

การประยุกต์ใช้แค็ตตาล็อกสินทรัพย์ทางภูมิสารสนเทศ

Spatial Temporal Asset Catalog Implementation



สิริญา แสนคำอ้อ

สหกิจศึกษาเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

(วิทยาการคอมพิวเตอร์)

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPATIAL TEMPORAL ASSET CATALOG IMPLEMENTATION



A COOPERATIVE EDUCATION SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENT FOR

THE DEGREE OF BACHELOR OF SCIENCE (COMPUTER SCIENCE)
DEPARTMENT OF COMPUTER SCIENCE, SCHOOL OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ **ACADEMIC YEAR 2022** อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อสหกิจศึกษา	การประยุกต์ใช้แค็ตตาล็อกสินทรัพย์ทางภูมิสารสนเทศ Spatial Temporal Asset Catalog Implementation
ชื่อนักศึกษา	นางสาวสิริญา แสนค าว์ รหัสนักศึกษา 62050238
ปริญญา	วิทยาศาสตร์บัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา	วิทยาการคอมพิวเตอร์
คณะ	วิทยาศาสตร์
มหาวิทยาลัย	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)
ปีการศึกษา	2565
อาจารย์ที่ปรึกษา	ผศ.ดร.กุลสวัสดิ์ จิตขจรวานิช

บทคัดย่อ

สหกิจศึกษานี้มีวัตถุประสงค์เพื่อศึกษาการประยุกต์ใช้แค็ตตาล็อกสินทรัพย์ทางภูมิสารสนเทศ โดยอ้างอิงจาก SpatioTemporal Asset Catalog Specification เพื่อพัฒนาเป็นส่วนเพิ่มเติมสำหรับ Vallaris Maps Platform ซึ่งจะเป็นอีกแนวทางหนึ่งในการเข้าถึง และค้นหาสินทรัพย์ทางภูมิสารสนเทศ โดยการนำเทคโนโลยี SpatioTemporal Asset Catalog (STAC) มาประยุกต์นั้น จะเป็นการทำแค็ตตาล็อกของข้อมูลที่มีคุณสมบัติของเวลา และสถานที่ ซึ่งเปรียบเสมือนสารบัญ หรือดัชนีของข้อมูลที่ใช้อ้างอิง ที่จะจัดเก็บสิ่งที่จะเชื่อมโยงไปยังข้อมูลที่ถูกเก็บไว้ในแหล่งจัดเก็บข้อมูล ทำให้การค้นหา และเข้าถึงข้อมูลของผู้ใช้ต้องการนั้น รวดเร็ว และใช้งานง่าย ทำให้สามารถตอบปัญหาเดิมที่การค้นหาข้อมูลนั้นมีเงื่อนไขที่จำกัด ยุ่งยาก ใช้เวลานาน และรูปแบบของการจัดท แค็ตตาล็อกข้อมูลนั้นไม่สามารถมีลูกเล่นในการจัดกลุ่มของข้อมูลตามที่ต้องการได้

ในโครงการสหกิจศึกษานี้ ได้มีการศึกษาข้อกำหนดต่างๆที่เกี่ยวข้อง และได้พัฒนา STAC API เพื่อใช้เป็นส่วนต่อประสานในการสร้าง และแก้ไขคอธิบายข้อมูลของสินทรัพย์ทางภูมิสารสนเทศให้อยู่ในรูปแบบของ STAC และสามารถค้นหา คอธิบายข้อมูล ซึ่งนำไปสู่สินทรัพย์ที่เก็บอยู่ในแหล่งข้อมูลได้ โดยมีกระบวนการพัฒนา STAC API ที่ได้พัฒนาขึ้นไปทดสอบกับ STAC API Validator เพื่อเป็นการรับรองว่า API ที่พัฒนานั้นเป็นไปตามความต้องการที่ข้อกำหนดได้ระบุไว้ และ STAC API ยังได้ถูกนำไปใช้ในโครงการจริงเพื่อตอบโจทย์ในการนำข้อมูลภูมิสารสนเทศมาจัดท ให้อยู่ในรูปแบบของแค็ตตาล็อกเพื่อใช้ในการเข้าถึงข้อมูล

คำสำคัญ : สินทรัพย์ทางภูมิสารสนเทศ, แค็ตตาล็อกของข้อมูล, STAC, REST API

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	Spatial Temporal Asset Catalog Implementation
Students	Miss Siriya Saenkhom-or Student ID 62050238
Degree	Bachelor of Science (Computer Science)
Department	Computer Science
School	Science
University	King Mongkut's Institute of Technology Ladkrabang (KMITL)
Academic Year	2022
Advisor	Asst.Prof.Dr. Kulsawasd Jitkajornwanich

Abstract

This cooperative education aims to study about Spatial Temporal Asset Catalog Implementation referencing SpatioTemporal Asset Catalog Specification to develop the new extension for Vallaris Maps Platform which will be another means to access and search spatial temporal asset. Implementing SpatioTemporal Asset Catalog (STAC) technology will be making a catalog for storing the metadata of the data that has properties of space and time, which that metadata will store the detail of where to get the actual data at the data store, leads to ease and convenience when searching and accessing the spatial temporal asset and also solves the limitation of searching and cataloging the spatial temporal data in the old days.

In this cooperative education project, various related specifications have been studied and STAC API has been implemented to be used as an interface for creating and editing metadata for the spatial temporal assets that follow STAC format and searching for metadata that leads to assets in the data store. The implemented STAC API also has been tested by STAC API Validator to assure that the API has followed the requirements and it also has been applied to real-world projects, bringing spatial temporal assets into the data catalog for another means to access the asset.

Keywords : Spatial Temporal Asset, Data Catalog, STAC, REST API

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การปฏิบัติงานสหกิจศึกษา และการจัดทำ สหกิจศึกษาเล่มนี้สำเร็จลุล่วงไปได้ด้วยดีโดยได้รับความร่วมมือ และความช่วยเหลือจากบุคคลหลายท่านที่ให้การปรึกษา และความช่วยเหลือในส่วนต่าง ๆ อันเป็นประโยชน์ต่อการปฏิบัติสหกิจศึกษาครั้งนี้

ขอกราบขอบพระคุณ ผศ.ดร.กุลสวัสดิ์ จิตขจรวานิช และ อ.สันธนะ อุ่ออุตมยิ่ง ที่ให้ความอนุเคราะห์ และคอยให้คำแนะนำ ในการปฏิบัติงานสหกิจศึกษาครั้งนี้

ขอกราบขอบพระคุณคุณประสงค์ ปทีปเพิ่มพงศ์ และบริษัท ไอบิทซ์ จ ทัต ที่ให้โอกาสข้าพเจ้าในการเข้ามาปฏิบัติสหกิจศึกษาในตำแหน่ง GIS Backend Developer และมีส่วนร่วมในการพัฒนาผลิตภัณฑ์ของบริษัท

ขอกราบขอบพระคุณคุณพงศกร อุตมบัว, คุณศตวรรษ อหารับ, คุณมรุตพงศ์ โฉมเฉลา และพี่ ๆ ทีมงานบริษัท ไอบิทซ์ จ ทัต ทุกคน ที่คอยให้คำแนะนำ ให้ความช่วยเหลือ ค ปรึกษา และชี้แนะแนวทาง ในการหา ความเข้าทฤษฎี และเทคโนโลยีต่าง ๆ อีกทั้งคำแนะนำ ในการพัฒนาระบบขึ้นมาให้สำเร็จลุล่วงไปได้ด้วยดี

ขอกราบขอบพระคุณคณาจารย์ทุกท่านในคณะวิทยาศาสตร์ ภาควิชาวิทยาการคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่คอยให้คำแนะนำ และคอยอบรมสั่งสอนให้ความรู้ตลอดเวลา 4 ปีจนกระทั่งสหกิจศึกษาสำเร็จไปได้ด้วยดี

สุดท้ายนี้ขอกราบขอบพระคุณครอบครัวที่คอยให้กำลังใจ และสนับสนุนเสมอมา รวมไปถึงรุ่นพี่ และมิตรสหายทุกท่านที่คอยให้ความช่วยเหลือ และให้คำปรึกษามาโดยตลอด

สิริญา แสนค ำฮ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง.....	ช
สารบัญรูป.....	ฉ
คำย่อสัญลักษณ์.....	ฎ
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของโครงการสหกิจศึกษา.....	2
1.3 ขอบเขตของโครงการสหกิจศึกษา.....	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	2
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	3
2.1 Spatiotemporal Asset.....	3
2.2 GeoJSON.....	3
2.3 Vallaris Maps Platform.....	5
2.4 OGC API - Features.....	5
2.5 SpatioTemporal Asset Catalog Specification.....	7
2.5.1 Item Specification.....	7
2.5.2 Catalog Specification.....	15
2.5.3 Collection Specification.....	19
2.5.4 STAC Model UML.....	26
2.5.5 Dynamic และ Static Catalogs.....	26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
3.7.3 การพัฒนา API ตามแบบสถาปัตยกรรม REST.....	59
บทที่ 4 ผลการวิจัยและการอภิปรายผล.....	65
4.1 ผลการทดสอบผ่าน STAC API Validator	65
4.2 ตัวอย่างการนำ STAC API ไปใช้ในโครงการจริง	66
4.2.1 ระบบสนับสนุนการตัดสินใจเพื่อการบริหารจัดการพื้นที่ภัยพิบัติ ของ GISTDA	67
4.2.2 ระบบ Platform บูรณาการข้อมูลสำหรับ Smart City ของ GISTDA	69
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....	79
5.1 สรุปผลการวิจัย.....	79
5.2 ข้อเสนอแนะ.....	79
เอกสารอ้างอิง.....	80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 2.1 แสดง Field เบื้องต้นที่เป็นส่วนประกอบของ STAC Item	8
ตารางที่ 2.2 แสดง Field เบื้องต้นที่เป็นส่วนประกอบของ Properties Object	9
ตารางที่ 2.3 แสดง Field ที่เป็นส่วนประกอบของ STAC Common Metadata	10
ตารางที่ 2.4 แสดง Field ที่เป็นส่วนประกอบของ Provider Object	11
ตารางที่ 2.5 แสดง Field ที่เป็นส่วนประกอบของ Link Object	12
ตารางที่ 2.6 แสดงประเภทความสัมพันธ์ที่เป็นไปได้ระหว่าง Item กับ เอกสารที่เชื่อมโยงไปถึง	12
ตารางที่ 2.7 แสดง Field เบื้องต้นที่เป็นส่วนประกอบของ Asset Object	13
ตารางที่ 2.8 แสดงชื่อ Role มาตรฐานที่ถูกกำหนดไว้	13
ตารางที่ 2.9 แสดง Field เบื้องต้นที่เป็นส่วนประกอบของ STAC Catalog	15
ตารางที่ 2.10 แสดง Field ที่เป็นส่วนประกอบของ Link Object	16
ตารางที่ 2.11 แสดงประเภทความสัมพันธ์ที่เป็นไปได้ระหว่าง Item กับ เอกสารที่เชื่อมโยงไปถึง	17
ตารางที่ 2.12 แสดง Media Types ที่ใช้สำหรับองค์ประกอบของ STAC	17
ตารางที่ 2.13 แสดง Field เบื้องต้นที่เป็นส่วนประกอบของ STAC Collection	19
ตารางที่ 2.14 แสดง Field ที่เป็นส่วนประกอบของ Provider Object	20
ตารางที่ 2.15 แสดง Field ที่เป็นส่วนประกอบของ Extent Object	21
ตารางที่ 2.16 แสดง Field ที่เป็นส่วนประกอบของ Spatial Extent Object	21
ตารางที่ 2.17 แสดง Field ที่เป็นส่วนประกอบของ Temporal Extent Object	22
ตารางที่ 2.18 แสดง Field เบื้องต้นที่เป็นส่วนประกอบของ Range Object	22
ตารางที่ 2.19 แสดง Field ที่เป็นส่วนประกอบของ Link Object	23
ตารางที่ 2.20 แสดงประเภทความสัมพันธ์ที่เป็นไปได้ระหว่าง Item กับ เอกสารที่เชื่อมโยงไปถึง	23
ตารางที่ 2.21 แสดง Field เบื้องต้นที่เป็นส่วนประกอบของ Asset Object	24
ตารางที่ 2.22 แสดง Endpoints ที่ขอก กำหนดCore ก กำหนดให้มี	27
ตารางที่ 2.23 แสดง Endpoints ที่ขอก กำหนดItem Search ก กำหนดให้มี	31
ตารางที่ 2.24 แสดง Query Parameters ที่เป็นตัวเลือกในการค้นหา	32
ตารางที่ 2.25 แสดง Endpoints ที่ขอก กำหนดCollections ก กำหนดให้มี	33
ตารางที่ 2.26 แสดง Endpoints ที่ขอก กำหนดFeatures ก กำหนดให้มี	33
ตารางที่ 2.27 แสดง Field เบื้องต้นที่เป็นส่วนประกอบของ Asset Object	34
ตารางที่ 3.1 แสดงรายละเอียดของโครงสร้างข้อมูลทีเก็บใน Collection “collections”	36

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง (ต่อ)

	หน้า
ตารางที่ 3.2 แสดงรายละเอียดของโครงสร้างข้อมูล provider	38
ตารางที่ 3.3 แสดงรายละเอียดของโครงสร้างข้อมูล extent.....	38
ตารางที่ 3.4 แสดงรายละเอียดของโครงสร้างข้อมูล spatial.....	39
ตารางที่ 3.5 แสดงรายละเอียดของโครงสร้างข้อมูล temporal.....	39
ตารางที่ 3.6 แสดงรายละเอียดของโครงสร้างข้อมูล asset	39
ตารางที่ 3.7 แสดงรายละเอียดของโครงสร้างข้อมูล range.....	40
ตารางที่ 3.8 แสดงรายละเอียดของโครงสร้างข้อมูลที่เก็บใน Collection “features”.....	40
ตารางที่ 3.9 แสดงรายละเอียดของโครงสร้างข้อมูล geometry	41
ตารางที่ 3.10 แสดงรายละเอียดของโครงสร้างข้อมูล asset	42
ตารางที่ 3.11 แสดงรายละเอียดของโครงสร้างข้อมูล common metadata.....	42
ตารางที่ 3.12 แสดงรายละเอียดของโครงสร้างข้อมูล provider	43
ตารางที่ 3.13 แสดงรายละเอียดของโครงสร้างข้อมูลที่เก็บใน Collection “queryables”	44
ตารางที่ 3.14 แสดงรายละเอียดของโครงสร้างข้อมูล property.....	45
ตารางที่ 3.15 แสดงรายละเอียดเกี่ยวกับ Use Case จัดการ STAC Catalog.....	48
ตารางที่ 3.16 แสดงรายละเอียดเกี่ยวกับ Use Case จัดการ STAC Collection.....	49
ตารางที่ 3.17 แสดงรายละเอียดเกี่ยวกับ Use Case จัดการ STAC Item.....	49
ตารางที่ 3.18 แสดงรายละเอียดเกี่ยวกับ Use Case Queryable.....	50
ตารางที่ 3.19 แสดงรายละเอียดเกี่ยวกับ Use Case ค้นหา STAC Item	50
ตารางที่ 3.20 แสดงการเปรียบเทียบของค์ประกอบของ Relational Database และ MongoDB.....	58
ตารางที่ 3.21 แสดงประเภทของ HTTP Request Method ที่ใช้งานบ่อย.....	61
ตารางที่ 3.22 แสดงหมวดหมู่ของ HTTP Status Code.....	63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

	หน้า
รูปที่ 2.1 ตัวอย่างของ GeoJSON FeatureCollection	4
รูปที่ 2.2 Vallaris Maps Platform	5
รูปที่ 2.3 Open Geospatial Consortium	5
รูปที่ 2.4 SpatioTemporal Asset Catalog	7
รูปที่ 2.5 ตัวอย่าง STAC Item	14
รูปที่ 2.6 ตัวอย่าง STAC Catalog.....	18
รูปที่ 2.7 ตัวอย่าง STAC Collection.....	25
รูปที่ 2.8 UML ของ STAC Model.....	26
รูปที่ 2.9 ตัวอย่าง Landing Page สำหรับ STAC API – Core.....	28
รูปที่ 3.1 Database Diagram ของแค็ตตาล็อกสินทรัพย์ทางภูมิสารสนเทศ 1 แค็ตตาล็อก	35
รูปที่ 3.2 System Architecture ของระบบในการบริหารจัดการ Spatial Temporal Asset Catalog	46
รูปที่ 3.3 Use Case Diagram ของระบบในการบริหารจัดการ Spatial Temporal Asset Catalog	47
รูปที่ 3.3 Sequence Diagram การค้นหาและเข้าถึงสินทรัพย์ทางภูมิสารสนเทศ	51
รูปที่ 3.4 ตัวอย่าง Index Table.....	54
รูปที่ 3.5 การทำ Replication ของ MongoDB.....	55
รูปที่ 3.6 การทำ Sharding กับ Collection ของ MongoDB	55
รูปที่ 3.7 GridFS ของ MongoDB	56
รูปที่ 3.8 ตัวอย่าง MongoDB Aggregation Pipeline.....	56
รูปที่ 3.9 ความสัมพันธ์ของแต่ละองค์ประกอบของ MongoDB	58
รูปที่ 3.10 ตัวอย่างโครงสร้างของ Document ใน MongoDB	59
รูปที่ 3.11 Flow การติดต่อของ Client และ Server ผ่าน REST API.....	61
รูปที่ 4.1 รายละเอียดคำสั่งที่ใช้ในการทดสอบ	65
รูปที่ 4.2 ผลลัพธ์ที่ได้จากการทดสอบ	66
รูปที่ 4.3 รายละเอียดเมนูของหน้าเว็บข้อมูลภัยพิบัติที่ ท่วมกรณีเลือกตามช่วงเวลาที่ให้มีให้	67
รูปที่ 4.4 รายละเอียดเมนูของหน้าเว็บข้อมูลภัยพิบัติที่ ท่วมกรณีกำหนดช่วงเวลาด้วยตนเอง	68
รูปที่ 4.5 รายละเอียดเมนูของหน้าเว็บบริการข้อมูลแค็ตตาล็อกข้อมูล (STAC).....	68
รูปที่ 4.6 Workflow การนำเข้าข้อมูลภาพถ่ายเทียม THEOS 1	69
รูปที่ 4.7 Pipeline ดาวเทียม THEOS 1	70

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

	หน้า
รูปที่ 4.8 ตัวอย่างภาพถ่ายเทียม THEOS 1 บริเวณจังหวัดขอนแก่น	70
รูปที่ 4.9 STAC Properties ของดาวเทียม THEOS 1.....	71
รูปที่ 4.10 STAC Catalog ของดาวเทียม THEOS 1	72
รูปที่ 4.11 STAC Item ของดาวเทียม THEOS 1	73
รูปที่ 4.12 หน้าจอแรกเมื่อผู้ใช้งานเข้ามาสู่ระบบการค้นหาภาพดาวเทียม	74
รูปที่ 4.13 เริ่มต้นด้วยการให้ผู้ใช้งานเลือกพื้นที่ที่ต้องการผ่านการสร้าง หรือ ส่งไฟล์ขึ้นมา.....	75
รูปที่ 4.14 เมื่อสร้างพื้นที่ที่ต้องการระบบจะแสดงพื้นที่ให้ผู้ใช้งานเห็น.....	75
รูปที่ 4.15 ผู้ใช้งานสามารถเลือกช่วงเวลาที่ต้องการ สามารถกำหนดเป็นช่วงเริ่มต้นและสิ้นสุด	76
รูปที่ 4.16 กำหนดประเภทของดาวเทียมที่ต้องการค้นหา แถบสีเขียวหมายถึง ต้องการค้นหาดาวเทียมนั้นๆ	76
รูปที่ 4.17 เมื่อเลือกและค้นหา ผลลัพธ์จะนำมาแสดงทางด้านขวา	77
รูปที่ 4.18 เลือกคุณภาพที่ต้องการในรูปแบบของ Thumbnail พร้อมคุณสมบัติต่างๆของภาพ	77
รูปที่ 4.19 สามารถเลือก Download ข้อมูลภาพได้ทันที.....	78
รูปที่ 4.20 เมื่อเลือกหลายภาพ สามารถไปจัดเก็บในระบบตะกร้าเพื่อ Download แบบหลายภาพ.....	78

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำย่อ/สัญลักษณ์

คำย่อ/สัญลักษณ์	คำอธิบาย
STAC	SpatioTemporal Asset Catalog คือแค็ตตาล็อกของสินทรัพย์ทางภูมิสารสนเทศ
JSON	JavaScript Object Notation เป็นข้อมูลรูปแบบ text ที่มีรูปแบบที่จะเก็บข้อมูลแบบ key, value
GeoJSON	มาจาก Geo + JSON เป็น Format ที่ออกแบบมาเพื่อใช้ในการแสดงคุณสมบัติทางภูมิศาสตร์อย่างง่ายพร้อมกับข้อมูลเชิงพื้นที่
OGC	Open Geospatial Consortium เป็นองค์กรนานาชาติที่พัฒนาและส่งเสริมมาตรฐานสำหรับการประมวลผลทางภูมิศาสตร์
WFS	Web Feature Service เป็นมาตรฐานที่อนุญาตให้ผู้ใช้งานสามารถดาวน์โหลดข้อมูลแผนที่ เพื่อนำมาแสดงผลหรือ นำมาใช้งานต่อ
OpenAPI	OpenAPI Specification เป็นมาตรฐานในการอธิบาย REST API ตัวหนึ่ง ที่ให้ทั้งคนและเครื่องคอมพิวเตอร์อ่านได้
RFC	Request for Comments เป็นเอกสารทางเทคนิคหรือบันทึกที่เกี่ยวข้องกับการทำงานของระบบอินเทอร์เน็ตเช่น Protocol, Procedure, Program หรือบันทึกการประชุม
Feature	คือสิ่งที่นำเสนอปรากฏการณ์บนโลก
FeatureCollection	คือชุดข้อมูลของสิ่งที่นำเสนอปรากฏการณ์บนโลก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากเทคโนโลยีที่ก้าวหน้า ทำให้ปัจจุบันข้อมูลที่มีอยู่บนโลกนี้เกิดขึ้นมาตลอดเวลา มีจำนวนมหาศาล หลายรูปแบบ มีมูลค่าในตัวเอง และต้องสามารถเชื่อถือได้ ในมุมมองของข้อมูลเชิงพื้นที่ก็เช่นกัน อาทิ ดาวเทียมสำรวจทรัพยากรธรรมชาติที่มีการส่งภาพถ่ายทางดาวเทียมมายังพื้นโลกในทุก ๆ วัน หรือเครื่องมือตรวจวัดตามจุดต่าง ๆ ที่มีการเก็บข้อมูลทุก ๆ นาที หรือทุก ๆ ชั่วโมง โดยข้อมูลเหล่านั้นก็สามารถมีได้หลายมิติ และรูปแบบที่สนใจ ทำให้ข้อมูลเชิงพื้นที่ที่เก็บอยู่ในแหล่งจัดเก็บข้อมูลนั้นมีจำนวนมหาศาล ซึ่งมูลค่าของข้อมูลเหล่านี้จะท าให้มนุษย์เข้าใจโครงสร้างลักษณะ หรือการเปลี่ยนแปลงของพื้นที่ที่สนใจมากยิ่งขึ้น อีกทั้งช่วยในการตัดสินใจ และวางแผน หรือคาดการณ์สิ่งที่จะเกิดขึ้นในอนาคตได้อย่างมีประสิทธิภาพ แต่ปัญหาที่เกิดขึ้นคือ การที่ผู้ใช้จะค้นหา และเข้าถึงข้อมูลที่น่าสนใจที่ถูกเก็บรวมกันกับข้อมูลมหาศาลนั้นเป็นไปได้ช้า และยุ่งยาก เป้าหมายของโครงการ สหกิจศึกษานี้จึงต้องการให้การค้นหา และเข้าถึงข้อมูลที่น่าสนใจของผู้ใช้เป็นไปได้อย่าง รวดเร็ว และเป็นไปตามมาตรฐานของการค้นหาข้อมูลในรูปแบบข้อมูลเชิงพื้นที่

Vallaris Maps Platform ซึ่งเป็นแพลตฟอร์มทางด้านภูมิสารสนเทศที่มีเครื่องมือสำหรับบริการจัดการข้อมูลเชิงพื้นที่ การวิเคราะห์ และนำไปแสดงผล ได้สังเกตเห็นถึงปัญหา และมูลค่าของข้อมูลและการเข้าถึงข้อมูลอีกรูปแบบหนึ่งที่สะดวกสบาย และรวดเร็ว จึงต้องการพัฒนาส่วนเพิ่มเติมที่จะใช้ในการจัดทำ Catalog ของข้อมูลที่สามารถใช้อ้างอิงข้อมูลนั้น ๆ ที่ถูกจัดเก็บใน Vallaris Maps Platform ได้อย่างง่าย และมีความถูกต้องสูง เพื่อเพิ่มความสะดวกในการค้นหาในภายหลัง และกำหนดให้เป็นอีกรูปแบบหนึ่งในการเข้าถึงข้อมูลของผู้ใช้งานใน Vallaris Maps Platform

การนาเทคโนโลยี SpatioTemporal Asset Catalog (STAC) มาประยุกต์ สามารถเข้ามาช่วยในการจัดทำ Catalog ซึ่งเปรียบเสมือนสารบัญ หรือดัชนีของข้อมูลที่ใช้อ้างอิง ที่จะจัดเก็บสิ่งที่จะเชื่อมโยงไปยังข้อมูลที่ถูกเก็บไว้ในแหล่งจัดเก็บข้อมูล ทำให้การค้นหา และเข้าถึงข้อมูลของผู้ใช้ต้องการนั้นรวดเร็ว และใช้งานง่าย ทำให้สามารถตอบปัญหาที่กล่าวไว้ข้างต้นได้ โดยการพัฒนาเทคโนโลยี STAC เป็นส่วนเพิ่มเติมสำหรับ Vallaris Maps Platform นั้นจะเป็นการสร้างทางเลือกใหม่ในการเข้าถึงข้อมูล ซึ่งเป็นการเพิ่มมูลค่าให้กับข้อมูลที่ถูกเก็บอยู่ใน Platform และตัว Platform เอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ของโครงการสหกิจศึกษา

1. เพื่อศึกษาการท าCatalog ของข้อมูลเชิงพื้นที่โดยอ้างอิงตามมาตรฐานของ STAC Specification
2. เพื่อจัดทำ ระบบในการบริหารจัดการ STAC เพื่อใช้ในการสร้าง Catalog ของข้อมูล และระบบในการค้นหาข้อมูลใน STAC เพื่อให้ผู้ใช้งานสามารถค้นหาข้อมูลที่ต้องการได้สะดวก และรวดเร็วยิ่งขึ้น
3. เพื่อสร้างส่วนเพิ่มเติมสำหรับ Vallaris Maps Platform ในส่วนของ Catalog ในการจัดเก็บข้อมูลที่ใช้อ้างอิง และสิ่งที่เกี่ยวข้องไปยังข้อมูลในแหล่งจัดเก็บข้อมูล และค้นหาข้อมูลที่ต้องการโดยแสดงผลในรูปแบบที่ก าหนด
4. เพื่อพัฒนาวิธีการเข้าถึงข้อมูลเชิงพื้นที่ในอีกรูปแบบหนึ่งของ Vallaris Maps Platform ที่เข้าใจง่าย สะดวกสบาย รวดเร็ว และมีรูปแบบเป็นไปตามมาตรฐาน

1.3 ขอบเขตของโครงการสหกิจศึกษา

1. ออกแบบ และพัฒนาระบบให้เข้ากับการใช้งานบน Vallaris Maps Platform
2. จัดทำฟังก์ชันในการจัดการ Catalog ของข้อมูลที่ใช้อ้างอิงที่มีรูปแบบหรือตามที่ใช้ต้องการ และฟังก์ชันค้นหา เพื่อให้สามารถค้นหาข้อมูลที่น่าสนใจได้ โดยมีรูปแบบตาม STAC Specification และมาตรฐาน OGC API
3. พัฒนาระบบในการบริหารจัดการ STAC และค้นหาข้อมูลที่เกี่ยวข้องใน STAC สำหรับส่วนเพิ่มเติม สำหรับการเข้าถึงข้อมูล ซึ่งเป็นส่วนหนึ่งของ Vallaris Maps Platform

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถใช้ส่วนเพิ่มเติมที่ได้พัฒนาขึ้นในการค้นหาข้อมูลที่ใช้อ้างอิงกับข้อมูลที่ถูกจัดเก็บใน Vallaris Maps Platform ได้ ภายใน Catalog ที่ถูกสร้างขึ้นตามแบบที่มาตรฐานระบุไว้
2. สามารถนำส่วนเพิ่มเติมที่พัฒนาขึ้น ไปเชื่อมต่อกับระบบส่วนอื่น ๆ ใน Vallaris Maps Platform หรือเชื่อมต่อกับระบบภายนอกในภายหลังได้
3. สามารถทำให้การค้นหา และเข้าถึงข้อมูลเชิงพื้นที่ใน Vallaris Maps Platform เป็นไปอย่างสะดวกสบาย ใช้งานง่าย และรวดเร็ว
4. ได้เรียนรู้ ผังผน และเข้าใจการน าเทคโนโลยี SpatioTemporal Asset Catalog (STAC) มาใช้ในการจัดทำ Catalog ของข้อมูลเชิงพื้นที่
5. ได้เปิดประสบการณ์ใหม่ ๆ ด้านการจัดการกับข้อมูลเชิงพื้นที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

การประยุกต์ใช้แค็ตตาล็อกสินทรัพย์ทางภูมิสารสนเทศ (Spatial Temporal Asset Catalog Implementation) เพื่อพัฒนาส่วนเพิ่มเติมสำหรับ Vallaris Maps Platform เป็นการจัดหาส่วนที่ใช้บริหารจัดการ และค้นหาข้อมูลที่ใช้อ้างอิงไปยังข้อมูลที่ถูกเก็บไว้ในแหล่งจัดเก็บข้อมูล ซึ่งมีหลักการ ทฤษฎีที่เกี่ยวข้องและเทคโนโลยีที่นำมาใช้ดังนี้

