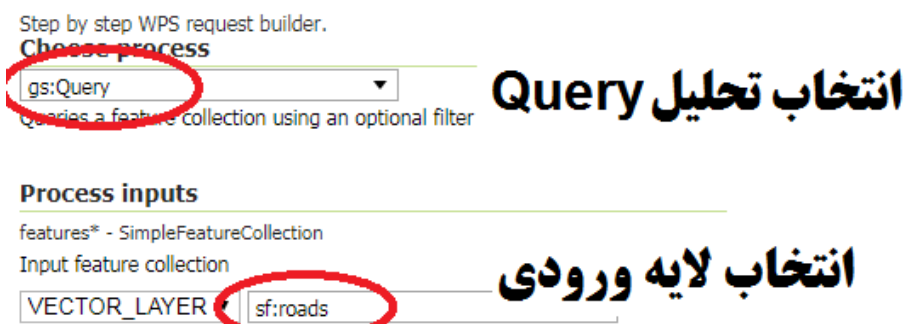
۴-۸- تحلیل Query

جدول ۱۶- تحلیل Query

نام تحلیل	نام تابع	توضیحات
Query	gs:Query	انتخاب عوارض مورد نظر از سرویس موجود مثال: مناطقی که شامل چاه آب می باشند

جهت اجرای تحلیل Query در نرم افزار ژئوسرور، مجدداً از بخش Demos وارد صفحه WPS request builder شده و از قسمت Choose process ، کلمه Query را جستجو و گزینه gs:Query را انتخاب می کنیم.

سپس از قسمت Input feature collection و گزینه پیش فرض VECTOR_LAYER لایه مورد نظر را به عنوان لایه ورودی انتخاب می نماییم.



شکل ۶۳- انتخاب تحلیل gs:Query و تعیین لایه ورودی

در مرحله بعد گزینه text/plain; subtype=cql را انتخاب نموده و کوئری مورد نظر را وارد می نماییم. همچنین می توان از عملگرهای منطقی AND و OR نیز در کوئری ها استفاده کرد.

filter - Filter

The filter to apply

TEXT text/plain; subtype=cql

انتخاب گزینه cql

cat = 4

وارد کردن کوئری مورد نظر

شکل ۶۴- وارد کردن کوئری مورد نظر

در نهایت به منظور اجرای تحلیل فوق گزینه Execute process را انتخاب می‌نماییم. خروجی این تحلیل به صورت زیر

است.

```
{
  "type": "FeatureCollection",
  "crs": {
    "type": "name",
    "properties": {
      "name": "EPSG:26713"
    }
  },
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "MultiLineString",
        "coordinates": [
          [
            [597332.327, 4915556.394], [597316.4589, 4915535.0197], [597289.7333, 4915523.9882],
            [597248.4048, 4915517.8066], [597113.9142, 4915571.8185], [597053.5918, 4915628.4181],
            [597023.1884, 4915630.1871], [596849.5359, 4915531.0368], [596802.7156, 4915533.3828],
            [596785.7347, 4915515.052], [596791.2691, 4915488.2277], [596832.7838, 4915414.5133],
            [596879.7619, 4915344.4692], [596865.2215, 4915322.4839], [596842.2106, 4915282.7947],
            [596809.6411, 4915170.5088], [596832.2587, 4915118.103], [596889.5597, 4915053.5686],
            [596909.67, 4915033.4823], [596934.1314, 4914972.5416], [596932.4041, 4914931.0653],
            [596947.0168, 4914904.869], [597032.9742, 4914822.7049], [597042.4098, 4914805.3975]]
          ]
        ],
        "properties": {
          "cat": 4,
          "label": "light-duty road, improved surface",
          "id": "roads.7"
        }
      },
      "type": "Feature",
      "geometry": {
        "type": "MultiLineString",
        "coordinates": [
          [
            [595108.335, 4915501.9404], [595083.4234, 4915495.1817], [594990.422, 4915486.4573],
            [594965.5047, 4915482.1382], [594904.0388, 4915507.6309], [594860.9133, 4915489.8577],
            [594729.1362, 4915423.1157], [594708.5174, 4915400.5083], [594604.6932, 4915339.9211],
            [594518.634, 4915222.0392], [594478.6309, 4915168.8983], [594430.0961, 4915124.2789],
            [594365.7964, 4915061.3312], [594323.3457, 4915014.8942], [594325.8899, 4914966.7176],
            [594351.4649, 4914949.6916], [594367.3137, 4914932.0363], [594421.5392, 4914882.7432],
            [594435.0172, 4914838.8577], [594432.6506, 4914810.7977], [594424.2185, 4914770.0207],
            [594405.4208, 4914755.2428], [594299.8409, 4914665.3771]]
          ]
        ],
        "properties": {
          "cat": 4,
          "label": "light-duty road, improved surface",
          "id": "roads.11"
        }
      },
      "type": "Feature",
      "geometry": {
        "type": "MultiLineString",
        "coordinates": [
          [
            [594401.4948, 4917077.5127], [594391.4337, 4917104.5325],
            [594374.3377, 4917135.603], [594231.1578, 4917265.8344], [594163.5481, 4917318.7602],
            [594145.8342, 4917354.0987], [594124.3388, 4917446.7599], [594089.4876, 4917530.8558],
            [594041.4948, 4917077.5127]
          ]
        ],
        "properties": {
          "cat": 4,
          "label": "light-duty road, improved surface",
          "id": "roads.12"
        }
      }
    ]
  ]
}
```

شکل ۶۵- خروجی حاصل از تحلیل Query

فصل پنجم

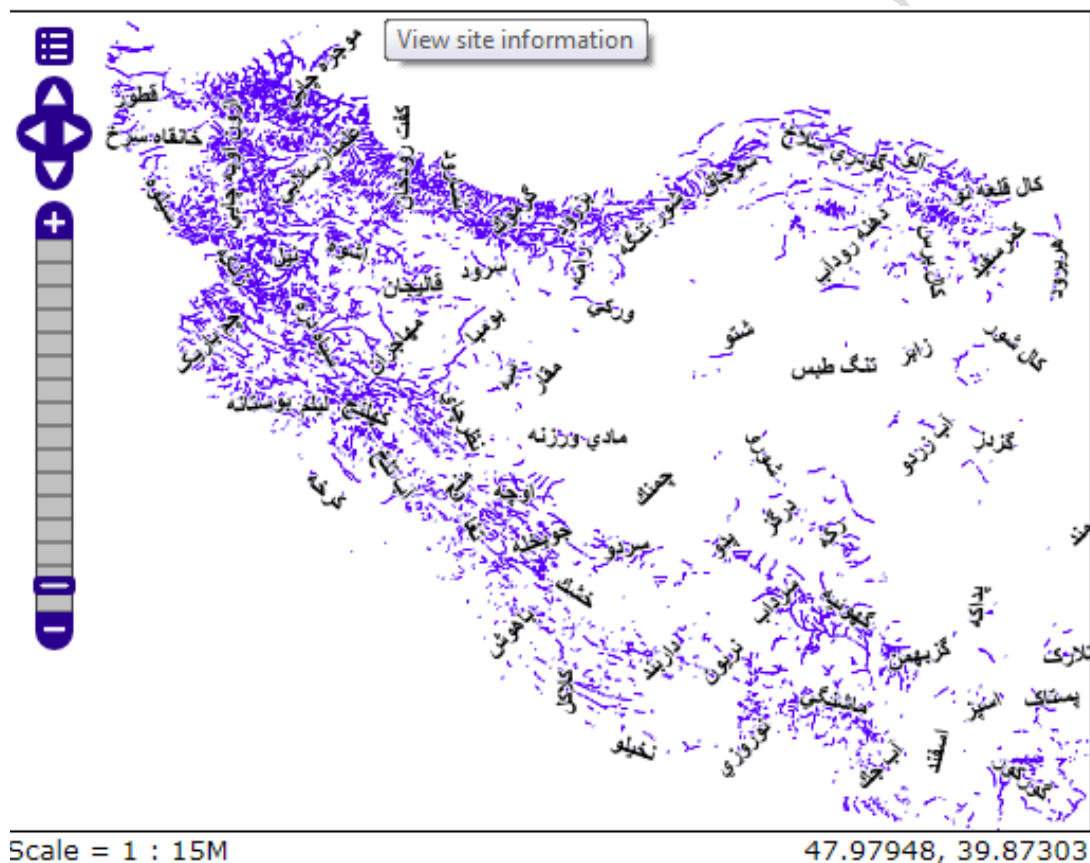
اجرای یک پروژه تحلیلی مکانی در
ژئوسرور

۵-۱- اجرای یک پروژه موردی

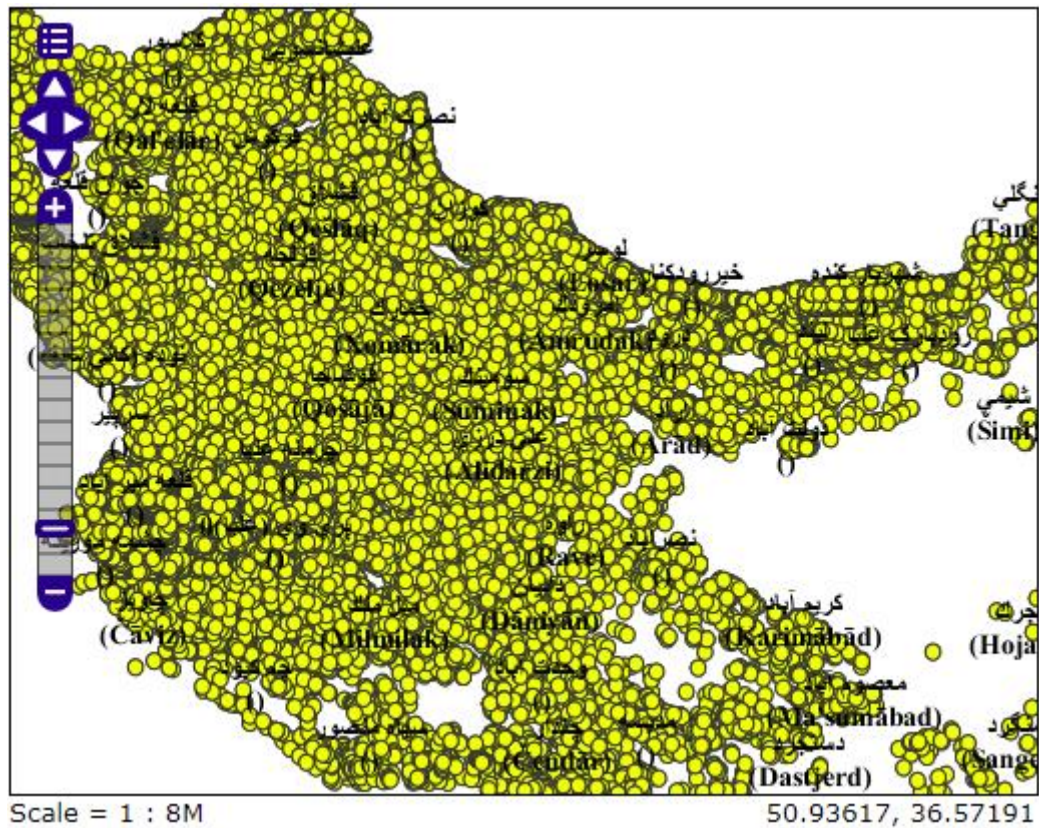
در این پروژه موردی، با استفاده از اطلاعات رودخانه‌های استان تهران، با استفاده از سرویس پردازشی پرسشی نام یک رود وارد شده و سپس بر اساس حریم ۱۰۰۰ متری بر روی این رودخانه روستاهایی که در برابر سیل در خطر واقع هستند نمایش داده می‌شود.

هدف: انتخاب روستاهایی که در حریم ۱۰۰۰ متری رودخانه خاص هستند.

فرضیات: سرویس لایه‌های رودخانه‌ها، روستاها موجود هستند.



شکل ۶۶- سرویس WMS لایه رودخانه‌ها



شکل ۶۷- سرویس WMS لایه روستاها

در این پروژه ابتدا یک رودخانه انتخاب می‌شود.. برای انتخاب رودخانه از سرویس Query استفاده می‌شود. به عنوان مثال رودخانه جاجرود را کاربر انتخاب می‌نماید. در سرویس لایه موجود هر رودخانه دارای یک فیلدی به نام mslink است که شماره‌های ۲۵۰۵۰۰۱ و ۳۶۱۶۰۰۳ مربوط به رودخانه جاجرود می باشد.

Step by step WPS request builder.

