

Spiderbot

API Documentation

June 21, 2015

Contents

Contents	1
1 Module model	2
1.1 Class PageResult	2
1.1.1 Class Variables	2
1.2 Class PageResultGoogle	2
1.2.1 Class Variables	2
1.3 Class Report	3
1.3.1 Class Variables	3
1.4 Class ReportGoogle	4
1.4.1 Class Variables	4
1.5 Class User	5
1.5.1 Class Variables	5
1.6 Class ReportRank	6
1.6.1 Class Variables	6
2 Module server	7
2.1 Functions	7
2.2 Variables	8
2.3 Class MainPage	8
2.3.1 Methods	8
2.4 Class login	8
2.4.1 Methods	9
2.5 Class logout	9
2.5.1 Methods	9
2.6 Class QueueValidation	9
2.6.1 Methods	10
2.7 Class Validation	10
2.7.1 Methods	11
2.8 Class Reports	11
2.8.1 Methods	11
2.9 Class ReportViewer	11
2.9.1 Methods	12
2.10 Class PageViewer	12
2.10.1 Methods	12
2.11 Class GetScanProgress	12
2.11.1 Methods	13

2.12	Class Rankings	13
2.12.1	Methods	13
2.13	Class GoogleValidation	13
2.13.1	Methods	13
2.14	Class GoogleReportViewer	14
2.14.1	Methods	14
2.15	Class GooglePageViewer	14
2.15.1	Methods	14
2.16	Class Users	14
2.16.1	Methods	15
2.17	Class EditUser	15
2.17.1	Methods	15
2.18	Class DeleteUser	15
2.18.1	Methods	16
2.19	Class CreateUser	16
2.19.1	Methods	16
2.20	Class DeleteReport	16
2.20.1	Methods	17
2.21	Class DeleteRankingReport	17
2.21.1	Methods	17
2.22	Class RankingReports	17
2.22.1	Methods	18
2.23	Class RankingReportViewer	18
2.23.1	Methods	18
2.24	Class CancelValidation	18
2.24.1	Methods	18
2.25	Class NotFound	19
2.25.1	Methods	19
3	Package spiderbot	20
3.1	Modules	20
3.2	Variables	20
4	Module spiderbot.parsers	21
4.1	Functions	21
4.2	Variables	23
5	Module spiderbot.validators	24
5.1	Functions	24
5.2	Variables	25

1 Module model

1.1 Class PageResult

google.appengine.ext.ndb.Model —
model.PageResult

Modelo de validación HTML, CSS, WCAG 2.0 o disponibilidad de una página web.

1.1.1 Class Variables

Name	Description
user	Usuario que ha realizado la validación Value: <code>ndb.StringProperty()</code>
validation_type	Tipo de validación, puede ser de tipo: <ul style="list-style-type: none"> • HTML • WCAG2-A • WCAG2-AA • CHECK AVAILABILITY Value: <code>ndb.StringProperty()</code>
url	URL validada Value: <code>ndb.StringProperty()</code>
content	Resultado de la validación con mensajes de error y advertencia Value: <code>ndb.TextProperty()</code>
state	Estado de la validación Value: <code>ndb.StringProperty()</code>
errors	Número de errores de la validación Value: <code>ndb.IntegerProperty()</code>
list_errors	Lista de errores de la validación Value: <code>ndb.TextProperty()</code>
number	Número de página para establecer un orden Value: <code>ndb.IntegerProperty()</code>

1.2 Class PageResultGoogle

google.appengine.ext.ndb.Model —
model.PageResultGoogle

Modelo de validación de Google para móviles de una página web.

1.2.1 Class Variables

Name	Description
user	Usuario que ha realizado la validación Value: <code>ndb.StringProperty()</code>
url	URL validada Value: <code>ndb.StringProperty()</code>
scoreUsability	Puntuación obtenida en el test de experiencia de usuario Value: <code>ndb.FloatProperty()</code>
scoreSpeed	Puntuación obtenida en el test de velocidad Value: <code>ndb.FloatProperty()</code>
content	Cadena JSON con el resultado de la validación estructurado Value: <code>ndb.TextProperty()</code>
number	Número de página para establecer un orden Value: <code>ndb.IntegerProperty()</code>

1.3 Class Report

google.appengine.ext.ndb.Model —
model.Report

Modelo de informe para validaciones HTML, WCAG 2.0 y disponibilidad de una web completa validada.

