

---

# **Zaguan Documentation**

***Release 3.2.0***

**Msa Team**

**Dec 17, 2019**



---

## Contents

---

<b>1</b>	<b>Supported toolkits</b>	<b>3</b>
<b>2</b>	<b>Installation</b>	<b>5</b>
<b>3</b>	<b>Other requirements</b>	<b>7</b>
<b>4</b>	<b>Table of contents</b>	<b>9</b>
4.1	zagan . . . . .	9
4.1.1	zagan . . . . .	9
4.1.1.1	Submodules . . . . .	9
4.1.1.2	Module contents . . . . .	14
4.2	examples . . . . .	15
4.2.1	Subpackages . . . . .	15
4.2.1.1	colors . . . . .	15
4.2.2	Module contents . . . . .	15
<b>5</b>	<b>Indices and tables</b>	<b>17</b>
	<b>Python Module Index</b>	<b>19</b>
	<b>Index</b>	<b>21</b>



Zaguan is a Framework to allow the easy creation of hibrid apps.

Zaguan is Free Software! you can check the code at <http://github.com/MSA-Argentina/zaguan>



# CHAPTER 1

---

## Supported toolkits

---

- GTK+ (Full support)
- QT4 No event caching or developer tools





## CHAPTER 2

---

### Installation

---

#### With easy\_install

```
sudo easy_install zaguan
```

#### With pip

```
sudo pip install zaguan
```

#### From source

```
git clone https://github.com/MSA-Argentina/zaguan.git  
sudo python setup.py install
```



## CHAPTER 3

---

### Other requirements

---

- python-webkit package (to use with GTK)
- QT4 (to use with QT)



## 4.1 zaguan

### 4.1.1 zaguan

#### 4.1.1.1 Submodules

#### actions

**class** `zaguan.actions.BaseActionController` (*controller*)

Bases: `object`

Clase base para las acciones que son enlazadas con los controladores que heredan de `WebContainerController`

**Variables** `__controler` (`zaguan.controller.WebContainerController`) – controlador con el cual interactua la clase.

`__init__` (*controller*)

Constructor de la clase. Establece el link con un controlador.

**Parameters** `controller` (`zaguan.controller.WebContainerController`) – el controlador a linkear.

**controller**

Getter para `__controller`.

**Returns** el controlador

**Return type** `zaguan.controller.WebContainerController`

**send\_command** (*\*args, \*\*kwargs*)

Envía un comando al controlador utilizando el método `WebContainerController.send_command()` pasando *\*args* y *\*\*kwargs*

## container

`zaguan.container.launch_browser(uri, debug=False, user_settings=None, window=None, webkit_version=None, debug_callback=None)`

Crea e inicializa el objeto browser.

### Parameters

- **debug** (*boolean*) – indica si se debe mostrar informacion de debug.
- **user\_settings** (*dict*) – diccionario que contiene las settings que se deben pasar al webkit.
- **window** (*Gtk.Window*) – objeto al que se le va a inyectar el browser.
- **webkit\_version** (*int*) – la version de webkit a usar. Puede ser 1 o 2.

### Returns

- **browser** (*WebKit2.WebView*) – la instancia de WebView.
- **\_web\_send** (*launcher\_browser.\_web\_send()*) – la funcion para ejecutar JS en el browser.
- **implementation** (*WebKitMethods* o *WebKit2Methods*) – instancia del wrapper Zaguan segun la version elegida de WebKit.

`launcher_browser._web_send(msg)`

Inyecta javascript de forma asíncrona en la vista. Esta funcion es uno de los valores a devolver por `launch_browser()`.

**Parameters** `msg` (*str*) – el codigo javascript a correr en el browser

## controller

**class** `zaguan.controller.WebContainerController`

Bases: `object`

Clase base para los controladores de contenedores web. Controla la interaccion con el browser en ambas direcciones.

### Variables

- **processors** (*list*) – lista de procesadores.
- **inspector** (*ZaguanInspector*) – se instancia solo cuando se usa WebKit version 1.

`__init__()`

Constructor de la clase. Inicializa `WebContainerController.processors` con una lista vacía.

**add\_processor** (*url\_word, instance=None*)

Agrega una instancia de la funcion `_inner` a la lista de procesadores para que desde el browser se pueda llamar a metodos de Python. Para mas detalles ver la documentacion de `_inner()`.

### Parameters

- **url\_word** (*str*) – Clave que linkea una URI con el procesador `instance`.
- **instance** (`zaguan.actions.BaseActionController`) – Una instancia de `BaseActionController` que debe tener implementados los metodos que se intentaran ejecutar desde el browser para la URI.

