ブラウザが自動で動くSeleniumを使ってみました
～Ruby編～

自己紹介

Name : さかもと そう
坂本 想

Company : ファーエンドテクノロジー株式会社
（入社２年目）

ネットワーク、AWS、Rubyなどを勉強中

Seleniumとは？

seleniumとは？

ブラウザのオートメーションツール

Seleniumとの出会い

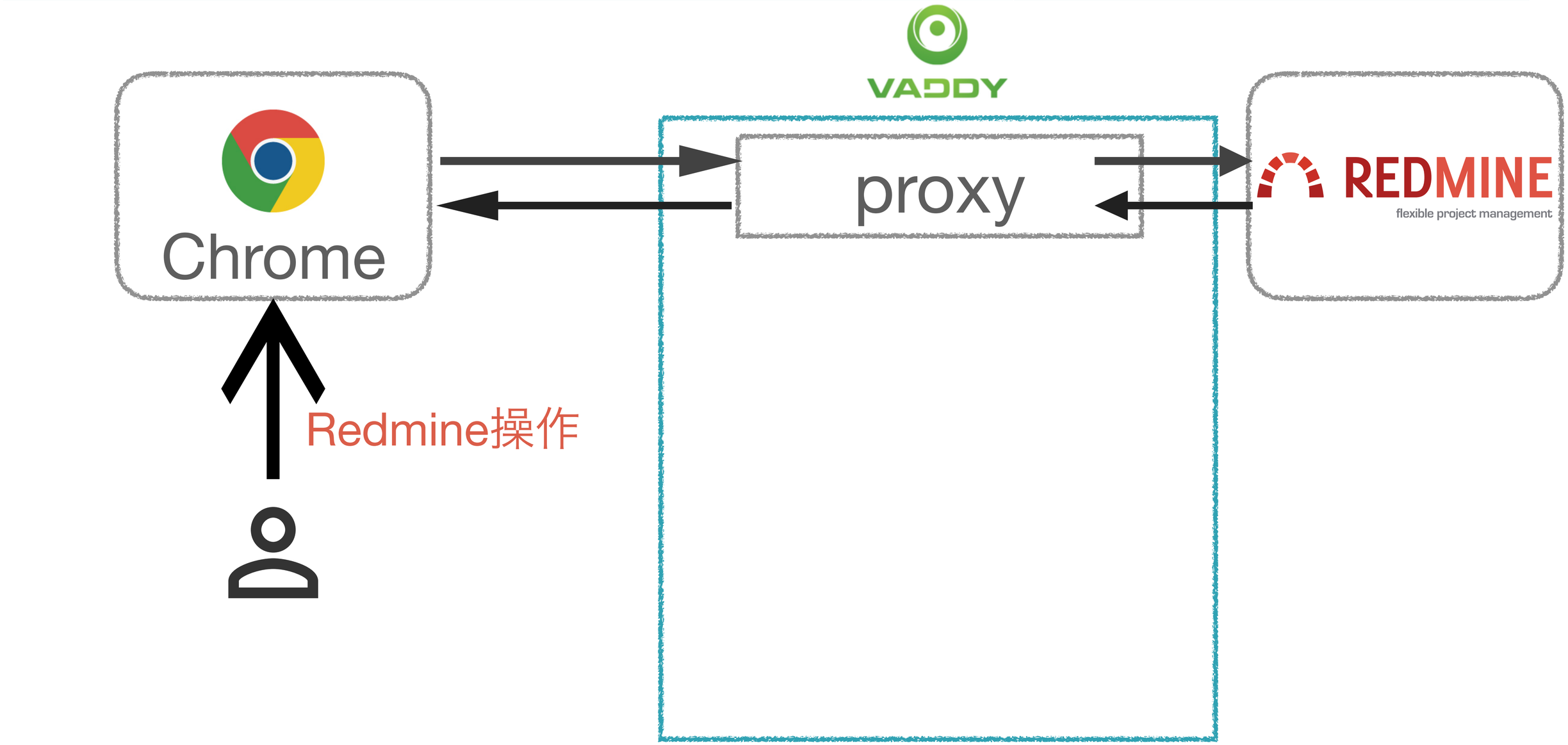
seleniumとの出会い

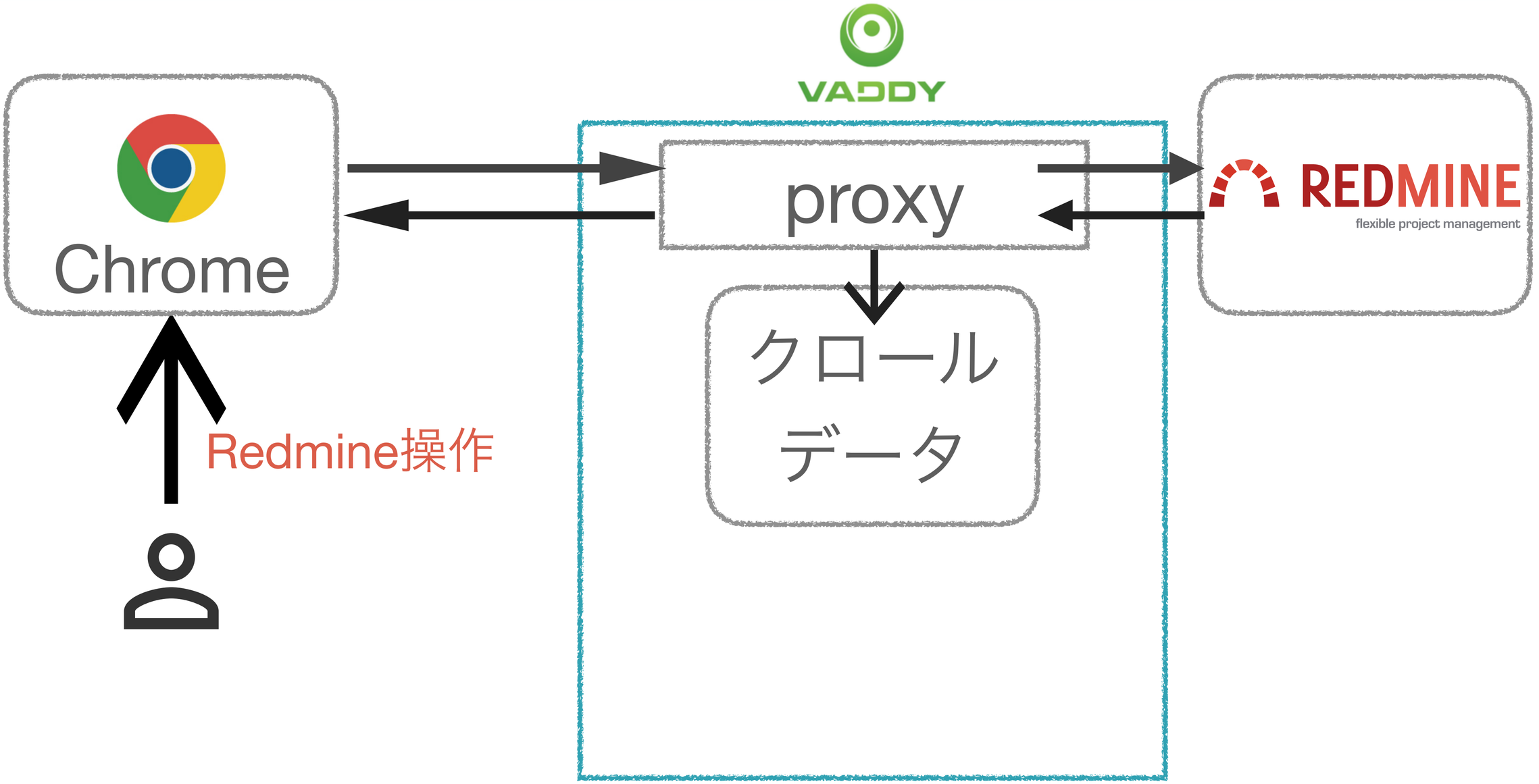
安心・安全

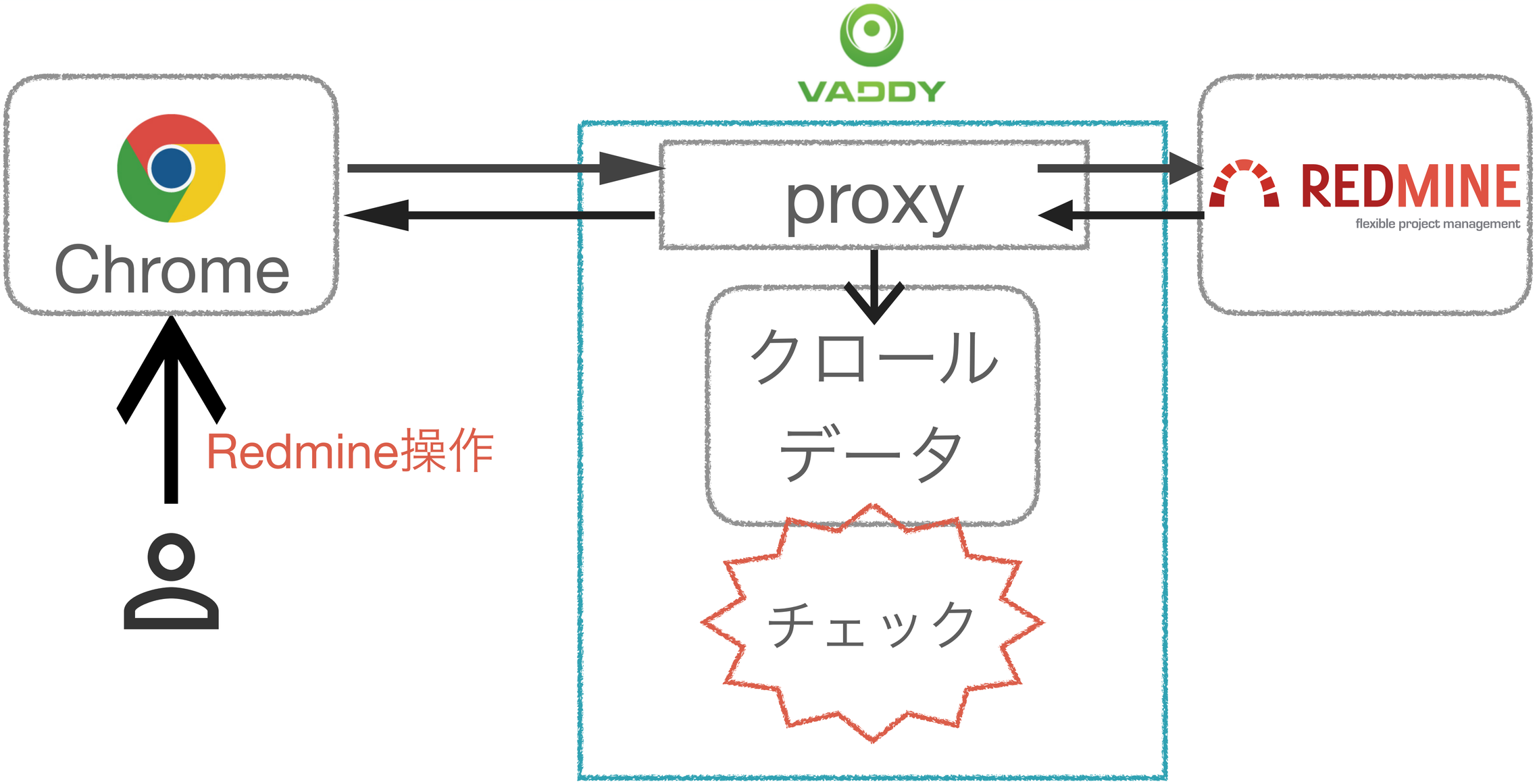
×

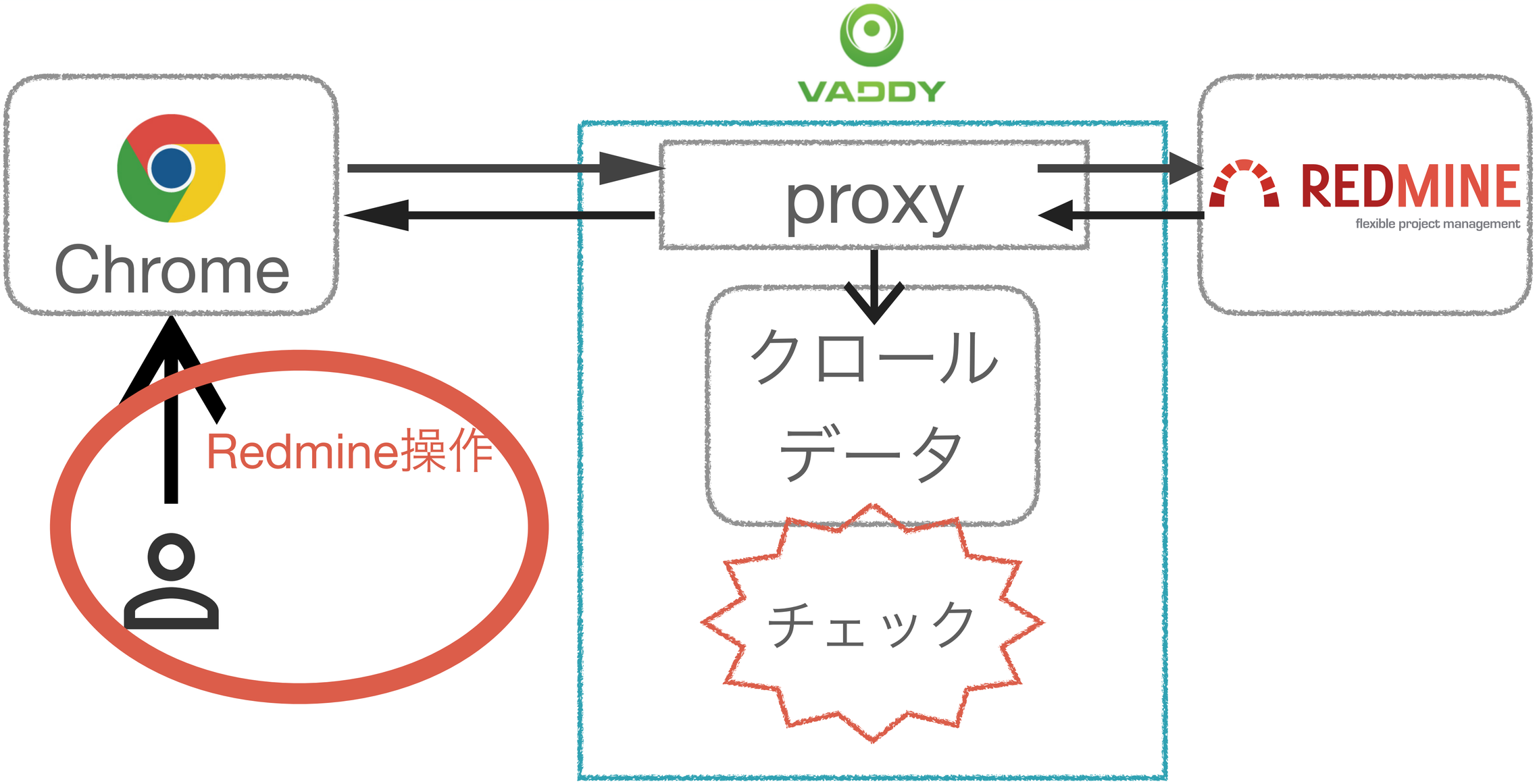
セキュリティチェック

セキュリティチェック？


VADDY
Chrome
proxy
REDMINE