1.3.1 Class Variables

Name	Description
web	Página raíz Value: <code>ndb.StringProperty()</code>
validation_type	Tipo de validación. Puede ser: <ul style="list-style-type: none"> • HTML • WCAG2-A • WCAG2-AA • CHECK AVAILABILITY Value: <code>ndb.StringProperty()</code>
user	Usuario que ha realizado la validación Value: <code>ndb.StringProperty()</code>
onlyDomain	¿Es de sólo dominio? Value: <code>ndb.BooleanProperty()</code>
isRank	¿Es un informe de Ranking? Value: <code>ndb.BooleanProperty(default= False)</code>
results	Lista de páginas validadas en el informe Value: <code>ndb.LocalStructuredProperty(PageResult, repeated= True, k...)</code>
pages	Número de páginas Value: <code>ndb.IntegerProperty()</code>
error_pages	Número de páginas con algún error Value: <code>ndb.IntegerProperty()</code>

continued on next page

Name	Description
errors	Número total de errores Value: <code>ndb.IntegerProperty()</code>
list_errors	Lista de errores Value: <code>ndb.TextProperty()</code>
score	Puntuación obtenida Value: <code>ndb.FloatProperty()</code>
date	Fecha en la que se realizó la validación Value: <code>ndb.DateProperty(auto_now= True)</code>
time	Hora en la que se realizó la validación Value: <code>ndb.TimeProperty(auto_now= True)</code>

1.4 Class ReportGoogle

google.appengine.ext.ndb.Model —
model.ReportGoogle

Modelo de informe de Google para móviles de una web completa validada.

1.4.1 Class Variables

Name	Description
web	Página raíz Value: <code>ndb.StringProperty()</code>
validation_type	Tipo de validación. Sólo puede tomar el valor MOBILE, pero se deja este campo para posibles futuras ampliaciones. Value: <code>ndb.StringProperty()</code>
user	Usuario que ha realizado la validación Value: <code>ndb.StringProperty()</code>
onlyDomain	¿Es de sólo dominio? Value: <code>ndb.BooleanProperty()</code>
isRank	¿Es un informe de Ranking? Value: <code>ndb.BooleanProperty(default= False)</code>
results	Lista de páginas validadas en el informe Value: <code>ndb.LocalStructuredProperty(PageResultGoogle, repeated= T...</code>
pages	Número de páginas Value: <code>ndb.IntegerProperty()</code>
scoreUsability	Puntuación media del test experiencia de usuario Value: <code>ndb.FloatProperty()</code>
scoreSpeed	Puntuación media del test de velocidad Value: <code>ndb.FloatProperty()</code>
score	Puntuación obtenida Value: <code>ndb.FloatProperty()</code>
date	Fecha en la que se realizó la validación Value: <code>ndb.DateProperty(auto_now= True)</code>

continued on next page

Name	Description
time	Hora en la que se realizó la validación Value: <code>ndb.TimeProperty(auto_now= True)</code>

1.5 Class User

google.appengine.ext.ndb.Model —
model.User

Modelo de usuario del sistema

1.5.1 Class Variables

Name	Description
name	Nombre. Identifica al usuario. Value: <code>ndb.StringProperty()</code>
full_name	Nombre completo Value: <code>ndb.StringProperty()</code>
group	Grupo de usuarios al que pertenece Value: <code>ndb.StringProperty()</code>
password	Contraseña Value: <code>ndb.StringProperty()</code>
email	Correo electrónico Value: <code>ndb.StringProperty()</code>
n_links	Variable de estado. Número de páginas que se van a validar cuando se está realizando una validación. Cuando no se está validado toma el valor -1 Value: <code>ndb.IntegerProperty(default=-1)</code>
root_link	Variable de estado. Enlace de la web que se está validando. Value: <code>ndb.StringProperty()</code>
validation_type	Variable de estado. Indica el tipo de validación que se está realizando. Value: <code>ndb.StringProperty()</code>
onlyDomain	Variable de estado. Indica si estamos realizando una validación de sólo dominio. Value: <code>ndb.BooleanProperty()</code>
lock	Variable de estado. Cerrojo que se activa cuando estamos realizando una validación y se desactiva cuando termina. Value: <code>ndb.BooleanProperty(default= False)</code>

