Selenium Webdriver

Purpose of the project

จุดประสงค์ของ Selenium Webdriver คือ เป็นเครื่องมือที่ช่วยให้สามารถสร้างโปรแกรมในการทดสอบ Web Application กับ Web Brower ได้หลายตัว เป็น Software Testing Framework ที่มีประสิทธิภาพตัวหนึ่งที่เอาไว้ใช้ในการทำ Automated Testing เขียน Test case เพื่อตรวจสอบว่าทำงานได้ตามที่คาดเอาไว้ไหม โดย Selenium Webdriver จะรองรับการเขียนโปรแกรมในภาษาต่างๆ เช่น C#, Java, Perl, PHP และ Ruby

Architectural patterns/styles

เพื่อให้เข้าใจ Selenium Webdriver Architecture เราควรรู้ก่อนว่า Webdriver API คืออะไร Selenium Webdriver API คือตัวช่วยในการสื่อสารระหว่างภาษากับเบราว์เซอร์ โดยเบราว์เซอร์แต่ละตัวมีตรรกะ (Logic) ในการดำเนินการบนเบราว์เซอร์ต่างกัน

รูปแบบสถาปัตยกรรมของ Selenium Webdriver คือ รูปแบบ Representational State Transfer (REST) ซึ่งประกอบด้วยองค์ประกอบหลัก 4 ส่วน ได้แก่

1. Selenium Client Library/Language Bindings

Selenium รองรับไลบรารีหลายตัว เช่น Java, Ruby, Python เป็นต้น โดย Selenium Developers ได้พัฒนาการเชื่อมโยงภาษาต่างๆเพื่อให้ Selenium รองรับหลายภาษา

2. JSON WIRE PROTOCOL Over HTTP Client

JSON ย่อมาจาก JavaScript Object Notation ใช้ในการถ่ายโอนข้อมูลระหว่าง Server และ Client บนเว็บ JSON Wire Protocol เป็น REST API ที่ถ่ายโอนข้อมูลระหว่าง HTTP Server BrowserDriver แต่ละตัวมี HTTP Server เป็นของตัวเอง

3. Browser Drivers

แต่ละเบราว์เซอร์จะมี browser driver แยกต่างหาก driver ของ Browser จะสื่อสารกับเบราว์เซอร์ที่เกี่ยวข้องโดยไม่เปิดเผยตรรกะภายใน (Internal Logic) ของฟังก์ชั่นการทำงานของเบราว์เซอร์ เมื่อ browser driver ได้รับคำสั่งใดๆ มา คำสั่งนั้นก็จะถูก executed บนเบราว์เซอร์นั้นๆ และตอบกลับไปในรูปแบบของ HTTP response

4. Browsers

Selenium รองรับหลายเบราว์เซอร์ เช่น Firefox, Chrome. IE, Safari เป็นต้น

Quality Attributes

Scenario 1 : Usability

- Source : ผู้ใช้งาน (User)
- Stimulus : ต้องการทำ Automated Testing ทดสอบ Web application
- Environment : Runtime
- Artifact : System
- Response : แสดงผลทดสอบ Web application

- Response Measure : ผู้ใช้งานพึงพอใจกับประสิทธิภาพการทำงานของการทดสอบ Web application

Scenario 2 : Performance

- Source : ผู้ใช้งาน
- Stimulus : คำสั่งของผู้ใช้งานต้องการทำ Automated Testing โดยมี 1000 Test case เพื่อทดสอบ Web application
- Environment : Normal mode
- Artifact : System
- Response : Processes all requests
- Response Measure : time the response takes (latency) < 3 min

Scenario 3 : Modifiability

- Source : นักพัฒนา (Developer)
- Stimulus : ต้องการเพิ่มเบราเซอร์ใหม่
- Environment : เวลาในการพัฒนา
- Artifact : Code
- Response : สามารถปรับเปลี่ยนได้โดยไม่ส่งผลข้างเคียงและผ่านการทดสอบ
- Response Measure : ใช้เวลาน้อยกว่า 72 ชั่วโมงในการทำและทดสอบการเปลี่ยนแปลง

