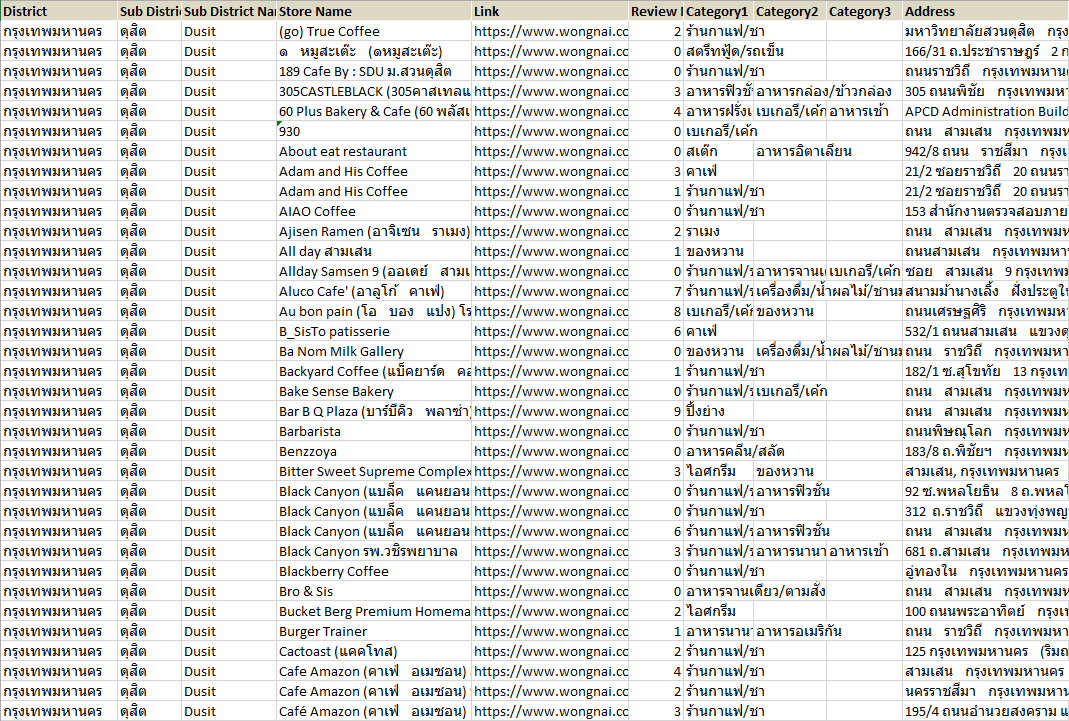
# ทำไมเราจึงต้องใช้โปรแกรม Web Scraper

ทางเหล่าคนเช็ค POI ที่ขยันขันแข็งได้ประสบปัญหาเป็นอย่างยิ่งในการตรวจสอบข้อมูลของสถานที่ เนื่องจากสาเหตุใหญ่ ๆ ดังนี้

* ข้อมูลมีจำนวนมาก จึงใช้เวลานานในการเช็ค
* บางครั้งคนเช็คอ่านหรือบันทึกข้อมูลผิดพลาด ส่งผลให้ข้อมูลคลาดเคลื่อน
* ทำนาน ๆ ก็จะเบื่อ เมื่อยล้าสายตา เสียสุขภาพจิต ชีวิตไม่สดใส

ซึ่งโปรแกรม Web Scraper สามารถดึงข้อมูลทั้งหมดที่เราต้องการจากเว็บไซต์ได้ในเวลาอันรวดเร็วและเก็บเป็น Excel อย่างมีระเบียบเพื่อให้คนเช็ค POI นำไปใช้งานต่อได้ง่าย โดยไม่กังวลว่าข้อมูลที่บันทึกมานั้นมีความผิดพลาดใด ๆ (คอมพิวเตอร์นั้นมีดี)