1.6 Class ReportRank

google.appengine.ext.ndb.Model —
model.ReportRank

Modelo de informe de Ranking

1.6.1 Class Variables

Name	Description
web	Página raíz Value: ndb.StringProperty()
user	Usuario que ha realizado la validación Value: ndb.StringProperty()
score	Puntuación global obtenida Value: ndb.FloatProperty()
html_test	Identificador del informe HTML/CSS asociado Value: ndb.StringProperty()
wcag2A_test	Identificador del informe WCAG 2.0 A asociado Value: ndb.StringProperty()
wcag2AA_test	Identificador del informe WCAG 2.0 AA asociado Value: ndb.StringProperty()
availability_test	Identificador del informe de disponibilidad asociado Value: ndb.StringProperty()
mobile_test	Identificador del informe para móviles de Google asociado Value: ndb.StringProperty()

2 Module server

2.1 Functions

insertReport(*username, validation_type, isRank*)

Inserta informe ejecutado por el usuario *username* de tipo *validation_type* e indicamos si se trata de parte de informe de ranking.

El informe obtiene las páginas analizadas de la base de datos usando los parámetros de entrada proporcionados. A partir de las páginas cuenta las páginas con error y la lista de errores. Con estos valores calcula la puntuación que se le asignará.

Si el informe es de Ranking y ya existía se actualiza en lugar de crear uno nuevo.

Parameters

username: nombre de usuario
(*type=str*)

validation_type: tipo de validación. Los valores que puede tomar son:

- HTML
- WCAG2.0-A
- WCAG2.0-AA
- CHECK AVAILABILITY

(*type=str*)

isRank: indica si es parte de un informe de ranking
(*type=bool*)

insertGoogleReport(*username, isRank*)

Inserta informe ejecutado por el usuario *username* del test de móviles de Google, e indicamos si se trata de un informe de ranking.

El informe obtiene las páginas analizadas de la base de datos usando los parámetros de entrada proporcionados. Obtiene la puntuación de usabilidad y velocidad de cada página, realizando una media de cada una para obtener la nota general de cada categoría. Finalmente calcula la nota final del test usando la media geométrica de ambas puntuaciones mencionadas anteriormente.

Si el informe es de Ranking y ya existía se actualiza en lugar de crear uno nuevo.

Parameters

username: nombre de usuario
(*type=str*)

isRank: indica si es parte de un informe de ranking
(*type=bool*)


main()

Main que ejecuta la aplicación WSGI de Webapp2

2.2 Variables

Name	Description
DEFAULT_MAX_PAGS	páginas por defecto Value: 25
DEFAULT_DEPTH	profundidad por defecto Value: 2
JINJA_ENVIRONMENT	entorno de plantillas jinja2 Value: jinja2.Environment(loader=jinja2.FileSystemLoader(os.pat...
head	cabecera html común Value: JINJA_ENVIRONMENT.get_template('template/head.html').rend...
footer	footer html común Value: JINJA_ENVIRONMENT.get_template('template/footer.html').re...
error_login	Value: "
urls	Lista de URLs del servicio asociadas a su clase controladora Value: [('/', MainPage), ('/login', login), ('/validation', Vali...
application	aplicación WSGI Webapp2 Value: webapp2.WSGIApplication(urls, debug=True)

2.3 Class MainPage

webapp2.RequestHandler  **server.MainPage**

Clase manejadora de la página principal

2.3.1 Methods

get(<i>self</i>)
Respuesta GET de la página principal

2.4 Class login

webapp2.RequestHandler  **server.login**

Clase manejadora de la página de login.

