Web testing at Corporama

30 / 11 / 2012

Nicolas Thauvin
<nicolas@corporama.com>
Corporama CTO
http://corporama.com

Agenda

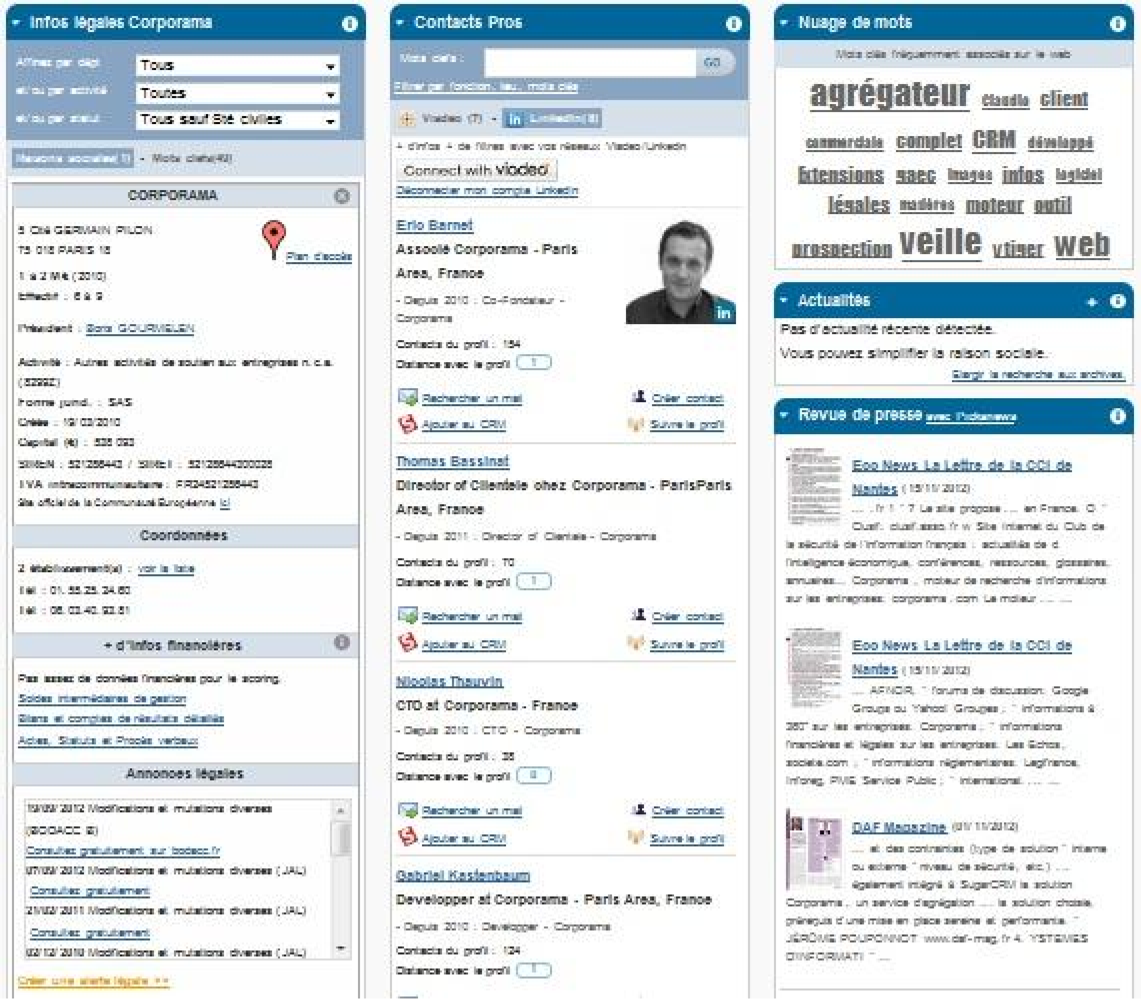
1. Why GUI tests / the needs
2. Initial version
3. Current version
4. Demo
5. Conclusion / what's next

Corporama
IMPROVE YOUR BUSINESS

Why GUI tests / The needs

- A lot of 'widgets'
- User specific
- External sources
- DB/CPU intensive
 (600k visits/month)
- A lot of Ajax

- Things behind the scene


Infos légales Corporama

Affiner par dépt : Tous
et/ou par activité : Toutes
et/ou par statut : Tous sauf Sté civiles

Raisons sociales(1) - Mots clefs(40)

CORPORAMA

3 Cité GERMAIN PILON
75 018 PARIS 18
Plan d'accès

1 à 2 M€ (2010)
Effectif : 6 à 9

Président : Boris GOURMELEN

Activité : Autres activités de soutien aux entreprises n.c.a. (8299Z)
Forme jurid. : SAS
Créée : 19/03/2010
Capital (€) : 535 093
SIREN : 521288443 / SIRET : 52128844300028
TVA intracommunautaire : FR24521288443
Site officiel de la Communauté Européenne ici

