

Architectural Patterns/Styles

1.Selenium WebDriver

Selenium (ซีลีเนียม) เป็น Software Testing Framework (ซอฟต์แวร์ ทดสอบ เฟรมเวิร์ค) ที่มีประสิทธิภาพตัวหนึ่ง เอาไว้ใช้สำหรับทำ Automated Testing (อัตโนมัติ ทดสอบ) เขียน Test Case (เทส เคส) เพื่อทดสอบเว็บ แอปพลิเคชัน

Selenium เป็น Open-Source จึงสามารถทำให้นั้นสามารถ Customize เพื่อเพิ่มประสิทธิภาพ WebDriver ได้ตามต้องการ สามารถไปร่วมกับ Tools อื่น ๆ ได้ตามต้องการ โดย ซีลีเนียมนั้น สามารถพัฒนาได้จากภาษาคอมพิวเตอร์ได้มากมาย ได้แก่ Java Python Ruby C# JavaScript PHP และ Perl และซีลีเนียมยังสามารถรันบนเว็บเบราว์เซอร์ต่าง ๆ ได้มากมาย เช่น Google Chrome Firefox Safari Opera Internet Explorer และอื่นๆ

โดย Selenium มี 2 แบบ ด้วยกัน คือ Selenium IDE คือ Add-On บน Firefox สามารถทำ Record & Playback คนที่ไม่มีพื้นฐานการเขียนโปรแกรมก็ทำได้ ส่วนอีกแบบ Selenium WebDriver คือ API ที่ใช้ในการ Drive เบราว์เซอร์ให้ทำตามที่เราต้องการ โดยเราสามารถใช้ Programming language ที่เรานัดในการสั่งได้

- Architectural patterns/styles



เป็นแบบ Client-server โดยเมื่อคุณไม่เรียกใช้เพื่อรันโปรแกรม เบราว์เซอร์จะเปิดตัวและจะนำทางไปยังเว็บไซต์ โดยวิธี Selenium สื่อสารกับ Selenium API ซึ่งจะส่งคำสั่งภาษาการเขียนโปรแกรมไปยังไดรเวอร์ของเบราว์เซอร์ผ่านโปรโตคอลของ JSON คำสั่งจะถูกส่งในรูปแบบของคำขอ JSON โดยที่โปรโตคอลจะแปลงเป็นคำขอ HTTPS ซึ่งไดรเวอร์เบราว์เซอร์จะใช้เซิร์ฟเวอร์ HTTP นี้เพื่อรับคำขอและส่งไปยัง

เซิร์ฟเวอร์ ซึ่งจะกรองคำสั่งที่จำเป็นต้องดำเนินการ สุดท้ายเซิร์ฟเวอร์ HTTP จะส่งการตอบกลับกลับไปยังสคริปต์ทดสอบ โดยที่ไดรเวอร์และ API แปลงเป็นรูปแบบ JSON และบันทึกผลลัพธ์

ซึ่ง Selenium WebDriver ประกอบด้วยองค์ประกอบหลัก 4 ส่วน ได้แก่

1. Selenium Client Libraries/Language Bindings

Selenium รองรับหลายไลบรารีเช่น Java, Ruby, Python เป็นต้น Selenium Developers ได้พัฒนาการเชื่อมโยงภาษาเพื่อให้ Selenium รองรับหลายภาษา

2. JSON WIRE PROTOCOL ผ่านไคลเอนต์ HTTP

JSON ย่อมาจาก JavaScript Object Notation ใช้ในการถ่ายโอนข้อมูลระหว่างเซิร์ฟเวอร์และไคลเอนต์บนเว็บ JSON Wire Protocol คือ REST API ที่ถ่ายโอนข้อมูลระหว่างเซิร์ฟเวอร์ HTTP BrowserDriver แต่ละตัว (เช่น FirefoxDriver, ChromeDriver เป็นต้น) มีเซิร์ฟเวอร์ HTTP ของตัวเอง

3. ไดรเวอร์เบราว์เซอร์

แต่ละเบราว์เซอร์มีไดรเวอร์เบราว์เซอร์แยกต่างหาก ไดรเวอร์ของเบราว์เซอร์จะสื่อสารกับเบราว์เซอร์ที่เกี่ยวข้องโดยไม่เปิดเผยตรรกะภายในของฟังก์ชันการทำงานของเบราว์เซอร์ เมื่อไดรเวอร์เบราว์เซอร์ได้รับคำสั่งใด ๆ คำสั่งนั้นจะถูกดำเนินการบนเบราว์เซอร์นั้น ๆ และการตอบสนองจะกลับไปในรูปแบบของการตอบสนอง HTTP