2.4.1 Methods

get(self)

Respuesta GET del formulario de login

post(self)

Respuesta POST del formulario de login.

Parameters

user: nombre de usuario

(*type=str*)

password: contraseña del usuario

(*type=str*)

2.5 Class logout

webapp2.RequestHandler — **server.logout**

Clase manejadora del logout

2.5.1 Methods

get(self)

Respuesta GET que cierra la sesión eliminando la cookie

2.6 Class QueueValidation

webapp2.RequestHandler — **server.QueueValidation**

Clase manejadora del encolado de validaciones.

2.6.1 Methods

post(*self*)

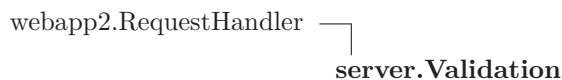
Respuesta POST al enviar el formulario de la página de inicio que se encarga de obtener los links a partir de la raíz y ejecutar el análisis seleccionado metiendo las tareas correspondientes en la cola de tareas.

Las variables de estado del usuario también se modifican usando los parámetros seleccionados en el formulario

Parameters

url:	web raíz que se va a analizar (<i>type=str</i>)
max_pags:	número máximo de páginas que se van a analizar (<i>type=int</i>)
depth:	profundidad del análisis (<i>type=int</i>)
onlyDomain:	indica si es sólo dominio (<i>type=bool</i>)
reg:	expresión regular que filtra páginas web del análisis (<i>type=str</i>)

2.7 Class Validation



Clase manejadora de la validación de una página.

2.7.1 Methods

post(self)

Respuesta POST que realiza una validación de una página en función de los parámetros que se le pasen por el método POST.

Esta página se almacenará en la base de datos.

Parameters

optradio: Opción que indica el tipo de página:

- HTML
- WCAG2.0-A
- WCAG2.0-AA
- CHECK AVAILABILITY

(*type=str*)

username: Nombre de usuario que realiza la validación

(*type=str*)

url: web que se va a validar

(*type=str*)

number: número de la página actual

(*type=int*)

2.8 Class Reports



Clase manejadora de la página de visualización e inserción de los informes.

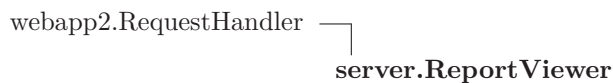
2.8.1 Methods

get(self)

Respuesta GET de la página de informes. Se encarga de mostrar los informes realizados por el usuario.

Además se encarga de insertar los informes cuando el proceso de validación termina.

2.9 Class ReportViewer



Clase manejadora de la visualización de un informe.

2.9.1 Methods

get(self)

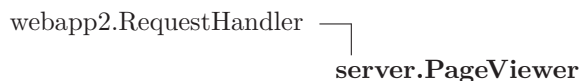
Respuesta GET de la vista general de un informe. Muestra el informe pasado como argumento al método GET.

Parameters

id: identificador del informe que se está visualizando.

(*type=long*)

2.10 Class PageViewer



Clase manejadora de la página de un informe.

2.10.1 Methods

get(self)

Respuesta GET de la página de un informe. Muestra el informe pasado como argumento al método GET.

Parameters

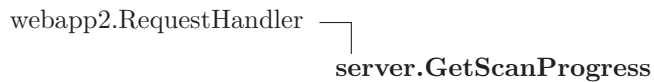
id: identificador del informe que se está visualizando.

(*type=long*)

number: número de página que se está visualizando

(*type=int*)

2.11 Class GetScanProgress



Clase manejadora del progreso de validación.

2.11.1 Methods

get(*self*)

Respuesta GET que obtiene el progreso de la validación que se está realizando. Este método es llamado por todas las páginas mediante AJAX.

2.12 Class Rankings



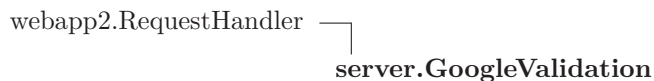
Clase manejadora de la página de visualización de los Rankings.

2.12.1 Methods

get(*self*)

Respuesta GET que obtiene las puntuaciones de los tests de ranking para mostrarlos.

