21.10. `urllib.robotparser` — Parser for robots.txt

Source code: Lib/urllib/robotparser.py

This module provides a single class, `RobotFileParser`, which answers questions about whether or not a particular user agent can fetch a URL on the Web site that published the `robots.txt` file. For more details on the structure of `robots.txt` files, see http://www.robotstxt.org/orig.html.

class `urllib.robotparser.`**`RobotFileParser`**(*url=''*)

This class provides methods to read, parse and answer questions about the `robots.txt` file at *url*.

`set_url`(*url*)

Sets the URL referring to a `robots.txt` file.

`read`()

Reads the `robots.txt` URL and feeds it to the parser.

`parse`(*lines*)

Parses the lines argument.

`can_fetch`(*useragent*, *url*)

Returns `True` if the *useragent* is allowed to fetch the *url* according to the rules contained in the parsed `robots.txt` file.

`mtime`()

Returns the time the `robots.txt` file was last fetched. This is useful for long-running web spiders that need to check for new `robots.txt` files periodically.

`modified`()

Sets the time the `robots.txt` file was last fetched to the current time.

`crawl_delay`(*useragent*)

Returns the value of the `Crawl-delay` parameter from `robots.txt` for the *useragent* in question. If there is no such parameter or it doesn't apply to the *useragent* specified or the `robots.txt` entry for this parameter has invalid syntax, return `None`.

New in version 3.6.

request_rate(*useragent*)

Returns the contents of the `Request-rate` parameter from `robots.txt` as a named tuple `RequestRate(requests, seconds)`. If there is no such parameter or it doesn't apply to the *useragent* specified or the `robots.txt` entry for this parameter has invalid syntax, return `None`.

New in version 3.6.

The following example demonstrates basic use of the `RobotFileParser` class:

```
>>> import urllib.robotparser
>>> rp = urllib.robotparser.RobotFileParser()
>>> rp.set_url("http://www.musi-cal.com/robots.txt")
>>> rp.read()
>>> rrate = rp.request_rate("*")
>>> rrate.requests
3
>>> rrate.seconds
20
>>> rp.crawl_delay("*")
6
>>> rp.can_fetch("*", "http://www.musi-cal.com/cgi-bin/search?city=Sar
False
>>> rp.can_fetch("*", "http://www.musi-cal.com/")
True
```