4. เบราว์เซอร์

Selenium รองรับเบราว์เซอร์หลายตัว เช่น Firefox, Chrome, IE, Safari เป็นต้น

- Quality attribute scenarios

- Modifiability เพราะเป็นโปรแกรม Open Source และมีการพัฒนาซอฟต์แวร์เป็นได้เรื่อยๆ ตัวอย่างซอฟต์แวร์ Katalon Automation Recorder มีพื้นฐานมาจาก Selenium IDE ของ firefox
- Performance เพราะสามารถเขียนสคริปต์ทดสอบในภาษาที่เว็บแอปพลิเคชันได้รับการเข้ารหัส ซึ่งจะทำให้รอบการทดสอบเร็วขึ้น

- Usability เพราะรองรับการรวมเข้ากับเฟรมเวิร์กต่างๆ รองรับการผลิตรวมกับเฟรมเวิร์กการทดสอบ เช่น TestNG เพื่อปรับปรุงการทดสอบและการรายงานอัตโนมัติ

(อ้างอิง : <https://www.interviewbit.com/blog/selenium-architecture/>

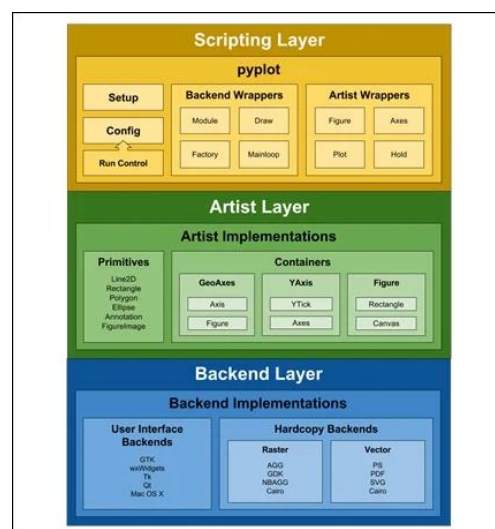
<https://medium.com/edureka/selenium-webdriver-architecture-565e2db26dd5>)

2. Matplotlib

Matplotlib คือไลบรารีหนึ่งของภาษาโปรแกรมมิ่ง Python สำหรับการ Visualize Data หรือก็คือการพล็อตข้อมูลออกมาเป็นรูป กราฟ ชาร์ตต่างๆ มี API เชิงวัตถุ สำหรับการฝังพล็อตลงในแอปพลิเคชันโดยใช้ชุดเครื่องมือ GUI ที่ใช้ งานทั่วไปเช่น Tkinter , wxPython , QtหรือGTK และสามารถใช้ร่วมกับ Numpy ที่เหมาะสำหรับการประมวลข้อมูลแบบตารางในปริมาณมากได้ และ Matplotlib เองก็นับว่าเป็นเครื่องมือไลบรารีแรกๆ ที่เหล่า Data Scientist มือใหม่ต่างก็ลองประเดิมฝีมือกันก่อน เพราะใช้งานง่าย และสามารถนำไปวิเคราะห์ข้อมูลได้หลากหลายรูปแบบ ซึ่งเป็นโอเพ่นซอร์สและสามารถใช้งานได้ฟรี

- Architectural patterns/styles

เป็นแบบ layers โดยเริ่มต้นจากติดตั้งโมดูลก่อนจากนั้นให้เราเรียกใช้งานโมดูล matplotlib กับ numpy เพื่อวาดกราฟและคำนวณค่าต่างๆ ได้ ต่อมากำหนดฟังก์ชันในการแสดงผลลัพธ์ของภาพ จากนั้นสร้างหน้าจอและสร้างตัวแปรในการรองรับภาพ 3 มิติ และการแสดงผลลัพธ์เป็นภาพแบบตะแกรงตะขாய



โดย Matplotlib ประกอบด้วย 3 ชั้นหลัก ได้แก่

1. Backend Layer

จัดการงานหนักทั้งหมดผ่านการสื่อสารไปยังชุดเครื่องมือ เช่น wxPython หรือภาษาการวาด เช่น PostScript ในเครื่อง เป็นเลเยอร์ที่ซับซ้อนที่สุดของไลบรารี Matplotlib