แหล่งอ้างอิง

https://economictimes.indiatimes.com/definition/selenium-web-driver

https://www.edureka.co/blog/selenium-webdriver-architecture/

https://aosabook.org/en/selenium.html

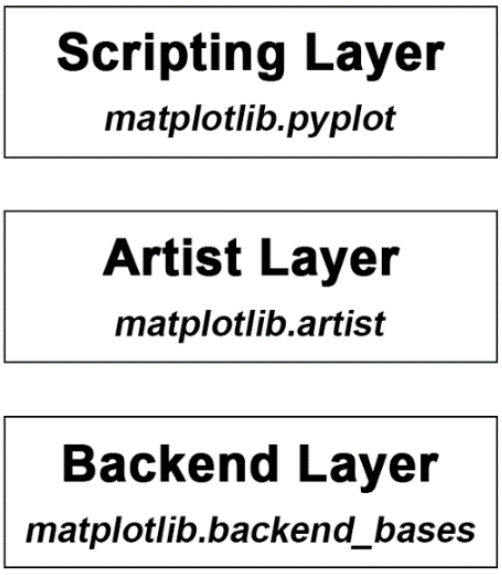
Matplotlib

Purpose of the project

จุดประสงค์ของ matplotlib คือ เป็นไลบรารีการ plotting บน Python ที่พร้อมรองรับ 2D เต็มรูปแบบและรองรับกราฟิก 3D ซึ่งใช้กันอย่างแพร่หลายใน Python scientific computing community

Library มุ่งเป้าไปที่ use cases การใช้งานที่หลากหลาย มันสามารถฝังกราฟิกใน user interface toolkit ของผู้ใช้ที่เลือก และปัจจุบันรองรับinteractive graphics บนระบบปฏิบัติการ desktop หลักทั้งหมดโดยใช้ GTK+, Qt, Tk, FLTK, wxWidgets และ Cocoa toolkits มันสามารถ called interactively จาก interactive Python shell เพื่อสร้างกราฟิกด้วยคำสั่งขั้นตอนง่าย ๆ เช่น Mathematica, IDL หรือ MATLAB อีกทั้งmatplotlib สามารถฝังใน headless webserver เพื่อ hardcopy ในรูปแบบ raster-based formats เช่น Portable Network Graphics (PNG) และรูปแบบเวกเตอร์ เช่น PostScript, Portable Document Format (PDF) และ Scalable Vector Graphics (SVG)

Architectural patterns/styles



รูปแบบสถาปัตยกรรมของ matplotlib คือ รูปแบบ Layer Architectural ซึ่งประกอบด้วย 3 Layers คือ

1. **Backend Layer** เป็น Layer ที่ใช้สำหรับการแสดงผลของ figure ที่สร้างขึ้นมา

2. **Artist Layer** เป็น Layer ที่ผู้ใช้งานสามารถเขียนโปรแกรมเพื่อใช้งานกับ matplotlib ได้ และช่วยควบคุมและปรับจูน UI ของ figure เช่น spines, tick direction, tick label size, tick label font, tick color

3. **Scripting Layer** เป็น Layer ที่ใช้สำหรับการเขียนโปรแกรม โดยส่วนนี้จะเป็นส่วนที่ผู้ใช้เขียนโปรแกรมเพื่อใช้งานกับ matplotlib เขียนโปรแกรมเพื่อสร้างกราฟ ออกแบบมาเพื่อให้ matplotlib ทำงานเหมือน MATLAB Script

Quality Attributes

Scenario 1 : Performance

- Source : ผู้ใช้งาน
- Stimulus : คำสั่งสร้างกราฟจากผู้ใช้งาน
- Environment : สถานะปกติ
- Artifact : เกิด Process Generate กราฟและจับเวลา
- Response : เวลาที่ใช้ในการสร้างกราฟ
- Response Measure : Latency (Min, Max, Average)

