

Publik - Development #22137

documentation en ligne : créer un outil de capture d'écran automatisé

27 février 2018 12:36 - Anonyme

Statut:	Nouveau	Début:	27 février 2018
Priorité:	Normal	Echéance:	
Assigné à:	Paul Marillonnet	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Club:	Non
Patch proposed:	Non		
Planning:			

Description

Suite à une discussion mail:
Pierre demandait:

On ne met (presque) pas de captures dans la doc parce que c'est trop pénible à maintenir. Vouloir mettre des captures dans la doc nous a juste empêché d'en avoir une pendant longtemps.

Pourtant, si on veut faire des (auto)formations en ligne, il faut des captures dans nos contenus.

Et donc je réfléchis à ce qu'on peut automatiser. Je vois des trucs genre <https://blog.codeship.com/automating-screenshots-in-documentation/> ou ça <https://github.com/aside-ufba/guide-automator> mais comme je suis pas sûr que ce soit une bonne voie à suivre, je préfère revenir à mon besoin.

Mon besoin c'est de pouvoir indiquer dans combo (puisque c'est notre outil de doc) un truc du genre `{{capture:http://entrouvert.com}}` et hop j'ai une capture de la page en question qui est mise à jour périodiquement sans que je m'en soucie.

Évidemment ça pose plein de problèmes :

- L'authentification pour les pages qui en demande une (mais on pourrait avoir ça en dur dans le script et n'utiliser que des sites où cette authent fonctionne)
- Une taille (hauteur) hétérogène des captures si on capture l'ensemble du contenu d'une page. Et ben tant pis, c'est mieux que rien, au rédacteur de ne pas faire les captures (de listings par exemple...) sur des pages dont il sait qu'elles pourraient s'allonger drastiquement avec le temps...
- La capture de parties inutiles (la capture de l'ensemble du contenu d'une page peut donner une capture de 800 px de haut alors qu'il y en a 200 d'utiles). On pourrait rajouter dans `{{capture:http://entrouvert.com}}` des coordonnées pour le coin supérieur gauche et le coin inférieur droit mais ça entraînera plus de maintenance, nécessairement.

Ce que j'imagine c'est dire que la doc suit, comme le code, les principes de l'intégration continue. Je fais l'effort (je l'ai fait ce matin d'ailleurs) de mettre la doc à jour à chaque mise à jour, tous les 15 jours.

Il n'y a pas de version de release Publik et pas de version de la doc actuellement, mais s'il devait y en avoir ils seraient en permanence identiques. Le principe des captures "automatiques" c'est bien que à chaque mise en prod, les captures soient refaites automatiquement (et bien sûr y aura parfois des soucis - pas facile à détecter - devant être corrigés à la main, c'est le jeu mon pov lulu et c'est pour ma pomme).

Victor:

Pour faciliter ça pourrait être une capture qui pointe directement vers

la page en question. plutôt que donner les coordonnées pour capturer l'image, le mec est rediriger directement vers la bonne interface. Ça suppose que le publik au bout est full acces libre en lecture (on que l'on communique en début de doc les identifiants. Pour le moment ça veut aussi dire full accès en modification et donc des choses cassées ponctuellement. Ça pose la question de la remise à zéro des données régulières et l'intégration des nouveaux contenus pour suivre la doc.

Bon c'est pas forcément plus simple...

Fred:

Mon besoin c'est de pouvoir indiquer dans combo (puisque c'est notre outil de doc) un truc du genre `{{capture:http://entrouvert.com}}` et hop j'ai une capture de la page en question qui est mise à jour périodiquement sans que je m'en soucie.

`{{capture:http://entrouvert.com}}`, mais on ne veut généralement pas une capture de toute la page, juste une partie de celle-ci, (source: nos notes de mise à jour).

Incidemment, cette restriction est un gros avantages, autant tout autour des choses bougent, autant concentrer la capture sur la zone pertinente améliore sa longévité.

[1] Je suis certain que Fred a déjà réfléchi sur le sujet par le passé et qu'il pourrait orienter les gens intéressés.

Pour le bandeau des écrans sur la page références du site Publik (<https://publik.entrouvert.com/static/img/screens.png>) j'avais tenté des bouts d'automatisation avec phantomjs et cie mais même pour ce cas a priori simple de captures "plein écran", j'ai plutôt abandonné l'idée.