2.13 Class GoogleValidation



Clase manejadora de la validación de un informe de Google para móviles.

2.13.1 Methods

post(*self*)

Respuesta POST que valida una página del test de móviles de Google.

Esta página se almacenará en la base de datos.

Parameters

url: página web que se validará
(*type=*str)

username: usuario que realiza la validación
(*type=*str)

number: número de página que se va a validar
(*type=*int)

2.14 Class `GoogleReportViewer`

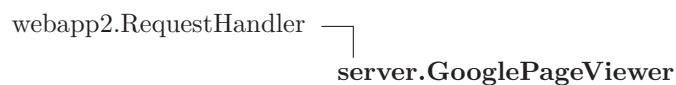


Clase manejadora de la visualización de un informe de Google para móviles.

2.14.1 Methods

<code>get(self)</code>
Respuesta GET de la vista general de un informe de test de móvil de Google.
Parameters
id: identificador del informe que se está visualizando (<i>type=long</i>)

2.15 Class `GooglePageViewer`



Clase manejadora de una página de un informe de Google para móviles.

2.15.1 Methods

<code>get(self)</code>
Respuesta GET de la vista de una página de un informe de test de móvil de Google.
Parameters
id: identificador del informe que se está visualizando (<i>type=long</i>)
number: número de página que se está visualizando (<i>type=int</i>)

2.16 Class `Users`



Clase manejadora de la página de edición de usuarios.

2.16.1 Methods

get(*self*)

Respuesta GET de la visualización de usuarios registrados en el sistema.

Obtiene la lista de usuarios registrados y muestra sus datos.

En caso de que el usuario que intente acceder a esta vista no sea del grupo "Admin" recibirá el mensaje: "Unauthorized".

2.17 Class *EditUser*



Clase manejadora del proceso de edición de un usuario.

2.17.1 Methods

post(*self*)

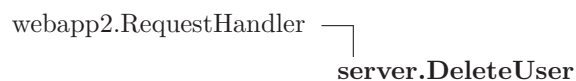
Respuesta POST que edita los campos del usuario pasados como parámetro. El identificador del usuario es su nombre, y este campo no será modificable.

En caso de que el usuario que intente acceder a este método no sea del grupo "Admin" recibirá el mensaje: "Unauthorized".

Parameters

edit-full_name:	nombre completo nuevo (<i>type=str</i>)
edit-email:	email nuevo (<i>type=str</i>)
edit-group:	grupo nuevo (<i>type=str</i>)
name:	identificador de usuario que se va a modificar (<i>type=str</i>)

2.18 Class *DeleteUser*



Clase manejadora del proceso de borrado de un usuario.

2.18.1 Methods

post(self)

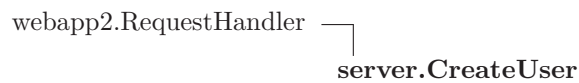
Respuesta POST que elimina un usuario de la base de datos identificado por name.

En caso de que el usuario que intente acceder a este método no sea del grupo "Admin" recibirá el mensaje: "Unauthorized".

Parameters

name: identificador del usuario que se va a eliminar
(*type=*str)

2.19 Class CreateUser



Clase manejadora del proceso de creación de un usuario.

2.19.1 Methods

post(self)

Respuesta POST que crea un usuario a partir de los campos pasados como parámetro. El identificador del usuario es su nombre, y este campo no será modificable.

En caso de que el usuario que intente acceder a este método no sea del grupo "Admin" recibirá el mensaje: "Unauthorized".

Parameters

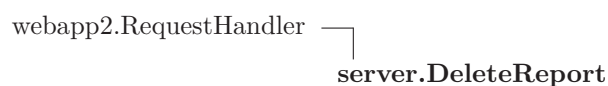
create-name: nombre e identificador de usuario
(*type=*str)

create-full_name: nombre completo
(*type=*str)

create-email: email
(*type=*str)

create-group: grupo
(*type=*str)

2.20 Class DeleteReport



Clase manejadora del proceso de borrado de un informe.

2.20.1 Methods

post(*self*)

Respuesta POST que elimina un informe de la base de datos a partir de su id pasada como parámetro.