Choose process

vec:Query

Queries a feature collection using an optional filter and an optional list of attributes to include. Can also be used to convert feature col

Process inputs

features* - SimpleFeatureCollection

Input feature collection

VECTOR_LAYER test:select_river3

attribute - String(0-2147483647)

Attribute to include in output

filter - Filter

The filter to apply

TEXT text/plain; subtype=cql

mslink =3616003.0 or mslink =2505001.0

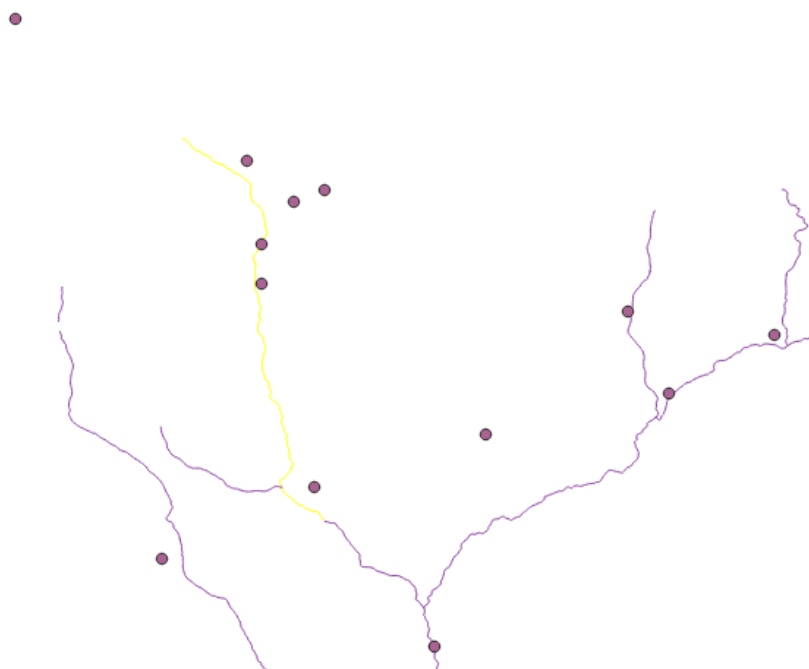
Process outputs

result* - SimpleFeatureCollection

The filtered feature collection

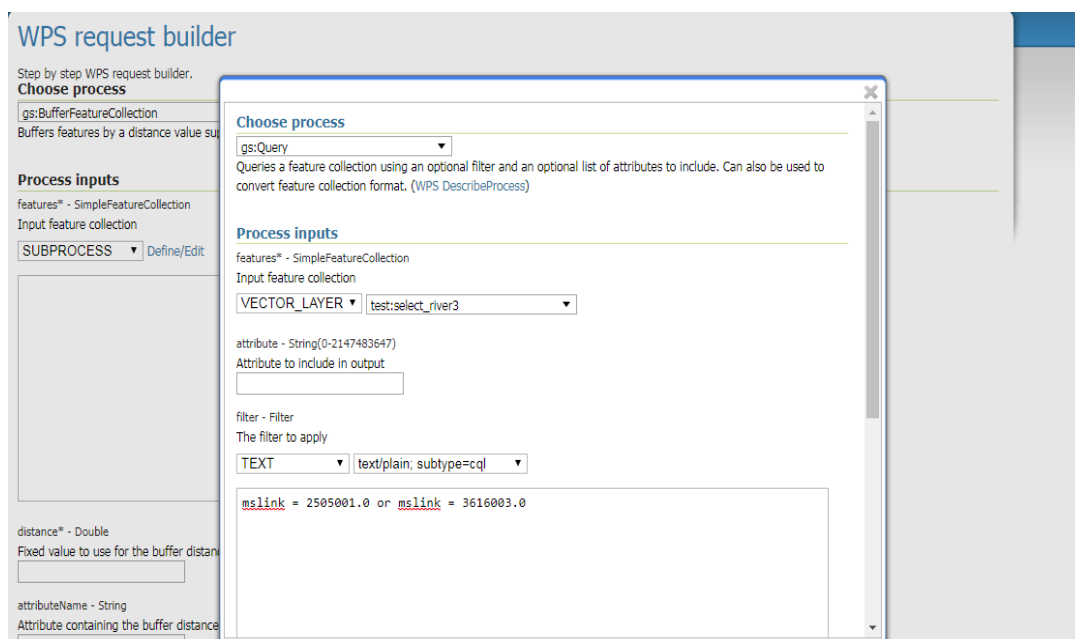
☒ Generate application/json

شکل ۶۸- انتخاب تحلیل Query و وارد نمودن کوئری مورد نظر



شکل ۶۹- نمایش رودخانه جاجرود

پس از انتخاب عارضه مورد نظر باید بر روی آن بافر زده شود تا حریم یک کیلومتری آن انتخاب گردد. برای انجام این کار در قسمت Choose Process گزینه BufferfeatureCollection را انتخاب می‌کنیم. سپس در قسمت input feature collection گزینه subprocess را انتخاب می‌شود. در این قسمت مجدد گزینه query را انتخاب کرده و بر روی لایه رودخانه جابروود را انتخاب می‌کنیم.



شکل ۷۰- انتخاب تحلیل Query به عنوان ورودی تحلیل Buffer

سپس پردازش بافر را به صورت شکل زیر تعریف می‌شود.

WPS request builder

Step by step WPS request builder.

Choose process

gs:BufferFeatureCollection

Buffers features by a distance value supplied either as a parameter or by a feature attribute. Calculates buffers based on Cartesian distances. (WPS DescribeProcess)

