



جمهوری اسلامی ایران
سازمان برنامه و بودجه کشور
سازمان نقشه‌برداری کشور

دستورالعمل پیاده‌سازی تحلیل‌های مکانی تحت وب (WPS)

نگارش ۱/۰

آخرین ویرایش : ۲۹-۰۱-۱۴۰۵

سازمان نقشه‌برداری کشور **http://www.ncc.gov.ir**

اداره کل نظارت، کنترل فنی و استاندارد **گروه استانداردسازی**

فروردین ۱۴۰۵

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 در سطح کارشناسی و اجرایی غیرقابل دسترس بوده‌اند. ازاین‌رو، تدوین سندی که ضمن حفظ دقت علمی، مفاهیم پایه را به زبانی ساده و قابل فهم ارائه داده و هم‌زمان جنبه‌های عملی و اجرایی پیاده‌سازی را به‌صورت مرحله‌به‌مرحله پوشش دهد، ضروری به نظر می‌رسد.

دستورالعمل حاضر با رویکردی کاربردمحور و با بهره‌گیری از تجربیات عملی کارشناسان سازمان نقشه‌برداری کشور تهیه شده است. در این دستورالعمل تلاش شده است مفاهیم نظری مورد نیاز به‌صورت مقدماتی و هدفمند بیان شده و سپس نحوه استفاده و پیاده‌سازی سرویس‌های پردازشی تحت وب در قالب مثال‌ها و سناریوهای عملی تشریح گردد. هدف اصلی، تسهیل درک، یادگیری و به‌کارگیری این سرویس‌ها برای کارشناسان و کاربران حوزه اطلاعات مکانی است تا زمینه لازم برای توسعه، بومی‌سازی و استفاده مؤثر از این فناوری در پروژه‌های عملی فراهم شود.

شایان ذکر است که در این نسخه از دستورالعمل، تمرکز اصلی بر تبیین مفاهیم پایه و نحوه پیاده‌سازی عملی سرویس‌های پردازشی تحت وب (WPS) بوده و مباحثی نظیر استقرار در بسترهای DevOps و رایانش ابری، بهره‌گیری از پردازش توزیع‌یافته و موازی مبتنی بر فناوری‌هایی مانند Dask و Apache Spark و همچنین چارچوب‌های پیشرفته امنیت، پایش و مقیاس‌پذیری سرویس‌ها، مورد پیاده سازی و برر سی قرار نگرفته‌اند. با این حال، این مو ضوعات به‌عنوان محورهای توسعه‌ای مهم شناسایی شده و پیش‌بینی می‌شود در نسخه‌های آتی دستورالعمل به‌صورت جامع‌تر و کاربردی‌تر مورد توجه و تشریح قرار گیرند.

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

فروردین ۱۴۰۵

جناب آقای پیمان بکتاش – معاون محترم فنی و تولید اطلاعات مکانی
جناب آقای حمیدرضا سیدین بروجنی – معاون محترم فناوری و ارائه اطلاعات مکانی
جناب آقای نوربخش عسکری – معاون محترم توسعه مدیریت و منابع
جناب آقای سید عبدالرضا سعادت – مدیرکل محترم نظارت ، کنترل فنی و استاندارد
جناب آقای رامین پاپی – مدیرکل محترم دفتر ریاست ، روابط عمومی و امور بین الملل
جناب آقای علی سلطانپور – مدیر محترم آبنگاری و امور جزر و مدی
سرکار خانم معصومه آمیغ پی – مدیر کل محترم نقشه برداری زمینی و بنیادی
جناب آقای منصور حسن پور – مدیر کل محترم نقشه برداری هوایی و فضایی
جناب آقای علی مدد – مدیر کل محترم سامانه های ملی و زیر ساخت های اطلاعات مکانی
جناب آقای مهدی سعیدی انجیله – مدیر محترم نقشه نگاری و اطلس های ملی
جناب آقای نعمت الله حبشی – مدیر محترم خدمات فنی
سرکار خانم فاطمه اسدی – مدیرکل محترم دفتر فناوری اطلاعات و امنیت فضای مجازی

موضوع : ابلاغ دستورالعمل تحلیل مکانی تحت وب (WPS)

سلام علیکم؛

در عصر حاضر، داده‌های مکانی به عنوان یکی از ارکان اساسی تصمیم‌گیری‌های سازمانی مطرح هستند. اهمیت این داده‌ها زمانی دوچندان می‌شود که بتوان خدمات تحلیلی پیشرفته‌ای را از طریق بسترهای نوین و در دسترس، مانند وب، به کاربران ارائه داد. در همین راستا، به پیوست **"دستورالعمل پیاده‌سازی تحلیل‌های مکانی تحت وب (WPS)"** به عنوان گامی بنیادین به منظور استانداردسازی، تسهیل دسترسی و ارتقاء چشمگیر کیفیت این خدمات، جهت بهره‌برداری و سایر اقدامات لازم ابلاغ می‌گردد. در این سند، تلاش شده است مفاهیم نظری مورد نیاز، به‌صورت هدفمند بیان شده و سپس نحوه استفاده و پیاده‌سازی سرویس‌های پردازشی تحت وب در قالب مثال‌ها و سناریوهای عملی تشریح گردد. هدف اصلی از تدوین این دستورالعمل، تسهیل درک، یادگیری و به‌کارگیری مؤثر سرویس‌های تحلیل مکانی تحت وب برای تمامی کارشناسان و کاربران حوزه اطلاعات مکانی است تا زمینه لازم برای توسعه، بومی‌سازی و استفاده مؤثرتر از این فناوری در پروژه‌های عملی در کشور عزیزمان ایران فراهم شود.

سیداسکندرصیدایی

بسمه تعالی

صورتجلسه تایید استاندارد / دستورالعمل سازمان نقشه برداری کشور

عنوان: تائیدیه نسخه دستورالعمل با عنوان "تحلیل مکانی تحت وب (WPS)"

جمع بندی:

- نسخه دستورالعمل با عنوان "تحلیل مکانی تحت وب (WPS)" در تاریخ ۱۴۰۴/۱۰/۲۳ مورد تائید می باشد.

پیمان بکتاش	حمیدرضا سیدین بروجنی	ساحل قلعه شاهینی
مهدی سعیدی انجیله	علی سلطانپور	حمیده چراغی
زهرا کشمیری	فرزانه ربیعی	

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

اعضای گروه تهیه کننده:

میلاد رفیعی طرقی (مسئول گروه)	سازمان نقشه‌برداری کشور	کارشناسی ارشد GIS
محسن یوسف زاده سورکی	سازمان نقشه‌برداری کشور	دانشجوی دکتری GIS
نیما قاسملو	سازمان نقشه‌برداری کشور	دکتری RS & GIS
زهرا رضائی	سازمان نقشه‌برداری کشور	دکتری RS & GIS

اعضای گروه نظارت :

مهدی سعیدی انجیله	سازمان نقشه‌برداری کشور	کارشناسی ارشد IT

اعضای گروه هدایت و راهبری:

پیمان بکتاش	سازمان نقشه‌برداری کشور	کارشناسی ارشد GIS
ساحل قلعه‌شاهینی	سازمان نقشه‌برداری کشور	کارشناسی ارشد RS & GIS

فهرست مطالب

فصل اول ۱

۱-۱- مقدمه ۳

۱-۲- تحلیل مکانی ۳

۱-۳- چالش‌های موجود در تحلیل‌های مکانی ۵

۱-۳-۱- حجم زیاد داده‌ها (Big Data) ۵

۱-۳-۲- تنوع داده‌ها و فرمت‌های مختلف ۶

۱-۳-۳- کارایی و مقیاس‌پذیری پردازش‌ها ۶

۱-۳-۴- یکپارچگی و ارتباط بین داده‌ها ۶

۱-۳-۵- دقت و صحت داده‌ها ۶

۱-۳-۶- قابلیت تعامل در سیستم‌های توزیع‌شده ۷

۱-۳-۷- پردازش و ذخیره‌سازی داده‌های سه‌بعدی و پیچیده ۷

۱-۴- مقدمه‌ای بر تحلیل‌های مکانی دسکتاپ و برخط ۷

۱-۵- مزایا و معایب تحلیل‌های مکانی دسکتاپ و برخط ۸

۱-۵-۱- تحلیل‌های مکانی دسکتاپ ۸

۱-۵-۲- تحلیل‌های مکانی برخط ۹

۱-۶- بررسی چالش‌های فنی و زیرساختی در پیاده‌سازی WPS ۹

۱-۶-۱- چالش‌های فنی در پیاده‌سازی WPS ۱۰

۱-۶-۲- چالش‌های زیرساختی در پیاده‌سازی WPS ۱۰

۱-۷- مجموعه استاندارد OGC در داده‌های مکانی ۱۱

۱-۷-۱-Web Map Service (WMS) ۱۲

۱-۷-۲-Web Feature Service (WFS) ۱۲

۱-۷-۳-Web Coverage Service (WCS) ۱۲

۱-۷-۴-Wep Processing Service (WPS) ۱۲

۱-۷-۵-Catalogue Service for the Web (CSW) ۱۳

۱-۷-۶-Sensor Observation Service (SOS) ۱۳
۱-۷-۷-Web Map Tile Service (WMTS) ۱۳
۱-۸- پیاده‌سازی‌های مختلف WPS و ارزیابی آن ها ۱۳
۱-۹- نیازمندی‌ها و مراحل پیاده‌سازی تحلیل مکانی تحت وب ۱۵
فصل دوم ۱۸
۲-۱- استاندارد WPS ۲۰
۲-۲- پیکربندی WPS ۲۱
۲-۲-۱-Service Metadata ۲۱
۲-۲-۲-Execution and resource settings ۲۳
۲-۲-۳-Process status page ۲۴
۲-۲-۴-WPS security page ۲۵
۲-۳- دامنه سرویس ۲۹
۲-۴- نمای کلی WPS ۲۹
۲-۵- توابع WPS ۲۹
۲-۵-۱- GetCapabilities ۳۰
۲-۵-۲- DescribeProcess ۳۱
۲-۵-۳- Execute ۳۲
۲-۶- تحلیل WPS ۳۳
۲-۶-۱-Process chaining ۳۴
۲-۶-۲-WPS Request Builder ۳۴
۲-۷- ماهیت عمومی و میان افزار WPS ۳۵
فصل سوم ۳۸
۳-۱- مراحل راه اندازی سرویس WPS در ژئوسرور ۴۰
۳-۲- نحوه پیاده سازی سرویس WPS در ژئوسرور ۴۲
فصل چهارم ۴۴
۴-۱- تحلیل Buffer ۴۶
۴-۲- تحلیل Intersection ۵۱

۴-۳- تحلیل Clip ۵۵
۴-۴- تحلیل Nearest ۵۸
۴-۵- تحلیل Within ۶۰
۴-۶- تحلیل Contains ۶۲
۴-۷- تحلیل Contour ۶۴
۴-۸- تحلیل Query ۶۷
فصل پنجم ۶۹
۵-۱- اجرای یک پروژه موردی ۷۱
فصل ششم ۷۹
۶-۱- فرآیند (process) در استاندارد OGC ۸۱
۶-۲- ورودی و خروجی داده در استاندارد OGC ۸۱
۶-۲-۱-LiteralData ۸۱
۶-۲-۲-ComplexData ۸۲
۶-۲-۳-BoundingBoxData ۸۲
۶-۳- انتقال داده به فرآیند (passing data to process) ۸۲
۶-۴- درخواست فرآیند به صورت همزمان و غیرهمزمان ۸۲
۶-۴-۱- حالت همزمان (Synchronous mode) ۸۲
۶-۴-۲- حالت غیرهمزمان (Asynchronous mode) ۸۳
۶-۵- وضعیت فرایند (process Status) ۸۳
۶-۶- کدگذاری درخواست HTTP GET و POST ۸۳
۶-۶-۱- جفت کلید-مقدار (Key-value pairs) ۸۳
۶-۶-۲- محتوای XML (XML payload) ۸۴
۶-۷- نصب و راه اندازی PyWPS ۸۴
۶-۸- فرآیندها ۸۴
۶-۸-۱- نوشتن یک فرایند (writing a process) ۸۴
۶-۹- پیاده‌سازی نمونه اجرایی بر اساس PyWPS ۸۵

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 سرویس‌های 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 ۳۳
شکل ۱۶- PROCESS CHAINING ۳۴
شکل ۱۷- GEOSERVER DEMOS ۳۴
شکل ۱۸- WPS REQUEST BUILDER ۳۵
شکل ۱۹- EXECUTE PROCESS IN NEW PAGE ۳۵
شکل ۲۰- صفحه دانلود GEOSERVER ۴۰
شکل ۲۱- دانلود نسخه‌های GEOSERVER ۴۰
شکل ۲۲- صفحه دانلود افزونه‌های WPS ۴۱
شکل ۲۳- قرار دادن افزونه WPS در مسیر LIB در نرم‌افزار GEOSERVER ۴۱
شکل ۲۴- استفاده از ماژول DEMOS برای ساخت درخواست‌های GEOSERVER ۴۲
شکل ۲۵- انتخاب گزینه WPS REQUEST BUILDER از ماژول DEMOS ۴۲
شکل ۲۶- انتخاب تحلیل مورد نظر منوی CHOOSE PROCESS ۴۳
شکل ۲۷- انتخاب تحلیل BUFFER از منوی CHOOSE PROCESS ۴۶
شکل ۲۸- انتخاب نوع ورودی از قسمت INPUT FEATURE COLLECTION ۴۶
شکل ۲۹- انتخاب لایه مورد نظر در حالت VECTOR_LAYER به عنوان لایه ورودی ۴۷
شکل ۳۰- انتخاب لایه مورد نظر در حالت TEXT به عنوان لایه ورودی ۴۷
شکل ۳۱- وارد کردن آدرس URL لایه ورودی مورد نظر در حالت REFERENCE ۴۸
شکل ۳۲- وارد کردن مقدار فاصله مورد نظر مربوط به تحلیل BUFFER ۴۸
شکل ۳۳- انتخاب فرمت استاندارد مورد نظر به عنوان خروجی تحلیل BUFFER ۴۹

شکل ۳۴- دریافت خروجی تحلیل BUFFER در زبان XML ۴۹
شکل ۳۵- نمایش انواع فرمت خروجی تحلیل‌های WPS در نرم‌افزار GEOSERVER ۵۰
شکل ۳۶- انتخاب تحلیل INTERSECTION از منوی CHOOSE PROCESS بخش WPS REQUEST BUILDER ۵۱
شکل ۳۷- انتخاب حالت SUBPROCESS به عنوان ورودی حالت اول در تحلیل INTERSECTION ۵۲
شکل ۳۸- انتخاب گزینه DEFINE/EDIT جهت معرفی پارامترهای تحلیل SUBPROCESS ۵۲
شکل ۳۹- انتخاب تحلیل مورد نظر به عنوان تحلیل دوم ۵۲
شکل ۴۰- انتخاب لایه ورودی از تحلیل BUFFER و تعیین مقدار فاصله ۵۳
شکل ۴۱- مشاهده کد XML حاصل از تحلیل BUFFER و انتخاب لایه دوم ۵۳
شکل ۴۲- انتخاب گزینه مربوطه به INTERSECTIONMODE ۵۴
شکل ۴۳- انتخاب فرمت استاندارد مورد نظر به عنوان خروجی تحلیل INTERSECTION ۵۴
شکل ۴۴- دریافت خروجی تحلیل INTERSECTION در زبان XML ۵۵
شکل ۴۵- انتخاب تحلیل CLIP از منوی CHOOSE PROCESS ۵۶
شکل ۴۶- انتخاب لایه ورودی و حالت CLIP GEOMETRY از تحلیل CLIP ۵۶
شکل ۴۷- انجام تنظیمات مورد نیاز CLIP GEOMETRY مربوط به تحلیل CLIP ۵۷
شکل ۴۸- انتخاب فرمت استاندارد مورد نظر به عنوان خروجی تحلیل CLIP ۵۷
شکل ۴۹- ایجاد فایل XML به عنوان خروجی تحلیل CLIP ۵۷
شکل ۵۰- انتخاب تحلیل NEAREST و لایه مورد نظر ۵۸
شکل ۵۱- انجام تنظیمات مورد نظر جهت وارد کردن مختصات نقطه‌ای و سیستم مختصات لایه مربوطه ۵۹
شکل ۵۲- معرفی کد EPSG سیستم مختصات و سیستم تصویر لایه مورد نظر ۵۹
شکل ۵۳- خروجی حاصل از تحلیل NEAREST ۵۹
شکل ۵۴- انتخاب تحلیل JTS:WITHIN ۶۰
شکل ۵۵- معرفی لایه اول به صورت TEXT در قالب GML3.1 ۶۱
شکل ۵۶- تعیین نوع ورودی و معرفی آدرس URL لایه دوم ۶۱
شکل ۵۷- فایل XML حاصل از تحلیل WITHIN ۶۲
شکل ۵۸- انتخاب تحلیل CONTAINS و انجام تنظیمات مورد نظر ۶۳
شکل ۵۹- فایل XML حاصل از تحلیل CONTAINS ۶۴
شکل ۶۰- انتخاب تحلیل CONTOUR و انجام تنظیمات مورد نظر ۶۵
شکل ۶۱- انتخاب دکمه EXECUTE PROCESS اجرای تحلیل CONTOUR و نمایش خروجی حاصل از آن ۶۵
شکل ۶۲- فایل XML حاصل از تحلیل CONTAINS ۶۶
شکل ۶۳- انتخاب تحلیل GS:QUERY و تعیین لایه ورودی ۶۷
شکل ۶۴- وارد کردن QUERY مورد نظر ۶۸
شکل ۶۵- خروجی حاصل از تحلیل QUERY ۶۸
شکل ۶۶- سرویس WMS لایه رودخانه‌ها ۷۱
شکل ۶۷- سرویس WMS لایه روستاها ۷۲

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

شکل ۱۰۲- تعریف متد HANDLER به منظور انجام پردازش مورد نیاز..... ۹۹
شکل ۱۰۳- ذخیره سازی خروجی تحلیل بافر به صورت GML..... ۱۰۰
شکل ۱۰۴- ساختار کلاس OGR..... ۱۰۰
شکل ۱۰۵- افزودن فرایند SIMPLE_BUFFER به لیست فرایندهای فعال..... ۱۰۱
شکل ۱۰۶- پاسخ سرور به درخواست GETCAPABILITIES..... ۱۰۲
شکل ۱۰۷- پاسخ سرور به درخواست DESCRIBEPROCESS..... ۱۰۳
شکل ۱۰۸- پاسخ سرور به درخواست EXECUTE..... ۱۰۴
شکل ۱۰۹- تبدیل خروجی به متر با تغییر WGS84 به سیستم تصویر UTM..... ۱۰۴

فهرست جداول

جدول ۱- پیاده سازی‌های WPS ۱۴
جدول ۲- المان‌های SERVICE METADATA ۲۲
جدول ۳- EXECUTION & RESOURCE SETTINGS ۲۳
جدول ۴- فازهای اجرایی در ژئوسرور ۲۵
جدول ۵- عملگرهای استاندارد WPS ۲۹
جدول ۶- اطلاعات موجود در پاسخ DESCRIBEPROCESS ۳۱
جدول۷- PPIO ۳۲
جدول ۸- دسته بندی پردازش ها در ژئوسرور ۳۷
جدول ۹- تحلیل BUFFER ۴۶
جدول ۱۰- تحلیل INTERSECTION ۵۱
جدول ۱۱- تحلیل CLIP ۵۵
جدول ۱۲- تحلیل NEAREST ۵۸
جدول ۱۳- تحلیل WITHIN ۶۰
جدول ۱۴- تحلیل CONTAINS ۶۲
جدول ۱۵- تحلیل CONTOUR ۶۴
جدول ۱۶- تحلیل QUERY ۶۷

فصل اول

مفاهیم پایه تحلیل های مکانی

۱-۱- مقدمه

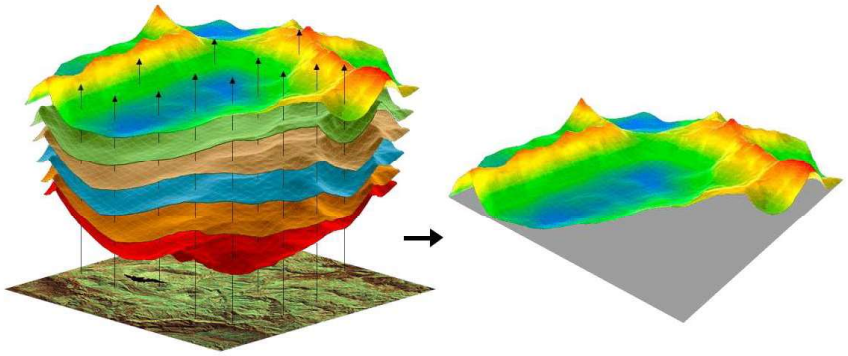
در دنیای امروز، دانش به‌عنوان بنیان اصلی پیشرفت شناخته می‌شود و توسعه پایدار مبتنی بر دانش، به‌عنوان هدف کلیدی بسیاری از کشورها از جمله ایران در نظر گرفته شده است. چنین دانشی با پردازش و تجزیه و تحلیل داده‌های با کیفیت و قابل اعتماد حاصل می‌گردد. تحقیق‌های علمی حاکی از آن است شکه بیش از ۷۰ درصد از داده‌هایی که سازمان‌ها برای تصمیم‌سازی و تصمیم‌گیری، مدیریت، برنامه‌ریزی و اجرا نیاز دارند، به‌طور مستقیم با مکان و ویژگی‌های مکانی مرتبط هستند. با توجه به پیچیدگی‌های موجود در پردازش این داده‌ها، نیاز به ابزارهایی برای تجزیه و تحلیل داده‌های مکانی در سطوح مختلف ضروری به نظر می‌رسد.

از زمان تأسیس OGC در سال ۱۹۹۴، استانداردسازی برای کشف، نمایش، تبادل و پردازش داده‌های مکانی در قالب سرویس‌های تحت وب به‌طور جدی پیگیری شده است. در این زمینه، سرویس‌های مبتنی بر Web Processing Service (WPS) یکی از ابزارهای اصلی برای پردازش داده‌های مکانی به‌صورت آنلاین و از راه دور هستند که می‌توانند در سیستم‌های اطلاعات مکانی (GIS) مورد استفاده قرار گیرند. این سرویس‌ها در راستای تسهیل تحلیل‌های مکانی و اشتراک‌گذاری پردازش‌های پیچیده در بستر وب، اهمیت زیادی دارند.

استاندارد WPS، راه‌حلی برای انجام پردازش‌های مکانی به‌طور آنلاین و در بستر وب فراهم می‌کند. این سرویس امکان اشتراک‌گذاری پردازش‌های پیچیده مکانی را فراهم آورده و به‌ویژه در پیاده‌سازی‌های سامانه‌های مرتبط با GIS در محیط‌های دیجیتال، اهمیت ویژه‌ای پیدا می‌کند.