`add_processor._inner(uri)`

Si alguna parte de la URI coincide con `url_word` entonces procesa el texto restante y lo divide en dos partes: el nombre del método y los datos (parámetros a pasar al metodo). Busca el método en la instancia de `BaseActionController` (instance) y si existe lo ejecuta.

**Parameters** `uri (str)` – la URI a parsear.

**Raises** `NotImplementedError` – excepcion que se lanza cuando el método que está llamando el browser no fué implementado en la instancia de `BaseActionController`.

`get_browser(uri, settings=None, debug=False, webkit_version=None, debug_callback=None)`

Obtiene el browser, el metodo para inyectar JS y la implementacion del wrapper de WebKit. Conecta el método `WebContainerController.on_navigation_requested` al evento `resource-request-starting` del browser, esto es para poder atender a nuevas peticiones de URI.

**Parameters**

- `uri (str)` – the URI of the HTML to open with the web view.
- `settings (list)` – the settings send to webkit.
- `debug (boolean)` – to indicate if it should output debug and add context menu and inspector.
- `webkit_version (int)` – the webkit gtk version (1 or 2)

**Returns** el objeto `WebView`

**Return type** `WebKit2.WebView`

`on_navigation_requested(webview, resource, request, *args)`

Este es el método que se ejecuta cada vez que en el browser se pide cargar una URI. Llama a `WebContainerController.process_uri()` con la URI que obtiene de la request que se pasa como parametro.

Los argumentos cambian segun la version de WebKit:

- Version 1: no existe documentacion.
- Version 2: [http://lazka.github.io/pgi-docs/WebKit2-4.0/classes/WebView.html#WebKit2.WebView.signals.resource\\_load\\_started](http://lazka.github.io/pgi-docs/WebKit2-4.0/classes/WebView.html#WebKit2.WebView.signals.resource_load_started)

`process_uri(uri)`

Por cada procesador registrado manda a procesar la URI.

**Parameters** `uri (str)` – la URI a procesar.

`send_command(command, data=None)`

Inyecta la ejecucion de un comando en el browser.

**Parameters**

- `command (str)` – el comando a ejecutar en el browser.
- `data (any)` – los datos que se envian como parametros del comando.

`set_screen(screen, **kwargs)`

Envía el comando ‘change\_screep’ al browser.

**Parameters**

- `screen (str)` – pantalla a la que se quiere cambiar.
- `kargs` – otros argumentos

## engines

**class** zaguan.engines.WebKit2Methods

Bases: `object`

Clase que implementa métodos estaticos para la version 2 de WebKit.

**static connect** (*browser, callback*)

Conecta el evento de navegación al browser.

### Parameters

- **browser** (`WebKit2.WebView`) – el objeto WebView destino.
- **callback** (`function`) – la función callback.

**static create\_browser** (*debug=False, cache\_model=None, process\_model=None*)

Crea la instancia de `WebKit2.WebView` y la configura correctamente.

### Parameters

- **debug** (`bool`) – indica si se debe abrir herramientas para desarrollador.
- **cache\_model** (`WebKit2.CacheModel`) – indica el comportamiento de la cache del browser.
- **process\_model** – sin uso.

**Returns** la instancia de `WebKit2.WebView` configurada.

**Return type** `WebKit2.WebView`

**static inject\_javascript** (*browser, script*)

Injecta JavaScript en un objeto `WebKit2.WebView`.

### Parameters

- **browser** (`WebKit2.WebView`) – el objeto WebView destino.
- **script** (`str`) – el script JS a correr.

**static open\_uri** (*browser, uri*)

Abre una URI en el browser.

### Parameters

- **browser** (`WebKit2.WebView`) – el objeto WebView destino.
- **uri** (`str`) – la URI del contenido a abrir en el browser.

**static print\_version** ()

Hace un print con la version de WebKit.

**static set\_settings** (*browser, user\_settings*)

Agrega las settings al browser.

### Parameters

- **browser** (`WebKit2.WebView`) – el objeto WebView destino.
- **user\_settings** – las settings a agregar al browser.

**class** zaguan.engines.WebKitMethods

Bases: `object`

Clase que implementa métodos estaticos para la version 1 de WebKit.



**static connect** (*browser, callback*)

Conecta el evento de navegación al browser.

**Parameters**

- **browser** (*WebView*) – el objeto WebView destino.
- **callback** (*function*) – la función callback.

**static create\_browser** (*debug=False, cache\_model=None, process\_model=None*)

Crea la instancia de `WebView` y la configura correctamente.

**Parameters**

- **debug** (*bool*) – indica si se debe abrir herramientas para desarrollador.
- **cache\_model** (*CacheModel*) – indica el comportamiento de la cache del browser.
- **process\_model** – sin uso.