(le code existe toujours dans le dépôt publik-website).

Pour revenir à l'automatisation des captures, ça s'inscrirait idéalement dans le cadre général de tests d'intégration, une fois qu'on aura des parcours complets réellement joués, on pourra indiquer qu'à telle ou telle étape on veut une capture, l'idée est de pouvoir ainsi détecter des changements inattendus mais rien n'empêcherait la documentation de puiser dans ces captures.

Historique

#1 - 27 février 2018 12:36 - Anonyme

- Tracker changé de Bug à Development

#2 - 27 février 2018 12:49 - Frédéric Péters

- Projet changé de Combo à Publik

(je déplace vers "publik", plus général, et qui contient les tickets concernant la documentation)

#3 - 27 février 2018 15:20 - Anonyme

Elias Showk a écrit :

Suite à une discussion mail:
Pierre demandait:

- La capture de parties inutiles (la capture de l'ensemble du contenu d'une page

peut donner une capture de 800 px de haut alors qu'il y en a 200 d'utiles). On pourrait rajouter dans `{{capture:http://entrouvert.com}}` des coordonnées pour le coin supérieur gauche et le coin inférieur droit mais ça entraînera plus de maintenance, nécessairement.

On remplace cette idée par le ciblage d'un élément HTML avec un sélecteur CSS standard

#4 - 12 mars 2018 11:36 - Anonyme

- *Sujet changé de documentation en ligne, créer un outil de capture d'écran automatisé à documentation en ligne : créer un outil de capture d'écran automatisé*

- *Assigné à mis à Anonyme*

#5 - 28 mars 2018 13:00 - Anonyme

- *Echéance 31 mars 2018 supprimé*

#6 - 28 mars 2018 13:00 - Anonyme (Notes privées)

J'avais mis comme date de livraison le 31/03, mais j'ai bien peur de devoir retirer cette promesse, avec d'autres priorités comme GNM

#7 - 04 juillet 2018 04:47 - Pierre Cros

Est-ce qu'on pourrait s'entendre sur une date et, dès à présent, sur une syntaxe à utiliser par le rédacteur de la doc dans combo pour générer une capture ?

#8 - 12 juillet 2018 08:58 - Frédéric Péters

Honnêtement, je ne nous vois pas créer un outil, ça implique trop. Pour moi ce qu'il faut c'est trouver un outil, il aura ses défauts et ses limites, on pourra faire avec ou y travailler.

Sans passer par l'étape de recherche, les choses ont peut-être évolué depuis, le seul truc raisonnable c'était robot framework, à nouveau pointé par <https://blog.codeship.com/automating-screenshots-in-documentation/> que tu reprends.

De là, restera la route de l'infrastructure, arriver à ce que ça ne soit pas trop loin du combo où s'édite la documentation, idées et discussions à avoir (par exemple si le code pour aller faire une capture n'était pas mêlé au texte mais dans une cellule dessous, pour ainsi ne pas avoir à risquer des choses avec ckeditor).

Bref, mon idée de chemin là-dessus :

- prendre robotframework (ça donne un outil immédiat, qui fonctionne en local pour avancer)
- poser les "scripts" robotframework dans ces cellules combo
- avoir de quoi les extraire, en donner l'accès à jenkins
- avoir jenkins lançant alors le robot framework, et rangeant de manière appropriée les captures
- avoir côté combo la récupération de ces captures (automatiquement, ou à la demande après validation, je ne sais pas, ici et ailleurs il reste quantité de questions)

#9 - 16 juillet 2018 11:46 - Paul Marillonnet

- *Assigné à changé de Anonyme à Paul Marillonnet*

#10 - 16 juillet 2018 14:07 - Christophe Siraut

Est-ce que la piste automatisation/amélioration de la prise de captures côté client a été suffisamment envisagée? Tiens l'outil Shutter semble vraiment pas mal pour combiner flexibilité et automatisation. <http://shutter-project.org/>

Autre piste il y a un plugin selenium pour firefox qui permet d'enregistrer/rejouer des sessions; ce serait utilisable tel quel avec un greffon de capture d'écran; voir si un greffon existe avec suffisamment de fonctionnalités.