۱-۲- تحلیل مکانی

تحلیل مکانی یکی از اجزای اساسی سامانه‌های اطلاعات مکانی محسوب می‌شود که داده‌های خام مکانی را به الگوهای معنادار تبدیل می‌کند. متخصصان این حوزه با بررسی موقعیت‌ها، ویژگی‌ها و روابط مکانی میان عوارض، قادر به شناسایی الگوها و بهبود فرآیند تصمیم‌گیری بوده و همچنین چالش‌های پیچیده دنیای واقعی را حل نمایند. (شکل ۱)



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

تحلیل مکانی شامل پردازش و تفسیر داده‌های مکانی به‌منظور استخراج اطلاعات کاربردی است. این تحلیل به پرسش‌هایی از جمله موارد زیر پاسخ می‌دهد:

- موقعیت مکانی عوارض خاص (مانند بیمارستان‌ها یا مناطق مستعد سیل)
- الگوهای موجود در داده‌های مکانی (مانند پراکندگی بیماری‌ها یا تراکم ترافیک)
- روابط متقابل میان عوارض (مانند تأثیر جنگل‌زدایی بر زیستگاه‌های حیات‌وحش)

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

- **پرس‌وجوهای توصیفی:** فیلتر نمودن داده‌ها بر اساس ویژگی‌های ذاتی مانند شناسایی شهرهای با جمعیت بیش از یک میلیون نفر
- **پرس‌وجوهای مکانی:** انتخاب عوارض بر مبنای موقعیت جغرافیایی مانند تعیین جاده‌های متأثر از سیلاب
- **تحلیل مجاورت:** ارزیابی عوارض در محدوده‌های فاصله‌ای مشخص مانند مدارس واقع در شعاع یک کیلومتری ایستگاه‌های پلیس
- **تحلیل همپوشانی:** تلفیق لایه‌های اطلاعاتی مختلف مانند ترکیب داده‌های خاک و بارش برای بهینه‌سازی کشت
- **تحلیل شبکه:** مدلسازی ارتباطات و مسیریابی مانند محاسبه کوتاه‌ترین مسیر برای توزیع کالا
- **زمین آمار:** شناسایی روندها و الگوهای مکانی مانند تعیین کانون‌های جرم‌خیز یا پیش‌بینی انتشار آلاینده‌ها

تحلیل مکانی در طیف وسیعی از صنایع، از برنامه‌ریزی شهری تا کشاورزی پیشرفته، به ابزاری ضروری برای تصمیم‌گیری‌های مبتنی بر شواهد تبدیل شده است. از جمله این موارد می‌توان به موارد زیر اشاره نمود:

- **برنامه‌ریزی شهری و توسعه شهرهای هوشمند**
  - مکان‌یابی بهینه خدمات شهری: انتخاب بهترین موقعیت برای احداث مدارس، بیمارستان‌ها و ایستگاه‌های حمل‌ونقل عمومی بر اساس معیارهای دسترسی و پراکندگی جمعیت
  - مدیریت هوشمند ترافیک: تحلیل الگوهای ترافیکی و بهینه‌سازی زمان‌بندی چراغ‌های راهنمایی
  - پهنه‌بندی کاربری اراضی: تعادل‌بخشی بین مناطق مسکونی، تجاری و فضای سبز با ملاحظه معیارهای توسعه پایدار
- **پایش و حفاظت محیط زیست**
  - پایش تغییرات پوشش جنگلی: ردیابی نرخ جنگل‌زدایی با پردازش تصاویر ماهواره‌ای چندزمانه
  - مدلسازی مخاطرات طبیعی: پیش‌بینی مناطق سیل‌گیر با تلفیق داده‌های توپوگرافی، هیدرولوژیکی و اقلیمی
  - مدیریت زیستگاه‌های حیات وحش: طراحی کریدورهای امن مهاجرت گونه‌ها با تحلیل الگوهای حرکتی جانوران
- **تحلیل‌های بازار و کسب‌وکار**
  - مکان‌یابی هوشمند مراکز فروش: تعیین موقعیت بهینه فروشگاه‌ها بر اساس تحلیل ترکیبی
  - بهینه‌سازی زنجیره تأمین: طراحی کارآمدترین مسیرهای توزیع برای کاهش هزینه‌های عملیاتی

- بازاریابی جغرافیایی: هدف‌گیری تبلیغاتی بر اساس الگوهای رفتاری مصرف‌کنندگان در مناطق مختلف

- **کشاورزی دقیق و هوشمند**
  - مدیریت نهاده‌های کشاورزی: تهیه نقشه‌های توصیه کود و آبیاری بر اساس تغییرپذیری مکانی ویژگی‌های خاک
  - پایش محصولات زراعی: پیش‌بینی عملکرد با استفاده از شاخص‌های گیاهی استخراج شده از تصاویر سنجش از دور
  - ارزیابی مخاطرات اقلیمی: تحلیل تأثیرات خشکسالی بر تولیدات کشاورزی و برنامه‌ریزی سازگاری

برخی از رایج‌ترین 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 قرارگیری 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- چالش‌های موجود در تحلیل‌های مکانی

تحلیل‌های مکانی به‌طور معمول شامل پردازش داده‌هایی هستند که از منابع مختلف، با مقیاس‌های زمانی و مکانی متفاوت، جمع‌آوری می‌شوند. این داده‌ها معمولاً شامل اطلاعات مکانی پیچیده‌ای مانند تصاویر ماهواره‌ای، مدل‌های ارتفاعی و داده‌های سنسور حاصل از محیط‌های واقعی هستند. پردازش این داده‌ها به‌ویژه در مقیاس‌های بزرگ می‌تواند چالش‌های متعددی به‌دنبال داشته باشد که در اینجا به برخی از مهم‌ترین آن‌ها اشاره می‌کنیم.

۱-۳-۱- حجم زیاد داده‌ها (Big Data)

یکی از بزرگ‌ترین چالش‌ها در تحلیل‌های مکانی، مدیریت و پردازش حجم بالای داده‌هاست. داده‌های مکانی معمولاً شامل اطلاعات با ابعاد زمانی و مکانی مختلف بوده که می‌تواند بسیار بزرگ و پیچیده باشند. به‌عنوان مثال، تصاویر ماهواره‌ای با وضوح بالا یا داده‌های حسگرهای محیطی می‌توانند 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 زمانیکه 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 سیستم‌های توزیع‌شده

با توجه به اینکه داده‌های مکانی معمولاً از منابع مختلف و در زمان‌های مختلف جمع‌آوری می‌شوند، نیاز به سیستم‌های توزیع‌شده برای پردازش داده‌ها وجود دارد. این سیستم‌ها باید توانایی پردازش داده‌ها را به‌صورت موازی و با استفاده از چندین سرور یا رایانه فراهم کنند. این چالش‌ها نیازمند معماری‌های پیچیده برای مدیریت و هماهنگی پردازش‌ها و ذخیره‌سازی داده‌ها هستند.

۱-۳-۷- پردازش و ذخیره‌سازی داده‌های سه‌بعدی و پیچیده

داده‌های سه‌بعدی، مانند مدل‌های شهری یا داده‌های مرتبط با ارتفاع، نیاز به پردازش‌هایی پیچیده دارند. این داده‌ها معمولاً حجم بالایی داشته و نیازمند الگوریتم‌هایی برای تجزیه و تحلیل دقیق و بهینه می‌باشند. علاوه بر این، جهت ذخیره‌سازی داده‌های سه‌بعدی، به زیرساخت‌هایی خاص برای حفظ و پردازش اطلاعات به‌صورت مؤثر و با کمترین هزینه نیاز است.

۱-۴- مقدمه‌ای بر تحلیل‌های مکانی دسکتاپ و برخط

تحلیل‌های مکانی همواره بخشی اساسی از سیستم‌های اطلاعات مکانی (GIS) بوده‌اند. در گذشته، این تحلیل‌ها عمدتاً در محیط‌های دسکتاپ و با استفاده از نرم‌افزارهایی مانند ArcGIS، QGIS و یا GRASS GIS انجام می‌شدند. در این روش، کاربران ملزم به نصب نرم‌افزارهای تخصصی بر روی سیستم‌های محلی خود بودند و انجام پردازش‌های پیچیده نیازمند سخت‌افزارهای قدرتمند و دانش فنی بالا بود. این رویکرد با محدودیت‌هایی همچون وابستگی به منابع محلی، دشواری در به‌روزرسانی الگوریتم‌ها و عدم امکان اشتراک‌گذاری ساده پردازش‌ها بین کاربران مختلف همراه بود.

با گسترش فناوری‌های مبتنی بر وب و نیاز روزافزون به دسترسی‌پذیری و تعامل‌پذیری در سرویس‌های مکانی، استانداردهای جدیدی توسط کنسرسیوم Open Geospatial Consortium (OGC) معرفی شدند. این تحول باعث شده که تحلیل‌های مکانی به‌راحتی از طریق اینترنت قابل دسترس بوده و امکان استفاده از سرورهای قدرتمند برای پردازش داده‌ها فراهم گردد. این روند نه تنها کارایی و دقت تحلیل‌ها را افزایش داده بلکه به تسهیل همکاری‌های بین‌سازمانی و اشتراک‌گذاری داده‌های مکانی کمک کرده است.

یکی از این استانداردهای کلیدی، Web Processing Service (WPS) است که امکان اجرای پردازش‌های مکانی را به‌صورت برخط فراهم می‌کند. این استاندارد، با تعریف یک چارچوب مناسب برای توصیف، اجرا و مدیریت پردازش‌های مکانی، تحولی اساسی در نحوه انجام تحلیل‌های مکانی ایجاد کرده است.

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

استفاده از سیستم‌های دسکتاپ و برخط برای تحلیل‌های مکانی هرکدام 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- تحلیل‌های مکانی برخط

مزایا:

- سیستم‌های برخط امکان دسترسی به داده‌ها و پردازش‌ها را از هر نقطه‌ای فراهم می‌آورند. کاربران می‌توانند بدون نیاز به نصب نرم‌افزارهای تخصصی، از خدمات پردازش مکانی برخط استفاده نموده که این امر موجب افزایش انعطاف‌پذیری می‌شود.
- سیستم‌های برخط به‌ویژه با استفاده از رایانش ابری و پردازش توزیع‌شده، قادرند داده‌های بزرگ و پیچیده را به‌صورت مقیاس‌پذیر پردازش کنند. این امر کاربران را قادر می‌سازد بدون نیاز به منابع سخت‌افزاری خاص، پردازش‌های پیچیده را با سرعت و کارایی بالا انجام دهند.
- یکی از مزایای مهم سیستم‌های برخط این است که امکان اشتراک‌گذاری داده‌ها و نتایج تحلیل‌ها را به‌صورت آنلاین فراهم می‌کنند. این امر به تیم‌ها و سازمان‌ها اجازه می‌دهد تا به‌طور همزمان و به‌صورت مؤثر با یکدیگر همکاری نموده و به داده‌ها دسترسی داشته باشند.
- در سیستم‌های برخط، به‌روزرسانی نرم‌افزارها و داده‌ها به‌صورت متمرکز و از طریق سرورهای مرکزی انجام می‌شود. این امر موجب می‌شود فرآیند نگهداری سیستم ساده‌تر شده و به‌روزرسانی‌ها به‌طور خودکار در دسترس کاربران قرار گیرد.

معایب:

- یکی از معایب اصلی سیستم‌های برخط این است که برای دسترسی به خدمات و انجام پردازش‌ها، 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.
- داده‌ها و پردازش‌ها در سیستم‌های برخط معمولاً در سرورهای از راه دور ذخیره می‌شوند. این موضوع می‌تواند مشکلات امنیتی و حریم خصوصی ایجاد کند، به‌ویژه در صورتی که داده‌های حساس و شخصی در میان باشد. اطمینان از حفاظت از داده‌ها در برابر تهدیدات امنیتی در سیستم‌های برخط می‌تواند چالش‌برانگیز باشد.
- در سیستم‌های برخط، کنترل داده‌ها به‌طور عمده در دست ارائه‌دهندگان سرویس قرار دارد. این موضوع ممکن است برای سازمان‌هایی که نیاز به کنترل دقیق‌تر و امنیت بیشتر دارند، مشکل‌آفرین باشد.

۱-۶- بررسی چالش‌های فنی و زیرساختی در پیاده‌سازی WPS

پیاده‌سازی استاندارد WPS می‌تواند مزایای زیادی از جمله دسترسی آسان، مقیاس‌پذیر، و امکان اشتراک‌گذاری پردازش‌ها را به‌صورت آنلاین داشته باشد. با این حال، چالش‌های 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 به‌صورت استاندارد تعریف شده و ورودی‌ها و خروجی‌ها به‌طور 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 به‌طور طبیعی برای پردازش داده‌های مکانی تحت وب طراحی شده است، این سرویس‌ها در مقیاس‌های بزرگ، ممکن است با مشکلات مقیاس‌پذیری مواجه شوند به‌ویژه زمانی که داده‌ها حجیم یا پیچیده باشند. برای پردازش‌های پیچیده و داده‌های بزرگ، نیاز به زیرساخت‌های محاسباتی قدرتمند و توزیع‌شده 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 سه‌بعدی) 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، به‌ویژه 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 به‌طور مؤثر مدیریت شده و سیستم به‌درستی 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 به‌صورت مؤثر انجام دهد. به‌ویژه 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 ذخیره‌سازی داده‌های بزرگ (مانند تصاویر ماهواره‌ای و مدل‌های سه‌بعدی) و مدیریت کارآمد آن‌ها نیز چالشی زیرساختی است.

- استاندارد WPS به‌عنوان یکی از استانداردهای OGC نیاز به پیاده‌سازی دقیق پروتکل‌ها و فرمت‌های باز دارد و ارتباطات میان سیستم‌های مختلف را تسهیل می‌کند. جهت پیاده‌سازی موفق این استاندارد و نیز تبادل داده‌ها و عملیات در بستر وب، لازم است زیرساخت‌های فناوری اطلاعات از استانداردهایی مانند SOAP، WSDL و XML پشتیبانی کنند. هرگونه ناسازگاری با این استانداردها می‌تواند منجر به بروز مشکلاتی در تعاملات بین سرویس‌ها و پردازش‌ها شود.
- یکی از چالش‌های اساسی در پیاده‌سازی WPS بر بستر وب، مسائل امنیتی و حفظ حریم خصوصی است. داده‌های با ماهیت مکانی ممکن است حساسیت زیادی داشته باشند، به‌ویژه زمانی که شامل اطلاعات خصوصی مانند مالکیت املاک یا داده‌های محیطی خاص باشد. اطمینان از امنیت داده‌ها، به‌ویژه 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 ویژگی‌های مهم سیستم‌های برخط، مقیاس‌پذیری است تا در شرایطی که حجم داده‌ها افزایش می‌یابد، عملکرد خود را حفظ کنند. این امر به‌ویژه در پردازش‌های پیچیده مکانی که نیازمند منابع بالای پردازشی هستند، اهمیت پیدا می‌کند. برای این کار، ممکن است نیاز به زیرساخت‌های توزیع‌شده‌ای مانند رایانش ابری یا سرورهای مجازی باشد که بتوانند منابع محاسباتی را به‌طور مقیاس‌پذیر تخصیص داده و از هرگونه خرابی یا کاهش عملکرد جلوگیری کنند.
- یکی دیگر از چالش‌های زیرساختی در پیاده‌سازی WPS، مدیریت تعداد درخواست‌های همزمان از کاربران مختلف است. در مواردی که تعداد درخواست‌های همزمان به‌طور 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 فناوری‌هایی از قبیل کش (caching) یا توزیع بار (load balancing) است تا پردازش‌ها به‌طور مؤثر تقسیم شده و سیستم بتواند به‌صورت پایدار به درخواست‌ها پاسخ دهد.

۱-۷- مجموعه استاندارد OGC در داده‌های مکانی

OGC (Open Geospatial Consortium) یک سازمان بین‌المللی است که استانداردهای مختلفی را برای تبادل، پردازش و نمایش داده‌های مکانی تعریف کرده است. این استانداردها در زمینه‌هایی مانند سیستم‌های اطلاعات مکانی (GIS)، پردازش‌های مکانی تحت وب، و وب‌سرویس‌ها مورد استفاده قرار می‌گیرند. در ادامه، به توضیح برخی از مهم‌ترین استانداردهای OGC و ارتباط آن‌ها با WPS می‌پردازیم.

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 به‌صورت تصاویر در فرمت‌های مختلف مانند PNG یا JPEG نمایش دهند.

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

Web Feature Service (WFS) -۲-۷-۱

یکی دیگر از استانداردهای پرکاربرد 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.

به‌طور 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 یک 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) به‌صورت 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.

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

استاندارد CSW یکی از استانداردهای OGC است که به‌صورت تحت وب برای جستجوی فراداده‌های مکانی، خدمات و منابع مرتبط استفاده می‌شود. این سرویس به عنوان یک کاتالوگ قابل جستجو (مانند یک "کتابخانه مکانی") عمل می‌کند و به کاربران و برنامه‌ها کمک می‌کند تا مجموعه داده‌های مرتبط (مانند تصاویر ماهواره‌ای، نقشه‌ها یا خدمات WMS/WFS) را بر اساس معیارهای فراداده به صورت تحت وب جستجو نمایند.

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

استاندارد SOS برای جمع‌آوری، مدیریت و ارائه داده‌های سنسوری محیطی مانند دما، رطوبت و فشار هوا طراحی شده است. این سرویس که زیرمجموعه استانداردهای OGC است به کاربران اجازه می‌دهد تا انواع داده‌های محیطی مختلف را درخواست کنند. استاندارد WPS می‌تواند داده‌های SOS را برای انجام پردازش‌های پیچیده مانند پیش‌بینی وضعیت هوا یا شبیه‌سازی تغییرات اقلیمی استفاده کند و تأثیرات این شرایط را بر اکوسیستم‌ها تحلیل کند.

Web Map Tile Service (WMTS) -۷-۷-۱

استاندارد WMTS یکی از استانداردهای OGC برای ارائه نقشه‌های بخش‌بندی‌شده تحت وب است. این سرویس با تقسیم نقشه به قطعات کوچک و قابل بارگذاری سریع، نمایش روان داده‌های مکانی را در وب و موبایل ممکن می‌سازد. WMTS به‌ویژه برای نمایش نقشه‌های پایه با عملکرد بالا مانند سامانه‌های ناوبری، نقشه‌های تعاملی و داشبوردهای مکانی مناسب است و امکان مشاهده نقشه در مقیاس‌های مختلف را بدون نیاز به بارگذاری کامل هر بار فراهم می‌کند.

۱-۸- پیاده‌سازی‌های مختلف WPS و ارزیابی آن ها

برای راه‌اندازی یک فرآیند پردازشی تحت وب مطابق با استانداردهای OGC، استفاده از استاندارد WPS ضروری است. این استاندارد امکان اجرای پردازش‌های مکانی-زمانی را به‌صورت تحت وب فراهم می‌کند. پیاده‌سازی‌های مختلفی از WPS وجود دارند که هر یک دارای ویژگی‌ها، مزایا و محدودیت‌های خاص خود هستند. انتخاب مناسب‌ترین گزینه از میان آن‌ها به عوامل کلیدی زیر بستگی دارد:

- نسخه‌ی استاندارد WPS مورد پشتیبانی
- ویژگی‌های فنی (مانند پردازش موازی، پشتیبانی از فرمت‌های مختلف داده)
- زیرساخت اجرایی (سیستم‌عامل، زبان برنامه‌نویسی، کتابخانه‌های وابسته)
- مقیاس‌پذیری و انعطاف‌پذیری

در جدول ۱ به بررسی مختصری از این پیاده‌سازی‌ها پرداخته شده است:

جدول ۱- پیاده سازی‌های WPS

پیاده‌سازی WPS	زبان برنامه‌نویسی	ویژگی‌ها	مزایا	معایب
52north	Java	- استفاده از Java Framework - پشتیبانی کامل از پروتکل‌های OGC (WPS, WMS, WFS) - مناسب پردازش‌های پیچیده و حجیم	- مقیاس‌پذیری بالا - پشتیبانی کامل از استانداردهای OGC - سادگی در پیاده‌سازی برای توسعه‌دهندگان	- نیاز به منابع محاسباتی بالا - پیچیدگی پیکربندی در پردازش‌های بسیار پیچیده
ZOO	Python, C, JavaScript	- معماری مدولار - پشتیبانی از چندین زبان برنامه‌نویسی - پشتیبانی از پردازش‌های موازی و توزیع‌شده - پیاده سازی استاندارد WPS 2.0.0	- مدولار و انعطاف‌پذیر - پشتیبانی از زبان‌های مختلف - مناسب برای سیستم‌های توزیع‌شده و مقیاس‌پذیر	- پیچیدگی در پیکربندی و نگهداری - نیاز به زیرساخت‌های قدرتمند برای پردازش‌های توزیع‌شده - نیاز به دانش فنی زیاد
ArcGIS Server esri	Python, JavaScript	- رابط گرافیکی کاربرپسند - بهینه‌شده برای پردازش‌های ArcGIS - پشتیبانی از فرمت‌های Esri ، GeoJSON - پشتیبانی از ArcGIS ModelBuilder، REST API و سرویس‌های پردازش مکانی	- یکپارچگی عالی با اکوسیستم ArcGIS - پشتیبانی قوی از تحلیل‌های مکانی	- محدودیت در سفارشی‌سازی خارج از ArcGIS - هزینه‌های لایسنس
deegree	Java	- پشتیبانی کامل از پروتکل‌های OGC (WPS, WMS, WFS) - یکپارچگی داده‌های مکانی از منابع مختلف - مناسب برای پروژه‌های مقیاس بزرگ	- پشتیبانی کامل از استانداردهای OGC - مقیاس‌پذیری بالا - مناسب برای سازمان‌ها و پروژه‌های بزرگ	- پیچیدگی در پیاده‌سازی - نیاز به منابع محاسباتی زیاد - نامناسب برای پروژه‌های کوچک
GeoServer	Java	- پشتیبانی کامل از پروتکل‌های OGC (WPS, WMS, WFS) - مناسب برای پروژه‌های کوچک تا متوسط - رابط گرافیکی کاربرپسند	- پشتیبانی از فرمت‌های مختلف داده - رابط کاربری ساده - پشتیبانی از پردازش‌های پیچیده	- کارایی محدود در پردازش‌های بسیار پیچیده - نیاز به پیکربندی اضافی برای مقیاس‌های بزرگ
PyWPS	Python	- پیاده‌سازی مبتنی بر Python - قابلیت گسترش و سفارشی‌سازی بالا - پشتیبانی از پردازش‌های پیچیده - ساده و سریع در توسعه فرآیندهای WPS	- سادگی در توسعه - مناسب برای پروژه‌های مقیاس کوچک و متوسط -پشتیبانی از پردازش‌های پیچیده در Python	-کارایی کمتر نسبت به دیگر پیاده‌سازی‌ها در پردازش‌های بسیار بزرگ - نیاز به دانش در برنامه‌نویسی Python

۱-۹- نیازمندی‌ها و مراحل پیاده‌سازی تحلیل مکانی تحت وب

پیاده‌سازی پردازش‌های مکانی آنلاین در محیط ژئوپورتال با استفاده از استاندارد WPS، نیازمند توجه به مجموعه‌ای از نیازمندی‌ها و مراحل خاص است. در ادامه به معرفی برخی از این موارد پرداخته شده است.

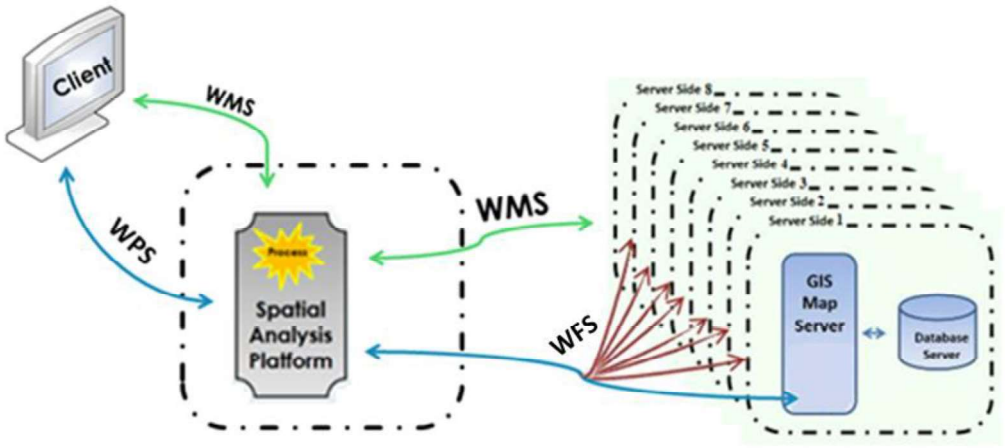
- زیرساخت سخت‌افزاری و محاسباتی:
  - ✓ تأمین سرورهای با قدرت پردازشی مناسب جهت انجام تحلیل‌های مکانی پیچیده به‌صورت آنلاین
  - ✓ پشتیبانی از پردازش موازی برای عملیات سنگین محاسباتی
  - ✓ امکان مدیریت داده‌های حجیم مکانی با کارایی مطلوب
- استانداردهای داده‌ای و فرمت‌های پشتیبانی:
  - ✓ استفاده از استاندارد WMS برای نمایش لایه‌ها
  - ✓ استفاده از استاندارد WFS برای انتقال داده‌های برداری
  - ✓ استفاده از استاندارد WCS برای داده‌های رستری
  - ✓ استفاده از استاندارد GML به‌عنوان فرمت استاندارد تبادل داده‌های مکانی

پیکربندی سرور پیشنهادی به منظور راه‌اندازی یک سامانه تحلیل مکانی تحت وب به صورت زیر می‌باشد:

- RAM ۱۶ گیگابایت
- CPU ۸ هسته‌ای
- SSD ۱۰۰ گیگابایت

عموماً این‌گونه فرض می‌شود که چون با استفاده از سرویس WMS امکان انجام تحلیل مکانی وجود ندارد، بنابراین تنها روش برای اجرای تحلیل مکانی، نیازمند دسترسی به اصل داده یا دسترسی به سرویس‌های WFS و WCS می‌باشد. این فرض، با رویکرد پیاده‌سازی تحلیل‌های مکانی در زیرساخت داده‌های مکانی (SDI) مغایرت داشته و در واقع نوعی بازگشت از تعامل مبتنی بر «سرویس‌های تحلیل مکانی» به تعامل مبتنی بر «اصل داده‌های مکانی» محسوب می‌شود.

شکل شماره ۲، نمودار استفاده از سرویس‌های WFS و WCS لایه‌های اطلاعاتی متولیان را به‌منظور انجام تحلیل مکانی در سمت سامانه‌های پردازشی نشان می‌دهد. در این رویکرد، تمامی لایه‌های دخیل لازم است از طریق سرویس‌های WFS و WCS متولیان داده به این سامانه‌های منتقل شده و سپس فرایند تحلیل مکانی بر روی آن‌ها اجرا شود.