flexible project management
Redmine操作

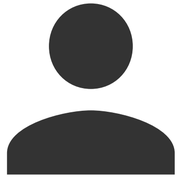

VADDY
Chrome
proxy
REDMINE
flexible project management
クロール
データ
Redmine操作


VADDY
Chrome
proxy
REDMINE
flexible project management
クロール
データ
チェック
Redmine操作

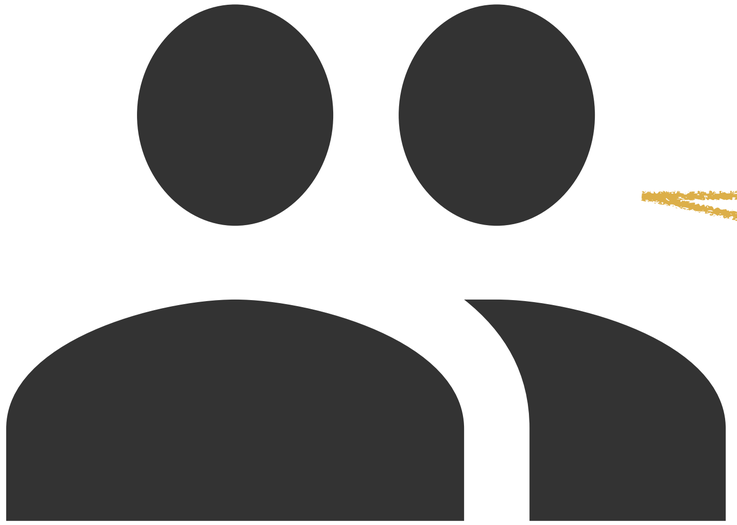

VADDY
Chrome
proxy
REDMINE
flexible project management
クロール
データ
チェック
Redmine操作

セキュリティチェック？

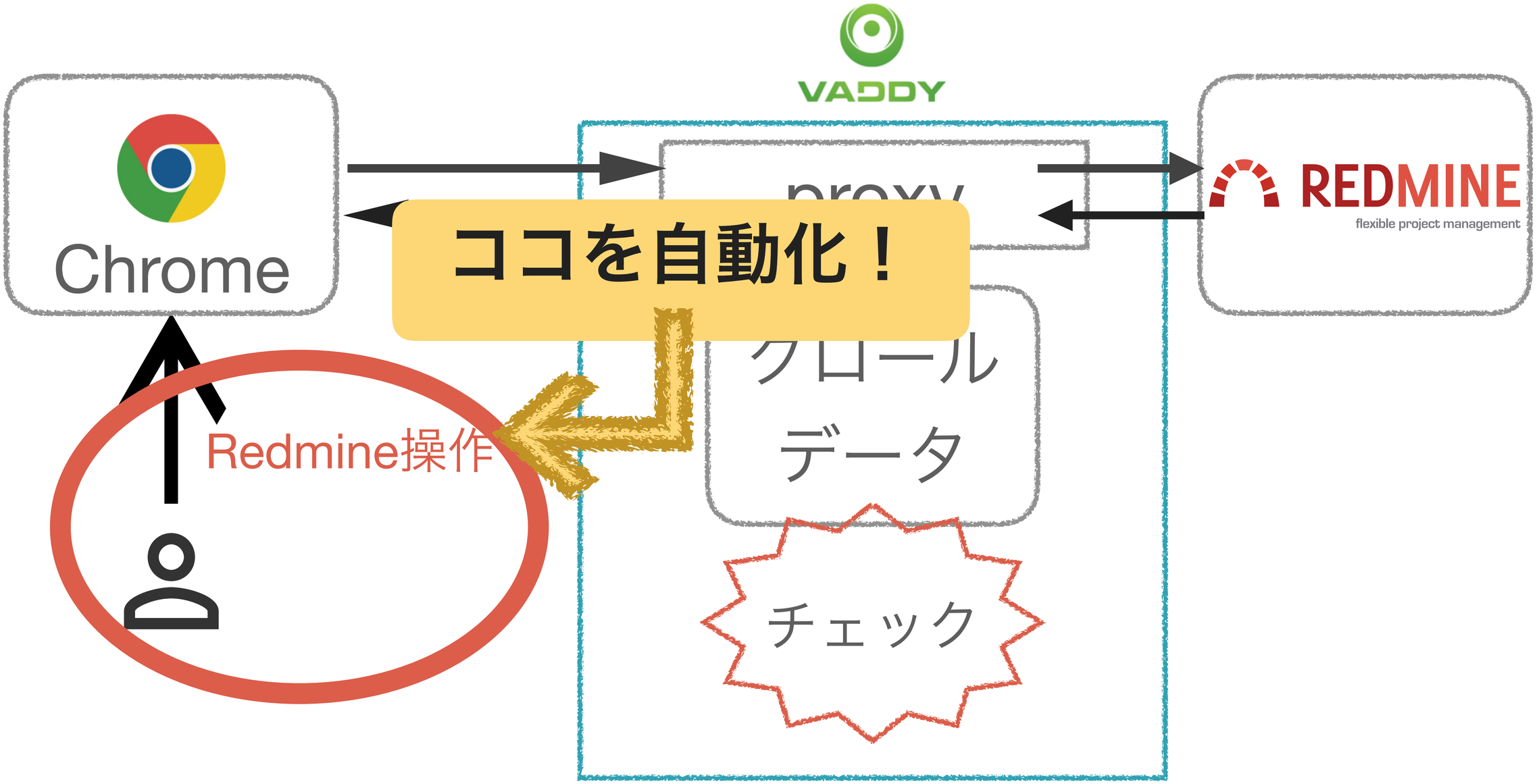
セキュリティチェック？