2.1 Spatiotemporal Asset

Spatiotemporal Asset หรือ สินทรัพย์ทางภูมิสารสนเทศ คือ ไฟล์ที่นำเสนอข้อมูลเกี่ยวกับโลก ที่ถูกเก็บข้อมูล ณ สถานที่ และเวลาที่แน่นอน แต่แรกเริ่ม Spatiotemporal Asset ให้ความสำคัญจุดหลักที่ภาพถ่ายดาวเทียม แต่ในปัจจุบันสามารถนำใช้ได้กับข้อมูลที่หลากหลายมากขึ้นที่อาจจะรวมถึงแหล่งข้อมูลต่าง ๆ เช่น เครื่องบินและโดรน และข้อมูลอื่น ๆ เช่น Hyperspectral optical, Synthetic Aperture Radar (SAR), วิดีโอ, Point Cloud, Lidar, DEM, Vector Data หรือแม้กระทั่ง Machine Learning labels และสามารถรวมไปถึงผลิตภัณฑ์จากดาวเทียม เช่น NDVI และ Mosaics เป็นต้น

2.2 GeoJSON

GeoJSON คือ รูปแบบการแลกเปลี่ยนข้อมูลทางภูมิศาสตร์ ซึ่งมีพื้นฐานมาจาก JavaScript Object Notation (JSON) โดยสามารถกำหนด JSON object ได้หลายประเภท เพื่อที่จะใช้ในการนำเสนอคุณลักษณะทางภูมิศาสตร์ คุณสมบัติอื่น ๆ และขอบเขตเชิงพื้นที่ของข้อมูลนั้น ๆ

GeoJSON อาจใช้เป็นตัวแทนของขอบเขตของพื้นที่ (Geometry) คุณลักษณะที่มีขอบเขตเชิงพื้นที่ (Feature) หรือรายการของคุณลักษณะ (FeatureCollection) โดย GeoJSON สามารถรองรับประเภทของ Geometry ดังนี้ Point, LineString, Polygon, MultiPoint, MultiLineString, MultiPolygon, และ GeometryCollection

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
  "type": "FeatureCollection",
  "features": [{
    "type": "Feature",
    "geometry": {
      "type": "Point",
      "coordinates": [102.0, 0.5]
    },
    "properties": {
      "prop0": "value0"
    }
  }, {
    "type": "Feature",
    "geometry": {
      "type": "LineString",
      "coordinates": [
        [102.0, 0.0],
        [103.0, 1.0],
        [104.0, 0.0],
        [105.0, 1.0]
      ]
    },
    "properties": {
      "prop0": "value0",
      "prop1": 0.0
    }
  }, {
    "type": "Feature",
    "geometry": {
      "type": "Polygon",
      "coordinates": [
        [
          [100.0, 0.0],
          [101.0, 0.0],
          [101.0, 1.0],
          [100.0, 1.0],
          [100.0, 0.0]
        ]
      ]
    },
    "properties": {
      "prop0": "value0",
      "prop1": {
        "this": "that"
      }
    }
  }
]
}

```

รูปที่ 2.1 ตัวอย่างของ GeoJSON FeatureCollection

(ที่มา: <https://en.wikipedia.org/wiki/GeoJSON>)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 Vallaris Maps Platform



รูปที่ 2.2 Vallaris Maps Platform

(ที่มา: <https://i-bitz.co.th/vallaris/>)

Vallaris Maps Platform เป็นแพลตฟอร์มทางด้านภูมิศาสตร์ ซึ่งเป็นเครื่องมือสำหรับบริหารจัดการข้อมูลทางด้านภูมิศาสตร์ และช่วยอำนวยความสะดวกในการจัดการข้อมูลด้านภูมิศาสตร์แก่หน่วยงานทั้งภาครัฐและเอกชนซึ่งกระจายกันอยู่หลายหลากที่ ไม่ว่าจะเป็น ไฟล์ หรือ ฐานข้อมูล โดย Vallaris Maps Platform จะมีเครื่องมือสำหรับนำเข้าข้อมูล ทำให้ข้อมูลถูกจัดเก็บอย่างมีระเบียบ เมื่อข้อมูลถูกจัดเก็บอย่างมีระเบียบแล้ว ช่วยให้สามารถนำข้อมูลทั้งหมดที่เก็บอยู่ใน Vallaris Maps Platform เอง และจากแหล่งข้อมูลภายนอกมาวิเคราะห์ร่วมกันได้ โดยผู้ใช้งานสามารถออกแบบการแสดงผลบนแผนที่ได้ด้วยตนเอง สิ่งสำคัญคือผู้ใช้งานสามารถเรียกบริการแผนที่ (Map Service) และบริการข้อมูล (Data Service) ไปใช้งานได้โดยง่าย สะดวก และรวดเร็ว

2.4 OGC API - Features



รูปที่ 2.3 Open Geospatial Consortium

(ที่มา: https://en.wikipedia.org/wiki/Open_Geospatial_Consortium)

OGC API เป็นกลุ่มของมาตรฐานที่พัฒนาขึ้นมาโดยกลุ่ม Open Geospatial Consortium เพื่อให้ทุกคนสามารถเข้าถึง หรือให้บริการข้อมูลภูมิสารสนเทศผ่านทางเว็บ มาตรฐานกลุ่มนี้ถูกพัฒนาขึ้นมาบน OGC Web Service ซึ่งเป็นมาตรฐานเดิม แต่ได้ออกแบบในรูปแบบ APIs ซึ่งจะเป็ประโยชน์ต่อการพัฒนาเว็บในสมัยใหม่ และมีลักษณะเป็นชุดเล็กๆ ทั้ในแง่ยต่อการที่นักพัฒนา และผู้ใช้งานจะอ้างอิงในการพัฒนาระบบหรือเข้าถึง และแลกเปลี่ยนข้อมูลทางภูมิสารสนเทศได้ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OGC API - Features เป็นหนึ่งในหมวดของกลุ่มมาตรฐาน OGC API ซึ่งเป็นมาตรฐานแบบหลายส่วนที่มีความสามารถในการสร้าง แก้ไข และค้นหาข้อมูลเชิงพื้นที่บนเว็บ และระบุข้อกำหนด และค าแนะน า ส าหรับ APIs ที่ต้องการปฏิบัติตามมาตรฐานการให้บริการข้อมูลคุณลักษณะ ในส่วนแกนหลักของมาตรฐานมีการอธิบายถึงความสามารถที่จ าเป็นที่ทุกบริการที่มีการปฏิบัติตามมาตรฐานนี้ต้องรองรับ และจ ากัดการเข้าถึงข้อมูลเชิงพื้นที่ ส าหรับความสามารถอื่นเพิ่มเติมที่เฉพาะเจาะจงจะถูกระบุไว้ในส่วนเพิ่มเติม ซึ่งตัวอย่างความสามารถที่คาดการณ์ไว้ในอนาคต ได้แก่ การรองรับการสร้าง และแก้ไขข้อมูล การรองรับ Data Model ที่ซับซ้อนมากขึ้น การสืบค้นที่ดีขึ้น และระบบพิกัดอ้างอิงเพิ่มเติม

โดยความตั้งใจของ OGC API Features นั้นต้องการให้ง่ายและมีความทันสมัยภายใต้รูปแบบของ Web Feature Services เดิม ซึ่ง OGC API Features ได้เปลี่ยนให้เป็นรูปแบบบริการสมัยใหม่ ที่มีการเปลี่ยนแปลงในเรื่องต่าง ๆ ดังนี้

1. Architecture: OGC API Features สนับสนุนการสื่อสารบน HTTP และ HTTPS. ส าหรับ WFS นั้น HTTP เป็นช่องทางหลักที่ใช้ในการสื่อสารข้อมูล

2. Encodings: WFS เดิมที่ผูกติดกับ XML อย่างมาก ขณะที่ OGC API Features เลือกให้สนับสนุน HTML, JSON และ XML เนื่องจากเป็นรูปแบบการเข้ารหัสข้อมูลที่ได้รับความนิยมปัจจุบัน (ไม่ได้มีข้อบังคับว่าจะต้องเป็นอย่างใดอย่างหนึ่ง และยังคงรองรับการเข้ารหัสในรูปแบบอื่นเช่นกัน) ซึ่งการสื่อสารบน HTTP นั้นถูกออกแบบมาให้สนับสนุนหลากหลายรูปแบบและก าหนดเงื่อนไขให้เหมาะแก่การแสดงผลบนลูกข่ายให้สามารถจัดการกับข้อมูลได้เหมาะสมที่สุด

3. Reuse: การใช้งานส่วนประกอบต่าง ๆ ได้ถูกก าหนดแบบย่อย ๆ ใน OGC API ที่เหมาะสมกับการใช้งานและการพัฒนา

4. Modularization: ถึงแม้ว่า WFS 2.0 และการค้นหาข้อมูลภายใต้ WFS 2.0 นั้นมีประสิทธิภาพสูง แต่ก็มี ความซับซ้อนต่อการใช้งาน ขณะที่ OGC API Features เป็นส่วนย่อย ๆ เพื่อสร้างเป็นองค์ประกอบใหญ่ คือ ในส่วนแรก เป็น แกนหลัก (Core) ที่ออกแบบมาเพื่อเข้าถึงข้อมูลโดยไม่กล่าวถึงการปรับปรุงแก้ไขข้อมูล และยังแยกออกจากการค้นหาและระบบพิกัด เช่นกัน

5. Schemas: เดิมที WFS จะต้องมี XML Schemas ควบคุมเพื่อให้เอกสารถูกต้อง แต่ส าหรับ OGC API Features นั้นไม่จ าเป็นต้องมี schemas ควบคุมที่ตายตัวดังเดิม

6. Security: WFS 2.0 ไม่ได้ระบุวิธีการรักษาความปลอดภัยของข้อมูล การแก้ไขถูกก าหนดใน OpenAPI ที่ใช้ Security schemas เช่น OAuth2 เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 SpatioTemporal Asset Catalog Specification



รูปที่ 2.4 SpatioTemporal Asset Catalog

(ที่มา: <https://github.com/radiantearth/stac-spec>)

SpatioTemporal Asset Catalog (STAC) เป็นการจัดทำ Catalog ของข้อมูลที่ใช้อ้างอิงไปยังสินทรัพย์ทางภูมิสารสนเทศ โดยมีข้อกำหนดในการพัฒนาตาม SpatioTemporal Asset Catalog Specification ซึ่งเป็นข้อกำหนดที่มีจุดมุ่งหมายในการสร้างมาตรฐานในการจัดโครงสร้างในการให้บริการ และค้นหาข้อมูลที่ใช้อ้างอิงไปยังสินทรัพย์ทางภูมิสารสนเทศ โดย STAC ได้ถูกออกแบบอย่างให้มีคุณสมบัติที่น้อยที่สุด และมีกลไกที่ยืดหยุ่น เพื่อรองรับการใช้งานที่หลากหลาย

STAC Specifications จึงเป็นประโยชน์ และมีความสำคัญต่อผู้ให้บริการข้อมูลทางภูมิสารสนเทศ เนื่องจากสามารถนำข้อกำหนด หรือมาตรฐาน และ API ที่ได้ถูกออกแบบไว้ ไปใช้เป็นแนวปฏิบัติในการพัฒนาระบบขององค์กรได้ และ STAC Specification ยังเป็นประโยชน์ต่อผู้ที่ต้องการเข้าถึงข้อมูลทางภูมิสารสนเทศ เนื่องจากสามารถใช้ Libraries และเครื่องมือที่ถูกพัฒนาขึ้นในการเข้าถึงข้อมูลที่ใช้อ้างอิงได้ทันที

โดยแกนหลักของ STAC Specifications ได้กำหนดประเภทของ JSON Object ที่เกี่ยวข้อง ด้วยความสัมพันธ์ที่เชื่อมโยงกัน ซึ่งประกอบไปด้วย Item, Catalog และ Collection ซึ่งมีรายละเอียดดังนี้

2.5.1 Item Specification

STAC Item เป็น GeoJSON Feature ที่สามารถมีคุณสมบัติหรือ field เพิ่มเติมนอกจากที่กำหนดไว้ได้ อีกทั้งยังเป็น object ที่มีความสำคัญที่สุดในระบบ STAC เนื่องจาก STAC Item ประกอบไปด้วยคำอธิบายข้อมูลที่อธิบายความสะดวกในการค้นหาข้อมูลที่สนใจ และ URI สำหรับเข้าถึงสินทรัพย์ที่เกี่ยวข้องที่อยู่ในแหล่งจัดเก็บข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดย Field เบื้องต้นที่เป็นส่วนประกอบของ STAC Item มีดังนี้

ตารางที่ 2.1 แสดง Field เบื้องต้นที่เป็นส่วนประกอบของ STAC Item

ชื่อ Field	ประเภทข้อมูล	จำเป็น	คำอธิบาย
type	string	✓	ประเภทของ GeoJSON Object สำหรับ Item ต้องมีค่าเป็น 'Feature'
stac_version	string	✓	เวอร์ชัน STAC ที่ Item ใช้
stac_extensions	[string]		ส่วนขยายที่ Item ใช้
id	string	✓	ใช้สำหรับระบุตัว Item ควรเป็นค่าที่ไม่ซ้ำใน Collection ที่ Item นั้นอยู่
geometry	GeoJSON Geometry Object null	✓	ระบุ Footprint ของสินทรัพย์ที่ Item อ้างถึง รูปแบบอ้างอิงตาม RFC 7946, หมวดที่ 3.1 โดย Footprint ควรเป็น GeoJSON geometry พื้นฐาน และพิกัดถูกระบุในลักษณะ Longitude/Latitude หรือ Longitude/Latitude/Elevation ตามมาตรฐาน WGS 84
bbox	[number]	✓	(*จำเป็นต้องระบุถ้า Geometry ไม่ใช่ null) เป็น Bounding Box ของสินทรัพย์ที่ Item อ้างถึง รูปแบบอ้างอิงตาม RFC 7946, หมวดที่ 5
properties	Properties Object	✓	รายการคอธิบายข้อมูลเพิ่มเติมสำหรับ Item
links	[Link Object]	✓	รายการของ Link Object ที่นำไปสู่ทรัพยากรที่ต้องการหรือ URLs ที่เกี่ยวข้อง
assets	Map<string, Asset Object>	✓	รายการของ Asset Object ที่สามารถดาวน์โหลดได้ โดยชื่อ key ห้ามซ้ำกัน

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

collection	string		Id ของ STAC Collection ที่ Item อ้างถึง จำเป็นต้องมี ถ้ามีความสัมพันธ์กับ Collection โดย Field ที่ ผู้ใช้งานสามารถค้นหา Item ที่เป็นของ Collection ที่ ต้องการได้
------------	--------	--	---

Properties Object

โดย Field ที่ใช้เป็น คอธิบายข้อมูลเพิ่มเติมสามารถถูกเพิ่มเข้าไปใน GeoJSON Object Properties นี้ได้ โดย Field เดียวที่ จำเป็นต้องมีคือ datetime ตามตารางด้านล่าง

ตารางที่ 2.2 แสดง Field เบื้องต้นที่เป็นส่วนประกอบของ Properties Object

ชื่อ Field	ประเภท ข้อมูล	จำเป็น	คอธิบาย
datetime	string null	✓	เป็นวันและเวลาของสินทรัพย์ที่สามารถค้นหาได้ โดยต้องเป็นเวลาแบบ UTC และมีรูปแบบตาม RFC 3339, หมวดที่ 5.6 สามารถเป็น null ได้ แต่ต้องมี field start_datetime และ end_datetime แทน

สำหรับ Field อื่นที่เพิ่มเติมเข้าไป ควรเป็น Field ที่เกี่ยวข้อง และจำเป็นต่อการค้นหา ซึ่งตัวอย่าง Field มีดังนี้

- **STAC Common Metadata** เป็นรายการของ Field ที่ใช้กันทั่วไปในทุกการใช้งาน
- **Extensions** เป็น Field ที่มีความจำเพาะเจาะจงมากขึ้น เช่น EO (Electro-Optical Extension), View (View Geometry Extension)
- **Custom Extensions** สามารถเพิ่ม Field นอกเหนือจากที่กล่าวมาได้ โดยแนะนำให้ใช้หลาย ๆ Field ในการอธิบาย แทนการใช้ Nested Object

STAC Common Metadata

เป็น Field ที่ใช้ทั่วไปใน STAC Item properties แต่ก็สามารถใช้กับส่วนอื่นได้ เช่น Item Asset หรือ Collection Asset โดย Field เหล่านี้ไม่จำเป็นต้องมี นอกเสียจากว่าจะถูกกำหนดไว้ในข้อกำหนดส่วนนั้น ๆ เช่น datetime ที่จำเป็นต้องมีใน STAC Item เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 แสดง Field ที่เป็นส่วนประกอบของ STAC Common Metadata

ชื่อ Field	ประเภทข้อมูล	คำอธิบาย
title	string	ชื่อเรียกสำหรับแสดงให้ผู้ใช้งาน
description	string	รายละเอียดที่ใช้อธิบาย Item สามารถใช้ syntax CommonMark 0.29 ในการอธิบายแบบ rich text
datetime	string null	เป็นวันและเวลาของสินทรัพย์ที่สามารถค้นหาได้ โดยต้องเป็นเวลาแบบ UTC และมีรูปแบบตาม RFC 3339, หมวดที่ 5.6 สามารถเป็น null ได้ แต่ต้องมี field start_datetime และ end_datetime แทน
created	string	วันและเวลาที่สร้างข้อมูลที่กล่าวถึง ในแบบ UTC และมีรูปแบบตาม RFC 3339, หมวดที่ 5.6
updated	string	วันและเวลาที่แก้ไขข้อมูลที่กล่าวถึง ในแบบ UTC และมีรูปแบบตาม RFC 3339, หมวดที่ 5.6
start_datetime	string	วันและเวลาเริ่มต้นของ Item ในแบบ UTC และมีรูปแบบตาม RFC 3339, หมวดที่ 5.6
end_datetime	string	วันและเวลาสิ้นสุดของ Item ในแบบ UTC และมีรูปแบบตาม RFC 3339, หมวดที่ 5.6
license	string	การอนุมัติของ Item สามารถมีค่าได้ทั้ง SPDX License identifier, 'various' ถ้ามีการใช้หลายการอนุมัติ หรือ 'proprietary' ส สำหรับกรณีอื่น ควรจะมีการระบุ Field นี้ใน Collection ด้วย
providers	[Provider Object]	รายการของผู้ให้บริการ ซึ่งสามารถระบุองค์กรที่เป็นผู้เก็บภาพ และประมวลผลข้อมูล หรือเป็นผู้เก็บและให้บริการข้อมูล ควรเรียงล าดับ โดยให้ผู้ให้บริการล่าสุดเป็นล าดับสุดท้ายในรายการ
platform	string	ชื่อของแพลตฟอร์มที่ไมซ์ แท็บเล็ต ที่เครื่องมือถูกติดตั้งอยู่
instruments	[string]	ชื่อของเครื่องมือ หรือเซนเซอร์ที่ใช้ในการจับภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้เผยแพร่ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

constellation	string	ชื่อกลุ่มของแพลตฟอร์มที่มีลักษณะคล้ายกัน
mission	string	ชื่อของเป้าหมายว่าต้องการเก็บข้อมูลใด
gsd	number	Ground Sample Distance ที่เซนเซอร์ตรวจจับได้ มีหน่วยเป็น เมตร และมีค่ามากกว่า 0

Provider Object

เป็นข้อมูลเกี่ยวกับผู้ให้บริการ โดยสามารถระบุองค์กรที่เป็นผู้เก็บภาพ และประมวลผลข้อมูล หรือเป็นผู้เก็บและให้บริการข้อมูล

ตารางที่ 2.4 แสดง Field ที่เป็นส่วนประกอบของ Provider Object

ชื่อ Field	ประเภทข้อมูล	จำเป็น	คำอธิบาย
name	string	✓	ชื่อองค์กร หรือบุคคล
description	string		รายละเอียดที่ใช้อธิบายผู้ให้บริการเพิ่มเติม สามารถใช้ Syntax CommonMark 0.29 ในการอธิบายแบบ Rich Text
roles	[string]		หน้าที่ของผู้ให้บริการ ค่าที่เป็นไปได้ ได้แก่ 'licensor', 'producer', 'processor' หรือ 'host'
url	string		URL ที่นำไปสู่หน้า Homepage ของผู้ให้บริการ

Link Object

เป็น Object ที่อธิบายถึงความสัมพันธ์กับ Entity อื่น ซึ่งแนะนำให้ผู้ให้บริการข้อมูลพยายามระบุ links ที่เชื่อมโยงกันให้ได้มากที่สุดเพื่อให้ผู้ใช้งานสามารถเข้าใจว่ามี Entity ใด ที่สามารถเชื่อมโยงไปถึงได้อีกบ้าง โดยรายละเอียดที่ต้องระบุใน Link Object แต่ละตัวมีดังตารางด้านล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.5 แสดง Field ที่เป็นส่วนประกอบของ Link Object

ชื่อ Field	ประเภทข้อมูล	จำเป็น	คำอธิบาย
href	string	✓	Link ในรูปแบบ URL สามารถเป็นได้ทั้ง Relative และ Absolute Link
rel	string	✓	ความสัมพันธ์ระหว่างเอกสารที่เรียกดูอยู่กับเอกสารที่ถูกเชื่อมโยงไปถึง
type	string		Media Type ของ Entity ที่อ้างอิงถึง
title	string		ชื่อที่สามารถอ่านเข้าใจได้ส สำหรับใช้แสดงกับ Link

ประเภทของความสัมพันธ์ระหว่าง Item กับเอกสารอื่น

ใน Field `rel` ของ Link Object ที่ใช้ในการระบุความสัมพันธ์ระหว่างเอกสารที่เรียกดูอยู่กับเอกสารที่ถูกเชื่อมโยงไปถึง สามารถมีความสัมพันธ์ที่เป็นไปได้ดังนี้

ตารางที่ 2.6 แสดงประเภทความสัมพันธ์ที่เป็นไปได้ระหว่าง Item กับ เอกสารที่เชื่อมโยงไปถึง

ประเภท	คำอธิบาย
self	(*แนะนำ ให้มีอย่างมาก) Absolute URL ที่เชื่อมโยงไปยัง Item ถ้าสามารถเข้าถึงได้ด้วย Public URL
root	URL ที่เชื่อมโยงไปยัง Root STAC Entity
parent	URL ที่เชื่อมโยงไปยัง STAC Entity ล ดับค่อนหน้า
collection	(*แนะนำ ให้มีอย่างมาก) URL ที่เชื่อมโยงไปยัง Collection โดยควรใช้แบบ Absolute URL จ าเป็นต้องมีความสัมพันธ์นี้ถ้าใน Item มี Field 'collection'
derived_from	URL ที่เชื่อมโยงไปยัง STAC Item ที่ถูกใช้เป็นข้อมูลเข้าเมื่อสร้าง Item

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Asset Object

เป็น object ที่ประกอบด้วย URI ที่นำพาไปหาข้อมูลที่เกี่ยวข้องกับ Item ที่สามารถดาวน์โหลดได้ ซึ่งสามารถมี Field เพิ่มเติมนอกเหนือจาก Field เบื้องต้นได้ โดย Field เบื้องต้นของ Item มีดังตารางด้านล่างนี้

ตารางที่ 2.7 แสดง Field เบื้องต้นที่เป็นส่วนประกอบของ Asset Object

ชื่อ Field	ประเภทข้อมูล	จำเป็น	คำอธิบาย
href	string	✓	URI ที่นำไปสู่สินทรัพย์ สามารถเป็นได้ทั้ง Relative และ Absolute URI
title	string		ชื่อเรียกสำหรับแสดงให้ผู้ใช้งาน
description	string		รายละเอียดเกี่ยวกับสินทรัพย์เพิ่มเติม
type	string		Media Type ของสินทรัพย์
roles	[string]		หน้าที่ หรือคุณสมบัติของสินทรัพย์นั้น คล้ายกับ 'rel' ของ links

ประเภทของ Asset Role

Field 'role' สามารถมีค่าเป็นอะไรก็ได้ อย่างไรก็ตาม มี Role มาตรฐานถูกกำหนดไว้ดังตารางด้านล่าง

ตารางที่ 2.8 แสดงชื่อ Role มาตรฐานที่ถูกกำหนดไว้

ชื่อ Role	คำอธิบาย
thumbnail	เป็นสินทรัพย์ที่แสดงถึงภาพ Thumbnail ของ Item โดยปกติแล้วจะเป็น True Color Image มี Resolution ต่ำ และเป็นไฟล์นามสกุล JPEG หรือ PNG ให้เหมาะสมกับการแสดงผลบนหน้าเว็บ
overview	เป็นสินทรัพย์ที่แสดงถึงภาพ หรือภาพรวมที่ใหญ่กว่า Thumbnail ของ Item เช่น ข้อมูลภาพ True Color ที่ประกอบด้วยหลาย Band
data	ข้อมูลของสินทรัพย์นั้น
metadata	คำอธิบายข้อมูลที่ใช้อธิบายข้อมูลเพิ่มเติมใน Item

ตัวอย่าง GeoJSON สำหรับข้อมูล Item

```

{
  "stac_version": "1.0.0",
  "stac_extensions": [],
  "type": "Feature",
  "id": "20201211_223832_CS2",
  "bbox": [
    172.91173669923782,
    1.3438851951615003,
    172.95469614953714,
    1.3690476620161975
  ],
  "geometry": {
    "type": "Polygon",
    "coordinates": [
      [
        [
          172.91173669923782,
          1.3438851951615003
        ],
        [
          172.95469614953714,
          1.3438851951615003
        ],
        [
          172.95469614953714,
          1.3690476620161975
        ],
        [
          172.91173669923782,
          1.3690476620161975
        ],
        [
          172.91173669923782,
          1.3438851951615003
        ]
      ]
    ]
  },
  "properties": {
    "datetime": "2020-12-11T22:38:32.125000Z"
  },
  "collection": "simple-collection",
  "links": [
    {
      "rel": "collection",
      "href": "./collection.json",
      "type": "application/json",
      "title": "Simple Example Collection"
    },
    {
      "rel": "root",
      "href": "./collection.json",
      "type": "application/json",
      "title": "Simple Example Collection"
    },
    {
      "rel": "parent",
      "href": "./collection.json",
      "type": "application/json",
      "title": "Simple Example Collection"
    }
  ],
  "assets": {
    "visual": {
      "href": "https://storage.googleapis.com/open-cogs/stac-examples/20201211_223832_CS2.tif",
      "type": "image/tiff; application=geotiff; profile=cloud-optimized",
      "title": "3-Band Visual",
      "roles": [
        "visual"
      ]
    },
    "thumbnail": {
      "href": "https://storage.googleapis.com/open-cogs/stac-examples/20201211_223832_CS2.jpg",
      "title": "Thumbnail",
      "type": "image/jpeg",
      "roles": [
        "thumbnail"
      ]
    }
  ]
}

```

รูปที่ 2.5 ตัวอย่าง STAC Item

(ที่มา: <https://github.com/radiantearth/stac-spec/blob/master/examples/simple-item.json>)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 Catalog Specification

STAC Catalog เป็น Object ลาดับบนสุดที่รวม Object Catalog, Collection และ Item อื่น ๆ เปรียบเสมือนโฟลเดอร์ที่ใช้จัดกลุ่ม หรือลำดับชั้นของข้อมูล เพื่อให้การเรียกดูนั้นเข้าใจง่ายมากขึ้น โดยใน Catalog ประกอบด้วยอะเรย์ของ Link Object ที่จะเชื่อมโยงไปยัง Object อื่น ๆ และยังสามารถเพิ่ม คอธิบายข้อมูลเพิ่มเติมเข้าไปเพื่อใช้ในการอธิบาย Object ที่อยู่ภายใต้ตนเอง ซึ่ง Object Catalog นี้จะเป็น จุดเริ่มต้นในการเข้าถึงหรือเรียกดู STAC

โดย Item สามารถมีความสัมพันธ์เชื่อมโยงกับ Catalog ได้โดยตรง หรือ Catalog ก็อาจมีความสัมพันธ์เชื่อมโยงกับ Catalog อื่น ๆ ได้ โดยในกรณีนี้จะเรียกว่า Sub-catalog ซึ่ง Catalog ที่เชื่อมโยงไปถึงนั้น ก็จะมี Links ที่นำไปสู่ Collections และ Items โดยในการพัฒนาเป็น Sub-catalog นั้นขึ้นอยู่กับตัวนักพัฒนาเอง แต่ก็สามารถนำแนวคิดนี้มาใช้ในการช่วย ทำให้การเรียกดูข้อมูลของผู้ใช้งานนั้นง่ายขึ้นได้

Field เบื้องต้นที่เป็นส่วนประกอบของ STAC Catalog มีดังนี้

ตารางที่ 2.9 แสดง Field เบื้องต้นที่เป็นส่วนประกอบของ STAC Catalog

ชื่อ Field	ประเภท ข้อมูล	จำเป็น	ค อธิบาย
type	string	✓	มีค่าเป็น 'Catalog' ถ้า Catalog นั้นปฏิบัติตามข้อกำหนดของ Catalog
stac_version	string	✓	เวอร์ชัน STAC ที่ Catalog ใช้
stac_extensions	[string]		ส่วนขยายที่ Catalog ใช้
id	string	✓	ใช้สำหรับระบุตัว Catalog ควรเป็นค่าที่ไม่ซ้ำ
title	string		เป็นชื่อเรียกที่สั้น มีความยาวไม่เกิน 1 บรรทัด
description	string	✓	รายละเอียดที่ใช้อธิบาย Catalog เพิ่มเติม สามารถใช้ Syntax CommonMark 0.29 ในการอธิบายแบบ Rich Text
links	[Link Object]	✓	รายการของเอกสารที่อ้างอิงไปถึงได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Link Object

เป็น Object ที่อธิบายถึงความสัมพันธ์กับ Entity อื่น ซึ่งแนะนำให้ผู้ให้บริการข้อมูลพยายามระบุ links ที่เชื่อมโยงกันให้ได้มากที่สุดเพื่อให้ผู้ใช้งานสามารถเข้าใจว่ามี Entity ใด ที่สามารถเชื่อมโยงไปถึงได้อีกบ้าง โดยรายละเอียดที่ต้องระบุใน Link Object แต่ละตัวมีดังตารางด้านล่าง

ตารางที่ 2.10 แสดง Field ที่เป็นส่วนประกอบของ Link Object

ชื่อ Field	ประเภทข้อมูล	จำเป็น	คำอธิบาย
href	string	✓	Link ในรูปแบบ URL สามารถเป็นได้ทั้ง Relative และ Absolute Link
rel	string	✓	ความสัมพันธ์ระหว่างเอกสารที่เรียกดูอยู่กับเอกสารที่ถูกเชื่อมโยงไปถึง
type	string		Media Type ของ Entity ที่อ้างอิงถึง
title	string		ชื่อที่สามารถอ่านเข้าใจได้ส สำหรับใช้แสดงกับ Link

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประเภทของความสัมพันธ์ระหว่าง Catalog กับเอกสารอื่น

ใน Field `rel` ของ Link Object ที่ใช้ในการระบุความสัมพันธ์ระหว่างเอกสารที่เรียกดูอยู่กับเอกสารที่ถูกเชื่อมโยงไปถึง สามารถมีความสัมพันธ์ที่เป็นไปได้ดังนี้

ตารางที่ 2.11 แสดงประเภทความสัมพันธ์ที่เป็นไปได้ระหว่าง Item กับ เอกสารที่เชื่อมโยงไปถึง

ประเภท	คำอธิบาย
self	(*แนะนำ ให้มีอย่างมาก) Absolute URL ที่เชื่อมโยงไปยัง Catalog ถ้าสามารถเข้าถึงได้ด้วย Public URL
root	(*แนะนำ ให้มีอย่างมาก) URL ที่เชื่อมโยงไปยัง Root STAC Catalog หรือ Collection สำหรับ Catalog ก็ควรมีความสัมพันธ์ประเภทนี้ ถึงแม้ว่าจะเชื่อมโยงกลับมาที่ตนเอง
parent	URL ที่เชื่อมโยงไปยัง STAC Entity ล ดับก่อนหน้า สำหรับ Catalog ที่ไม่ใช่ Root ควรมีความสัมพันธ์ประเภทนี้เชื่อมโยงไปถึงล ดับก่อนหน้า
child	URL ที่เชื่อมโยงไปยัง STAC Entity ล ดับถัดไป
item	URL ที่เชื่อมโยงไปยัง STAC Item

Media Types

เป็นการแจ้งรายละเอียดเกี่ยวกับเนื้อหาภายในเอกสารที่อ้างอิง หรือเชื่อมโยงไปถึงว่ามี Common Media Type เป็นอย่างไร ซึ่งจะถูกใช้ระบุใน Field `'type'` ที่ถูกใช้ใน Link และ Asset Object ซึ่งรายละเอียดของ STAC Media Types มีดังตารางด้านล่างนี้

ตารางที่ 2.12 แสดง Media Types ที่ใช้สำหรับองค์ประกอบของ STAC

Media Type	คำอธิบาย
application/geo+json	STAC Item
application/json	STAC Catalog
application/json	STAC Collection

ตัวอย่าง JSON สำหรับข้อมูล Catalog