อีกทั้งเราสามารถใช้โปรแกรมเพื่อดึงข้อมูลชุดเดิมจากเว็บไซต์ได้ตลอดเวลา อีกทั้งความแม่นยำและความเร็วของโปรแกรมช่วยให้เราสามารถหาความแตกต่างระหว่างข้อมูลเก่ากับใหม่ได้ง่ายขึ้นมาก ด้วยการใช้ฟังก์ชันใน Excel เพื่อหาผลลัพธ์ความแตกต่าง และหา POI ใหม่หรือ POI ที่ถูกเปลี่ยนแปลง

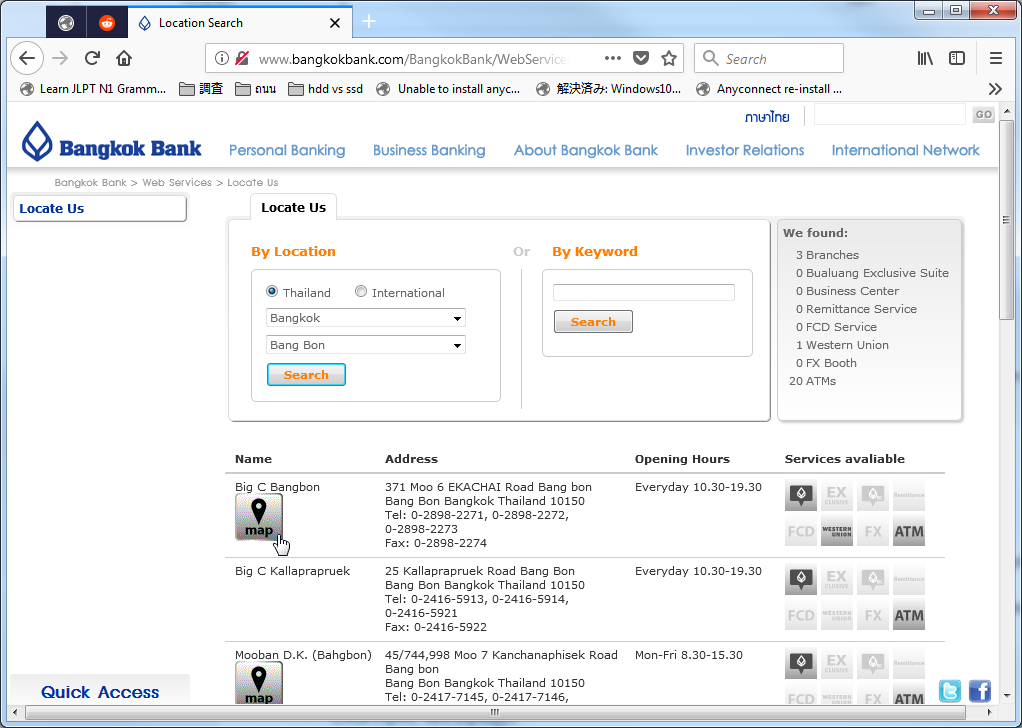
จากเหตุผลดังกล่าว ทางผู้มีอำนาจจึงตัดสินใจซื้อโปรแกรม Web Scraper เพื่อให้ชาวคนเช็ค POI มีความสุขและสนุกในการทำงานมากขึ้น (คาดว่า)