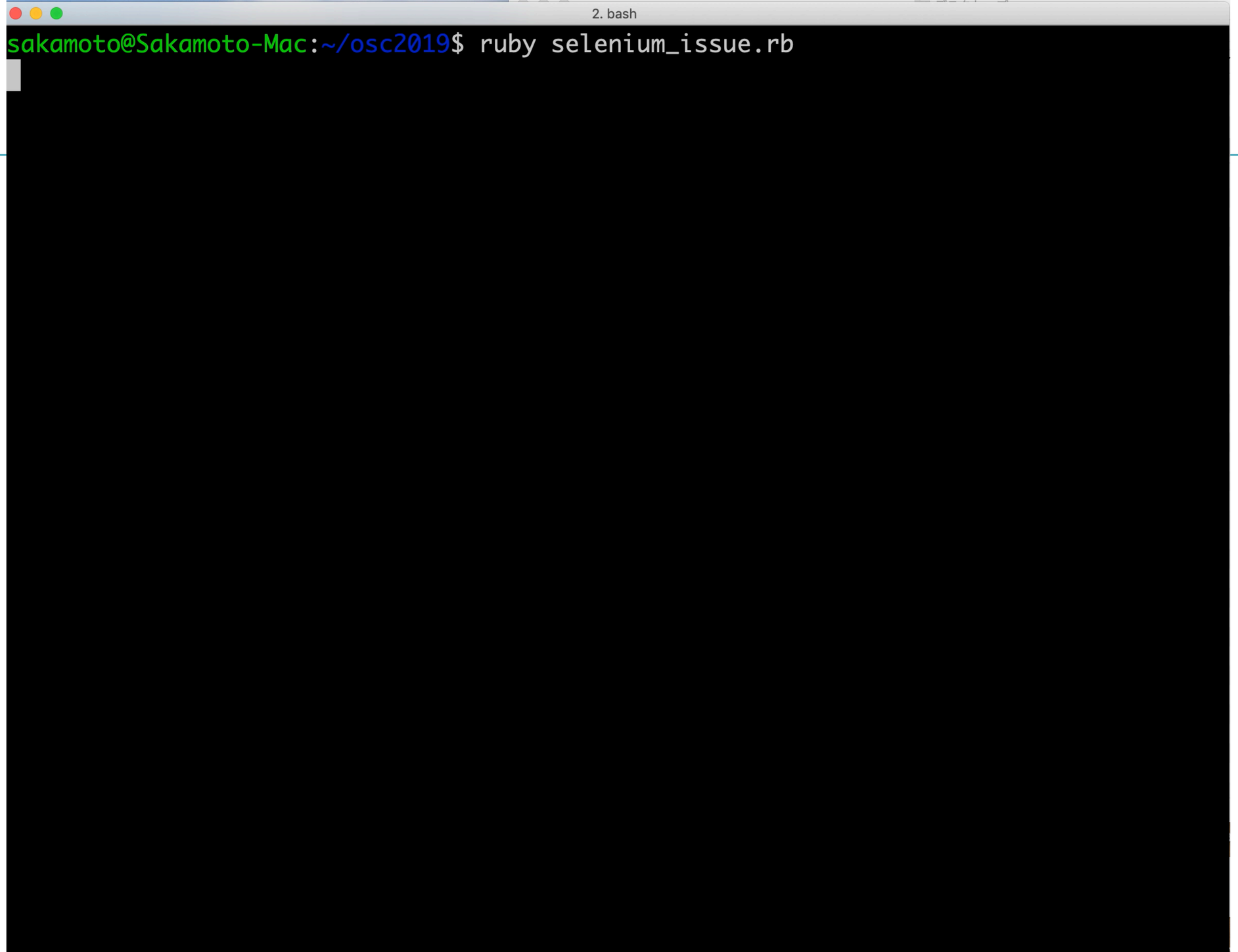
自動化できるといいね



Seleniumを使ってRedmine
クロールデータ作成を
自動化しよう


VADDY
Chrome
proxy
クロール
データ
チェック
REDMINE
flexible project management
ココを自動化！
Redmine操作

Seleniumで動くRedmine


2. bash
sakamoto@Sakamoto-Mac:~/osc2019$ ruby selenium_issue.rb

seleniumで動くredmine

```
require 'selenium-webdriver'

1  driver = Selenium::WebDriver.for :chrome

   # redmineにログイン
2  driver.navigate.to('http://vaddy-redmine.farend.ne.jp/redmine/login')