Process inputs

features* - SimpleFeatureCollection

Input feature collection

SUBPROCESS Define/Edit

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0"
service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier/>
  <wps>DataInputs/>
  <wps:ResponseForm/>
</wps:Execute>
```

distance* - Double

Fixed value to use for the buffer distance

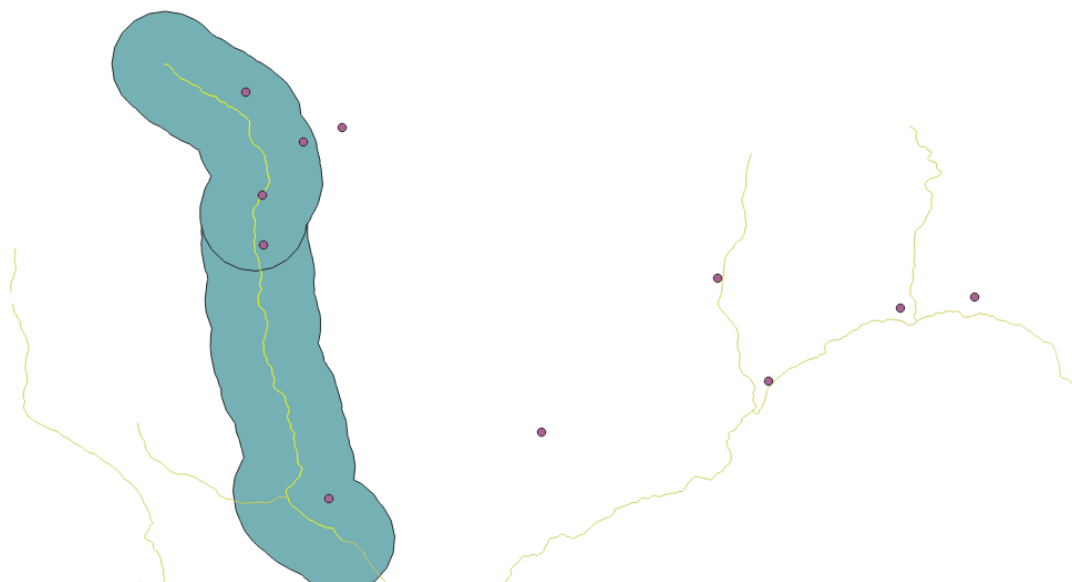
1000

attributeName - String

Attribute containing the buffer distance value

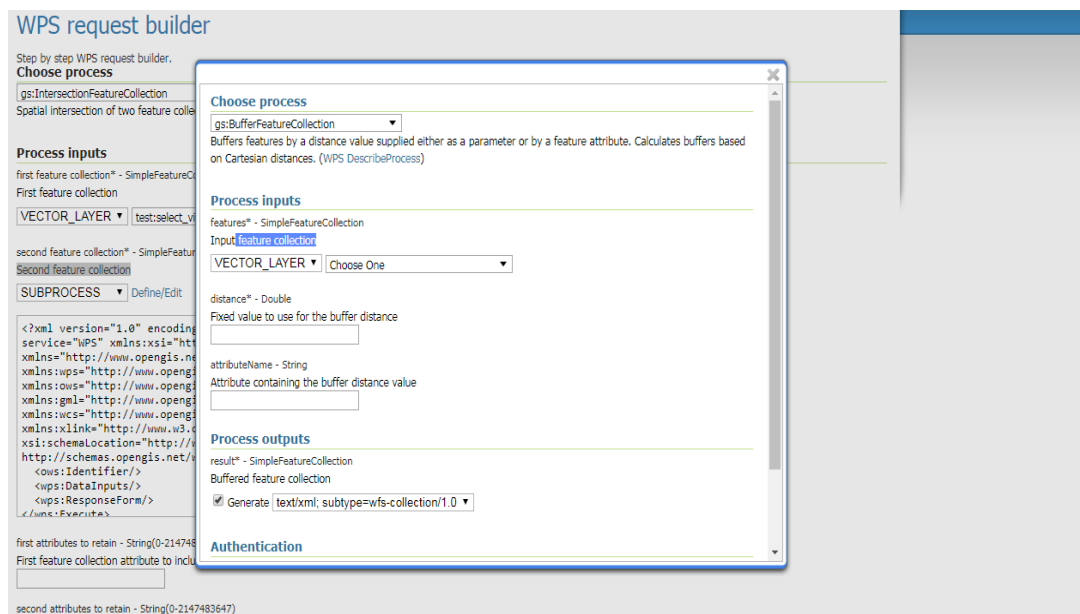
شکل ۷۱- فایل XML حاصل از تحلیل Query به عنوان ورودی تحلیل Buffer

خروجی این تحلیل در شکل ۷۲ نشان داده شده است.



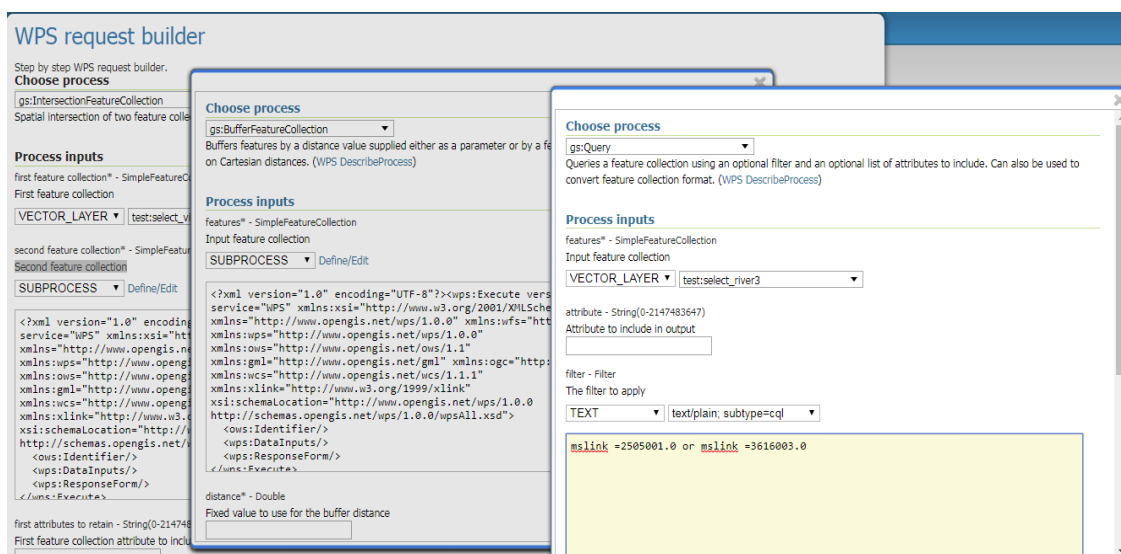
شکل ۷۲- سرویس نمایشی حاصل از تحلیل Buffer بر روی رودخانه جاجرود

سپس از ایجاد لایه حریم رودخانه مورد نظر، حال باید روستاهایی که در این منطقه قرار داده شده‌اند انتخاب گردند. بدین منظور باید از تحلیل intersection استفاده شود. و در آن روستاهایی که با این عارضه تقاطع دارند را مشخص گردد. برای این منظور در قسمت WPS request builder و در قسمت Choose process گزینه gs: IntersectionFeatureCollection انتخاب می‌گردد. این تابع دارای دو ورودی است که ورودی اول آن لایه روستاها است و ورودی دوم آن نتیجه خروجی حریم رودخانه جاجرود است. بدین منظور در قسمت اول لایه village انتخاب شده و در قسمت دوم خروجی تحلیل حریم قرار می‌گیرد. به همین منظور از قسمت Second feature collection گزینه subprocess را انتخاب نموده و در پنجره باز شده تحلیل bufferfeaturecollection را انتخاب نمود.



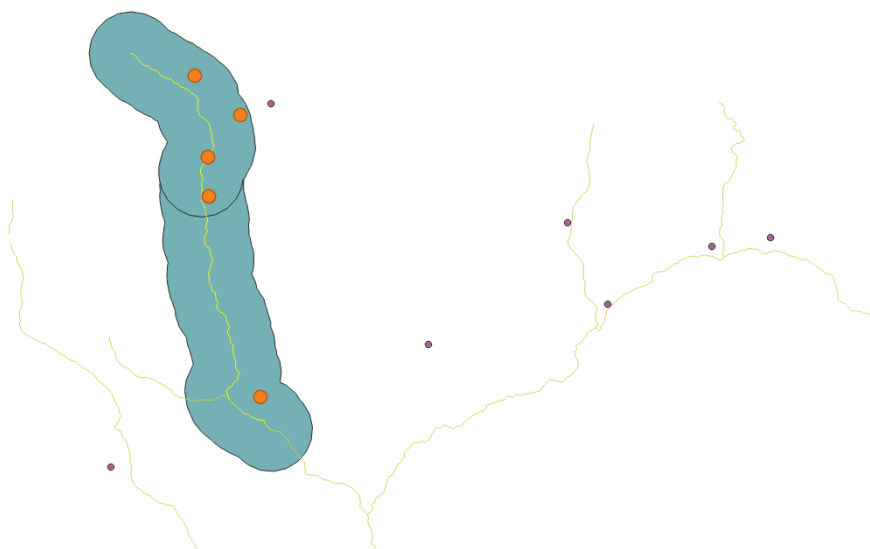
شکل ۷۳- انتخاب تحلیل Buffer به عنوان ورودی لایه دوم

حال در این قسمت برای ایجاد حریم برای عارضه رودخانه جابرو در این قسمت از subprocess را انتخاب نمود و در این قسمت از تحلیل query استفاده نموده و رودخانه جابرو انتخاب می‌شود.



شکل ۷۴- وارد کردن کوئری مورد نظر به منظور انتخاب رودخانه جابرو

خروجی این تحلیل در شکل ۷۵ نمایش داده شده است. از این تحلیل می‌توان برای مشخص نمودن روستاهای در معرض خط سیل استفاده نمود.



شکل ۷۵- خروجی حاصل از تحلیل فوق و نمایش روستاهای در معرض خطر سیل

لازم به ذکر است که این تحلیل‌ها می‌تواند به این صورت ایجاد شود که هریک از لایه‌های رودخانه و نقاط روستایی هریک از یک منبع دریافت شده باشند و خروجی آن به صورت سرویس در اختیار کاربران قرار گیرد.

خروجی XML مورد نظر برای اجرای این تحلیل که به صورت زنجیره وار تعریف شده است در شکل ۷۶ آورده شده

است.

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>gs:IntersectionFeatureCollection</ows:Identifier>
  <wps>DataInputs>
    <wps:Input>
      <ows:Identifier>first feature collection</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2" xmlns:test="localhost:8080/test">
            <wfs:Query typeName="test:select_village3"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>second feature collection</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wps" method="POST">
        <wps:Body>
          <?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
          <ows:Identifier>gs:BufferFeatureCollection</ows:Identifier>
          <wps>DataInputs>
            <wps:Input>
```

```

<ows:Identifier>features</ows:Identifier>
<wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wps" method="POST">
  <wps:Body>
    <wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
      xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
      xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
      xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
      xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
      <ows:Identifier>vec:Query</ows:Identifier>
      <wps:DataInputs>
        <wps:Input>
          <ows:Identifier>features</ows:Identifier>
          <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
            <wps:Body>
              <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2" xmlns:test="localhost:8080/test">
                <wfs:Query typeName="test:select_river3"/>
              </wfs:GetFeature>
            </wps:Body>
          </wps:Reference>
        </wps:Input>
        <wps:Input>
          <ows:Identifier>filter</ows:Identifier>
          <wps>Data>
            <wps:ComplexData mimeType="text/plain; subtype=cql"><![CDATA[mslink =3616003.0 or mslink
=2505001.0]]></wps:ComplexData>
          </wps>Data>
        </wps:Input>
      </wps>DataInputs>
      <wps:ResponseForm>
        <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
          <ows:Identifier>result</ows:Identifier>
        </wps:RawDataOutput>
      </wps:ResponseForm>
    </wps:Execute>
    </wps:Body>
  </wps:Reference>
</wps:Input>
<wps:Input>
  <ows:Identifier>distance</ows:Identifier>
  <wps>Data>
    <wps:LiteralData>1000</wps:LiteralData>
  </wps>Data>
</wps:Input>
</wps>DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
  </wps:Body>
</wps:Reference>
</wps:Input>
</wps>DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

شکل ۷۶- خروجی XML تحلیل Chain

فصل ششم

پیاده سازی استاندارد WPS بر مبنای
PyWPS

۶-۱- فرآیند (process) در استاندارد OGC

فرایند محاسباتی p تابعی است که به ازای هر ورودی یک خروجی متناظر را برمی گرداند.

$$p: X \rightarrow Y$$

در حقیقت، یک فرایند محاسباتی شامل برخی عملیات مکانی است به طوریکه می تواند یک عملیات ساده (جمع دو لایه رستری) و یا عملیات دارای پیچیدگی بالا (مدل تغییر وضعیت جوی) را شامل شود. فرایند چیزی است که شما به عنوان یک کاربر pywps در معرض مردم قرار داده و اجازه پردازش داده را به آنها می دهید.

هر فرایند دارای ویژگی های زیر می باشد:

- **Identifier** شناسه منحصر به فرد فرایند
- **Title** عنوانی خوانا برای انسان
- **Abstract** توصیفی بیشتر در مورد فرایند که چه کاری انجام میدهد و چطور قرار است مورد استفاده قرار گیرد.
- مجموعه ای از ورودی ها و خروجی ها (inputs and outputs)

۶-۲- ورودی و خروجی داده در استاندارد OGC

استاندارد OGC سه نوع ورودی و خروجی داده (data inputs and outputs) تعریف می کند که عبارتست از: LiteralData, ComplexData, BoundingBoxData

تمامی نوع داده ها می بایست دارای ویژگی های زیر باشند:

- **Identifier** شناسه منحصر به فرد فرایند
- **Title** عنوانی خوانا برای انسان
- **Abstract** توصیفی بیشتر در مورد ورودی ها و خروجی ها
- **minOccurs** حداقل وقوع ورودی
- **maxOccurs** حداکثر تعداد بارهایی که ورودی یا خروجی موجود می باشد

۶-۲-۱- LiteralData

هر رشته متنی غالباً کوتاه می باشد به طوریکه برای پارامترهای تکی نظیر اعداد و یا پارامترهای متنی به کار میرود. WPS اجازه تعریف مقادیر مجاز را به سرور می دهد. این تعریف در قالب لیست و یا فواصل مقادیر مجاز به همراه نوع داده (integer, float, string) می باشد. ویژگی های اضافی نظیر units و encoding نیز می تواند تعیین گردد.

۶-۲-۲-ComplexData

این نوع داده معمولاً به صورت فایل‌های رستری و یا برداری می‌باشد که می‌توان با تعیین mimeType, XML schema و یا encoding جزئیات بیشتری را مشخص کرد. به طور معمول حداقل نیازمندی برای شناسایی داده، ورودی mimeType می‌باشد. به طور معمول برای داده‌های برداری GML به صورت application/gml+xml و برای داده‌های رستری، به صورت image/tiff; subtype=geotiff می‌باشد.

۶-۲-۳-BoundingBoxData

این نوع داده در مشخصات OGC OWS به صورت دو جفت مختصات (برای فضای دو و سه بعدی) تعیین می‌گردد که می‌توانند به صورت WGS84 یا EPSG رمزگذاری شوند. در دنیای واقعی این نوع داده خیلی به صورت رایج استفاده نمی‌شود.

۶-۳-انتقال داده به فرآیند (passing data to process)

- به طور معمول سه روش به منظور انتقال داده ورودی از کلاینت به سرور وجود دارد:
- در این روش، داده روی سرور موجود باشد، که این روش ساده ترین حالت است.
 - در این روش، داده به همراه درخواست به سرور ارسال گردد. در این حالت داده که جزیی از XML می‌باشد توسط پروتکل HTTP POST فرستاده می‌شود و می‌تواند به صورت متنی (JSON و GML) و یا رستری باشد.
 - در این روش، ارسال اطلاعات به صورت reference link بوده و نیازی به انتقال داده نیست. این روش اگر چه بیشتر بر روی داده‌های از نوع ComplexData اعمال می‌شود، اما در داده‌های از نوع LiteralData و BoundingBoxData نیز می‌تواند مورد استفاده قرار گیرد.

۶-۴-درخواست فرآیند به صورت همزمان و غیرهمزمان

دو حالت برای اجرای نمونه فرآیند وجود دارد: همزمان و غیر همزمان (Synchronous vs asynchronous process request).

۶-۴-۱-حالت همزمان (Synchronous mode)

کلاینت درخواست Execute را به سرور ارسال نموده و در حالت ارتباط باز با سرور منتظر می‌ماند تا زمانی که فرآیند محاسبه شده و پاسخ نهایی بازگردانده شود. این روش در محاسبات سریع می‌تواند مفید واقع شود.

۶-۴-۲- حالت غیرهمزمان (Asynchronous mode)

در این حالت، کلاینت درخواست Execute را به همراه بیان صریح حالت غیر همزمان ارسال می نماید. در صورت پشتیبانی توسط فرآیند، سرور پاسخ ProcessAccepted را فوراً با URL می فرستد به طوریکه کلاینت می تواند به طور متناوب وضعیت فرآیند اجرا را بررسی کند.

۶-۵- وضعیت فرایند (process Status)

- وضعیت فرایند در رابطه با چگونگی پیشرفت محاسبات گزارش می دهد و به صورت یکی از چهار حالت زیر می باشد:
- ProcessAccepted : فرایند توسط سرور مورد پذیرش قرار گرفته و اجرای فرایند به زودی آغاز می گردد.
 - ProcessStarted : محاسبات فرایند آغاز شده و به صورت متنی گزارشی در رابطه با وضعیت محاسبات ارائه می دهد. (به طور مثال - "33% - Caculationg buffer")
 - ProcessFinished : نمونه فرایند به طور موفق به انجام محاسبات پرداخته و پاسخ نهایی به کلاینت بازگشت داده می شود و یا در مکان نهایی ذخیره می گردد.
 - ProcessFailed : در نمونه فرایند مشکلی وجود داشته که منجر به ایجاد exception توسط سرور می گردد. علاوه بر این پیامی در مورد اینکه چه چیزی ممکن است اشتباه باشد نمایش داده می شود.

۶-۶- کدگذاری درخواست HTTP GET و POST

این درخواست می تواند هم به صورت جفت کلید-مقدار (key-value pairs or KVP) و یا محتوای XML (XML) (payload) رمزگذاری گردد.

۶-۶-۱- جفت کلید-مقدار (Key-value pairs)

معمولاً توسط متد درخواست HTTP GET رمزگذاری شده به طور مستقیم در URL ارسال می شود. مقادیر و کلیدها توسط علامت مساوی (=) از همدیگر جدا می شوند. همچنین هر جفت از جفت دیگر نیز توسط علامت & از همدیگر جدا خواهند شد (از ابتدای درخواست با علامت ? می باشد). مثالی از درخواست Get Capabilities در زیر مشاهده می شود.

`http://127.0.0.1:8000/?request=GetCapabilities&service=WPS&version=1.0.0`

در مثال بالا سه جفت پارامتر ورودی مشاهده می گردد: request, service, و version به همراه مقادیر WPS، GetCapabilities و 1.0.0.

۶-۲-۶- محتوای XML (XML payload)

داده‌های XML از طریق درخواست HTTP POST ارسال میشوند. با افزودن پارامترهای بیشتر به سند XML، امکان پردازش ساختارهای پیچیده فراهم می‌شود. همچنین کلاینت میتواند کلیه داده‌ها (شامل داده‌های برداری مانند GML و داده‌های رستری کدگذاری شده با base64) را رمزگذاری کند.

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:GetCapabilities language="cz" service="WPS" 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/wpsGetCapabilities_request.xsd">
  <wps:AcceptVersions>
    <ows:Version>1.0.0</ows:Version>
  </wps:AcceptVersions>
</wps:GetCapabilities>
```

۶-۷- نصب و راه اندازی PyWPS

راحتترین روش به منظور نصب PyWPS استفاده از PIP (Python Package Index) می‌باشد (روش پیشنهادی).

```
pip install pywps
```

در این صورت آخرین نسخه PyWPS از PIP نصب می‌شود.