```

{
  "id": "examples",
  "type": "Catalog",
  "title": "Example Catalog",
  "stac_version": "1.0.0",
  "description": "This catalog is a simple demonstration of an example
catalog that is used to organize a hierarchy of collections and their
items.",
  "links": [
    {
      "rel": "root",
      "href": "./catalog.json",
      "type": "application/json"
    },
    {
      "rel": "child",
      "href": "./extensions-collection/collection.json",
      "type": "application/json",
      "title": "Collection Demonstrating STAC Extensions"
    },
    {
      "rel": "child",
      "href": "./collection-only/collection.json",
      "type": "application/json",
      "title": "Collection with no items (standalone)"
    },
    {
      "rel": "child",
      "href": "./collection-only/collection-with-schemas.json",
      "type": "application/json",
      "title": "Collection with no items (standalone with JSON Schemas)"
    },
    {
      "rel": "item",
      "href": "./collectionless-item.json",
      "type": "application/json",
      "title": "Collection with no items (standalone)"
    },
    {
      "rel": "self",
      "href": "https://raw.githubusercontent.com/radiantearth/stac-
spec/v1.0.0/examples/catalog.json",
      "type": "application/json"
    }
  ]
}

```

รูปที่ 2.6 ตัวอย่าง STAC Catalog

(ที่มา: <https://github.com/radiantearth/stac-spec/blob/master/examples/catalog.json>)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3 Collection Specification

STAC Collection เป็น Object ที่ใช้อธิบายกลุ่มของ Item ที่เกี่ยวข้อง ซึ่งมีโครงสร้าง Field คล้าย Catalog แต่มีการเพิ่ม Field สำหรับใช้ในการอธิบายกลุ่มของ Item เพิ่มเติม โดย Collection สามารถมี Parent เป็น Catalog หรือ Collection ก็ได้ และสามารถมี Child เป็น Item, Catalog หรือ Collection ก็ได้

STAC Collection มีองค์ประกอบที่ยืดหยุ่นได้ในการเพิ่มคำอธิบายข้อมูลเพิ่มเติมสำหรับกลุ่มของ Item โดยทั่วไปแล้วจะใช้สำหรับอธิบายกลุ่มของสินทรัพย์ที่ถูกกำหนดด้วยคุณสมบัติเดียวกัน และใช้คำอธิบายข้อมูลภาพรวมด้วยกัน

STAC Collection ถูกแสดงในรูปแบบ JSON โดย JSON ใด ๆ ที่มี Field ที่บังคับเป็นส่วนประกอบครบ ก็จะเป็น STAC Collection ที่ถูกต้อง อีกทั้งยังเป็น STAC Catalog ที่ถูกต้องด้วย ซึ่ง STAC Collection ยังสอดคล้องกับ Collection JSON ที่ถูกระบุไว้ใน OGC API - Features แต่ขยายเพิ่มเติมออกมาด้วยการเพิ่ม Field เข้าไป

Field เบื้องต้นที่เป็นส่วนประกอบของ STAC Collection มีดังนี้

ตารางที่ 2.13 แสดง Field เบื้องต้นที่เป็นส่วนประกอบของ STAC Collection

ชื่อ Field	ประเภทข้อมูล	จำเป็น	คำอธิบาย
type	string	✓	มีค่าเป็น 'Collection'
stac_version	string	✓	เวอร์ชัน STAC ที่ Collection ใช้
stac_extensions	[string]		ส่วนขยายที่ Collection ใช้
id	string	✓	ใช้สำหรับระบุตัว Collection ควรเป็นค่าที่ไม่ซ้ำกันในผู้ให้บริการเดียวกัน
title	string		เป็นชื่อเรียกที่สั้น มีความยาวไม่เกิน 1 บรรทัด
description	string	✓	รายละเอียดที่ใช้อธิบาย Collection เพิ่มเติม สามารถใช้ syntax CommonMark 0.29 ในการอธิบายแบบ rich text
keywords	[string]		รายการของ Keyword ที่ใช้บอกรายละเอียดของ Collection

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

license	string	✓	การอนุมัติของ Collection สามารถมีค่าได้ทั้ง SPDX License identifier, 'various' ถ้ามีการใช้หลายการอนุมัติ หรือ 'proprietary' สำหรับกรณีอื่น
providers	[Provider Object]		รายการของผู้ให้บริการ ซึ่งสามารถระบุองค์กรที่ เป็นผู้เก็บภาพ และประมวลผลข้อมูล หรือเป็นผู้เก็บและให้บริการข้อมูล ควรเรียงล าดับโดยให้ ผู้ให้บริการล่าสุดเป็นล าดับสุดท้ายในรายการ
extent	Extent Object	✓	ขอบเขตเชิงพื้นที่ และเวลา
summaries	Map<string, [*]Range Object JSON Schema Object>		(*แนะนำ ให้มีอย่างมาก) รายการสรุปของคุณสมบัติต่าง ๆ สามารถมีค่าได้ทั้งในรูปแบบกลุ่มของค่าที่เจาะจง ช่วงหรือสถิติค่ามากที่สุด ต ่าสุด หรือ JSON Schema
links	[Link Object]	✓	รายการของเอกสารที่อ้างไปถึงได้
assets	Map<string, Asset Object>		รายการของ Asset Object ที่สามารถดาวน์โหลดได้ โดยชื่อ Key ห้ามซ้ำกัน

Provider Object

เป็นข้อมูลเกี่ยวกับผู้ให้บริการ โดยสามารถระบุองค์กรที่เป็นคนเก็บภาพ และประมวลผลข้อมูล หรือเป็นผู้เก็บและให้บริการข้อมูล

ตารางที่ 2.14 แสดง Field ที่เป็นส่วนประกอบของ Provider Object

ชื่อ Field	ประเภท ข้อมูล	จ า เป็น	ค า ธิบาย
name	string	✓	ชื่อองค์กร หรือบุคคล
description	string		รายละเอียดที่ใช้อธิบายผู้ให้บริการเพิ่มเติม สามารถใช้ syntax เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกำหนดการเขียนที่รองรับการเขียนโปรแกรมในวงใช้ประโยชน์ด้านการค้า

roles	[string]		หน้าที่ของผู้ให้บริการ ค่าที่เป็นไปได้ ได้แก่ 'licensor', 'producer', 'processor' หรือ 'host'
url	string		URL ที่น ไปสู่หน้า Homepage ของผู้ให้บริการ

Extent Object

เป็น Object ที่ใช้อธิบายถึงขอบเขตของ Collection ในเชิงของพื้นที่ และเวลา โดยมีรายละเอียดที่ต้องระบุไว้ดังตารางด้านล่าง

ตารางที่ 2.15 แสดง Field ที่เป็นส่วนประกอบของ Extent Object

ชื่อ Field	ประเภทข้อมูล	จำเป็น	คำอธิบาย
spatial	Spatial Extent Object	✓	ขอบเขตเชิงพื้นที่ที่เป็นไปได้ที่ครอบคลุม Collection
temporal	Temporal Extent Object	✓	ขอบเขตเชิงเวลาที่เป็นไปได้ที่ครอบคลุม Collection

Spatial Extent Object

เป็น Object ที่ใช้อธิบายขอบเขตเชิงพื้นที่ของ Collection

ตารางที่ 2.16 แสดง Field ที่เป็นส่วนประกอบของ Spatial Extent Object

ชื่อ Field	ประเภทข้อมูล	จำเป็น	คำอธิบาย
bbox	[[number]]	✓	ขอบเขตเชิงพื้นที่ที่เป็นไปได้ที่ครอบคลุม Collection โดยในอระเรย์ชั้นนอกสามารถแยกเป็นอีกขอบเขตเชิงพื้นที่ที่ใช้อธิบาย Bounding Boxes ของสินทรัพย์ของ Collection นี้โดยใช้ 2D หรือ 3D geometries โดยอ้างอิงตามระบบพิกัด WGS 84

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Temporal Extent Object

เป็น Object ที่ใช้อธิบายขอบเขตเชิงเวลาของ Collection

ตารางที่ 2.17 แสดง Field ที่เป็นส่วนประกอบของ Temporal Extent Object

ชื่อ Field	ประเภทข้อมูล	จำเป็น	คำอธิบาย
interval	[[string null]]	✓	ขอบเขตเชิงเวลาที่เป็นไปได้ที่ครอบคลุม Collection โดยในอะเรย์ ชั้นนอกสามารถแยกเป็นอีกขอบเขตเชิงเวลา โดยช่วงเวลาลาดับแรก จะใช้ในการอธิบายภาพรวมขอบเขตเวลาของข้อมูล ซึ่งรูปแบบของวัน และเวลาต้องอ้างอิงตาม RFC 3339, หมวด 5.6 และเป็นแบบ UTC และสำหรับเวลาที่เป็นช่วงเปิด สามารถกำหนดค่าให้ วันเวลาเริ่มต้น หรือ วันเวลาสิ้นสุด ใดๆอย่างหนึ่งเป็น null แต่ทั้ง 2 ค่าไม่ควร เป็น null พร้อมกัน

Range Object

สำหรับใน Field 'summaries' ที่ค่าต่อเนื่อง สามารถใช้ข้อมูลทางสถิติในการแสดงแทนได้ โดยระบุค่าตาม Field ที่จำเป็นและสามารถเพิ่มค่าทางสถิติอื่น ๆ เข้าไปอีกได้ ซึ่งรายละเอียด Field เบื้องต้นมีดังตารางด้านล่าง

ตารางที่ 2.18 แสดง Field เบื้องต้นที่เป็นส่วนประกอบของ Range Object

ชื่อ Field	ประเภทข้อมูล	จำเป็น	คำอธิบาย
minimum	number string	✓	ค่าต่ำสุด
maximum	number string	✓	ค่าสูงสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Link Object

เป็น Object ที่อธิบายถึงความสัมพันธ์กับ Entity อื่น ซึ่งแนะนำให้ผู้ให้บริการข้อมูลพยายามระบุ links ที่เชื่อมโยงกันให้ได้มากที่สุดเพื่อให้ผู้ใช้งานสามารถเข้าใจว่ามี Entity ใด ที่สามารถเชื่อมโยงไปถึงได้อีกบ้าง โดยรายละเอียดที่ต้องระบุใน Link Object แต่ละตัวมีดังตารางด้านล่าง

ตารางที่ 2.19 แสดง Field ที่เป็นส่วนประกอบของ Link Object

ชื่อ Field	ประเภทข้อมูล	จำเป็น	คำอธิบาย
href	string	✓	Link ในรูปแบบ URL สามารถเป็นได้ทั้ง Relative และ Absolute Link
rel	string	✓	ความสัมพันธ์ระหว่างเอกสารที่เรียกดูอยู่กับเอกสารที่ถูกเชื่อมโยงไปถึง
type	string		Media Type ของ Entity ที่อ้างอิงถึง
title	string		ชื่อที่สามารถอ่านเข้าใจได้ส สำหรับใช้แสดงกับ Link

ประเภทของความสัมพันธ์ระหว่าง Collection กับเอกสารอื่น

ใน Field rel ของ Link Object ที่ใช้ในการระบุความสัมพันธ์ระหว่างเอกสารที่เรียกดูอยู่กับเอกสารที่ถูกเชื่อมโยงไปถึง สามารถมีความสัมพันธ์ที่เป็นไปได้ดังนี้

ตารางที่ 2.20 แสดงประเภทความสัมพันธ์ที่เป็นไปได้ระหว่าง Item กับ เอกสารที่เชื่อมโยงไปถึง

ประเภท	คำอธิบาย
self	(*แนะนำ ให้มีอย่างมาก) Absolute URL ที่เชื่อมโยงไปยัง Item ถ้าสามารถเข้าถึงได้ด้วย Public URL
root	URL ที่เชื่อมโยงไปยัง Root STAC Entity ส าหรับ Collection ก็ควรมีความสัมพันธ์ประเภทนี้ ถึงแม้ว่าจะอาจจะเชื่อมโยงกลับมาที่ตนเอง
parent	URL ที่เชื่อมโยงไปยัง STAC Entity ล าดับก่อนหน้า ส าหรับ Collection ที่ไม่ใช่ Root ควรมีความสัมพันธ์ประเภทนี้เชื่อมโยงไปถึง ล าดับก่อนหน้า

child	URL ที่เชื่อมโยงไปยัง STAC Entity ล ดับถัดไป
item	URL ที่เชื่อมโยงไปยัง Item โดยทุก Item ที่ถูกอ้างอิงจาก Collection ต้องมี Links ที่เชื่อมโยงกลับมาหา Collection ด้วยความสัมพันธ์ประเภท 'collection'
license	URLs ของการอนุญาตส สำหรับ Collection ควรจะระบุไว้ด้วย ถ้าหากใน Field license ถูกระบุไว้ว่า 'proprietary' หรือ 'various'
derived_from	URL ที่เชื่อมโยงไปยัง STAC Collection ที่ถูกใช้เป็นข้อมูลเข้าเมื่อสร้าง Collection

Asset Object

เป็น object ที่ประกอบด้วย URI ที่นำพาไปหาข้อมูลที่เกี่ยวข้องกับ Collection ที่สามารถดาวน์โหลดได้ ซึ่งสามารถมี Field เพิ่มเติมนอกเหนือจาก Field เบื้องต้นได้ โดย Field เบื้องต้นของ Collection มีดังตารางด้านล่างนี้

ตารางที่ 2.21 แสดง Field เบื้องต้นที่เป็นส่วนประกอบของ Asset Object

ชื่อ Field	ประเภทข้อมูล	จำเป็น	คำอธิบาย
href	string	✓	URI ที่นำสู่สินทรัพย์ สามารถเป็นได้ทั้ง Relative และ Absolute URI
title	string		ชื่อเรียกส สำหรับแสดงให้ผู้ใช้งาน
description	string		รายละเอียดเกี่ยวกับสินทรัพย์เพิ่มเติม
type	string		Media Type ของสินทรัพย์
roles	[string]		หน้าที่ หรือคุณสมบัติของสินทรัพย์นั้น คล้ายกับ rel ของ links

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง JSON สำหรับข้อมูล Collection

```

{
  "id": "simple-collection",
  "type": "Collection",
  "stac_extensions": [
    "https://stac-extensions.github.io/eo/v1.0.0/schema.json",
    "https://stac-extensions.github.io/projection/v1.0.0/schema.json",
    "https://stac-extensions.github.io/view/v1.0.0/schema.json"
  ],
  "stac_version": "1.0.0",
  "description": "A simple collection demonstrating core catalog fields with links to a couple of items",
  "title": "Simple Example Collection",
  "providers": [
    {
      "name": "Remote Data, Inc",
      "description": "Producers of awesome spatiotemporal assets",
      "roles": [
        "producer",
        "processor"
      ],
      "url": "http://remotedata.io"
    }
  ],
  "extent": {
    "spatial": {
      "bbox": [
        [
          172.91173669923787,
          1.3438851951615003,
          172.95469614953714,
          1.3690476520161975
        ]
      ]
    },
    "temporal": {
      "interval": [
        [
          "2020-12-11T22:38:32.125Z",
          "2020-12-14T18:02:31.437Z"
        ]
      ]
    }
  },
  "license": "CC-BY-4.0",
  "summaries": {
    "platform": [
      "cool_sat1",
      "cool_sat2"
    ],
    "constellation": [
      "ion"
    ],
    "instruments": [
      "cool_sensor_v1",
      "cool_sensor_v2"
    ],
    "gsd": {
      "minimum": 0.512,
      "maximum": 0.66
    },
    "eo:cloud_cover": {
      "minimum": 1.2,
      "maximum": 1.2
    },
    "proj:epsg": {
      "minimum": 32659,
      "maximum": 32659
    },
    "view:sun_elevation": {
      "minimum": 54.9,
      "maximum": 54.9
    },
    "view:off_nadir": {
      "minimum": 3.8,
      "maximum": 3.8
    },
    "view:sun_azimuth": {
      "minimum": 135.7,
      "maximum": 135.7
    }
  },
  "links": [
    {
      "rel": "root",
      "href": "./collection.json",
      "type": "application/json",
      "title": "Simple Example Collection"
    },
    {
      "rel": "item",
      "href": "./simple-item.json",
      "type": "application/geo+json",
      "title": "Simple Item"
    },
    {
      "rel": "item",
      "href": "./core-item.json",
      "type": "application/geo+json",
      "title": "Core Item"
    },
    {
      "rel": "item",
      "href": "./extended-item.json",
      "type": "application/geo+json",
      "title": "Extended Item"
    },
    {
      "rel": "self",
      "href": "https://raw.githubusercontent.com/radiantearth/stac-spec/v1.0.0/examples/collection.json",
      "type": "application/json"
    }
  ]
}

```

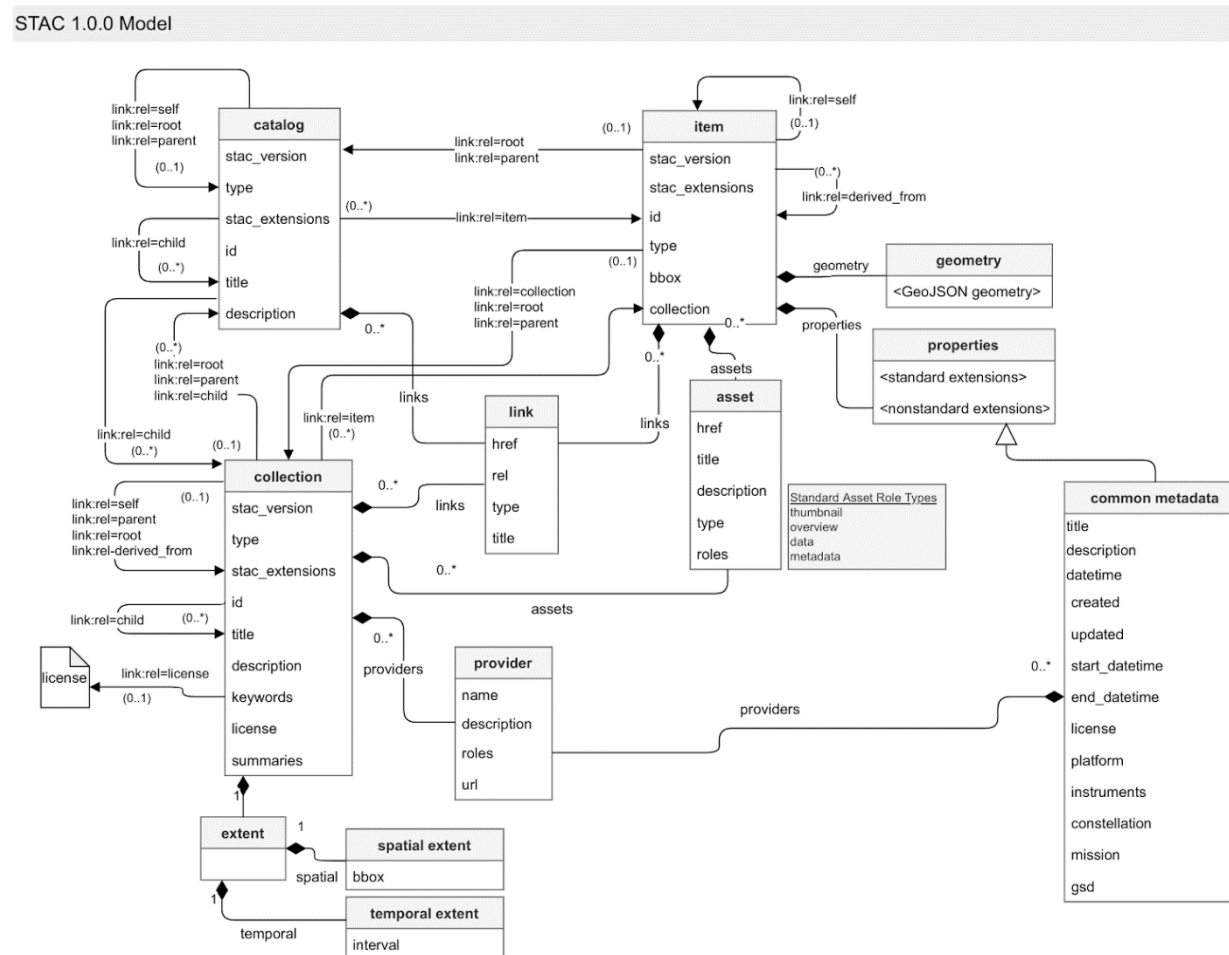
รูปที่ 2.7 ตัวอย่าง STAC Collection

(ที่มา: <https://github.com/radiantearth/stac-spec/blob/master/examples/collection.json>)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.4 STAC Model UML

ส่วนประกอบและความสัมพันธ์ที่เป็นไปได้ของ STAC Object แต่ละประเภทสามารถแสดงได้ด้วยแผนภาพ UML นี้



รูปที่ 2.8 UML ของ STAC Model

(ที่มา: <https://github.com/radianteearth/stac-spec/blob/master/STAC-UML.pdf>)

2.5.5 Dynamic และ Static Catalogs

การจัดท าคatalog ของข้อมูลที่ใช้อ้างอิงตาม STAC Specifications นี้ สามารถมีได้ 2 รูปแบบ ได้แก่ Static Catalog และ Dynamic Catalog โดย Static Catalog นั้นจะเป็นการจัดท าคatalog ในรูปแบบกลุ่มของ flat file ที่ถูกเก็บอยู่บน Web Server หรือ Object Store เช่น Amazon S3 หรือ Google Cloud Storage ส่วน Dynamic Catalog นั้นจะเป็น Catalog ที่จัดท าคatalog อย่าง Dynamic โดยส่วนใหญ่จะมีระบบด้านหลังเพื่อที่จะใช้ในการเตรียม Response ออกมาตามที่ถูกร้องขอเข้าไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 SpatioTemporal Asset Catalog API Specification

SpatioTemporal Asset Catalog API เป็นรูปแบบ Dynamic ของ SpatioTemporal Asset Catalog ที่กำหนด JSON-based web API สำหรับเรียกดู และค้นหา SpatioTemporal Asset Catalog (STAC) Object ซึ่งในแกนหลักของ STAC Specification ได้กล่าวถึงแต่โครงสร้าง และภาษาที่ใช้อธิบายสินทรัพย์ แต่ในความเป็นจริงแล้วผู้ใช้งานต้องการเข้าถึงกลุ่มของข้อมูลภายใน Catalog โดยการค้นหาแบบมีช่วงเวลาที่เฉพาะเจาะจง ต้องการระบุบริเวณพื้นที่ที่สนใจ หรือค้นหาตามคุณสมบัติที่ตรงตามเงื่อนไขที่ต้องการ ใน STAC API Specification จึงได้พัฒนาข้อกำหนดในส่วนของการพารามิเตอร์ที่ใช้ในการสืบค้นเหล่านั้น และระบบที่สอดคล้องกับมาตรฐานที่จะตอบ STAC Item objects ที่ตรงกับที่ผู้ใช้งานต้องการกลับไป ซึ่งฟังก์ชันในการท างานสามารถเพิ่มเติมได้โดยอ้างอิงมาตรฐานกลุ่ม OGC API โดยเฉพาะมาตรฐานในหมวด OGC API - Features

โดยรายละเอียดของข้อกำหนด STAC API Specification แต่ละหมวดมีดังนี้

2.6.1 Core

ในแกนหลักของ STAC API Specification นั้นจะระบุถึงการตอบ STAC Catalog ที่ตรงตามข้อกำหนด กลับมา พร้อมกับรายละเอียดว่าสอดคล้องกับข้อกำหนดส่วนใดบ้าง ซึ่งในส่วนของ links ของ Catalog นั้นจะเป็นส่วนที่เป็นจุดเชื่อมโยงไปยังข้อมูลส่วนอื่น ๆ โดยดูจากความสัมพันธ์ที่ระบุไว้

โดย Endpoints ที่ต้องพัฒนาในส่วนของ Core มีดังนี้

ตารางที่ 2.22 แสดง Endpoints ที่ข้อกำหนด Core กำหนดให้มี

Endpoint	สิ่งที่ตอบกลับ	คำอธิบาย
/	Catalog	Landing page ที่เชื่อมโยงกับ API Capabilities
/api	ไม่ได้จำกัด	คำอธิบาย Service ที่เปิดให้บริการ จะถูกเรียกจาก ความสัมพันธ์ประเภท 'service-desc' ของ links
/catalogs/{catalogId}	Catalog	Child Catalog Object (*แนะนำ ให้ใช้ Endpoint ในรูปแบบนี้ถ้ามีโครงสร้างของ Sub-catalog อยู่ใน STAC)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง JSON สำหรับ Landing Page ของ STAC API - Core