شکل ۲- استفاده از سرویس‌های مکانی WFS یا WCS در تحلیل مکانی

چنین ساختاری معایبی را به همراه دارد، که از آن جمله می‌توان به موارد زیر اشاره کرد:

- سرویس WFS به‌دلیل دارا بودن مختصات مکانی و اطلاعات توصیفی عوارض، عملکرد نسبتا کندی دارد. از جمله مهم‌ترین عوامل مؤثر در این زمینه، می توان به موارد ذیل اشاره کرد:
  - حجم داده: هر چه داده‌های مکانی و توصیفی بیشتر باشد، Query 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.
  - تأخیر شبکه[۲] : سرعت اتصال شبکه بین سرور متولیان داده و سرور ملی همچنین می تواند بر سرعت انتقال داده تأثیرگذار بوده باشد.
- در دسترس قرار دادن سرویس WFS یا WCS سبب ایجاد یک سری 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.

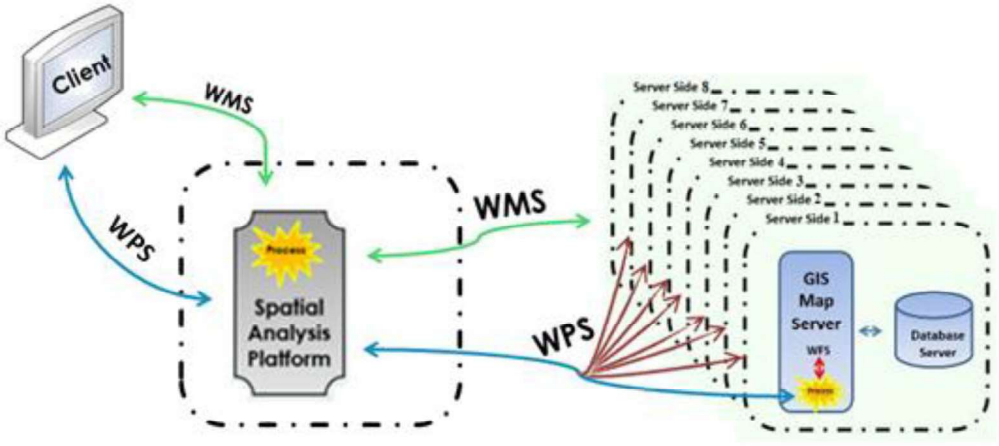
تذکر: چنانچه بعضی از لایه‌های مکانی مورد نظر به دلیل پایین بودن حجم داده و یا تمایل متولی مربوطه در اشتراک‌گذاری آن باعث عدم بروز مشکلات فوق‌الذکر گردد، می‌توان از سرویس‌های WFS یا WCS جهت انجام تحلیل‌های مکانی استفاده نمود.

گزینه جایگزین به منظور رفع و کاستن از مشکلات مطرح شده، استفاده از سرویس WPS و پردازش داده در سمت سرور متولیان داده و سپس ارسال نتایج به ژئوپرتال می‌باشد (شکل ۳).

---

[۱] Server Load

[۲] Netwrok Latency



شکل ۳ – تحلیل مکانی تحت وب بدون استفاده از سرویس‌های مکانی WFS یا WCS

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 یک سری معیارها، خروجی نهایی را ایجاد می‌کند.

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

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

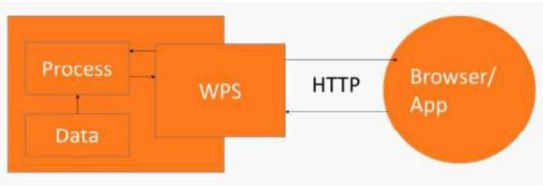
از جمله مباحث مهم دیگر، وجود پیاده‌سازی پایتون برای کتابخانه‌هایی نظیر OGR/GDAL (انجام تحلیل‌های مکانی) و PROJ (تبدیل سیستم‌های مختصات) می‌باشد. به همین دلایل، پیاده سازی PyWPS از این بین انتخاب شده است که در ادامه به طور مبسوط به توضیح و تشریح و مثال‌هایی در این رابطه پرداخته خواهد شد. در فصول آتی این دستورالعمل، 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).



شکل ۴- استاندارد WPS

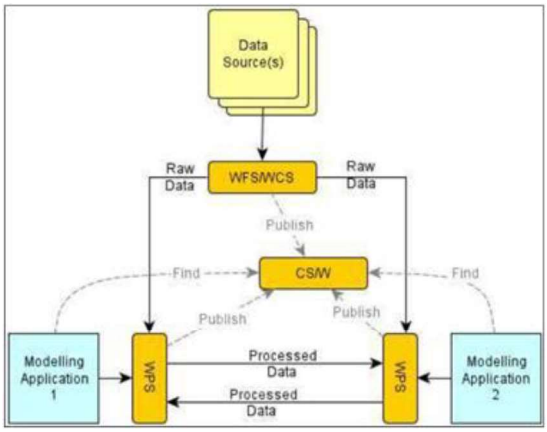
یکی از مزایای اصلی WPS، استقلال آن از فناوری‌های سمت سرور است. این استاندارد با پنهان کردن جزئیات پیاده‌سازی، امکان فراخوانی سرویس‌های پردازشی را روی هر پلتفرمی فراهم می‌کند. به‌علاوه، انعطاف‌پذیری بالایی در پشتیبانی از انواع داده‌های مکانی (برداری و رستری) و پردازش‌های مختلف - از تعیین حریم رودخانه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



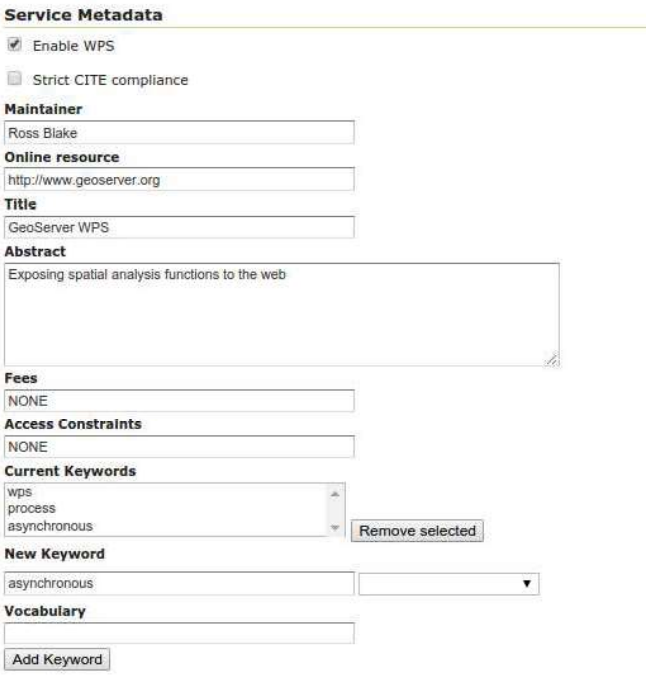
شکل ۵- دیاگرام سرویس‌های WFS، CSW و WPS

۲-۲- 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 می باشد. این محدودیت برای پردازش ورودی‌ها اعمال می شود، بنابراین حتی پردازش‌هایی که سبب خلاصه سازی داده میشوند و نتایج کمی را برمیگردانند در صورت اعمال بر داده‌های زیاد، تحت تأثیر 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.

۲-۲-۱- Service Metadata

بخش Service Metadata در بین تمامی سرویس‌ها (WMS, WFS, WCS و WPS) مشترک می‌باشد. این بخش از یک سری تعاریف فراداده‌ای رایج برای توصیف سرویس مدنظر و نشان دادن محدودیت‌ها استفاده می‌کند.



شکل ۶– Service Metadata

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

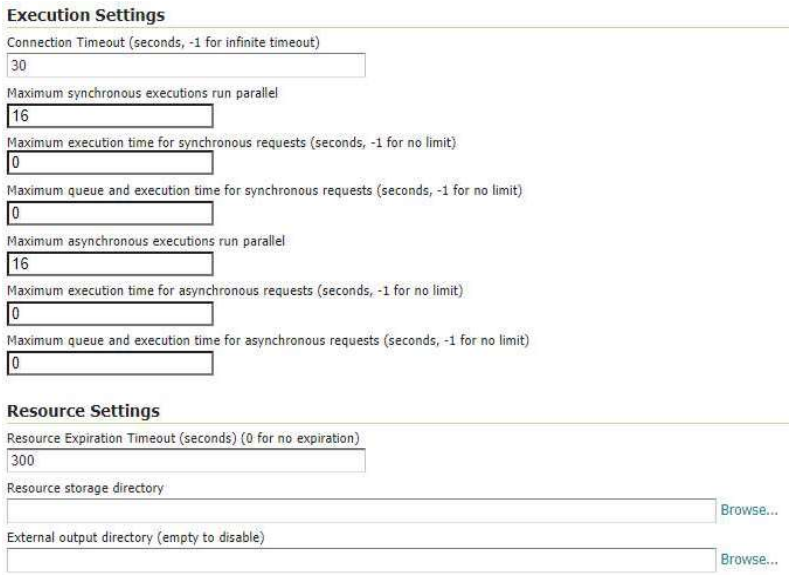
جدول ۲– المان‌های Service Metadata

فیلد	شرح مختصر
Enabled WPS	فعال بودن یا نبودن سرویس WPS را مشخص میکند. در صورت غیرفعال بودن درخواست‌های سرویس مذبور پردازش نخواهند شد.
Strict CITE compliance	در صورت انتخاب، انطباق کامل با CITE[۱] را تحمیل می کند. 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.
Abstract	یک توضیحی با اطلاعات بیشتر در مورد سرویس فراهم میکند.

۱ Compliance Interoperability Test Engine (CITE)

نشانگر هزینه‌ی اعمال شده توسط فراهم کننده‌ی سرویس به منظور استفاده از سرویس می‌باشد. کلمه‌ی کلیدی NONE کلمه‌ی رزرو شده به معنی بدون هزینه بوده که در بیشتر حالات استفاده می‌گردد.	***Fees***
قیودی که توسط فراهم کننده‌ی سرویس اعمال می‌شود را توصیف می‌کند. کلمه‌ی کلیدی NONE کلمه‌ی رزرو شده می باشد که نشانگر عدم اعمال هیچگونه قیود دسترسی بوده و در بیشتر حالات استفاده می‌گردد.	***Access Constraints***
لیستی از لغاتی که مرتبط با سرویس بوده و به منظور بهبود جستجو و فهرست نویسی استفاده می‌شود.	***Keywords***

۲-۲-۲- Execution and resource settings



شکل ۷- Execution & Resource Settings

المان‌های شکل ۷ در جدول ۳ شرح مختصری داده می شوند.

جدول ۳- Execution & Resource Settings

شرح مختصر	فیلد
Excecution settings	
ثانیه‌هایی که سرویس WPS منتظر می‌ماند قبل از اینکه بخواهد ارتباط مبتنی بر HTTP از راه دور را متوقف کند.	***Connection timeout***
حداکثر تعداد پردازش‌های همزمان که به طور موازی در یک زمان مشخص اجرا می‌شوند.	***Maximum synchronous executions run parallel***
حداکثر زمانی که یک پردازش همزمان مجاز به اجرا می باشد. پردازش‌هایی که در حالت همزمان اجرا می‌شوند باید در زمان مجاز تمام گردند. در غیر این صورت به طور اتوماتیک کنار گذاشته می‌شوند. این درخواستها بدلیل اینکه منجر به انتظار کلاینت برای پاسخ از طریق اتصال HTTP می‌باشند، باید زمان‌های نسبتاً کوتاهی انتخاب گردد (به طور مثال ۶۰ ثانیه).	***Maximum execution time for synchronous requests***
حداکثر زمانی که یک پردازش مجاز به ماندن در نوبت و اجرایش می باشد. پردازش‌هایی که در حالت همزمان اجرا می‌شوند باید در زمان مجاز تمام گردند. در غیر این صورت به طور	***Maximum queue and execution time for synchronous requests***

	اتوماتیک کنار گذاشته می‌شوند. این درخواست‌ها بدلیل اینکه منجر به انتظار کلاینت برای پاسخ از طریق اتصال HTTP می‌باشند، باید زمان‌های نسبتا کوتاهی انتخاب گردد (به طور مثال ۶۰ ثانیه).
Maximum asynchronous executions run parallel	حداکثر تعداد پردازشهای 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. پردازش‌هایی که در حالت غیرهمزمان اجرا می‌شوند باید در زمان محدود تعین شده به اتمام برسند در غیر این صورت به طور اتوماتیک کنار گذاشته می‌شوند.
Maximum queue and execution time for asynchronous requests	حداکثر زمانی که یک پردازش غیرهمزمان مجاز به ماندن در نوبت و اجرایش می‌باشد. پردازش‌هایی که در حالت غیر همزمان اجرا می‌شوند باید در زمان محدود مجاز تمام گردند. در غیر این صورت به طور اتوماتیک کنار گذاشته می‌شوند.
Resource settings	
Resource expiration timeout	تعداد ثانیه‌هایی که نتیجه‌ی اجرای غیرهمزمان برای کاربر به منظور بازیابی در دسترس خواهد بود. به محض اتمام این زمان، این منابع مستعد پاکسازی می‌گردند (که در بازه‌های زمانی مشخص انجام می‌گیرد).
Resource storage directory	مکانی روی دیسک که ورودی‌ها، منابع خروجی و موقت مرتبط با یک پردازش مشخص نگهداری می‌شوند. مکان پیش فرض temp/wps در data directory خود ژئوسرور می‌باشد.
External output directory	برخی پردازش‌ها اجازه‌ی ذخیره‌سازی خروجی‌ها را در یک دایرکتوری خارجی نیز می‌دهند.

۲-۲-۳- Process status page

علاوه بر موارد مذکور، صفحه‌ی وضعیت پردازش در بخش About and Status موجود می‌باشد که در رابطه با پردازش‌های در حال اجرا و مواردی که اخیراً به پایان رسیده‌اند گزارش می‌دهد (شکل ۸).



شکل ۸- Process Status

جدول حاصله حاوی یک سری اطلاعات می‌باشد، که ستون‌های آن به شرح زیر می‌باشند:

- S/A: اجرا به صورت همزمان یا غیرهمزمان (synchronous or asynchronous)
- Node: نام ماشین یا سیستمی که پردازش را اجرا می‌کند.
- User: کاربری که فرایند اجرا را آغاز کرده است.
- Process name: پردازش اصلی که در حال اجرا می‌باشد. (chain process در این جدول نشان داده نخواهد شد.)

- Created: زمانی که پردازش ایجاد شده است.
- Phase: وضعیت موجود در چرخه‌ی عمر پروژه
- Progress: پیشرفت فعلی
- Task: در طی فرآیند پردازش، در حال حاضر مشغول انجام چه کاری می‌باشد.

فازهای اجرایی در ژئوسرور می‌تواند 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 می‌باشد.
RUNNING	پردازش یا در حال بازیابی و تجزیه کردن[1] ورودی‌ها برای محاسبه و یا نوشتن نتایج می‌باشد.
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 پردازش‌های همزمان را نیز می‌دهد. زمانی که نادیده گرفتن پردازش به اتمام رسید، اجرای پردازش از جدول فوق ناپدید می‌شود.

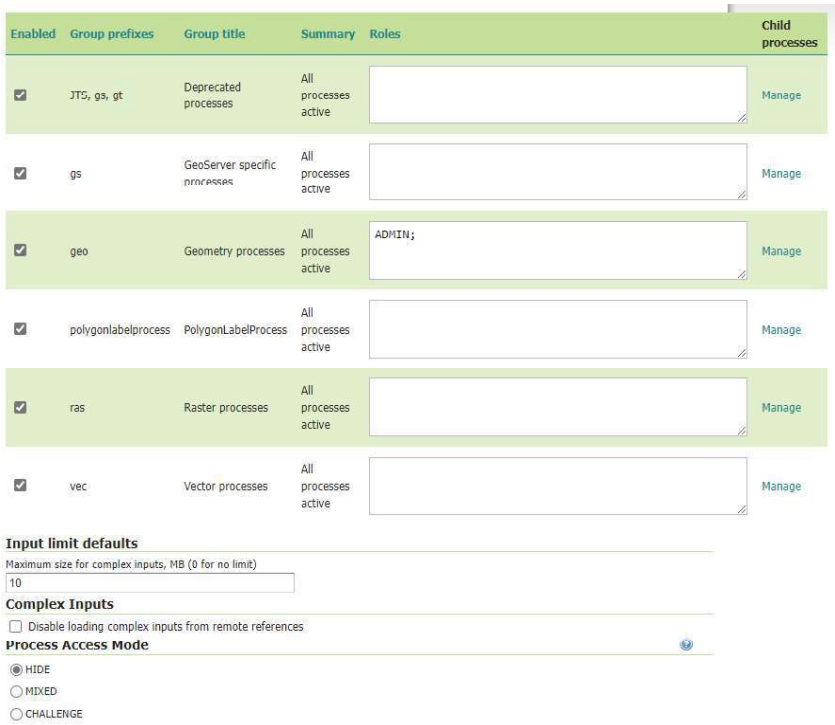
لازم به ذکر است پردازش‌های اتمام یافته نیز می توانند نادیده گرفته شوند. این کار منجر به حذف آنی تمام منابع مرتبط با پردازش بر روی دیسک می‌گردد.

۲-۲-۴- WPS security page

امنیت سرویس ژئوسرور به طور معمول مبتنی بر تنظیمات کلی امنیت OGC می باشد. اما در مورد WPS نیاز به محدود کردن دسترسی به تک تک پردازش‌ها نیز احساس می‌شود. به همان صورتی که امنیت داده، دسترسی به لایه‌ها را محدود می‌کند.

---

[1] parsing



شکل ۹– WPS Security

تنظیمات امنیتی WPS می‌تواند در بخش Security و سپس WPS security انجام شود (شکل ۹).

در حقیقت، لیست role هایی که به یک پروسه یا مجموعه‌ای از پروسه‌ها ضمیمه می‌شود مشخص کننده‌ی آن است که کدام کاربر می‌تواند به کدام پردازش/ مجموعه پردازش‌ها دسترسی داشته باشد. در صورتی که لیست role ها خالی باشد نشان‌دهنده 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 هیچ‌کس 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 شده‌اند. اطلاعات وارد شده در این ستون همچنین قابلیت تکمیل خودکار و Copy و Paste لیست role ها را از یک پردازش به دیگری فراهم می‌کند (شکل ۱۰).



شکل ۱۰- Roles

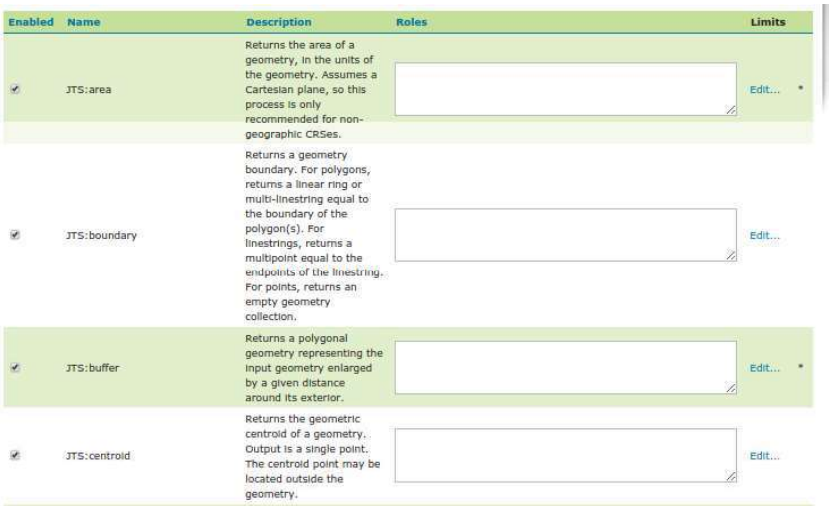
در ادامه، سه بخش انتهایی Input limit defaults، Complex Inputs و Process Access Mode در صفحه WPS security توضیح داده می‌شوند.

❖ Input limit defaults

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

- حداکثر حجم ورودی‌های Complex
- محدوده‌ی مقادیر قابل قبول برای مقادیر عددی
- حداکثر تعدد ورودی‌های تکرارپذیر

حداکثر حجم می‌تواند در صفحه‌ی WPS Security page یک مقدار کلی پیش فرض داده شود. همچنین تعریف محدودیت به ازای هر پردازش در قسمت Limits ممکن می‌باشد: در هر گروه وارد manage شده و به ازای پردازش مدنظر در ستون Limits و اعمال محدودیت مدنظر می‌شود.



شکل ۱۱- WPS Security : manage

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



شکل ۱۲– نمونه ای از محدودیت ها

این محدودیت می‌تواند بر تمامی input های یک پردازش مشخص که قابلیت آن را داشته باشند اعمال گردد. (شکل ۱۲)

❖ Complex Inputs

به صورت پیش‌فرض، 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 Excecute 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 نحوه‌ی 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 capabalities برای کاربر فعلی 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 capabalities 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 ساز و کار عمومی، به منظور توصیف و فعال‌سازی فرآیندهای مکانی تحت وب، توصیف داده‌های مورد نیاز جهت اطلاعات ورودی و خروجی یک فرآیند، مشخص می‌شود. این داده‌ها می‌توانند از طریق شبکه و سرور قابل دسترس باشند. داده‌ها می‌تواند شامل فرمت‌های داده رستری مانند GeoTIFF یا استانداردهای تبادل داده مانند GML باشد. ورودی‌های داده می‌توانند به‌صورت استاندارد با سرویس‌های استاندارد OGC در ارتباط باشند. به عنوان مثال، یک ورودی داده برای عملیات تقاطع می‌تواند یک پلیگون باشد که در پاسخ به درخواست WFS به‌دست می‌آید، در این صورت ورودی داده WPS از پرسش(Query) حاصل از WFS خواهد بود.

۲-۴- نمای کلی WPS

فعال کردن پردازش مکانی در اینترنت مستلزم توسعه طیف گسترده‌ای از خدمات وب برای پشتیبانی از عملیات مکانی تجزیه‌ناپذیر (اتمیک) و همچنین قابلیت‌های مدلسازی پیچیده است. استاندارد کردن روش فراخوانی این فرآیندها به منظور کاهش میزان برنامه‌نویسی مورد نیاز و تسهیل در اجرا و استفاده از سرویس‌های جدید اهمیت زیادی دارد. سرویس WPS برای دستیابی به این اهداف در نظر گرفته شده است.

۲-۵- توابع WPS

استاندارد WPS نسخه‌ی ۱,۰,۰ سه عملگر را برای بررسی و اجرای پردازش‌های مکانی تعریف می‌کند، به نحوی که ژئوسرور قابلیت‌های این سه عملگر را با دو شبه‌عملگر[۱] گسترش می‌دهد. 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 عملیات‌ها عبارتند از:

جدول ۵- عملگرهای استاندارد WPS

عملیات	توضیحات
GetCapabilities	این عملگر یک سند XML برمیگرداند که قابلیتهای WPS، تحلیلها و نحوه فراخوانی آنها را بیان می کند.

[۱] pseudo-operator

DescribeProcess	این عملگر نیز یک سند XML برمیگرداند که تحلیل مورد نظر را با جزییات توصیف میکند و شامل جزییاتی به منظور فراهم اوردن ورودیهای مناسب و فهم پاسخ می باشد.
Execute	عملگری است که کلاینت در زمان نیاز به فراخوانی یک تحلیل خاص صدا می‌زند.
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.

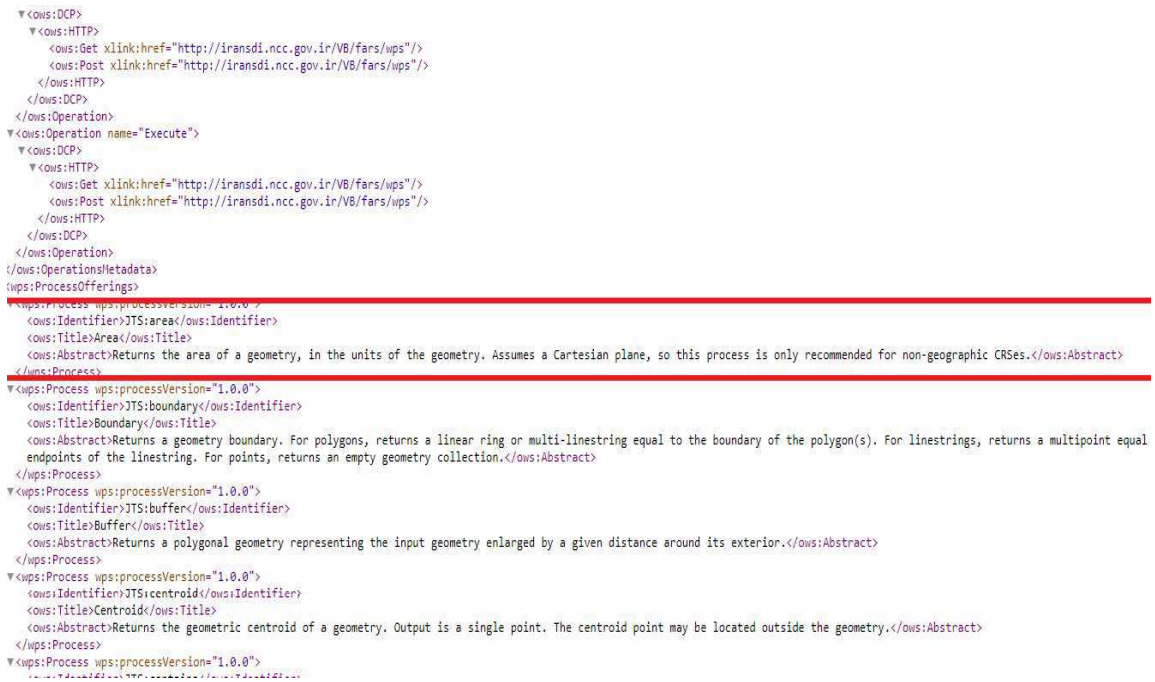
۲-۵-۱- GetCapabilities

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 پشتیبانی‌شده را بازمی‌گرداند.

