**Mobile Testing with serenity**

Appium adalah kerangka kerja otomatisasi seluler yang digunakan untuk menjalankan pengujian otomatisasi pada perangkat seluler. Beberapa informasi penting tentang Appium adalah sebagai berikut:

1. Dukungan Bahasa Pemrograman: Appium mendukung berbagai bahasa pemrograman seperti Java, Python, JavaScript, Ruby, dan masih banyak lagi. Hal ini memungkinkan pengguna untuk menulis skrip pengujian dalam bahasa pemrograman yang mereka pilih.
2. Dibangun di atas Selenium: Appium dibangun di atas pustaka Selenium, yang merupakan kerangka kerja otomatisasi web yang populer. Dengan memanfaatkan Selenium WebDriver, Appium memungkinkan pengguna untuk mengotomatisasi pengujian pada perangkat seluler dengan menggunakan bahasa pemrograman yang kompatibel dengan Selenium.
3. Dukungan untuk Android, iOS, dan Windows: Appium adalah kerangka kerja yang lintas platform, yang berarti dapat digunakan untuk menguji aplikasi pada platform Android, iOS, dan Windows. Ini memberikan fleksibilitas kepada pengguna untuk mengotomatisasi pengujian pada berbagai perangkat seluler.

Dengan menggunakan Appium, pengembang dan pengujian dapat mengotomatisasi interaksi pengguna pada aplikasi seluler, seperti mengklik tombol, memasukkan teks, menggulir, dan banyak lagi. Ini membantu dalam memastikan kualitas aplikasi seluler dengan mengotomatisasi pengujian dan mengidentifikasi bug atau masalah yang mungkin terjadi selama pengembangan aplikasi.

Benar, dalam Appium terdapat dua komponen utama yang digunakan:

1. Server Appium: Server Appium berfungsi sebagai jembatan antara perangkat pengujian (seperti perangkat Android atau iOS) dan skrip pengujian yang dijalankan. Ada dua cara untuk menggunakan Server Appium:  
   a. Appium Desktop: Appium Desktop adalah aplikasi desktop yang menyediakan antarmuka grafis untuk mengelola server Appium. Dengan menggunakan Appium Desktop, pengguna dapat mengatur preferensi server, mengelola pengaturan perangkat, mengawasi log, dan menjalankan skrip pengujian.  
   b. Appium melalui command-line: Appium juga dapat diinstal melalui NPM (Node Package Manager) dan dijalankan melalui command-line. Dengan pendekatan ini, pengguna dapat menggunakan perintah terminal untuk mengatur dan menjalankan server Appium.
2. Klien Appium: Klien Appium adalah pustaka yang ditulis dalam berbagai bahasa pemrograman (seperti Java, Python, JavaScript, Ruby, dll.) yang memungkinkan pembuatan skrip pengujian. Pustaka ini berfungsi untuk mengirimkan perintah ke server Appium dan mengontrol perangkat pengujian. Dengan menggunakan klien Appium, pengguna dapat mengotomatisasi interaksi pengguna pada aplikasi seluler yang diuji, seperti mengklik tombol, memasukkan teks, memverifikasi elemen, dan lain-lain.

Kombinasi antara server Appium dan klien Appium memungkinkan pengguna untuk mengendalikan perangkat pengujian secara jarak jauh dan mengautomatisasi pengujian pada aplikasi seluler melalui skrip pengujian yang ditulis dalam bahasa pemrograman yang diinginkan.

Kerangka kerja Mentimun (Cucumber) adalah kerangka kerja yang mendukung pembangunan berbasis perilaku (Behavior-Driven Development/BDD) untuk pengujian otomatis. Beberapa informasi penting tentang Mentimun adalah sebagai berikut:

1. Pendekatan BDD: Mentimun menggunakan pendekatan BDD, yang berfokus pada kolaborasi antara stakeholder dalam pengembangan perangkat lunak. Dalam BDD, skenario pengujian ditulis dalam bahasa yang dapat dimengerti oleh semua pihak yang terlibat, termasuk pengembang, pengujian, dan pemangku kepentingan bisnis.
2. Sintaks Gherkin: Mentimun menggunakan sintaks Gherkin untuk menulis skenario pengujian. Gherkin adalah bahasa yang mudah dibaca dan ditulis yang memungkinkan kolaborasi dan pemahaman yang lebih baik antara berbagai pihak yang terlibat. Sintaks Gherkin terdiri dari kata kunci seperti "Given" (Diberikan), "When" (Ketika), dan "Then" (Maka), yang digunakan untuk menggambarkan langkah-langkah dalam skenario pengujian.
3. Dukungan Bahasa Pemrograman: Mentimun mendukung berbagai bahasa pemrograman, termasuk Java, Ruby, JavaScript, Python, dan lainnya. Hal ini memungkinkan pengguna untuk menulis skrip pengujian dalam bahasa pemrograman yang mereka pilih dan nyaman.
4. Dukungan dengan Intellij IDEA IDE: Mentimun memiliki fitur dan dukungan yang kuat dengan Intellij IDEA IDE (Integrated Development Environment). IDE ini menyediakan alat-alat yang berguna untuk mengelola dan menjalankan skrip Mentimun, serta memberikan fitur debugging yang bermanfaat.
5. Dokumentasi: Untuk informasi lebih lanjut tentang Mentimun dan cara menggunakannya, Anda dapat mengunjungi situs resmi Mentimun di<https://cucumber.io/>. Di situs tersebut, Anda dapat menemukan dokumentasi yang komprehensif tentang penggunaan dan fitur Mentimun.
6. Serenity: Mentimun memiliki ketergantungan dengan Serenity, sebuah kerangka kerja pengujian yang mencakup pelaporan yang kaya fitur dan manajemen skenario. Dengan menggunakan Mentimun, Anda sudah memiliki ketergantungan Serenity secara otomatis, sehingga Anda tidak perlu menambahkan ketergantungan Mentimun secara terpisah.

Mentimun (Cucumber) adalah kerangka kerja yang kuat dan fleksibel untuk pengujian otomatis berbasis perilaku. Dengan pendekatan BDD dan sintaks Gherkin yang mudah dibaca, Mentimun memungkinkan kolaborasi yang baik antara stakeholder dan memberikan kejelasan dalam menggambarkan perilaku aplikasi yang diuji.