Scenario 2 : Usability

- Source : ผู้ใช้งาน
- Stimulus : ใช้งาน matplotlib ในการสร้างกราฟด้วย Python
- Environment : Deployment, Runtime, Integration
- Artifact : เกิด Process Generate กราฟ
- Response : สำเร็จหรือไม่ ได้ตามต้องการหรือไม่

- Response Measure : มีการแสดงผล

Scenario 3 : Integrability

- Source : ผู้ใช้งาน
- Stimulus : Integrate matplotlib ให้ใช้งานกับภาษา Python ได้
- Environment : Integration
- Artifact : ระบบทั้งหมด
- Response : Integrate สำเร็จหรือไม่
- Response Measure : การแสดงผล ความสำเร็จในการใช้งาน
-

แหล่งอ้างอิง

https://www.aosabook.org/en/matplotlib.html

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

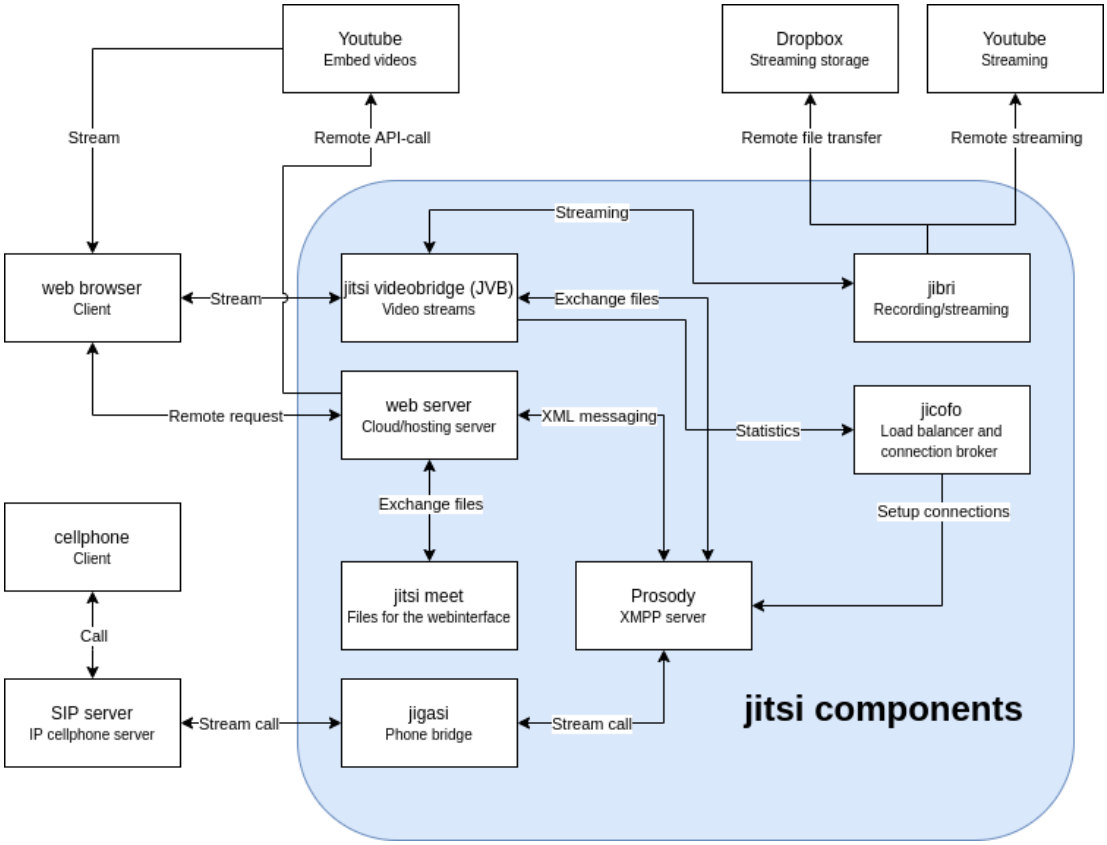
https://medium.com/dataseries/mastering-matplotlib-part-1-a480109171e3