```

{
  "stac_version": "1.0.0",
  "id": "example-stac",
  "title": "A simple STAC API Example",
  "description": "This Catalog aims to demonstrate the a simple landing page",
  "type": "Catalog",
  "conformsTo" : [
    "https://api.stacspec.org/v1.0.0-rc.1/core"
  ],
  "links": [
    {
      "rel": "self",
      "type": "application/json",
      "href": "https://stac-api.example.com"
    },
    {
      "rel": "root",
      "type": "application/json",
      "href": "https://stac-api.example.com"
    },
    {
      "rel": "service-desc",
      "type": "application/vnd.oai.openapi+json;version=3.0",
      "href": "https://stac-api.example.com/api"
    },
    {
      "rel": "service-doc",
      "type": "text/html",
      "href": "https://stac-api.example.com/api.html"
    },
    {
      "rel": "child",
      "type": "application/json",
      "href": "https://stac-api.example.com/catalogs/sentinel-2",
    },
    {
      "rel": "child",
      "type": "application/json",
      "href": "https://stac-api.example.com/catalogs/landsat-8",
    }
  ]
}

```

รูปที่ 2.9 ตัวอย่าง Landing Page สำหรับ STAC API – Core

(ที่มา: <https://github.com/radiantearth/stac-api-spec/blob/main/core/README.md>)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจัดทำโครงสร้างลำดับชั้นของ Catalog

สำหรับนักพัฒนาที่ต้องการนำ STAC API ไปใช้ในการค้นหาเป็นหลัก ควรพิจารณาการพัฒนามุมมองการเรียกดูข้อมูลในรูปแบบที่เป็นกราฟระบุทิศทาง (Directed Graph) ของแค็ตตาล็อก โดยที่ความสัมพันธ์แบบเป็นแม่ลูกกัน (child) ทำให้เกิดเป็นต้นไม้ และ Catalog ก็สามารถถูกเรียกดูได้ด้วยการร้องขอเดียว เนื่องจากแต่ละ Catalog นั้นมีขนาดเล็กจนไม่จำเป็นต้องมีการจัดลำดับชั้น

ในขณะที่ OGC API – Features ระบุไว้ว่า Item ต้องเป็นส่วนหนึ่งของ Collection แต่ก็ไม่ได้หมายความว่า Collection จำเป็นต้องเป็นส่วนหนึ่งของต้นไม้ แต่ถ้าหากว่ามี Collection เป็นส่วนหนึ่งของต้นไม้ ข้อกำหนดที่แนะนำว่าควรมีแค่ 1 Collection ในเส้นทางนั้น อีกทั้ง Collection ต้องไม่มีลูกเป็น Collection

โดยโครงสร้างที่เป็นไปได้ในการจัดโครงสร้างต้นไม้ของแค็ตตาล็อกเป็นได้ 2 รูปแบบหลัก ๆ ซึ่งใน 2 รูปแบบนี้จะต่างกันแค่มีหรือไม่มี Collection เป็นส่วนประกอบของต้นไม้

- Catalog (root) → Catalog* → Item (recommended)
- Catalog (root) → Collection → Catalog* → Item

การจัดโครงสร้างสามารถทำให้นักพัฒนารวมกลุ่มของ Collection เข้าด้วยกัน แล้วสร้างเป็นกลุ่มย่อยของ Item ที่อยู่ภายใน Collection เช่น ถ้าหากมี Collection ของแต่ละภาพถ่ายดาวเทียมแต่ละแหล่งดังนี้ Landsat 8 Collection 1, Landsat 8 Surface Reflectance, Sentinel-2 L1C, Sentinel-2 L2A, Sentinel-5P UV Aerosol Index, Sentinel-5P Cloud, MODIS MCD43A4, MODIS MOD11A1, และ MODIS MYD11A1 ก็สามารถนำเสนอข้อมูลเหล่านี้ในรูปแบบของ Catalog ที่รวมกลุ่มเข้าด้วยกันเป็นแค็ตตาล็อกของ Landsat, Sentinel, และ MODIS ซึ่งนำเสนอเป็นแผนผังได้ดังนี้

Dimension: Root → Product → Platform → Instrument

Example:

- / root catalog
 - child -> /catalogs/landsat
 - child -> /catalogs/landsat_7
 - child -> /catalogs/landsat_8
 - child -> /catalogs/landsat_8_c1
 - child -> /catalogs/landsat_8_sr
 - child -> /catalogs/sentinel
 - child -> /catalogs/sentinel_2
 - child -> /catalogs/sentinel_2_l1c
 - child -> /catalogs/sentinel_2_l2a

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเฉพาะเท่านั้น ไม่ควรเผยแพร่ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิคัดลอกเอกสารทุกครั้งที่มีการนำไปใช้

- child -> /catalogs/sentinel_5p
 - child -> /catalogs/sentinel_5p_uvai
 - child -> /catalogs/sentinel_5p_cloud
- child -> /catalogs/modis
 - child -> /catalogs/modis_mcd43a4
 - child -> /catalogs/modis_mod11a1
 - child -> /catalogs/modis_myd11a1

ในอีกรูปแบบหนึ่งของการจัดโครงสร้างก็สามารถแบ่ง Collection ให้เป็นกลุ่มเล็ก ๆ เช่น ในโครงสร้างของข้อมูล Landsat 8 Collection 1 สามารถจัดกลุ่มได้เป็น Path, Row, และ Date ที่เป็นการแบ่งกริดของข้อมูล ซึ่งนำเสนอเป็นแผนผังได้ดังนี้

Dimension: Root → Instrument → Path → Row → Date

Example:

- / (root)
 - /catalogs/landsat_8_c1
 - /catalogs/landsat_8_c1/139
 - /catalogs/landsat_8_c1/139_045
 - /catalogs/landsat_8_c1/139_045_20170304
 - /collections/landsat_8_c1/items/LC08_L1TP_139_045_20170304_20170316_01_T1
 - /catalogs/landsat_8_c1/139_045_20170305
 - /collections/landsat_8_c1/items/LC08_L1TP_139_045_20170305_20170317_01_T1
 - /catalogs/landsat_8_c1/139_046
 - /catalogs/landsat_8_c1/139_046_20170304
 - /collections/landsat_8_c1/items/LC08_L1TP_139_046_20170304_20170316_01_T1
 - /catalogs/landsat_8_c1/139_046_20170305
 - /collections/landsat_8_c1/items/LC08_L1TP_139_046_20170305_20170317_01_T1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยสามารถจัดโครงสร้างของแค็ตตาล็อกในรูปแบบที่หลากหลายเพื่อที่จะนำพาไปสู่ Item เช่นส าหรับ ข้อมูลของ Landsat 8 อาจสามารถให้ผู้ใช้งานเข้าถึงได้จาก Date แล้ว Path แล้ว Row หรือ Path แล้ว Row แล้ว Date ดังแผนผังด้านล่างนี้

1. Catalog → Catalog (product) → Catalog (date) → Catalog (path) → Catalog (row)
2. Catalog → Catalog (product) → Catalog (path) → Catalog (row) → Catalog (date)

การจัดโครงสร้างของแค็ตตาล็อกก็สามารถมีได้หลายรูปแบบเพื่อให้รองรับกับข้อมูลที่ต้องการ น าเสนอ ดังนั้นจึงต้องมีการวิเคราะห์เพื่อหาโครงสร้างที่เหมาะสมในการจัดโครงสร้างในการเข้าถึง เพื่อให้ตอบกับความ ต้องการของผู้ใช้งาน

2.6.2 Item Search

ฟังก์ชัน Item Search เป็นฟังก์ชันที่ถูกใช้บ่อยสุด โดยจะถูกน าไปใช้ใน Endpoint /search ซึ่งในฟังก์ชัน นี้ได้น าพารามิเตอร์ที่ใช้ในการค้นหาที่ถูกระบุใน OGC API - Features - Part 1: Core มาใช้และเพิ่ม พารามิเตอร์ไปอีกเล็กน้อย โดยฟังก์ชัน Item Search นี้เป็นโครงสร้างแบบง่ายที่สามารถเรียกใช้ในการค้นหา STAC Item Objects ซึ่งรูปแบบของผลลัพธ์ที่ตอบกลับมามาจะต้องอยู่ในรูปแบบ GeoJSON FeatureCollection ที่ถูกต้องที่ประกอบด้วย STAC Item ที่ตรงกับการค้นหา โดย Endpoints ที่ต้องพัฒนาในส่วนของ Item Search มีดังนี้

ตารางที่ 2.23 แสดง Endpoints ที่ข้อมู ก าหนดItem Search ก าหนดให้มี

Endpoint	สิ่งที่ตอบกลับ	ค าอธิบาย
/search	Item Collection	Search Endpoint ไว้ส าหรับค้นหา STAC Item ที่ตรงเงื่อนไขกับที่ผู้ใช้งาน ระบุ โดยผู้ใช้งานจะระบุเงื่อนไขมาใน Query Parameters หรือ Request Body

เอกสารนี้เป็นเอกสารที่สงวนไว้ส าหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับ Query Parameters ที่สามารถใช้เป็นตัวเลือกในการค้นหา Item มีดังตารางด้านล่างนี้

ตารางที่ 2.24 แสดง Query Parameters ที่เป็นตัวเลือกในการค้นหา

Parameter	ประเภทข้อมูล	Source API	คำอธิบาย
limit	integer	OGC API - Features	จำนวนมากที่สุดของผลลัพธ์ที่จะตอบกลับมาแต่ละครั้ง
bbox	[number]	OGC API - Features	ขอบเขต Bounding Box ที่ต้องการค้นหา
datetime	string	OGC API - Features	วันเวลาเจาะจง หรือเป็นช่วงเวลา (คั่นด้วยเครื่องหมาย /) โดยอยู่ในรูปแบบของ RFC 3339 หมวด 5.6 สำหรับเวลาที่เป็นช่วงเปิดให้ใช้เครื่องหมาย ..
intersects	GeoJSON Geometry	STAC	ค้นหา Item โดยทวิภาค Intersect กันกับ Geometry ที่ผู้ใช้งานระบุเข้ามา
ids	[string]	STAC	รายการของ Item id ที่ต้องการ
collections	[string]	STAC	รายการของ Collection ID ที่ Item อยู่ภายใต้ ที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.3 Collections

ฟังก์ชัน Collections เป็นอีกหนึ่งฟังก์ชันที่ช่วยอำนวยความสะดวก ซึ่งใช้ในการเรียกรายการของ STAC Collection entities ที่มีอยู่ ผ่าน Endpoint /collections และสามารถเข้าถึง STAC Collection แต่ละตัว โดยระบุ Collection ID ผ่าน Endpoint /collections/{collectionId}

โดย Endpoints ที่ต้องพัฒนาในส่วนของ Collections มีดังนี้

ตารางที่ 2.25 แสดง Endpoints ที่ขอ กำหนด Collections กำหนดใหม่

Endpoint	สิ่งที่ตอบกลับ	คำอธิบาย
/collections	JSON	รายการของ Collection ที่อยู่ภายใต้ Catalog และ links
/collections/{collectionId}	Collection	Collection JSON เดี่ยว ๆ

2.6.4 Features

ฟังก์ชัน Features นี้เป็นอีกหนึ่งฟังก์ชันที่สําคัญสำหรับ STAC API เพื่อที่จะใช้ในการให้บริการ OGC API - Features Endpoint สำหรับ STAC Collection และ Item ซึ่งรายการของ STAC Collection สามารถเข้าถึงได้โดย Endpoint /collections และมี Endpoint /collections/{collectionId}/items สำหรับรายการของ STAC Item ที่อยู่ภายใน 1 Collection ซึ่งสำหรับ STAC แล้ว นี่หมายถึงการนำ OGC API - Features Core มาใช้งาน เช่นเดียวกับ GeJSON และ OpenAPI เนื่องจาก STAC อยู่ในรูปแบบของ GeJSON อยู่เสมอ และ OpenAPI ใช้ในการระบุ STAC API

โดย Endpoints ที่ต้องพัฒนาในส่วนของ Features มีดังนี้

ตารางที่ 2.26 แสดง Endpoints ที่ขอ กำหนด Features กำหนดใหม่

Endpoint	สิ่งที่ตอบกลับ	คำอธิบาย
/conformance	JSON	มาตรฐานที่ API ปฏิบัติตาม
/collections	JSON	รายการของ Collection ที่อยู่ภายใต้ Catalog และ links
/collections/{collectionId}	Collection	Collection JSON เดี่ยว ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

/collections/{collectionId}/items	ItemCollection	GeoJSON FeatureCollection ของ Item ที่อยู่ภายใต้ Collection
/collections/{collectionId}/items/{featureId}	Item	Item เดี่ยว ๆ (GeoJSON Feature)

2.6.5 Extensions & Fragments

STAC API และ OGC API - Features อนุญาตให้มีการเพิ่มส่วนขยายสำหรับฟังก์ชันเพิ่มเติมได้ ยกตัวอย่างเช่นการเพิ่ม 'sortBy' เพื่อให้สามารถเรียงลำดับผลลัพธ์ได้ ซึ่งการท ให้ส่วนขยายสามารถใช้งานได้ด้วย Item Search และ OGC API - Features นั้นในส่วนของ STAC จะถูกระบุในส่วนของ Fragments

Field ที่เป็นส่วนประกอบของ ItemCollection

เป็น Object ที่ใช้อธิบาย STAC ItemCollection โดยมีรายละเอียดดังตารางด้านล่าง

ตารางที่ 2.27 แสดง Field เบื้องต้นที่เป็นส่วนประกอบของ Asset Object

ชื่อ Field	ประเภทข้อมูล	จำเป็น	คำอธิบาย
type	string	✓	ต้องเป็น FeatureCollection เพื่อให้สอดคล้องกับ GeoJSON
features	[STAC Item]	✓	อะเรย์ของ Item Object สามารถเป็นอะเรย์เปล่าได้
links	[Link Object]		อะเรย์ของ Links ที่เกี่ยวข้องกับ ItemCollection นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

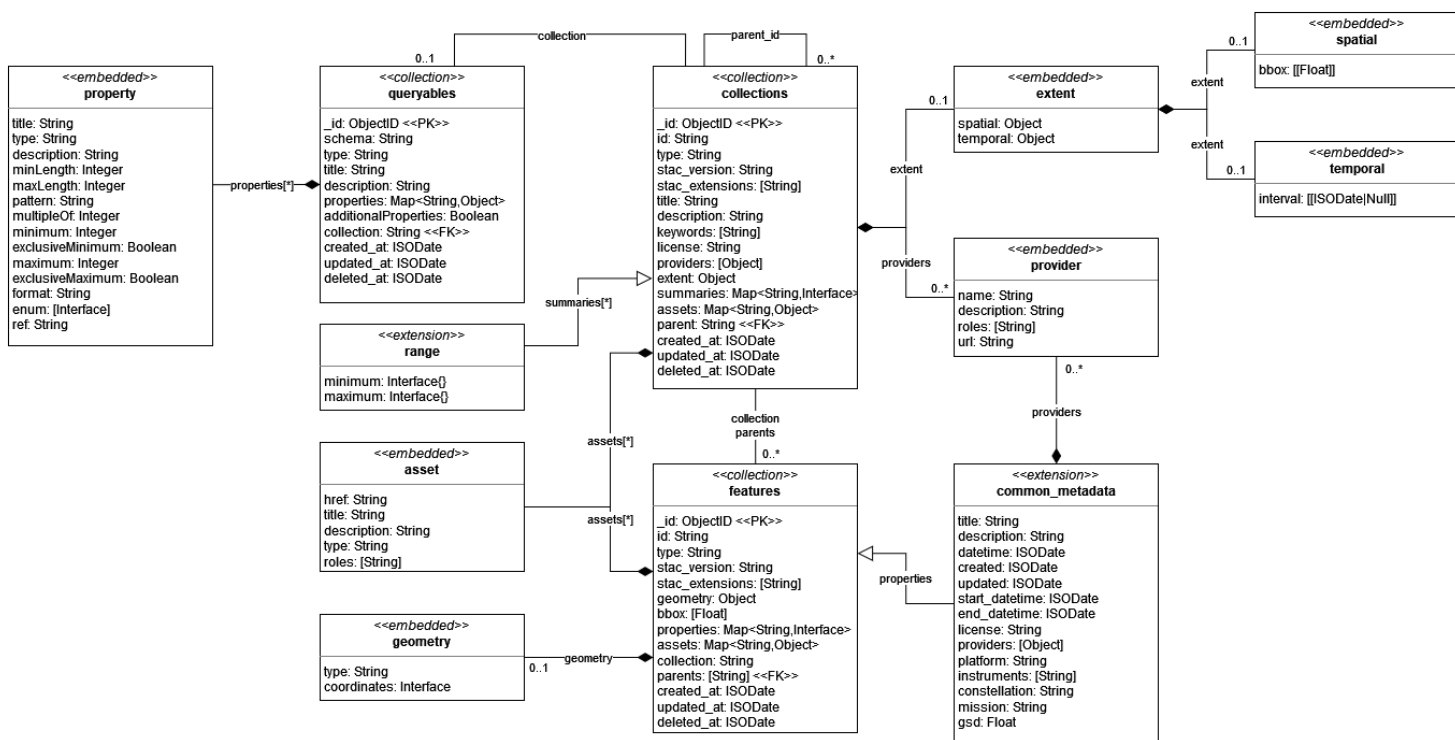
วิธีการดำเนินงานวิจัย

จากการศึกษา ทาความเข้าใจ และตีความข้อกำหนด SpatioTemporal Asset Catalog Specification เพื่อดาเนินโครงการสหกิจศึกษา การประยุกต์ใช้แค็ตตาล็อกสินทรัพย์ทางภูมิสารสนเทศ (Spatio Temporal Asset Catalog Implementation) ทาให้สามารถวางแผน และออกแบบระบบในมุมมองต่าง ๆ ออกมาได้ดังนี้

3.1 Database Diagram

จากการวิเคราะห์ STAC UML ในรูปที่ 2.8 และกรณีที่จะถูกนำไปใช้งาน สามารถออกแบบ Database Diagram ของแค็ตตาล็อกสินทรัพย์ทางภูมิสารสนเทศ 1 แค็ตตาล็อก โดยมีรายละเอียดดังนี้

Database Diagram ของ แค็ตตาล็อกสินทรัพย์ทางภูมิสารสนเทศ 1 แค็ตตาล็อก



รูปที่ 3.1 Database Diagram ของแค็ตตาล็อกสินทรัพย์ทางภูมิสารสนเทศ 1 แค็ตตาล็อก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากแผนผัง จะเห็นว่าส่วนหลัก ๆ ที่จะนำไปสร้างเป็น Collection ในฐานข้อมูลมีอยู่ 3 ส่วน คือ collections, features และ queryables โดยมีรายละเอียดดังนี้

- collections จะมีความสัมพันธ์ที่วนกลับเข้าตัวเอง เนื่องจาก collection แต่ละตัวสามารถมี parent ที่เป็นประเภท collection เหมือนกันได้ โดยในแต่ละ collection สามารถไม่มีหรือมี parent ได้แค่ 1 ตัว
- features จะมีความสัมพันธ์กับ collections 2 ความสัมพันธ์คือ collection และ parents โดยจะเป็นความสัมพันธ์จาก features ไปหา collections โดยในความสัมพันธ์ collection 1 feature จะมีความสัมพันธ์ได้กับ 1 collection และ 1 collection สามารถมีความสัมพันธ์ได้กับหลาย feature และความสัมพันธ์ parents คือ 1 feature สามารถมีความสัมพันธ์แบบ parent ได้กับ collection หลายตัว
- queryables จะมีความสัมพันธ์กับ collections โดยในแต่ละ collection จะไม่มีหรือมี queryable ได้แค่ 1 ตัว

สำหรับในส่วนอื่น ๆ ที่เป็น <<embedded>> และ <<extension>> จะเป็นส่วนที่เป็นรายละเอียดสำหรับค่าในบาง field ที่เก็บในรูปแบบ Object หรือ Map

3.2 Data Dictionary

จาก Database Diagram ในข้อที่ 3.1 สามารถเขียน Data Dictionary ของแค็ตตาล็อกสินทรัพย์ทางภูมิสารสนเทศ 1 แค็ตตาล็อก โดยมีรายละเอียดดังนี้

3.2.1 ข้อมูลกลุ่มของข้อมูล

ในฐานข้อมูลจะใช้ชื่อ Collection ว่า “collections” สำหรับเก็บรายละเอียดกลุ่มของข้อมูล และความสัมพันธ์กับ Object อื่นที่เกี่ยวข้องเพื่อใช้ในการประกอบเป็น Response เมื่อผู้ใช้งานร้องขอเข้ามาได้

ตารางที่ 3.1 แสดงรายละเอียดของโครงสร้างข้อมูลที่เก็บใน Collection “collections”

ลำดับ	ฟิลด์	ประเภทข้อมูล	คำอธิบาย
1	_id	ObjectId	(PK) รหัสข้อมูล
2	id	String	รหัสข้อมูลกลุ่มของข้อมูล
3	type	String	ประเภทของกลุ่มข้อมูล ตัวเลือกมีดังนี้ 1. Catalog 2. Collection

4	stac_version	String	เวอร์ชันของ STAC ที่กลุ่มของข้อมูลอ้างอิง
5	stac_extensions	[String]	ส่วนขยายที่กลุ่มของข้อมูลอ้างอิง
6	title	String	ชื่อเรียกกลุ่มของข้อมูล
7	description	String	รายละเอียดของกลุ่มของข้อมูล
8	keywords	[String]	รายการคาสถาฐ์ของกลุ่มของข้อมูล
9	license	String	ใบอนุญาตที่กลุ่มของข้อมูลอ้างอิง
10	providers	[Object]	<<embedded>> รายการผู้ให้บริการข้อมูลที่กลุ่มของข้อมูลเก็บ
11	extent	Object	<<embedded>> ขอบเขตของข้อมูลที่กลุ่มของข้อมูลเก็บ
12	summaries	Map<String, Interface>	<<extension>> ภาพรวมคุณสมบัติต่าง ๆ ของข้อมูลที่กลุ่มของข้อมูลเก็บ
13	assets	Map<String, Object>	<<embedded>> รายการสินทรัพย์ที่ผู้ใช้งานสามารถดาวน์โหลดได้
14	parent	String	(FK) รหัสข้อมูลของ Object ที่มีความสัมพันธ์เป็นแม่ของกลุ่มของข้อมูล
15	created_at	ISODate	วัน-เวลา ที่สร้างข้อมูล
16	updated_at	ISODate	วัน-เวลา ที่ปรับปรุงข้อมูลล่าสุด
17	deleted_at	ISODate	วัน-เวลา ที่ลบข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1.1 ข้อมูลผู้ให้บริการ

ข้อมูลเก็บในฟิลด์ที่ชื่อว่า “providers” เพื่อเก็บรายการของผู้ให้บริการข้อมูล โดยมีโครงสร้างแต่ละตัว ดังนี้

ตารางที่ 3.2 แสดงรายละเอียดของโครงสร้างข้อมูล provider

ลำดับ	ฟิลด์	ประเภทข้อมูล	คำอธิบาย
1	name	String	ชื่อองค์กร หรือบุคคลที่ให้บริการข้อมูล
2	description	String	รายละเอียดของผู้ให้บริการ
3	roles	[String]	หน้าที่ของผู้ให้บริการ
4	url	String	URL ที่น ไปสู่หน้า Homepage ของผู้ให้บริการ

3.2.1.2 ข้อมูลขอบเขต

ข้อมูลเก็บในฟิลด์ที่ชื่อว่า “extent” เพื่อเก็บรายละเอียดของขอบเขตของข้อมูล โดยมีโครงสร้างแต่ละตัว ดังนี้

ตารางที่ 3.3 แสดงรายละเอียดของโครงสร้างข้อมูล extent

ลำดับ	ฟิลด์	ประเภทข้อมูล	คำอธิบาย
1	spatial	Object	<<embedded>> ข้อมูลขอบเขตเชิงพื้นที่
2	temporal	Object	<<embedded>> ข้อมูลขอบเขตเชิงเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1.2.1 ข้อมูลขอบเขตเชิงพื้นที่

ข้อมูลเก็บในฟิลด์ที่ชื่อว่า “spatial” เพื่อเก็บรายละเอียดของขอบเขตเชิงพื้นที่ของข้อมูล โดยมีโครงสร้างแต่ละตัว ดังนี้

ตารางที่ 3.4 แสดงรายละเอียดของโครงสร้างข้อมูล spatial

ลำดับ	ฟิลด์	ประเภทข้อมูล	คำอธิบาย
1	bbox	[[Float]]	ขอบเขตเชิงพื้นที่ ในรูปแบบของ Bounding Box

3.2.1.2.2 ข้อมูลขอบเขตเชิงเวลา

ข้อมูลเก็บในฟิลด์ที่ชื่อว่า “temporal” เพื่อเก็บรายละเอียดของขอบเขตเชิงเวลาของข้อมูล โดยมีโครงสร้างแต่ละตัว ดังนี้

ตารางที่ 3.5 แสดงรายละเอียดของโครงสร้างข้อมูล temporal

ลำดับ	ฟิลด์	ประเภทข้อมูล	คำอธิบาย
1	interval	[[ISODate null]]	ขอบเขตเชิงเวลา โดยเก็บค่าเป็นช่วงเวลา โดยมีวัน-เวลา เริ่มต้น และสิ้นสุด

3.2.1.2.3 ข้อมูลสินทรัพย์

ข้อมูลเก็บในฟิลด์ที่ชื่อว่า “assets” เพื่อเก็บรายละเอียดของสินทรัพย์ที่สามารถดาวน์โหลดได้ โดยมีโครงสร้างแต่ละตัว ดังนี้

ตารางที่ 3.6 แสดงรายละเอียดของโครงสร้างข้อมูล asset

ลำดับ	ฟิลด์	ประเภทข้อมูล	คำอธิบาย
1	href	String	URI ที่นำไปสู่สินทรัพย์
2	title	String	ชื่อเรียกของสินทรัพย์
3	description	String	รายละเอียดของสินทรัพย์

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4	type	String	Media Type ของสื่อบทประพันธ์
5	roles	[String]	หน้าที่ หรือคุณสมบัติของสื่อบทประพันธ์

3.2.1.4 ข้อมูลช่วงข้อมูล

ข้อมูลสามารถเป็นโครงสร้างในการอธิบายรายละเอียดของแต่ละ key ในฟิลด์ ที่ชื่อว่า “summaries” เพื่อเก็บรายละเอียดชุดข้อมูล โดยมีโครงสร้างแต่ละตัว ดังนี้

ตารางที่ 3.7 แสดงรายละเอียดของโครงสร้างข้อมูล range

ลำดับ	ฟิลด์	ประเภทข้อมูล	คำอธิบาย
1	minimum	Interface	ค่าต่ำสุด
2	maximum	Interface	ค่าสูงสุด

3.2.2 ข้อมูลสิ่งที่น่าสนใจปรากฏการณ์บนโลก

ในฐานข้อมูลจะใช้ชื่อ Collection ว่า “features” สำหรับเก็บรายละเอียดของข้อมูลสิ่งที่น่าสนใจปรากฏการณ์บนโลก และความสัมพันธ์กับ Object อื่นที่เกี่ยวข้องเพื่อใช้ในการประกอบเป็น Response เมื่อผู้ใช้งานร้องขอเข้ามาได้

ตารางที่ 3.8 แสดงรายละเอียดของโครงสร้างข้อมูลที่เก็บใน Collection “features”

ลำดับ	ฟิลด์	ประเภทข้อมูล	คำอธิบาย
1	_id	ObjectId	(PK) รหัสข้อมูล
2	id	String	รหัสข้อมูลสิ่งที่น่าสนใจปรากฏการณ์บนโลก
3	type	String	ประเภทของข้อมูลสิ่งที่น่าสนใจปรากฏการณ์บนโลก
4	stac_version	String	เวอร์ชันของ STAC ที่ข้อมูลสิ่งที่น่าสนใจปรากฏการณ์บนโลกอ้างอิง
5	stac_extensions	[String]	ส่วนขยายที่ข้อมูลสิ่งที่น่าสนใจปรากฏการณ์บนโลกอ้างอิง

6	geometry	Object	<<embedded>> รูปร่างของข้อมูล
7	bbox	[Float]	ขอบเขตของข้อมูล ในรูปแบบ Bounding Box
8	properties	Map<String, Interface>	<<extension>> คุณสมบัติของข้อมูลสิ่งทีนำเสนอ ปรากฏการณ์บนโลก
9	assets	Map<String, Object>	<<embedded>> รายการสินทรัพย์ที่ผู้ใช้งานสามารถดาวน์โหลดได้
10	collection	String	รหัสข้อมูลของกลุ่มของข้อมูล ประเภท Collection ที่ข้อมูล สิ่งทีนำเสนอปรากฏการณ์บนโลกอยู่ภายใต้
11	parents	[String]	(FK) รายการรหัสข้อมูลของ Object ที่มีความสัมพันธ์เป็นแม่ ของกลุ่มของข้อมูล
12	created_at	ISODate	วัน-เวลา ที่สร้างข้อมูล
13	updated_at	ISODate	วัน-เวลา ที่ปรับปรุงข้อมูลล่าสุด
14	deleted_at	ISODate	วัน-เวลา ที่ลบข้อมูล

3.2.2.1 ข้อมูลรูปร่าง

ข้อมูลเก็บในฟิลด์ที่ชื่อว่า “geometry” เพื่อเก็บรายละเอียดของรูปร่างของข้อมูล โดยมีโครงสร้างแต่ละตัว ดังนี้

ตารางที่ 3.9 แสดงรายละเอียดของโครงสร้างข้อมูล geometry

ลำดับ	ฟิลด์	ประเภทข้อมูล	คำอธิบาย
1	type	String	ประเภทของรูปร่าง
2	coordinates	Interface	พิกัดแต่ละจุดยอดของรูปร่างตามประเภทของรูปร่างที่ได้ระบุไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.2 ข้อมูลสินทรัพย์

ข้อมูลเก็บในฟิลด์ที่ชื่อว่า “assets” เพื่อเก็บรายละเอียดของสินทรัพย์ที่สามารถดาวน์โหลดได้ โดยมีโครงสร้างแต่ละตัว ดังนี้

ตารางที่ 3.10 แสดงรายละเอียดของโครงสร้างข้อมูล asset

ลำดับ	ฟิลด์	ประเภทข้อมูล	คำอธิบาย
1	href	String	URI ที่นำไปสู่สินทรัพย์
2	title	String	ชื่อเรียกของสินทรัพย์
3	description	String	รายละเอียดของสินทรัพย์
4	type	String	Media Type ของสินทรัพย์
5	roles	[String]	หน้าที่ หรือคุณสมบัติของสินทรัพย์

3.2.2.3 ข้อมูลคำอธิบายข้อมูลทั่วไป

ข้อมูลสามารถใช้ในการอธิบายรายละเอียดของสิ่งที่มีเลขปรากฏการณ์บนโลก โดยอยู่ภายใต้ฟิลด์ที่ชื่อว่า “properties” เพื่อเก็บคุณสมบัติของสิ่งที่มีเลขปรากฏการณ์บนโลก โดยมีโครงสร้างแต่ละตัว ดังนี้

ตารางที่ 3.11 แสดงรายละเอียดของโครงสร้างข้อมูล common metadata

ลำดับ	ฟิลด์	ประเภทข้อมูล	คำอธิบาย
1	title	String	ชื่อเรียกของข้อมูล
2	description	String	รายละเอียดของข้อมูล
3	datetime	ISODate	เป็นวัน-เวลาของสินทรัพย์
4	created	ISODate	วัน-เวลา ที่สร้างข้อมูล
5	updated	ISODate	วัน-เวลา ที่ปรับปรุงข้อมูลล่าสุด
