

---

# **Flask-DebugToolbar Documentation**

***Release 0.11.0***

**Matt Good**

**Mar 02, 2020**



---

## Contents

---

<b>1</b>	<b>Installation</b>	<b>3</b>
<b>2</b>	<b>Usage</b>	<b>5</b>
<b>3</b>	<b>Configuration</b>	<b>7</b>
<b>4</b>	<b>Panels</b>	<b>9</b>
4.1	Built-In Panels . . . . .	9
<b>5</b>	<b>Contributing</b>	<b>15</b>
<b>6</b>	<b>Thanks</b>	<b>17</b>
<b>7</b>	<b>Indices and tables</b>	<b>19</b>



This extension adds a toolbar overlay to Flask applications containing useful information for debugging.



# CHAPTER 1

---

## Installation

---

Installing is simple with `pip`:

```
$ pip install flask-debugtoolbar
```





## CHAPTER 2

---

### Usage

---

Setting up the debug toolbar is simple:

```
from flask import Flask
from flask_debugtoolbar import DebugToolbarExtension

app = Flask(__name__)

# the toolbar is only enabled in debug mode:
app.debug = True

# set a 'SECRET_KEY' to enable the Flask session cookies
app.config['SECRET_KEY'] = '<replace with a secret key>'

toolbar = DebugToolbarExtension(app)
```

The toolbar will automatically be injected into HTML responses when debug mode is on. In production, setting `app.debug = False` will disable the toolbar.

This extension also supports the Flask app factory pattern by separately creating the toolbar and later initializing it for an app:

```
toolbar = DebugToolbarExtension()
# Then later on.
app = create_app('the-config.cfg')
toolbar.init_app(app)
```



---

Configuration

---

The toolbar support several configuration options:

Name	Description	Default
DEBUG_TB_ENABLED	Enable the toolbar?	<code>app.debug</code>
DEBUG_TB_HOSTS	Whitelist of hosts to display toolbar	any host
DEBUG_TB_INTERCEPT_REDIRECTS	Should intercept redirects?	True
DEBUG_TB_PANELS	List of module/class names of panels	enable all built-in panels
DEBUG_TB_PROFILER_ENABLED	Enable the profiler on all requests	False, user-enabled
DEBUG_TB_TEMPLATE_EDITOR_ENABLED	Enable the template editor	False

To change one of the config options, set it in the Flask app's config like:

```
app.config['DEBUG_TB_INTERCEPT_REDIRECTS'] = False
```



### 4.1 Built-In Panels

#### 4.1.1 Versions

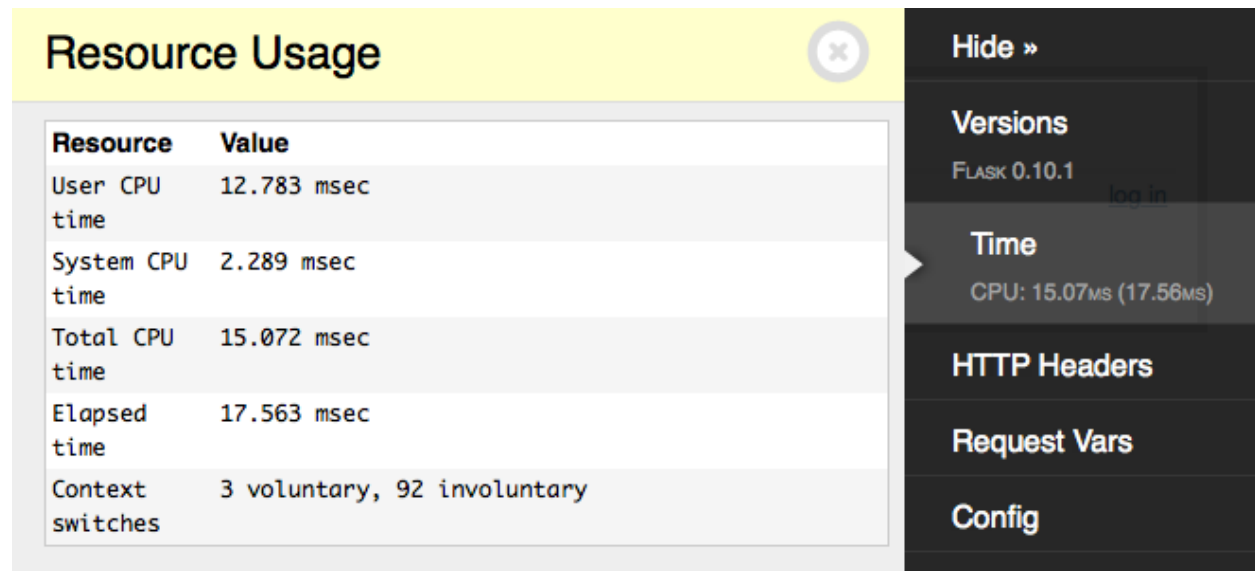
`flask_debugtoolbar.panels.versions.VersionDebugPanel`