   driver.find_element(:id, 'username').send_keys 'admin'
   driver.find_element(:id, 'password').send_keys 'admin'
   driver.find_element(:id, 'login-submit').click
   # 新しいチケット作成画面へ移動
   driver.navigate.to('http://vaddy-redmine.farend.ne.jp/redmine/projects/ecookbook/issues/new')

   # 新しいチケット(ticket1)を作成
3  driver.find_element(:id, 'issue_subject').send_keys 'ticket1'
   driver.find_element(:id, 'issue_description').send_keys 'ticket1'
4  driver.find_element(:xpath, '//*[@id="issue-form"]/input[3]').click

   driver.quit
```

seleniumで動くredmine

```
require 'selenium-webdriver'

driver = Selenium::WebDriver.for :chrome

# redmineにログイン
driver.navigate.to('http://vaddy-redmine.farend.ne.jp/redmine/login')
driver.find_element(:id, 'username').send_keys 'admin'
driver.find_element(:id, 'password').send_keys 'admin'
driver.find_element(:id, 'login-submit').click
# 新しいチケット作成画面へ移動
driver.navigate.to('http://vaddy-redmine.farend.ne.jp/redmine/projects/ecookbook/issues/new')

# 新しいチケット(ticket1)を作成
driver.find_element(:id, 'issue_subject').send_keys 'ticket1'
driver.find_element(:id, 'issue_description').send_keys 'ticket1'
driver.find_element(:xpath, '//*[@id="issue-form"]/input[3]').click

driver.quit
```