در صورتی که توسعه PyWPS مد نظر باشد، می‌توان آخرین نسخه آن را از repository موجود در ^۱github دریافت نموده و علاوه بر نصب نیازمندی‌های اجرا، به نصب نیازمندی‌های توسعه‌ای آن نیز پرداخت.

```
pip install -r requirements.txt
```

```
pip install -r requirements-dev.txt #For development tools
```

۶-۸- فرآیندها

Pywps از طریق فرایندها و سرویس‌ها فعالیت می‌کند. در حقیقت یک فرآیند، یک کلاس پایتون است که حاوی متد handler و لیستی از input ها و output ها می‌باشد. به‌طور پیش‌فرض، Pywps هیچ فرایند از پیش تعریف شده‌ای در خود ندارد و این کار بر عهده کاربر Pywps است.

۶-۸-۱- نوشتن یک فرایند (writing a process)

یک فرایند به‌صورت یک کلاس که از Process ارث بری می‌کند نوشته می‌شود. نمونه Process به‌منظور پیکربندی، نیاز به ویژگی‌های زیر دارد:

- Identifier شناسه منحصر به فرد فرایند

^۱<https://github.com/geopython/pywps>

- Title عنوان فرایند
- inputs لیستی از ورودی های فرایند
- outputs لیستی از خروجی های فرایند
- handler متدی است که object هایی از نوع request و response را دریافت نموده و به پردازش داده های ورودی، انجام آنالیز مکانی (با استفاده از کتابخانه هایی نظیر GDAL/OGR) و بازگرداندن خروجی می پردازد.

۶-۹- پیاده سازی نمونه اجرایی بر اساس PyWPS

هدف از پیاده سازی این نمونه اجرایی، ایجاد سرویس های پردازشی ساده از عملگرهای اصلی WPS یعنی GetCapabilities, DescribeProcess و Execute مبتنی بر وب می باشد.

در نمونه اول، کاربر یک ورودی متنی را به عنوان "name" وارد کرده و عبارت "Hello "name" بازگردانده شود به نحوی که به جای name، کلمه دریافت شده از کاربر درج شود.

در نمونه بعد، یک سرویس تحلیلی بافر به صورت تحت وب پیاده سازی شده است که در آن کاربر مختصات (x و y) و فاصله مورد نظر را وارد نموده و پس از اجرا، بافری در اطراف آن نقطه ایجاد شده است.

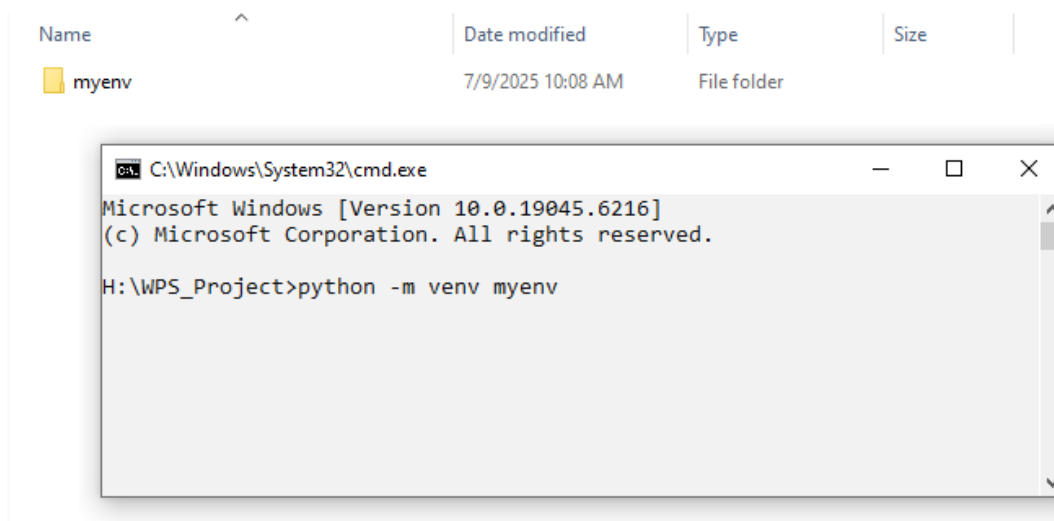
در این راستا از چارچوب PyWPS به منظور پیاده سازی استاندارد WPS مبتنی بر OGC و چارچوب Django برای ایجاد بستر تحت وب و تعامل با PyWPS استفاده خواهد شد و در مرحله بعد به ایجاد سرویس مبتنی بر وب برای تحلیل مکانی بافر اقدام می گردد.

Django یک چارچوب قدرتمند برای توسعه وب است که بر پایه زبان برنامه نویسی پایتون ساخته شده است. این چارچوب به دلیل ساختار ماژولار، امنیت بالا و قابلیت توسعه پذیری، یکی از محبوب ترین گزینه ها برای توسعه برنامه های تحت وب محسوب می شود. در این بین همانطور که اشاره شد PyWPS به عنوان یک پیاده سازی متن باز استاندارد WPS از OGC در زبان پایتون، به توسعه دهندگان اجازه می دهد تا پردازش های مکانی را به صورت سرویس های تحت وب ارائه دهند. ترکیب Django و PyWPS امکان ایجاد سامانه های قدرتمند پردازش مکانی را فراهم می کند که در آن مدیریت کاربران، امنیت، مسیریابی درخواست ها و تعامل با پایگاه داده به صورت منسجم و ساختاریافته انجام می شود. این رویکرد به ویژه برای پیاده سازی پردازش های مکانی در بسترهای سازمانی یا ژئوپورتال ها بسیار مفید و کاربردی می باشد.

هر بار که یک package در پایتون نصب می شود، در دسترس کل سیستم قرار می گیرد. از آنجایی که پروژه های مختلف نیازهای متفاوتی دارند، این امر می تواند منجر به ایجاد تعارضات شود. برای مقابله با این موضوع، پایتون از محیط های مجازی استفاده می کند که به صورت مجموعه های مستقل از یکدیگر فعالیت می کنند. استفاده از محیط مجازی هنگام نوشتن کد در پایتون بسیار مهم است تا از تعارضات package ها جلوگیری شود. بدین منظور در مرحله اول، در یک دایرکتوری دلخواه به ایجاد یک محیط مجازی (virtual environment) می پردازیم.

```
python -m venv myenv
```

پس از اجرای دستور فوق در cmd فولدري با نام 'myenv' در دایرکتوري مورد نظر ايجاد مي‌گردد.



شکل ۷۷- ايجاد محيط مجازی myenv

تذکر: نمونه‌اي اجرائي اين دستورالعمل با استفاده از پایتون نسخه 3.9.11 در سیستم عامل ویندوز پیاده‌سازی شده است.

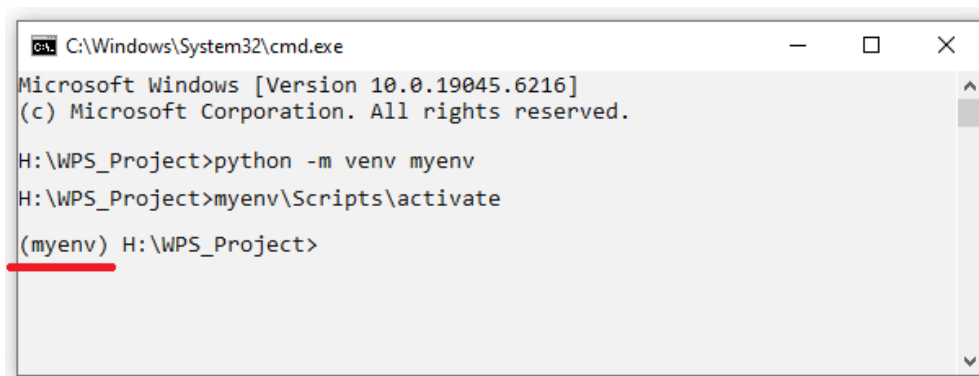
توجه: برای دانلود نسخه 3.9.11 پایتون و ساير فايل‌های مورد نیاز از لینک زیر استفاده نمایید.

<https://mega.nz/folder/ko8WGLSY#BuC1Xc4qBrRdMOTyWREtZg>

سپس با دستور زیر محیط مجازی ايجاد شده را فعال کنید.

myenv\Scripts\activate

در صورتي که مراحل به درستي انجام شده باشد، عبارت (myenv) در ابتدای خط فرمان مطابق شکل زیر نمایش داده خواهد شد که نشان‌دهنده فعال بودن محیط مجازی است.



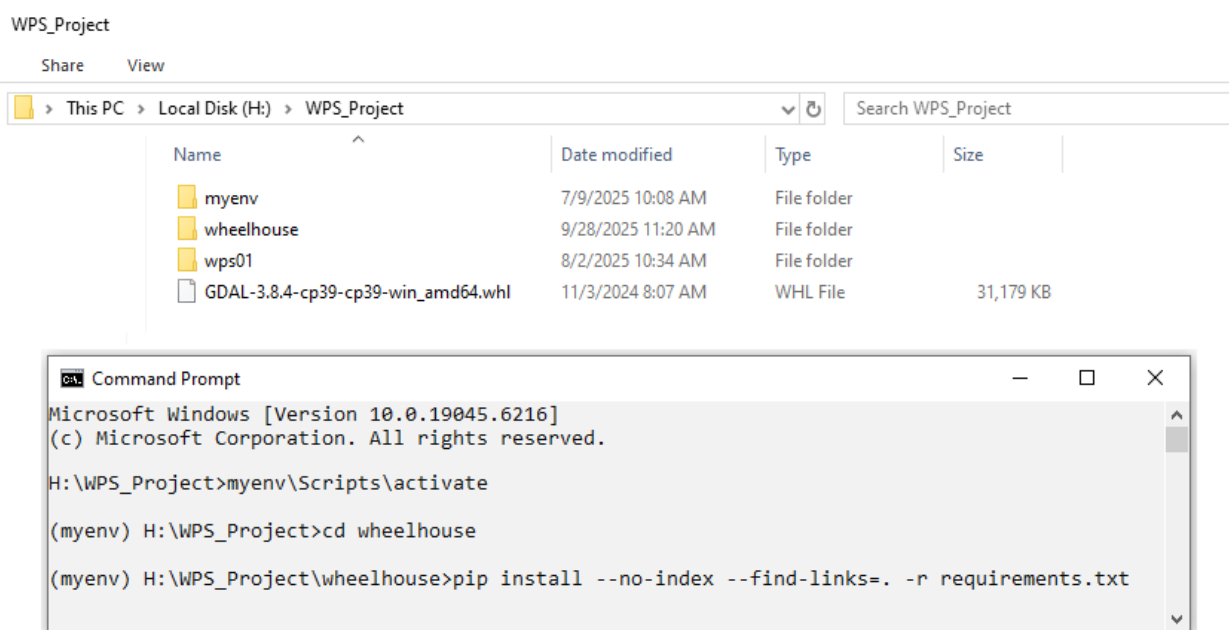
شکل ۷۸- فعال‌سازی محیط مجازی myenv

در مرحله بعد به نصب Django و PyWPS از طریق PIP می‌پردازیم. کتابخانه‌های Django و PyWPS^۱ منتشر شده است و می‌توان آنها را با استفاده از دستور زیر در محیط مجازی پایتون نصب کرد.

```
python -m pip install django pywps
```

توجه: در صورتی که بخواهید کتابخانه‌های مورد نیاز را به صورت آفلاین نصب نمایید، فولدر wheelhouse.rar را در مسیر دایرکتوری پروژه (فولدر WPS_Project) از حالت فشرده خارج نمایید. سپس در صورت فعال بودن محیط مجازی، با استفاده از دستور زیر به نصب کتابخانه‌های مورد نیاز بپردازید.

```
pip install --no-index --find-links=. -r requirements.txt
```



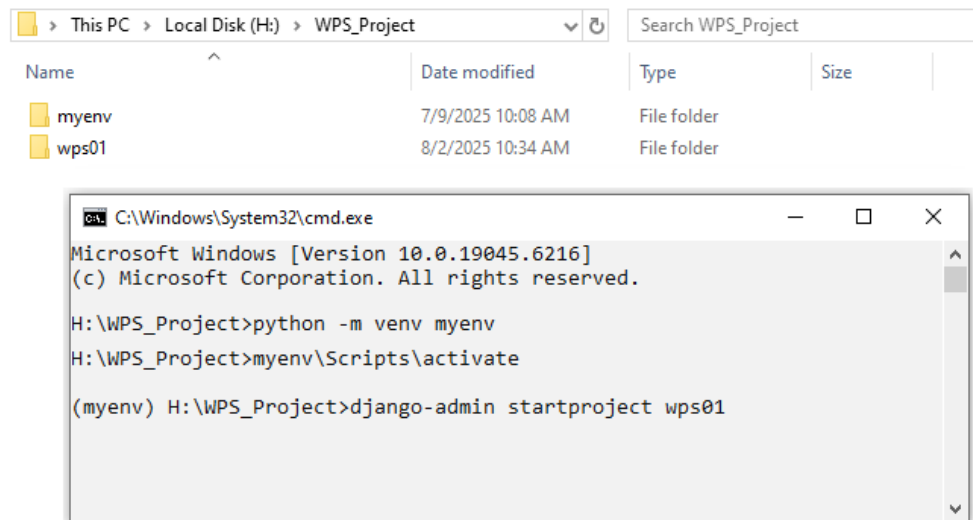
شکل ۷۹- نصب آفلاین کتابخانه‌ها

پس از نصب Django، می‌توان با استفاده از دستور django-admin یک چارچوب اولیه (اسکلت) برای پروژه خود ایجاد کرد. هنگامی که یک پروژه جدید ایجاد می‌کنید، Django حداقل نیازمندی‌هایی که برای شروع لازم است را در یک دایرکتوری جدید با نامی که توسط خودمان مشخص می‌شود، نصب می‌کند. لازم به ذکر است چارچوب Django یک ابزار خط فرمان به نام django-admin را شامل می‌شود که برای ایجاد پروژه‌های جدید استفاده می‌شود.

