

---

# **Autobahn-Sync Documentation**

*Release 0.3.2*

**Emmanuel Leblond**

**Jul 13, 2017**



---

# Contents

---

<b>1</b>	<b>Contents</b>	<b>3</b>
1.1	Tutorial . . . . .	3
1.2	API Reference . . . . .	3
<b>2</b>	<b>Indices and tables</b>	<b>9</b>
	<b>Python Module Index</b>	<b>11</b>



Autobahn integration with [crochet](#) to provide WAMP for synchronous applications.

Originally based on the work of [Sam & Max](#) (warning: French, pr0n and awesomeness inside !).



***Tutorial*** A quick tutorial to start with the project

***API Reference*** The complete API documentation

## Tutorial

Todo...

## API Reference

### Default API

Autobahn-Sync exposes an initialized `autobahn_sync.AutobahnSync` at the root of the module as a quick&easy access to the API:

```
import autobahn_sync
autobahn_sync.run(url=MY_ROUTER_URL, realm=MY_REALM)

@autobahn_sync.subscribe('com.app.event')
def on_event(e):
    print('%s happened' % e)

autobahn_sync.publish('com.app.event', 'trigger !')
```

### Advanced API

With the need to connect to multiple realms/routers, the default API is not enough and you should create other instances of `autobahn_sync.AutobahnSync`.

**class** autobahn\_sync.core.**AutobahnSync** (*authmethods=None*)

Main class representing the AutobahnSync application

**on\_challenge** (*func*)

Decorator providing a callback to the onChallenge event, use this instead of subclassing autobahn.twisted.wamp.ApplicationSession.onChallenge()

---

**Note:** This decorator can only be used before `AutobahnSync.run()` is called given the Challenge event is triggered at this time

---

**register** (*procedure=None, options=None*)

Decorator for the `AutobahnSync.session.register()`

---

**Note:** This decorator can be used before `AutobahnSync.run()` is called. In such case the actual registration will be done at `run()` time.

---

**run** (*url=u'ws://127.0.0.1:8080/ws', realm=u'realm1', authmethods=None, authid=None, authrole=None, authextra=None, blocking=False, callback=None, \*\*kwargs*)  
Start the background twisted thread and create the WAMP connection

**Parameters blocking** – If `False` (default) this method will spawn a new

thread that will be used to run the callback events (e.i. registered and subscribed functions). If `True` this method will not return and use the current thread to run the callbacks. :param `callback`: This callback will be called once init is done, use it with `blocking=True` to put your WAMP related init

**run\_in\_twisted** (*url=u'ws://127.0.0.1:8080/ws', realm=u'realm1', authmethods=None, authid=None, authrole=None, authextra=None, callback=None, \*\*kwargs*)

Start the WAMP connection. Given we cannot run synchronous stuff inside the twisted thread, use this function (which returns immediately) to do the initialization from a spawned thread.

**Parameters callback** – function that will be called inside the spawned thread.

Put the rest of you init (or you main loop if you have one) inside it

**Parameters**

- **authmethods** – Passed to `autobahn.wamp.protocol.ApplicationSession.join()`
- **authid** – Passed to `autobahn.wamp.protocol.ApplicationSession.join()`
- **authrole** – Passed to `autobahn.wamp.protocol.ApplicationSession.join()`
- **authextra** – Passed to `autobahn.wamp.protocol.ApplicationSession.join()`

---

**Note:** This function must be called instead of `AutobahnSync.run()` if we are calling from twisted application (typically if we are running our application inside crossbar as a `wsgi` component)

---

**session**

Return the underlying `session.SyncSession` object if available or raise an `exceptions.NotRunningError`



**stop ()**  
 Terminate the WAMP session

---

**Note:** If the `AutobahnSync.run ()` has been run with `blocking=True`, it will returns then.

---

**subscribe (topic, options=None)**  
 Decorator for the `AutobahnSync.session.subscribe ()`

---

**Note:** This decorator can be used before `AutobahnSync.run ()` is called. In such case the actual registration will be done at `run ()` time.

---

**class** `autobahn_sync.session.SyncSession (callbacks_runner, on_challenge_callback)`  
 Synchronous version of `autobahn.twisted.wamp.ApplicationSession`

**call (\*args, \*\*kwargs)**  
 Call a remote procedure.  
 Replace `autobahn.wamp.interface.IApplicationSession.call ()`

**leave (\*args, \*\*kwargs)**  
 Actively close this WAMP session.  
 Replace `autobahn.wamp.interface.IApplicationSession.leave ()`

**publish (\*args, \*\*kwargs)**  
 Publish an event to a topic.  
 Replace `autobahn.wamp.interface.IApplicationSession.publish ()`

**register (\*args, \*\*kwargs)**  
 Register a procedure for remote calling.  
 Replace `autobahn.wamp.interface.IApplicationSession.register ()`

**subscribe (\*args, \*\*kwargs)**  
 Subscribe to a topic for receiving events.  
 Replace `autobahn.wamp.interface.IApplicationSession.subscribe ()`

## Exceptions

---

**Note:** Most exceptions are republished from `autobahn.wamp.exception` and `twisted.internet.error`

---

**exception** `autobahn_sync.exceptions.Error`  
 Base class for all exceptions related to WAMP.

**exception** `autobahn_sync.exceptions.SessionNotReady`  
 The application tried to perform a WAMP interaction, but the session is not yet fully established.