1

2

3

4

seleniumで動くredmine

❶

ブラウザ起動

```
driver = Selenium::WebDriver.for :chrome
```

seleniumで動くredmine

❶

ブラウザ起動

```
driver = Selenium::WebDriver.for :chrome
```

:firefox

でもOK

seleniumで動くredmine

```
require 'selenium-webdriver'

driver = Selenium::WebDriver.for :chrome

# redmineにログイン
driver.navigate.to('http://vaddy-redmine.farend.ne.jp/redmine/login')
driver.find_element(:id, 'username').send_keys 'admin'
driver.find_element(:id, 'password').send_keys 'admin'
driver.find_element(:id, 'login-submit').click
# 新しいチケット作成画面へ移動
driver.navigate.to('http://vaddy-redmine.farend.ne.jp/redmine/projects/ecookbook/issues/new')

# 新しいチケット(ticket1)を作成
driver.find_element(:id, 'issue_subject').send_keys 'ticket1'
driver.find_element(:id, 'issue_description').send_keys 'ticket1'
driver.find_element(:xpath, '//*[@id="issue-form"]/input[3]').click

driver.quit
```

1

②

3

4

seleniumで動くredmine

② ページ遷移

指定したURLのページに
アクセスできる

```
driver.navigate.to('http://vaddy-redmine.farend.ne.jp/redmine/login')
```

seleniumで動くredmine

```
require 'selenium-webdriver'

driver = Selenium::WebDriver.for :chrome

# redmineにログイン
driver.navigate.to('http://vaddy-redmine.farend.ne.jp/redmine/login')
driver.find_element(:id, 'username').send_keys 'admin'
driver.find_element(:id, 'password').send_keys 'admin'
driver.find_element(:id, 'login-submit').click
# 新しいチケット作成画面へ移動
driver.navigate.to('http://vaddy-redmine.farend.ne.jp/redmine/projects/ecookbook/issues/new')

# 新しいチケット(ticket1)を作成
driver.find_element(:id, 'issue_subject').send_keys 'ticket1'
driver.find_element(:id, 'issue_description').send_keys 'ticket1'
driver.find_element(:xpath, '//*[@id="issue-form"]/input[3]').click

driver.quit
```

1
2
3
4

seleniumで動くredmine

③

要素取得

```
driver.find_element(:id, 'issue_subject').send_keys 'ticket1'
```

要素を指定

seleniumで動くredmine

③

:name, :xpath

でもOK

要素取得

```
driver.find_element(:id, 'issue_subject').send_keys 'ticket1'
```

要素を指定

seleniumで動くredmine

③

テキスト入力

```
driver.find_element(:id, 'issue_subject').send_keys 'ticket1'
```

テキストの内容

seleniumで動くredmine

```
require 'selenium-webdriver'

driver = Selenium::WebDriver.for :chrome

# redmineにログイン
driver.navigate.to('http://vaddy-redmine.farend.ne.jp/redmine/login')
driver.find_element(:id, 'username').send_keys 'admin'
driver.find_element(:id, 'password').send_keys 'admin'
driver.find_element(:id, 'login-submit').click
# 新しいチケット作成画面へ移動
driver.navigate.to('http://vaddy-redmine.farend.ne.jp/redmine/projects/ecookbook/issues/new')

# 新しいチケット(ticket1)を作成
driver.find_element(:id, 'issue_subject').send_keys 'ticket1'
driver.find_element(:id, 'issue_description').send_keys 'ticket1'
driver.find_element(:xpath, '//*[@id="issue-form"]/input[3]').click

driver.quit
```

1

2

3

4

seleniumで動くredmine

④

クリック

```
driver.find_element(:id, 'login-submit').click
```

クリック

「こんなこともできるの？」と思ったこと

「こんなこともできるの？」と思ったこと

全選択

待機処理

~応用編~

右クリック

「こんなこともできるの？」と思ったこと

全選択

Notes

Edit Preview B I U S C H1 H2 H3 pre

```
def test
  puts "test"
end
```

「こんなこともできるの？」と思ったこと

全選択

Mac :　　command + a

Windows :　　control + a

「こんなこともできるの？」と思ったこと

全選択

使うメソッドは?

A send_keys

「こんなこともできるの？」と思ったこと

全選択

他にも、、
・カーソルを先頭に移動
・ファイル添付

使うメソッドは?