2. Artist Layer

อนุญาตให้ควบคุมและปรับแต่ง Matplotlib ได้อย่างสมบูรณ์ figure— คอนเทนเนอร์ระดับบนสุดสำหรับองค์ประกอบพล็อตทั้งหมด

3. Scripting Layer

เป็นชั้นบนสุดที่ออกแบบมาเพื่อให้ Matplotlib ทำงานเหมือนสคริปต์ MATLAB เป็นชุดของฟังก์ชันรูปแบบคำสั่งและถือเป็นเลเยอร์ที่ง่ายที่สุดในการใช้งาน

- Quality attribute scenarios

- Modifiability เพราะเป็นโปรแกรม Open Source สามารถใช้งานได้ฟรี
- Usability เพราะขอบเขตการทำงานจริงๆกว้างขวาง สามารถใช้เขียนโปรแกรมจัดการรูปภาพได้ และใช้สร้าง GUI อย่างง่ายได้
- Portability เพราะสามารถใช้ได้หลายวิธี Python scripts iPython shells และ Jupyter Notebooks

(อ้างอิง : <https://medium.com/@codingpilot25/architecture-of-matplotlib-1a2d44370f5a>

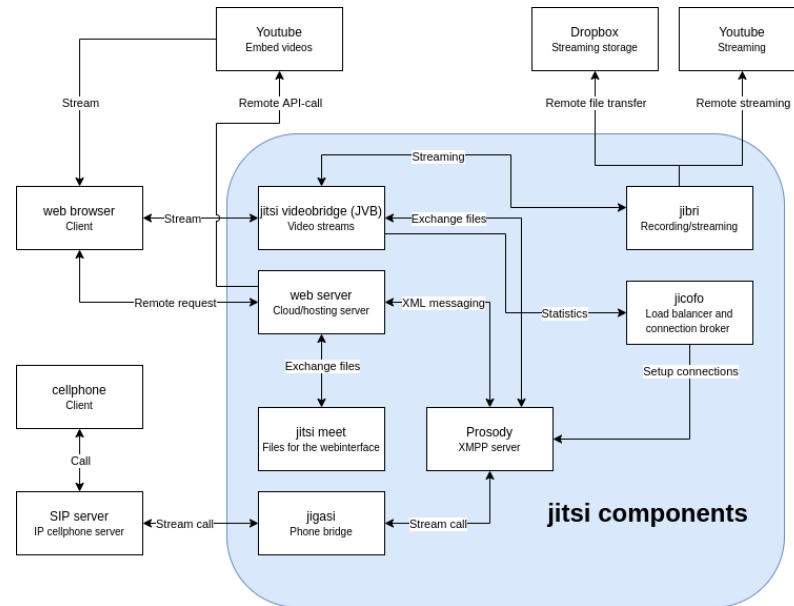
<https://medium.datadriveninvestor.com/data-visualization-with-python-matplotlib-architecture-6b05af533569>)

3. Jitsi

Jitsi Meet คือซอฟต์แวร์การประชุมทางวิดีโอที่ใช้เพื่อสร้างกลุ่มและวิดีโอแชทแบบ 1:1 กับทุกคนได้ แต่มันไม่เหมือนกับโซลูชันการประชุมทางวิดีโออื่นๆ ซอฟต์แวร์การประชุมทางวิดีโออื่นๆ ทั้งหมดที่อาจเคยใช้ จำเป็นต้องให้ลงทะเบียนและสร้างบัญชีเพื่อเริ่มใช้บริการ Jitsi Meet ไม่จำเป็นต้องใช้บัญชี ซึ่งเป็นแพลตฟอร์มที่เน้นความเป็นส่วนตัวและเชื่อมั่นในการจัดเก็บและแชร์ข้อมูลผู้ใช้โดยเริ่มจากระดับพื้นฐานที่สุด เหมาะสำหรับผู้ที่ไม่ต้องการแชร์ที่อยู่อีเมลกับแอปของบุคคลที่สาม เมื่อพูดถึงเรื่องความปลอดภัย เรายังคงได้รับการ