# เป้าหมายของโปรแกรม

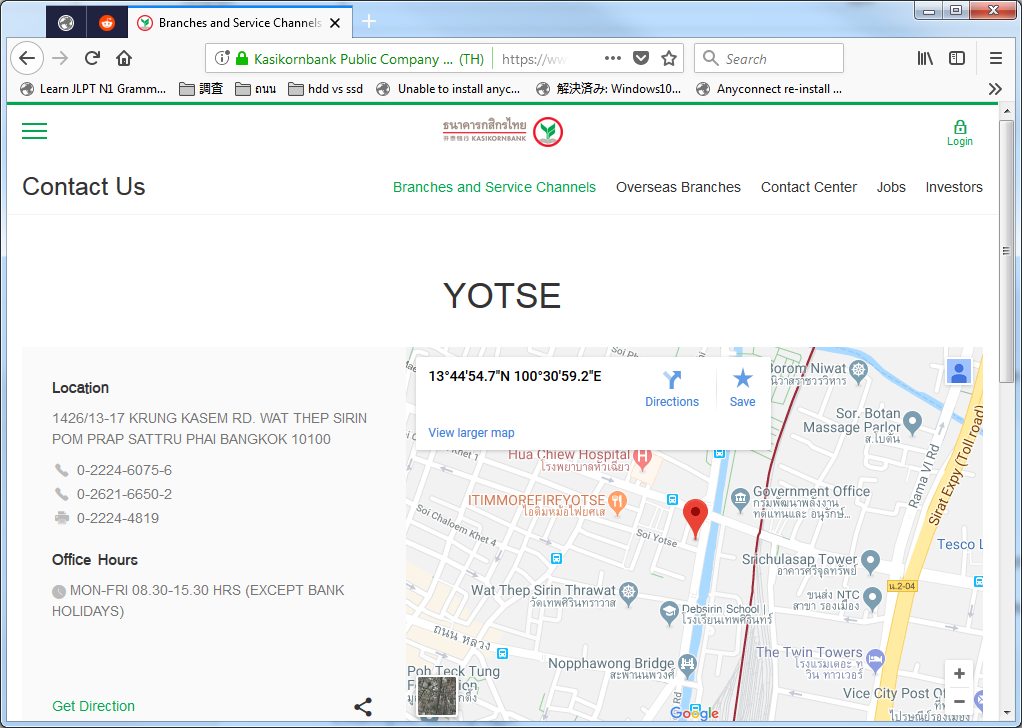
* ดึงข้อมูลจากเว็บที่มีร้านสาขาเยอะ หรือรวบรวมรายชื่อสถานที่ อาทิเช่น ธนาคาร ปั้มน้ำมัน หรือ wongnai อาบอบนวด เป็นต้น
* ตัวอย่างข้อมูลที่สำคัญ
  + พิกัดของ POI
  + ชื่อของ POI
  + ที่อยู่ของ POI
  + เบอร์ติดต่อ อีเมล์ เว็บไซต์
* บันทึกการทำงานของโปรแกรมที่ดึงข้อมูลจากเว็บไซต์อย่างเป็นขั้นตอนและนำไปใช้งานในภายหลังได้
* นำผลลัพธ์ที่บันทึกบน Excel ไปใช้ในการตรวจสอบได้ อีกทั้งสามารถใช้ในการวิเคราะห์ข้อมูลได้ง่าย

