

toreator

Tor relay parser and UI

Developer documentation

Libor Polčák



TARZAN project VI20172020062 document
Faculty of Information Technology, Brno University of Technology

Last change: April 7, 2020

toreator— Developer documentation

Libor Polčák

Faculty of Information Technology, Brno University of Technology, e-mail:
`polcak@fit.vutbr.cz`

This software processes data about Tor network. The main source is Tor CollecTor¹, which is available for research purposes. The tool augment the data with data from MaxMind GeoLite2 data² and data fetched from public DNS.

CollecTor contains all registered public nodes that were usable as entry nodes, exit nodes, and relays. Neither Tor bridges nor Tor users are stored in the database.

Conceptually, this tool was inspired by the service Exonerator³ and Torstatus⁴. This database differs from:

- *Exonerator* because it contains more details about the Tor nodes;
- *Torsatus* because it contains historical data.

The database is available as a REST API⁵ and allows to fetch structured data as JSON (for machine processing) and HTML (for human processing). Data encoding is controlled by the `Accept` header.

You should set up the program (see install guide) so that the database is automatically updated. CollecTor updates once per hour. MaxMind updates once per week (ASN database) or month (location database).

1 REST API endpoints

The API contains the following endpoints:

- Route `’/’`
- Route `’/addresses/’`
- Route `’/addresses/<address_detail>/’`
- Route `’/addresses/<addr>/date/’`
- Route `’/addresses/<addr>/date/<date>/’`
- Route `’/addresses/<addr>/time/’`
- Route `’/addresses/<addr>/time/<time>/’`

¹ <https://metrics.torproject.org/collector.html>

² <http://www.maxmind.com/>

³ <https://exonerator.torproject.org/>

⁴ <http://torstatus.blutmagie.de/>

⁵ You can try <http://toreator.fit.vutbr.cz/> for an example instance of the API. The FIT Toreator database was created as part of the *Integrated platform for analysis of digital data from security incidents* project. This database contains data about Tor network since January 2008.

- Route `'/addresses/<addr>/month/'`
- Route `'/addresses/<addr>/month/<month>/'`
- Route `'/addresses/<addr>/year/'`
- Route `'/addresses/<addr>/year/<year>/'`

The endpoints contains the following parameters:

- `<addr>` is an IPv4 or IPv6⁶ address.
- `<address_detail>/'` can be:
 - `<addr>`: displays all information known about the IP address,
 - `<network_address>-<prefix_len>` where `<network_address>` is an IPv4 or IPv6 network and `<prefix_len>` si the number of bits in the network prefix (e.g. 147.229.0.0-16).
 Depending on the address family and prefix length:
 - * For IPv4:
 - Prefix length `/0-7` displays all `/8` networks with at least one known Tor relay.
 - Prefix length `/8-15` displays all `/16` networks with at least one known Tor relay.
 - Prefix length `/16-32` displays all known Tor relays in the network.
 - * For IPv6:
 - Prefix length `/0-15` displays all `/16` networks with at least one known Tor relay.
 - Prefix length `/16-31` displays all `/32` networks with at least one known Tor relay.
 - Prefix length `/32-128` displays all known Tor relays in the network.
- `<date>` is in the format `%Y-%m-%d`.
- `<time>` is in the format `%Y-%m-%d %H:%M:%S`.
- `<month>` is in the format `%Y-%m`.
- `<year>` is a 4-digit year.

2 Return value

The result HTTP body for JSON encoding contains an object with keys:

- `result`: object containing data depending on the endpoint type.
- `related`: list of links where the inner list contains a description of the related resources and the URL of the resource.

⁶ <https://tools.ietf.org/html/rfc5952>

The items describing Tor nodes are inspired by the Tor directory protocol⁷. For example, `inconsensus_val_after` represents the `valid-after` time⁸, `supported_proto` represents the `proto`⁹.

Besides the items from Tor directory protocol, the Toreator database contains (if available):

1. `reverse_name` is an object containing:
 - `dns`: contains a string with the reverse domain name,
 - `queried`: contains a list of timestamps (date, time string) specifying the time of DNS query (note that the queries might be DNS-cached).
2. `geolite_maxmind_geolocation`: object with data from the MaxMind database including the `geolocation_source` attribute giving credit the database.