Par rapport à la génération automatisée côté serveur, je ne comprends pas le besoin de définir programmatiquement une sous-zone de page: cette zone est susceptible d'évoluer et de bouger, et nécessaire d'adapter régulièrement le code en conséquence.

(désolé si mes questions ont déjà été traitées)

(Le besoin c'est bien 1. "un moyen graphique de dessiner le rectangle de coordonnées" 2. "voici une liste d'URL et une liste de coordonnées; fais-moi des captures d'écran"?)

#11 - 01 août 2018 13:40 - Paul Marillonnet

Sauf avis contraire, après lecture des avis de chacun, je commence avec robotframework (en dépit de son incapacité à capturer parfaitement des sous-éléments du DOM).

#12 - 01 août 2018 15:11 - Paul Marillonnet

Après un premier épiluchage de la documentation, j'ai l'impression qu'utiliser le binding python de selenium plutôt que la librairie native ScreenShot de RFW est **vraiment** plus simple -- et aussi plus adapté au besoin -- si on omet les problématiques de test évoquées par Frédéric.

Donc je vois deux options pour l'instant :

- soit je me creuse la tête pour avoir la balise personnalisée de template, intégrée dans combo, qui permet de mentionner directement l'URL de la page à capturer.
- soit j'écris un script utilisant le binding python-selenium, qui va bêtement, à intervalle régulier (défini dans un crontab), générer toutes les captures. Le script placera et nommera les captures de façon à ce que la documentation récupère la dernière version de ces captures.

J'avoue que l'option 1 me paraît un peu over-engineered, mais je me plante peut-être.

Dans tous les cas, c'est vrai que ce serait un pas de plus vers du contenu de documentation automatiquement mis à jour pour suivre le cycle de release Publik.

#13 - 01 août 2018 15:29 - Paul Marillonnet

Je vais essayer d'avoir une authentification sur mon Publik local avec selenium, pour des captures de pages à des URI restreintes (le backoffice, par exemple).

Edit: Ah oui, et j'essaie aussi d'avoir la capture de sous-éléments du DOM.

#14 - 01 août 2018 16:09 - Paul Marillonnet

Apparemment on n'est pas les seuls à se poser la question des captures partielles [1].

Une solution, pas entièrement fonctionnelle, proposée dans [1], est :

- prendre une capture de la fenêtre active
- déterminer les coordonnées et la taille du sous-élément du DOM à capturer
- sauvegarder la capture tronquée aux dimensions et à l'emplacement du sous-élément.

Reste que dans l'idée, le driver ouvre le navigateur dans une fenêtre de taille définie. Et donc, il faut prendre en compte le scrolling de la page si l'on veut capture des sous-éléments.

Il faudrait peut-être procéder comme ça :

- déterminer l'emplacement et les dimensions du sous-élément à capturer
- scroller pour faire coïncider le coin haut-gauche de l'élément au coin haut gauche de la page visible
- itérer pour scroller la page visible de manière à capturer, en plusieurs morceaux, tout le sous-élément
- recoller les bouts :) (lib Pillow)
- sauvegarde l'image entière du sous-élément

[1] <https://stackoverflow.com/questions/15018372/how-to-take-partial-screenshot-with-selenium-webdriver-in-python>

#15 - 01 août 2018 17:09 - Benjamin Dauvergne

J'ai tout testé (notamment ces scripts) et sur ma sid je n'ai jamais réussi à les faire marcher (soit on arrive pas à prendre un screenshot complet, Chrome/Firefox, soit le scrolling de la fenêtre ne marche pas, Firefox, ce qui revient au même, même en faisant un screenshot partiel on arrive pas à mettre l'élément dans la fenêtre).

Si tu te lances là dedans, teste les différentes techniques et si tu arrives au but documente ici la liste des versions parce que pour moi d'une version à l'autre ça semble totalement changer.

#16 - 02 août 2018 12:03 - Paul Marillonnet

- Fichier *screenshot.png* ajouté
- Fichier *selenium_scroll.py* ajouté

Pour cette histoire de scrolling je pense avoir quelque chose qui fonctionne (le script en fichier joint). Je mets aussi une capture pour montrer le résultat.