تذکر: توجه شود که این دستور، با ابزار تحت وب "Django Admin" که در مدیریت داده‌ها و کاربران مورد استفاده قرار می‌گیرد، اشتباه گرفته نشود.

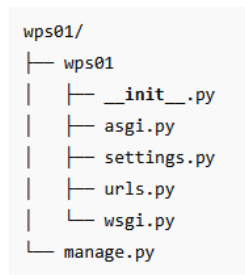
با استفاده از دستور زیر، به ایجاد یک پروژه Django با نام 'wps01' خواهیم پرداخت.

```
django-admin startproject wps01
```



شکل ۸۰- ایجاد پروژه Django با نام Wps01

پس از اجرای دستور فوق، فولدري با نام 'wps01' در دایرکتوري اصلي ايجاد می‌گردد. ساختار دایرکتوري پروژه جديد در شکل ۸۱ قابل مشاهده است.



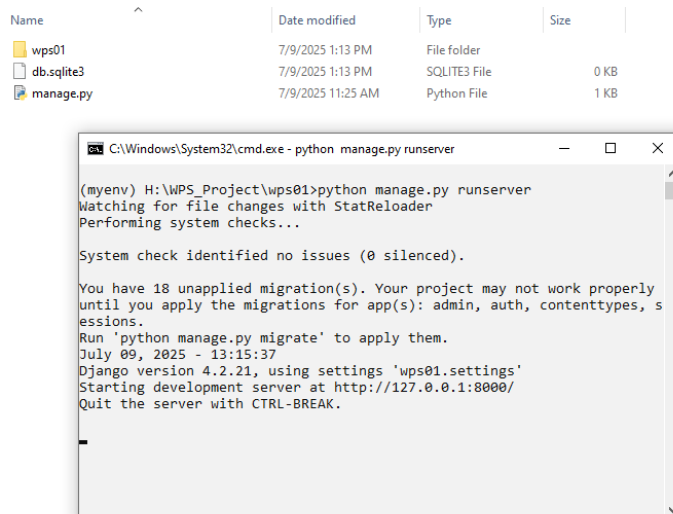
شکل ۸۱- ساختار پروژه ی Django

دایرکتوري پروژه (wps01) شامل دایرکتوري پیکربندی است که همان نام پروژه را دارد (wps01/wps01)، همچنین شامل محیط مجازی و فایل اجرایی manage.py نیز می‌باشد. فایل manage.py تقریباً همان کارهایی را انجام می‌دهد که django-admin انجام می‌دهد، با این تفاوت که به پروژه wps01 متصل بوده و از تنظیمات آن آگاه است. به منظور چک کردن درستی کار تا به اینجا می‌توان سرور توسعه‌ی Django را با استفاده از اسکریپت manage.py و دستور runserver اجرا کرد. باید توجه نمود که این کار را باید داخل دایرکتوري اصلي پروژه‌ی wps01 انجام شود.

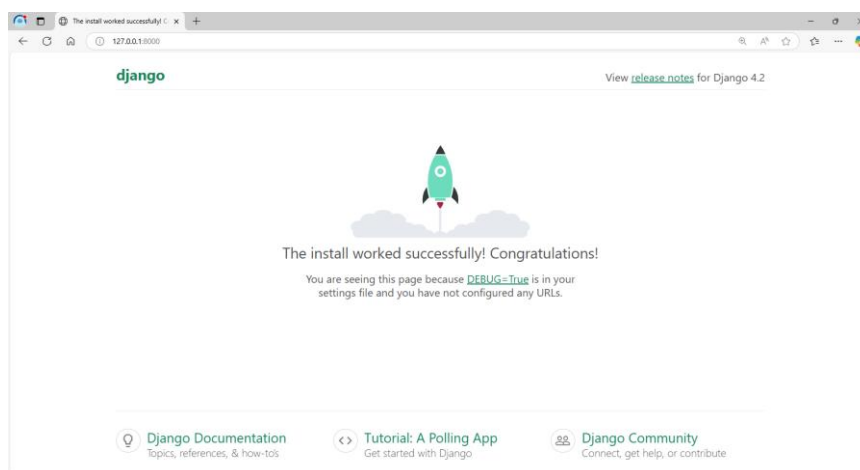
```

cd wps01
python manage.py runserver
  
```

به‌طور پیش‌فرض، این سرور به پورت ۸۰۰۰ گوش می‌دهد. آدرس http://127.0.0.1:8000/ همان جایی است که می‌توانید پروژه‌ی Django خود را در مرورگر مشاهده کرد.



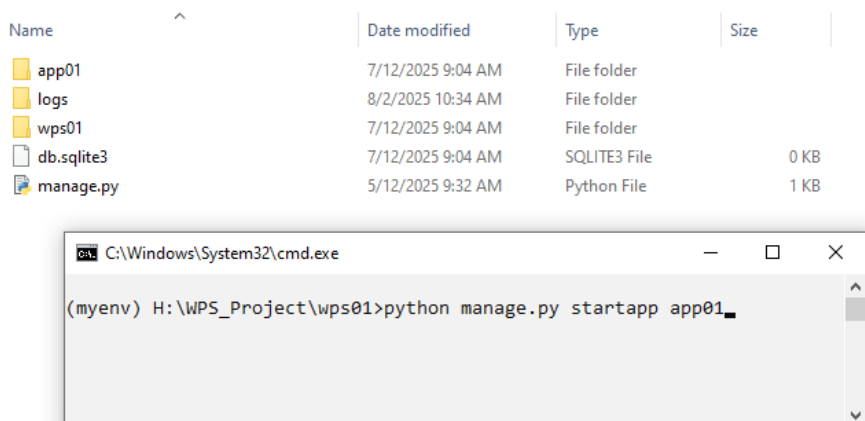
شکل ۸۲- اجرای سرور توسعه ای Django



شکل ۸۳- مشاهده ی پروژه ی Django در مرورگر

در این مرحله با دستور زیر به ایجاد یک app با نام app01 در درون پروژه می پردازیم.

python manage.py startapp app01



شکل ۸۴- ایجاد یک app در درون پروژه ی Django

به منظور ثبت برنامه‌ی جدید ایجاد شده، فایل settings.py را در مسیر wps01/wps01/settings.py باز کنید و نام برنامه‌ی خود را به لیست INSTALLED_APPS اضافه کنید.

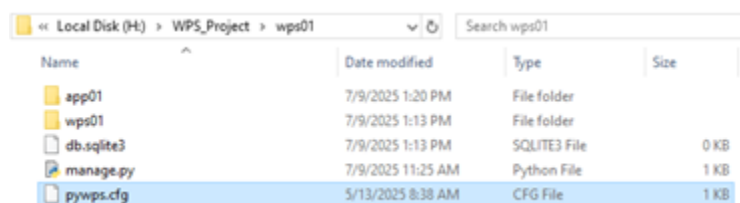
```

33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40     'app01',|
41 ]
--

```

شکل ۸۵- اضافه کردن نام app01 به تنظیمات پروژه

سپس ترجیحا در root directory به ایجاد یک فایل پیکربندی (Configuration File) با نام pywps.cfg اقدام میکنیم.



Name	Date modified	Type	Size
app01	7/9/2025 1:20 PM	File folder	
wps01	7/9/2025 1:13 PM	File folder	
db.sqlite3	7/9/2025 1:13 PM	SQLITE3 File	0 KB
manage.py	7/9/2025 11:25 AM	Python File	1 KB
pywps.cfg	5/13/2025 8:38 AM	CFG File	1 KB

شکل ۸۶- اضافه کردن فایل پیکربندی pywps.cfg به پروژه

این فایل می‌تواند شامل بخش‌ها و پارامترهای خاصی باشد که نحوه عملکرد سرور WPS، مکان ذخیره فایل‌های موقت، گزینه‌های ثبت گزارش (logging) و آدرس‌های URL که سرور به آن‌ها پاسخ می‌دهد را تعیین می‌کنند. به طور مثال بخش [metadata:main] که در شکل ۸۷ هم نشان داده شده است، در پاسخ به GetCapabilities نیز مورد استفاده قرار میگیرد.

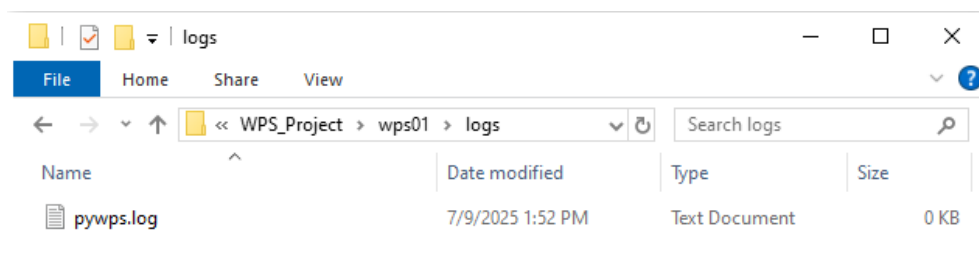
```

1  [metadata:main]
2  identification_title=PyWPS server
3  identification_abstract=PyWPS development server
4  identification_keywords_type=theme
5  identification_fees=None
6  identification_accessconstraints=None
7  provider_name= ...
8  provider_url= ...
9  contact_name=6948
10 contact_position=Dev
11 contact_address=Tehran
12 contact_city=Tehran
13 contact_stateorprovince=None
14 contact_postalcode=000 000
15 contact_country=Iran
16 contact_phone=+00 00 00 00 00
17 contact_fax=+00 00 00 00 00
18 contact_email=6948@ncc.gov.ir
19 contact_url=6948.ncc.gov.ir
20 contact_hours=07:00:00UTC
21 contact_instructions=...
22 contact_role=GISExpert
23
24 [server]
25 maxsingleinputsize=1mb
26 maxrequestsize=3mb
27 url=http://localhost:8000/wps
28 outputurl=http://localhost:8000/outputs/
29 outputpath=outputs
30 workdir=workdir
31 wd_inp_subdir=inputs
32 wd_out_subdir=outputs
33 maxprocesses=100
34 parallelprocesses=20
35
36 [logging]
37 level=INFO
38 file=logs/pywps.log
39 database=sqlite:///logs/pywps-logs.sqlite3
40 format=%(asctime)s] [% (levelname)s] file=%(pathname)s line=%(lineno)s module=%(module)s function=%(funcName)s %(message)s

```

شکل ۸۷- بخش هایی از پیکربندی شامل فراداده

حال به ایجاد مسیر logs/pywps.log همانطور که در بخش [logging] تعریف کردیم، به منظور ذخیره سازی log ها اقدام میکنیم. در غیر این صورت در هنگام اجرا با خطا مواجه می شویم.