6	start_datetime	ISODate	วัน-เวลาเริ่มต้นสำหรับข้อมูล

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7	end_datetime	ISODate	วัน-เวลาสิ้นสุด สำหรับข้อมูล
8	license	String	ใบอนุญาตที่อ้างอิง
9	providers	[Object]	<<embedded>> รายการผู้ให้บริการข้อมูล
10	platform	String	แพลตฟอร์มที่เครื่องมือนี้ถูกติดตั้งอยู่
11	instruments	[String]	รายการของเครื่องมือ หรือเซนเซอร์ที่ใช้
12	constellation	String	ชื่อกลุ่มที่แพลตฟอร์มอยู่ภายใต้
13	mission	String	ชื่อของเป้าหมายว่าต้องการเก็บข้อมูลใด
14	gsd	Float	Ground Sample Distance ที่เซนเซอร์ตรวจจับได้ มีหน่วยเป็น เมตร และมีค่ามากกว่า 0

3.2.2.3.1 ข้อมูลผู้ให้บริการ

ข้อมูลเก็บในฟิลด์ที่ชื่อว่า “providers” เพื่อเก็บรายการของผู้ให้บริการข้อมูล โดยมีโครงสร้างแต่ละตัว ดังนี้

ตารางที่ 3.12 แสดงรายละเอียดของโครงสร้างข้อมูล provider

ลำดับ	ฟิลด์	ประเภทข้อมูล	คำอธิบาย
1	name	String	ชื่อบุคคล หรือบุคคลที่ให้บริการข้อมูล
2	description	String	รายละเอียดของผู้ให้บริการ
3	roles	[String]	หน้าที่ของผู้ให้บริการ
4	url	String	URL ที่ขึ้น ไปสู่หน้า Homepage ของผู้ให้บริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 ข้อมูลคุณสมบัติที่สามารถค้นหาได้

ในฐานข้อมูลจะใช้ชื่อ Collection ว่า “queryables” สำหรับเก็บรายละเอียดของข้อมูลคุณสมบัติที่สามารถค้นหาได้ เพื่อเป็นข้อมูลประกอบข้อจำกัดในการข้อมูลที่ผู้ใช้งานต้องการ

ตารางที่ 3.13 แสดงรายละเอียดของโครงสร้างข้อมูลที่อยู่ใน Collection “queryables”

ลำดับ	ฟิลด์	ประเภทข้อมูล	คำอธิบาย
1	_id	ObjectID	(PK) รหัสข้อมูล
2	schema	String	ข้อกำหนดของ schema ที่อ้างอิง
3	type	String	ประเภทของข้อมูล
4	title	String	ชื่อเรียกของ schema นี้
5	description	String	คำอธิบายของ schema นี้
6	properties	Map<String,Object>	รายละเอียดของแต่ละ ฟิลด์ ที่อนุญาตให้ค้นหา
7	additionalProperties	Boolean	รูปแบบของการอนุญาตให้สามารถค้นหาด้วยฟิลด์ที่นอกเหนือจากที่ระบุไว้
8	collection	String	(FK) รหัสข้อมูลของกลุ่มของข้อมูล ประเภท Collection ที่ข้อมูลคุณสมบัติที่สามารถค้นหาได้ระบุถึง
9	created_at	ISODate	วัน-เวลา ที่สร้างข้อมูล
10	updated_at	ISODate	วัน-เวลา ที่ปรับปรุงข้อมูลล่าสุด
11	deleted_at	ISODate	วัน-เวลา ที่ลบข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3.1 ข้อมูลคุณสมบัติ

เป็นรายละเอียดของข้อมูลที่เก็บในแต่ละ key ของฟิลด์ “properties” เพื่อเก็บรายการของคุณสมบัติของแต่ละ key โดยมีโครงสร้างแต่ละตัว ดังนี้

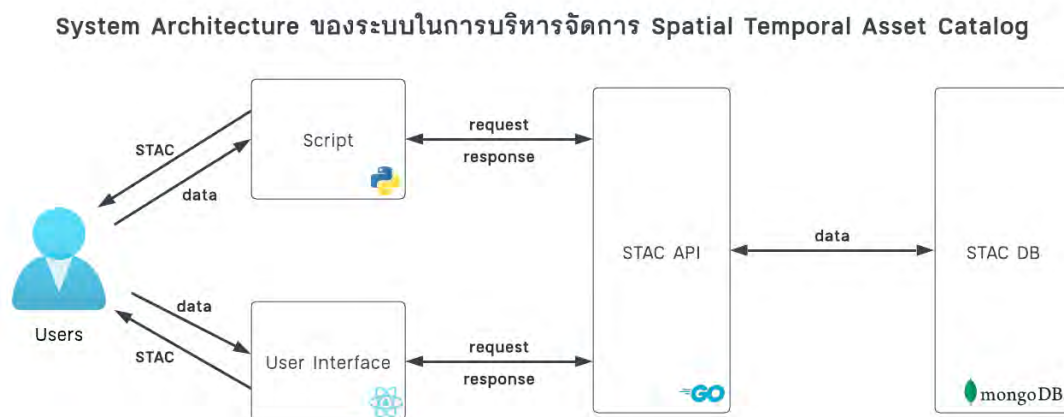
ตารางที่ 3.14 แสดงรายละเอียดของโครงสร้างข้อมูล property

ลำดับ	ฟิลด์	ประเภทข้อมูล	คำอธิบาย
1	title	String	ชื่อเรียก
2	type	String	ประเภทข้อมูล
3	description	String	คำอธิบาย
4	minLength	Integer	ความยาวขั้นต่ำของค่า
5	maxLength	Integer	ความยาวขั้นสูงของค่า
6	pattern	String	รูปแบบของค่าที่อนุญาต
7	multipleOf	Integer	ตัวกำหนดรูปแบบของค่าที่อนุญาต
8	minimum	Integer	ค่าต่ำสุด
9	exclusiveMinimum	Boolean	สถานะระบุว่าค่าต่ำสุดที่รองรับรวมค่าที่ระบุใน minimum ด้วยหรือไม่
10	maximum	Integer	ค่าสูงสุด
11	exclusiveMaximum	Boolean	สถานะระบุว่าค่าสูงสุดที่รองรับรวมค่าที่ระบุใน maximum ด้วยหรือไม่
12	format	String	รูปแบบของค่าที่รองรับ
13	enum	[Interface]	รายการของค่าที่อนุญาต
14	ref	String	โครงสร้างที่อ้างอิงมาจากภายนอก

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้ทำซ้ำโดยไม่ได้รับอนุญาตจากทางสถาบันฯ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 System Architecture

System Architecture ของระบบในการบริหารจัดการ Spatial Temporal Asset Catalog อธิบายถึงการติดต่อกันของแต่ละส่วน โดยมีรายละเอียดดังรูปที่ 3.2



รูปที่ 3.2 System Architecture ของระบบในการบริหารจัดการ Spatial Temporal Asset Catalog

ส่วนที่ 1 อธิบายถึงช่องทางในการที่ Users สามารถมีปฏิสัมพันธ์กับระบบได้ โดยสามารถทำได้ 2 ช่องทาง คือ การใช้สคริปต์คำสั่งในการท างาน โดยในรูปแบบนี้จะเกี่ยวข้องไปถึงการท างานแบบอัตโนมัติ หรือการติดต่อผ่าน User Interface โดยในรูปแบบนี้ Users จะเป็นการจัดการด้วยตัวเอง

ส่วนที่ 2 STAC API เป็นส่วนของ logic ของระบบในการท างานฟังก์ชันต่าง ๆ

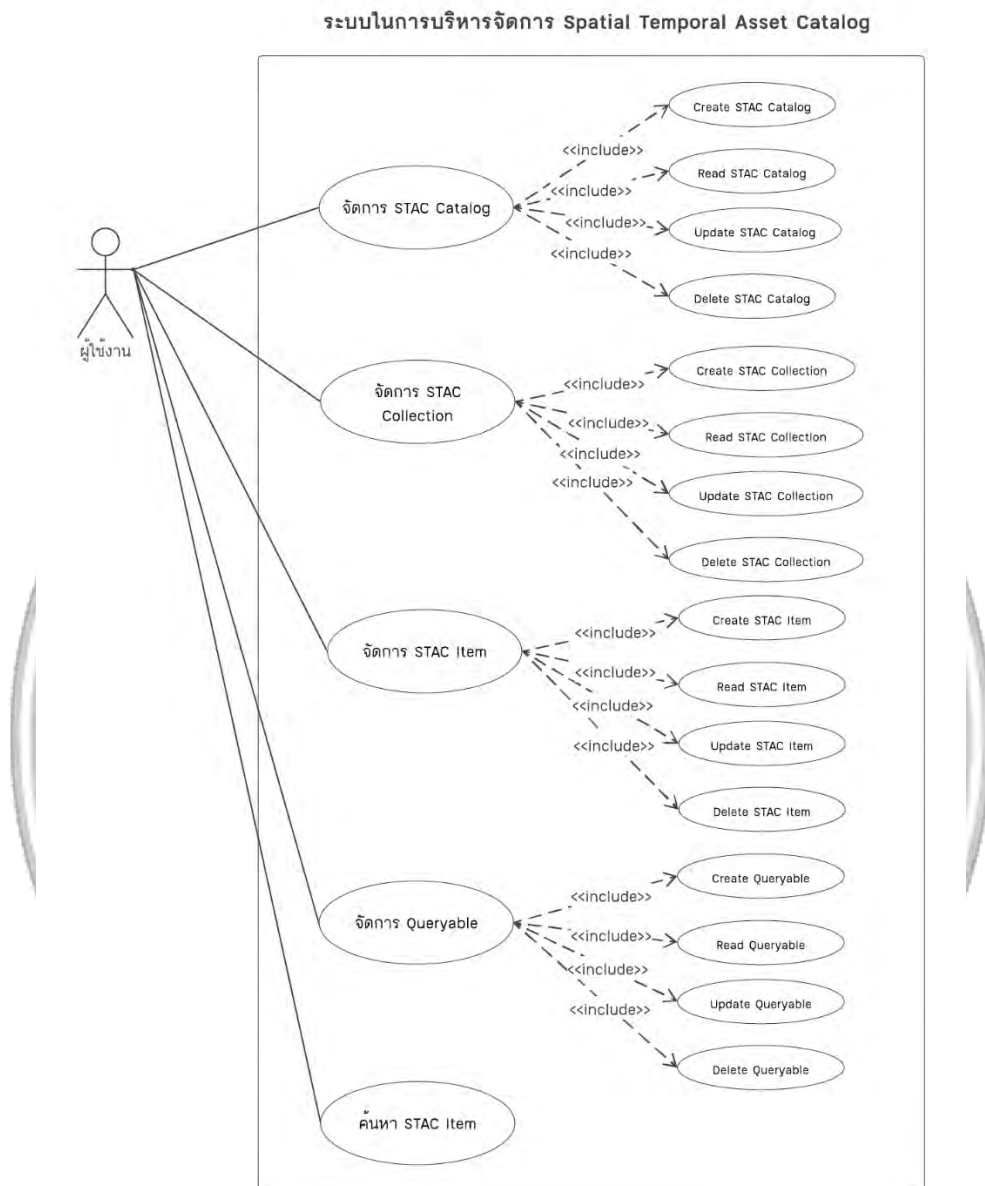
ส่วนที่ 3 STAC Database เป็นส่วนเก็บข้อมูลทั้งหมดเกี่ยวกับระบบ

ในการพัฒนาโครงการสหกิจศึกษาที่จะกล่าวถึงในข้อต่อ ๆ ไปนั้น จะเป็นรายละเอียดของส่วนที่ 2 logic ของระบบในการบริหารจัดการและค้นหาข้อมูลใน STAC ซึ่งพัฒนาเป็น REST API ด้วยภาษาโปรแกรม Go เพื่อติดต่อกับข้อมูลที่เก็บอยู่ในฐานข้อมูล mongoDB ในส่วนที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 Use Case Diagram

Use Case Diagram นี้เป็นระบบในการบริหารจัดการ Spatial Temporal Asset Catalog มีผู้กระทำการ 1 คือผู้ใช้งาน โดยมีรายละเอียดดังรูปที่ 3.3



รูปที่ 3.3 Use Case Diagram ของระบบในการบริหารจัดการ Spatial Temporal Asset Catalog

จากแผนภาพด้านบน ในกรณีของการบริหารจัดการแค็ตตาล็อกสินทรัพย์ทางภูมิสารสนเทศ 1 แค็ตตาล็อก ผู้ใช้งานสามารถจัดการ STAC Catalog ได้ กล่าวคือ สามารถสร้าง เรียกดู แก้ไข และลบข้อมูล STAC object ประเภท Catalog ได้

ผู้ใช้งานสามารถจัดการ STAC Collection ได้ กล่าวคือ สามารถสร้าง เรียกดู แก้ไข และลบข้อมูล STAC object ประเภท Collection ได้ ผู้ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้ใช้งานสามารถจัดการ STAC Item ได้ กล่าวคือ สามารถสร้าง เรียกดู แก้ไข และลบข้อมูล STAC object ประเภท Item ได้

ผู้ใช้งานสามารถจัดการ Queryable ได้ กล่าวคือ สามารถสร้าง เรียกดู แก้ไข และลบข้อมูล Queryable ได้

ผู้ใช้งานสามารถค้นหา STAC Item ตามที่ต้องการได้ โดยระบุเงื่อนไขตามพารามิเตอร์ที่ใช้ในการสืบค้น โดยระบบจะตอบรายการของ STAC Item ที่เงื่อนไขตรงตามที่ผู้ใช้งานระบุไว้กลับไปให้ผู้ใช้งาน ในรูปแบบของ GeoJSON

3.5 Use Case Description

จาก Use Case Diagram ในข้อที่ 3.4 สามารถเขียน Use Case Description ของระบบในการบริหารจัดการ Spatial Temporal Asset Catalog โดยมีรายละเอียดดังนี้

3.5.1 คำอธิบาย Use Case จัดการ STAC Catalog

ตารางที่ 3.15 แสดงรายละเอียดเกี่ยวกับ Use Case จัดการ STAC Catalog

ชื่อ	จัดการ STAC Catalog
คำอธิบาย	สามารถสร้าง เรียกดู แก้ไข และลบข้อมูล STAC object ประเภท Catalog ได้
ข้อมูลเข้า	Catalog ID, STAC object ประเภท Catalog
ข้อมูลออก	STAC Catalog ที่ตรงตามข้อ กหนด
เงื่อนไข	ชื่อ STAC Catalog ไม่สามารถซ้ำกันได้ และต้องระบุรายละเอียดที่จ าเป็นมาให้ครบ
หมายเหตุ	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.2 คำอธิบาย Use Case จัดการ STAC Collection

ตารางที่ 3.16 แสดงรายละเอียดเกี่ยวกับ Use Case จัดการ STAC Collection

ชื่อ	จัดการ STAC Collection
คำอธิบาย	สามารถสร้าง เรียกดู แก้ไข และลบข้อมูล STAC object ประเภท Collection ได้
ข้อมูลเข้า	Catalog ID, Collection ID, STAC object ประเภท Collection
ข้อมูลออก	STAC Collection ที่ตรงตามข้อกำหนด
เงื่อนไข	ต้องท การสร้าง STAC Catalog มาก่อน, ชื่อ STAC Collection ไม่สามารถซ้ำกันได้ และซ้ำกับชื่อ STAC Catalog ไม่ได้ และต้องระบุรายละเอียดที่จำเป็นมาให้ครบ
หมายเหตุ	-

3.5.3 คำอธิบาย Use Case จัดการ STAC Item

ตารางที่ 3.17 แสดงรายละเอียดเกี่ยวกับ Use Case จัดการ STAC Item

ชื่อ	จัดการ STAC Item
คำอธิบาย	สามารถสร้าง เรียกดู แก้ไข และลบข้อมูล STAC object ประเภท Item ได้
ข้อมูลเข้า	Catalog ID, Collection ID, Item ID, STAC object ประเภท Item
ข้อมูลออก	STAC Item ที่ตรงตามข้อกำหนด
เงื่อนไข	ต้องท การสร้าง STAC Catalog และ STAC Collection มาก่อน, ชื่อ STAC Item ไม่สามารถซ้ำกันได้ และต้องระบุรายละเอียดที่จำเป็นให้ครบ
หมายเหตุ	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.4 คำอธิบาย Use Case จัดการ Queryable

ตารางที่ 3.18 แสดงรายละเอียดเกี่ยวกับ Use Case Queryable

ชื่อ	จัดการ Queryable
คำอธิบาย	สามารถสร้าง เรียกดู แก้ไข และลบข้อมูล Queryable ได้
ข้อมูลเข้า	รายละเอียด Queryable
ข้อมูลออก	Schema ของ Queryable ของแต่ละ STAC
เงื่อนไข	ต้องท การสร้าง root STAC Catalog ก่อน
หมายเหตุ	-

3.5.5 คำอธิบาย Use Case ค้นหา STAC Item

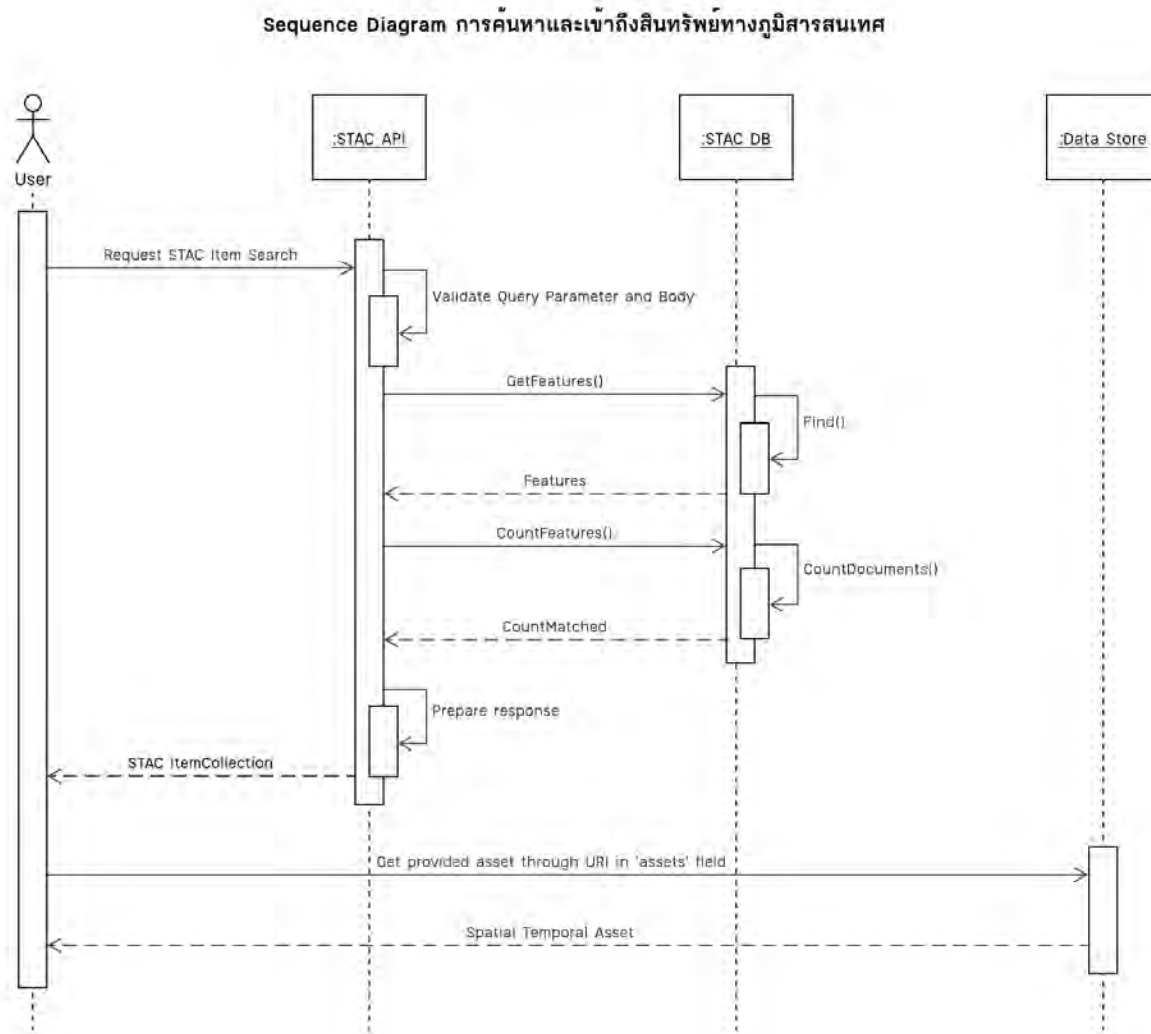
ตารางที่ 3.19 แสดงรายละเอียดเกี่ยวกับ Use Case ค้นหา STAC Item

ชื่อ	ค้นหา STAC Item
คำอธิบาย	สามารถค้นหา STAC Item ตามที่ต้องการได้ โดยระบุเงื่อนไขตามพารามิเตอร์ที่ใช้ในการสืบค้น
ข้อมูลเข้า	Catalog ID, เงื่อนไขตามแต่ละพารามิเตอร์ที่ใช้ในการสืบค้น
ข้อมูลออก	รายการของ STAC Item ที่มีเงื่อนไขตรงกับที่ป้อนเข้ามา
เงื่อนไข	ต้องค้นหาใน root STAC Catalog ที่มีอยู่จริง และหากต้องการค้นหาด้วยเงื่อนไขภายใน properties ต้องกำหนด Queryable ก่อน
หมายเหตุ	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 Sequence Diagram

Sequence Diagram นี้แสดงถึงลำดับการทำงานและการติดต่อสื่อสารกันระหว่าง Object ในระบบ กรณีที่ผู้ใช้งานจะค้นหา และเข้าถึงสินทรัพย์ทางภูมิสารสนเทศผ่าน STAC API โดยมีรายละเอียดดังรูปที่ 3.4



รูปที่ 3.3 Sequence Diagram การค้นหาและเข้าถึงสินทรัพย์ทางภูมิสารสนเทศ

จากแผนภาพด้านบน

- 1) User ร้องขอเข้ามาผ่าน Endpoint STAC Item Search โดยกรอกเงื่อนไขที่ต้องการค้นหาผ่าน Query Parameter หรือ Body
- 2) ในส่วน STAC API เมื่อถูกร้องขอเข้ามา จะทำการตรวจสอบเงื่อนไขที่ user กรอกเข้ามา หากไม่ถูกต้อง ก็จะมีการแจ้งเออเร่อกลับไป
- 3) จากนั้น STAC API จะไปค้นหา record ของข้อมูลใน collection ใน STAC DB ที่ตรงตามเงื่อนไข

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับอาจารย์ใช้ในการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4) และจะมีการเรียก STAC DB เพื่อนับจำนวนของผลลัพธ์ทั้งหมดที่ตรงตามเงื่อนไข แล้วตอบกลับไปให้ STAC API
- 5) จากนั้นฝั่ง STAC API จะเตรียมโครงสร้างสิ่งที่จะตอบกลับไปให้ user ให้ตรงตามโครงสร้างของ STAC ItemCollection
- 6) แล้วจึงตอบกลับ STAC ItemCollection นั้นไปให้ user
- 7) เมื่อ user ต้องการสินทรัพย์ที่ STAC Item ที่ระบบตอบกลับไปอ้างอิงถึง รายละเอียดเหล่านั้นจะถึงระบอบอยู่ใน field 'assets' และเมื่อ user ไปตาม URI ที่ระบุไว้ ก็จะพา user ไปสินทรัพย์นั้นที่ถูกเก็บอยู่ในแหล่งจัดเก็บข้อมูล ซึ่ง user สามารถนำสินทรัพย์ไปใช้ต่อได้

3.7 เทคโนโลยีที่ใช้ในการพัฒนา

ในการพัฒนาระบบในการบริหารจัดการ Spatial Temporal Asset Catalog ของส่วน Backend Programming มีเทคโนโลยีที่ใช้ในการพัฒนาดังนี้

3.7.1 Go Programming Language

ภาษา Go หรือ Golang เป็นภาษาโปรแกรมแบบกระบวนคำสั่ง (Procedural Programming Language) ที่ถูกพัฒนาขึ้นที่ Google โดย Robert Griesemer, Rob Pike และ Ken Thompson ในปี ค.ศ. 2007 เพื่อมาใช้ในการแก้ปัญหา หรือข้อจำกัดที่พบในภาษาโปรแกรมที่ถูกใช้ในบริษัท และถูกเปิดตัวเป็นภาษาโปรแกรมแบบ Open-source ในปี ค.ศ. 2009

จุดเด่นของภาษา Go

Go เป็น Compiled Language คือ คอมไพเลอร์ (Compiler) จะอ่าน Source Code และแปลผลให้เป็นภาษาเครื่อง (Executable File) ที่สามารถทำงานบนแพลตฟอร์มนั้น ๆ ได้เลย

Go เป็น Statically-typed Language คือ ตัวแปรต้องกำหนดชนิดของข้อมูล (Data Type) ตั้งแต่ในตอนประกาศตัวแปร ซึ่งคอมไพเลอร์จะสามารถตรวจสอบชนิดของข้อมูลได้ว่าถูกต้องหรือไม่ ตั้งแต่ช่วงของการคอมไพล์ (Compile)

ลักษณะของภาษา Go

- มีไวยากรณ์ (Syntax) คล้ายภาษา C โดยเน้นความง่ายในการเขียนและการอ่าน
- ใช้เวลาในการคอมไพล์น้อย
- สามารถทำ Concurrent Programming ได้ง่าย เพราะถูกออกแบบมาสำหรับเขียนโปรแกรมแบบ Parallel โดยเฉพาะ
- มีระบบ Package ทั้งแบบที่เป็น Standard Libraries และ External Libraries เพื่อใช้ในการเขียน

เอกสารนี้ ประกอบกันให้ได้เป็นโปรแกรม หรือระบบขึ้นมามากกว่าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีการคืนหน่วยความจำจาก Object ที่ไม่ใช้งานแล้ว (Garbage Collection) ทำให้โปรแกรมเมอร์ไม่ต้องคอยจัดการเอง
- เหมาะสำหรับการไปเขียนโปรแกรมแนว System Programming เช่นการสร้าง API Server หรือ Network Application เป็นต้น

ตัวอย่างการนำไปใช้งาน

- Web Development & Backend Development

Go เป็นภาษาที่เหมาะสมสำหรับการทำ Web Development มาก ๆ เพราะ Go สามารถใช้สร้างระบบที่รองรับการทำงานใน Scale ใหญ่ที่มี Request จำนวนมากได้ ซึ่ง Go ถูกออกแบบมาเพื่อทำงานประเภทนี้โดยเฉพาะ ยกตัวอย่างเช่น มี HTTP Package อยู่ใน Standard Library ของ Go เลยโดยไม่ต้องลง Library เพิ่มเติม หรือจะใช้ Framework ต่างๆ ช่วยให้การทำ Web Development ได้ง่ายและสะดวกขึ้น เช่น Gin Gonic, Martini, Buffalo

- DevOps Automation

Go เป็นภาษาที่มี Syntax ที่ง่ายและเบา ทำให้เหมาะกับการใช้สร้าง Script ง่าย ๆ ในการทำ Tasks ต่าง ๆ ได้ดี โดยเฉพาะ Tasks ในด้านงานของ DevOps เช่นการทำ Task Automation ต่าง ๆ ใน CI/CD Pipeline เป็นต้น นอกจากนี้ Go ยังสามารถถูก Compile ได้อย่างรวดเร็ว จึงเหมาะกับงานประเภทนี้มากเช่นกัน

- Cloud Computing Systems

Go ยังสามารถใช้ในการพัฒนา Cloud Computing Systems ที่ Scalable ได้ดีมาก ๆ ด้วยยกตัวอย่างเช่น Docker และ Kubernetes ซึ่ง 2 โปรเจกต์นี้นั้นใช้ Go เป็นภาษาหลักในการพัฒนา อีกทั้ง Dropbox ก็ได้เปลี่ยนระบบหลังบ้านจาก Python มาเขียนใหม่ด้วยภาษา Go แทนเช่นกัน

- Command Line Interface Tools

Go ยังสามารถนำมาใช้สร้าง command-line interfaces หรือ CLI ได้ง่าย ๆ ยกตัวอย่างเช่น docker-cli ของ Docker และ kubectl ของ Kubernetes ซึ่ง Go เองก็มี Framework ตัวช่วยอำนวยความสะดวกในการสร้าง CLI เช่น ใช้ Cobra ร่วมกับ Viper

- Other Applications

Go ยังสามารถนำมาใช้สำหรับ Application อื่น ๆ ได้อีกมากมาย เช่น การทำ Robot, Drone, IoT (Internet of Things) ด้วย Tools อย่าง Gobot

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.2 MongoDB

MongoDB เป็น Open-source Document Database ที่สามารถใช้งานข้ามแพลตฟอร์มได้ มีประสิทธิภาพสูง ตอบสนองการบริการได้ดี และยังสามารถ Scale ได้ง่าย ซึ่งเป็นฐานข้อมูลแบบ NoSQL คือไม่มีความสัมพันธ์ (Relation) ของตารางแบบ Relational Database ทั่ว ๆ ไป แต่จะเก็บข้อมูลเป็นแบบ JSON (JavaScript Object Notation) แทน

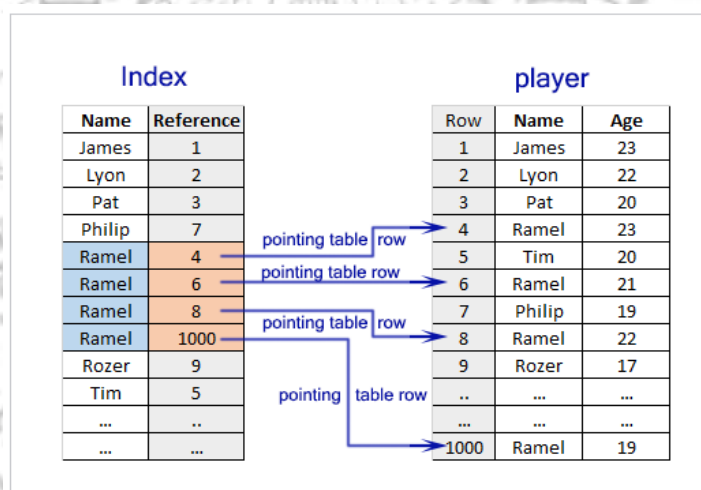
คุณสมบัติหลักของ MongoDB

- Ad-hoc queries

MongoDB รองรับการค้นหาตาม Field ช่วงของค่าที่สนใจ และการค้นหาโดยใช้ Regular Expression โดยในการ Query แต่ละครั้งสามารถตอบกลับมาแค่ Field ที่ต้องการได้ และผู้ใช้งานยังสามารถเขียนฟังก์ชัน JavaScript ร่วมด้วยได้

- Indexing

Field ของ MongoDB Document สามารถจัดทาดัชนีด้วย Primary และ Secondary Index ที่มีความสามารถในการค้นหาข้อมูลได้อย่างรวดเร็วในข้อมูลที่มีปริมาณมาก ๆ ได้และสามารถค้นหาข้อมูลได้จาก ข้อมูลในส่วนใดก็ได้

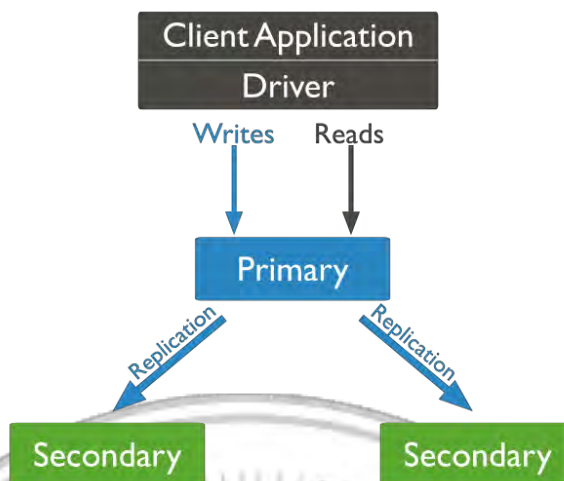


รูปที่ 3.4 ตัวอย่าง Index Table

(ที่มา: <https://www.w3resource.com/mongodb/introduction-to-mongodb-index.php>)

- Replication

Replication คือการกระจายข้อมูลท่ามกลาง MongoDB Servers หลาย ๆ ตัว (หรือ หลาย ๆ Node) โดย MongoDB สามารถกระจายข้อมูลไปยัง 1 หรือมากกว่านั้นและข้อมูลจะ Sync กันตลอดเวลาเมื่อมีข้อมูลเปลี่ยนแปลง ซึ่ง Replication แบบนี้จะทำงานผ่านสิ่งที่เรียกว่า Replica Set โดยที่ Replica Sets ก็คือกลุ่มของ Nodes ที่ถูกตั้งค่าให้ Sync กันอัตโนมัติ และนอกจากการ Sync ข้อมูลแล้ว หาก Node หลัก (Primary Node) เสียหายหรือไม่สามารถเข้าถึงได้ มันก็จะทำ Automatic Failover ให้เราอีกด้วย



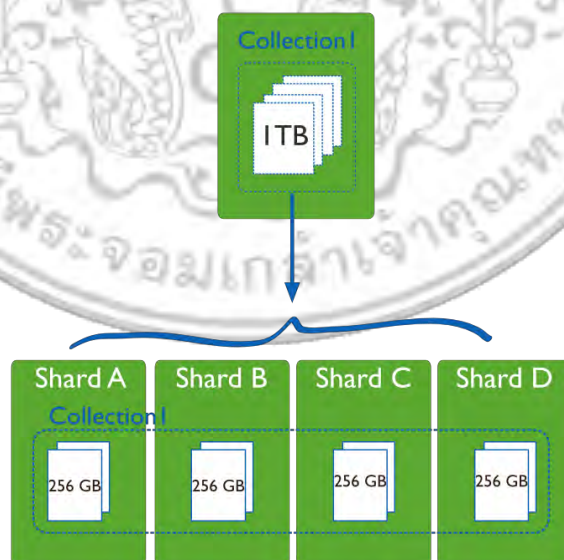
รูปที่ 3.5 การทำ Replication ของ MongoDB

(ที่มา: <https://www.khomkrit.com/mongodb-data-replication/>)

• Load balancing

MongoDB สามารถ Scale ตามแนวนอนได้โดยการใช้ Sharding ซึ่งผู้ใช้งานจะต้องเลือก Shard Key เพื่อใช้ในการแบ่งข้อมูลใน Collection เป็นท่อน ๆ และกระจายข้อมูลไปเก็บกันในแต่ละ Shard เพื่อให้ประสิทธิภาพในการเขียนและอ่านข้อมูลเป็นไปได้ดีขึ้น

MongoDB ยังสามารถทำงานได้บน Server หลายเครื่อง ทำให้สามารถช่วยแบ่งเบาภาระงานกันได้ หรือมีการคัดลอกข้อมูลเก็บไว้เพื่อให้ระบบยังสามารถทำงานต่อได้ในกรณีที่เกิดความขัดข้องของอุปกรณ์



รูปที่ 3.6 การทำ Sharding กับ Collection ของ MongoDB

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือสงวนชื่อผู้เผยแพร่และเนื้อหา ซึ่งผู้เผยแพร่เอกสารฉบับนี้มิได้ให้การรับประกันหรือการันตีในด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **File storage**

MongoDB สามารถใช้เป็น File System เรียกว่า GridFS พร้อมกับคุณสมบัติในการทำ Load Balancing และ Data Replication บนหลาย ๆ เครื่อง สำหรับใช้ในการเก็บข้อมูล ซึ่งฟังก์ชัน Grid File System นี้จะช่วยนักพัฒนาในการจัดการไฟล์ และเนื้อหา โดยจะแบ่งไฟล์เป็นส่วน ๆ แล้วเก็บแยกกัน เพื่อจะรองรับการเพิ่มขึ้น หรือ ลดของปริมาณข้อมูลได้

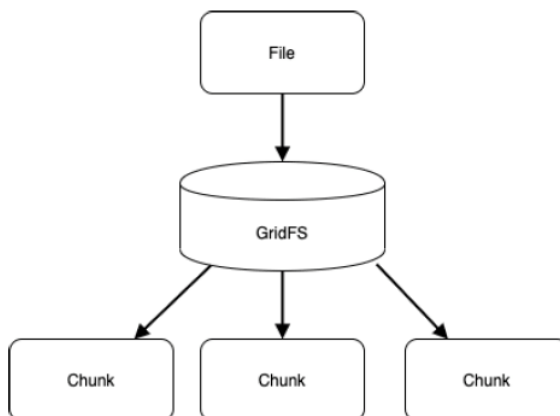


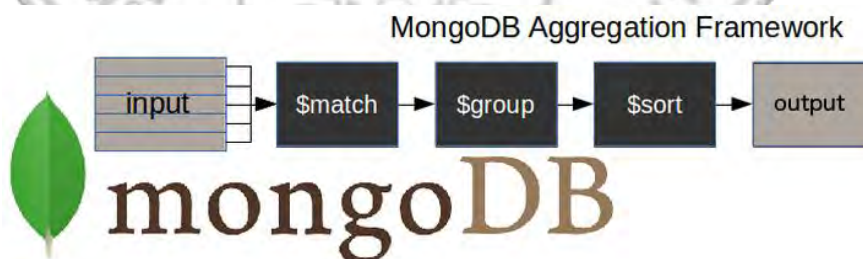
Fig 7: MongoDB GridFS

รูปที่ 3.7 GridFS ของ MongoDB

(ที่มา: <https://dev.to/nyashanziramasanga/document-store-database-mongodb-part-3-4226>)

- **Aggregation**

MongoDB ได้จัดเตรียมการทำ Aggregation ไว้ 3 รูปแบบ คือ Aggregation Pipeline, Map-reduce Function และ Single-purpose Aggregation Method ซึ่ง Aggregation Framework ของ MongoDB จะช่วยให้ผู้ใช้งานได้ผลลัพธ์เหมือนกับคำสั่ง GROUP BY ของ SQL



รูปที่ 3.8 ตัวอย่าง MongoDB Aggregation Pipeline

(ที่มา: <https://studio3t.com/knowledge-base/articles/mongodb-aggregation-framework/>)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Server-side JavaScript execution**

ภาษาโปรแกรม JavaScript สามารถถูกใช้ในการเขียน Query ฟังก์ชัน Aggregation เช่น MapReduce หรือถูกส่งไปหาฐานข้อมูลโดยตรงเพื่อให้ สังกูทท งาน

- **Capped collections**

MongoDB รองรับ Collection ที่มีขนาดตายตัว เรียกว่า Capped Collection โดย Collection ประเภทนี้จะรักษาลำดับในการเพิ่มข้อมูล และเมื่อถึงขนาดที่กำหนดไว้ ก็จะมีพฤติกรรมเหมือน Circular Queue ซึ่งจะทยอยลบข้อมูลเก่า ๆ แล้วเอาข้อมูลใหม่มาใส่แทนได้ ซึ่งจะเหมาะแก่การนำมาทำ Log เป็นอย่างมาก เพราะจะ Clear ข้อมูลเก่า ๆ ที่เก็บไว้นานเกินไปไว้ให้อัตโนมัติ ทำให้ข้อมูลไม่ใหญ่กว่าที่กำหนดไว้

- **Transactions**

MongoDB รองรับ Multi-document Transactions สำหรับกรณีที่ต้องการ Atomicity ในการอ่านหรือเขียน หลาย Document และด้วย Distributed Transactions ทำให้ธุรกรรมสามารถถูกใช้ข้าม Operation, Collection, Database, Document, Shard หลายส่วนได้

องค์ประกอบของ MongoDB

- **Database**

เป็นส่วนที่ใช้เก็บ Collection ไว้ด้วยกัน โดย Database แต่ละตัวก็จะมี File และ File System เป็นของตัวเอง ซึ่งโดยทั่วไปแล้วในหนึ่ง MongoDB Server ก็จะมีหลาย Database เป็นส่วนประกอบ

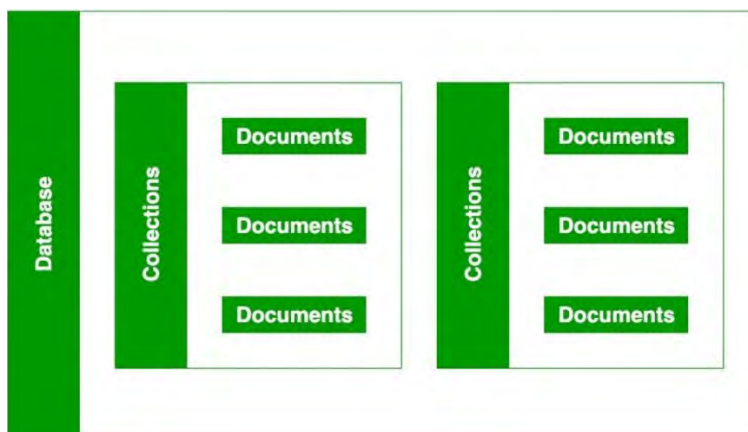
- **Collection**

เป็นกลุ่มของ MongoDB Document เปรียบเทียบได้กับ Table ใน Relational Database ทั่วไป แต่แตกต่างกันที่ Collection ไม่จำเป็นต้องมีโครงสร้างของข้อมูล (Schema) เหมือนกันก็สามารถบันทึกข้อมูลได้ ทำให้สามารถเก็บข้อมูลที่มีโครงสร้างต่างกันไว้ใน Collection เดียวกันได้

- **Document**

เป็นชุดข้อมูลที่มีลักษณะเป็น key-value ซึ่ง Document ที่อยู่ใน Collection เดียวกันไม่จำเป็นต้องมีข้อมูลหรือโครงสร้างเหมือนกันได้ และยังสามารถเก็บค่าได้หลายชนิดของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 ความสัมพันธ์ของแต่ละองค์ประกอบของ MongoDB

(ที่มา: <https://www.geeksforgeeks.org/mongodb-database-collection-and-document/>)

เปรียบเทียบองค์ประกอบของ Relational Database และ MongoDB

ตารางที่ 3.20 แสดงการเปรียบเทียบองค์ประกอบของ Relational Database และ MongoDB

Relational Database	MongoDB
Database	Database
Table	Collection
Record / Row	Document / Object
Column	Field / Key
Value	Value
Table Join	Embedded Documents
Primary Key	Primary Key (key เริ่มต้น ‘_id’ จะถูกกำหนดโดย MongoDB เอง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างของ Document ที่เก็บใน MongoDB