A send_keys

「こんなこともできるの？」と思ったこと

全選択

```
driver.find_element(:id, 'issue_notes').send_keys(:command, 'a')
```

「こんなこともできるの？」と思ったこと

全選択

macの場合

```
driver.find_element(:id, 'issue_notes').send_keys(:command, 'a')
```

「こんなこともできるの？」と思ったこと

全選択

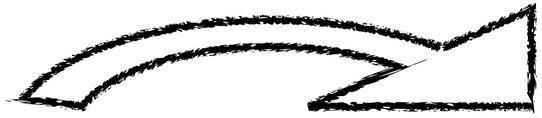
```
driver.find_element(:id, 'issue_notes').send_keys(:command, 'a')
```

windowsの場合　:control

「こんなこともできるの？」と思ったこと

待機処理

添付



REDMINE

flexible project management

「こんなこともできるの？」と思ったこと

待機処理

「こんなこともできるの？」と思ったこと

待機処理

```
sleep 5
```

「こんなこともできるの？」と思ったこと

待機処理

「こんなこともできるの？」と思ったこと

待機処理

waitを使う

```
# waitに50秒のタイマーを持たせる
wait = Selenium::WebDriver::Wait.new(:timeout => 50)

# 指定した要素が表示されるまで待つ
wait.until{driver.find_element(:id, 'test').displayed?}
# 指定した要素が表示されなくなるまで待つ
wait.until{!driver.find_element(:id, 'test').displayed?}
```

「こんなこともできるの？」と思ったこと

待機処理

waitを使う

```
# waitに50秒のタイマーを持たせる
wait = Selenium::WebDriver::Wait.new(:timeout => 50)

# 指定した要素が表示されるまで待つ
wait.until{driver.find_element(:id, 'test').displayed?}
# 指定した要素が表示されなくなるまで待つ
wait.until{!driver.find_element(:id, 'test').displayed?}
```

① ② ③

「こんなこともできるの？」と思ったこと

待機処理

waitを使う

```
# waitに50秒のタイマーを持たせる
wait = Selenium::WebDriver::Wait.new(:timeout => 50)

# 指定した要素が表示されるまで待つ
wait.until{driver.find_element(:id, 'test').displayed?}
# 指定した要素が表示されなくなるまで待つ
wait.until{!driver.find_element(:id, 'test').displayed?}
```

「こんなこともできるの？」と思ったこと

待機処理

waitを使う

```
# waitに50秒のタイマーを持たせる
wait = Selenium::WebDriver::Wait.new(:timeout => 50)

# 指定した要素が表示されるまで待つ
wait.until{driver.find_element(:id, 'test').displayed?}
# 指定した要素が表示されなくなるまで待つ
wait.until{!driver.find_element(:id, 'test').displayed?}
```

1

2

3

「こんなこともできるの？」と思ったこと

右クリック ~応用編~

右クリックのメソッド見当たらなかった、、

複数のメソッド組み合わせ ☺

「こんなこともできるの？」と思ったこと

右クリック ~応用編~

macの場合

control +クリック

＝

右クリック

windowsの場合

「Shift」＋「F10」

らしいです！

「こんなこともできるの？」と思ったこと

右クリック ~応用編~

使うメソッドは?

A click、key_down、key_up

「こんなこともできるの？」と思ったこと

右クリック ~応用編~

```
driver.action.key_down(:control).click(driver.find_element(:id, 'test'))
  .key_up(:control).perform
```

key_down . . . キーボードを押下した状態のままにする

「こんなこともできるの？」と思ったこと

右クリック ~応用編~

```
driver.action.key_down(:control).click(driver.find_element(:id, 'test'))
  .key_up(:control).perform
```

key_down . . . キーボードを押下した状態のままにする

key_up . . . 押下したキーボードを離す

Seleniumを使ってみて

seleniumを使ってみて

すごく難しそう

全選択、右クリックなどの
細かい操作はできなさそう

エラーばっか出そう

seleniumを使ってみて

すごく難しそう

全選択、右クリックなどの
細かい操作はできなさそう

エラーはよく出そう

使い方は簡単

細かい操作も
できる

情報が多いため
すぐ解決できる

seleniumを使ってみて

すごく難しそう

全選択、右クリックなどの
細かい操作はできなさそう

エラーはよく出そう

画面が動いて
楽しい

使い方は簡単

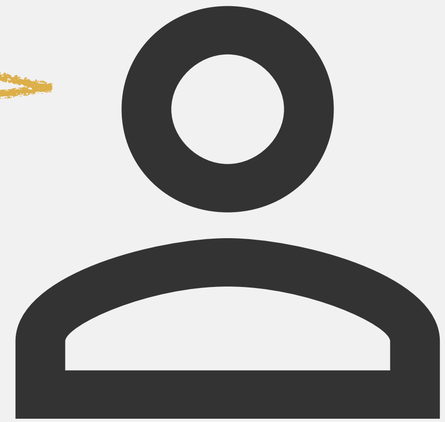
細かい操作も
できる

情報が多いため
すぐ解決できる

seleniumを使ってみて

こんなメソッドもある

こんなこともできる！



ご清聴ありがとうございました。