คู่มือ Jitsi Meet เป็นแพลตฟอร์มที่มีการเข้ารหัสและปลอดภัยอย่างสมบูรณ์ ขณะนี้ได้เพิ่มการเข้ารหัสแบบ end-to-end แม้ว่าจะยังอยู่ในรุ่นเบต้าก็ตาม ซึ่งทำให้แพลตฟอร์มนี้ปลอดภัยยิ่งขึ้นสำหรับผู้ใช้ และยังเป็นโอเพ่นซอร์ส 100% ดังนั้นนักพัฒนาจึงสามารถติดตั้งและรันเวอร์ชันของ Jitsi Meet บนเซิร์ฟเวอร์ของตนได้ เช่น Brave Together โดยเบราว์เซอร์ Brave

- Architectural patterns/styles



เป็นแบบ Client-Server การเชื่อมต่อภายนอกสามารถแบ่งออกเป็นสองกลุ่มหลัก ประการแรก การเชื่อมต่อระหว่างไคลเอนต์ที่ร้องขอการเชื่อมต่อวิดีโอหรือเสียงที่ดำเนินการผ่านคำขอระยะไกลและสตรีมข้อมูล ประเภทที่สองของการเชื่อมต่อภายนอกคือการเชื่อมต่อไปยังบริการภายนอกที่ช่วยจัดเก็บบันทึก สตรีมบันทึก สตรีมวิดีโอ หรือช่วยสร้างการประชุม

โดย Jitsi มีส่วนประกอบดังนี้

1. Jitsi Meet

แอปพลิเคชัน JavaScript ที่เข้ากันได้กับ WebRTC ซึ่งใช้ Jitsi Videobridge เพื่อจัดการประชุมทางวิดีโอคุณภาพสูงและปรับขนาดได้ สร้างจาก React และ React Native

2. Jitsi Videobridge (JVB)

เซิร์ฟเวอร์ที่เข้ากันได้กับ WebRTC ซึ่งออกแบบมาเพื่อกำหนดเส้นทางสตรีมวิดีโอระหว่างผู้เข้าร่วมในการประชุม

3. Jitsi Conference Focus (jicofo)

องค์ประกอบโฟกัสฝั่งเซิร์ฟเวอร์ที่ใช้ในการประชุม Jitsi Meet ที่จัดการเซสชันสื่อและทำหน้าที่เป็นตัวจัดสรรภาระงานระหว่างผู้เข้าร่วมแต่ละคนกับสะพานวิดีโอ

4. Jitsi Gateway to SIP (jigasi)

แอปพลิเคชันฝั่งเซิร์ฟเวอร์ที่อนุญาตให้ไคลเอนต์ SIP ปกติเข้าร่วมการประชุม Jitsi Meet

5. Jitsi Broadcasting Infrastructure (jibri)

ชุดเครื่องมือสำหรับบันทึกและ/หรือสตรีมการประชุม Jitsi Meet ที่ทำงานโดยเปิดใช้อินสแตนซ์ Chrome ที่แสดงผลในเฟรมบัพเฟอร์เสมือน และบันทึกและเข้ารหัสเอาต์พุตด้วย ffmpeg

● Quality attribute scenarios

- Modifiability 100% ดังนั้นนักพัฒนาจึงสามารถติดตั้งและรันเวอร์ชันของ Jitsi Meet บนเซิร์ฟเวอร์ของตนได้ เช่น Brave Together โดยเบราว์เซอร์ Brave
- Usability เพราะคุณสมบัติมากมาย เช่น การแชร์หน้าจอ การสตรีมสด การบันทึก ยกมือ เบลอพื้นหลัง ปิดเสียงทุกคน (หรือทั้งหมดยกเว้นหนึ่งรายการ) มุมมองไทม์ และอื่นๆ อีกมากมาย
- Security เพราะไม่จำเป็นต้องใช้บัญชี ซึ่งเป็นแพลตฟอร์มที่เน้นความเป็นส่วนตัวและเชื่อมั่นในการจัดเก็บและแชร์ข้อมูลผู้ใช้โดยเริ่มจากระดับพื้นฐานที่สุด และเป็นแพลตฟอร์มที่มีการเข้ารหัสและปลอดภัยอย่างสมบูรณ์

(อ้างอิง : <https://helplogics.net/th/jitsi-meet->

<https://helplogics.net/th/jitsi-meet-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3%E0%B9%81%E0%B8%A5%E0%B8%B0%E0%B9%83%E0%B8%8A%E0%B9%89%E0%B8%87%E0%B8%B2%E0%B8%99%E0%B8%AD%E0%B8%A2%E0%B9%88%E0%B8%B2%E0%B8%87>)