# ตัวอย่างเว็บไซต์

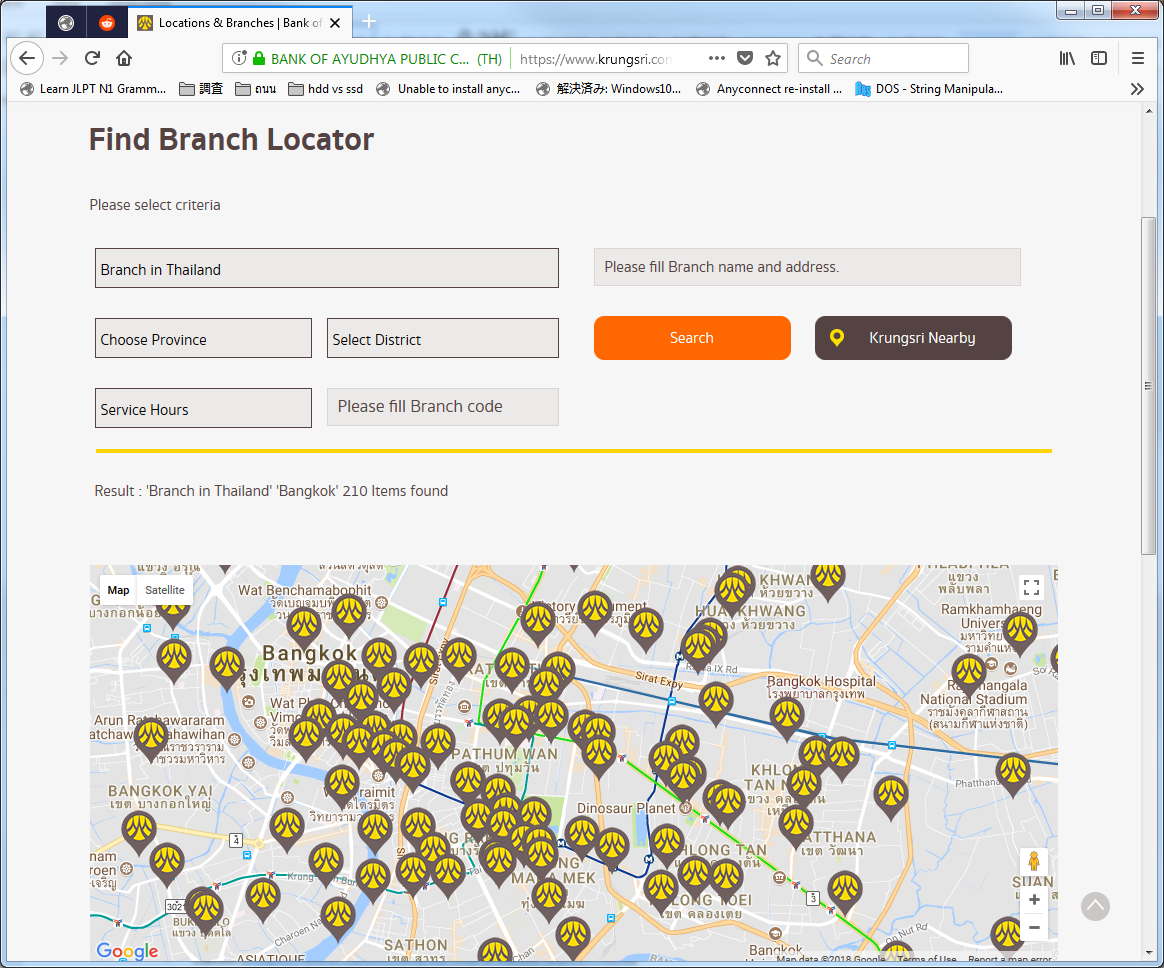
* ธนาคารกรุงเทพ (<http://www.bangkokbank.com/BangkokBank/WebServices/LocateUs/Pages/Default.aspx>)