Parameters

id: identificador del informe
(*type=long*)

2.21 Class DeleteRankingReport

webapp2.RequestHandler —
server.DeleteRankingReport

Clase manejadora del proceso de borrado de un informe de Ranking.

2.21.1 Methods

post(*self*)

Respuesta POST que elimina un informe de Ranking de la base de datos a partir de su id pasada como parámetro. Elimina también, por tanto, los cinco informes asociados a éste.

Parameters

id: identificador del informe
(*type=long*)

2.22 Class RankingReports

webapp2.RequestHandler —
server.RankingReports

Clase manejadora de la página de visualización e inserción de los informes de Ranking.

2.22.1 Methods

get(*self*)

Respuesta GET que muestra los informes de Ranking.

Si se ha completado una validación de tipo Ranking se insertarán en la base de datos los cinco informes asociados y el informe de tipo Ranking que los referencia. Al insertar este informe se calcula la puntuación global, que se trata de la media geométrica de la puntuación de los cinco informes asociados.

Si ya existía un informe de Ranking de la misma web se actualiza en lugar de crear uno nuevo.

2.23 Class *RankingReportViewer*



Clase manejadora de la visualización de un informe de Raking.

2.23.1 Methods

get(*self*)

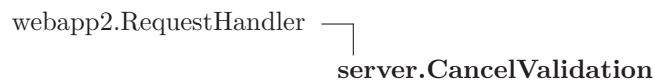
Respuesta GET de la vista de un informe de Ranking. Muestra los informes asociados al informe de Ranking pasado como argumento al método GET.

Parameters

id: identificador del informe de Ranking que se está visualizando.

(*type=long*)

2.24 Class *CancelValidation*



Clase manejadora del proceso de cancelación de una validación en curso.

2.24.1 Methods

get(*self*)

Respuesta POST que cancela el proceso de validación de una web.

Para la cancelación se detienen las tareas que se encuentran en la cola, se eliminan las páginas guardadas en la base de datos hasta el momento y se reinician las variables de estado del usuario.

2.25 Class `NotFound`

`webapp2.RequestHandler` —
`server.NotFound`

Clase manejadora de la página de error al no encontrar el recurso indicado.

2.25.1 Methods

<code>get(self)</code>
Respuesta GET de la página que maneja las peticiones no encontradas en el servidor. Devuelve un mensaje de error diciendo que el recurso solicitado no se ha encontrado.

3 Package spiderbot

3.1 Modules

- **parsers** (*Section 4, p. 21*)
- **validators** (*Section 5, p. 24*)

3.2 Variables

Name	Description
<code>__package__</code>	Value: None

4 Module *spiderbot.parsers*

4.1 Functions

parseHTMLValidation(*validation, url*)

Da formato legible parseando el diccionario *validation* obtenido de haber validado el estándar HTML de la url pasada como parámetro.

Parameters

validation: diccionario JSON con el resultado de validar la url
(*type=dict*)

url: url que se va a parsear
(*type=str*)

Return Value

diccionario que contiene los resultados de realizar el parseo:

- out: texto de salida formateado
- errors: número de errores encontrados en la validación
- state: estado de la validación

(*type=dict*)

parseCSSValidation(*validation, url*)

Da formato legible parseando el diccionario *validation* obtenido de haber validado el estándar CSS de la url pasada como parámetro.

Parameters

validation: diccionario JSON con el resultado de validar la url
(*type=dict*)

url: url que se va a parsear
(*type=str*)

Return Value

diccionario que contiene los resultados de realizar el parseo:

- out: texto de salida formateado
- errors: número de errores encontrados en la validación
- state: estado de la validación
- list_errors: lista de errores encontrados en la validación

(*type=dict*)

parseAvailabilityValidation(*code*)

Da formato legible parseando el código HTTP de respuesta del servidor.

Parameters

code: código HTTP que devuelve el servidor
(*type=int*)

Return Value

diccionario que contiene los resultados de realizar el parseo:

- out: texto de salida formateado
- errors: número de errores encontrados en la validación
- state: estado de la validación

(*type=dict*)

parseWCAGValidation(*validation, url*)

Da formato legible parseando el diccionario validation obtenido de haber validado el estándar WCAG 2.0 de la url pasada como parámetro.