```
{
  _id: ObjectId(7df78ad8902c)
  title: 'MongoDB Overview',
  description: 'MongoDB is no sql database',
  by: 'tutorials point',
  url: 'http://www.tutorialspoint.com',
  tags: ['mongodb', 'database', 'NoSQL'],
  likes: 100,
  comments: [
    {
      user:'user1',
      message: 'My first comment',
      dateCreated: new Date(2011,1,20,2,15),
      like: 0
    },
    {
      user:'user2',
      message: 'My second comments',
      dateCreated: new Date(2011,1,25,7,45),
      like: 5
    }
  ]
}
```

รูปที่ 3.10 ตัวอย่างโครงสร้างของ Document ใน MongoDB

(ที่มา: https://www.tutorialspoint.com/mongodb/mongodb_overview.htm)

3.7.3 การพัฒนา API ตามแบบสถาปัตยกรรม REST

REST ย่อมาจาก Representational State Transfer คือรูปแบบของสถาปัตยกรรมซอฟต์แวร์ ที่ใช้ประโยชน์จากเทคโนโลยี Web protocol เพื่อใช้ในการสร้าง Web Service ซึ่งคุณ Roy Fielding Ph.D. ได้อธิบายไว้ในวิทยานิพนธ์ ตั้งแต่ปี ค.ศ. 2000

Architectural properties

โดยข้อกำหนดของ REST Architectural Style ทั้ง 7 ประกอบไปด้วย

- **Performance:** ประสิทธิภาพในการโต้ตอบ
- **Scalability:** มีความสามารถในการปรับขยายได้ทำให้สามารถรองรับส่วนประกอบจำนวนมาก
- **Simplicity:** ความเรียบง่ายของอินเตอร์เฟซที่เหมือนกัน
- **Modifiability:** ความสามารถในการปรับเปลี่ยนส่วนประกอบเพื่อตอบสนองความต้องการที่เปลี่ยนแปลง (แม้ในขณะที่แอปพลิเคชันทำงานอยู่)
- **Visibility:** การมองเห็นการสื่อสารระหว่าง Component ผ่าน Service Agents
- **Portability:** ง่ายต่อการโยกย้ายในส่วนของ Program และ Data
- **Reliability:** ความน่าเชื่อถือในการต้านทานความล้มเหลวในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RESTful Web Services (RWS)

คือ Web Service ที่ใช้ REST architectural style เป็นแบบในการพัฒนา โดย RESTful Web service อนุญาตให้ระบบ Request และเข้าถึง Resource บนเว็บโดยใช้ชุดคำสั่งที่กำหนดเอาไว้ล่วงหน้า โดยที่การโต้ตอบของระบบที่ใช้ REST จะอยู่บนพื้นฐานของ Hypertext Transfer Protocol (HTTP) ซึ่ง Request จะส่งค่าของ URI ที่กำหนด และขอ Response กลับมาเป็น Payload ในแบบ HTML, XML, JSON หรือรูปแบบอื่น ๆ

โดย RESTful จะประกอบไปด้วย

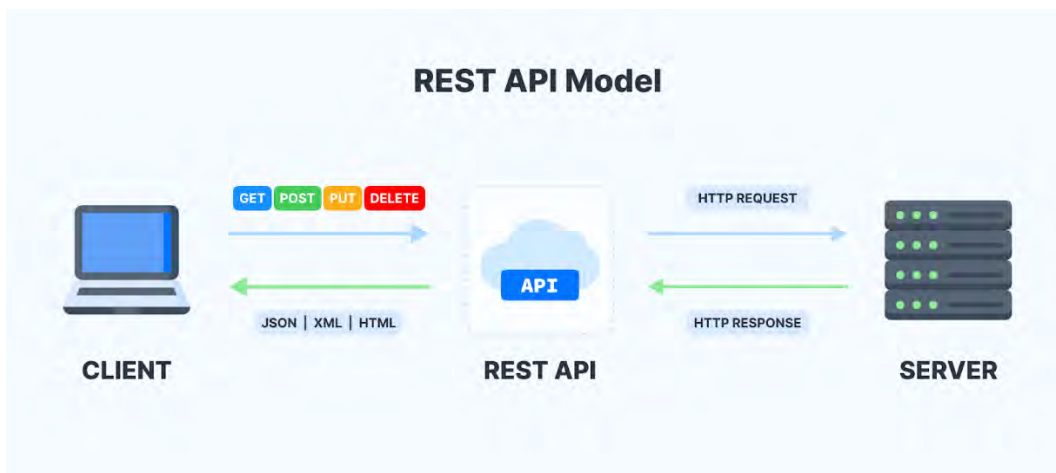
- **Client** คือ ผู้ที่จะเข้ามาร้องขอ Resources
- **Server** คือ ผู้เป็นผู้ให้บริการ Resources

Architectural constraints of RESTful API

6 ข้อจำกัดของ RESTful API ซึ่งถือเป็นสิ่งที่สำคัญในการสร้าง RESTful API ตามมาตรฐานซึ่งทำให้ง่ายต่อการพัฒนา และทำให้เป็นที่ยอมรับ (หากไม่ทำตามให้ครบทั้ง 6 ข้อจะไม่ถือว่าเป็น RESTful API ยกเว้นข้อที่เป็น Optional)

- **Client-server Architecture:** Client ไม่จำเป็นต้องรู้อะไรเกี่ยวกับ Business logic ภายใน ไม่มีหน้าที่เกี่ยวกับการจัดเก็บข้อมูล ส่วน Server มีหน้าที่เก็บ Resource และไม่จำเป็นต้องรู้เกี่ยวกับ UI Frontend หรือสถานะของผู้ร้องขอ
- **Statelessness:** เมื่อมีการส่ง Request รับ Response จาก Server เรียบร้อยแล้ว Server จะไม่เก็บข้อมูล Session ไว้ เพื่อลดภาระของ Server
- **Cacheability:** สามารถ Cache Response ได้ โดย Response จะต้องสามารถกำหนดได้ว่าจะ Cache หรือไม่ เพื่อป้องกันไม่ให้ User หรือ Client ได้รับข้อมูลเก่าในการ Request ครั้งถัดไป
- **Layered System:** ปกติ Client ไม่รู้ว่าที่ทำการเชื่อมต่อนั้น ได้เชื่อมต่อโดยตรงกับ Server ปลายทาง หรือไปยังตัวกลางอื่น ๆ ระหว่างทาง ทำให้สามารถออกแบบระบบให้เป็นชั้น ๆ เช่นการเพิ่ม Layer ส าหรับจัดการเรื่องความปลอดภัยไว้บนสุดหรือตัวกลางอาจจะทำการรวบรวม Response จากหลาย ๆ Server เพื่อตอบกลับไปในแต่ละครั้ง หรือทำ Load Balancing เพื่อเพิ่มประสิทธิภาพของระบบ
- **Code on demand (Optional):** Server สามารถขยาย หรือปรับแต่งการทำงานของ Client ได้ชั่วคราว โดยการส่งโค้ดที่สามารถทำงานได้ เช่น client-side scripts ใน JavaScript
- **Uniform interface:** เป็นพื้นฐานของการออกแบบและพัฒนาระบบ RESTful ใด ๆ ซึ่งจะช่วยลดความซับซ้อน และแยกส่วนออกจากกันเพื่อให้แต่ละส่วนสามารถพัฒนาขึ้นไปได้อย่างอิสระ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 Flow การติดต่อของ Client และ Server ผ่าน REST API

(ที่มา: <https://velog.io/@yhko1992/REST-API%EC%97%90-%EB%8C%80%ED%95%B4-%EC%84%A4%EB%AA%85%ED%95%B4%EB%B3%B4%EC%9E%90>)

HTTP Request

เป็นสิ่งที่เราส่งไปให้ Server ปลายทาง โดย HTTP Request จะประกอบไปด้วย

- **Request-Line** คือส่วนที่ระบุ HTTP Method, URI และ Version ของ Protocol ที่ใช้
- **Headers** คือส่วนที่ระบุข้อมูล และกฎต่าง ๆ ในการเชื่อมต่อ
- **Body** คือเนื้อหาของการร้องขอ ซึ่งจะถูกใช้เมื่อเราต้องการส่งข้อมูลไปให้กับเว็บเพจนั้น ๆ ประมวลผล

HTTP Request Method

คือส่วนที่ใช้กำหนดประเภทการกระทำของ ร้องขอ โดยจะมีอยู่ 5 Methods ที่ใช้งานบ่อย ดังตารางด้านล่าง

ตารางที่ 3.21 แสดงประเภทของ HTTP Request Method ที่ใช้งานบ่อย

Method	คำอธิบาย
GET	ใช้สำหรับร้องขอข้อมูลจากปลายทางที่กำหนดไว้ ซึ่งควรจะใช้ในการเรียกดูข้อมูลเท่านั้น และต้องไม่มีผลกระทบใด ๆ กับข้อมูล โดยสามารถระบุค่ามากับ Path Parameters หรือ Query Parameters เพื่อเป็นตัวเลือกในการค้นหา หรือระบุถึงข้อมูลที่เจาะจง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

POST	ใช้สำหรับสร้างข้อมูลใหม่โดยส่งข้อมูลผ่าน HTTP Body หรือเป็นการเรียกให้คาสั่งหรือฟังก์ชันทำงานเมื่อส่งรายละเอียดไปพร้อมกับ HTTP Body โดยจะต้องระบุค่าตามรูปแบบที่ระบบเข้าใจ
PUT	ใช้สำหรับแทนที่ข้อมูลที่มีทั้งหมดด้วยข้อมูลใหม่ที่ส่งเข้าไป โดยจะใช้ในการแก้ไขข้อมูลที่เก็บอยู่ในระบบ ซึ่งจะต้องมีการระบุค่าที่อ้างอิงถึงข้อมูลที่ต้องการแก้ไขมาทั้งกับ Path Parameters หรือ Query Parameters และรายละเอียดของข้อมูลใหม่มาทั้งกับ HTTP Body
PATCH	ใช้สำหรับแก้ไขรายละเอียดของข้อมูลบางส่วน ซึ่งจะต้องมีการระบุค่าที่อ้างอิงถึงข้อมูลที่ต้องการแก้ไขมาทั้งกับ Path Parameters หรือ Query Parameters และรายละเอียดของข้อมูลที่ต้องการแก้ไขหรือเพิ่มเติมมาทั้งกับ HTTP Body
DELETE	ใช้สำหรับลบข้อมูลที่มีอยู่ของปลายทางที่กำหนดไว้ ซึ่งจะต้องมีการระบุค่าที่อ้างอิงถึงข้อมูลที่ต้องการลบมาทั้งกับ Path Parameters หรือ Query Parameters

สำหรับรายละเอียดของ HTTP Request Method เพิ่มเติมสามารถอ่านได้จาก URL นี้ <https://www.rfc-editor.org/rfc/rfc9110.html#name-methods>

HTTP Response

คือข้อมูลที่ Server ส่งกลับไปยัง Client ตามที่ทาง Client ได้ร้องขอมา จะประกอบไปด้วย

- **Status-Line:** จะเริ่มด้วย Version ของ Protocol ตามด้วย Status-code และ Reason-Phase จบด้วย CRLF ทุกส่วนจะถูกคั่นด้วยช่องว่าง
- **Response Header:** คือรายละเอียดเพิ่มเติมของ Response นอกเหนือจากที่ระบุใน Status-line สามารถระบุหรือไม่ระบุไว้ตอน Response กลับมาได้
- **Message-body:** ข้อมูลหรือเนื้อหาของข้อมูลจริงที่ ทาง Server ส่งกลับไปยัง Client ที่จะต้องแสดงลงบน Web Browser

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HTTP Status Code

เป็นเลข จำนวนเต็ม 3 หลัก ที่จะแบ่งหมวดหมู่ตามเลขหลักหน้าสุด โดยจะมี 5 หมวด ดังนี้

ตารางที่ 3.22 แสดงหมวดหมู่ของ HTTP Status Code

หมวด	คำอธิบาย
1xx	Informational คือได้รับการร้องขอเรียบร้อยแล้ว กำลัง ำกระบวนการถัดไป
2xx	Success คือการกระทำนั้นได้รับ เข้าใจ และตอบรับเรียบร้อยแล้ว
3xx	Redirection คือต้องมีการกระทำเพิ่มเติม เพื่อให้การร้องขอนั้นสมบูรณ์
4xx	Client Error คือการร้องขอมานั้นอยู่ในรูปแบบที่ผิด หรือไม่สามารถเติมเต็มให้ได้
5xx	Server Error คือฝั่ง Server ไม่สามารถเติมเต็มการร้องขอที่ส่งเข้ามาได้

โดยตัวอย่างของ HTTP Status Code ที่มักจะพบบ่อย มีดังนี้

- หมวด 2xx Success
 - **200 OK** เป็นมาตรฐานของ HTTP Response นั้น Success ส าหรับ GET, PUT, PATCH หรือ POST
 - **201 Created** เป็น Response ส าหรับข้อมูลใหม่ได้ถูกสร้างขึ้น ใช้ส าหรับPOST หรือบาง Request ที่เป็น PUT
 - **204 No Content** เป็น Response ส าหรับ Request ที่ดำเนินการ Success แต่ไม่ได้ Return ข้อมูลกลับไป
- หมวด 3xx Redirection

○ **304 Not Modified** เป็น Status Code ที่บอกว่า Client ได้รับการ Response แล้วอยู่ในเอกสารนี้เป็นเอกสาร Cache และไม่จำเป็นต้องส่งผ่านข้อมูลเดิมอีกครั้ง ม่อนุญาตให้เข้าไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- หมวด 4xx Client Error
 - **400 Bad Request** บอกว่า Request ที่ส่งมาโดย Client นั้นไม่ถูกต้อง เาเนิหาร และ Server ไม่เข้าใจว่า Request เกี่ยวกับอะไร
 - **401 Unauthorized** บอกว่า Client ไม่ได้ร้บอนุญาตในการเข้าถึงข้อมูลและควรส่ง Credential มาพร้อม Request
 - **403 Forbidden** บ่งบอกว่า Request นั้นถูกต้องและ Client ได้รับการอนุญาตเข้าระบบ แต่ Client ไม่ได้รับการอนุญาตให้เข้าถึงข้อมูลด้วยเหตุผลบางประการ
 - **404 Not Found** บ่งบอกว่า Resource ที่ Request มานั้น ยังไม่มีอยู่ ณ ขณะนั้น หรือระบบไม่รู้จัก URL ที่ User Request เข้ามา
 - **410 Gone** บ่งบอกว่าข้อมูลที่ต้องการนั้นไม่มีอยู่แล้ว หรืออาจจะย้ายไปที่อื่น
- หมวด 5xx Server Error
 - **500 Internal Server Error** บอกว่าการ Request นั้นถูกต้อง แต่ Server ได้เจอกับสถานการณ์ที่รับมือไม่ได้
 - **503 Service Unavailable** บอกว่า Server ใช้การไม่ได้ หรือไม่วางที่จะรับและด าเนินการ Request โดยส่วนใหญ่แล้ว Server อยู่ในช่วงบ ารุงรักษา

สำหรับรายละเอียดของ HTTP Status Code เพิ่มเติมสามารถอ่านได้จาก URL นี้ <https://www.iana.org/assignments/http-status-codes/http-status-codes.xhtml>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการวิจัยและการอภิปรายผล

จากการพัฒนาระบบบริหารจัดการแค็ตตาล็อกสินทรัพย์ทางภูมิสารสนเทศโดยประยุกต์ใช้แค็ตตาล็อกสินทรัพย์ทางภูมิสารสนเทศ (Spatial Temporal Asset Catalog Implementation) ได้นำ API ที่ได้พัฒนาขึ้นไปทดสอบกับ Validator ที่ทางผู้พัฒนาข้อกำหนดได้จัดทำ ขึ้นมา และนำไปใช้งานในโครงการจริง เพื่อเป็นการรับรองว่า API ที่ได้พัฒนาขึ้นในโครงการสหกิจศึกษานี้เป็นไปตามความต้องการของข้อกำหนดและสามารถใช้งานได้จริง โดยมีรายละเอียดดังนี้

4.1 ผลการทดสอบผ่าน STAC API Validator

การทดสอบเป็นการนำ URL ของแค็ตตาล็อกสินทรัพย์ทางภูมิสารสนเทศไปทดสอบผ่าน Python Library stac-api-validator โดยมีรายละเอียดคำสั่งในการเรียกใช้งานรูปที่ 4.1

ซึ่งรายละเอียดเพิ่มเติมของ library สามารถอ่านได้จาก URL นี้ <https://github.com/stac-utils/stac-api-validator>

โดยคำสั่งที่ได้มีการระบุถึง URL ของ STAC catalog ที่ต้องการทดสอบ หมายเลขของข้อกำหนดที่ต้องการทดสอบ และ id ของ STAC Collection ที่จะใช้ในการทดสอบ และค่าของ geometry ที่จะใช้ในการทดสอบครั้งนี้

```
stac-api-validator \
  --root-url http://localhost:5001/core/api/stac/1.0/validation-stac/ \
  --conformance core \
  --conformance children \
  --conformance browseable \
  --conformance item-search \
  --conformance features \
  --conformance collections \
  --collection collection-wetland \
  --geometry '{"type": "Polygon", "coordinates": [[[-124.5849609375,32.491230287947594],
[-114.06005859375,32.491230287947594],[-114.06005859375,42.01665183556825],
[-124.5849609375,42.01665183556825],[-124.5849609375,32.491230287947594]]]}'
```

รูปที่ 4.1 รายละเอียดคำสั่งที่ใช้ในการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลที่ได้จากการทดสอบมีรายละเอียดดังรูปที่ 4.2