**Returns** la instancia de `WebView` configurada.

**Return type** `WebView`

**static get\_inspector** (*browser*)

Obtiene el Inspector de la instancia de `WebKit`.

**Parameters** **browser** (*WebView*) – el objeto WebView destino.

**Returns** el inspector de zaguan.

**Return type** `Inspector`

**static inject\_javascript** (*browser, script*)

Injecta JavaScript en un objeto `WebView`.

**Parameters**

- **browser** (*WebView*) – el objeto WebView destino.
- **script** (*str*) – el script JS a correr.

**static open\_uri** (*browser, uri*)

Abre una URI en el browser.

**Parameters**

- **browser** (*WebView*) – el objeto WebView destino.
- **uri** (*str*) – la URI del contenido a abrir en el browser.

**static print\_version** ()

Hace un print con la version de `WebKit`.

**static set\_settings** (*browser, user\_settings*)

Agrega las settings al browser.

**Parameters**

- **browser** (*WebView*) – el objeto WebView destino.
- **user\_settings** – las settings a agregar al browser.

`zaguan.engines.get_wk_implementation` (*webkit\_version*)

Retorna el webkit wrapper de Zaguan segun la version.

**Parameters** **webkit\_version** (*int*) – la version de webkit a usar. Puede ser 1 o 2.

## functions

`zaguan.functions.asynchronous_gtk_message` (*fun*)

Genera un wrapper de la funcion que se pasa como parametro (*fun*) para ejecutarla de forma asíncrona usando `GLib.idle_add`.

**Parameters** *fun* (*function*) – la funcion a wrappear

**Returns** la funcion wrappeada.

**Return type** *fun2*

`asynchronous_gtk_message.worker` (*param*)

Ejecuta una funcion con sus *\*args* y *\*\*kargs*. Obtiene la funcion y sus parámetros de *param*

**Parameters** *param* (*tuple*) – tupla que debe contener (*funcion\_a\_ejecutar*, *\*args*, *\*\*kargs*)

`asynchronous_gtk_message.fun2` (*\*args*, *\*\*kargs*)

Ejecuta la funcion `GLib.idle_add` pasándole un `asynchronous_gtk_message.worker` y como *\*data* la tupla (*fun*, *\*args*, *\*\*kargs*)

### 4.1.1.2 Module contents

**class** `zaguan.Zaguan` (*uri*, *controller=None*)

Bases: `object`

Esta clase administra una ventana y le inyecta un browser obtenido desde `WebContainerController.get_browser()`. El browser muestra la *uri*.

#### Variables

- **controller** (`zaguan.controller.WebContainerController`) – se usa para crear el browser.
- **uri** (*str*) – indica los recursos que se requieren mostrar en el browser.
- **on\_close** – funcion que se ejecuta al cerrar.

`__init__` (*uri*, *controller=None*)

Constructor de Zaguan

#### Parameters

- **uri** (*str*) – la URI del HTML a visualizar en el browser.
- **controller** (`zaguan.controller.WebContainerController`) – el controlador.

**quit** (*widget*, *event*)

Funcion que se ejecuta al cerrar la ventana. Ejecuta `on_close` con los parametros.

#### Parameters

- **widget** – parametro que toma cuando *Window* llama al callback al salir
- **event** – parametro que toma cuando *Window* llama al callback al salir

**run** (*settings=None*, *window=None*, *debug=False*, *on\_close=None*)

Si no se pasa una ventana, la crea y la muestra. Obtiene un browser desde `controller` y lo inyecta en la ventana.

#### Parameters

- **settings** (*list*) – lista de settings a enviar al webkit.

- **window** (*Gtk.Window*) – la ventana que contiene el browser donde se visualizará el HTML.
- **debug** (*bool*) – indica si se debe abrir las herramientas para desarrollador del browser.
- **on\_close** (*function*) – funcion que se ejecutará al cerrar.

## 4.2 examples

### 4.2.1 Subpackages

#### 4.2.1.1 colors

##### Submodules

##### actions

**class** zaguan.examples.colors.actions.**ColorsControllerActions** (*controller*)

Bases: *zaguan.actions.BaseActionController*

Thsi are the actions for the colors controller.

**document\_ready** ()

Action excecuted when the document is ready.

**log** (*data*)

Action excecuted when ‘log’ is called and debug is True.

**select\_color** (*data*)

Action excecuted when ‘select\_color’ is called.