Parameters

validation: diccionario JSON con el resultado de validar la url
(*type=dict*)

url: url que se va a parsear
(*type=str*)

Return Value

diccionario que contiene los resultados de realizar el parseo:

- out: texto de salida formateado
- errors: número de errores encontrados en la validación
- state: estado de la validación
- list_errors: lista de errores encontrados en la validación

(*type=dict*)

parseGoogleValidation(*validation*, *url*)

Da formato legible parseando el diccionario validation obtenido de haber validado el estándar CSS de la url pasada como parámetro.

Parameters

validation: diccionario JSON con el resultado de validar la url
(type=dict)

url: url que se va a parsear
(type=str)

Return Value

diccionario que contiene los resultados de realizar el parseo:

- scoreUsability: puntuación de experiencia de usuario
- scoreSpeed: puntuación de velocidad de carga
- ruleNames: lista con los nombres de las reglas
- ruleImpacts: lista con los impactos de las reglas
- types: lista con los tipos de variable
- summaries: lista de resúmenes de las reglas
- urlBlocks: lista de bloques de URLs

(type=dict)

4.2 Variables

Name	Description
<code>__package__</code>	Value: None

5 Module *spiderbot.validators*

5.1 Functions

validate(*filename*, *page_type*)

Valida una web HTML o CSS y devuelve un JSON resultado como diccionario.

Parameters

filename: nombre de URL HTTP
page_type: tipo de página que se quiere validar. Puede ser:

- html
- css

filename: (*type=str*)
page_type: (*type=str*)

Return Value

diccionario JSON que contiene los resultados de la validación. Si la validación no es correcta devuelve "

(*type=dict*)

validateWCAG(*filename*, *guide*)

Valida una web con el estándar WCAG 2.0

Parameters

filename: nombre de URL HTTP
(*type=str*)

guide: directrices de WCAG 2.0. Puede ser:

- WCAG2-A
- WCAG2-AA
- WCAG2-AAA

(*type=str*)

Return Value

diccionario que contiene los resultados de la validación. Si la validación no es correcta devuelve "

(*type=dict*)

checkAvailability(*filename*)

Comprueba la disponibilidad de una web.

Parameters

filename: nombre de la URL HTTP
(*type=str*)

Return Value

código que devuelve el servidor ante la petición GET
(*type=int*)

getAllLinks(*root, depth, max_pages, onlyDomain, reg='.*'*)

Algoritmo principal de la araña. Obtiene los links de la página raíz *root* con una búsqueda en anchura, determinada por una profundidad de búsqueda *depth* y número máximo de páginas *max_pages*. Puede filtrar páginas si se indica que sólo busque en su propio dominio o usar una expresión regular *reg*.

Parameters

root: página raíz
(*type=str*)

depth: profundidad de búsqueda
(*type=int*)

max_pages: links que se obtendrán como máximo
(*type=int*)

onlyDomain: indica si se quieren filtrar páginas que estén sólo en su dominio
(*type=bool*)

reg: expresión regular para filtrar páginas en la obtención de links
(*type=str*)

Return Value

lista de enlaces compuesta por objetos de tipo tupla con dos valores:

- 0: nombre del enlace
- 1: tipo de enlace: html o css

(*type=list*)

html_decode(*s*)

Returns the ASCII decoded version of the given HTML string. This does NOT remove normal HTML tags like <p>.

GoogleMobileValidation(*filename*)

Valida la página *filename* usando el validador de Google para móviles.

