

Integrando Travis CI en su proyecto Github

Publicado el 11 de enero de 2017 Última actualización el 18 de enero de 2017



Travis CI

[Travis CI](#) es un servicio de integración de código abierto alojado, distribuido y continuo que se utiliza para construir y probar proyectos alojados en GitHub.

Integración continua (CI)

La integración continua es un proceso de ejecución de todas las [pruebas unitarias](#) en el último código que se envía para verificar si los componentes recién agregados no han roto ninguna funcionalidad anterior. Es muy importante desde las perspectivas de prueba e implementación. Previene los problemas de reversiones posteriores al despliegue. Son lentos y reducen la fiabilidad de los productos.

Otras opciones

Puede usar [Jenkins] (<https://www.codementor.io/slavko/tutorials/setup-configure-jenkins-recommended-plugins-du107z6nr>) " Configurar y configurar Jenkins para su equipo ") y muchos otros servidores de integración continua para para este propósito, pero para usarlos, tendrá que configurarlos en un servidor y mantener este servidor utilizando su [ingeniero de desarrollo](#) dedicado , es decir, si tiene uno. Si puede lograrlo, le proporcionarán una gran cantidad de Control y flexibilidad.

Por qué Travis CI

Supongamos que no desea contratar a un ingeniero dedicado de operaciones de desarrollo o que no necesita dedicar un servidor a CI, pero desea utilizar CI, ya que es muy importante para su producto. Travis CI viene a su ayuda aquí. No tiene que mantener su propio servidor ni configurar Travis en ningún lado. Solo incluya algunos archivos en su repositorio para que Travis los lea y ya está.

Travis tiene soporte para muchos idiomas y plataformas. También lo uso para mis bibliotecas Swift Cocoapod.

En este tutorial, proporciono pasos para integrar Travis CI en su repositorio público de Github.

Pasos para integrar el repositorio público de Ruby on Rails Github con Travis CI.

- Habilite Travis desde la configuración de su repositorio → Integración y servicios → servicios

- Vaya a la consola de Travis desde el enlace allí. Si no tiene una cuenta de Travis, cree una y enumerará todos sus repositorios.
- Habilite el repositorio que desea integrar presionando el interruptor al lado.
- Ahora, cree un `.travis.yml` archivo similar a este:

```
language: ruby

rvm:

  - "2.1.3"

env:

  - DB=postgresql

script:

  - RAILS_ENV=test bundle exec rake db:create db:migrate --trace

  - bundle exec rspec test/

before_script:

  - cp config/database.travis.yml config/database.yml

  - mysql -e 'create database mydb_test'

bundler_args: --binstubs=./bundler_stubs
```

- El archivo anterior debe agregarse al directorio raíz de su repositorio, es decir, de nivel superior.
- Agrega un `config/database.travis.yml`

```
default: &default

adapter: postgresql

test:

  <<: *default

database: mydb_test

username: travis

encoding: utf8
```

Ahora presione los cambios y deje que cree la primera compilación.

Terminando

Ahora, cada vez que envíe cambios a la rama maestra de su repositorio (puede especificar una rama diferente a través de la configuración en la configuración del repositorio de Travis), las pruebas de su unidad se ejecutarán automáticamente y cualquier falla se enviará por correo a su cuenta de correo electrónico adjunta de Github.