Jitsi

Purpose of the project

จุดประสงค์ของ Jitsi คือ เป็น Open-source projects สำหรับการสร้าง video conference และ chat เป็นแอปพลิเคชั่นที่ช่วยให้ผู้คนสามารถโทรผ่านวิดีโอและสนทนา แชร์เดสก์ท็อป และแลกเปลี่ยนไฟล์และข้อความ ที่สำคัญกว่านั้นคืออนุญาตให้ผู้คนทำสิ่งนี้ผ่านโปรโตคอลต่างๆ ตั้งแต่ XMPP มาตรฐาน (Extensible Messaging and Presence Protocol) และ SIP (Session Initiation Protocol) ไปจนถึงกรรมสิทธิ์เช่น Yahoo! และ Windows Live Messenger (MSN) โดย Jitsi ทำงานบน Microsoft Windows, Apple Mac OS X, Linux และ FreeBSD ส่วนใหญ่เขียนด้วยภาษา Java แต่ก็มีส่วนที่เขียนด้วย Native code เป็นโปรเจคที่เปิดโอกาสให้ผู้ใช้งานสามารถเข้ามาเพิ่มเติม แก้ไข หรือปรับปรุงโปรแจคได้

Architectural patterns/styles



รูปแบบสถาปัตยกรรมของ Jitsi คือ รูปแบบ Layer Architectural ซึ่งประกอบไปด้วย

1. **Jitsi Meet** – JavaScript application ที่เข้ากันได้กับ WebRTC ซึ่งใช้ Jitsi Videobridge เพื่อจัดการประชุมทางวิดีโอคุณภาพสูงและปรับขนาดได้ สร้างจาก React และ React Native

2. **Jitsi Videobridge (JVB)** – Server ที่เข้ากันได้กับ WebRTC ซึ่งออกแบบมาเพื่อกำหนดเส้นทางสตรีมวิดีโอระหว่างผู้เข้าร่วมในการประชุม

3. **Jitsi Conference Focus (jicofo)** – Server-side focus component ที่ใช้ในการประชุม Jitsi Meet ที่จัดการ Media sessions และทำหน้าที่เป็นตัว Load balancer ระหว่างผู้เข้าร่วมแต่ละคนกับ Viodeobridge

4. **Jitsi Gateway to SIP (jigasi)** – Server-side application ที่อนุญาตให้ไคลเอ็นต์ SIP ปกติเข้าร่วมการประชุม Jitsi Meet

5. **Jitsi Broadcasting Infrastructure (jibri)** – ชุดเครื่องมือสำหรับการบันทึกหรือสตรีมการประชุม

Quality Attributes

Scenario 1 : Usability

- Source : ผู้ใช้งาน (User)
- Stimulus : ผู้ใช้งานจะทำการ Video Conference โดยใช้ Jitsi
- Environment : Runtime
- Artifact : System
- Response : สามารถเข้าร่วมสำเร็จหรือไม่
- Response Measure : สามารถเข้าร่วม Video Conference ได้ภายในเวลาน้อยกว่า 1 นาที

Scenario 2 : Performence

- Source : ผู้ใช้งาน (User)
- Stimulus : คำสั่งของผู้ใช้งานสร้าง Video Conference
- Environment : Normal mode
- Artifact : System
- Response : ระบบทำตามคำสั่งของผู้ใช้งาน
- Response Measure : Average latency < 1000ms

Scenario 3 : Security

- Source : ผู้ใช้งานที่ไม่ได้รับอนุญาต
- Stimulus : ผู้ใช้งานที่ไม่ได้รับอนุญาตจะเข้าร่วม Video Conference โดยใช้ Jitsi
- Environment : Normal operation
- Artifact : System
- Response : ไม่สามารถเข้าร่วม Video Conference ได้
- Response Measure : สามารถตรวจจับผู้ใช้งานที่ไม่ได้รับอนุญาตแล้วจะเข้าร่วม Video Conference ในไม่กี่วินาที
-

แหล่งอ้างอิง

https://www.aosabook.org/en/jitsi.html

https://jitsi.github.io/handbook/docs/architecture/