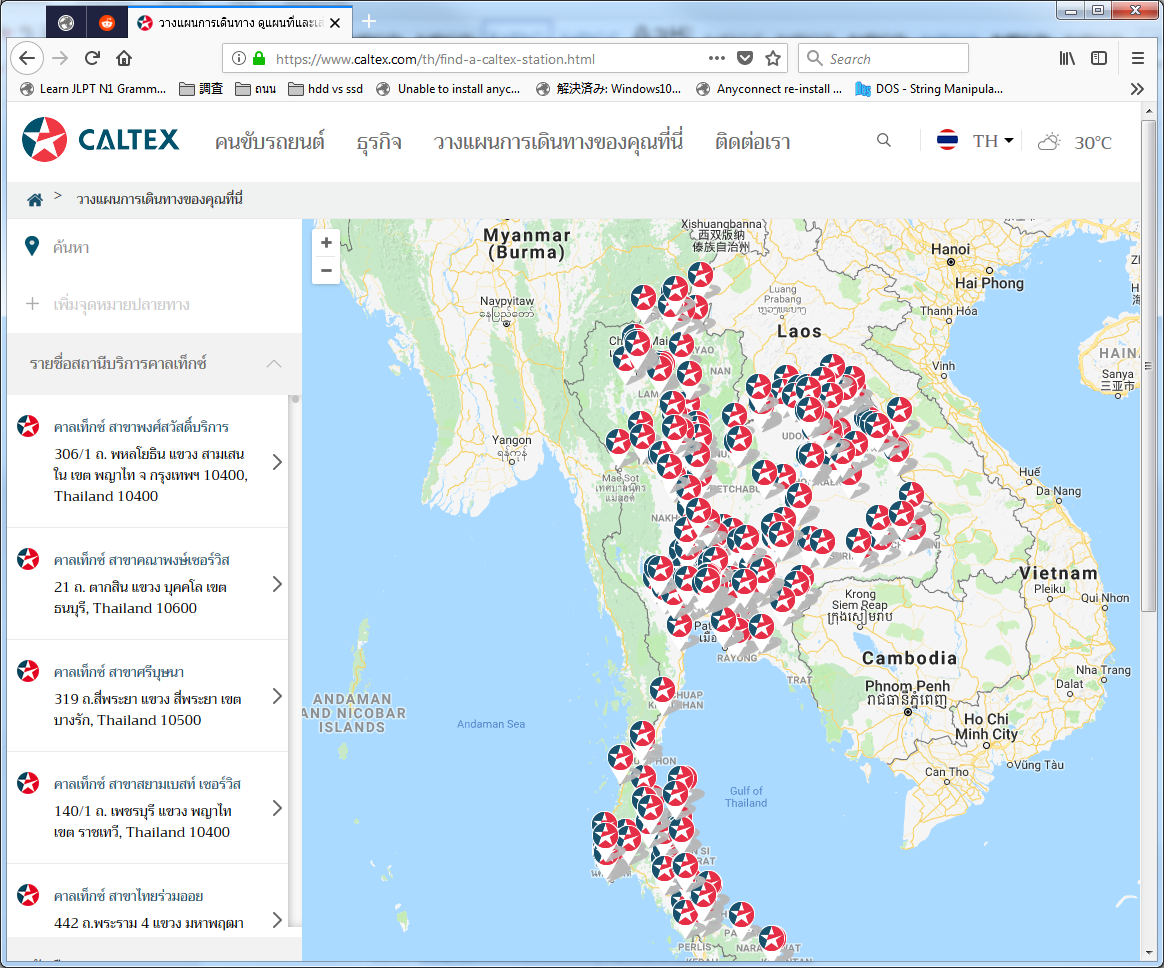
* ธนาคารกสิกรไทย (<https://www.kasikornbank.com/en/branch>)



