gotana Documentation

Release 0.1

Jacek Nosal

Apr 20, 2018

Contents

CHAPTER 1

Configuration

1.1 project

Default: `This parameter is mandatory`

```
Name of the project used internally by the engine
```

1.2 tcpaddress

Default: `Optional parameter`

```
Host and Port combination that telnet console will bind to, e.g: localhost:7654
```

1.3 httpaddress

Default: `Optional parameter`

```
Host and Port combination that HTTP API server will bind to, e.g: localhost:5555
```

1.4 redisaddress

Default: `Optional parameter`

```
Host and Port combination of redis server, which is required for http api frontend as␣
↪well as storage.
```

1.5 scrapers

Default: `This parameter is mandatory`

```
List of scrapers that will be executed by the engine
```

CHAPTER 2

Scraper Configuration

2.1 name

Default: `This parameter is mandatory`

```
Internal name of the scraper
```

2.2 url

Default: `This parameter is mandatory`

```
Base url which will be used to start crawling
```

2.3 requestlimit

Default: `1 millisecond`

```
Number of millisecond to wait between requests
```

2.4 patterns

Default: `Optional parameter`

```
List of patterns to validate url that's currently being scraped against. See patterns␣
↪configuration.
```

2.5 extractor

Default: `Optional parameter`

```
Short name of extractor struct which implements Extractable interface, by defualt␣
↪LinkExtractor (link) is used.
```

CHAPTER 3

Patterns Configuration

3.1 type

Default: `This parameter is mandatory`

```
Either contains or regexp. First one uses string matching, the latter relies on␣
↪regular expression.
```

3.2 pattern

Default: `This parameter is mandatory`

```
Value that's used as string to match against or regexp expression depending on the␣
↪type of pattern.
```

CHAPTER 4

Example configuration

```
project: test
tcpaddress: localhost:7654
redisaddress: localhost:6379
httpaddress: localhost:5555
scrapers:
- name: golang
  url: http://golangweekly.com
  requestlimit: 200
  patterns:
  - type: contains
    pattern: /issues
- name: scrapinghub
  url: https://blog.scrapinghub.com
  requestlimit: 200
```

CHAPTER 5

Exports

CHAPTER 6

Extensions

CHAPTER 7

Items

CHAPTER 8

Middleware

CHAPTER 9

Patterns

CHAPTER 10

TCP Server

Gotana offers telnet console for inspecting crawlers and controlling the engine. The idea behind this service is to offer simple remote control over the scrapers.

10.1 Available commands

Name	Description
HELP	Displays list of available commands
LIST	Displays lists of available scrapers
STATS	Displays statistics of currently running scrapers
MIDDLEWARE	Display installed middleware
EXTENSIONS	Display installed extensions

10.2 Usage

```
telnet localhost 7654
HELP
-------------------------------------
Available commands: LIST, STATS, HELP, STOP
STATS
-------------------------------------
Total scrapers: 1. Total requests: 45
-------------------------------------
-------------------------------------------------------------------------------------------
↪----
<Scraper: golangweekly.com>. Crawled: 45, successful: 44, failed: 1. Scraped: 44,␣
↪saved: 9
-------------------------------------------------------------------------------------------
↪----
-------------------------------------------------------
```

```
Currently fetching: http://golangweekly.com/rss/14p9ef33
--------------------------------------------------------
LIST
------------------------
Running scrapers: golang
STOP
--------------------
Stopping scrapers...
```

10.3 Configuration

10.3.1 tcpaddress

Default: `Optional parameter`

```
Host and Port combination that telnet console will bind to, e.g: localhost:7654
```

CHAPTER 11

Tutorial

11.1 Tutorial

This tutorial introduces **gotana** core concepts by example.

11.1.1 Getting started

11.1.2 Basic usage

CHAPTER 12

Info

12.1 Additional