نمونه‌ای از درخواست GetCapabilities که از طریق روش HTTP Get ارسال می شود، به صورت ذیل می باشد.

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

پاسخ دریافتی به صورت شکل ۱۳ قابل مشاهده می‌باشد.



شکل ۱۳- پاسخ درخواست GetCapability

همانطور که در شکل ۱۳ نشان داده شده است خلاصه اطلاعاتی در مورد سرویس‌های تحلیلی پیاده‌سازی شده در یک سرور را نمایش می‌دهد.

DescribeProcess -۲-۵-۲

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 می‌تواند 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 می‌باشند.

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” *(geometry, mandatory)*
	distance: “The distance (same unit of measure as the geometry)” *(double, mandatory)*
	quadrant segments: “Number of quadrant segments. Use > 0 for round joins, 0 for flat joins, < 0 for mitred joins” *(integer, optional)*
	capstyle: “The buffer cap style, round, flat, square” *(literal value, optional)*
Out	One of the GeoServer supported formats for Geometry encoding

به طور کلی، تحلیل‌های GeoServer از ورودی‌های پیچیده‌ای نظیر geometry، feature collection و raster استفاده می‌کند. فرمت‌های موجود برای ورودی و خروجی هر کدام از پردازش‌ها/تحلیل‌ها معمولاً توسط کلاس‌های PPIO[۱] و رمزگشا / رمزگذارهای عمومی[۲] مشخص می‌گردند که توسط ساز و کار WPS به هر کدام از تحلیل‌ها منتصب می‌شود. لازم به ذکر است مجموعه PPIO ها با نصب یک سری افزونه قابلیت گسترش دارند.

در هسته مرکزی WPS، PPIO های زیر موجود می‌باشند.

جدول۷- PPIO

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

Execute -۳-۵-۲

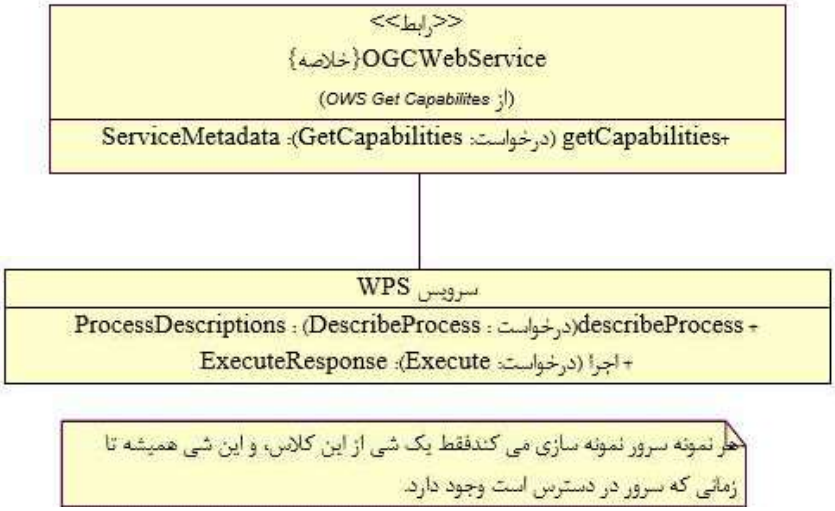
این عملیات به کاربر اجازه می‌دهد تا با استفاده از مقادیر پارامتر ورودی ارائه شده و برگرداندن خروجی‌های تولید شده، یک فرآیند مشخص را اجرا کند که توسط WPS پیاده سازی شده است. این عملیات شباهت‌های زیادی به سایر سرویس‌های تحت وب استاندارد OGC از جمله WMS، WFS و WCS دارند.

شکل ۱۵ یک نمودار UML ساده است که رابط WPS را خلاصه می‌کند. این نمودار کلاس نشان می‌دهد که کلاس واسط WPS عملیات GetCapabilities را از کلاس رابط OGCWebService به ارث می‌برد و عملیات DescribeProcess و Execute را اضافه می‌کند. تست این عملیات به طور کامل در قسمت مراحل راه‌اندازی سرویس WPS در ژئوسرور توضیح داده خواهد شد.

---

[۱] Process Parameter IO classes

[۲] generic encoders/decoders



شکل ۱۵- نمودار UML رابط WPS

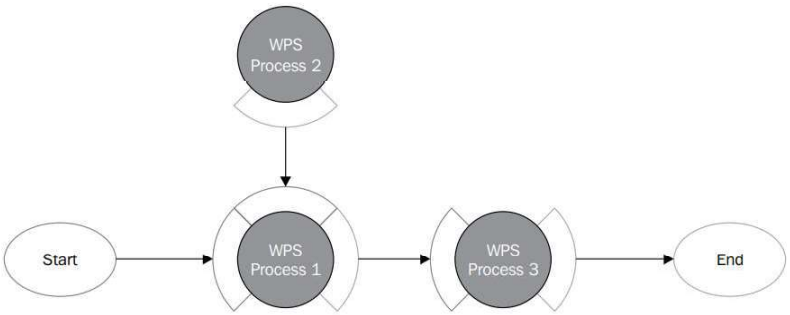
برای مثال، حالت ساده فرآیندی را در نظر بگیرید که می‌تواند تقاطع دو پلیگون را نشان دهد. پاسخ به درخواست GetCapabilities ممکن است نشان دهد که WPS از عملیاتی به نام "تقاطع" پشتیبانی می‌کند و این عملیات محدود به تقاطع یک پلیگون با پلیگون دوم است. پاسخ به یک درخواست DescribeProcess برای فرآیند «تقاطع» ممکن است نشان دهد که به دو ورودی نیاز دارد، یعنی: «FirstPolygon» و «SecondPolygon»، و این ورودی‌ها باید در GML ۲,۲ ارائه شوند. علاوه بر این، این فرآیند یک خروجی را در GML ۲,۲ یا GML ۳,۱ تولید می‌کند و می‌تواند به عنوان یک منبع در دسترس وب تحویل داده شود. کلاینت فرآیند را با فراخوانی عملیات Execute اجرا می‌کند و ممکن است انتخاب کند که دو پلیگون ورودی را که مستقیماً در داخل درخواست تعبیه شده‌اند 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 برمی‌گرداند که ورودی‌ها و خروجی‌ها را شناسایی می‌کند، نشان می‌دهد 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 می‌شود. یک تحلیل WPS می‌تواند شامل چندین ورودی دارای انواع مختلف باشد که به صورت value یا reference می‌تواند مشخص گردد. همچنین تحلیل WPS می تواند خروجی‌های متعددی را برگرداند، به همین دلیل یک تحلیل صرفاً به یک پارامتر خروجی وابسته نمی‌باشد. استفاده از خروجی یک تحلیل به عنوان ورودی یک تحلیل دیگر نیز امکان پذیر می‌باشد که از آن با WPS process chaining یاد می‌شود.

۲-۶-۱- Process chaining

انجام یک تحلیل WPS به منظور اجرای یک سری تحلیل‌ها صرفاً امکان استفاده محدودی از تحلیل‌های مکانی را میسر می‌کند. در صورتی که هدف، انجام یک سری آنالیزهای پیچیده‌تر باشد، نیاز به اجرای تعداد زیادی تحلیل‌های WPS می‌باشد. به طوری که هر بار خروجی ذخیره و در تحلیل بعدی استفاده شود (شکل ۱۶).



شکل ۱۶- Process Chaining

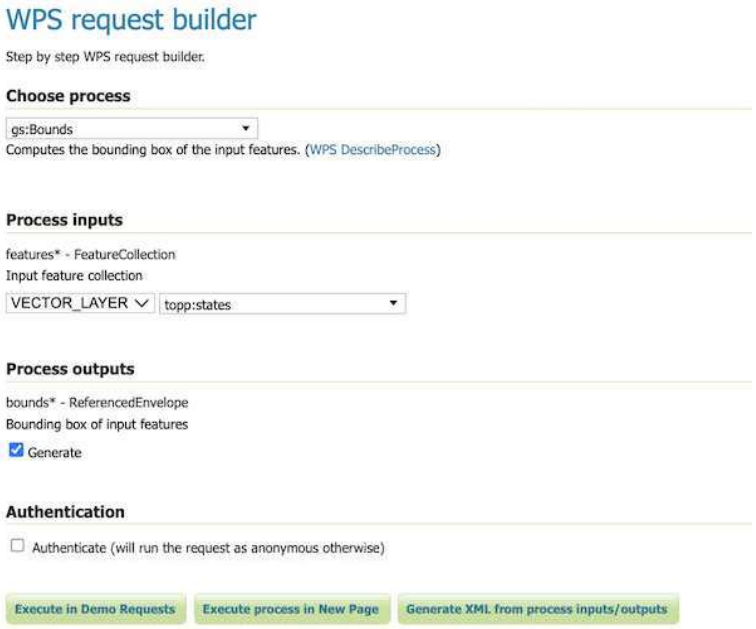
۲-۶-۲- WPS Request Builder

افزونه WPS ژئوسرور شامل یک Request Builder به منظور تست پردازش‌های WPS نیز می‌باشد. به منظور دسترسی به آن ابتدا Demos انتخاب و سپس WPS Request Builder از لیست انتخاب می‌شود (شکل ۱۷).



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 می باشد. 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 (شکل ۱۸).



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 که با Execute Process in New Page نتایج در پنجره‌ای نمایش داده می‌شود (شکل ۱۹).



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.

۲-۷- ماهیت عمومی و میان افزار WPS

WPS امکان ارائه داده‌های ورودی را به دو روش مختلف فراهم می‌کند. داده‌ها را می‌توان 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 می‌کند. در حالت دوم، WPS به عنوان سرویس میان‌افزار برای داده‌ها، با به دست آوردن داده‌ها از یک منبع خارجی به منظور اجرای یک فرآیند عمل می‌کند.

WPS به رابط‌های نرم‌افزاری موجود اجازه می‌دهد تا به عنوان خدمات وب به شبکه ارائه شوند. بنابراین پیاده سازی WPS را می‌توان میان افزار برای نرم‌افزار در نظر گرفت. در حقیقت، WPS یک رابط عمومی است که فرایندهای اجرایی آن‌ها را با ورودی و خروجی‌های مربوطه پشتیبانی می‌کند. WPS را می‌توان به عنوان یک مدل انتزاعی از یک وب سرویس در نظر گرفت که برای آن باید پروفایل‌هایی برای پشتیبانی از موضوعات مورد استفاده توسعه داده شود و برای پشتیبانی از قابلیت همکاری استاندارد شود. همانند سایر مشخصات OGC GML، این توسعه، انتشار و پذیرش پروفایل‌ها است که کاربردهای خاص این مشخصات را تعریف می‌کند.

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

- درخواست‌ها را برای اجرای فرآیند رمزگذاری کنید.
- پاسخ‌های حاصل از اجرای فرآیند را رمزگذاری کنید.
- داده‌ها و فراداده‌ها را در ورودی/خروجی‌های اجرای فرآیند جاسازی کنید.
- ورودی/خروجی داده‌های قابل دسترسی تحت وب را مرجع کنید.
- از فرآیندهای طولانی مدت پشتیبانی کنید.
- اطلاعات وضعیت فرآیند را بازگردانید.
- خطاهای پردازش را برگردانید.
- ذخیره‌سازی خروجی‌های فرآیند را درخواست کنید.

در حقیقت، سرویس WPS یک متد به منظور انتشار پردازش‌های مکانمند توصیف و ارائه می‌دهد. اما به طور خاص مشخص نمی‌کند که آن پردازش‌ها چه باید باشند. بنابراین سرورهایی که سرویس WPS را پیاده می‌کنند در رابطه با اینکه چه پردازش‌هایی را پیاده و به چه صورت پیاده کنند، آزادی عمل کاملی دارند. این بدین معنی است که انتظار نمی‌رود یک درخواست پردازشی که برای یک نوع از WPS طراحی شده بر روی نوع دیگر این سرویس کار بکند.

ژئوسرور پردازش‌ها را بر اساس موضوع و کتابخانه‌های مورد استفاده، دسته‌بندی می‌کند. این دسته‌ها به واسطه یک پیشوند مشخص می‌شوند 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 JTS Topology Suite[۱] ساخته می‌شوند. 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 می‌باشد. سرویس WPS ژئوسرور یک سری از این توابع را در گروه پردازش‌های geo پیاده می‌کند. اسم و تعریف توابع میتواند تغییر کنند که برای دیدن مجموعه‌ی کاملی از آن می‌توان به سند Capabilities سرویس WPS مراجعه کرد.

سرویس WPS ژئوسرور شامل ی کسری پردازش‌هاست که صرفاً برای کار با ژئوسرور ایجاد شده‌اند. این پردازش‌ها معمولاً توابع خاص ژئوسرور نظیر bounds و reprojection می‌باشند (GeoServer-specific). این پردازش‌ها از یک سری اتصال داخلی به سرویس‌های WFS/WCS ژئوسرور برای خواندن و نوشتن داده استفاده می‌کنند.

[۱] JTS

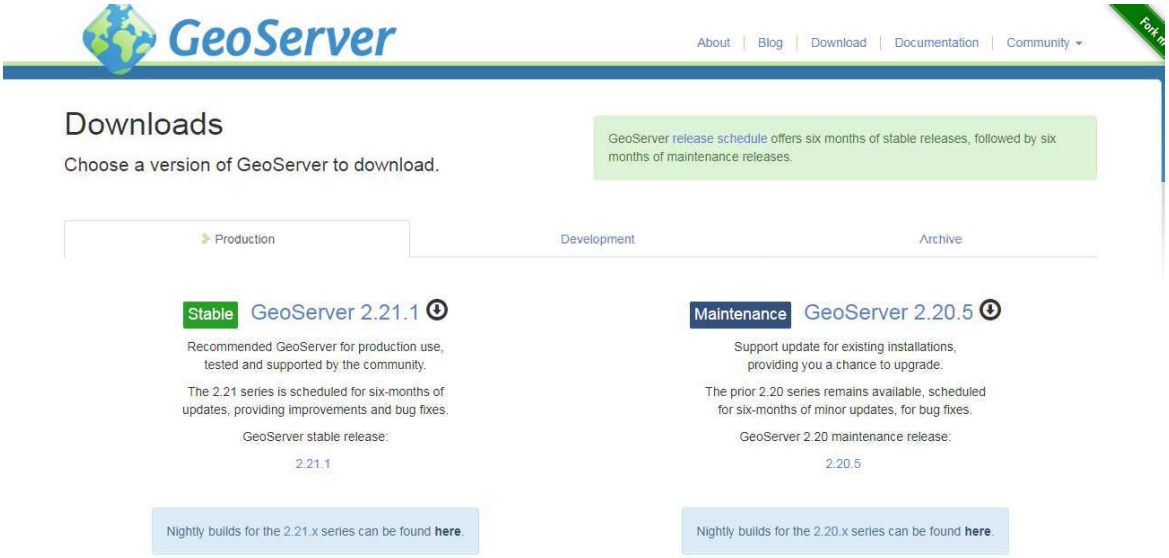
فصل سوم

نصب و راه اندازی WPS در ژئوسرور

۳-۱- مراحل راه اندازی سرویس WPS در ژئوسرور

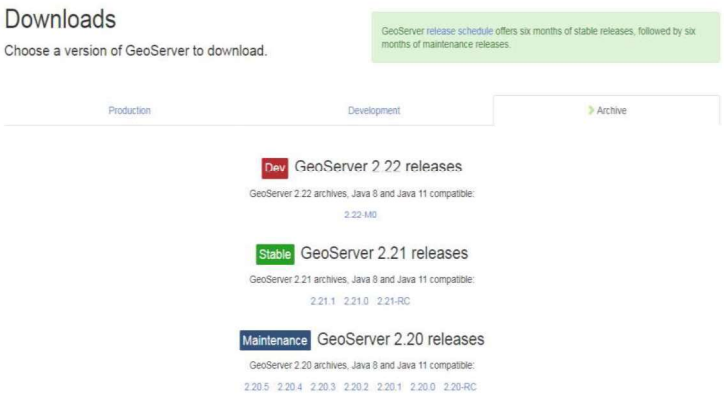
سرویس WPS به طور پیش‌فرض بخشی از ژئوسرور نمی‌باشد، اما به عنوان یک افزونه در دسترس و قابل نصب است. جهت دانلود و نصب آن مراحل زیر را دنبال کنید:

۱. وارد سایت ژئوسرور به مسیر https://geoserver.org شده و لینک download را انتخاب کنید (شکل ۲۰).



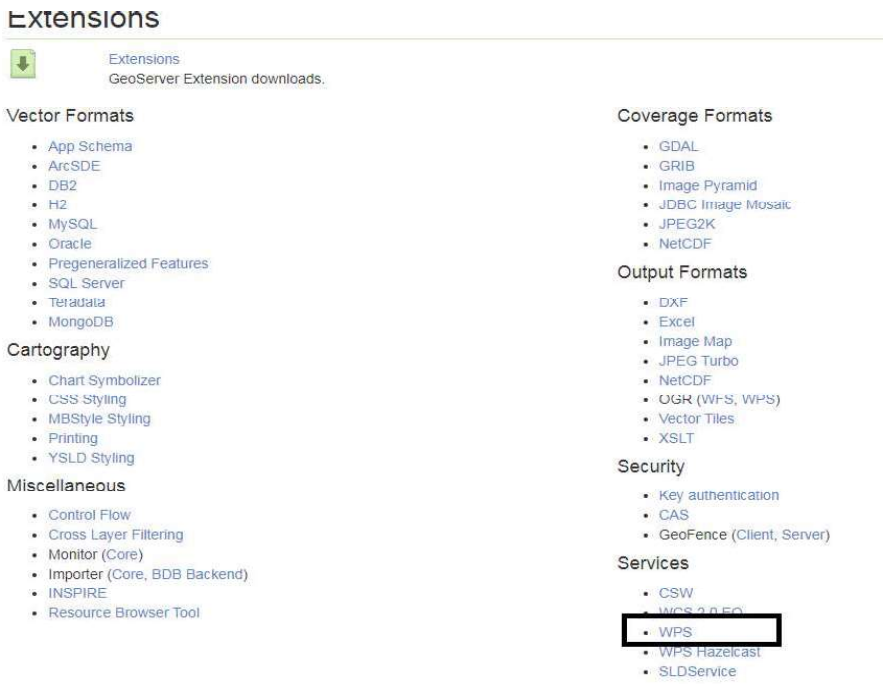
شکل ۲۰- صفحه دانلود GeoServer

۲. از تب Archive صفحه‌ای را که با نسخه GeoServer شما دقیقاً مطابقت دارد انتخاب نمایید (شکل ۲۱).



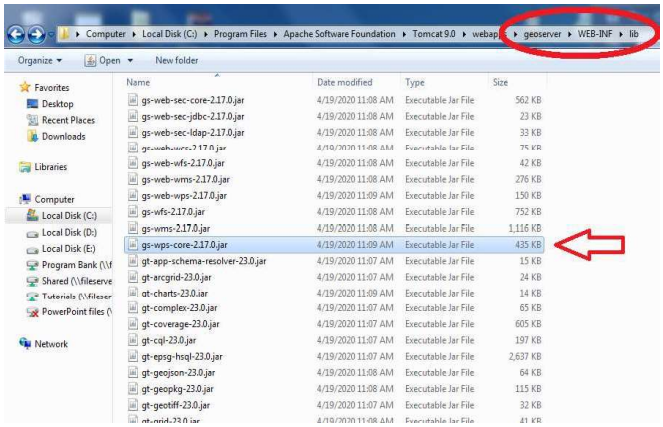
شکل ۲۱- دانلود نسخه‌های GeoServer

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 حتماً نسخه افزونه با نسخه نرم‌افزار ژئوسرور مطابقت داشته باشد، در غیر این صورت با خطا مواجه خواهید شد (شکل ۲۲).



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).



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

۵. نرم‌افزار را Restart نمایید.

۳-۲- نحوه پیاده سازی سرویس WPS در ژئوسرور

بعد از نصب افزونه WPS و به منظور پیاده‌سازی تحلیل‌های مکانی موجود در ژئوسرور، وارد نرم‌افزار ژئوسرور شده و روی بخش Demos کلیک نمایید (شکل ۲۴).



شکل ۲۴- استفاده از ماژول Demos برای ساخت درخواست‌های GeoServer

از صفحه GeoServer Demos گزینه WPS request Builder را انتخاب کنید (شکل ۲۵).

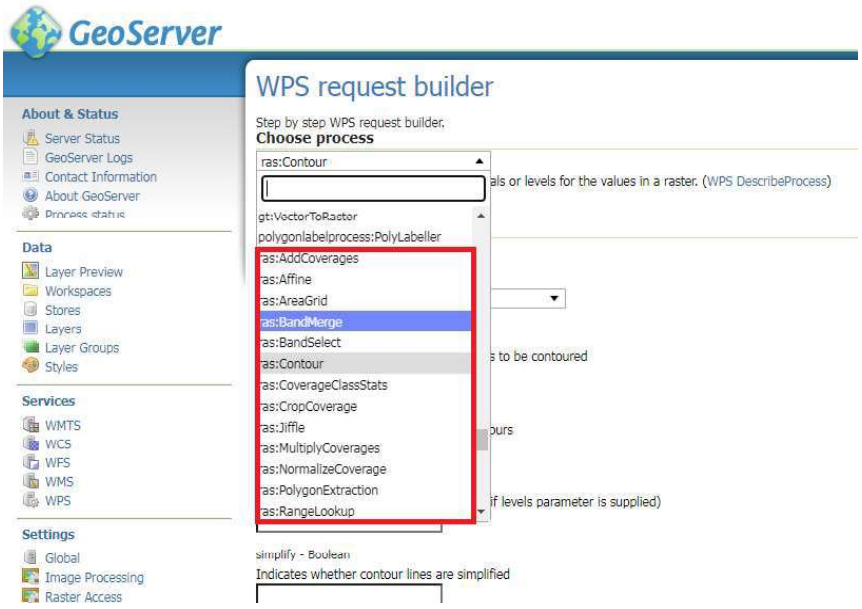


شکل ۲۵- انتخاب گزینه WPS request builder از ماژول Demos

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 می‌کند که عمدتاً می توان آن‌ها را به سه نوع بزرگ طبقه‌بندی نمود. برای توابع برداری سه حالت زیر در نظر گرفته می‌شوند 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 ویژگی‌های 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 پرس‌وجوگر**: این نوع فرآیندها را می‌توان به عنوان یک توسعه WFS در نظر گرفت. از طریق آن‌ها می‌توان پرس و جوها و توابع پیچیده را برای هر نوع هندسه خارجی نیز صادر کرد.
- **فرآیندهای هندسی**: توابع هندسی مثل بافر و تقاطع که بر روی منابع داخلی و خارجی کار می‌کنند.

برای استفاده از تحلیل‌های داده‌های رستری می‌توان در طبقه‌بندی مشاهده شده، تحلیل‌های با پیشوند :Ras را انتخاب نمود (شکل ۲۶).



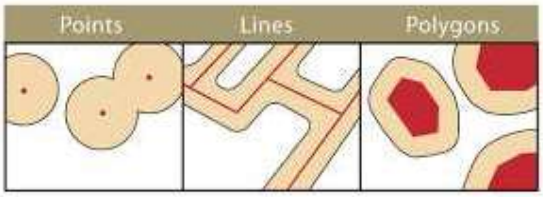
شکل ۲۶- انتخاب تحلیل مورد نظر منوی Choose process

فصل چهارم

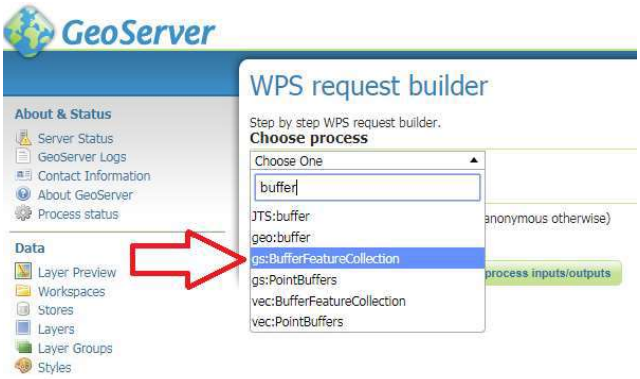
انواع تحلیل‌های مکانی پایه در ژئوسرور

۴-۱- تحلیل Buffer

جدول ۹- تحلیل Buffer

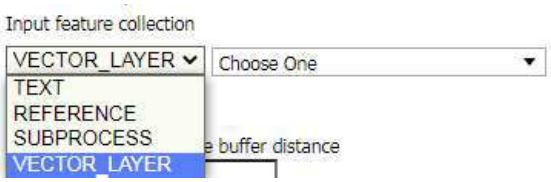
نام تحلیل	نام تابع	توضیحات
Buffer	gs:BufferFeatureCollection	ایجاد محدوده‌ای در اطراف عوارض مثال: محدوده ۵۰۰ متری اطراف میدان آزادی 

به منظور پیاده‌سازی تحلیل Buffer در نرم‌افزار ژئوسرور، از صفحه WPS request builder و از قسمت Choose process، کلمه buffer جستجو و گزینه gs:BufferFeatureCollection انتخاب می‌شود (شکل ۲۷).