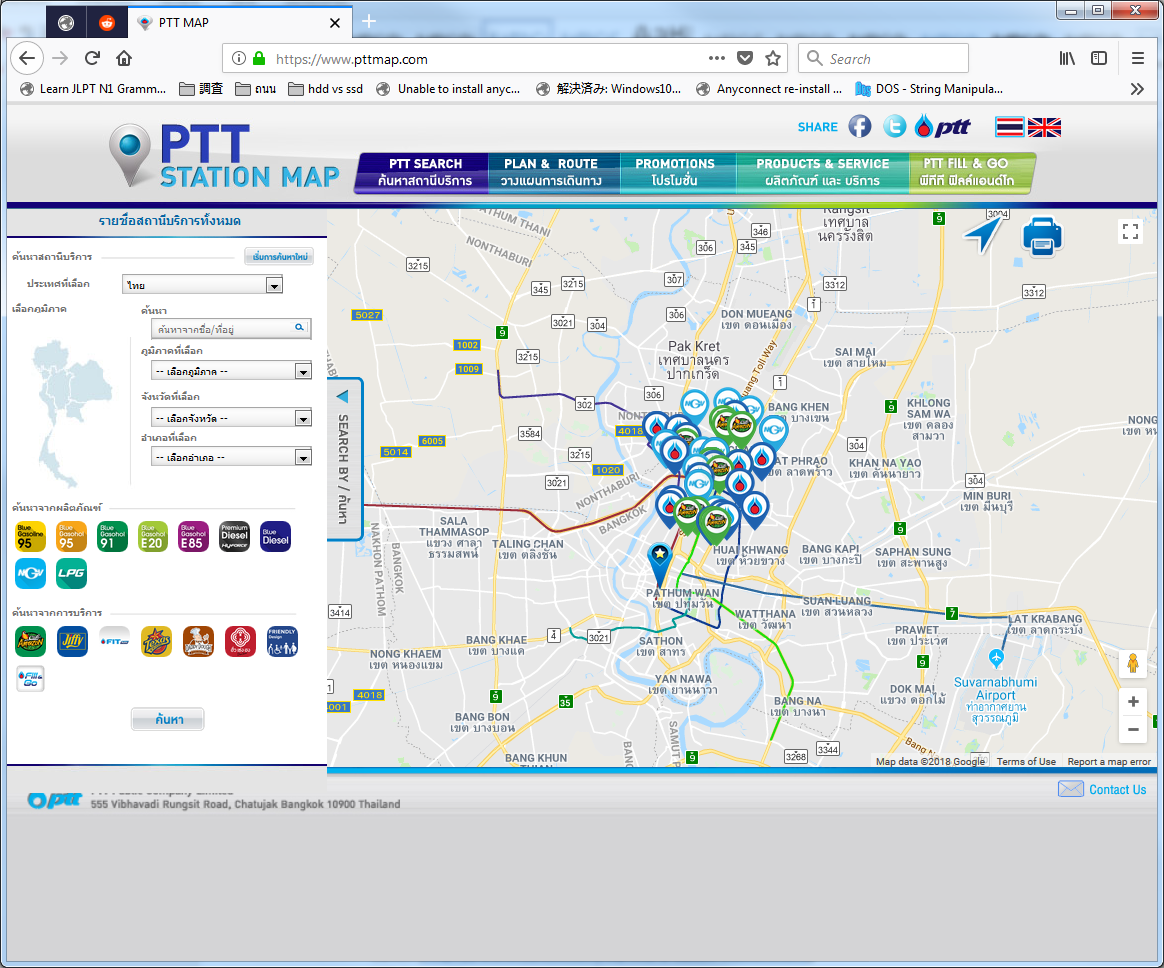
* ธนาคารกรุงศรีอยุธยา (<https://www.krungsri.com/bank/en/Bank-Locations-and-Branches/BranchLocations.html>)



