

1. Selenium WebDriver

Purpose

เพื่อใช้ทดสอบเว็บแอปพลิเคชันในเบราว์เซอร์ต่างๆ

Architectural patterns/styles

ประกอบด้วยองค์ประกอบหลักสี่ส่วน คือ

1. Selenium Client library ให้การสนับสนุนไลบรารีต่างๆ เช่น Ruby, Python, Java เป็นต้น
2. JSON wire protocol over HTTP เป็นมาตรฐานเปิดที่มีกลไกการขนส่งสำหรับการถ่ายโอนข้อมูลระหว่างไคลเอนต์และเซิร์ฟเวอร์บนเว็บ รองรับโครงสร้างข้อมูลต่างๆ เช่น อาร์เรย์และอ็อบเจกต์ ซึ่งให้อ่านและเขียนข้อมูลจาก JSON ได้ง่ายขึ้น
3. Browser Drivers ไดรเวอร์เฉพาะสำหรับแต่ละเบราว์เซอร์และโดยไม่เปิดเผยตรรกะภายในของการทำงานของเบราว์เซอร์ ไดรเวอร์ของเบราว์เซอร์จะโต้ตอบกับเบราว์เซอร์ที่เกี่ยวข้องโดยสร้างการเชื่อมต่อที่ปลอดภัย ไดรเวอร์เบราว์เซอร์เหล่านี้มีความเฉพาะเจาะจงสำหรับภาษาที่ใช้สำหรับกรณีทดสอบอัตโนมัติ เช่น C# , Python , Java เป็นต้น
4. Browsers ให้การสนับสนุนเบราว์เซอร์หลายตัว เช่น Chrome, Firefox, Safari, Internet Explorer เป็นต้น

Quality Attribute

compatibility, flexibility, responsibility

Resource

<https://www.browserstack.com/guide/selenium-webdriver-tutorial#:~:text=Selenium%20WebDriver%20is%20a%20web,language%20to%2>

2. Matplotlib

Purpose

เพื่อสร้างการแสดงผลภาพแบบสแตติก ภาพเคลื่อนไหว และการวางแผนกราฟิก

Architectural patterns/styles

ประกอบด้วยองค์ประกอบหลักสามส่วน คือ

1. Back-end layer จัดเตรียมการใช้งานคลาสอินเทอร์เฟซนามธรรมสามคลาส

2. Artist layer ทุกสิ่งที่เราเห็นในรูปของ matplotlib คือตัวอย่างของศิลปิน ตัวอย่าง: ชื่อเรื่อง บรรทัด
ป้ายกำกับ รูปภาพ เป็นต้น

3. Scripting layer ประกอบด้วย pyplot เป็นหลัก ซึ่งเป็นอินเทอร์เฟซที่เบากว่าเลเยอร์ Artist

Quality Attribute

flexibility, probability, visibility

Resource

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

3. Joomla

Purpose

เพื่อจัดการเนื้อหาแบบโอเพ่นซอร์ส (CMS) ซึ่งใช้ในการสร้างเว็บไซต์และแอปพลิเคชันออนไลน์

Architectural patterns/styles

ประกอบด้วยองค์ประกอบหลักเจ็ดส่วน คือ

1. Database ชุดของข้อมูลและสามารถจัดเก็บ จัดการ และจัดระเบียบในลักษณะเฉพาะได้ ฐานข้อมูลเก็บข้อมูลผู้ใช้ เนื้อหา และข้อมูลที่จำเป็นอื่น ๆ
2. Joomla Framework คือชุดของซอฟต์แวร์โอเพ่นซอร์สที่สร้าง Joomla CMS ได้รับการพัฒนาให้มีความยืดหยุ่นมากขึ้นและแบ่งกรอบงานออกเป็นแพ็คเกจโมดูลาร์เดียว ซึ่งช่วยให้แต่ละแพ็คเกจพัฒนาได้ง่ายขึ้น
3. Components เป็นแอปพลิเคชันขนาดเล็ก ประกอบด้วยสองส่วนคือ ผู้ดูแลระบบและไซต์ เมื่อใดก็ตามที่มีการโหลดหน้า Components จะถูกเรียกเพื่อแสดงเนื้อหาของหน้าหลัก ส่วนผู้ดูแลระบบจัดการองค์ประกอบที่แตกต่างกันและส่วนไซต์ช่วยในการแสดงหน้าเมื่อมีการร้องขอโดยผู้เยี่ยมชมไซต์ Components เป็นหน่วยหน้าที่หลักของ Joomla
4. Modules เป็นส่วนเสริมที่ใช้ในการแสดงหน้าใน Joomla นอกจากนี้ยังใช้เพื่อแสดงข้อมูลใหม่จากส่วนประกอบ มักจะดูเหมือนกล่องเช่นโมดูลการเข้าสู่ระบบ ในผู้ดูแลระบบ Joomla โมดูลจะได้รับการจัดการโดยตัวจัดการโมดูล จะแสดงเนื้อหาและรูปภาพใหม่เมื่อโมดูลเชื่อมโยงกับส่วนประกอบ Joomla
5. Plugin เป็นส่วนขยายของ Joomla ซึ่งมีความยืดหยุ่นและมีประสิทธิภาพมากสำหรับการขยายเฟรมเวิร์ก ประกอบด้วยรหัสบางส่วนที่ใช้ในการเรียกใช้งานทริกเกอร์เหตุการณ์เฉพาะ โดยทั่วไปจะใช้เพื่อจัดรูปแบบผลลัพธ์ของส่วนประกอบหรือโมดูลเมื่อสร้างเพจ ฟังก์ชันปลั๊กอินที่เกี่ยวข้องกับเหตุการณ์จะดำเนินการตามลำดับเมื่อมีเหตุการณ์เฉพาะเกิดขึ้น
6. Templates เป็นตัวกำหนดรูปแบบลักษณะของเว็บไซต์ Joomla เทมเพลตที่ใช้มี 2 ประเภทคือ Front-end และ Back-end เทมเพลต Back-end ใช้เพื่อควบคุมฟังก์ชันโดยผู้ดูแลระบบ โดยที่เทมเพลต Front-end เป็นวิธีนำเสนอเว็บไซต์แก่ผู้ใช้ เทมเพลตนั้นง่ายต่อการสร้างหรือปรับแต่งไซต์ของคุณ ให้ความยืดหยุ่นสูงสุดในการจัดรูปแบบไซต์ของคุณ
7. Web Server เป็นเซิร์ฟเวอร์ที่ผู้ใช้ได้ตอบ มันส่งหน้าเว็บให้กับลูกค้า HTTP (Hyper Text Transfer Protocol) ใช้ในการสื่อสารระหว่างไคลเอนต์และเซิร์ฟเวอร์

Quality Attribute

compatibility, flexibility, responsibility

Resource

https://www.tutorialspoint.com/joomla/joomla_architecture.html