Shows the installed Flask version. The expanded view displays all installed packages and their versions as detected by `setuptools`.

#### 4.1.2 Time

`flask_debugtoolbar.panels.timer.TimerDebugPanel`

Shows the time taken to process the current request. The expanded view includes the breakdown of CPU time, by user and system, wall clock time, and context switches.



**Resource Usage**

Resource	Value
User CPU time	12.783 msec
System CPU time	2.289 msec
Total CPU time	15.072 msec
Elapsed time	17.563 msec
Context switches	3 voluntary, 92 involuntary

**Hide »**

- Versions**  
FLASK 0.10.1 [log in](#)
- Time**  
CPU: 15.07ms (17.56ms)
- HTTP Headers**
- Request Vars**
- Config**

### 4.1.3 HTTP Headers

`flask_debugtoolbar.panels.headers.HeaderDebugPanel`

Displays the HTTP headers for the current request.



**HTTP Headers**

Key	Value
CONTENT_TYPE	
HTTP_ACCEPT	text/html,application/xhtml+xml,application/javascript;q=0.9,*/*;q=0.8
HTTP_ACCEPT_ENCODING	gzip, deflate
HTTP_ACCEPT_LANGUAGE	en-us
HTTP_CONNECTION	keep-alive
HTTP_HOST	127.0.0.1:5000
HTTP_USER_AGENT	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_6_8; rv:2.6.1) AppleWebKit/600.2.5 (KHTML, like Gecko) Safari/600.2.5
QUERY_STRING	

**Hide »**

- Versions**  
FLASK 0.10.1 [log in](#)
- Time**  
CPU: 15.07ms (17.56ms)
- HTTP Headers**
- Request Vars**
- Config**

### 4.1.4 Request Vars

`flask_debugtoolbar.panels.request_vars.RequestVarsDebugPanel`

Displays details of the Flask request-related variables, including the view function parameters, cookies, session variables, and GET and POST variables.

The screenshot displays the Flask-DebugToolbar interface. The main panel, titled 'Request Vars', is highlighted in yellow. It contains several sections: 'View information' with a table showing 'View Function' as `__main__.show_entries`, 'args' as an empty list, and 'kwargs' as `None`; 'COOKIES Variables' with a table showing a single entry `'fldt' u'hide'`; 'SESSION Variables' with the message 'No SESSION data'; 'GET Variables' with 'No GET data'; and 'POST Variables' with 'No POST data'. On the right, a sidebar contains links to 'Hide »', 'Versions' (showing 'FLASK 0.10.1' and a 'log in' link), 'Time' (showing 'CPU: 15.07ms (17.56ms)'), 'HTTP Headers', 'Request Vars' (which is the active panel), 'Config', 'Templates' (showing '1 RENDERED'), 'SQLAlchemy' (showing '0 QUERIES'), and 'Logging'.

View Function	args	kwargs
<code>__main__.show_entries</code>	<code>[]</code>	<code>None</code>

Variable	Value
<code>'fldt'</code>	<code>u'hide'</code>

#### 4.1.5 Config

`flask_debugtoolbar.panels.config_vars.ConfigVarsDebugPanel`

Shows the contents of the Flask application's config dict `app.config`.

Key	Value
APPLICATION_ROOT	None
DATABASE	'/tmp/flaskr.db'
DEBUG	True
DEBUG_TB_ENABLED	True
DEBUG_TB_HOSTS	()
DEBUG_TB_INTERCEPT_REDIRECTS	True
DEBUG_TB_PANELS	('flask_debugtoolbar.panels.ve
JSON_AS_ASCII	True
JSON_SORT_KEYS	True
JSONIFY_PRETTYPRINT_REGULAR	True
LOGGER_NAME	'__main__'
MAX_CONTENT_LENGTH	None
PASSWORD	'default'
PERMANENT_SESSION_LIFETIME	datetime.timedelta(31)

#### 4.1.6 Templates

`flask_debugtoolbar.panels.template.TemplateDebugPanel`

Shows information about the templates rendered for this request, and the value of the template parameters provided.