شکل ۸۸- اضافه کردن فایل log به پروژه

به منظور مدیریت بهتر، در درون app01 به ایجاد فولدری با نام processes اقدام شده به نحوی که فرآیندهای PyWPS در آن ایجاد و تعریف می‌گردند. در درون این فولدر به ایجاد فایل پایتونی به نام seyhello اقدام میکنیم. در نمونه اجرایی اول فرایند ما دارای یک ورودی و یک خروجی که هر دو از نوع literal هستند، می‌باشد:

- LiteralData input نام از نوع رشته

- LiteralData output الحاق نام ورودی به Hello – از نوع رشته

نام پیشنهادی برای این فرایند sayHello می‌باشد (seyhello.py) که میتوان با ورود کلاسهای و ماژول‌های مورد نیاز به کدزنی آن پرداخت.

```
1 from pywps import Process, LiteralInput, LiteralOutput, UOM
2
```

شکل ۸۹- فراخوانی کلاسهای مورد نیاز برای فرایند SayHello

در مرحله بعد پس از تعریف فرایند PyWPS با نام SayHello به تعریف ورودی و خروجی پرداخته می‌شود. فرایند مورد نظر دارای یک ورودی و یک خروجی همانطور که در بالا اشاره شد می‌باشد.

```
3 class SayHello(Process):
4     def __init__(self):
5         inputs = [LiteralInput('name', 'Input name', data_type='string')]
6         outputs = [LiteralOutput('response',
7                                'Output response', data_type='string')]
```

شکل ۹۰- تعریف ورودی و خروجی‌ها در فرایند SayHello

در این قسمت فرآیند sayHello مقداردهی اولیه می‌شود. `super(SayHello, self).__init__` باعث شده sayHello از کلاس پدر خود یعنی process ارث‌بری کرده و بتواند ویژگی‌های آن را تنظیم کند. که در آن پارامترهایی نظیر عنوان، خلاصه، نسخه و لیست ورودی و خروجی‌ها برای فرآیند مذکور تنظیم می‌گردند.

```
9 super(SayHello, self).__init__(
10     self._handler,
11     identifier='say_hello',
12     title='Process Say Hello',
13     abstract='Returns a literal string output with Hello plus the input name',
14     version='0.0.0.1',
15     inputs=inputs,
16     outputs=outputs,
17     store_supported=True,
18     status_supported=True
19 )
```

شکل ۹۱- مقداردهی اولیه فرآیند SayHello

در انتها به تعریف متد handler پرداخته که در داخل آن پردازش انجام میگیرد. در حقیقت این متد به عنوان هسته ی فرآیند PyWPS در نظر گرفته می شود. در این نمونه ی اجرایی handler به گرفتن ورودی و چسباندن آن (concatenation) به کلمه ی Hello می پردازد. و در انتها خروجی را بر می گرداند.

```

22 def _handler(self, request, response):
23     response.outputs['response'].data = 'Hello ' + request.inputs['name'][0].data
24     response.outputs['response'].uom = UOM('unity')
25     return response
26

```

شکل ۹۲- تعریف متد handler به منظور انجام پردازش مورد نیاز

به واسطه ی استقرار و پیاده سازی یک سرویس WPS در مسیر wps01/wps01/wsgi.py تغییرات زیر را اعمال میکنیم. لازم به ذکر است فرایند تعریف شده در مرحله ی قبلی اینجا معرفی می شود. با انجام این کار در حقیقت کل پروژه به سرویس WPS تبدیل می گردد.

```

1  """
2  WSGI config for wps01 project.
3
4  It exposes the WSGI callable as a module-level variable named ``application``.
5
6  For more information on this file, see
7  https://docs.djangoproject.com/en/4.2/howto/deployment/wsgi/
8  """
9
10 from pywps.app.Service import Service
11 from app01.processes.sayhello import SayHello
12
13 processes = [
14     SayHello(),
15 ]
16
17 application = Service(processes, ['pywps.cfg'])
18

```

شکل ۹۳- افزودن فرایند SayHello به لیست فرایندهای فعال

حال با قرار گرفتن در مسیر دایرکتوری root دستور زیر را اجرا و سرور توسعه ای Django را اجرا میکنیم.
python manage.py runserver

Name	Date modified	Type	Size
app01	7/12/2025 9:04 AM	File folder	
logs	8/2/2025 10:34 AM	File folder	
wps01	7/12/2025 9:04 AM	File folder	
db.sqlite3	7/12/2025 9:04 AM	SQLite3 File	0 KB
manage.py	5/12/2025 9:32 AM	Python File	1 KB
pywps.cfg	5/13/2025 8:38 AM	CFG File	1 KB

```

C:\Windows\System32\cmd.exe - python manage.py runserver

(myenv) H:\WPS_Project\wps01>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
October 06, 2025 - 15:08:50
Django version 4.2.21, using settings 'wps01.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
    
```

شکل ۹۴- اجرای سرور توسعه ای Django

پس از ایجاد این فرایند می توان درخواست های GetCapabilites، DescribeProcess و Execute را ارسال و پاسخ آن را مشاهده نمود.

:GetCapabilites

request •

http://127.0.0.1:8000/?request = GetCapabilities&service = WPS&version = 1.0.0

response •

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<!-- PyWPS 4.6.0 -->
<wps:Capabilities service="WPS" version="1.0.0" xml:lang="en-US" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0,./wpsGetCapabilities_response.xsd" updateSequence="1">
  <ows:ServiceIdentification>
    <ows:Title>PyWPS server</ows:Title>
    <ows:Abstract>PyWPS development server</ows:Abstract>
    <ows:Keywords>
      <ows:Keyword>PyWPS</ows:Keyword>
      <ows:Keyword>WPS</ows:Keyword>
      <ows:Keyword>OGC</ows:Keyword>
      <ows:Keyword>processing</ows:Keyword>
      <ows:Type codeSpace="ISOTC211/19115">theme</ows:Type>
    </ows:Keywords>
    <ows:ServiceType>WPS</ows:ServiceType>
    <ows:ServiceTypeVersion>1.0.0</ows:ServiceTypeVersion>
    <ows:ServiceTypeVersion>2.0.0</ows:ServiceTypeVersion>
    <ows:Fees>None</ows:Fees>
    <ows:AccessConstraints>None</ows:AccessConstraints>
  </ows:ServiceIdentification>
  <ows:ServiceProvider>
    <ows:ProviderName>...</ows:ProviderName>
    <ows:ProviderSite xlink:href="...">
    <ows:ServiceContact>
      <ows:IndividualName>6948</ows:IndividualName>
      <ows:PositionName>Dev</ows:PositionName>
      <ows:ContactInfo>
        <ows:Phone>
          <ows:Voice>+00 00 00 00 00</ows:Voice>
          <ows:Facsimile>
        </ows:Phone>
      <ows:Address>
        <ows:DeliveryPoint>
          <ows:City>Tehran</ows:City>
          <ows:AdministrativeArea>
          <ows:PostalCode>000 000</ows:PostalCode>
          <ows:Country>Iran</ows:Country>
          <ows:ElectronicMailAddress>6948@ncc.gov.ir</ows:ElectronicMailAddress>
        </ows:Address>
      <ows:ContactInfo>
    </ows:ServiceContact>
    </ows:ServiceProvider>
  </ows:OperationsMetadata>
  <ows:Operation name="GetCapabilities">
    <ows:DCP>
      <ows:HTTP>
        <ows:Get xlink:href="http://localhost:8000/wps?">
        </ows:HTTP>
      </ows:DCP>
    </ows:Operation>
  <ows:Operation name="DescribeProcess">
    <ows:DCP>
      <ows:HTTP>
        <ows:Get xlink:href="http://localhost:8000/wps?">
        <ows:Post xlink:href="http://localhost:8000/wps?">
        </ows:HTTP>
      </ows:DCP>
    </ows:Operation>
  <ows:Operation name="Execute">
    <ows:DCP>
      <ows:HTTP>
        <ows:Get xlink:href="http://localhost:8000/wps?">
        <ows:Post xlink:href="http://localhost:8000/wps?">
        </ows:HTTP>
      </ows:DCP>
    </ows:Operation>
  </ows:OperationsMetadata>
</wps:ProcessOfferings>
<wps:Process wps:processVersion="0.0.0.1">
  <ows:Identifier>say_hello</ows:Identifier>
  <ows:Title>Process Say Hello</ows:Title>
  <ows:Abstract>
    Returns a literal string output with Hello plus the input name
  </ows:Abstract>
</wps:Process>
</wps:ProcessOfferings>
</wps:Languages>
<wps:Default>
  <ows:Language>en-US</ows:Language>
</wps:Default>
<wps:Supported>
  <ows:Language>en-US</ows:Language>
</wps:Supported>
</wps:Languages>
</wps:Capabilities>
```

شکل ۹۵- پاسخ سرور به درخواست GetCapabilities

:DescribeProcess

request •

http://127.0.0.1: 8000/? request = DescribeProcess&service = WPS&identifier = say_hello&version = 1.0.0

response •

```
<!-- PyWPS 4.6.0 -->
<wps:ProcessDescriptions xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 ../wpsDescribeProcess_response.xsd" service="WPS" version="1.0.0" xml:lang="en-US">
  <ProcessDescription wps:processVersion="0.0.0.1" storeSupported="true" statusSupported="true">
    <ows:Identifier>say_hello</ows:Identifier>
    <ows:Title>Process Say Hello</ows:Title>
    <ows:Abstract>
      Returns a literal string output with Hello plus the input name
    </ows:Abstract>
    <DataInputs>
      <Input minOccurs="1" maxOccurs="1">
        <ows:Identifier>name</ows:Identifier>
        <ows:Title>Input name</ows:Title>
        <ows:Abstract>
          <LiteralData>
            <ows:DataType ows:reference="http://www.w3.org/TR/xmlschema-2/string">string</ows:DataType>
          </LiteralData>
        </ows:Abstract>
      </Input>
    </DataInputs>
    <ProcessOutputs>
      <Output>
        <ows:Identifier>response</ows:Identifier>
        <ows:Title>Output response</ows:Title>
        <ows:Abstract>
          <LiteralOutput>
            <ows:DataType ows:reference="http://www.w3.org/TR/xmlschema-2/string">string</ows:DataType>
          </LiteralOutput>
        </ows:Abstract>
      </Output>
    </ProcessOutputs>
  </ProcessDescription>
</wps:ProcessDescriptions>
```

شکل ۹۶- پاسخ سرور به درخواست DescribeProcess

Execute

request •

http://127.0.0.1: 8000/? request = Execute&service = WPS&identifier = say_hello&version = 1.0.0&datainputs = name = Mohsen

response •

```
<wps:ExecuteResponse xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 ../wpsExecute_response.xsd" service="WPS" version="1.0.0" xml:lang="en-US" serviceInstance="http://localhost:8000/wps?request=GetCapabilities&service=WPS" statusLocation="">
  <wps:Process wps:processVersion="0.0.0.1">
    <ows:Identifier>say_hello</ows:Identifier>
    <ows:Title>Process Say Hello</ows:Title>
    <ows:Abstract>
      Returns a literal string output with Hello plus the input name
    </ows:Abstract>
  </wps:Process>
  <wps:Status creationTime="2025-05-14T08:31:26Z">
    <wps:ProcessSucceeded>PyWPS Process Process Say Hello finished</wps:ProcessSucceeded>
  </wps:Status>
  <wps:ProcessOutputs>
    <wps:Output>
      <ows:Identifier>response</ows:Identifier>
      <ows:Title>Output response</ows:Title>
      <ows:Abstract>
        <wps:Data>
          <wps:LiteralData uom="urn:ogc:def:uom:OGC:1.0:unity" dataType="string">Hello Mohsen</wps:LiteralData>
        </wps:Data>
      </ows:Abstract>
    </wps:Output>
  </wps:ProcessOutputs>
</wps:ExecuteResponse>
```

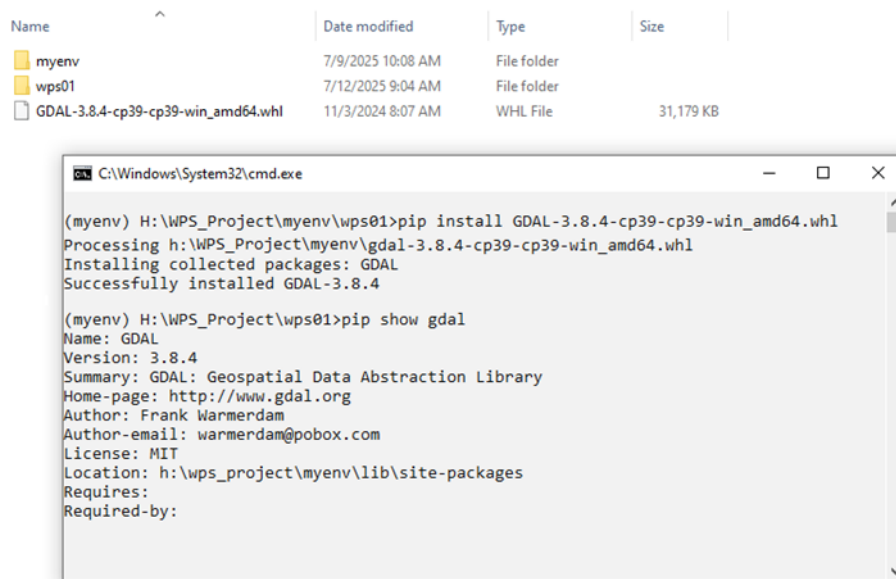
شکل ۹۷- پاسخ سرور به درخواست Execute

همانطور که دیده می شود در درخواست نام Mohsen ارسال و در پاسخ Hello Mohsen خروجی بازگردانده شد.

در مرحله ی بعد سراغ اجرای تحلیل مکانی بافر (Buffer) مبتنی بر وب می رویم. شیوه ی کار همانند نمونه ی اجرایی قبلی می باشد با این تفاوت که در این تحلیل نیازمند استفاده از کتابخانه ی GDAL/OGR می باشیم. بدین منظور پس

از دانلود wheel فایل آن متناسب با نسخه ی پایتون 3.9.11 و سیستم عامل خود، به نصب آن در محیط مجازی می پردازیم. در ویندوز، به دلیل نبودن ابزارهای کامپایل پیش فرض، توصیه می شود از نسخه های wheel استفاده شود. تذکر: نمونه اجرایی این دستورالعمل با استفاده از GDAL نسخه 3.8.4 پیاده سازی شده است. در این مرحله با استفاده دستور زیر به نصب GDAL می پردازیم.