```
(venv) siriyas@SiriyaS-MacBook-Pro python % stac-api-validator \
--root-url http://localhost:5001/core/api/stac/1.0/validation-stac/ \
--conformance core \
--conformance children \
--conformance browseable \
--conformance item-search \
--conformance features \
--conformance collections \
--collection collection-wetland \
--geometry '{"type": "Polygon", "coordinates": [[[-124.5849609375, 32.491230287947594], [-114.06005859375, 32.491230287947594], [-114.06005859375, 42.01665183556825], [-124.5849609375, 42.01665183556825], [-124.5849609375, 32.491230287947594]]]}'
INFO:stac_api_validator.validations:Validating STAC API - Core conformance class.
INFO:stac_api_validator.validations:Validating STAC API - Browseable conformance class.
INFO:stac_api_validator.validations:Browseable validation is not yet implemented.
INFO:stac_api_validator.validations:Validating STAC API - Children conformance class.
INFO:stac_api_validator.validations:Children validation is not yet implemented.
INFO:stac_api_validator.validations:Validating STAC API - Collections conformance class.
INFO:stac_api_validator.validations:Validating STAC API - Features conformance class.
INFO:stac_api_validator.validations:Skipping STAC API - Features - Filter Extension conformance class.
INFO:stac_api_validator.validations:Validating STAC API - Item Search conformance class.
INFO:stac_api_validator.validations:Skipping STAC API - Item Search - Filter Extension conformance class.
Warnings:
- STAC API Specification v1.0.0-rc.2 is the latest version, but API advertises an older version or older versions.
- / : Link[rel=service-doc] should exist
- [Collections] GET http://localhost:5001/core/api/stac/1.0/validation-stac/collections/collection-wetland has these stac-check recommendations: ['STAC Best Practices: ', ' Object should be called 'collection.json' not 'collection-wetland.json', ', ' Links in catalogs and collections should always have a 'title' field', ', '']
- [Collections] GET http://localhost:5001/core/api/stac/1.0/validation-stac/collections/collection-wetland has these stac-check recommendations: ['STAC Best Practices: ', ' Object should be called 'collection.json' not 'collection-wetland.json', ', ' Links in catalogs and collections should always have a 'title' field', ', '']
- [Features] GET http://localhost:5001/core/api/stac/1.0/validation-stac/collections/collection-wetland/items/vallaris-2020-001-01-05 has these stac-check recommendations: ['STAC Best Practices: ', ' Item name 'vallaris-2020-001-01-05' should only contain Searchable identifiers', ' Identifiers should consist of only lowercase characters, numbers, '-', and '-', ', '']
- [Features] GET http://localhost:5001/core/api/stac/1.0/validation-stac/collections/collection-wetland/items/vallaris-2020-001-01-05 has these stac-check recommendations: ['STAC Best Practices: ', ' Item name 'vallaris-2020-001-01-05' should only contain Searchable identifiers', ' Identifiers should consist of only lowercase characters, numbers, '-', and '-', ', '']
Errors:
- Error with pystac: <urlopen error [SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: unable to get local issuer certificate (_ssl.c:997)>
(venv) siriyas@SiriyaS-MacBook-Pro python %
```

รูปที่ 4.2 ผลลัพธ์ที่ได้จากการทดสอบ

ผลลัพธ์ของการทดสอบ มีการระบุถึงหมวดหมู่ของข้อบกพร่องที่ทดสอบ Warnings และ Errors ของการทดสอบ โดยในการทดสอบครั้งนี้ รายละเอียดของ Warnings แนะนำ ถึง Best Practices ที่ควรปฏิบัติตาม และในส่วนของ Errors มีรายละเอียดเกี่ยวกับข้อผิดพลาด SSL: CERTIFICATE_VERIFY_FAILED ซึ่งอาจจะเป็นข้อผิดพลาดทางฝั่ง Network

4.2 ตัวอย่างการนำ STAC API ไปใช้ในโครงการจริง

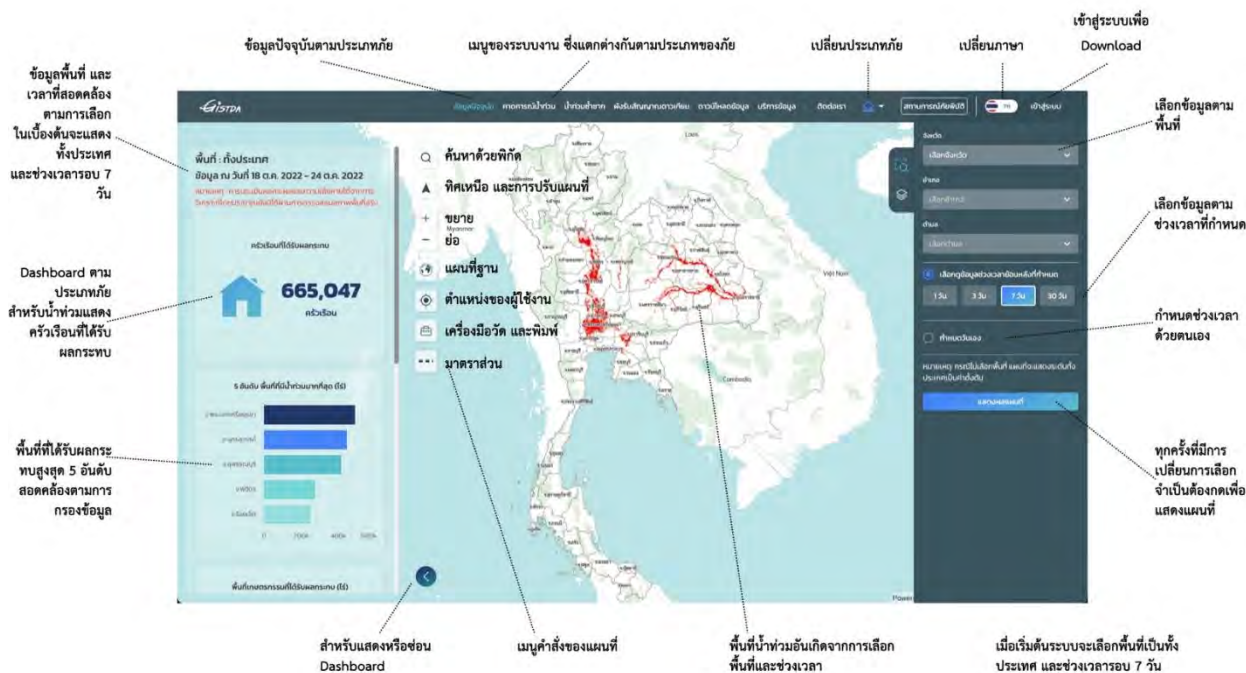
STAC API ที่ได้พัฒนาขึ้นมา นั้น ได้ไปเป็นส่วนหนึ่งของผลิตภัณฑ์ Vallaris Maps และถูกนำไปเป็นส่วนหนึ่งของโครงการที่บริษัทได้พัฒนาขึ้น โดยเริ่มตั้งแต่การนำข้อมูลเข้าให้อยู่ในรูปแบบของ แค็ตตาล็อกสินทรัพย์ทางภูมิสารสนเทศ การเรียกดูข้อมูลภายในแค็ตตาล็อก และการค้นหาข้อมูล ปัจจุบันปี พ.ศ.2565 ได้นำไปใช้ทั้งหมด 2 โครงการ และจะมีการนำไปใช้งานในโครงการของปีต่อ ๆ ไป โดยแต่ละโครงการมีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1 ระบบสนับสนุนการตัดสินใจเพื่อการบริหารจัดการพื้นที่ภัยพิบัติ ของ GISTDA

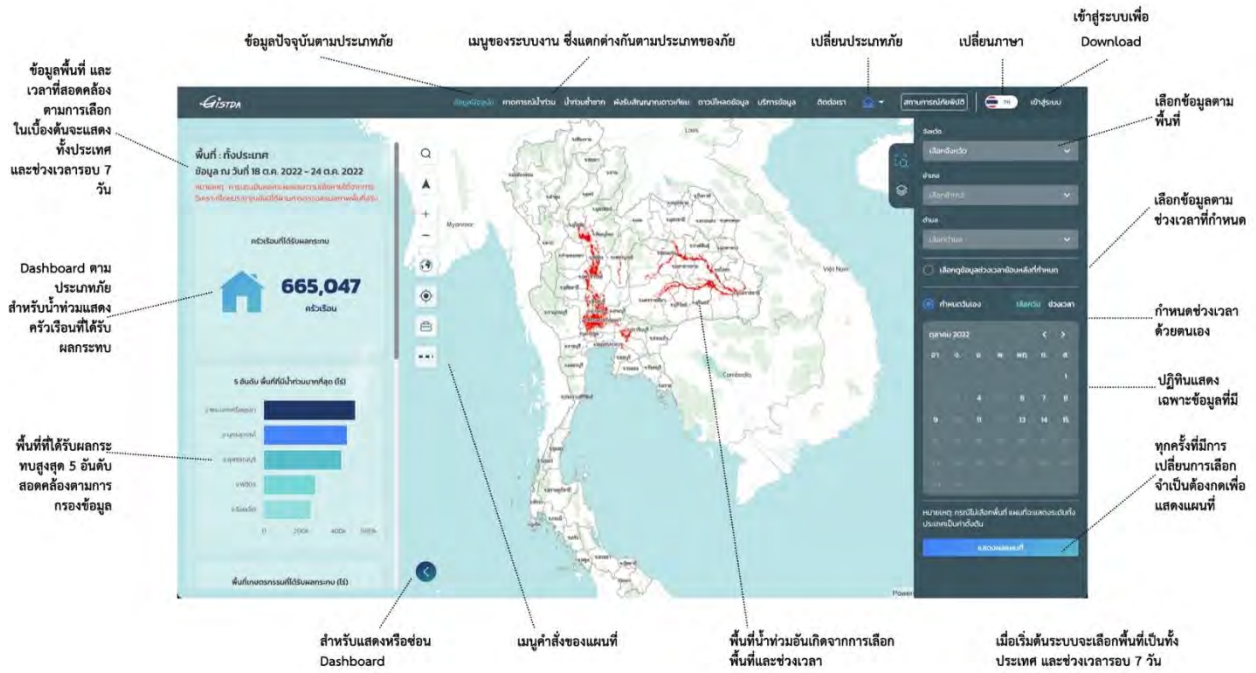
สามารถเข้าถึงได้ที่ URL: <http://disaster.gistda.or.th/>

ระบบสนับสนุนการตัดสินใจเพื่อการบริหารจัดการพื้นที่ภัยพิบัติจัดท ขึ้นเพื่อให้นักวิทยาศาสตร์ นักวิชาการ หรือผู้ที่สนใจเข้ามา สรพพื้นที่เสี่ยงภัยพิบัติทางธรรมชาติ ได้แก่ น้ท่วม ไฟป่า และภัยแล้ง โดย ภายในระบบผู้ใช้งานสามารถเลือกช่วงเวลาที่น่าสนใจ แล้วระบบจะมีการค านวนค่าทางสถิติโดยสรุปให้ พร้อมกับ แสดงบริเวณพื้นที่เสี่ยงบนแผนที่ และสามารถดาวน์โหลดข้อมูลที่ต้องการ อีกทั้งยังมี Data Service ส ำหรับให้ นักพัฒนาน ำไปเรียกใช้ต่อได้



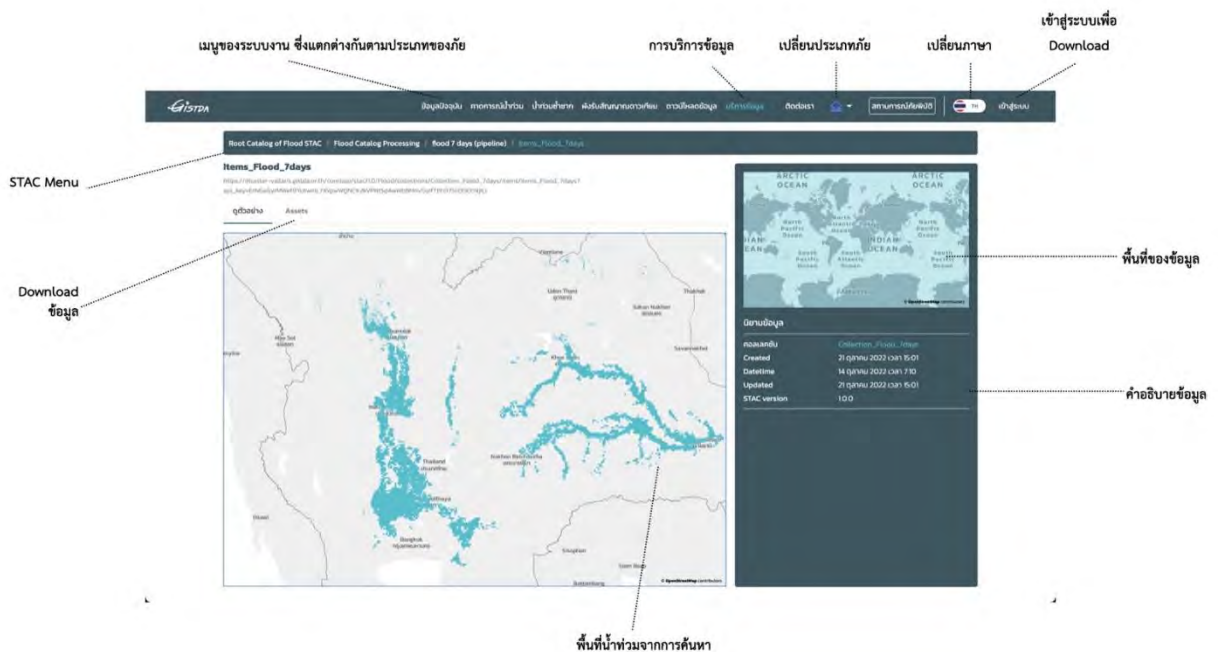
รูปที่ 4.3 รายละเอียดเมนูของหน้าเว็บข้อมูลภัยพิบัติน้ท่วมกรณีเลือกตามเวลาที่มีให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 รายละเอียดเมนูของหน้าเว็บข้อมูลภัยพิบัติน้ำท่วมกรณีกำหนดช่วงเวลาด้วยตนเอง

รูปด้านบนแสดงหน้าเว็บของภัยพิบัติน้ำท่วม ในกรณีที่ผู้ใช้งานต้องการดูข้อมูลตามช่วงเวลาที่มีไว้ให้ หรือกำหนดช่วงเวลาด้วยตนเอง ระบบจะไปค้นหาข้อมูลนั้นผ่าน STAC API ที่พัฒนาขึ้น เมื่อมีข้อมูลจึงจะนำข้อมูลนั้นออกมาแสดงบนแผนที่ ทำให้ระบบฝั่งหน้าเว็บไม่จำเป็นต้องติดต่อกับฐานข้อมูลโดยตรง ทำให้การทำงานของหน้าเว็บนั้นเร็วขึ้น



รูปที่ 4.5 รายละเอียดเมนูของหน้าเว็บบริการข้อมูลแค็ตตาล็อกข้อมูล (STAC)

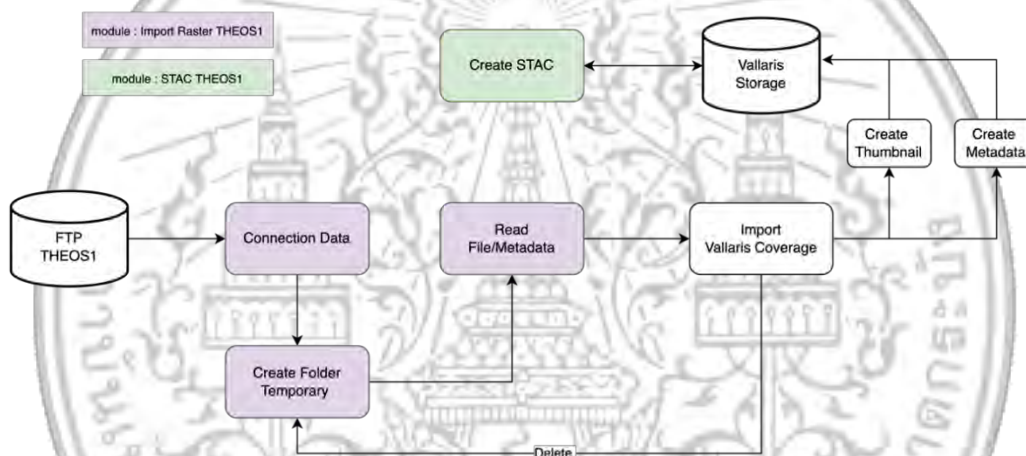
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของการเรียกดูข้อมูลในแค็ตตาล็อกข้อมูล STAC ได้เป็นการเรียกข้อมูลออกมาแสดงผ่าน STAC API โดยให้ความรู้สึกกับผู้ใช้งานเหมือนการเปิดไฟล์เดอร์เข้าไปเรื่อย ๆ จนเจอไฟล์ข้อมูล ซึ่งเมื่อคลิกดู STAC Object แต่ละตัว ก็จะคุณสมบัติของ Object นั้นระบุไว้ด้วย และเมื่อเข้าไปถึงหน้าของข้อมูล ก็จะมีเมนู Assets ในการที่มี URL ให้ผู้ใช้งานสามารถดาวน์โหลดไปใช้งานต่อได้

4.2.2 ระบบ Platform บูรณาการข้อมูลสำหรับ Smart City ของ GISTDA

4.2.2.1 การพัฒนา Module ในการนำเข้าข้อมูลภาพถ่ายดาวเทียม THEOS 1

Module ที่พัฒนาขึ้นมีชื่อว่า “Import Raster THEOS1” สามารถอ่านไฟล์ภาพถ่ายของดาวเทียม THEOS 1 ได้ทั้งแบบภาพ Multi Spectral (1A, 2A) และ Panchromatic (1A, 2A) ตัว Module จะมีการอ่านข้อมูลไฟล์และ Metadata ของไฟล์ เพื่อนำไปใช้ในกระบวนการนำเข้าข้อมูลใน Vallaris Maps และนำข้อมูล Metadata ใน STAC



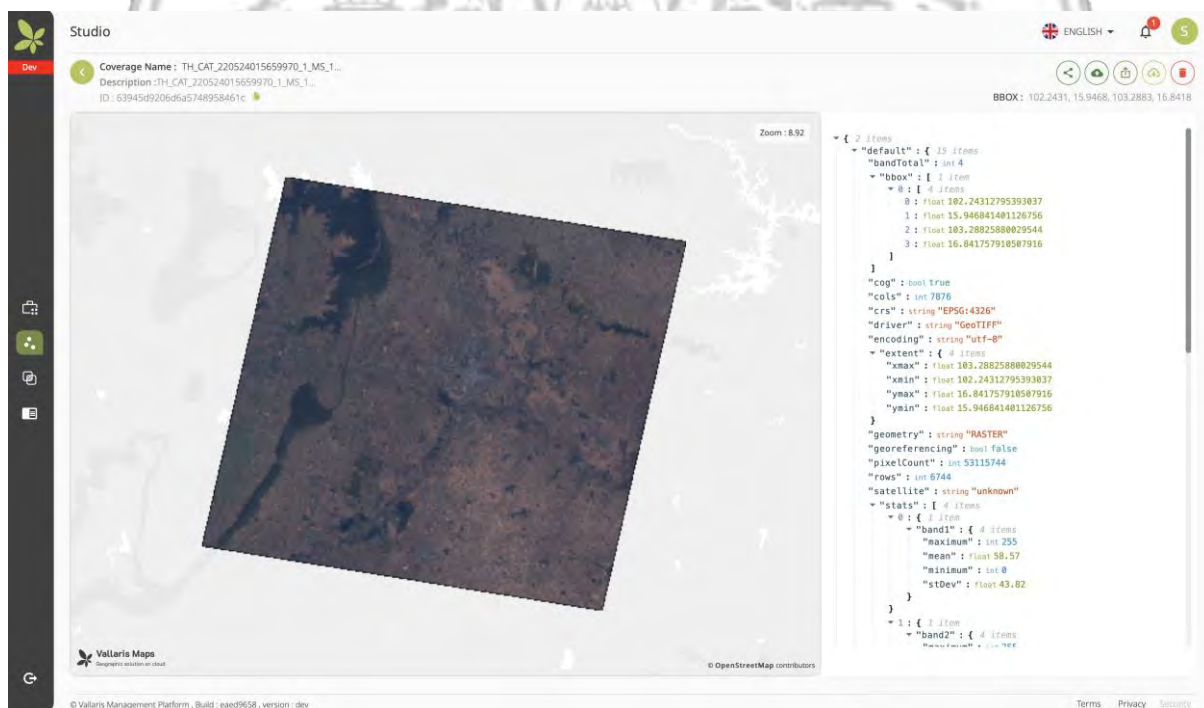
รูปที่ 4.6 Workflow การนำเข้าข้อมูลภาพถ่ายดาวเทียม THEOS 1

จากแผนภาพด้านบนเมื่อมีการนำเข้าข้อมูลภาพถ่ายดาวเทียม และเก็บลงแหล่งข้อมูล ก็จะมีการเรียกใช้ Module เพื่อใช้ในการสร้างแค็ตตาล็อกของข้อมูลในรูปแบบ STAC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 Pipeline ดาวเทียม THEOS 1



รูปที่ 4.8 ตัวอย่างภาพถ่ายเทียม THEOS 1 บริเวณจังหวัดขอนแก่น

ภาพถ่ายวันที่ 14 เดือน ธันวาคม ปี ค.ศ. 2021

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2.2 การจัดการข้อมูลในรูปแบบมาตรฐาน STAC

ในการสร้างแค็ตตาล็อกของข้อมูลภาพถ่ายดาวเทียมให้อยู่ในรูปแบบของ STAC ได้มีการออกแบบโครงสร้างของ STAC สำหรับภาพถ่ายดาวเทียม บริษัทได้ออกแบบ จา แนวตามคุณลักษณะของแต่ละดาวเทียมเพื่อให้เป็นประโยชน์ต่อการเข้าถึง และการค้นหาภาพถ่ายดาวเทียมได้อย่างมีประสิทธิภาพที่สุด ซึ่งการค้นหาข้อมูลภาพถ่ายดาวเทียมระบบจะหา การค้นหาจาก โครงสร้าง STAC ที่ก หนดไว้โดยท า การค้นหาที่ค าอธิบายของแต่ละภาพถ่ายใต้งี้อื่นไขที่บรรจุอยู่ในคุณสมบัติ (Property) โดยมีรายละเอียดโครงสร้าง STAC ของ THEOS 1 ดังนี้

Dimension: THEOS1 → Imaging mode → Year → Month

STAC Properties:

```
{
  "datetime": "2022-01-09T03:18:27.648759Z",
  "grid_reference": "0265-0318",
  "imaging_date": "2022-01-09",
  "imaging_mode": "MS",
  "imaging_time": "03:18:27.648759",
  "instrument": "TOP",
  "instrument_index": "2",
  "mission": "THEOS",
  "mission_index": "1",
  "revolution_number": 69226,
  "satellite_azimuth_angle": 100.126587,
  "satellite_incidence_angle": 2.159915,
  "scene_processing_level": 0,
  "shift_value": 0,
  "sun_azimuth": 142.080199,
  "sun_elevation": 42.053161,
  "viewing_angle_across_track": -1.909339,
  "viewing_angle_along_track": 0.090809
}
```

รูปที่ 4.9 STAC Properties ของดาวเทียม THEOS 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

STAC Catalog ของดาวเทียม THEOS 1

```
{
  "id": "THEOS1",
  "type": "Catalog",
  "stac_version": "1.0.0",
  "title": "THEOS 1",
  "description": "This is root catalog for THEOS1",
  "links": [
    {
      "href": "https://v2k-dev.vallarismaps.com/core/api/stac/1.0
/THEOS1/",
      "rel": "self",
      "type": "application/json"
    },
    {
      "href": "https://v2k-dev.vallarismaps.com/core/api/stac/1.0
/THEOS1/",
      "rel": "root",
      "type": "application/json"
    },
    {
      "href": "https://v2k-dev.vallarismaps.com/core/api/stac/1.0
/THEOS1/conformance",
      "rel": "conformance",
      "type": "application/json",
      "title": "OGC API conformance classes implemented by this
server"
    },
    {
      "href": "https://v2k-dev.vallarismaps.com/core/api/stac/1.0
/THEOS1/api",
      "rel": "service-desc",
      "type": "application/vnd.oai.openapi+json;version=3.0",
      "title": "the API definition"
    },
    {
      "href": "https://v2k-dev.vallarismaps.com/core/api/stac/1.0
/THEOS1/queryables",
      "rel": "queryables",
      "type": "application/schema+json",
      "title": "queryables of this catalog"
    }
  ],
}
```