##### controller

**class** zaguan.examples.colors.controller.**ColorsController**

Bases: *zaguan.controller.WebContainerController*

**ready** ()

##### run

zaguan.examples.colors.run.**load\_browser** ()

zaguan.examples.colors.run.**load\_window** ()

##### settings

##### web\_server

##### Module contents

### 4.2.2 Module contents



## CHAPTER 5

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



### Z

- zagan, [14](#)
- zagan.actions, [9](#)
- zagan.container, [10](#)
- zagan.controller, [10](#)
- zagan.engines, [12](#)
- zagan.examples, [15](#)
- zagan.examples.colors, [15](#)
- zagan.examples.colors.actions, [15](#)
- zagan.examples.colors.controller, [15](#)
- zagan.examples.colors.run, [15](#)
- zagan.examples.colors.settings, [15](#)
- zagan.functions, [14](#)





## Symbols

`__init__()` (*zaguan.Zaguan* method), 14  
`__init__()` (*zaguan.actions.BaseActionController* method), 9  
`__init__()` (*zaguan.controller.WebContainerController* method), 10

## A

`add_processor()` (*zaguan.controller.WebContainerController* method), 10  
`asynchronous_gtk_message()` (in module *zaguan.functions*), 14  
`asynchronous_gtk_message.fun2()` (in module *zaguan.functions*), 14  
`asynchronous_gtk_message.worker()` (in module *zaguan.functions*), 14

## B

*BaseActionController* (class in *zaguan.actions*), 9

## C

*ColorsController* (class in *zaguan.examples.colors.controller*), 15  
*ColorsControllerActions* (class in *zaguan.examples.colors.actions*), 15  
`connect()` (*zaguan.engines.WebKit2Methods* static method), 12  
`connect()` (*zaguan.engines.WebKitMethods* static method), 12  
`controller` (*zaguan.actions.BaseActionController* attribute), 9  
`create_browser()` (*zaguan.engines.WebKit2Methods* static method), 12  
`create_browser()` (*zaguan.engines.WebKitMethods* static method), 13

## D

`document_ready()` (*zaguan.examples.colors.actions.ColorsControllerActions* method), 15

## G

`get_browser()` (*zaguan.controller.WebContainerController* method), 11  
`get_inspector()` (*zaguan.engines.WebKitMethods* static method), 13  
`get_wk_implementation()` (in module *zaguan.engines*), 13

## I

`inject_javascript()` (*zaguan.engines.WebKit2Methods* static method), 12  
`inject_javascript()` (*zaguan.engines.WebKitMethods* static method), 13

## L

`launch_browser()` (in module *zaguan.container*), 10  
`launcher_browser._web_send()` (in module *zaguan.container*), 10  
`load_browser()` (in module *zaguan.examples.colors.run*), 15  
`load_window()` (in module *zaguan.examples.colors.run*), 15  
`log()` (*zaguan.examples.colors.actions.ColorsControllerActions* method), 15

## O

`on_navigation_requested()` (*zaguan.controller.WebContainerController* method), 11

`open_uri()` (*zaguan.engines.WebKit2Methods static method*), 12  
`open_uri()` (*zaguan.engines.WebKitMethods static method*), 13

**P**

`print_version()` (*zaguan.engines.WebKit2Methods static method*), 12  
`print_version()` (*zaguan.engines.WebKitMethods static method*), 13  
`process_uri()` (*zaguan.controller.WebContainerController method*), 11

*zaguan.engines (module)*, 12  
*zaguan.examples (module)*, 15  
*zaguan.examples.colors (module)*, 15  
*zaguan.examples.colors.actions (module)*, 15  
*zaguan.examples.colors.controller (module)*, 15  
*zaguan.examples.colors.run (module)*, 15  
*zaguan.examples.colors.settings (module)*, 15  
*zaguan.functions (module)*, 14

## Q

`quit()` (*zaguan.Zaguan method*), 14

## R

`ready()` (*zaguan.examples.colors.controller.ColorsController method*), 15  
`run()` (*zaguan.Zaguan method*), 14

## S

`select_color()` (*zaguan.examples.colors.actions.ColorsControllerActions method*), 15  
`send_command()` (*zaguan.actions.BaseActionController method*), 9  
`send_command()` (*zaguan.controller.WebContainerController method*), 11  
`set_screen()` (*zaguan.controller.WebContainerController method*), 11  
`set_settings()` (*zaguan.engines.WebKit2Methods static method*), 12  
`set_settings()` (*zaguan.engines.WebKitMethods static method*), 13

## W

*WebContainerController (class in zaguan.controller)*, 10  
*WebContainerController.add\_processor.\_inner()* (*in module zaguan.controller*), 10  
*WebKit2Methods (class in zaguan.engines)*, 12  
*WebKitMethods (class in zaguan.engines)*, 12

## Z

*Zaguan (class in zaguan)*, 14  
*zaguan (module)*, 14  
*zaguan.actions (module)*, 9  
*zaguan.container (module)*, 10  
*zaguan.controller (module)*, 10