```
pip install GDAL-3.8.4-cp39-cp39-win_amd64.whl
```



شکل ۹۸- نصب GDAL با استفاده از wheel فایل

کتابخانه (GDAL/OGR) مخفف (Geospatial Data Abstraction Library) یک ابزار قدرتمند و متن باز برای خواندن، نوشتن، و پردازش داده های مکانی است. این کتابخانه از فرمت های بسیار متنوعی برای داده های رستری (مانند TIFF، JPEG، GeoTIFF) و برداری (مانند Shapefile، GeoJSON، و KML) پشتیبانی می کند. بخش GDAL برای داده های رستری و بخش OGR برای داده های برداری طراحی شده اند، اما امروزه هر دو معمولاً به صورت یکپارچه استفاده می شوند. GDAL قابلیت هایی مانند برش، تغییر سیستم مختصات، تبدیل فرمت، و انجام تحلیل های مکانی را فراهم می کند و در پروژه های معروفی مانند QGIS، MapServer، و حتی Google Earth Engine نقش کلیدی دارد. این کتابخانه در زبان های مختلف از جمله Python، C/C++، Java و .NET قابل استفاده است و به عنوان یکی از ارکان اصلی پردازش داده های مکانی در حوزه GIS شناخته می شود.

حال به ایجاد فرایند بافر (buffer.py) در فولدر processes می پردازیم.

در ابتدا کتابخانه های مورد استفاده را وارد میکنیم. به دلیل کار با داده های برداری و سیستم های مختصات مازول های ogr و osr فراخوانی می گردند.

```

1  """
2  # Name: buffer
3  # Description: performs buffer with simple inputs - x and y
4  #
5  # Author: Mohsen Yousefzadeh
6  #
7  # Created: ----
8  # Copyright: (c) 6948
9  # Licence: NCC - GIS
10 """
11
12 from pywps import Process, LiteralInput, ComplexOutput, Format, FORMATS
13 from osgeo import ogr, osr

```

شکل ۹۹- فراخوانی کلاسها و ماژول‌های مورد نیاز

در این فرایند ما سه ورودی داریم، دو ورودی به عنوان مختصات x و y که از نوع عددی float می باشند و یک بافر با اعمال محدودیت روی مقادیر مجاز ورودی. این ویژگی سبب می شود کاربر فقط بتواند از مقادیر مجاز و کنترل شده استفاده کند. ورودی ها همه از نوع literalInput بوده ولی در خروجی استفاده از complexOutput اجازه استفاده از فرمت های پیچیده تری نظیر GML را به ما میدهد. با تعیین `as_reference=False` خروجی مستقیماً داخل پاسخ قرار داده می شود (نه به صورت لینک).

```

15 class simple_buffer(Process):
16     def __init__(self):
17
18         inputs = [LiteralInput('x', 'X coordinate', data_type='float'),
19                   LiteralInput('y', 'Y coordinate', data_type='float'),
20                   LiteralInput('buffer', 'Buffer size', data_type='float',
21                               allowed_values=(0, 1, 10, (10, 10, 100), (100, 100, 1000)))]
22
23
24         outputs = [ComplexOutput('buff_out', 'Buffered file',
25                                  supported_formats=[
26                                      Format('application/gml+xml')
27                                  ],
28                                  as_reference=False
29                              )]
30

```

شکل ۱۰۰- تعریف ورودی و خروجی ها در فرایند بافر

سپس به مقداردهی اولیه (initialization) کلاس فرآیند می پردازیم.

```

31         super(simple_buffer, self).__init__(
32             self._handler,
33             identifier='buffer',
34             version='0.1',
35             title="GDAL Buffer process",
36             abstract="The process returns buffers around the simple input features using GDAL",
37             profile='',
38             inputs=inputs,
39             outputs=outputs,
40             store_supported=True,
41             status_supported=True
42         )

```

شکل ۱۰۱- مقداردهی اولیه فرآیند بافر

سپس با استفاده از تابع handler ورودی ها دریافت، هندسه ایجاد، اعمال بافر صورت گرفته و خروجی در قالب فایل

GML ذخیره می گردد.

```

44 def _handler(self, request, response):
45     x = request.inputs['x'][0].data
46     y = request.inputs['y'][0].data
47     buffer_size = float(request.inputs['buffer'][0].data)
48     # Create an OGR geometry for the point
49     point = ogr.Geometry(ogr.wkbPoint)
50     point.AddPoint(x, y)
51     # Create a buffer around the point
52     buff = point.Buffer(buffer_size)
53     srs = osr.SpatialReference()
54     srs.ImportFromEPSG(4326)
55     # Create output GML
56     driver = ogr.GetDriverByName('GML')
57     out = 'point_buffer.gml'
58     outSource = driver.CreateDataSource(out, ["XSI:SCHEMAURI=http://schemas.opengis.net/gml/2.1.2/feature.xsd"])
59     outLayer = outSource.CreateLayer('buffered_point', srs=srs)
60     # Create a feature and set the geometry to the buffer
61     outFeature = ogr.Feature(outLayer.GetLayerDefn())
62

```

شکل ۱۰۲- تعریف متد handler به منظور انجام پردازش مورد نیاز

```

63     outFeature.SetGeometry(buff)
64     outLayer.CreateFeature(outFeature)
65     # Destroy feature and close data source
66     outFeature = None
67     outSource = None
68     # Set the output
69     response.outputs['buff_out'].output_format = FORMATS.GML
70     response.outputs['buff_out'].file = out
71     return response
72

```

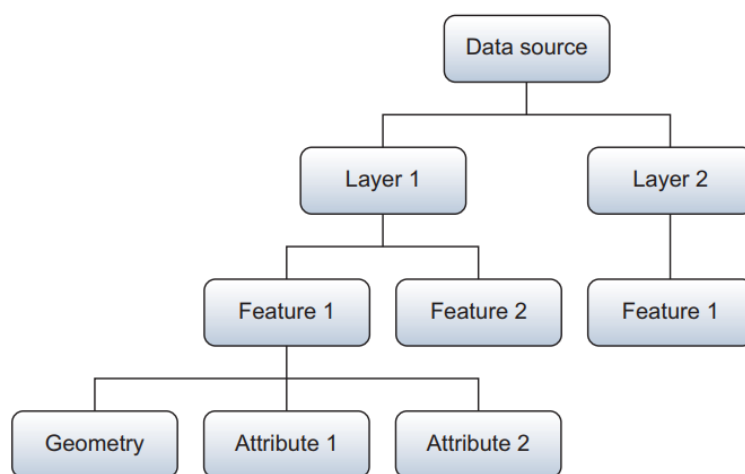
شکل ۱۰۳- ذخیره سازی خروجی تحلیل بافر به صورت GML

خطوط ۴۶ الی ۴۸ در کد بالا ورودی ها را که شامل x و y و سایز بافر می باشد، دریافت می کند. خطوط ۵۰ الی

۵۳ نقطه ای ایجاد کرده و با استفاده از Buffer(). به ایجاد بافری حول آن می پردازد. در خطوط ۵۴ و ۵۵ سیستم مختصات

آن به WGS84 تنظیم می‌گردد. در خطوط ۵۷ الی ۶۰ ابتدا از درایور GML برای تولید فایل GML استفاده شده و سپس فایل خروجی با نام point_buffer ایجاد و لایه ای با سیستم مختصات تعریف شده از قبل ایجاد می‌شود. در خطوط بعدی کد feature جدید ایجاد و هندسه ی بافر به آن اختصاص یافته و feature در لایه ذخیره می‌گردد. خطوط ۶۶ و ۶۷ نیز به منظور ازادسازی حافظه نوشته شده است. در انتها نیز فرمت خروجی به GML تنظیم شده و خروجی فرآیند به فایل point_buffer.gml اشاره می‌کند.

ساختار کلی OGR در شکل زیر به منظور درک بهتر آورده شده است. هر data source می‌تواند شامل چندین لایه باشد. هر لایه می‌تواند شامل چندین feature بوده و هر feature شامل geometry و یک یا چند attribute می‌باشد.



شکل ۱۰۴- ساختار کلاس OGR

در مرحله بعد همانند نمونه ی اجرایی قبلی در فایل wsgi موجود در مسیر wps01/wps01/wsgi.py تغییرات زیر را اعمال می‌کنیم.

```

1  """
2  WSGI config for wps01 project.
3
4  It exposes the WSGI callable as a module-level variable named ``application``.
5
6  For more information on this file, see
7  https://docs.djangoproject.com/en/4.2/howto/deployment/wsgi/
8  """
9
10 from pywps.app.Service import Service
11 from app01.processes.sayhello import SayHello
12 from app01.processes.buffer import simple_buffer
13
14 processes = [
15     SayHello(),
16     simple_buffer(),
17 ]
18
19 application = Service(processes, ['pywps.cfg'])
  
```

شکل ۱۰۵- افزودن فرایند Simple_buffer به لیست فرایندهای فعال

کد تکمیل می باشد و میتوان پس از اجرای سرور توسعه ای Django به ارسال درخواست و دریافت پاسخ اقدام نمود.

GetCapabilities

request •

<http://127.0.0.1:8000/?request=GetCapabilities&service=WPS&version=1.0.0>

response •

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<!-- PyWPS 4.6.0 -->
<wps:Capabilities service="WPS" version="1.0.0" xml:lang="en-US" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 ../wpsGetCapabilities_response.xsd" updateSequence="1">
  <ows:ServiceIdentification>
    <ows:Title>PyWPS server</ows:Title>
    <ows:Abstract>PyWPS development server</ows:Abstract>
    <ows:Keywords>
      <ows:Keyword>PyWPS</ows:Keyword>
      <ows:Keyword>WPS</ows:Keyword>
      <ows:Keyword>OGC</ows:Keyword>
      <ows:Keyword>processing</ows:Keyword>
      <ows:Type codeSpace="ISOTC211/19115">theme</ows:Type>
    </ows:Keywords>
    <ows:ServiceType>WPS</ows:ServiceType>
    <ows:ServiceTypeVersion>1.0.0</ows:ServiceTypeVersion>
    <ows:ServiceTypeVersion>2.0.0</ows:ServiceTypeVersion>
    <ows:Fees>None</ows:Fees>
    <ows:AccessConstraints>None</ows:AccessConstraints>
  </ows:ServiceIdentification>
  <ows:ServiceProvider>
    <ows:ProviderName>...</ows:ProviderName>
    <ows:ProviderSite xlink:href="...">
    <ows:ServiceContact>
      <ows:IndividualName>6948</ows:IndividualName>
      <ows:PositionName>Dev</ows:PositionName>
      <ows:ContactInfo>
        <ows:Phone>
          <ows:Voice>+00 00 00 00 00</ows:Voice>
          <ows:Facsimile>
        </ows:Phone>
        <ows:Address>
          <ows:DeliveryPoint>
          <ows:City>Tehran</ows:City>
          <ows:AdministrativeArea>
          <ows:PostalCode>000 000</ows:PostalCode>
          <ows:Country>Iran</ows:Country>
          <ows:ElectronicMailAddress>6948@ncc.gov.ir</ows:ElectronicMailAddress>
        </ows:Address>
      </ows:ContactInfo>
    </ows:ServiceContact>
  </ows:ServiceProvider>
  <ows:OperationsMetadata>
    <ows:Operation name="GetCapabilities">
      <ows:DCP>
        <ows:HTTP>
          <ows:Get xlink:href="http://localhost:8000/wps">
          </ows:HTTP>
        </ows:DCP>
      </ows:Operation>
    <ows:Operation name="DescribeProcess">
      <ows:DCP>
        <ows:HTTP>
          <ows:Get xlink:href="http://localhost:8000/wps">
          <ows:Post xlink:href="http://localhost:8000/wps">
          </ows:HTTP>
        </ows:DCP>
      </ows:Operation>
    <ows:Operation name="Execute">
      <ows:DCP>
        <ows:HTTP>
          <ows:Get xlink:href="http://localhost:8000/wps">
          <ows:Post xlink:href="http://localhost:8000/wps">
          </ows:HTTP>
        </ows:DCP>
      </ows:Operation>
    </ows:OperationsMetadata>
  </ows:ProcessOfferings>
  <ows:Process wps:processVersion="0.0.1">
    <ows:Identifier>say_hello</ows:Identifier>
    <ows:Title>Process Say Hello</ows:Title>
    <ows:Abstract>
      Returns a literal string output with Hello plus the input name
    </ows:Abstract>
  </ows:Process>
  <ows:Process wps:processVersion="0.1">
    <ows:Identifier>buffer</ows:Identifier>
    <ows:Title>GDAL Buffer process</ows:Title>
    <ows:Abstract>
      The process returns buffers around the simple input features using GDAL
    </ows:Abstract>
  </ows:Process>
</wps:ProcessOfferings>
<wps:Languages>
  <ows:Default>
    <ows:Language>en-US</ows:Language>
  </ows:Default>
  <ows:Supported>
    <ows:Language>en-US</ows:Language>
  </ows:Supported>
</wps:Languages>
</wps:Capabilities>
```