Coordonnées

2 établissement(s) : voir la liste
Tél : 01. 55.25.24.60
Tél : 06.03.40.93.81

+ d'infos financières

Pas assez de données financières pour le scoring.
Soldes intermédiaires de gestion
Bilans et comptes de résultats détaillés
Actes, Statuts et Procès verbaux

Annonces légales

10/09/2012 Modifications et mutations diverses (BODACC B)
Consultez gratuitement sur bodacc.fr
07/09/2012 Modifications et mutations diverses (JAL)
Consultez gratuitement
21/02/2011 Modifications et mutations diverses (JAL)
Consultez gratuitement
02/12/2010 Modifications et mutations diverses (JAL)

Créer une alerte légale >>

Contacts Pros

Mots clefs : GO
Filtrer par fonction, lieu, mots clés

Viadeo (7) - LinkedIn (8)
+ d'infos + de filtres avec vos réseaux Viadeo/LinkedIn
Connect with viadeo
Déconnecter mon compte LinkedIn

Eric Barnet
Associé Corporama - Paris Area, France
- Depuis 2010 : Co-Fondateur - Corporama
Contacts du profil : 184
Distance avec le profil 1
Rechercher un mail
Créer contact
Ajouter au CRM
Suivre le profil

Thomas Bassinet
Director of Clientele chez Corporama - ParisParis Area, France
- Depuis 2011 : Director of Clientele - Corporama
Contacts du profil : 70
Distance avec le profil 1
Rechercher un mail
Créer contact
Ajouter au CRM
Suivre le profil

Nicolas Thauvin
CTO at Corporama - France
- Depuis 2010 : CTO - Corporama
Contacts du profil : 38
Distance avec le profil 0
Rechercher un mail
Créer contact
Ajouter au CRM
Suivre le profil

Gabriel Kastenbaum
Developper at Corporama - Paris Area, France
- Depuis 2010 : Developper - Corporama
Contacts du profil : 124
Distance avec le profil 1

Nuage de mots

Mots clés fréquemment associés sur le web

agrégateur claudia client commerciale complet CRM développé extensions gaec images infos logiciel légales matières moteur outil prospection veille vtiger web

Actualités

Pas d'actualité récente détectée.
Vous pouvez simplifier la raison sociale.
Elargir la recherche aux archives.

Revue de presse avec Pickanews

Eco News La Lettre de la CCI de Nantes (15/11/2012)
... .fr 1 " 7 Le site propose ... en France. O " Clusif: clusif.asso.fr w Site Internet du Club de la sécurité de l'information français : actualités de d l'intelligence économique, conférences, ressources, glossaires, annuaires... Corporama , moteur de recherche d'informations sur les entreprises: corporama .com Le moteur ... ...

Eco News La Lettre de la CCI de Nantes (15/11/2012)
... AFNOR, " forums de discussion: Google Groups ou Yahoo! Groupes ; " informations à 360° sur les entreprises: Corporama ; " informations financières et légales sur les entreprises: Les Echos, societe.com ; " informations réglementaires: Legifrance, Inforeg, PME Service Public ; " international ... ...

DAF Magazine (01/11/2012)
... et des contraintes (type de solution " interne ou externe " niveau de sécurité, etc.) ... également intégré à SugarCRM la solution Corporama , un service d'agrégation ... la solution choisie, prérequis d'une mise en place sereine et performante. " JÉRÔME POUPONNOT www.daf-mag.fr 4. YSTEMES D'INFORMATI " ...

Features of our GUI tests system (version 1.Myriad)

- A GUI test = a set of actions in a browser, the automated way.

- We want to test Ajax, so we need to control browsers.

- We use Selenium and https://github.com/charpi/erl_selenium (old RC API, not WebDriver).

- An erlang module per feature to be tested. Automated detection with *_gui suffix

Features of our GUI tests system (version 1.Myriad)

What *gui_tests.sh "<tests to launch>"* does:

1. Define a few variables for the tests (to be read with os:getenv/1) : Host, Port, browser to be used...
2. Compile code, restart yaws test node
3. Start Selenium in a VNC instance or on display (Debug) using a custom profile
4. Fill database using our production import scripts
5. Create tests users (one per offer)
6. Start tests (sequencially) with subsystems (like fake SMTP server, mock_internet)

Intercept external calls : data:api/2

* Intermediate layer between code and data (ie: external store)

* eunit tests declare their own data_fun with expected clauses

```
api (http_request, {Method, URL, Headers, Body, Timeout, Options}) ->
...
{Host, Port, Path} =
    case application:get_env(www, http_proxy) of
        {ok, {Proxy_host, Proxy_port}} -> {Proxy_host, Proxy_port, URL};
        _ -> ...
    end,
lhttpc:request(Host, Port, ....).
```