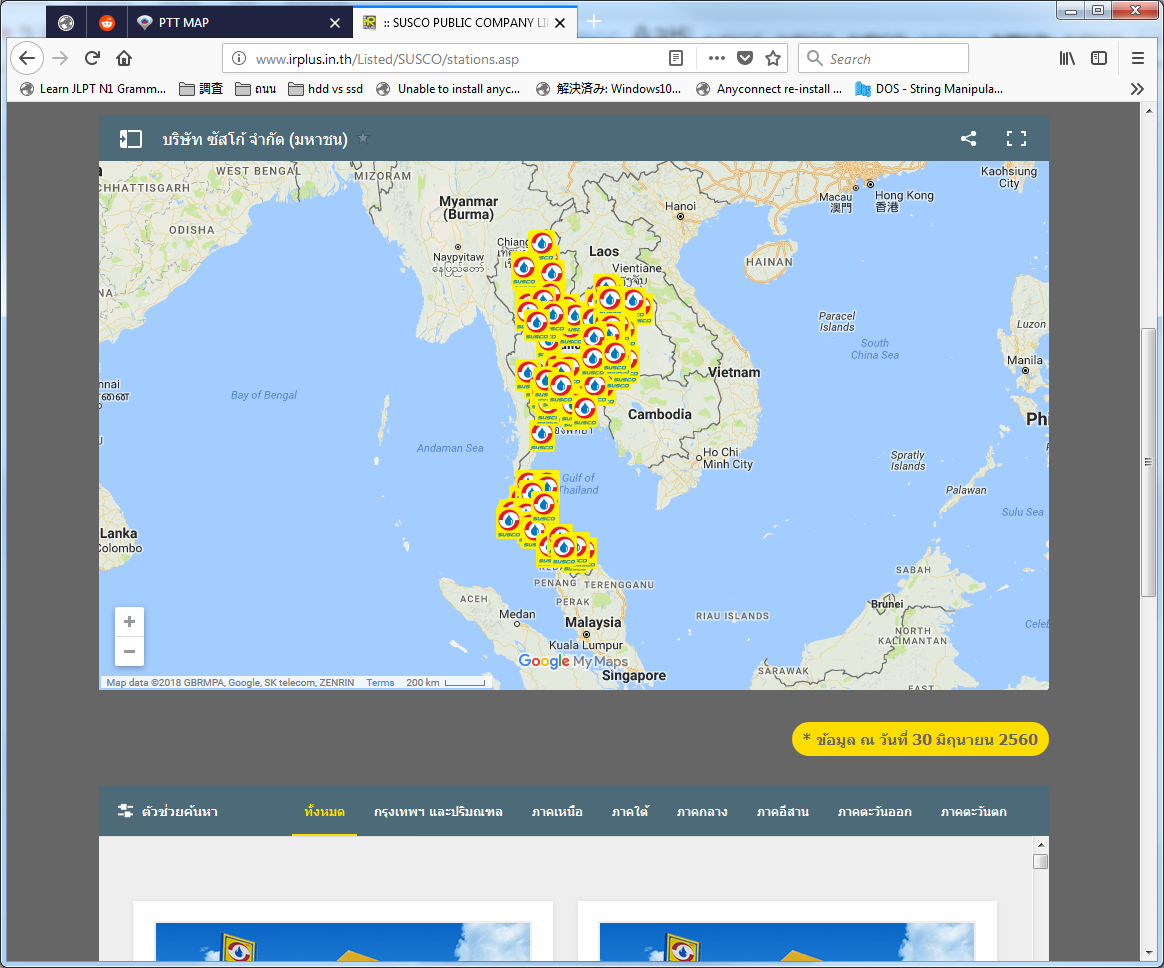
* Caltex (<https://www.caltex.com/th/find-a-caltex-station.html>)



* ปตท (<https://www.pttmap.com/>)



* ซัสโก้ (<http://www.irplus.in.th/Listed/SUSCO/stations.asp>)



จะเห็นได้ว่าเว็บไซต์ส่วนมาก จะมีการแสดงสถานที่บน google map และมีลิสต์สาขาควบอยู่ด้วย ภารกิจของเราคือดึงข้อมูลบนเว็บไซต์ที่เป็นเป้าหมายของเรา (พิกัด ชื่อสาขา ที่อยู่ เป็นต้น)

# การทำงานทั่วไปของโปรแกรม

ในหัวข้อนี้ หลังจากที่เราได้กล่าวถึงการทำงานคร่าว ๆ และเป้าหมายไปแล้ว เราก็จะมาคลายข้อสงสัยกันว่าแท้จริงแล้วโปรแกรมมีการทำงานอย่างไร

รูปร่างหน้าตาของโปรแกรม

เปิดเว็บไซต์เป้าหมาย และเข้าไปยังหน้าเพจที่เราต้องการ

ลอกพฤติกรรมมนุษย์ได้ เช่น เปิดลิงค์ พิมพ์คียเวิร์ดและกดคลิดปุ่มค้นหา

เลือกตัวอักษร ภาพ หรือลิงค์ที่ต้องการเก็บไว้บน Excel โปรแกรมจะเลือกข้อมูลที่คล้ายกันให้อัตโนมัติ