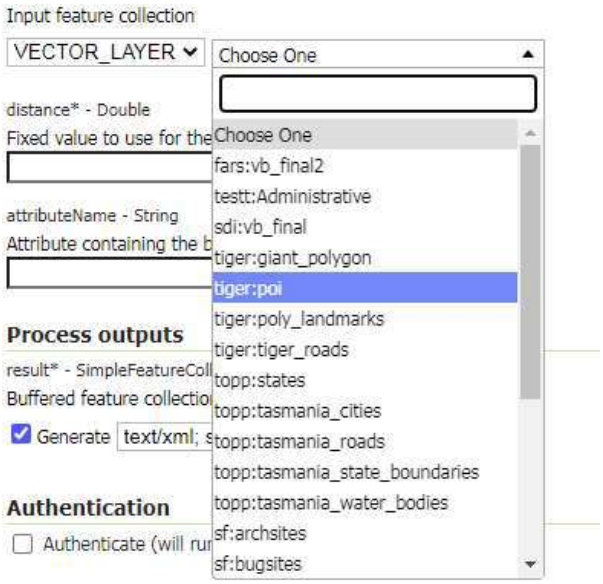
شکل ۲۷- انتخاب تحلیل Buffer از منوی Choose process

از قسمت Input feature collection که به صورت پیش‌فرض گزینه VECTOR_LAYER انتخاب شده است. ابتدا در این قسمت ورودی‌های مختلف ارائه خواهد شد (شکل ۲۸).



شکل ۲۸- انتخاب نوع ورودی از قسمت Input feature collection

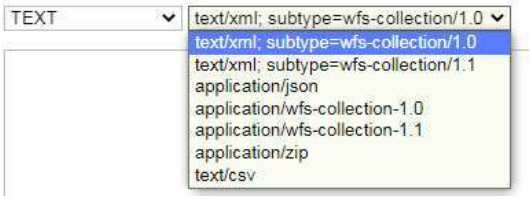
همانطور که در شکل ۲۹ نشان داده شده است ورودی‌ها در ۴ دسته قرار خواهند گرفت. حالت پیش‌فرض Vector_Layer است که به طور پیش‌فرض یکی از لایه‌های موجود در یک GeoServer قابل انتخاب خواهد بود.



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.



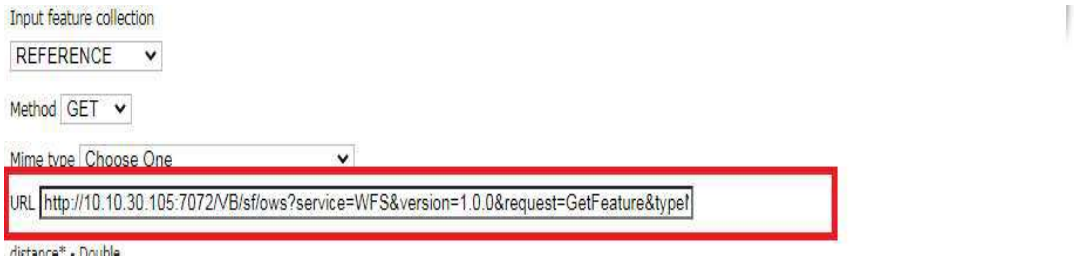
شکل ۳۰– انتخاب لایه مورد نظر در حالت TEXT به عنوان لایه ورودی

به عنوان مثال وضعیت 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","propertie
s":{"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"}}
```

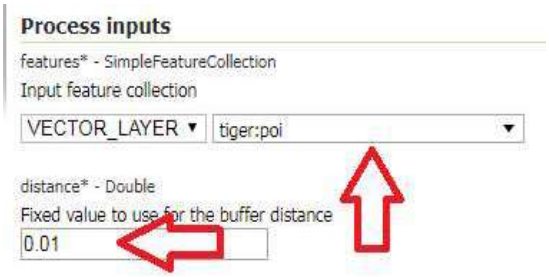

برای انتخاب ورودی‌ها در حالت Reference می‌توان اطلاعات ورودی سرویس WPS را از یک سرور خارجی با استفاده از استاندارد WFS دریافت نمود (شکل ۳۱).



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

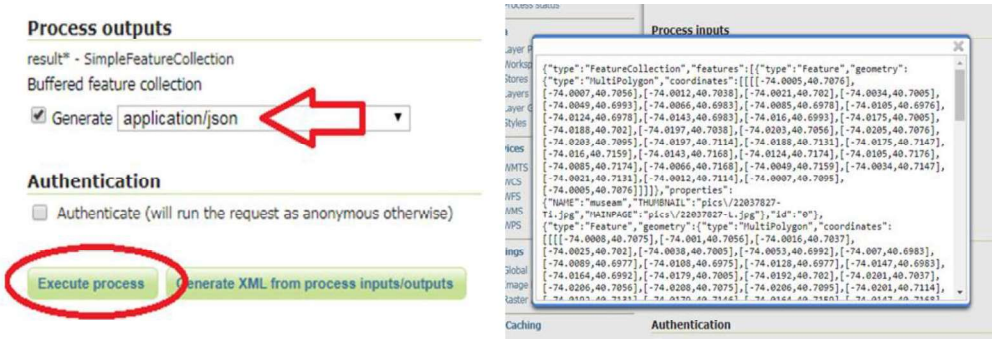
حالت Subprocess یک انتخاب مهم برای ساخت مدل‌های تحلیلی می‌باشد. در این حالت ورودی یک تحلیل از یک تحلیل دیگر به دست می‌آید. پس از تعریف تحلیل بافر، نمونه استفاده از این حالت را در قسمت تحلیل تقاطع به طور کامل توضیح خواهیم داد. در این حالت به عنوان مثال، ورودی تقاطع حریم یک لایه دیگر خواهد بود.

لایه مدنظر را انتخاب و سپس در قسمت distance مقدار فاصله مورد نظر با توجه به سیستم مختصات و سیستم تصویر لایه وارد می‌شود (شکل ۳۲).



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 precess 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).

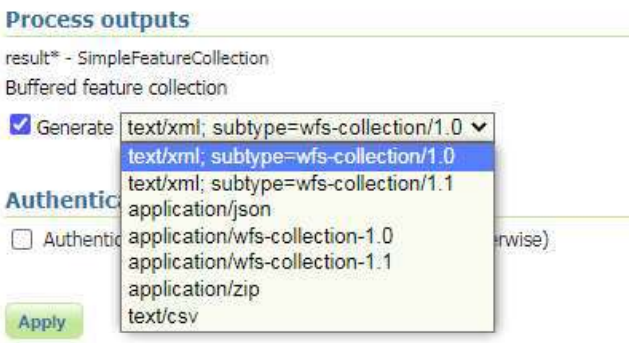


```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://
  <ows:Identifier>vec:BufferFeatureCollection</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>features</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2" xmlns:tiger="http://www
            <wfs:Query typeName="tiger:poi"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>distance</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>0.01</wps:LiteralData>
      </wps:Data>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="application/json">
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>
```

شکل ۳۴– دریافت خروجی تحلیل Buffer در زبان XML

تعریف خروجی‌ها:

پس از اجرای تحلیل می‌توان نوع خروجی تحلیل را مشخص نمود (شکل ۳۵).

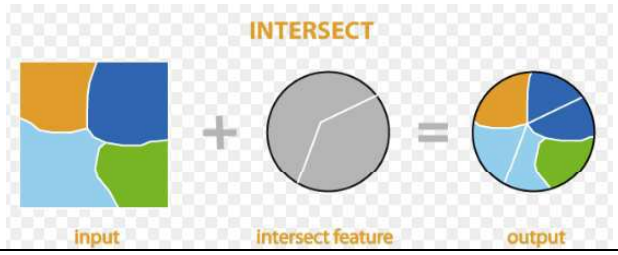


شکل ۳۵- نمایش انواع فرمت خروجی تحلیل‌های WPS در نرم‌افزار GeoServer

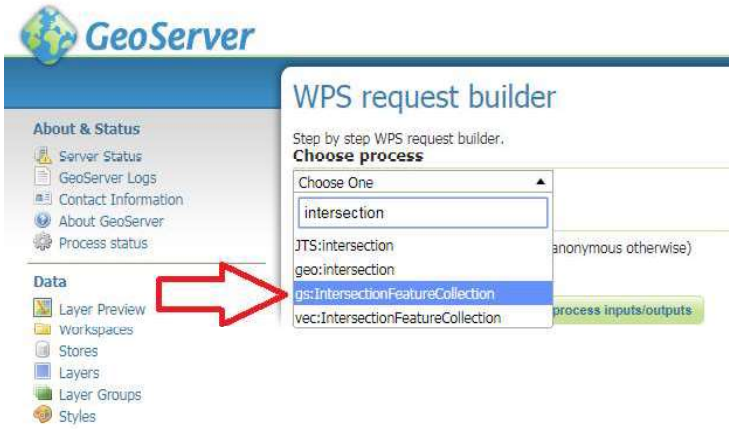
خروجی‌ها می‌تواند به صورت استاندارد WFS، GeoJSON و یا به صورت ShapeFile زیپ شده تعریف گردد. در هنگام استفاده از ShapeFile زیپ شده، خروجی پس از اجرای تحلیل دانلود می‌شود.

۴-۲- تحلیل Intersection

جدول ۱۰- تحلیل Intersection

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

جهت اجرای تحلیل Intersection در نرم‌افزار 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 کنید (شکل ۳۶).



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

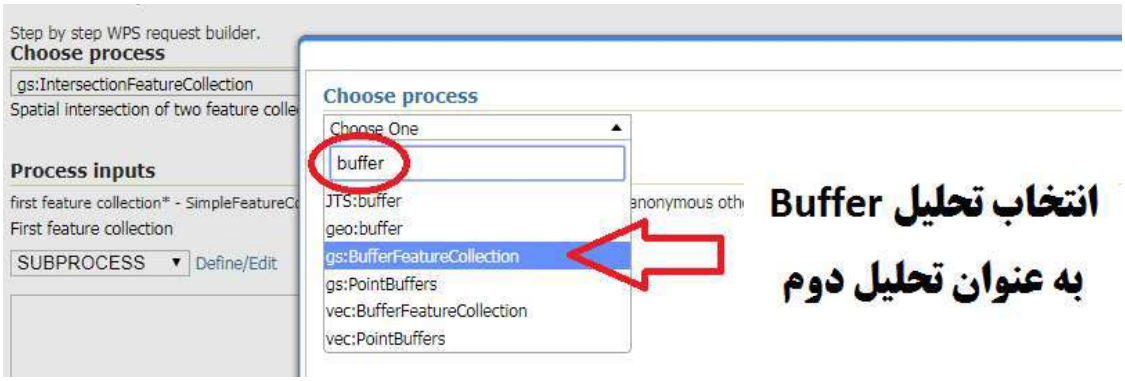
از قسمت First feature collection که به صورت پیش‌فرض گزینه VECTOR_LAYER انتخاب شده است که با توجه به توضیحات قسمت قبل می‌توان از چهار حالت موجود انتخاب شود. در این قسمت حالت SubProcess توضیح داده می‌شود. در این حالت کاربر می‌تواند به عنوان ورودی تحلیل جدید، خروجی یک تحلیل دیگر را قرار دهد به عنوان مثال برای ورودی تقاطع، ابتدا می‌توان بر روی یک لایه حریم ایجاد نموده و خروجی را با یک لایه دیگر تقاطع دهد (شکل ۳۷ تا ۴۰).



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



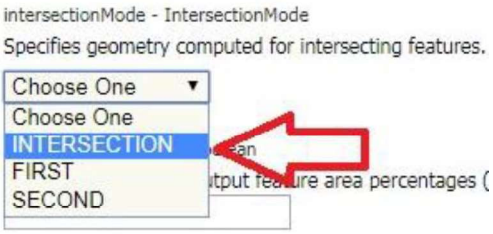
شکل ۴۰– انتخاب لایه ورودی از تحلیل **Buffer** و تعیین مقدار فاصله

پس از انجام تنظیمات فوق، بر روی دکمه Apply کلیک نموده و سپس از قسمت Second feature collection لایه دوم را انتخاب کنید (شکل ۴۱).



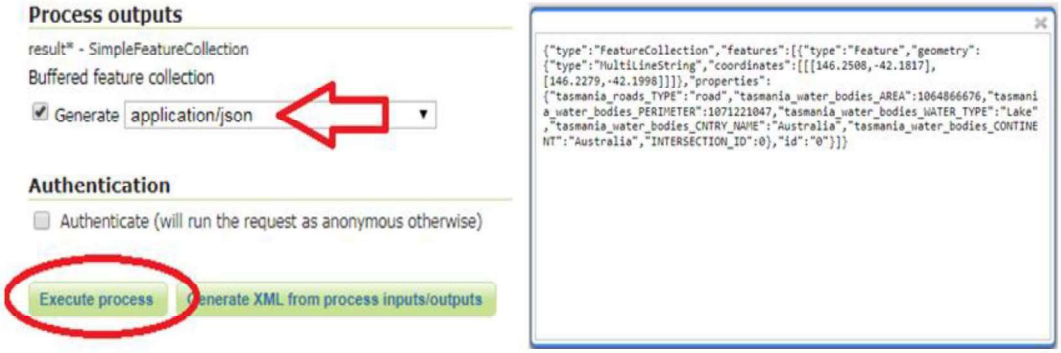
شکل ۴۱– مشاهده کد XML حاصل از تحلیل **Buffer** و انتخاب لایه دوم

در مرحله بعد intersectionMode را روی گزینه INTERSECTION قرار دهید (شکل ۴۲).



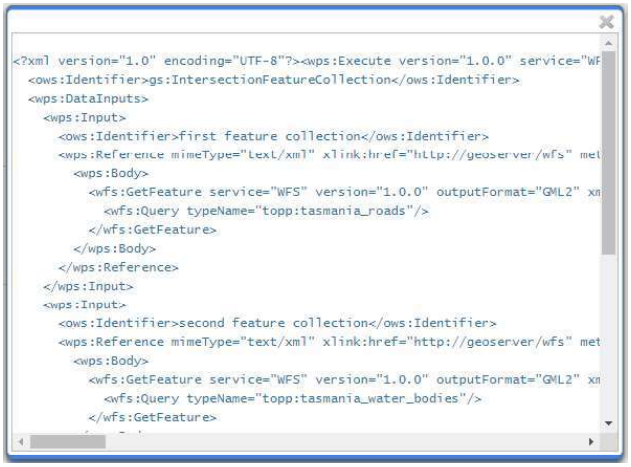
شکل ۴۲- انتخاب گزینه مربوطه به intersectionMode

در قسمت Process outputs و از Generate نوع خروجی فایل بر اساس یک فایل متنی قابل انتخاب است. با انتخاب گزینه Execute precess تحلیل فوق اجرا و خروجی در صفحه نمایش قابل مشاهده است (شکل ۴۳).



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

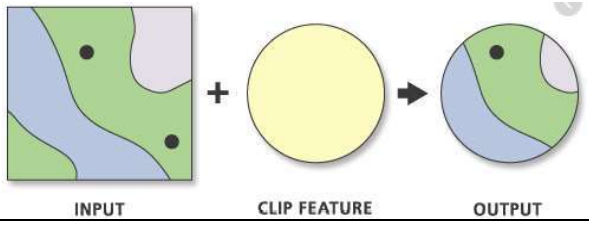
همچنین برای ایجاد درخواست بر اساس یک فایل XML می‌توان از گزینه Generate XML from process inputs/outputs استفاده کرد (شکل ۴۴).



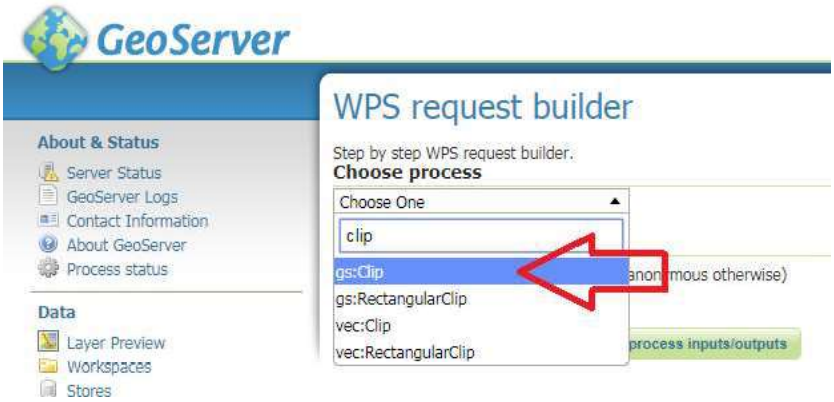
شکل ۴۴– دریافت خروجی تحلیل Intersection در زبان XML

۴-۳- تحلیل Clip

جدول ۱۱– تحلیل Clip

نام تحلیل	نام تابع	توضیحات
Clip	gs:Clip	برش زدن عوارض داخل محدوده موردنظر مثال: برش زدن محدوده‌ای خاص از شهر تهران 

جهت اجرای تحلیل Clip در نرم‌افزار 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 کنید (شکل ۴۵).



شکل ۴۵- انتخاب تحلیل Clip از منوی Choose process

از قسمت Input feature collection که به صورت پیش‌فرض گزینه VECTOR_LAYER انتخاب شده است، لایه مورد نظر را معرفی و سپس از قسمت Clip Geometry گزینه Reference را انتخاب نمایید (شکل ۴۶).



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 کنید (شکل ۴۷).



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 precess تحلیل فوق اجرا می‌گردد (شکل ۴۸).



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

همچنین برای ایجاد درخواست برا اساس یک فایل XML می‌توان از گزینه Generate XML from process inputs/outputs استفاده کرد (شکل ۴۹).

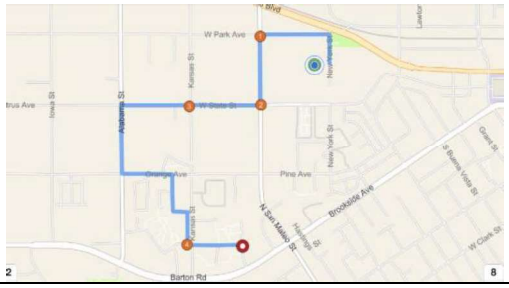


```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WF
  <ows:Identifier>gs:Clip</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>features</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" met
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2" xn
            <wfs:Query typeName="sf:bugsites"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>clip</ows:Identifier>
      <wps:Reference mimeType="application/json" xlink:href="http://localhost:
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="application/json">
```

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	نزدیک‌ترین عارضه به نقطه موردنظر مثال: نزدیک‌ترین مسیر به سازمان نقشه‌برداری 

جهت اجرای تحلیل Nearest در نرم‌افزار ژئوسرور، مجددا از بخش Demos وارد صفحه WPS request builder شده و از قسمت Choose process، کلمه Nearest را جستجو و گزینه gs:Nearest را انتخاب کرد.

از قسمت Input feature collection که به صورت پیش‌فرض گزینه VECTOR_LAYER انتخاب شده است، لایه مورد نظر را انتخاب نمایید (شکل ۵۰).



شکل ۵۰- انتخاب تحلیل Nearest و لایه مورد نظر

سپس از قسمت Point گزینه TEXT، و سپس گزینه application/wkt را انتخاب نموده و مختصات نقطه مورد نظر را وارد نمایید. به منظور معرفی سیستم مختصات و سیستم تصویر لایه ورودی روی گزینه Find کلیک کنید (شکل ۵۱).



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 کنید (شکل ۵۲).



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

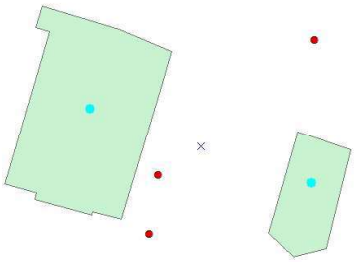
در قسمت Generate از بخش Process outputs گزینه application/json را جهت نحوه نمایش خروجی انتخاب نموده و جهت اجرای تحلیل Nearest بر روی گزینه Execute process کلیک کنید (شکل ۵۳).



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 کنید (شکل ۵۴).



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 نمایید (شکل ۵۵).



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

سپس از قسمت Second input geometry گزینه REFERENCE را انتخاب و بعد از تعیین حالت ورودی یا Mime type بر اساس text/xml;subtype=gml/3.1.1، آدرس URL لایه دوم را وارد کنید (شکل ۵۶).



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).

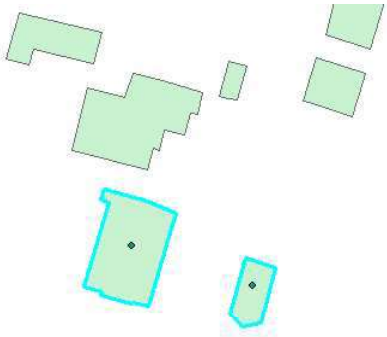


```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" 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>JTS:within</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>a</ows:Identifier>
      <wps:Data>
        <wps:ComplexData mimeType="text/xml;
subtype=gml/3.1.1"><![CDATA[<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=DescribeFeatu
reType&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">
<gml:pos>590232 4915039</gml:pos>
</gml:Point>
</sf:the_geom>
<sf:cat>1</sf:cat>
<sf:str1>Beetle site</sf:str1>
</sf:bugsites>
<sf:bugsites gml:id="bugsites.2">
<sf:the_geom>
<gml:Point srsName="http://www.opengis.net/gml/srs/epsg.xml#26713" srsDimension="2">
<gml:pos>590430 4915204</gml:pos>
</gml:Point>
```

شکل ۵۷- فایل XML حاصل از تحلیل within

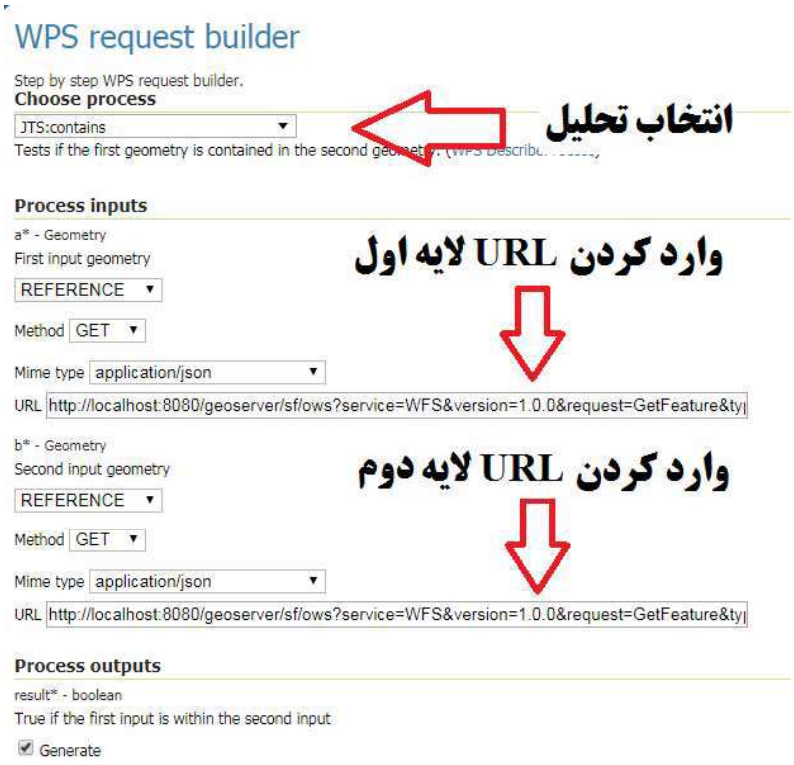
۴-۶- تحلیل Contains

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 کنید.

از قسمت First input geometry گزینه REFERENCE را انتخاب و URL لایه اول را وارد نمایید. سپس از قسمت Second input geometry مجدداً گزینه REFERENCE را انتخاب و URL لایه دوم را وارد کنید (شکل ۵۸).



شکل ۵۸– انتخاب تحلیل Contains و انجام تنظیمات مورد نظر

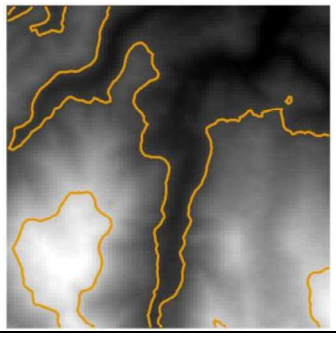
در نهایت پس از انتخاب دکمه Generate XML from process inputs/outputs فایل XML تحلیل contains به صورت زیر قابل دسترسی خواهد بود (شکل ۵۹).

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WF
  <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 منحنی‌های میزان با استفاده از فایل DEM 

با استفاده از این تحلیل می‌توان با معرفی فایل DEM به نرم‌افزار ژئوسرور و انجام تنظیمات مربوطه، منحنی‌میزان‌هایی با فاصله تعیین شده بدست آورد و از آن‌ها خروجی گرفت.

جهت اجرای تحلیل Contour در نرم‌افزار 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 کنید. از قسمت Input raster لایه مورد نظر خود را انتخاب و سپس از قسمت Interval فاصله منحنی میزان را وارد نمایید (شکل ۶۰).



شکل ۶۰– انتخاب تحلیل Contour و انجام تنظیمات مورد نظر

به منظور اجرای تحلیل Contour مطابق شکل ۶۱ عمل کنید.



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).

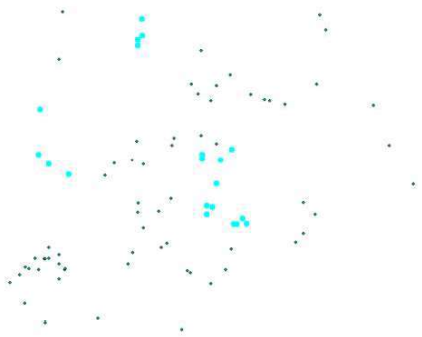