Variable	Value
entries	<input type="checkbox"/>
g	<flask.g of 'flaskr'>
request	<Request 'http://127.0.0.1:5000/' [GET]>
session	<SecureCookieSession {}>

#### 4.1.7 SQLAlchemy

`flask_debugtoolbar.panels.sqlalchemy.SQLAlchemyDebugPanel`

Shows SQL queries run during the current request.

**Note:** This panel requires using the [Flask-SQLAlchemy](#) extension in order to record the queries. See the Flask-



SQLAlchemy [Quickstart](#) section to configure it.

For additional details on query recording see the `get_debug_queries()` documentation.

**Note:** SQL syntax highlighting requires [Pygments](#) to be installed.

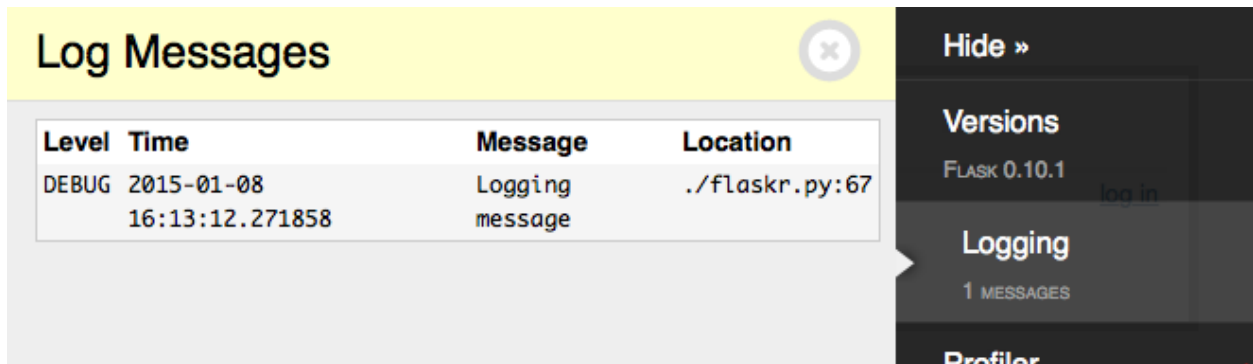


(ms)	Action	Context	Query
0.2339		./flaskr.py:71 (show_entries)	select title, text from entries

#### 4.1.8 Logging

`flask_debugtoolbar.panels.logger.LoggingPanel`

Displays log messages recorded during the current request.



Level	Time	Message	Location
DEBUG	2015-01-08 16:13:12.271858	Logging message	./flaskr.py:67

#### 4.1.9 Route List

`flask_debugtoolbar.panels.route_list.RouteListDebugPanel`

Displays the Flask URL routing rules.

#### 4.1.10 Profiler

`flask_debugtoolbar.panels.profiler.ProfilerDebugPanel`

Reports profiling data for the current request. Due to the performance overhead, profiling is disabled by default. Click the checkmark to toggle profiling on or off. After enabling the profiler, refresh the page to re-run it with profiling.

View: 1.25ms

Calls	Total Time (ms)	Per Call (ms)	Cumulative Time (ms)	Per Call (ms)	Function
1	0.185	0.1850	0.185	0.1850	{method 'execu
1	0.11	0.1100	0.11	0.1100	{_sqlite3.conn
5	0.065	0.0130	0.075	0.0150	<werkzeug/urls
2	0.037	0.0185	0.152	0.0760	<werkzeug/urls
10	0.031	0.0031	0.557	0.0557	./templates/la
5	0.031	0.0062	0.053	0.0106	./templates/sh
2	0.031	0.0155	0.128	0.0640	<werkzeug/rout

Hide »

Versions

FLASK 0.10.1

Logging

1 MESSAGES

Profiler

VIEW: 1.25MS

## CHAPTER 5

---

### Contributing

---

Fork us on [GitHub](#)



## CHAPTER 6

---

### Thanks

---

This was based on the original [django-debug-toolbar](#). Thanks to [Michael van Tellingen](#) for the original development of this Flask extension, and to all the [individual contributors](#).



## CHAPTER 7

---

### Indices and tables

---

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