Pour les versions c'est :

```
$ sudo apt search python-selenium
En train de trier... Fait
Recherche en texte intégral... Fait
python-selenium/unstable,testing,now 3.8.0+dfsg1-3 all [installé]
  Python bindings for Selenium
```

et

```
$ sudo apt search chromium-driver
En train de trier... Fait
```

Recherche en texte intégral... Fait
chromium-driver/unstable,testing,now 68.0.3440.75-2 amd64 [installé]
navigateur web - prise en charge de WebDriver

#17 - 02 août 2018 14:35 - Benjamin Dauvergne

Tant mieux si ça marche, bon maintenant il faudrait faire marcher ça avec robotframework (parce que c'est pas des copies d'écran de page statiques qu'on veut mais des copies après avoir déroulé un script dans l'application).

#18 - 08 août 2018 14:01 - Paul Marillonnet

Est-ce que les fonctions de navigation de selenium te paraissent insuffisantes pour cela ? (<http://selenium-python.readthedocs.io/navigating.html>)

#19 - 05 septembre 2018 21:04 - Benjamin Dauvergne

Oui tout me va, robotframework utilise selenium donc c'est possible, je dis juste que tu vas te faire chier à écrire du python là où robotframework fournit un DSL adapté et extensible, mais c'est toi qui vois.

#20 - 08 octobre 2018 12:54 - Frédéric Péters

Veille, <https://splinter.readthedocs.io/en/latest/>

#21 - 08 novembre 2018 17:13 - Paul Marillonnet

- *Club mis à Non*

Frédéric Péters a écrit :

Veille, <https://splinter.readthedocs.io/en/latest/>

Merci, plus pratique en effet, avec l'option `full=True` de la fonction de prise de capture. Plus besoin du code mentionné plus haut.

Pour cette histoire de parcours, ça me va de les détailler à la main une fois. Genre discuter avec Pierre des différents parcours, et ensuite écrire un fichier texte où ligne par ligne on aurait

```
<XPath de l'élément HTML> <Action à effectuer (clic, saisie clavier, capture)>
```

Ensuite un simple script basé sur la lib splinter qui prend un tel fichier en entrée, et qui produit les captures en reproduisant les parcours.

#22 - 12 novembre 2018 14:44 - Paul Marillonnet

Paul Marillonnet a écrit :

Genre discuter avec Pierre des différents parcours, et ensuite écrire un fichier texte [...].

Bon, je dois d'abord prouver que ça marche. Genre partir sur un parcours basique : s'authentifier en tant qu'agent dans le BO, aller dans la fabrique de formulaires, et prendre un capture de l'interface de la fabrique sans la barre latérale.

#23 - 12 novembre 2018 15:20 - Pierre Cros

J'ai l'impression que vous vous êtes focalisés sur une histoire de parcours (Selenium, Robotframework) qui ne correspond pas à ce que j'avais en tête. Je reviens sur ce que j'indiquais dans le descriptif initial du besoin :

Mon besoin c'est de pouvoir indiquer dans combo (puisque c'est notre outil de doc) un truc du genre `{{capture:http://entrouvert.com}}` et hop j'ai une capture de la page en question qui est mise à jour périodiquement sans que je m'en soucie.

Avec <https://splinter.readthedocs.io/en/latest/> ça deviendrait je crois `{{capture:XPath-Vers-UN-ELEMENT-A-CAPTURER}}` et ça veut dire que les rédacteurs de doc qui veulent des captures doivent apprendre à écrire des Xpaths, pourquoi pas.

Le boulot pour l'outil que doit développer Paul ce serait alors me semble-t-il :

- repérer les tags `{{capture:xpath}}` dans la doc
- faire les captures vers ces xpath périodiquement (en réglant les pbs éventuels d'authent)
- poser les captures dans les ressources combo

Et il y aurait un boulot à faire par je ne sais pas qui pour que les tags `{{capture:xpath}}` d'une cellule donnent, dans le rendu html, un ``.

Désolé si tout ça n'est pas faisable/réaliste mais c'est ce que j'avais en tête.

#24 - 12 novembre 2018 15:25 - Frédéric Péters

capture:http://entrouvert.com

Mais à partir du moment où il est question de backoffice il y a nécessairement un parcours pour se connecter.