We mock the Internet

mock_internet is a process that runs as a proxy and matches the longest URL prefix in an ETS table -> we can pass the tests without an internet access

the *_gui:mocks/1 Callback :

```
mocks () ->
 [{"crm.zoho.com/crm", "<html><body>OK</body></html>"},
  {"crm.zoho.com/crm/WebToLeadForm##actionType=social_pro12345a",
  fun () ->
   someone ! got_zoho_request,
   mock:http_reply("../fxt/zoho_reply.html")
  end}].
```

GUI test sample

Sample from social_pro_gui.erl:

```
test_not_logged(Session) ->
    ok = gui_tests:logout(Session),
    ok = gui_tests:search(Session, "Apple"),
    Xpath = "//div[@id='social_pro']/div/b/text()",
    Text = "Tous les profils Viadeo et Linkedin"
        " de la société à filtrer et exporter",
    gui_tests:check_text(Session, Xpath, Text),
    Teaser_x = "//a[@id='social_pro-teaser']",
    {ok, none} = selenium:cmd(Session, click, [Teaser_x]),
    Teaser_png_x = "//img[@src='/images/social_pro_teaser.png']",
    {ok, none} = selenium:cmd(Session, waitForElementPresent, [Teaser_png_x]),
    ?assertEqual(1, gui_tests:close_dialog_boxes(Session)).
```

What's wrong with version 1

- Selenium (old RC) is slow, as it relies on a JS interface.

- In some browsers, the JS *itself* is slow (Ajax in IE..., various initializations).

- Very long time to start a browser session

- Duration (**50 tests**): **20min**.

- Incompatible with a reactive continuous integration system and a growing test set.

- Order matters. Tests pass when user A is used in test 1 then in test 2. Not test 2 then test 1

- Things to optimize in the tests sequence

- Some random timeouts in Ajax calls. Use *waitForElementPresent, waitForTextPresent* & al.

Version 2

- Make it distributed (Erlang's way !)

A new callback function: *index/0*

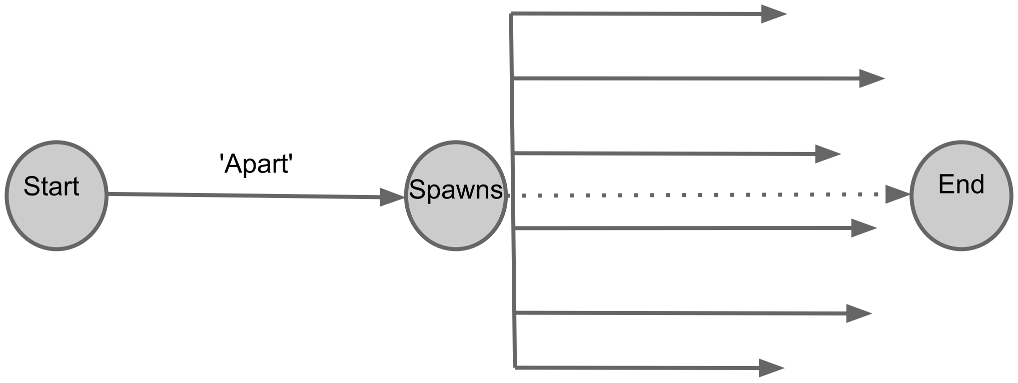
```
index () ->
     [{test_coupon_from_freemium_then_renew, [{user, "freemium"}]},
      {test_error, [{user, anonymous}, apart]},
      {test_offer_after_trial, [{user, anonymous}]},
      {test_defered, [{user, "freemium"}, {search, "Apple"}]}].
```

- One user per test (or anymous).
- DB creation on GUI tests startup + automatic login at the beginning of test. Prevents bad profile reutilisation
- selenium_pool to (re)allocate sessions (similar to Selenium Grid).
- selenium_pool can also be used during development in an erlang Shell.

Version 2 : queues

Test queues : A single "apart" then $((`grep -c vendor_id /proc/cpuinfo`))" concurrent processes

* A browser instance per queue (one per visible processor)
* Record test durations in a DETS table. Used for next run distribution order



Version 2 : debug

* Distributed tests often mean 'messy log files' or one log per test

* We use the messy one, tagged with the queue Pid (easier to spot interactions)

* When a test fails, it generates a screenshot of the browser view with the test name as file name

* export Debug=true :

 - Runs browsers on current X server

 - Keep mock_internet running at the end

Good enough ?

Latest duration: 9min

(including 164 GUI tests)

Conclusion / What's next

- Good speed up and catches major regressions
- On our staging server, GUI tests act as a load tester (while running eunits in parallel)
- Selenium approach is ok for functional testing, but is not efficient to spot browser-specific bugs
  * bugs are likely to be catched by JS Lint or similar
  * CSS / layout issues are very hard to detect (screenshot comparison tend to result in false positives and easily misses real problems)
- We may release parts of our code on github... yet the system is built-in for Corporama use

Corporama
IMPROVE YOUR BUSINESS

Merci !

Questions ?