**exception** `autobahn_sync.exceptions.SerializationError`  
 Exception raised when the WAMP serializer could not serialize the application payload (`args` or `kwargs` for `CALL`, `PUBLISH`, etc).

**exception** `autobahn_sync.exceptions.ProtocolError`  
 Exception raised when WAMP protocol was violated. Protocol errors are fatal and are handled by the WAMP implementation. They are not supposed to be handled at the application level.

**exception** `autobahn_sync.exceptions.TransportLost`

Exception raised when the transport underlying the WAMP session was lost or is not connected.

**exception** `autobahn_sync.exceptions.ApplicationError` (*error, \*args, \*\*kwargs*)

Base class for all exceptions that can/may be handled at the application level.

**error\_message** ()

Get the error message of this exception.

**Returns** The error message.

**Return type** `str`

**exception** `autobahn_sync.exceptions.NotAuthorized`

Not authorized to perform the respective action.

**exception** `autobahn_sync.exceptions.InvalidUri`

The URI for a topic, procedure or error is not a valid WAMP URI.

**exception** `autobahn_sync.exceptions.ConnectionRefusedError` (*osError=None, string=''*)

Connection was refused by other side

**exception** `autobahn_sync.exceptions.AbortError`

Error raised when the soutes respond with an ABORT message to our HELLO

**exception** `autobahn_sync.exceptions.AlreadyRunningError`

Error raised when trying to `run()` multiple time an `autobahn_sync.AutobahnSync`

**exception** `autobahn_sync.exceptions.NotRunningError`

Error raised when trying to `stop()` multiple time an `autobahn_sync.AutobahnSync`

## Flask extension

Flask extension provide an easier integration of Autobahn-Sync by doing it configuration in three ways (by order of priority):

- Configuration explicitly passed in `init_app`
- Configuration present in `app.config`
- Default configuration

Variable name	Description
<code>AUTOBAHN_ROUTER</code>	WAMP router to connect to (default: <code>ws://localhost:8080/ws</code> )
<code>AUTOBAHN_REALM</code>	WAMP realm to connect to (default: <code>realm1</code> )
<code>AUTOBAHN_IN_TWISTED</code>	Set to <code>true</code> if the code is going to run inside a Twisted application (default: <code>false</code> )

**class** `autobahn_sync.extensions.flask.FlaskAutobahnSync` (*app=None, \*\*config*)

Inherit from `autobahn_sync.AutobahnSync` to integrate it with Flask.

**Parameters**

- **app** – Flask app to configure, if provided `init_app()` is automatically called
- **config** – remaining kwargs will be passed to `init_app` as configuration

**init\_app** (*app, router=None, realm=None, in\_twisted=None*)

Configure and call the `AutobahnSync.start()` method

**Parameters**

- **app** – Flask app to configure
- **router** – WAMP router to connect to
- **realm** – WAMP realm to connect to
- **in\_twisted** – Is the code is going to run inside a Twisted application

---

**Note:** The config provided as argument will overwrite the one provided by `app.config`

---

Also see the [examples](#) for more usecases.



## CHAPTER 2

---

### Indices and tables

---

- [genindex](#)
- [modindex](#)
- [search](#)



**a**

`autobahn_sync.core`, 3  
`autobahn_sync.exceptions`, 5  
`autobahn_sync.extensions.flask`, 6  
`autobahn_sync.session`, 5





**A**

AbortError, 6  
 AlreadyRunningError, 6  
 ApplicationError, 6  
 autobahn\_sync.core (module), 3  
 autobahn\_sync.exceptions (module), 5  
 autobahn\_sync.extensions.flask (module), 6  
 autobahn\_sync.session (module), 5  
 AutobahnSync (class in autobahn\_sync.core), 3

**C**

call() (autobahn\_sync.session.SyncSession method), 5  
 ConnectionRefusedError, 6

**E**

Error, 5  
 error\_message() (autobahn\_sync.exceptions.ApplicationError method), 6

**F**

FlaskAutobahnSync (class in autobahn\_sync.extensions.flask), 6

**I**

init\_app() (autobahn\_sync.extensions.flask.FlaskAutobahnSync method), 6  
 InvalidUri, 6

**L**

leave() (autobahn\_sync.session.SyncSession method), 5

**N**

NotAuthorized, 6  
 NotRunningError, 6

**O**

on\_challenge() (autobahn\_sync.core.AutobahnSync method), 4

**P**

ProtocolError, 5  
 publish() (autobahn\_sync.session.SyncSession method), 5

**R**

register() (autobahn\_sync.core.AutobahnSync method), 4  
 register() (autobahn\_sync.session.SyncSession method), 5  
 run() (autobahn\_sync.core.AutobahnSync method), 4  
 run\_in\_twisted() (autobahn\_sync.core.AutobahnSync method), 4

**S**

SerializationError, 5  
 session (autobahn\_sync.core.AutobahnSync attribute), 4  
 SessionNotReady, 5  
 stop() (autobahn\_sync.core.AutobahnSync method), 4  
 subscribe() (autobahn\_sync.core.AutobahnSync method), 5  
 subscribe() (autobahn\_sync.session.SyncSession method), 5  
 SyncSession (class in autobahn\_sync.session), 5

**T**

TransportLost, 5