กด run หรือ extract เพื่อดึงข้อมูล

กำหนดการทำงานเป็นรายวัน รายสัปดาห์ รายเดือนได้

และอื่น ๆ อีกมากมายที่คร้านจะกล่าวถึง

# เปรียบเทียบหลาย ๆ โปรแกรม Web Scraper

Web Scraper เป็นชื่อเรียกรวม ๆ สำหรับโปรแกรมประเภทนี้ ซึ่งจากการทดลองใช้งาน ก็พบโปรแกรมหลาย ๆ รูปแบบ ตั้งแต่ฝากให้โปรแกรมจัดการทุกอย่าง หรือใช้งานง่ายแต่ไม่แรง ไปจนถึง ต้องเขียนเป็นโค้ดเพื่อดึงข้อมูล แต่ในที่สุดก็มาทางสายกลางคือ โปรแกรมมีอินเตอร์เฟสใช้งานง่าย แต่ผู้ใช้ก็สามารถกำหนดการทำงานของโปรแกรมได้ลึกและเต็มที่ ดังที่ศาสนาพุทธได้สอนไว้ แต่ไม่ค่อยทำกัน

ฟังก์ชันพื้นฐานที่ทุกโปรแกรมมีนั้น จะเป็นดังที่กล่าวไปในหัวข้อก่อน คือ เข้าถึงเว็บเพจ กำหนดข้อมูลที่จะดึง และทำการรันโปรแกรม แต่รายละเอียดที่มากกว่านั้นจะเป็นไปตามด้านล่าง

* ทุกโปรแกรมจะมีเบราว์เซอร์ของตัวเอง แต่คุณภาพนั้นจะแตกต่างกันไป บางอันเปิดเว็บไซต์เฉพาะแล้วจะโหลดข้อมูลไม่ติด หรือบางอันก็งอแงพังบ่อย พวกนี้เราจะตัดทิ้งไป
* บางอันจะมีฟังก์ชันทำงานแบบวนลูป ช่วยให้เราดึงข้อมูลที่เป็นชุด ๆ ออกมาได้
* โปรแกรมบางตัวจะมีการทำงานแบบ ปล่อยให้เรียนรู้ โดยใส่ชุดเว็บเพจลงไปเยอะ ๆ เพื่อให้มันทำงานได้เอง โปรแกรมชนิดนี้จะมีข้อดีคือ หากเว็บไซต์มีการเปลี่ยนแปลงอินเตอร์เฟสภายหลัง โปรแกรมก็ยังทำงานได้ถูกต้อง
* แต่ปกติจะแพงมาก เลยไม่เอา

# Content Grabber tool usage example (true usage)

* 2. start from tool interface -> website
* 3. start from fujirestaurant (describe as 'noob')
* 4. show real extract process
* 5. show debug and exported data
* 6. show how to run data
* 7. show how to run with KBANK
* 8. show how to input multiple link
* 9. show how to cut word to get coordination from link
* 9.5 show human method to get information from wongnai (province -> district -> next page -> store detail)
* 10. show real example with wongnai-full
* 11. describe knowledge in html is need (with first list where auto detect not work)
* 12. show sub-link page and how they process
* 13. show pagination
* 14. show detailed page and end

# Con/Limitation in functionality

* Can be slow
* Required knowledge in html, some experience tinkering with tool
* Buggy at times

# Manual

* run multiple agent at once
* select element
* select list of element
* choose command
* manually select element via HTML(XPath) when interface selection fail
* set page/action delay
* extract element
* transform text/get specific data from element
* add multiple link
* choose category -> do loop
* add list of keyword into search bar
* click next page
* (special) extract coordinate from google map
* (important) know website structure/data and our objective clearly before creating a extractor