araldo Documentation

Release 0.1

Bernhard Biskup

November 24, 2012

CONTENTS

Contents:

Warning: This is alpha-stage software. Use with caution.

CHAPTER

ONE

ARALDO

araldo provides a simple, extensible communication relay (based on gevent and WebSockets) between multiple endpoints

Resources:

- Repository and issue tracker:
- Documentation of in-development version

1.1 Use cases

- Push notification from server to Browser
- Bidirectional communication between browsers
- Bidirectional communication between browsers and a backend
- Interoperability between different message-oriented middleware products

1.2 Architecture

araldo is a central server component that provides an arbitrary amount of endpoints to which clients may connect, or which connect to other servers.

1.2.1 Routes

araldo uses the concept of *routes* for setting up communication. Rather than relying on clients subscribing/publishing to/from particular channels, the channel setup and routing between endpoints is done inside *araldo*.

1.2.2 Implementation

araldo uses gevent, a corouting-based networking library. *gevent* uses green threads (greenlets) and asynchronous IO. Therefore, the *araldo* server is single-threaded with respect to native threads, but highly concurrent with respect to green threads.

1.2.3 Plugins

Available Plugins

araldo consists of a core that is run as a server process, and several plugins. Available plugins are loaded automatically. Out of the box, *araldo* provides a WebSocket server that allows multiple clients to communicate with each other via routes.

Currently the following plugins are available:

Name	Description	Link
araldo-redis	Communication via Redis PubSub	http://bitbucket.org/ganymed/araldo-redis
araldo-websocket	Communication with an HTML5 WebSocket server	http://bitbucket.org/ganymed/araldo-websocket

Extensibility

araldo uses the Python entry-point mechanism for plugin discovery.

1.3 Installation

To install *araldo* along with some plugins, type:

```
pip install araldo
```

1.4 Configuration

Command line parameters:

```
Usage: server.py [options]

Options:
  -h, --help                  show this help message and exit
  -c CONFIG, --config=CONFIG
                              Configuration path
```

1.4.1 Configuration File

The configuration file in *YAML* format mainly contains

- global settings
- plugin instances, with plugin-specific settings
- routes (connections between endpoints)

The standard configuration file is `araldo.yaml` in the current directory.

1.4.2 Global parameters

Parameter	Type	Description
server-port	int	TCP port of web (WebSocket) server
log-level	str	One of `debug`, `info`, `warning`, `error`, `fatal`. Verbosity of logging

1.4.3 Plugin-Specific parameters

Plugins are configured under the toplevel configuration key `plugins`.

All plugins

Parameter	Type	Description
name	str	Unique name for the plugin instance
id	str	Identifier for the plugin type

1.4.4 Routing

Routes map messages from an inbound source to multiple outbound endpoints. Routing is configured under the toplevel configuration key `routes`. Each route consists of a key that references a name of an endpoint. The value is a list of names referencing other endpoints.

Sample configuration:

```
global:
    server-port: 54321
    log-level: debug

plugins:
  araldo.marshalling:
    - name: marshal-json
      id: marshal-json
  araldo.endpoints.endpoint:
    - name: mock_1
      id: endpoint-mock
      channel: channel_1
    - name: redis_2
      id: endpoint-mock
      channel: channel_2
    - name: redis_3
      id: endpoint-mock
      channel: channel_3

routes:
  redis_1:
    - redis_2
    - redis_3
```

1.5 Platform & System Requirements

- *araldo* has been tested under Python 2.6 and 2.7. Python 3.x is not supported because not all libraries it depends on currently support 3.x.
- *araldo* was tested exclusively under Linux.

CHAPTER

TWO

API

Top-level application package

2.1 Application

Handles WebSocket requests and HTTP requests

exception araldo.app.**AppException**(*msg*)
An internal exception

exception araldo.app.**TooLong**
Indicates a gevent timeout

```
>>> t = TooLong()
```

class araldo.app.**WebSocketApp**(*config*, *queue*, *plugin_manager*)
Provides bidirectional communication: - inbound via WebSocket - outbound via regular HTTP request

class araldo.app.**WebSocketApp**(*config*, *queue*, *plugin_manager*)
Provides bidirectional communication: - inbound via WebSocket - outbound via regular HTTP request

__call__(*environ*, *start_response*)
main WSGI method

2.2 Server

Launches WSGI server process hosting WebSocketApp

araldo.server.**main**()
Server main method

araldo.server.**parse_args**(*args*)
Get parser for command line parameters

araldo.server.**setup_logging**(*config*)
Configure logger, log level, etc.

araldo.server.**setup_plugins**(*logger*, *config*)
Load and instantiate plugins

araldo.server.**setup_sending**(*config*, *queue*, *plugin_manager*)
Set up sending of outbound messages

araldo.server.**setup_signals**()
Setup OS signals for graceful termination of server

araldo.server.**sig_handler**(*signum*, *frame*)
Handles abortion; e.g. by pressing CTRL+C

araldo.server.**start_server**(*config*, *port*, *queue*, *plugin_manager*, *start_method=<function <lambda> at 0x42c4b90>*)
Launches WSGI server that will listen forever

2.3 Communication Endpoints

Communication endpoints

class araldo.endpoints.**EndPointBase**(***kwargs*)
Abstract base class for Araldo endpoints

Concrete classes must implement Greenlet's _run method to process incoming messages and to enque them into gevent_queue

config()
Plugin configuration sub-object

description()
Textual description of plugin

gevent_queue
gevent target queue

marshalling()
Short, human-readable name marshalling used

name()
Short, human-readable name of plugin

plugin_manager()
plugin manager for loading other plugins

send(*message*)
Send message to backend

exception araldo.endpoints.**PluginException**(*msg*)
An end-point related exception

CHAPTER

THREE

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

PYTHON MODULE INDEX

a