#25 - 12 novembre 2018 15:35 - Pierre Cros

D'ac (et je mentionnais l'authent mais en me disant qu'il y avait peut-être moyen d'utiliser juste une URL pour ça, visiblement à tort). Ça reste quand même très réduit comme parcours : pointer vers la page de login, remplir login et mdp, valider.

#26 - 13 novembre 2018 11:55 - Benjamin Dauvergne

Pierre Cros a écrit :

D'ac (et je mentionnais l'authent mais en me disant qu'il y avait peut-être moyen d'utiliser juste une URL pour ça, visiblement à tort). Ça reste quand même très réduit comme parcours : pointer vers la page de login, remplir login et mdp, valider.

Ça suppose de pointer vers un site déjà initialisé (workflow, formulaires, pages combo, etc.), de ne remplir aucun champ, impossible de prendre une capture d'un truc créer pour l'occasion, ni d'une page qui ne s'afficherait que suite à un POST (ou une quelconque action en fait, genre une popup qui ne s'affiche que si on clique sur un truc).

#27 - 13 novembre 2018 12:06 - Pierre Cros

J'avais pas pensé aux popup (tant pis pour elle, il y en a peu, et y aura pas de captures les concernant), on aura aussi le problème sur certains listings ou indicateurs statistiques pour les lesquels on peut utiliser des paramètres qui n'apparaissent pas dans l'URL (même punition).

Mais pour le reste, oui, je pensais bien utiliser un site référence et qui bouge pas ou peu au niveau de son contenu : <https://matrik.test.entrouvert.org/> (sauf si vous avez mieux).

Et les rédacteurs de la doc devront aussi éviter d'être crétins en mettant des captures de trucs qui seraient rapidement caduques sur le site en question.

#28 - 14 novembre 2018 10:49 - Benjamin Dauvergne

Pierre Cros a écrit :

Et les rédacteurs de la doc devront aussi éviter d'être crétins en mettant des captures de trucs qui seraient rapidement caduques sur le site en question.

Parfait, l'important c'est de borner ce qui est attendu sinon on ne fermera jamais ce ticket, si on a une solution 20/80 en vue prenons la.

#29 - 15 novembre 2018 17:38 - Paul Marillonnet

Ma vision des choses pour avancer doucement :

- créer une balise de type inclusion_tag appelée capture dans combo/public/templatetags/combo.py
- écrire le gabarit correspondant, qui prenne un chemin vers une image et qui l'affiche
- intégrer splinter dans le code de la balise pour afficher la capture d'une page quelconque
- faire en sorte que l'URL de cette page quelconque soit un paramètre de la balise
- faire en sorte que l'élément XPath associé à une URL soit un paramètre d'appel de la balise
- gérer la connexion/déconnexion pour les pages à accès restreint

Et peut-être d'autres choses encore, une fois que tout ceci fonctionnera (mise en cache ? thumbnail ?).

#30 - 15 novembre 2018 17:48 - Frédéric Péters

- créer une balise de type inclusion_tag appelée capture dans combo/public/templatetags/combo.py

Nope, les cellules textes ne vont pas se mettre à interpréter du django qui serait tapé dedans; il faut faire avec le balisage HTML offert/éditable par ckeditor, on pourrait par exemple s'approprier longdesc (éditable dans l'onglet "avancé" d'une image). En mode basique, ça prendrait capture de l'URL mentionnée, en mode moins basique il y aurait <https://url/...#expression> xpath et l'expression serait utilisée pour pointer un élément. En mode parcours, il y aurait <https://dev.entrouvert.org/.../wiki/doc/parcours/> avec dans la page la description du parcours.

Une fois qu'on a ainsi de quoi marquer les pages avec l'info souhaitée, le reste c'est juste un script qui passe sur toutes les cellules texte, choppe les longdesc, en joue le contenu, pose l'image dans /media/..., modifie éventuellement l'attribut src de la balise, etc.

#31 - 15 novembre 2018 17:59 - Paul Marillonnet

Ok, au temps pour moi, je n'avais compris qu'on irait taper tout ça directement dans des cellules textes.

Fichiers

selenium_scroll.py	1,24 ko	02 août 2018	Paul Marillonnet
screenshot.png	791 ko	02 août 2018	Paul Marillonnet