شکل ۱۰۶- پاسخ سرور به درخواست GetCapabilities

همانطور که دیده می‌شود فرایند جدید (بافر) در XML خروجی اضافه شده است.

:DescribeProcess

request •

http://127.0.0.1: 8000/? request = DescribeProcess&service = WPS&identifier = buffer&version = 1.0.0

response •

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<!-- PyWPS 4.6.0 -->
<wps:ProcessDescriptions xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 ../wpsDescribeProcess_response.xsd" service="WPS" version="1.0.0" xml:lang="en-US">
  <ProcessDescription wps:processVersion="0.1" storeSupported="true" statusSupported="true">
    <ows:Identifier>buffer</ows:Identifier>
    <ows:Title>GDAL Buffer process</ows:Title>
    <ows:Abstract>
      The process returns buffers around the simple input features using GDAL
    </ows:Abstract>
    <DataInputs>
      <Input minOccurs="1" maxOccurs="1">
        <ows:Identifier>x</ows:Identifier>
        <ows:Title>X coordinate</ows:Title>
        <ows:Abstract>
          <LiteralData>
            <ows:DataType ows:reference="http://www.w3.org/TR/xmlschema-2/#float">float</ows:DataType>
          </LiteralData>
        </Input>
        <Input minOccurs="1" maxOccurs="1">
          <ows:Identifier>y</ows:Identifier>
          <ows:Title>Y coordinate</ows:Title>
          <ows:Abstract>
            <LiteralData>
              <ows:DataType ows:reference="http://www.w3.org/TR/xmlschema-2/#float">float</ows:DataType>
            </LiteralData>
          </Input>
          <Input minOccurs="1" maxOccurs="1">
            <ows:Identifier>buffer</ows:Identifier>
            <ows:Title>Buffer size</ows:Title>
            <ows:Abstract>
              <LiteralData>
                <ows:DataType ows:reference="http://www.w3.org/TR/xmlschema-2/#float">float</ows:DataType>
              </ows:AllowedValues>
                <ows:Value>0</ows:Value>
                <ows:Value>1</ows:Value>
                <ows:Value>10</ows:Value>
                <ows:Range ows:rangeClosure="closed">
                  <ows:MinimumValue>10</ows:MinimumValue>
                  <ows:MaximumValue>100</ows:MaximumValue>
                  <ows:Spacing>10</ows:Spacing>
                </ows:Range>
                <ows:Range ows:rangeClosure="closed">
                  <ows:MinimumValue>100</ows:MinimumValue>
                  <ows:MaximumValue>1000</ows:MaximumValue>
                  <ows:Spacing>100</ows:Spacing>
                </ows:Range>
              </ows:AllowedValues>
            </LiteralData>
          </Input>
        </DataInputs>
        <ProcessOutputs>
          <Output>
            <ows:Identifier>buff_out</ows:Identifier>
            <ows:Title>Buffered file</ows:Title>
            <ows:Abstract>
              <ComplexOutput>
                <Default>
                  <Format>
                    <MimeType>application/gml+xml</MimeType>
                  </Format>
                </Default>
                <Supported>
                  <Format>
                    <MimeType>application/gml+xml</MimeType>
                  </Format>
                </Supported>
              </ComplexOutput>
            </Output>
          </ProcessOutputs>
        </ProcessDescription>
      </wps:ProcessDescriptions>
```

شکل ۱۰۷- پاسخ سرور به درخواست DescribeProcess

Execute

request •
 http://127.0.0.1: 8000/? request = Execute&service = WPS&identifier = buffer&version = 1.0.0&DataInputs = x = 38.0;y = 28.0;buffer = 10

response •

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?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/1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" service="WPS" version="1.0.0" xsi:schemaLocation="http://www.w3.org/2001/XMLSchema-instance http://www.w3.org/2001/XMLSchema-instance">
  <wps:Process wps:processVersion="0.1">
    <ows:Identifier>buffer</ows:Identifier>
    <ows:Title>GDAL Buffer process</ows:Title>
    <ows:Abstract>
      The process returns buffers around the simple input features using GDAL.
    </ows:Abstract>
    <wps:Process>
      <wps:Status creationTime="2025-05-17T07:57:25Z">
        <wps:ProcessSucceeded>PyWPS Process GDAL Buffer process finished</wps:ProcessSucceeded>
      </wps:Status>
      <wps:ProcessOutputs>
        <wps:Output>
          <ows:Identifier>buff_out</ows:Identifier>
          <ows:Title>Buffered file</ows:Title>
          <ows:Abstract>
            <wps:Data>
              <wps:ComplexData mimeType="application/gml+xml" encoding="utf-8" schema="http://www.opengis.net/gml/2.1.2/feature.xsd">
                <gml:boundedBy>
                  <gml:Envelope srsName="urn:ogc:def:crs:EPSG:4326">
                    <gml:lowerCorner>18 28</gml:lowerCorner>
                    <gml:upperCorner>38 48</gml:upperCorner>
                  </gml:Envelope>
                </gml:boundedBy>
                <ogr:featureMember>
                  <ogr:buffered_point gml:id="buffered_point.0">
                    <gml:boundedBy>
                      <gml:Envelope srsName="urn:ogc:def:crs:EPSG:4326">
                        <gml:lowerCorner>18 28</gml:lowerCorner>
                        <gml:upperCorner>38 48</gml:upperCorner>
                      </gml:Envelope>
                    </gml:boundedBy>
                    <ogr:geometryProperty>
                      <gml:Polygon srsName="urn:ogc:def:crs:EPSG:4326" gml:id="buffered_point_geom.0">
                        <gml:exterior>
                          <gml:LinearRing>
                            <gml:posList>
                              28 48 27.4766404375706 47.9862953475457 26.9547153673235 47.9452189536827 26.4356553495977 47.8768834059514 25.9208830918224 47.7814760073381 25.4118095489748 47.6592582628907 24.9098300562505 47.5105651629515 24.416320504547 47.335804264972 23.932633569242 47.135454576426 23.4600950026045 46.9100652418837 23.0 46.6602540378444 22.5536096498497 46.3867056794542 22.1221474770753 46.0901699437495 21.7067960895016 45.7714596145697 21.3086939364114 45.4314482547739 20.9289321881345 45.0710678118655 20.5685517452261 44.6913060635886 20.2285403854303 44.2932039104984 19.9098300562505 43.877852529247 19.6132943205458 43.4463903501503 19.3397459621556 43.0 19.0899347581163 42.5399049973955 18.864545423574 42.067366430758 18.664195735028 41.583679495453 18.4894348370485 41.0901699437495 18.3407417371093 40.5881904510252 18.2185239926619 40.079169081776 18.1231165940486 39.5643446504023 18.0547810463173 39.0452846326765 18.0137046524543 38.5233595624294 18.38 18.0137046524543 37.4766404375706 18.0547810463173 36.9547153673235 18.1231165940486 36.4356553495977 18.2185239926619 35.9208830918224 18.3407417371093 35.4118095489748 18.4894348370485 34.9098300562505 18.664195735028 34.416320504547 18.864545423574 33.932633569242 19.0899347581163 33.4600950026045 19.3397459621556 33.0 19.6132943205458 32.5536096498497 19.9098300562505 32.1221474770753 20.2285403854303 31.7067960895016 20.5685517452261 31.3086939364114 20.9289321881345 30.9289321881345 21.3086939364114 30.5685517452261 21.7067960895016 20.2285403854303 22.1221474770753 20.9098300562505 22.5536096498497 20.6132943205458 23.0 29.3397459621556 23.4600950026045 29.0899347581163 23.932633569242 28.864545423574 24.416320504547 28.664195735028 24.9098300562505 28.4894348370485 25.4118095489748 28.3407417371093 25.9208830918224 28.1231165940486 26.9547153673235 28.0547810463173 27.4766404375706 28.0137046524543 28 28 28.5233595624294 28.0137046524543 29.0452846326765 28.0547810463173 29.5643446504023 28.1231165940486 30.079169081776 28.2185239926619 30.5881904510252 28.3407417371093 31.0901699437495 28.4894348370485 31.583679495453 28.664195735028 32.067366430758 28.864545423574 32.5399049973955 29.0899347581163 33.0 29.3397459621556 33.4463903501503 29.6132943205458 33.877852529247 29.9098300562505 34.2932039104984 30.2285403854303 34.6913060635886 30.5685517452261 35.0710678118655 30.9289321881345 35.4314482547739 31.3086939364114 35.7714596145697 31.7067960895016 36.0901699437495 32.1221474770753 36.3867056794542 32.5536096498497 36.6602540378444 33.0 36.9100652418837 33.4600950026045 37.135454576426 33.932633569242 37.335804264972 34.416320504547 37.5105651629515 34.9098300562505 37.6592582628907 35.4118095489748 37.7814760073381 35.9208830918224 37.8768834059514 36.4356553495977 37.9452189536827 36.9547153673235 37.9862953475457 37.4766404375706 38 38 37.9862953475457 38.5233595624294 37.9452189536827 39.0452846326765 37.8768834059514 39.5643446504023 37.7814760073381 37.6592582628907 40.5881904510252 37.5105651629515 41.0901699437495 37.335804264972 41.583679495453 37.135454576426 42.067366430758 36.9100652418837 42.5399049973955 36.6602540378444 43.0 36.3867056794542 43.4463903501503 36.0901699437495 43.877852529247 35.7714596145697 44.2932039104984 35.4314482547739 44.6913060635886 35.0710678118655 45.0710678118655 34.6913060635886 45.4314482547739 34.2932039104984 45.7714596145697 33.877852529247 46.0901699437495 33.4463903501503 46.3867056794542 33.0 46.6602540378444 32.5399049973955 46.9100652418837 32.067366430758 47.135454576426 31.583679495453 47.335804264972 31.0901699437495 47.5105651629515 30.5881904510252 47.6592582628907 30.079169081776 47.7814760073381 29.5643446504023 47.8768834059514 29.0452846326765 47.9452189536827 28.5233595624294 47.9862953475457 28 48

```

شکل ۱۰۸- پاسخ سرور به درخواست Execute

خروجی در فرمت GML و در سیستم مختصات WGS84 نمایش داده می شود. لازم به ذکر است خروجی بافر ایجاد شده به درجه می باشد (۱۰ درجه). به منظور تبدیل آن به متر با تغییر خط ۵۵ کد در فرایند buffer.py از EPSG:4326 به EPSG:32636 خروجی بافری به طول ۱۰ متر حول نقطه ی ورودی در سیستم تصویر UTM زون ۳۶ ایجاد خواهد شد.

55 srs.ImportFromEPSG(32636)]

شکل ۱۰۹- تبدیل خروجی به متر با تغییر WGS84 به سیستم تصویر UTM



**Islamic Republic of Iran
Plan and Budget Organization
National Cartographic Center**

Technical Guide for Web Processing Service (WPS)

No:

Last Edition: 06-22-2025

National Cartographic Center GIS Department

www.ncc.gov.ir

<http://ncc.gov.ir>

2025