

Automatització de proves funcionals amb Selenium i Github Actions





Objectius:

Comprendre Selenium

Aprendre els fonaments de Selenium i com s'utilitza per a realitzar proves funcionals, tant per a aplicacions web com per a aplicacions mòbils.

Automatitzar proves amb Github Actions

Integrar Selenium amb Github Actions per Actions per a la creació de fluxos de treball treball automatitzats que executen proves proves de manera periòdica o després de després de cada canvi de codi.

Millorar la qualitat del codi

Entendre com les proves automatitzades amb Selenium i Github Actions contribueixen a identificar errors d'una manera ràpida i eficaç, ajudant a garantir la qualitat del codi.

Relació amb conceptes previs

1

Proves unitàries

Les proves unitàries aïllen i validen el funcionament correcte de mòduls de codi. Aquestes proves són fonamentals per assegurar que cada part del codi funciona correctament per si sola.

2

Proves funcionals

Les proves funcionals validen el funcionament de l'aplicació com a un tot. Aquestes proves simulen el comportament de l'usuari, validant la interfície i la lògica de l'aplicació.

3

Automatització

L'automatització d'aquestes proves amb Github Actions simplifica el procés de validació, assegurant que el codi funciona correctament amb cada canvi.



Unit Test

Què és Selenium?

Selenium és un conjunt de eines que permet automatitzar proves funcionals en navegadors web.

Aquestes proves simulen les accions d'un usuari real, validant la funcionalitat del lloc web.



Per què utilitzar Selenium?



Simulació d'Accions d'Usuari

Selenium permet simular el comportament d'un usuari real, com ara omplir ara omplir formularis, clicar botons i navegar per enllaços.



Validació de Comportament

A través de proves d'automatització, s'asseguren que l'aplicació funciona correctament segons les especificacions del projecte.



Integració amb Proves Unitàries

Selenium complementa les proves unitàries, enfocant-se en el comportament extern de l'aplicació, com ara interaccions amb la interfície d'usuari.

Components principals de Selenium



WebDriver	Control dels navegadors web per per simular accions de l'usuari. l'usuari.
Compatibilitat	Funcionalitat amb diferents navegadors com Chrome, Firefox, Firefox, Edge, etc.
Mode headless	Execució de proves sense visualitzar la interfície del navegador, ideal per a servidors. servidors.

Què és Github Actions?

GitHub Actions és una plataforma d'automatització de fluxos de treball per a repositoris de GitHub. Aquesta plataforma permet crear automatitzacions per a tasques repetitives, com ara la compilació del codi, l'execució de proves i el desplegament d'aplicacions. Mitjançant l'ús d'accions, es poden definir fluxos de treball complexos que s'executen en resposta a esdeveniments com ara la creació d'una nova branca, l'enviament d'un commit o la creació d'una nova publicació.



Per què utilitzar Github Actions?

Automatització de proves

S'executen automàticament cada vegada que es fa un canvi al codi.

Aquest fet permet detectar problemes abans que el codi es desplegui.

Millora la qualitat del codi

Els errors es detecten abans que arribin a producció.

El codi es testa de forma consistent, independentment del desenvolupador.

Què és la integració contínua (CI)?



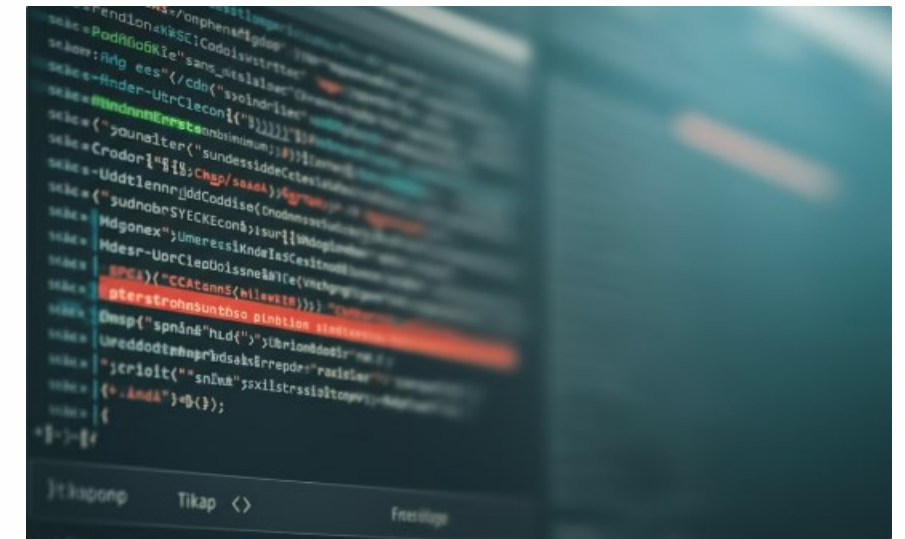
Integració freqüent

La integració contínua implica integrar i provar els canvis de codi amb freqüència, freqüència, com ara diverses vegades al dia.



Automatització de proves

Automatitzar les proves després de cada canvi de codi assegura una detecció ràpida d'errors.



Detecció ràpida d'errors

La integració contínua permet identificar i identificar i corregir errors de manera primerenca, abans que es converteixin en en problemes importants.



Relació entre Selenium i Github Actions

1

Selenium

Selenium és una eina per a proves funcionals. Validem el funcionament extern extern de l'aplicació, com l'interfície d'usuari.

2

Github Actions

Github Actions automatitza aquestes proves. S'executa cada vegada que s'actualitza el codi.

3

Benefici combinat

Garanteix una qualitat de codi consistent. Redueix el treball manual repetitiu i augmenta la confiança en l'aplicació.

Beneficis generals

1 Estalvi de Temps

L'automatització redueix el temps dedicat a les proves manuals.

2 Reducció d'Errors

Les proves automatitzades minimitzen els errors humans, millorant la fiabilitat.

3 Millora de la Qualitat

Els tests funcionals garanteixen que l'aplicació funciona correctament i compleix els requisits.

4 Integració Continua

La integració contínua (CI) garanteix que les modificacions al codi no introdueixen errors.



Extensions possibles



Explorar Django

Per aquells interessats en desenvolupar aplicacions web més complexes, Django és una opció poderosa.



Proves avançades

Automatitzar proves de seguretat, com detectar entrades malicioses en formularis, és essencial per a la seguretat de les aplicacions.



Aprendre més sobre CI/CD

Implementar processos complets de desenvolupament i desplegament contínuament automatitzats permet crear aplicacions més robustes i eficients.