```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WF
  <ows:Identifier>ras:Contour</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>data</ows:Identifier>
      <wps:Reference mimeType="image/tiff" xlink:href="http://geoserver/wcs" n
        <wps:Body>
          <wcs:GetCoverage service="WCS" version="1.1.1">
            <ows:Identifier>test:sfdem_clip</ows:Identifier>
            <wcs:DomainSubset>
              <ows:BoundingBox crs="http://www.opengis.net/gml/srs/epsg.xml#26
                <ows:LowerCorner>592537.7250000006 4918748.525000004</ows:Lowe
                <ows:UpperCorner>595027.7250000006 4921238.525000004</ows:Uppe
              </ows:BoundingBox>
            </wcs:DomainSubset>
            <wcs:Output format="image/tiff"/>
          </wcs:GetCoverage>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
```

شکل ۶۲- فایل XML حاصل از تحلیل Contains

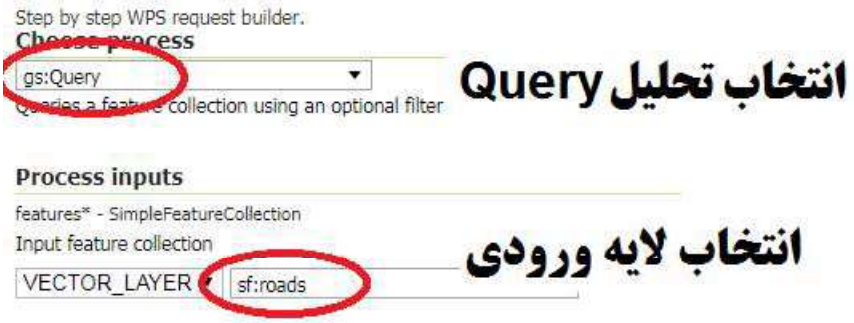
۴-۸- تحلیل Query

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 می‌باشند 

جهت اجرای تحلیل Query در نرم‌افزار 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 کنید.

سپس از قسمت Input feature collection و گزینه پیش فرض VECTOR_LAYER لایه مورد نظر را به عنوان لایه ورودی انتخاب نمایید (شکل ۶۳).



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

در مرحله بعد گزینه text/plain; subtype=cql را انتخاب نموده و Query مورد نظر را وارد نمایید. همچنین می‌توان از عملگرهای منطقی AND و OR نیز در Query ها استفاده کرد (شکل ۶۴).



شکل ۶۴- وارد کردن Query مورد نظر

در نهایت به منظور اجرای تحلیل فوق گزینه Execute process را انتخاب نمایید. خروجی این تحلیل به صورت زیر است (شکل ۶۵).



شکل ۶۵- خروجی حاصل از تحلیل Query

فصل پنجم

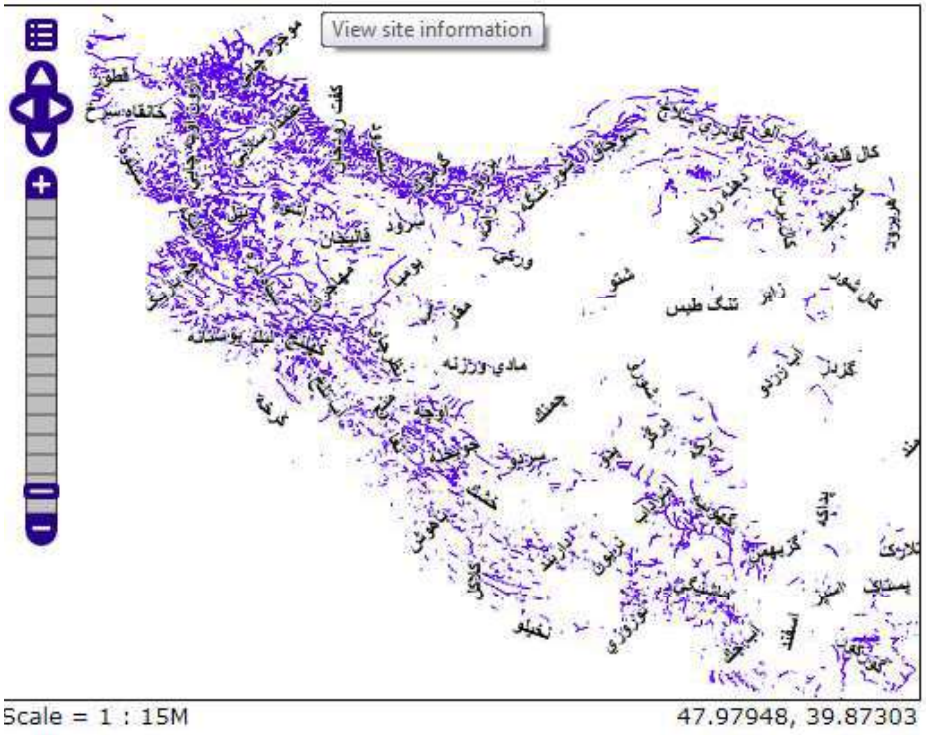
اجرای یک پروژه تحلیلی مکانی در ژئوسرور

۵-۱- اجرای یک پروژه موردی

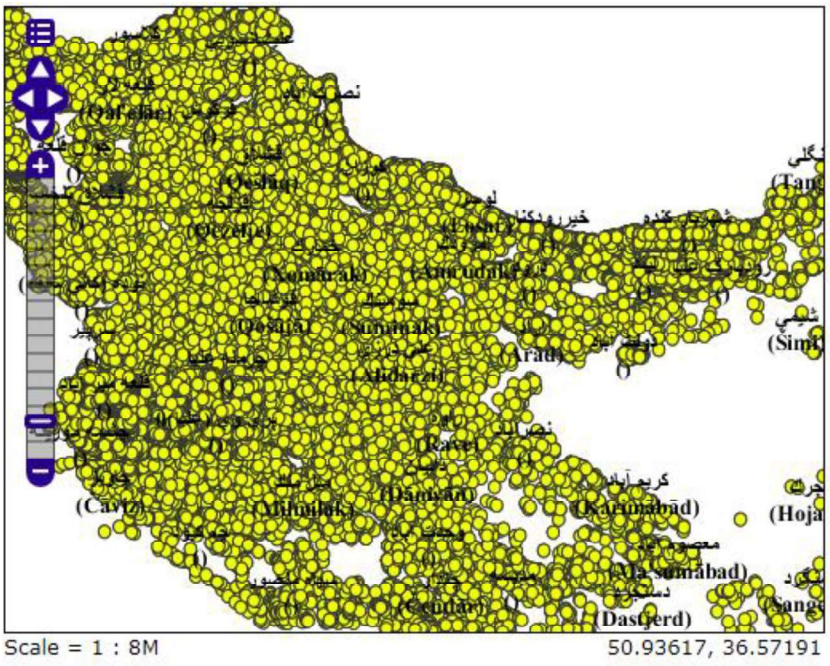
در این پروژه موردی، با استفاده از اطلاعات رودخانه‌های استان تهران، با استفاده از سرویس پردازشی پرسشی نام یک رود وارد شده و سپس بر اساس حریم ۱۰۰۰ متری بر روی این رودخانه روستاهایی که در برابر سیل در خطر واقع هستند نمایش داده می‌شود (شکل ۶۶ و ۶۷).

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

فرضیات: سرویس لایه‌های رودخانه‌ها، روستاها موجود هستند.

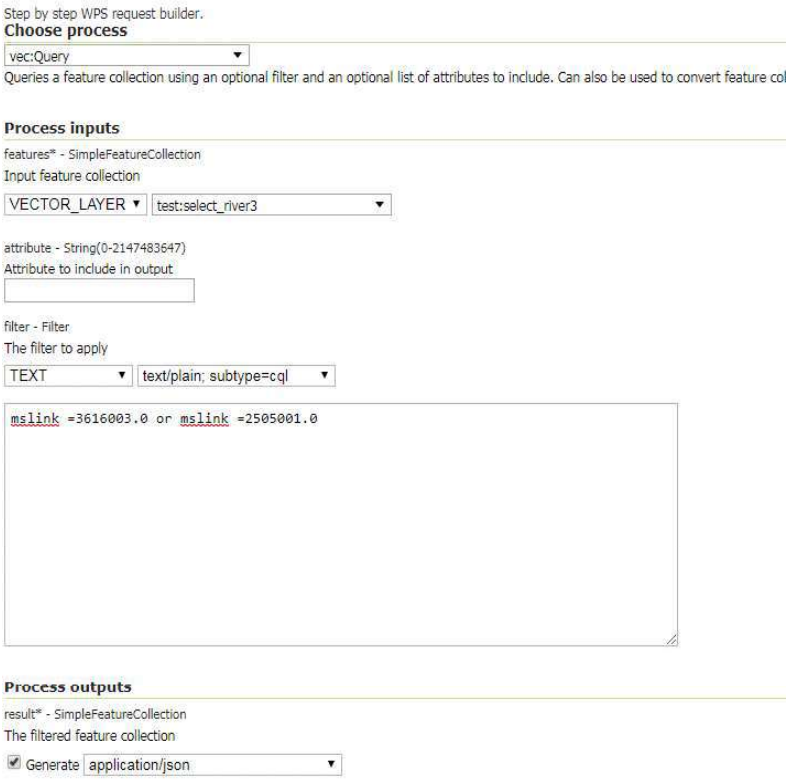


شکل ۶۶- سرویس WMS لایه رودخانه‌ها

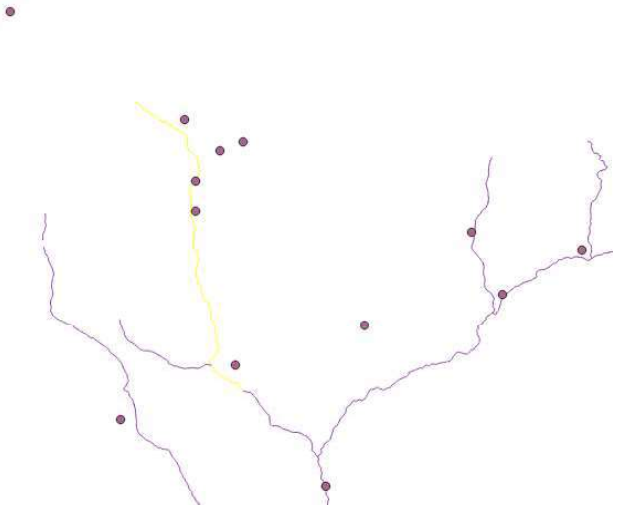


شکل ۶۷- سرویس WMS لایه روستاها

در این پروژه ابتدا یک رودخانه انتخاب می‌شود. برای انتخاب رودخانه از سرویس Query استفاده می‌شود. به عنوان مثال رودخانه جاجرود توسط کاربر انتخاب می‌شود. در سرویس لایه موجود هر رودخانه دارای یک فیلد به نام mslink است که شماره‌های ۲۵۰۵۰۰۱ و ۳۶۱۶۰۰۳ مربوط به رودخانه جاجرود می‌باشد (شکل ۶۸ و ۶۹).



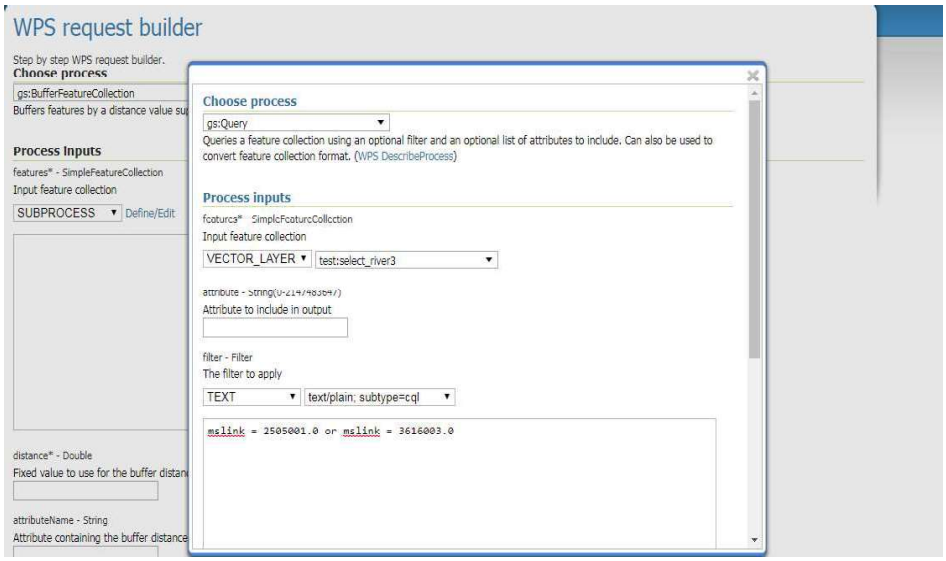
شکل ۶۸– انتخاب تحلیل Query و وارد نمودن Query مورد نظر



شکل ۶۹– نمایش رودخانه جاجرود

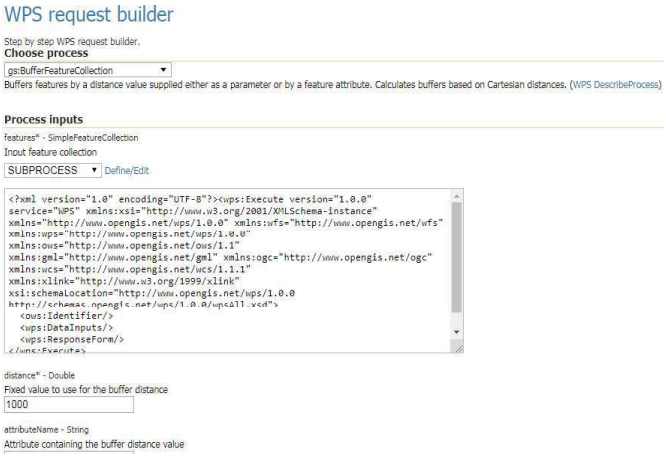
پس از انتخاب عارضه مورد نظر باید بر روی آن بافر زده شود تا حریم یک کیلومتری آن انتخاب گردد. برای انجام این کار در قسمت Choose Process گزینه BufferfeatureCollection را انتخاب کنید. سپس در قسمت input feature

collection گزینه subprocess را انتخاب نمایید. در این قسمت مجدد گزینه query را انتخاب کرده و بر روی لایه رودخانه جاجرود انتخاب شود (شکل ۷۰).



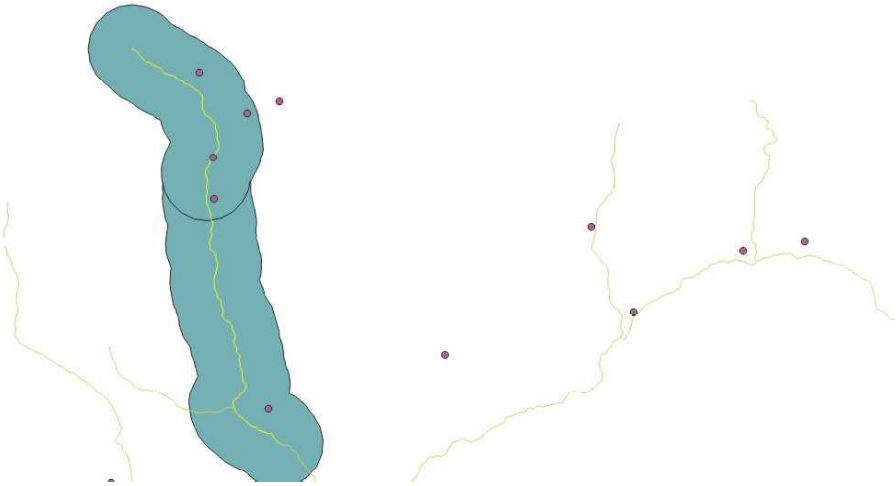
شکل ۷۰- انتخاب تحلیل Query به عنوان ورودی تحلیل Buffer

سپس پردازش بافر به صورت شکل ۷۱ تعریف می‌شود.



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

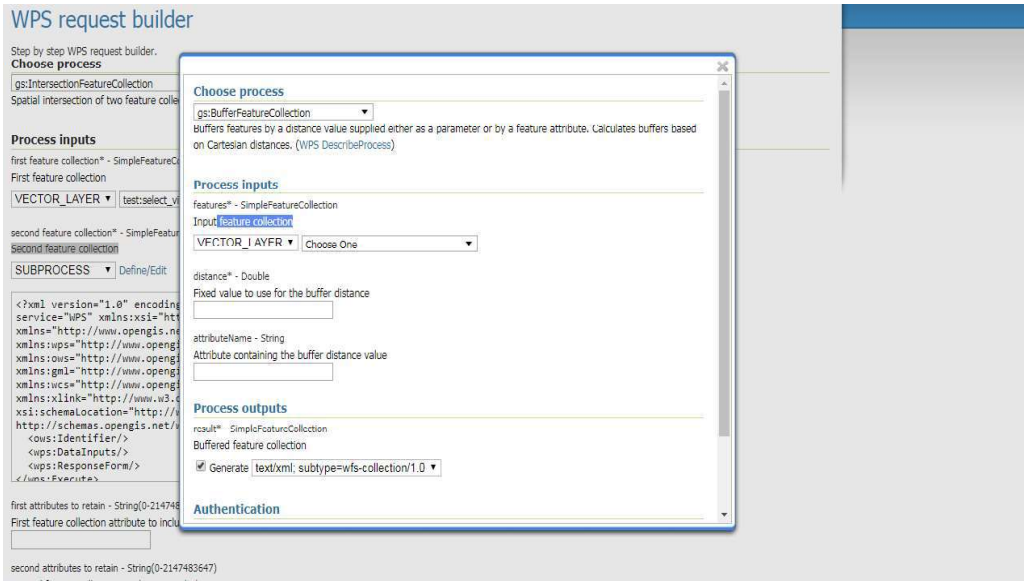
خروجی این تحلیل در شکل ۷۲ نشان داده شده است.



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

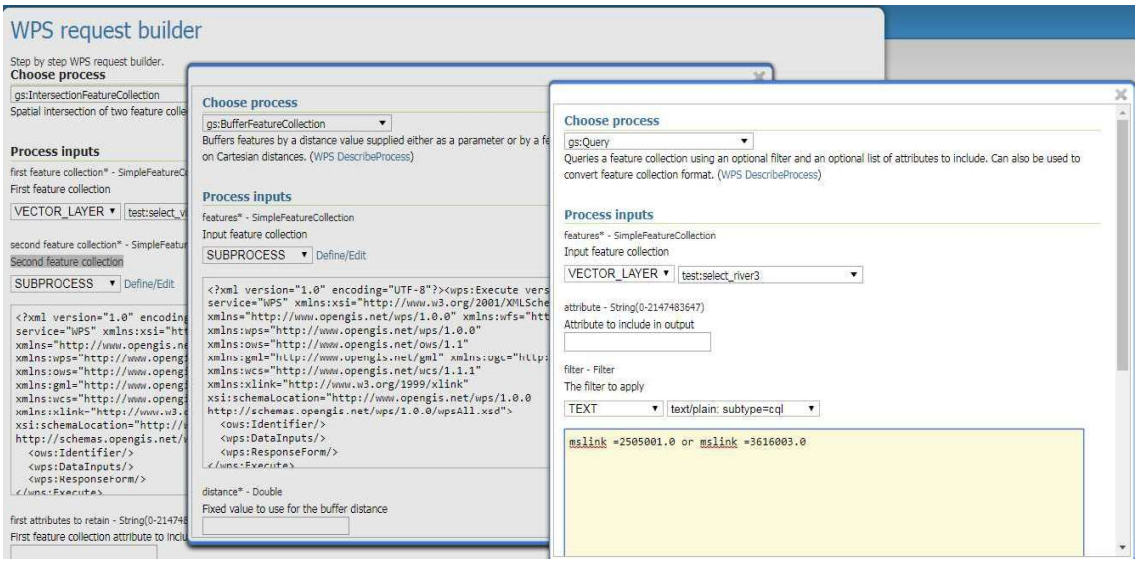
پس از ایجاد لایه حریم رودخانه مورد نظر، حال باید روستاهایی که در این منطقه قرار داده شده‌اند انتخاب گردند. بدین منظور باید از تحلیل intersection استفاده شود. و در آن روستاهایی که با این عارضه تقاطع دارند مشخص گردد.

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 لایه 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 و سپس در پنجره باز شده تحلیل bufferfeaturecollection انتخاب می‌شود (شکل ۷۳).



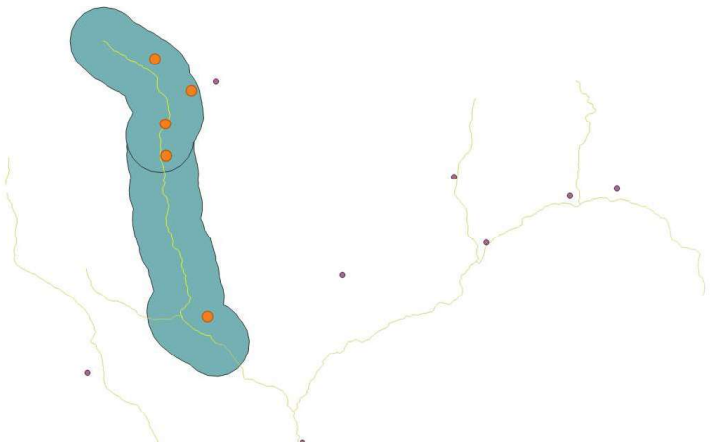
شکل ۷۳– انتخاب تحلیل Buffer به عنوان ورودی لایه دوم

حال در این قسمت برای ایجاد حریم برای عارضه رودخانه جاجرود مجدد در این قسمت از subprocess را انتخاب و در این قسمت از تحلیل query استفاده نموده و رودخانه جاجرود انتخاب می‌شود(شکل ۱۷).



شکل ۷۴- وارد کردن Query مورد نظر به منظور انتخاب رودخانه جاجرود

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 برمی‌گرداند.

$$p: X \to Y$$

در حقیقت، یک فرایند محاسباتی شامل برخی عملیات مکانی است به‌طوریکه می‌تواند 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 می‌دهید.

هر فرایند دارای ویژگیهای 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 ورودی‌ها و خروجی‌ها (**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 می‌کند که عبارت است از:
LiteralData, ComplexData, BoundingBoxData

تمامی نوع داده‌ها می‌بایست دارای ویژگی‌های زیر باشند:

- Identifier شناسه منحصر به فرد فرایند
- Title عنوانی خوانا برای انسان
- Abstract توصیفی بیشتر در مورد ورودی‌ها و خروجی‌ها
- minOccurs حداقل وقوع ورودی
- maxOccurs حداکثر تعداد دفعاتی که ورودی یا خروجی موجود می‌باشد

۶-۲-۱- LiteralData

هر رشته متنی غالباً کوتاه می‌باشد به طوریکه برای پارامترهای تکی نظیر اعداد و یا پارامترهای متنی به کار می‌رود. WPS اجازه تعریف مقادیر مجاز را به سرور می‌دهد. این تعریف در قالب لیست و یا فواصل مقادیر مجاز به همراه نوع داده (integer, float, string) می‌باشد. ویژگی‌های اضافی نظیر units و یا encoding نیز می‌تواند 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 داده‌ها 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. استفاده از این نوع داده در دنیای واقعی چندان رواج ندارد.

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:

- روش اول: داده روی سرور موجود باشد، که این روش ساده ترین حالت است.
- روش دوم: داده به همراه درخواست به سرور ارسال گردد. در این حالت داده که بخشی از XML می‌باشد توسط پروتکل HTTP POST فرستاده می‌شود و می‌تواند به صورت متنی (JSON و GML) و یا رستری باشد.
- روش سوم: ارسال اطلاعات به صورت reference link بوده و نیازی به انتقال داده نیست. این روش اگرچه بیشتر بر روی داده‌های 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 می‌نماید. در صورت پشتیبانی توسط فرآیند، سرور پاسخ ProcessAccepted را فوراً با URL می‌فرستد به طوریکه کلاینت می‌تواند به طور متناوب وضعیت فرآیند اجرا را بررسی کند.

۶-۵- وضعیت فرایند (process Status)

وضعیت فرایند در رابطه با چگونگی پیشرفت محاسبات گزارش می‌دهد و به‌صورت یکی از چهار حالت زیر می‌باشد:

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

۶-۶- کدگذاری درخواست HTTP GET و POST

این درخواست می‌تواند 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 می‌شود. مقادیر و کلیدها توسط علامت مساوی (=) از همدیگر جدا می‌شوند. همچنین هر جفت از جفت دیگر نیز توسط علامت & از همدیگر جدا خواهند شد (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 می‌گردد:

service, request و version به همراه مقادیر WPS، GetCapabilities و 1.0.0.

۶-۶-۲- 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 مدنظر باشد، می‌توان آخرین نسخه آن را از repository موجود در github[۱] 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 ارث‌بری می‌کند، نوشته می‌شود. نمونه Process به‌منظور پیکربندی، نیاز به ویژگی‌های زیر دارد:

[۱] https://github.com/geopython/pywps

- Identifier شناسه منحصر به فرد فرایند
- Title عنوان فرایند
- inputs لیستی از ورودی‌های فرایند
- outputs لیستی از خروجی‌های فرایند
- handler متدی است که object هایی از نوع request و response را دریافت نموده و به پردازش داده‌های ورودی، انجام آنالیز مکانی (با استفاده از کتابخانه‌هایی نظیر GDAL/OGR) و بازگرداندن خروجی می‌پردازد.

۶-۹- پیاده‌سازی نمونه اجرایی بر اساس PyWPS

هدف از پیاده‌سازی این نمونه اجرایی، ایجاد سرویس‌های پردازشی ساده از عملگرهای اصلی WPS یعنی GetCapabilites، DescribeProcess و Execute مبتنی بر وب می‌باشد.

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 پیاده‌سازی شده است که در آن کاربر مختصات (y و x) و فاصله مورد نظر را وارد نموده و پس از اجرا، بافری در اطراف آن نقطه ایجاد شده است.

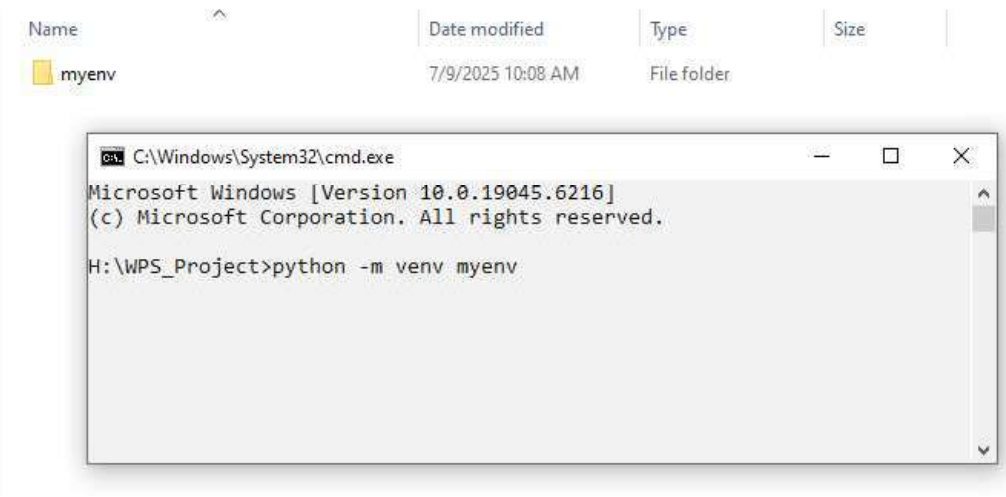
در این راستا از چارچوب PyWPS به منظور پیاده‌سازی استاندارد WPS مبتنی بر OGC و چارچوب Django برای ایجاد بستر تحت وب و تعامل با PyWPS استفاده خواهد شد و در مرحله بعد به ایجاد سرویس مبتنی بر وب برای تحلیل مکانی بافر اقدام می‌گردد.

Django یک چارچوب قدرتمند برای توسعه وب است که بر پایه زبان برنامه‌نویسی پایتون ساخته شده است. این چارچوب به دلیل ساختار ماژولار، امنیت بالا و قابلیت توسعه‌پذیری، یکی از محبوب‌ترین گزینه‌ها برای توسعه برنامه‌های تحت وب محسوب می‌شود. در این بین همانطور که اشاره شد PyWPS به عنوان یک پیاده‌سازی متن‌باز استاندارد WPS از OGC در زبان پایتون، به توسعه‌دهندگان اجازه می‌دهد تا پردازش‌های مکانی را به صورت سرویس‌های تحت وب ارائه دهند. ترکیب Django و PyWPS امکان ایجاد سامانه‌های قدرتمند پردازش مکانی را فراهم می‌کند که در آن مدیریت کاربران، امنیت، مسیریابی درخواست‌ها و تعامل با پایگاه‌داده به صورت منسجم و ساختاریافته انجام می‌شود. این رویکرد به‌ویژه برای پیاده‌سازی پردازش‌های مکانی در بسترهای سازمانی یا ژئوپورتال‌ها بسیار مفید و کاربردی می‌باشد.

هر بار که یک package در پایتون نصب می‌شود، در دسترس کل سیستم قرار می‌گیرد. از آنجایی که پروژه‌های مختلف نیازهای متفاوتی دارند، این امر می‌تواند منجر به ایجاد تعارضات شود. برای مقابله با این موضوع، پایتون از محیط‌های مجازی استفاده می‌کند که به صورت مجموعه‌های مستقل از یکدیگر فعالیت می‌کنند. 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 یک محیط مجازی (virtual environment) ایجاد می‌شود.

python -m venv myenv

پس از اجرای دستور فوق در cmd فولدری با نام 'myenv' در دایرکتوری مورد نظر ایجاد می‌گردد(شکل ۷۷).



شکل ۷۷- ایجاد محیط مجازی myenv

تذکر: نمونه اجرایی این دستورالعمل با استفاده از پایتون نسخه 3.9.11 در سیستم عامل ویندوز پیاده‌سازی شده است.

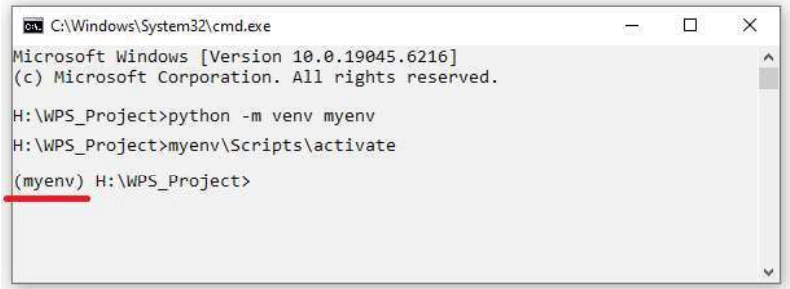
توجه: برای دانلود نسخه 3.9.11 پایتون و سایر فایل‌های مورد نیاز از لینک زیر استفاده نمایید.

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

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

myenv\Scripts\activate

در صورتی که مراحل به درستی انجام شده باشد، عبارت (myenv) در ابتدای خط فرمان مطابق شکل ۷۸ نمایش داده خواهد شد که نشان‌دهنده فعال بودن محیط مجازی است.



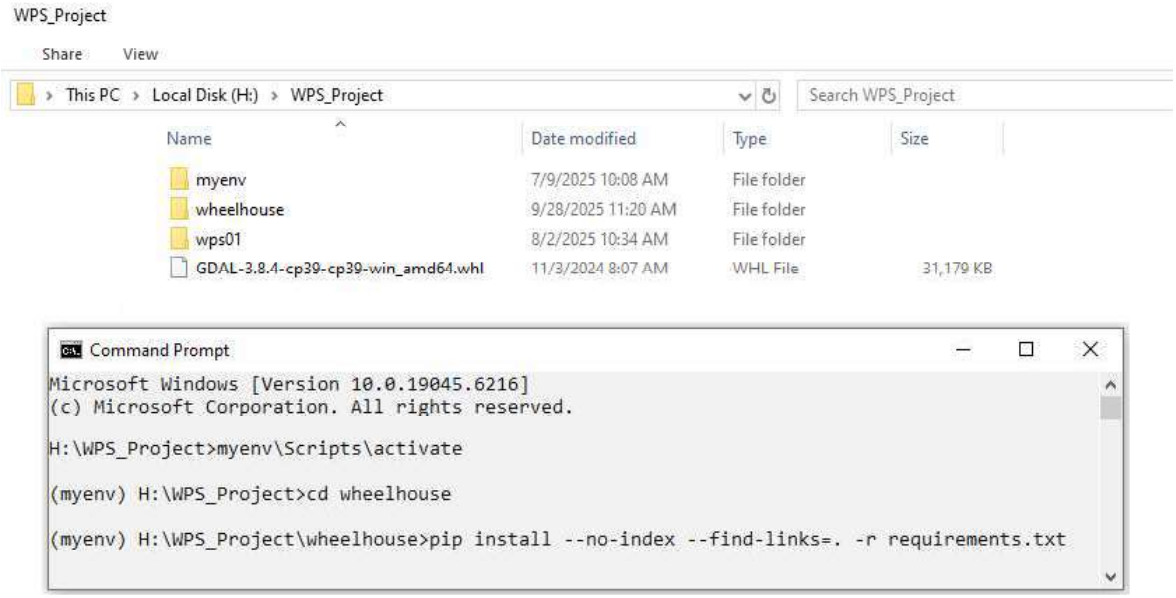
شکل ۷۸- فعالسازی محیط مجازی myenv

در مرحله بعد Django و PyWPS از طریق PIP نصب می‌شود. کتابخانه‌های Django و PyWPS در PyPI[1] 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 (شکل ۷۹).

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



شکل ۷۹– نصب آفلاین کتابخانه‌ها

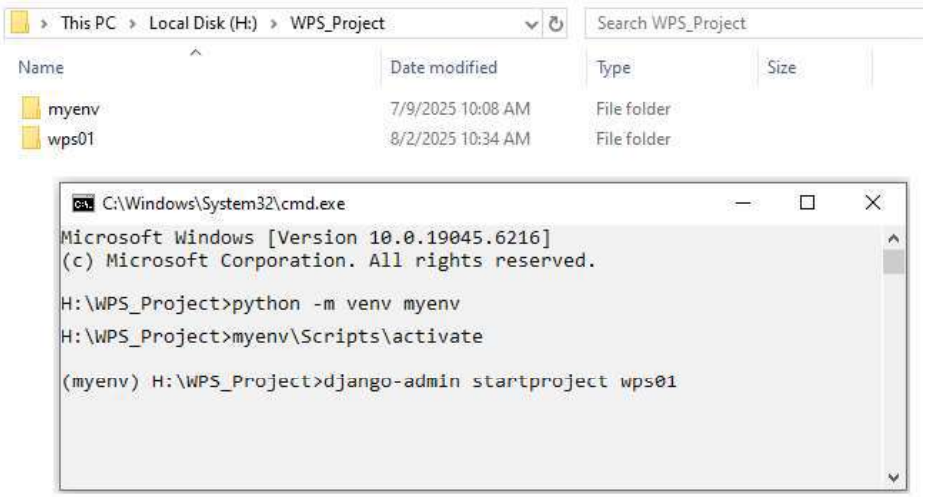
پس از نصب Django، می‌توان با استفاده از دستور django-admin یک چارچوب اولیه (اسکلت) برای پروژه خود ایجاد کرد. هنگامی که یک پروژه جدید ایجاد می‌کنید، Django حداقل نیازمندی‌هایی که برای شروع لازم است را در یک دایرکتوری جدید با نامی که توسط کاربر مشخص می‌شود، نصب می‌کند. لازم به ذکر است چارچوب Django یک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

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، یک پروژه Django با نام 'wps01' ایجاد خواهد شد (شکل ۸۰).

[1] https://pypi.org/

django-admin startproject wps01



شکل ۸۰- ایجاد پروژه Django با نام WPS01

پس از اجرای دستور فوق، فولدری با نام 'wps01' در دایرکتوری اصلی ایجاد می‌گردد. ساختار دایرکتوری پروژه جدید در شکل ۸۱ قابل مشاهده است.

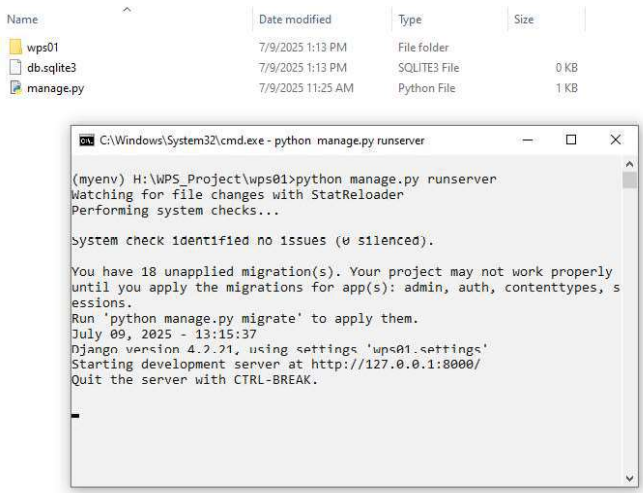
```
wps01/
├── wps01
│   ├── __init__.py
│   ├── asgi.py
│   ├── settings.py
│   ├── urls.py
│   └── wsgi.py
└── manage.py
```

شکل ۸۱- ساختار پروژه‌ی Django

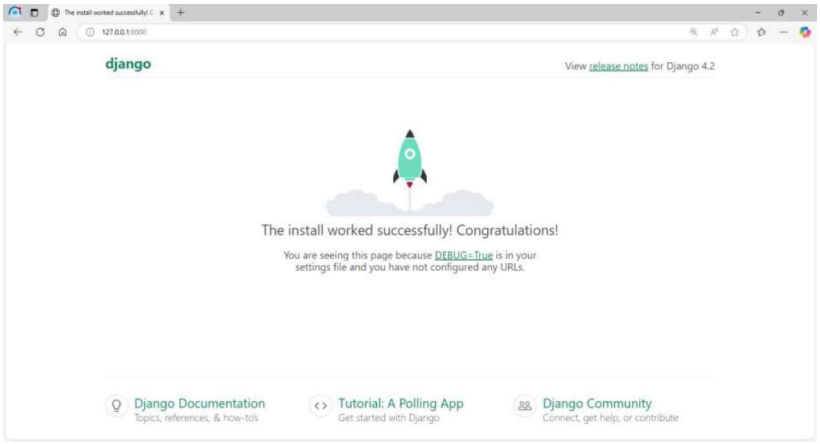
دایرکتوری پروژه (wps01) شامل دایرکتوری پیکربندی است که همان نام پروژه را دارد (wps01/wps01)، همچنین شامل محیط مجازی و فایل اجرایی manage.py نیز می‌باشد. فایل manage.py تقریباً همان کارهایی را انجام می‌دهد که django-admin انجام می‌دهد، با این تفاوت که به پروژه wps01 متصل بوده و از تنظیمات آن آگاه است. به منظور چک کردن درستی کار تا به اینجا می توان سرور توسعه‌ی Django را با استفاده از اسکریپت manage.py و دستور runserver اجرا کرد. باید توجه نمود که این کار باید داخل دایرکتوری اصلی 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 می‌توان پروژه‌ی Django خود را در مرورگر مشاهده کرد (شکل ۸۲ و ۸۳).



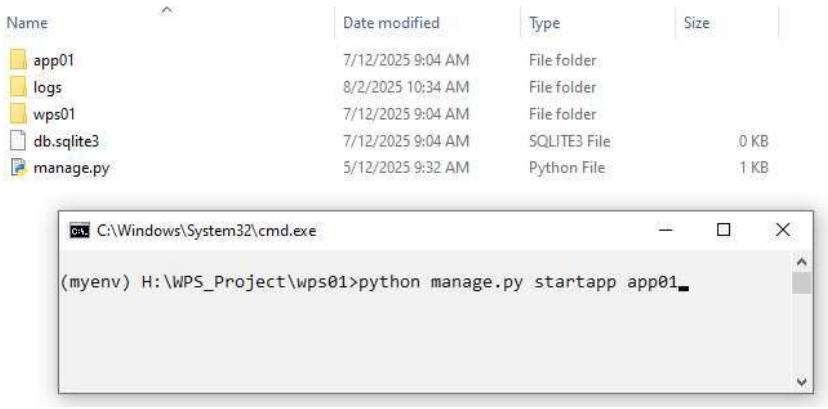
شکل ۸۲– اجرای سرور توسعه ای Django



شکل ۸۳– مشاهده‌ی پروژه‌ی Django در مرورگر

در این مرحله با دستور زیر به ایجاد یک app با نام app01 در درون پروژه پرداخته می‌شود (شکل ۸۴).