```

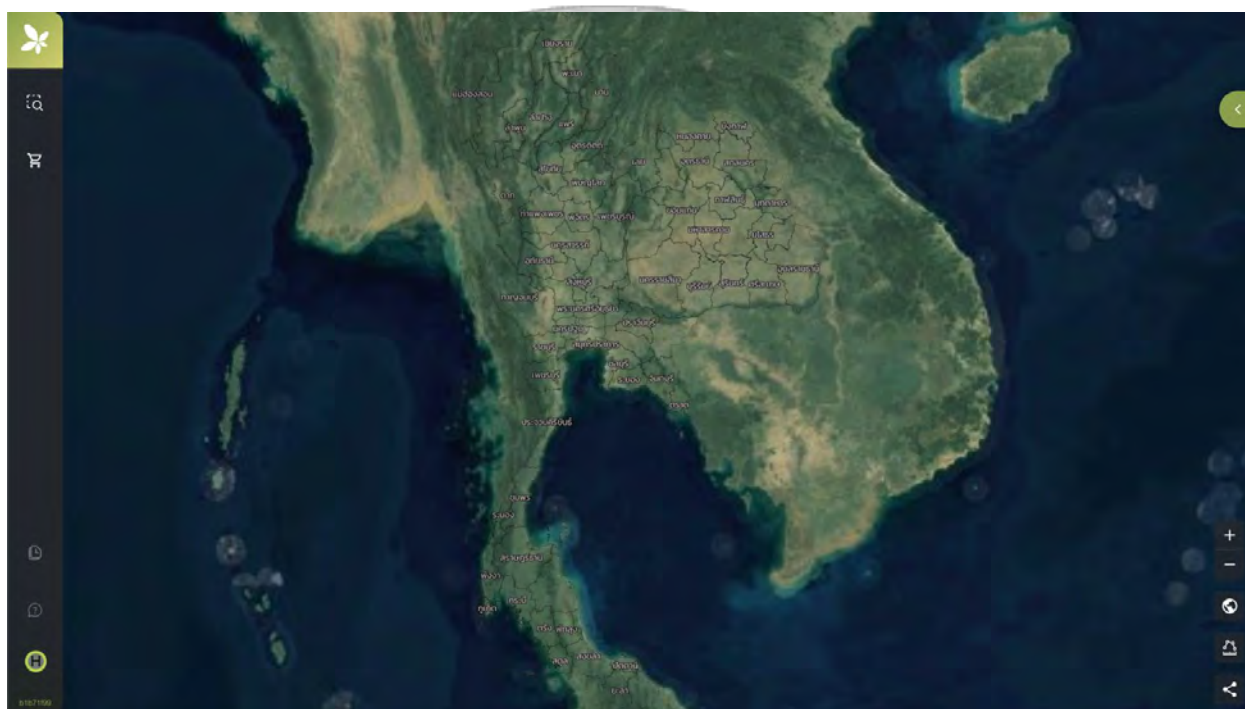
    {
      "href": "https://v2k-dev.vallarismaps.com/core/api/stac/1.0
/THEOS1/collections",
      "rel": "data",
      "type": "application/json",
      "title": "collection list"
    },
    {
      "href": "https://v2k-dev.vallarismaps.com/core/api/stac/1.0
/THEOS1/search",
      "rel": "search",
      "type": "application/geo+json",
      "title": "item search"
    },
    {
      "href": "https://v2k-dev.vallarismaps.com/core/api/stac/1.0
/THEOS1/children",
      "rel": "children",
      "type": "application/json",
      "title": "child list"
    },
    {
      "href": "https://v2k-dev.vallarismaps.com/core/api/stac/1.0
/THEOS1/catalogs/THEOS1_MS",
      "rel": "child",
      "type": "application/json",
      "title": "THEOS1 Catalog imaging mode Multi Spectral (MS)"
    },
    {
      "href": "https://v2k-dev.vallarismaps.com/core/api/stac/1.0
/THEOS1/catalogs/THEOS1_PAN",
      "rel": "child",
      "type": "application/json",
      "title": "THEOS1 Catalog imaging mode Panchromatic (PAN)"
    }
  ],
  "conformsTo": [
    "https://api.stacspec.org/v1.0.0-rc.1/core",
    "https://api.stacspec.org/v1.0.0-rc.1/children",
    "https://api.stacspec.org/v1.0.0-rc.1/browseable",
    "https://api.stacspec.org/v1.0.0-rc.1/collections",
    "https://api.stacspec.org/v1.0.0-rc.1/ogcapi-features",
    "https://api.stacspec.org/v1.0.0-rc.1/item-search",
    "http://www.opengis.net/spec/ogcapi-features-1/1.0/conf/core",
    "http://www.opengis.net/spec/ogcapi-features-1/1.0/conf/oas30",
    "http://www.opengis.net/spec/ogcapi-features-1/1.0/conf/geojson"
  ]
}
```

รูปที่ 4.10 STAC Catalog ของดาวเทียม THEOS 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

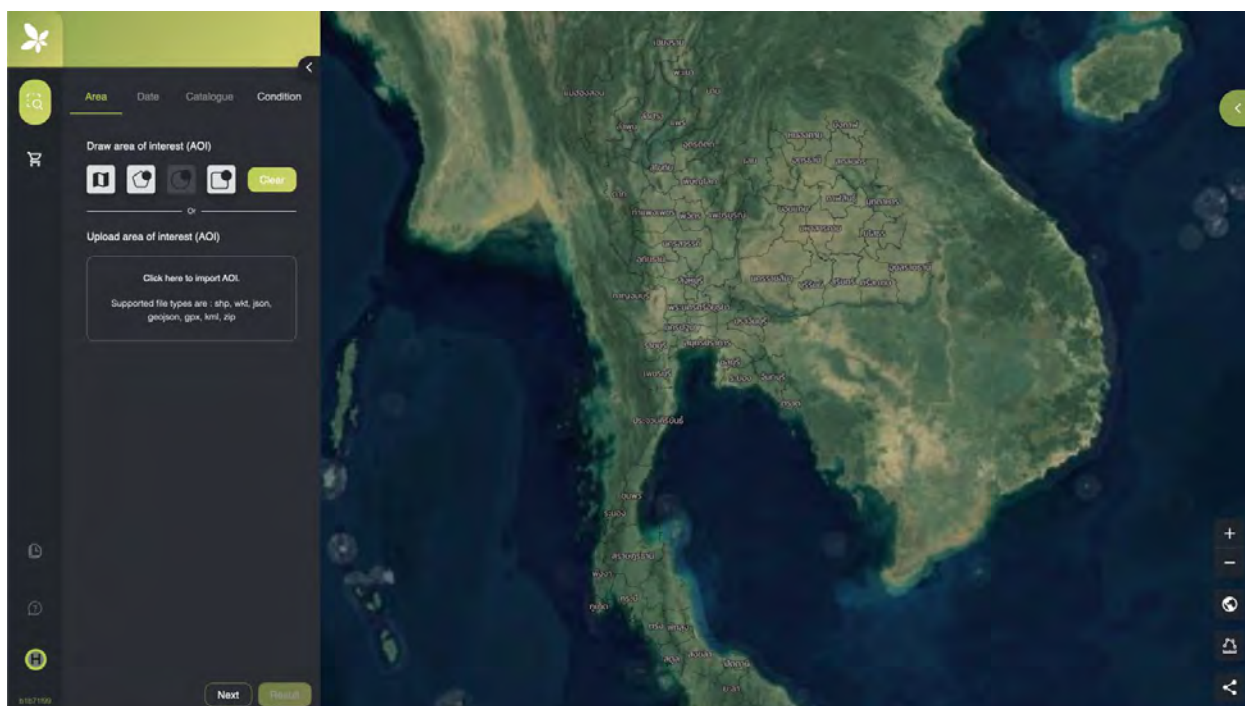
4.2.2.3 ระบบการค้นหาข้อมูลภาพถ่ายดาวเทียม

ระบบการค้นหาข้อมูลภาพถ่ายดาวเทียมถูกสร้างขึ้นเพื่อให้ใช้งานบนเว็บไซต์ เป็นหลักเนื่องจากมีระบบการดาวน์โหลดข้อมูลขนาดใหญ่ที่เหมาะสมกับการท างานบนเครื่องคอมพิวเตอร์ ในการค้นหาข้อมูลภาพถ่ายดาวเทียมจะเข้าถึงผ่าน STAC API ที่มีรูปแบบการค้นหาข้อมูลไปที่แต่ละรายการของภาพถ่ายดาวเทียม ทำให้การค้นหาสามารถทราบถึงพื้นที่ และช่วงเวลาของภาพ ได้อย่างรวดเร็ว ประกอบกับคุณสมบัติของ STAC ที่สามารถกำหนดคุณสมบัติของแต่ละข้อมูลว่าจะอนุญาตให้ค้นหาด้วยเงื่อนไขอะไรบ้างผ่าน Query Parameter

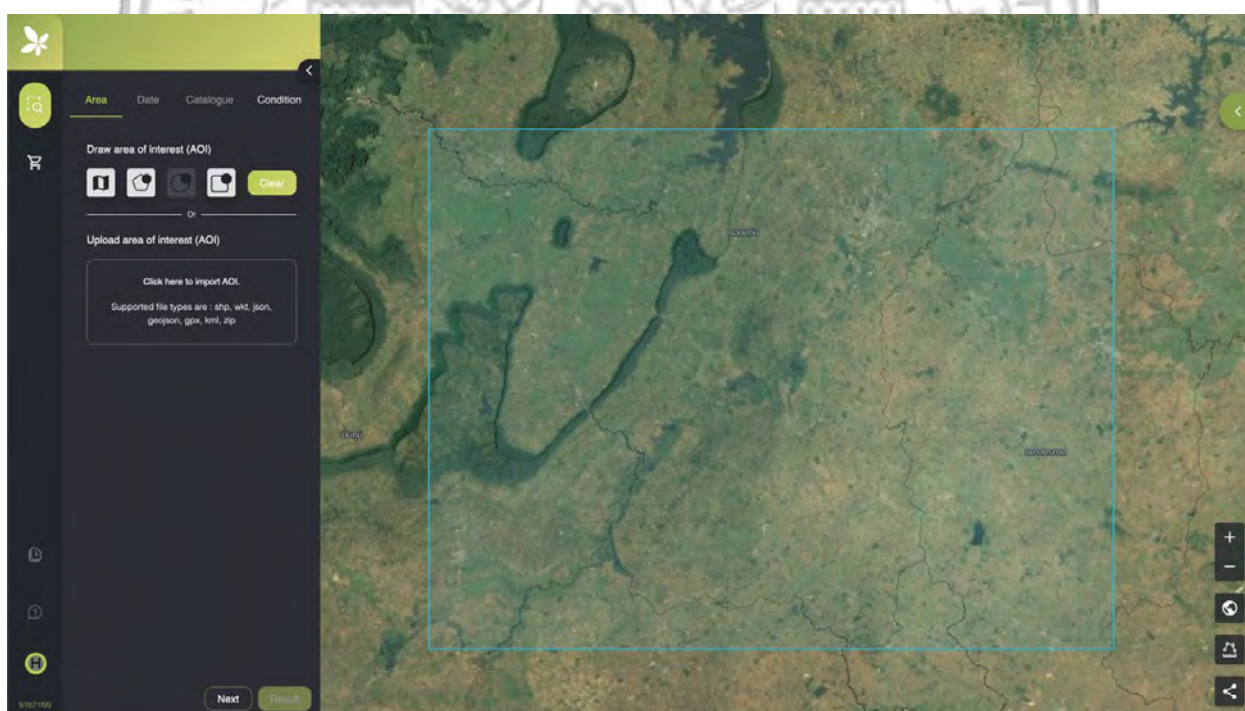


รูปที่ 4.12 หน้าจอแรกเมื่อผู้ใช้งานเข้ามาสู่ระบบการค้นหาภาพถ่ายดาวเทียม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

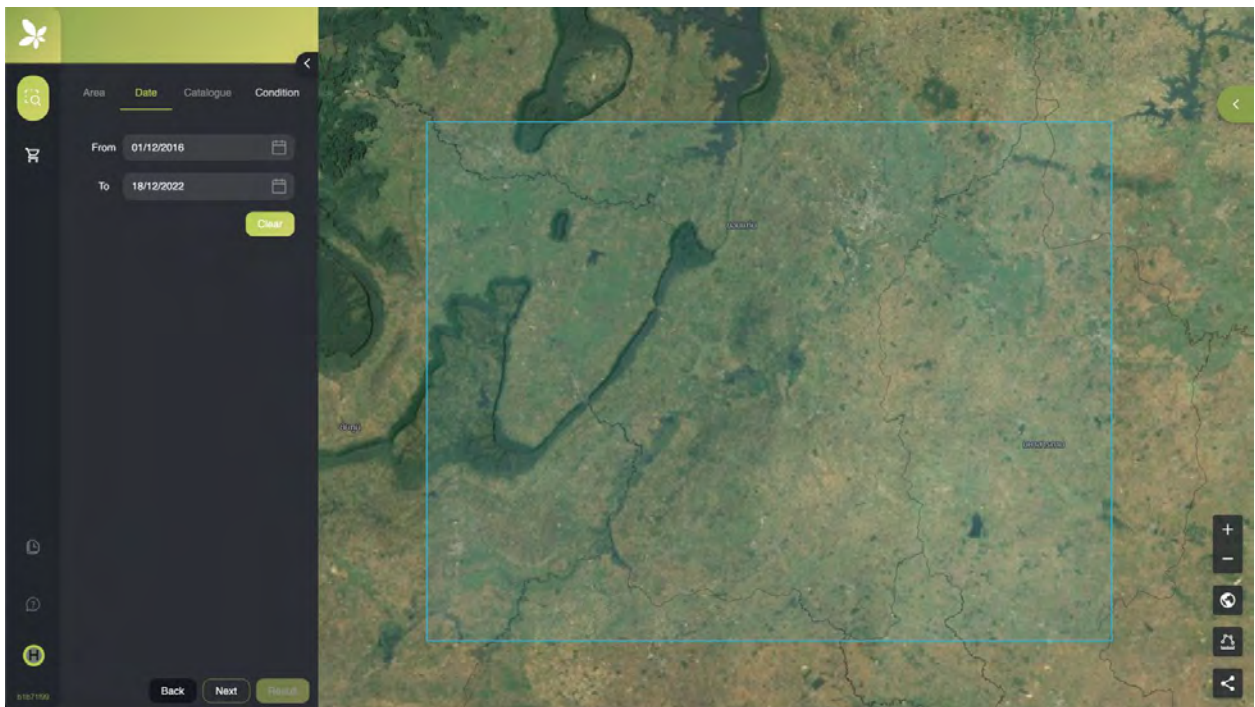


รูปที่ 4.13 เริ่มต้นด้วยการให้ผู้ใช้งานเลือกพื้นที่ที่ต้องการผ่านการสร้าง หรือ ส่งไฟล์ขึ้นมา

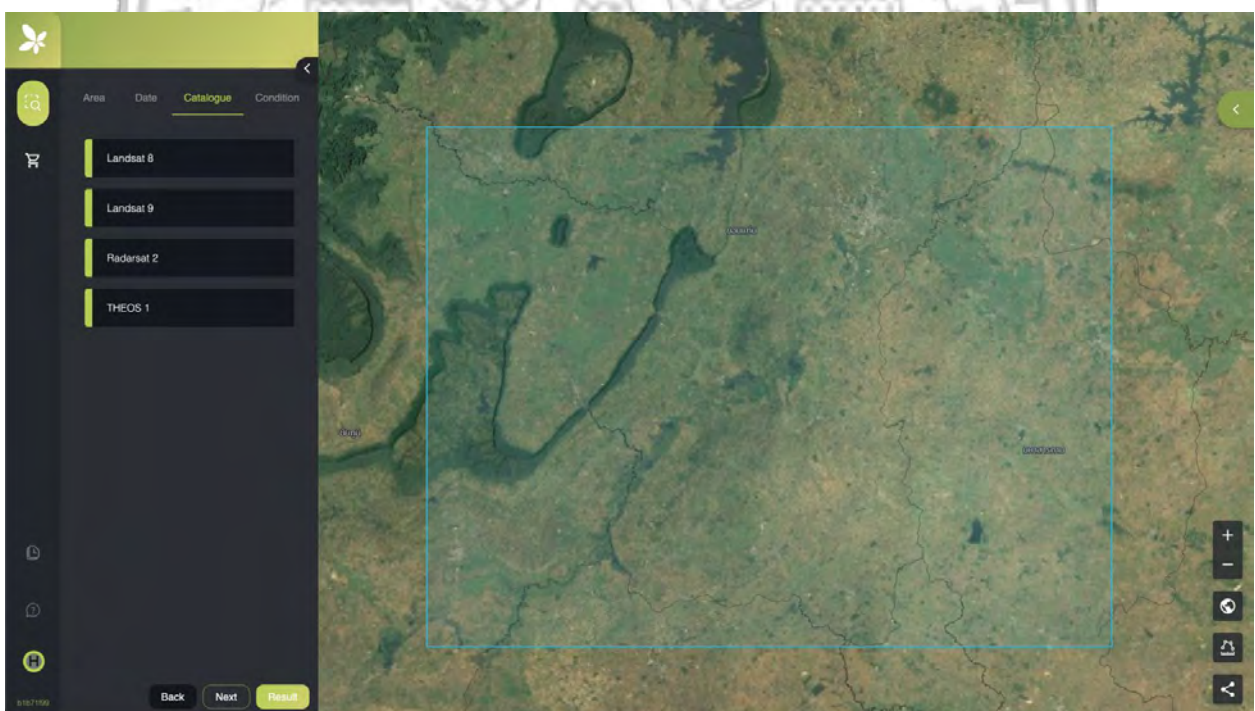


รูปที่ 4.14 เมื่อสร้างพื้นที่ที่ต้องการระบบจะแสดงพื้นที่ให้ผู้ใช้งานเห็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

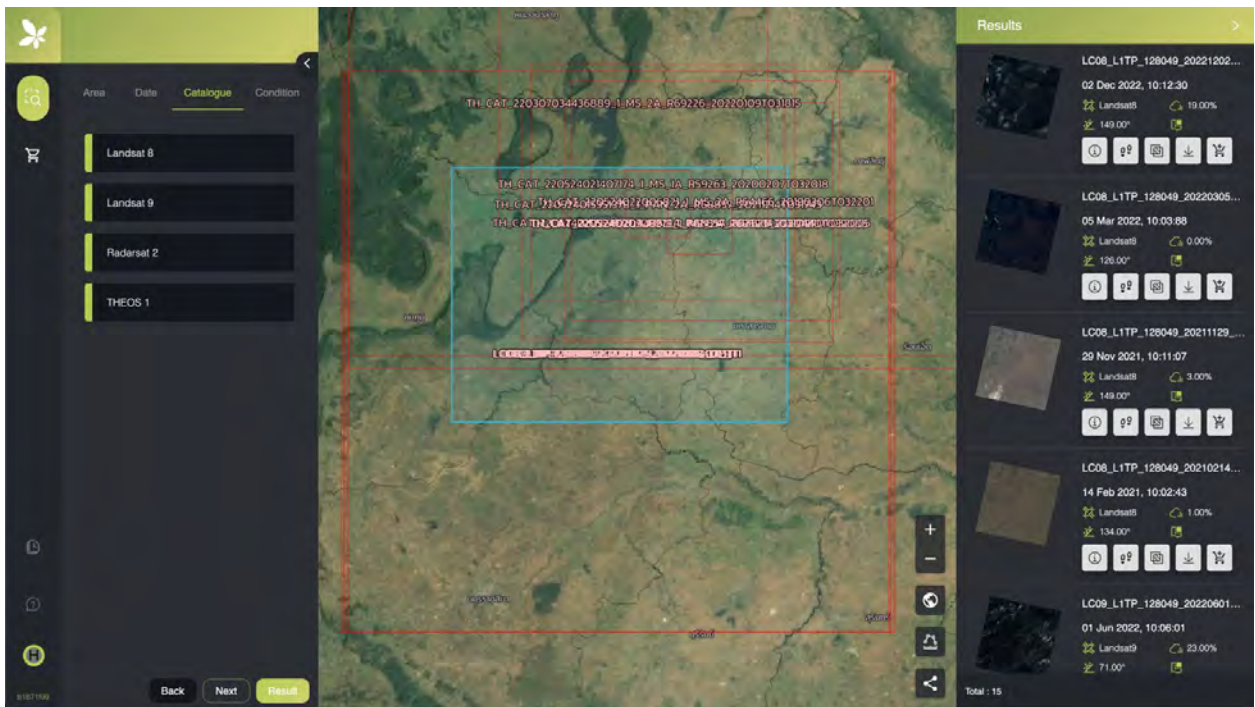


รูปที่ 4.15 ผู้ใช้งานสามารถเลือกช่วงเวลาที่ต้องการ สามารถกำหนดเป็นช่วงเริ่มต้นและสิ้นสุด

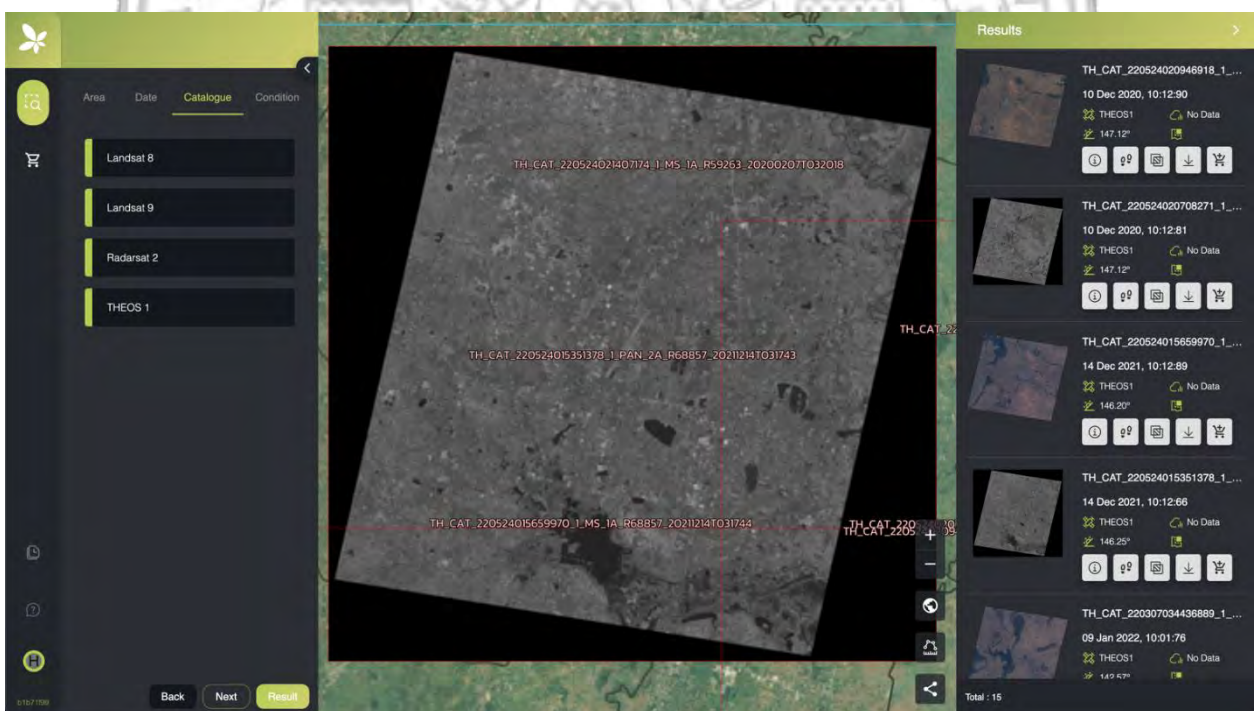


รูปที่ 4.16 กำหนดประเภทของดาวเทียมที่ต้องการค้นหา แถบสีเขียวหมายถึง ต้องการค้นหาดาวเทียมนั้น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

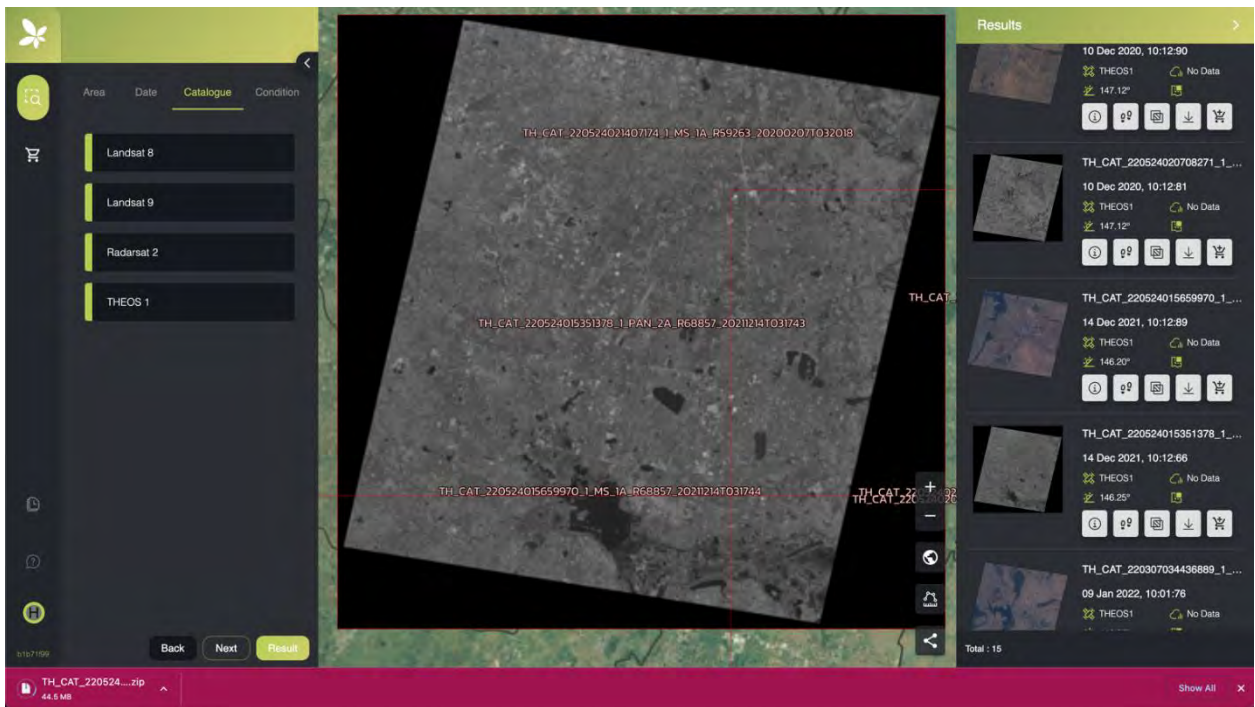


รูปที่ 4.17 เมื่อเลือกและค้นหา ผลลัพธ์จะนำมาแสดงทางด้านขวา

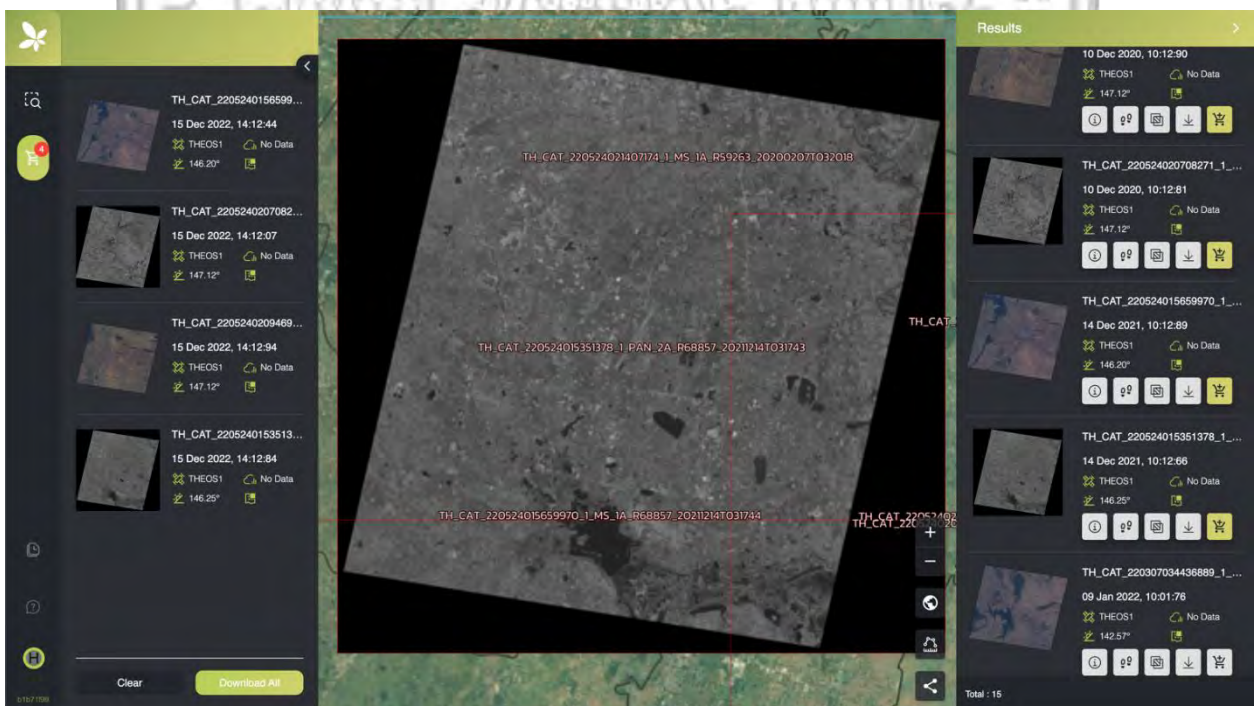


รูปที่ 4.18 เลือกดูภาพที่ต้องการในรูปแบบของ Thumbnail พร้อมคุณสมบัติต่าง ๆ ของภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.19 สามารถเลือก Download ข้อมูลภาพได้ทันที
(มีการกำหนดสิทธิ์การเข้าถึงภาพผ่านระบบบริหารจัดการ)



รูปที่ 4.20 เมื่อเลือกหลายภาพ สามารถไปจัดเก็บในระบบตะกร้าเพื่อ Download แบบหลายภาพ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่นอญูดหมายไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

5.1.1 ได้ศึกษา และท ความเข้าใจเกี่ยวกับการท า แค็ตตาล็อกของข้อมูลทางภูมิสารสนเทศ ตามข้อกำหนด SpatioTemporal Asset Catalog Specification

5.1.2 ได้พัฒนาระบบในการบริหารจัดการ Spatial Temporal Asset Catalog และการค้นหาสินทรัพย์ที่ต้องการ

5.1.3 ระบบ STAC API ในการบริหารจัดการ Spatial Temporal Asset Catalog ที่ได้พัฒนาขึ้นสามารถน าสินทรัพย์ทางภูมิสารสนเทศมาสร้างเป็นแค็ตตาล็อกของข้อมูลได้ และสามารถเรียกดูข้อมูลในรูปแบบตามข้อกำหนดระบุไว้

5.1.4 ระบบ STAC API ในการค้นหาข้อมูลที่ต้องการที่ได้พัฒนาขึ้นสามารถค้นหาข้อมูลที่อ้างอิงไปยังสินทรัพย์ภูมิสารสนเทศ ทาให้อานขความสะดวกในการค้นหาข้อมูลในพื้นที่ และช่วงเวลาที่น่าสนใจได้

5.2 ข้อเสนอแนะ

5.2.1 สามารถน าไปต่อยอดในงานอื่น ๆ ที่มีการท าแค็ตตาล็อกของข้อมูลทางภูมิสารสนเทศ หรืองานที่แค็ตตาล็อกของข้อมูลทางภูมิสารสนเทศเข้าไปช่วยในการเข้าถึง หรือค้นหาข้อมูลได้สะดวกสบายมากยิ่งขึ้น

5.2.2 เนื่องจากการน าข้อมูลเข้าให้อยู่ในรูปแบบของแค็ตตาล็อกสินทรัพย์ทางภูมิสารสนเทศสามารถเป็นไปได้หลายแบบ จึงควรเพิ่มเติมส่วนที่จะให้ผู้ใช้งานระบุโครงสร้างที่ต้องการ แล้วระบบท ากาน น าข้อมูลเข้าให้อัตโนมัติ

5.2.3 ในส่วนของการค้นหาข้อมูล ควรเก็บรายละเอียดเพิ่มให้รองรับการค้นหาด้วยเงื่อนไขที่ซับซ้อนได้

5.2.4 น าระบบไปทดสอบกับทางมาตรฐานให้ได้รับการรับรอง เพื่อเพิ่มมูลค่าของระบบ และผลิตภัณฑ์อื่นที่น ไปต่อยอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- สยามรัฐออนไลน์. 2565. เปิดตัว "Vallaris Maps Platform" เครื่องมือจัดการข้อมูลด้านภูมิศาสตร์ ตอบ
โจทย์ขับเคลื่อนองค์กรด้วยนวัตกรรม. [Online]. เข้าถึงได้จาก: <https://siamrath.co.th/n/342703>.
- Arab, S., Bianglae, A. and Udombua P. 2020. OGC API - FEATURES PART 1 – CORE. [PDF
File]. Available from: OSGeo Thailand -OGC API Features Part1.pdf.
- Butler, H., Daly, M., Doyle, A., Gillies, S., Hagen, S. and Schaub, T. 2016. The GeoJSON
Format. [Online]. Available: <https://www.rfc-editor.org/rfc/rfc7946>.
- Chaiyo Hosting. 2022. MongoDB Hosting Cloud MongoDB - NoSQL Database เจ้าแรกในไทย.
[Online]. Available: [https://www.chaiyohosting.com/mongodb-hosting.html#:~:text=MongoDB เป็นฐานข้อมูล หรือ,การเลือกใช้งาน MongoDB](https://www.chaiyohosting.com/mongodb-hosting.html#:~:text=MongoDB%20เป็นฐานข้อมูล%20หรือ,การเลือกใช้งาน%20MongoDB).
- Kamol, T. 2019. http คืออะไร?. [Online]. Available: <https://medium.com/icreativesystems/basic-http-3a2b05e5aa19>.
- Montha, S. 2019. REST กับ RESTful API ต่างกันนะรู้ยัง. [Online]. Available: <https://iamgique.medium.com/restful-api-กับ-rest-api-ต่างกันนะรู้ยัง-2c70c42990e3>.
- Open Geospatial Consortium. 2019. OGC API - Features - Part 1: Core. [Online]. Available:
<http://docs.opengeospatial.org/is/17-069r3/17-069r3.html>.
- Radiant Earth Foundation. 2022. SpatioTemporal Asset Catalog API specification - an API to
make geospatial assets openly searchable and crawlable. [Online]. Available:
<https://github.com/radiantearth/stac-api-spec>.
- Radiant Earth Foundation. 2022. SpatioTemporal Asset Catalog specification – making
geospatial assets openly searchable and crawlable. [Online]. Available:
<https://github.com/radiantearth/stac-spec>.
- Skooldio. 2021. Golang คืออะไร? ดียังไง? ท ไม่นคนใช้เยอะ? รวมสิ่งที่ควรรู้เกี่ยวกับ Golang. [Online].
Available: <https://blog.skooldio.com/what-is-golang/>.
- Thongjor, N. 2017. เรียนรู้การใช้ภาษา Go ใน 15 นาที. [Online]. Available:
<https://www.babelcoder.com/blog/articles/intro-to-golang>.
- Tutorialspoint. 2022. MongoDB – Overview. [Online]. Available:
https://www.tutorialspoint.com/mongodb/mongodb_overview.htm.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้