Parameters

filename: nombre de la URL HTTP
(*type=str*)

Return Value

diccionario JSON que contiene los resultados de la validación. Si la validación no es correcta devuelve "

(*type=dict*)

5.2 Variables

Name	Description
html_validator_url	URL del validador HTML Value: 'http://validator.w3.org/check'

continued on next page

Name	Description
css_validator_url	URL del validador CSS Value: 'http://jigsaw.w3.org/css-validator/validator'
wcag_validator_url	URL del validador WCAG Value: 'http://achecker.ca/checkacc.php'
ACHECKER_ID	Clave para usar la API de AChecker Value: '8d50ba76d166da61bdc9dfa3c97247b32dd1014c'
pagespeed_api_url	URL del validador de Google para móviles Value: 'https://www.googleapis.com/pagespeedonline/v2/runPagespeed'
GOOGLE_API_KEY	Clave para usar la API de Google Value: 'AIzaSyDmU2XjCiUJdMyDDhX8AQ8n-4vye8i5Uzk'

Index

- model (*module*), 2–6
 - model.PageResult (*class*), 2
 - model.PageResultGoogle (*class*), 2–3
 - model.Report (*class*), 3–4
 - model.ReportGoogle (*class*), 4–5
 - model.ReportRank (*class*), 5–6
 - model.User (*class*), 5
- server (*module*), 7–19
 - server.CancelValidation (*class*), 18
 - server.CancelValidation.get (*method*), 18
 - server.CreateUser (*class*), 16
 - server.CreateUser.post (*method*), 16
 - server.DeleteRankingReport (*class*), 17
 - server.DeleteRankingReport.post (*method*), 17
 - server.DeleteReport (*class*), 16–17
 - server.DeleteReport.post (*method*), 17
 - server.DeleteUser (*class*), 15–16
 - server.DeleteUser.post (*method*), 16
 - server.EditUser (*class*), 15
 - server.EditUser.post (*method*), 15
 - server.GetScanProgress (*class*), 12–13
 - server.GetScanProgress.get (*method*), 13
 - server.GooglePageViewer (*class*), 14
 - server.GooglePageViewer.get (*method*), 14
 - server.GoogleReportViewer (*class*), 13–14
 - server.GoogleReportViewer.get (*method*), 14
 - server.GoogleValidation (*class*), 13
 - server.GoogleValidation.post (*method*), 13
 - server.insertGoogleReport (*function*), 7
 - server.insertReport (*function*), 7
 - server.login (*class*), 8–9
 - server.login.get (*method*), 9
 - server.login.post (*method*), 9
 - server.logout (*class*), 9
 - server.logout.get (*method*), 9
 - server.main (*function*), 7
 - server.MainPage (*class*), 8
 - server.MainPage.get (*method*), 8
 - server.NotFound (*class*), 18–19
 - server.NotFound.get (*method*), 19
 - server.PageViewer (*class*), 12
 - server.PageViewer.get (*method*), 12
 - server.QueueValidation (*class*), 9–10
 - server.QueueValidation.post (*method*), 10
 - server.RankingReports (*class*), 17–18
 - server.RankingReports.get (*method*), 18
 - server.RankingReportViewer (*class*), 18
 - server.RankingReportViewer.get (*method*), 18
 - server.Rankings (*class*), 13
 - server.Rankings.get (*method*), 13
 - server.Reports (*class*), 11
 - server.Reports.get (*method*), 11
 - server.ReportViewer (*class*), 11–12
 - server.ReportViewer.get (*method*), 12
 - server.Users (*class*), 14–15
 - server.Users.get (*method*), 15
 - server.Validation (*class*), 10–11
 - server.Validation.post (*method*), 11
- spiderbot (*package*), 20
 - spiderbot.parsers (*module*), 21–23
 - spiderbot.parsers.parseAvailabilityValidation (*function*), 21
 - spiderbot.parsers.parseCSSValidation (*function*), 21
 - spiderbot.parsers.parseGoogleValidation (*function*), 22
 - spiderbot.parsers.parseHTMLValidation (*function*), 21
 - spiderbot.parsers.parseWCAGValidation (*function*), 22
 - spiderbot.validators (*module*), 24–26
 - spiderbot.validators.checkAvailability (*function*), 24
 - spiderbot.validators.getAllLinks (*function*), 24
 - spiderbot.validators.GoogleMobileValidation (*function*), 25
 - spiderbot.validators.html_decode (*function*), 25
 - spiderbot.validators.validate (*function*), 24
 - spiderbot.validators.validateWCAG (*function*), 24