```
python manage.py startapp app01
```



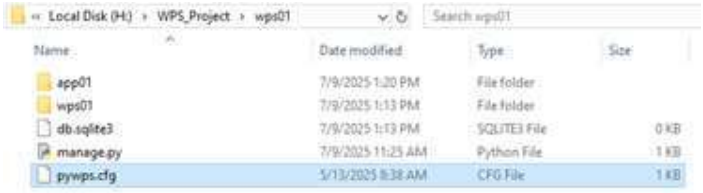
شکل ۸۴– ایجاد یک app در درون پروژه‌ی Django

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 (شکل ۸۵).

```
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 ترجیحاً در root directory به ایجاد یک فایل پیکربندی (Configuration File) با نام pywps.cfg اقدام کنید (شکل ۸۶).



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 GetCapabilties نیز مورد استفاده قرار می‌گیرد.

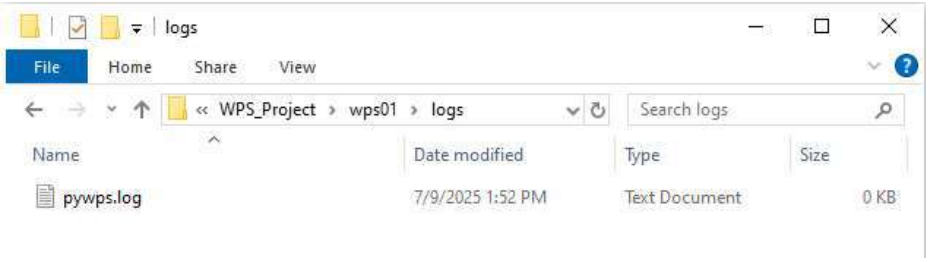
```
[metadata:main]
identification_title=PyWPS server
identification_abstract=PyWPS development server
identification_keywords_type=theme
identification_fees=None
identification_accessconstraints=None
provider_name= ...
provider_url= ...
contact_name=6948
contact_position=Dev
contact_address=Tehran
contact_city=Tehran
contact_stateorprovince=None
contact_postalcode=000 000
contact_country=Iran
contact_phone=+00 00 00 00 00
contact_fax=+00 00 00 00 00
contact_email=6948@ncc.gov.ir
contact_url=6948.ncc.gov.ir
contact_hours=07:00:00UTC
contact_instructions=...
contact_role=GISExpert

[server]
maxsingleinputsize=1mb
maxrequestsize=3mb
url=http://localhost:8000/wps
outputurl=http://localhost:8000/outputs/
outputpath=outputs
workdir=workdir
wd_inp_subdir=inputs
wd_out_subdir=outputs
maxprocesses=100
parallelprocesses=20

[logging]
level=INFO
file=logs/pywps.log
database=sqlite:///logs/pywps-logs.sqlite3
format=%(asctime)s] [%(levelname)s] file=%(pathname)s line=%(lineno)s module=%(module)s function=%(funcName)s %(message)s
```

شکل ۸۷– بخشهایی از پیکربندی شامل فراداده

حال همانطور که در بخش [logging] اشاره شد، به منظور ذخیره سازی logها، بایستی اقدام به ایجاد مسیر logs/pywps.log کنید، در غیر این صورت در هنگام اجرا با خطا مواجه خواهید شد (شکل ۸۸).



شکل ۸۸- اضافه کردن فایل log به پروژه

به منظور مدیریت بهتر، در درون app01 به ایجاد فولدری با نام processes اقدام کنید به نحوی که فرآیندهای PyWPS در آن ایجاد و تعریف گردند. داخل این فولدر فایل پایتونی به نام seyhello ایجاد کنید.

در نمونه اجرایی اول فرایند مربوطه دارای یک ورودی و یک خروجی که هر دو از نوع literal هستند، می باشد:

- 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 کلاس‌ها و ماژول‌های مورد نیاز به کدزنی آن پرداخت (شکل ۸۹).

```
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. همانطور که در بالا اشاره شد فرایند مورد نظر دارای یک ورودی و یک خروجی می‌باشد (شکل ۹۰).

```
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 تا بتواند ویژگیهای آن را تنظیم کند. که در آن پارامترهایی نظیر عنوان، خلاصه، نسخه و لیست ورودی و خروجی‌ها برای فرآیند مذکور تنظیم می‌گردند (شکل ۹۱).

```
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 که داخل آن پردازش انجام می‌گیرد پرداخته خواهد شد. در حقیقت این متد به عنوان هسته‌ی فرآیند PyWPS در نظر گرفته می‌شود. در این نمونه‌ی اجراییhandler به گرفتن ورودی و چسباندن آن (concatenation) به کلمه‌ی Hello می‌پردازد. و در انتها خروجی را بر می‌گرداند (شکل ۹۲).

```
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 واسطه‌ی استقرار و پیاده سازی یک سرویس WPS در مسیر wps01/wps01/wsgi.py تغییرات زیر اعمال می‌شوند. لازم به ذکر است فرایند تعریف شده در مرحله‌ی قبلی اینجا معرفی می‌شود. با انجام این کار در حقیقت کل پروژه به سرویس WPS تبدیل می‌گردد (شکل ۹۳).

```python
"""
WSGI config for wps01 project.

It exposes the WSGI callable as a module-level variable named ``application``.

For more information on this file, see
https://docs.djangoproject.com/en/4.2/howto/deployment/wsgi/
"""

from pywps.app.Service import Service
from app01.processes.sayhello import SayHello

processes = [
    SayHello(),
    ]

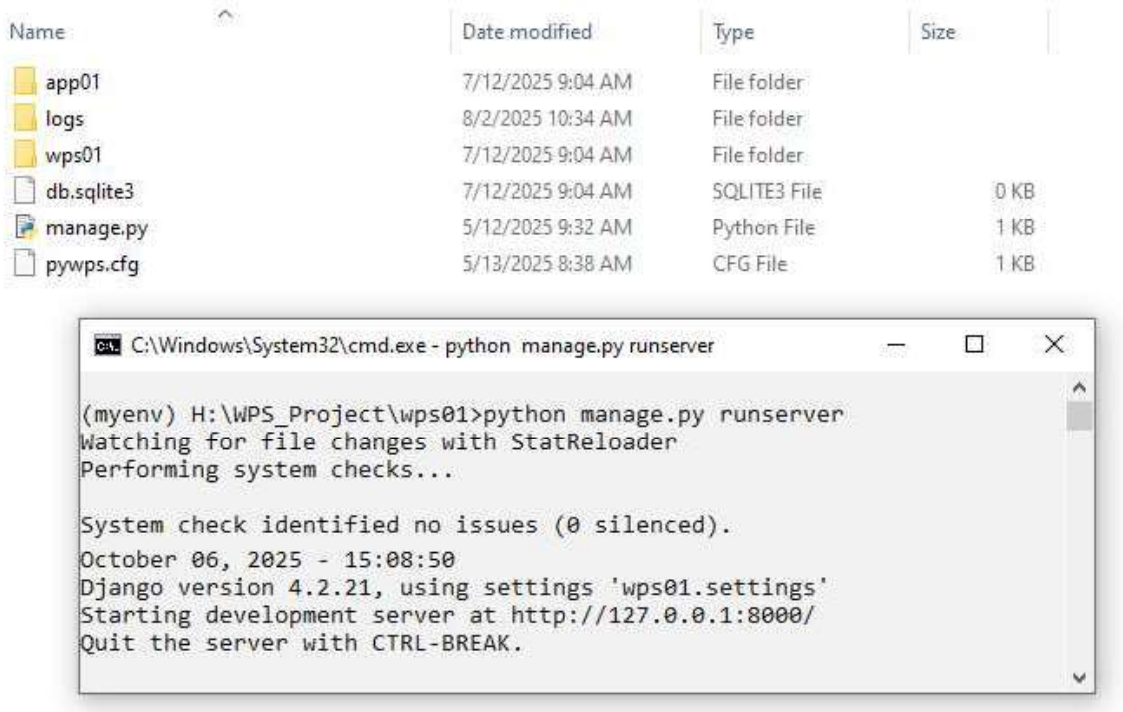
application = Service(processes, ['pywps.cfg'])

```

شکل ۹۳– افزودن فرایند SayHello به لیست فرایندهای فعال

حال با قرار گرفتن در مسیر دایرکتوری root دستور زیر و سرور توسعه‌ای Django اجرا می‌شود (شکل ۹۴).

python manage.py runserver



شکل ۹۴– اجرای سرور توسعه ای Django

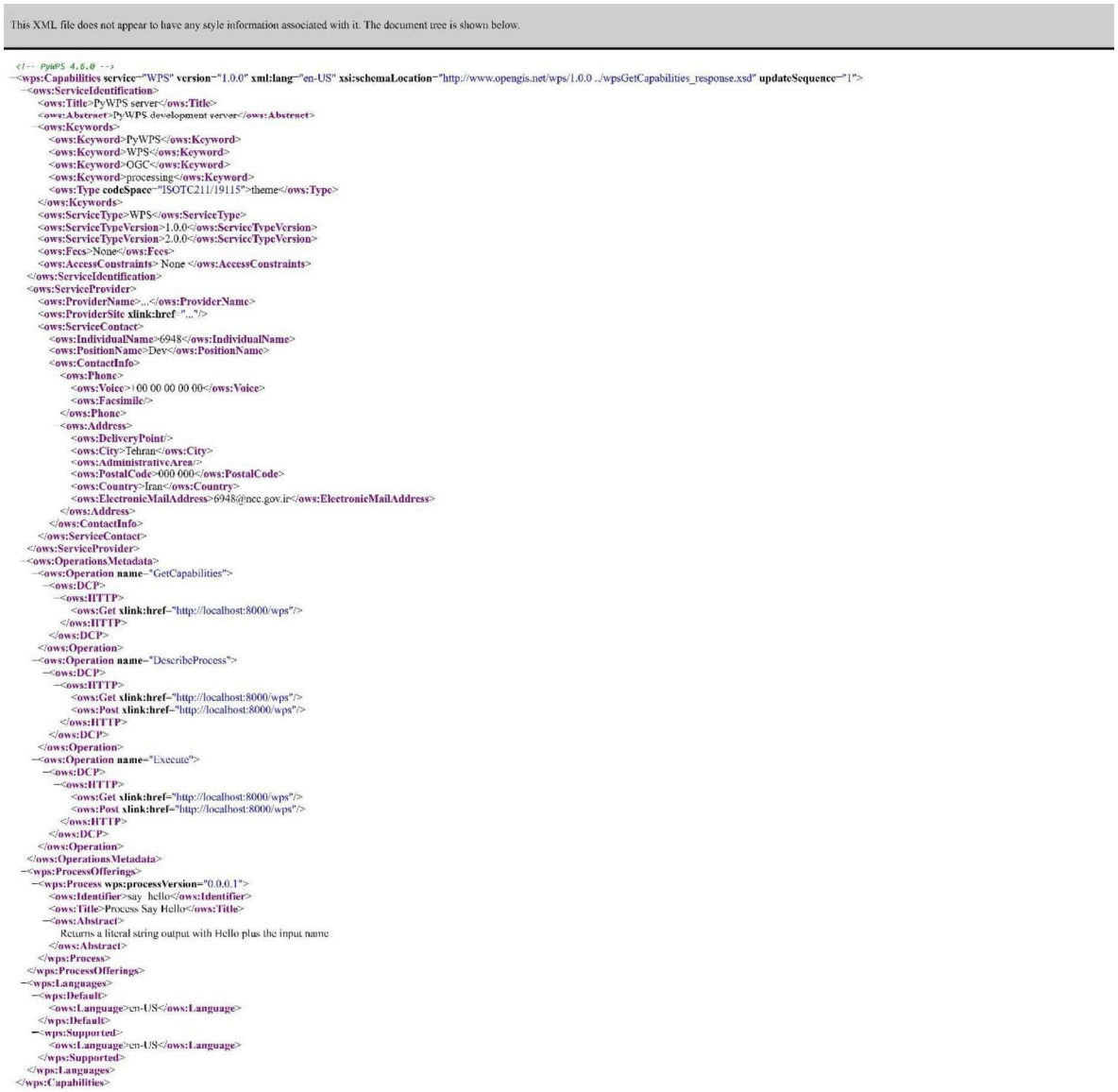
پس از ایجاد این فرایند می‌توان درخواست‌های GetCapabilites، DescribeProcess و Execute را ارسال و پاسخ آن را مشاهده نمود (شکل ۹۵، ۹۶ و ۹۷).

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="0.0.1">
      <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>
      </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>
      </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&amp;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>
    </wps:Output>
  </wps:ProcessOutputs>
</wps:ExecuteResponse>
```

شکل ۹۷- پاسخ سرور به درخواست Execute

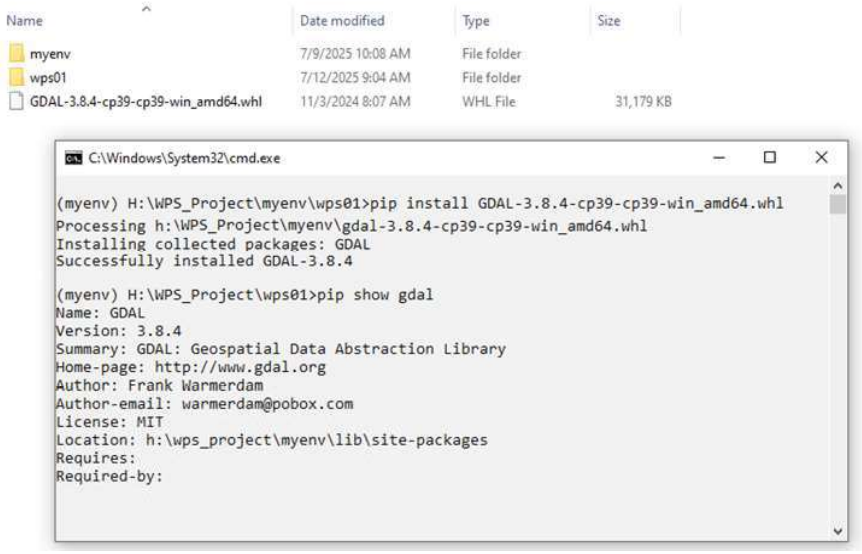
همان‌طور که دیده می‌شود در درخواست نام Mohsen ارسال و در پاسخ Hello Mohsen در XML خروجی بازگردانده شد.

مرحله‌ی بعدی اجرای تحلیل مکانی بافر (Buffer) مبتنی بر وب می‌باشد. شیوه‌ی کار همانند نمونه‌ی اجرایی قبلی می‌باشد با این تفاوت که در این تحلیل نیاز به استفاده از کتابخانه‌ی GDAL/OGR می‌باشد. بدین منظور پس از دانلود wheel فایل آن متناسب با نسخه‌ی پایتون 3.9.11 و سیستم عامل خود، به نصب آن در محیط مجازی اقدام می‌شود. در ویندوز، به دلیل نبود ابزارهای کامپایل پیش‌فرض، توصیه می‌شود از نسخه‌های whl. استفاده شود.

تذکر: نمونه اجرایی این دستورالعمل با استفاده از GDAL نسخه 3.8.4 پیاده‌سازی شده است.

در این مرحله با استفاده دستور زیر به نصب GDAL می‌پردازیم (شکل ۹۸).

```
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 متن‌باز برای خواندن، نوشتن، و پردازش داده‌های مکانی است. این کتابخانه از فرمت‌های بسیار متنوعی برای داده‌های رستری (مانندTIFF ، JPEG، وGeoTIFF) و برداری (مانندShapefile، GeoJSON، وKML) پشتیبانی می‌کند. بخش GDAL برای داده‌های رستری و بخش OGR برای داده‌های برداری طراحی شده‌اند، اما امروزه هر دو معمولاً به صورت یکپارچه استفاده می‌شوند. GDAL قابلیت‌هایی مانند برش، تغییر سیستم مختصات، تبدیل فرمت، و انجام تحلیل‌های مکانی را فراهم می‌کند و در پروژه‌های معروفی مانند QGIS، MapServer، و حتیGoogle Earth Engine نقش کلیدی دارد. این کتابخانه در زبان‌های مختلف از جمله Python، C/C++، Java و .NET. قابل استفاده است و به عنوان یکی از ارکان اصلی پردازش داده‌های مکانی در حوزه GIS شناخته می‌شود.

حال ایجاد فرایند بافر (buffer.py) در فولدر processes توضیح داده می‌شود.

در ابتدا کتابخانه‌های مورد استفاده وارد می‌شود که به دلیل کار با داده‌های برداری و سیستم‌های مختصات ماژول‌های ogr و osr فراخوانی می‌گردند (شکل ۹۹).

```
"""
# Name: buffer
# Description: performs buffer with simple inputs - x and y
#
# Author:   Mohsen Yousefzadeh
#
# Created:  ----
# Copyright:    (c) 6948
# Licence:  NCC - GIS
"""

from pywps import Process, LiteralInput, ComplexOutput, Format, FORMATS
from osgeo import ogr, osr
```

شکل ۹۹- فراخوانی کلاس ها و ماژول های مورد نیاز

در این فرایند سه ورودی وجود دارد، دو ورودی به عنوان مختصات x و y که از نوع عددی float می‌باشند و یک بافر با اعمال محدودیت روی مقادیر مجاز ورودی. این ویژگی سبب می‌شود کاربر فقط بتواند از مقادیر مجاز و کنترل شده استفاده کند. ورودی‌ها همه از نوع literalInput بوده ولی در خروجی استفاده از complexOutput اجازه استفاده از فرمت‌های پیچیده‌تری نظیر GML را به کاربر می‌دهد. با تعیین `as_reference=False` خروجی مستقیماً داخل پاسخ قرار داده می‌شود (نه به صورت لینک) (شکل ۱۰۰).

```
class simple_buffer(Process):
    def __init__(self):

        inputs = [LiteralInput('x', 'X coordinate', data_type='float'),
                  LiteralInput('y', 'Y coordinate', data_type='float'),
                  LiteralInput('buffer', 'Buffer size', data type='float',
                               allowed_values=(0, 1, 10, (10, 10, 100), (100, 100, 1000)))]


        outputs = [ComplexOutput('buff_out', 'Buffered file',
                                 supported_formats=[
                                          Format('application/gml+xml')
                                          ],
                                 as_reference=False
                                 )]
```

شکل ۱۰۰- تعریف ورودی و خروجیها در فرایند بافر

سپس به مقداردهی اولیه (initialization) کلاس فرآیند پرداخته می‌شود (شکل ۱۰۱).

```
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 ورودی‌ها دریافت، هندسه ایجاد، اعمال بافر صورت گرفته و خروجی در قالب فایل GML ذخیره می گردد (شکل ۱۰۲ و ۱۰۳).

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

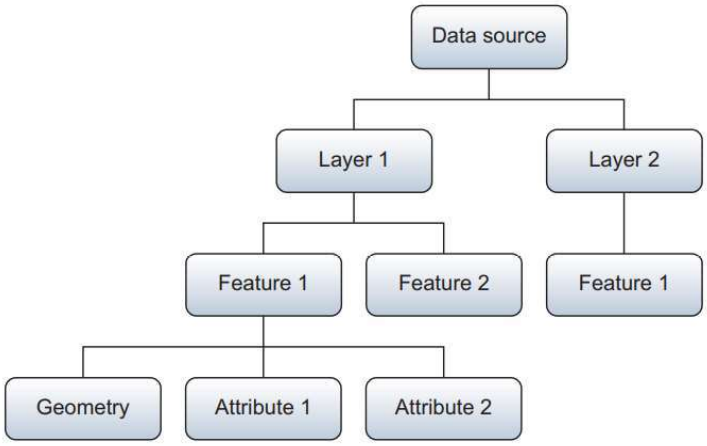
شکل ۱۰۲– تعریف متد handler به منظور انجام پردازش مورد نیاز

```
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 ورودی‌ها را که شامل x و y و سایز بافر می‌باشد، دریافت می‌کند. خطوط ۵۰ الی ۵۳ نقطه‌ای ایجاد کرده و با استفاده از Buffer() به ایجاد بافری حول آن می‌پردازد. در خطوط ۵۴ و ۵۵ سیستم مختصات آن 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 لایه‌ای با سیستم مختصات تعریف شده از قبل ایجاد می‌شود. در خطوط بعدی کد feature جدید ایجاد و هندسه‌ی بافر به آن اختصاص یافته و feature در لایه ذخیره می‌گردد. خطوط ۶۶ و ۶۷ نیز به منظور آزادسازی حافظه نوشته شده است. در انتها نیز فرمت خروجی به GML تنظیم شده و خروجی فرآیند به فایل point_buffer.gml اشاره می‌کند.

به منظور درک بهتر، ساختار کلی OGR در شکل ۱۰۴ آورده شده است. هر data source می تواند شامل چندین لایه باشد. هر لایه می تواند شامل چندین feature بوده و هر feature شامل geometry و یک یا چند attribute می‌باشد.



شکل ۱۰۴- ساختار کلاس OGR

در مرحله بعد همانند نمونه‌ی اجرایی قبلی در فایل wsgi موجود در مسیر wps01/wps01/wsgi.py تغییرات زیر اعمال می‌شود (شکل ۱۰۵).

```
"""
WSGI config for wps01 project.

It exposes the WSGI callable as a module-level variable named ``application``.

For more information on this file, see
https://docs.djangoproject.com/en/4.2/howto/deployment/wsgi/
"""

from pywps.app.Service import Service
from app01.processes.sayhello import SayHello
from app01.processes.buffer import simple_buffer

processes = [
    SayHello(),
    simple_buffer(),
    ]

application = Service(processes, ['pywps.cfg'])
```

شکل ۱۰۵– افزودن فرایند Simple_buffer به لیست فرایندهای فعال

کد تکمیل می‌باشد و می‌توان پس از اجرای سرور توسعه‌ای Django به ارسال درخواست و دریافت پاسخ اقدام نمود(شکل ۱۰۶، ۱۰۷ و ۱۰۸).

GetCapabilites:

- request

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

- response



شکل ۱۰۶- پاسخ سرور به درخواست GetCapabilities

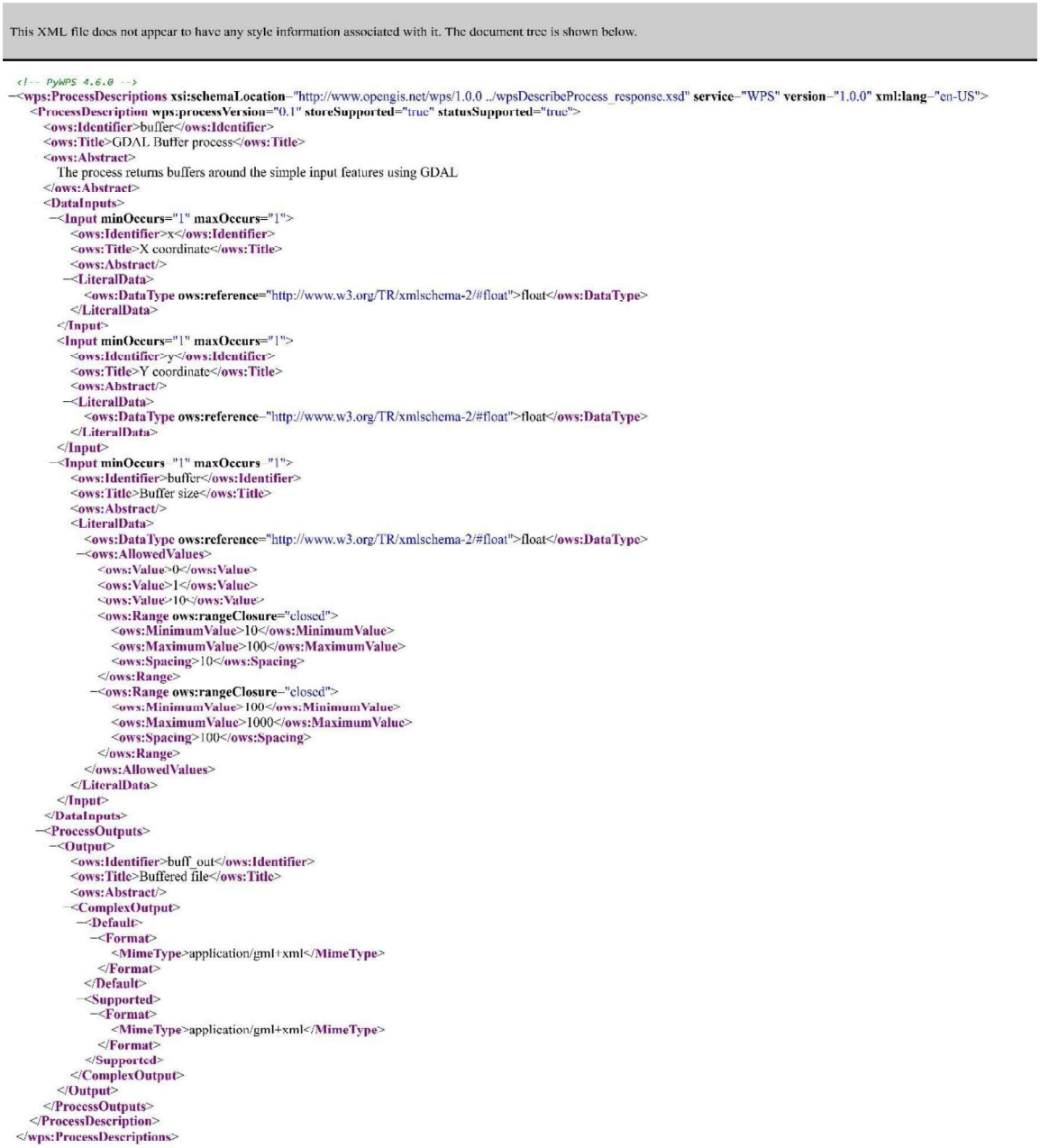
همان‌طور که دیده می‌شود فرایند جدید (بافر) در XML خروجی اضافه شده است.

: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>
```

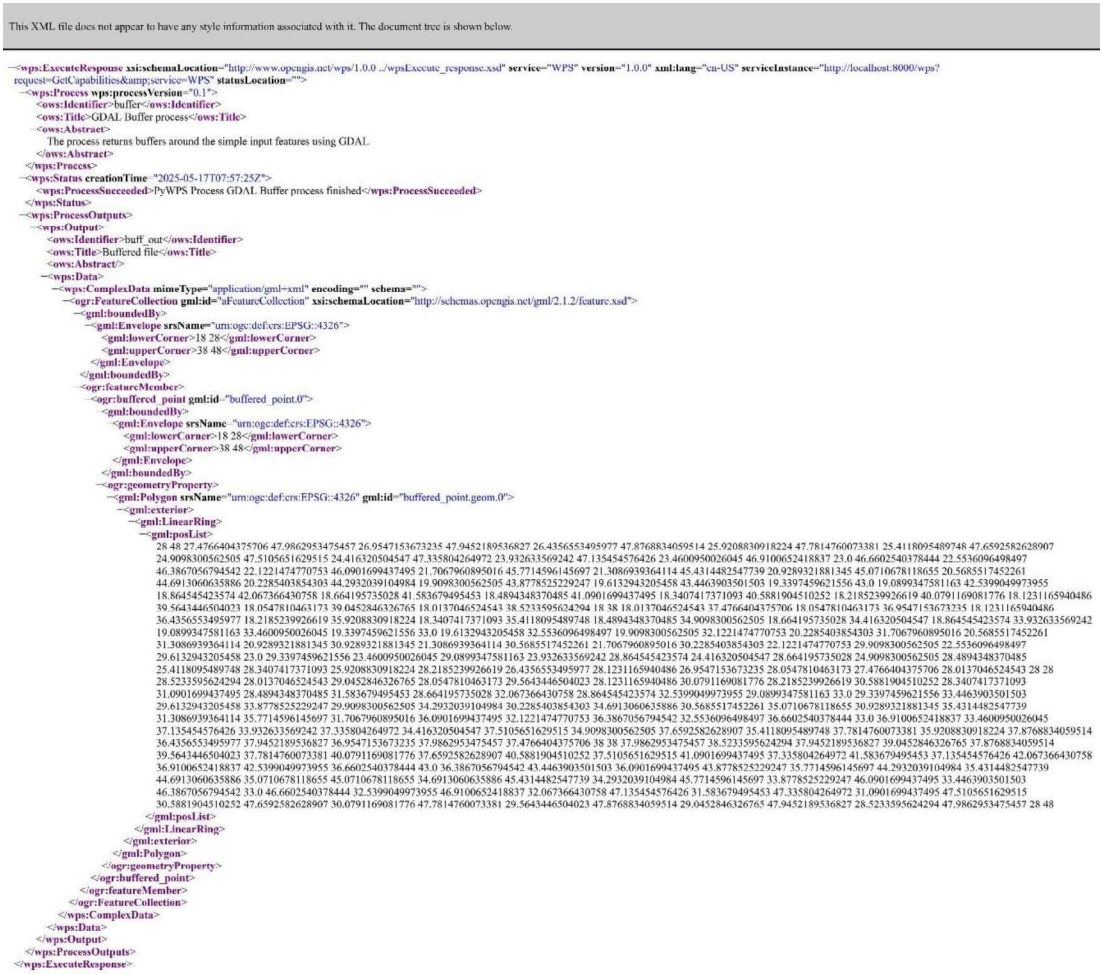
شکل ۱۰۷– پاسخ سرور به درخواست DescribeProcess

: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.

<wps:ExecuteResponse xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 ../wpsExecute_responseOnly.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="0.1">
<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="" schema="">
<ogr:FeatureCollection gml:id="aFeatureCollection" xsi:schemaLocation="http://schemas.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 27.47664044375706 47.98629534754 57 26.95471536732 35 47.94521895368 27 26.43565534959 77 47.87688340595 14 25.92088309182 24 47.7814760073381 25.4118095489748 47.65925826289 07
24.90983005625 05 47.51056516295 15 24.41632050454 7 47.3358042649 72 23.93263356924 2 47.1354545764 26 23.46009500260 45 46.91006521888 32 23.0 46.66025403784 44 22.5536096498497
46.3867056794542 22.1221474770753 46.09016994374 95 21.70679608950 16 45.77145961456 97 21.30869393641 14 45.43144825477 39 20.92893218813 45 45.07106781186 55 20.5685517452261
44.69130606358 86 20.2285403854303 44.2932039104984 19.9098300562505 43.8778525229247 19.6132943205458 43.4463903501503 19.3397459621556 43.0 19.0899347581163 42.5399049973955
18.8645454523574 42.0673664307 58 18.6641957350 28 41.5836794954 53 18.4894348370485 41.0901699437495 18.3407417371093 40.5881904510252 18.2185239926619 40.0791169081776 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 18.6641957350 28 34.4163205045 47 18.8645454523574 33.9326335692 42
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 21.3086939364114 30.5685517452261 21.7067960895016 30.2285403854303 22.1221474770753 29.9098300562505 22.5536096498497
29.6132943205458 23.0 29.3397459621556 23.4600950026045 29.0899347581163 23.9326335692424 28.8645454523574 24.4163205045 47 28.6641957350 28 24.9098300562505 28.4894348370485
25.4118095489748 28.3407417371093 25.9208830918224 28.2185239926619 26.4356553495977 28.1231165940486 26.9547153673235 28.0547810463173 27.4766404375706 28.0137046524543 28 28
28.5233595624294 28.0137046524543 29.0452846326765 28.0547810463173 29.5643446504023 28.1231165940486 30.0791169081776 28.2185239926619 30.5881904510252 28.3407417371093
31.0901699437495 28.4894348370485 31.5836794954 53 28.6641957350 28 32.0673664307 58 28.8645454523574 32.5399049973955 29.0899347581163 33.0 29.3397459621556 33.4463903501503
29.6132943205458 33.8778525229247 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 33.9326335692424 37.3358042649 72 34.4163205045 47 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 40.0791169081776 37.6592582628907 40.5881904510252 37.5105651629515 41.0901699437495 37.3358042649 72 41.5836794954 53 37.1354545764 26 42.0673664307 58
36.9100652418837 42.5399049973955 36.6602540378444 43.0 36.3867056794542 43.4463903501503 36.0901699437495 43.8778525229247 35.7714596145697 44.2932039104984 35.4314482547739
44.6913060635886 35.0710678118655 45.0710678118655 34.6913060635886 45.4314482547739 34.2932039104984 45.7714596145697 33.8778525229247 46.0901699437495 33.4463903501503
46.3867056794542 33.0 46.6602540378444 32.5399049973955 46.9100652418837 32.0673664307 58 47.1354545764 26 31.5836794954 53 47.3358042649 72 31.0901699437495 47.5105651629515
30.5881904510252 47.6592582628907 30.0791169081776 47.7814760073381 29.5643446504023 47.8768834059514 29.0452846326765 47.9452189536827 28.5233595624294 47.9862953475457 28 48
</gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</ogr:geometryProperty>
</ogr:buffered_point>
</ogr:featureMember>
</ogr:FeatureCollection>
</wps:ComplexData>
</wps:Data>
</wps:Output>
</wps:ProcessOutputs>
</wps:ExecuteResponse>

شکل ۱۰۸– پاسخ سرور به درخواست Execute

خروجی در فرمت GML و در سیستم مختصات WGS84 نمایش داده می‌شود. لازم به ذکر است خروجی بافر ایجاد شده به درجه می‌باشد (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 نقطه‌ی ورودی در سیستم تصویر UTM زون ۳۶ ایجاد خواهد شد (شکل ۱۰۹).

```
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: 04-18-2026

National Cartographic Center **Standard Department**

www.ncc.gov.ir **http://standard.ncc.gov.ir**

2026