3 Examples

```
$ curl -H "Accept: application/json" \  
http://toreator.fit.vutbr.cz/addresses/147.229.13.223/  
{  
  "result": [  
    {  
      "inconsensus_val_until": [  
        "2013-04-03",  
        "00:00:00"  
      ],  
      "nickname": "default",  
      "bandwidth": {  
        "Unmeasured": "1",  
        "Bandwidth": "20"  
      },  
      "publication": [  
        "2013-04-02",  
        "20:08:40"  
      ],  
      "inconsensus_val_after": [  
        "2013-04-02",  
        "21:00:00"  
      ],  
      "dirport": "9030",  
      "digest": "pkYCSI2r79XMzeFFVLbF+73Sr3w",  
      "identity": "M5dvWj/g37C7nYQ2QHw9bA1QNT8",  
      "orport": "443",  
      "flags": [  

```

⁷ <https://gitweb.torproject.org/torspec.git/tree/dir-spec.txt>

⁸ <https://gitweb.torproject.org/torspec.git/tree/dir-spec.txt#n309>

⁹ <https://gitweb.torproject.org/torspec.git/tree/dir-spec.txt#n820>

```

"Exit",
"Fast",
"Running",
"V2Dir",
"Valid"
],
"inconsensus_fresh_until": [
"2013-04-02",
"22:00:00"
],
"allow_ports": "reject 25,119,135-139,445,563,1214,4661-4666,6346-6429,6699,6881-6999",
"ip": "147.229.13.223",
"version": "Tor 0.2.3.25"
},
{
"inconsensus_val_until": [
"2013-04-03",
"09:00:00"
],
"nickname": "default",
"bandwidth": {
"Unmeasured": "1",
"Bandwidth": "20"
},
"publication": [
"2013-04-02",
"21:39:09"
],
"inconsensus_val_after": [
"2013-04-02",
"22:00:00"
],
...

```

```

$ curl -H "Accept: application/json" \
"http://toreator.fit.vutbr.cz/addresses/193.23.244.244/time/2020-04-01%2012:00:00/"
{
"result": [
{
"digest": "rIQiTgTCxfwKlD3u537s8QhWlaE",
"nickname": "dannenbergl",
"bandwidth": {
"Bandwidth": "13"
},
"orport": "443",
"geolite_maxmind_geolocation": {

```

```
"2020-03-03 00:00:00": [
  [
    "geolocation source",
    "This product includes GeoLite2 data created by MaxMind, available from <a href=\"http://www.maxmind.com\">http://www.maxmind.com</a>.",
  ],
  [
    "network",
    "193.23.244.0/24"
  ],
  [
    "continent",
    "Europe"
  ],
  [
    "country code",
    "DE"
  ],
  [
    "country",
    "Germany"
  ],
  [
    "time zone",
    "Europe/Berlin"
  ],
  [
    "inside EU",
    "True"
  ],
  [
    "timestamp",
    "2020-03-03 00:00:00"
  ]
],
"publication": [
  "2020-04-01",
  "07:26:10"
],
"supported_proto": [
  "Cons=1-2",
  "Desc=1-2",
  "DirCache=1-2",
  "HSDir=1-2",
  "HSIntro=3-5",
]
```

```

"HSRend=1-2",
"Link=1-5",
"LinkAuth=1,3",
"Microdesc=1-2",
"Relay=1-2",
"Padding=2",
"FlowCtrl=1"
],
"geolite_maxmind_as": {
"2020-03-31 00:00:00": [
[
"geolocation source",
"This product includes GeoLite2 data created by MaxMind,
available from <a href=\"http://www.maxmind.com\">http://www.maxmind.com</a>."
],
[
"AS network",
"193.23.244.0/24"
],
[
"AS number",
"50472"
],
[
"AS organization",
"Chaos Computer Club e.V."
],
[
"timestamp",
"2020-03-31 00:00:00"
]
]
},
"inconsensus_val_after": [
"2020-04-01",
"09:00:00"
],
"dirport": "80",
"inconsensus_fresh_until": [
"2020-04-01",
"13:00:00"
],
"inconsensus_val_until": [
"2020-04-01",
"15:00:00"
]

```



```
],
"identity": "e+aD5l1IFBMhxe2S8HXFU2SscSM",
"reverse_name": {
"dns": "dannenbergtorauth.de",
"queried": [
[
"2020-04-01",
"09:15:03"
],
[
"2020-04-01",
"10:22:17"
],
[
"2020-04-01",
"11:13:29"
],
[
"2020-04-01",
"12:15:06"
]
]
},
"flags": [
"Authority",
"Running",
"Stable",
"V2Dir",
"Valid"
],
"ipv6": [
[
"2001:678:558:1000::244",
"443"
]
],
"allow_ports": "reject 1-65535",
"ip": "193.23.244.244",
"version": "Tor 0.4.2.7"
}
],
"related": [
[
"Show the IP address 193.23.244.244 activity on 2020-04-01",
"/addresses/193.23.244.244/date/2020-04-01/"
]
]
```

```
],
[
  "Show the IP address 193.23.244.244 activity in 2020-04",
  "/addresses/193.23.244.244/month/2020-04/"
],
[
  "Show the IP address 193.23.244.244 activity in 2020",
  "/addresses/193.23.244.244/year/2020/"
],
[
  "Show dates when the IP address 193.23.244.244 was active",
  "/addresses/193.23.244.244/date/"
],
[
  "Show months when the IP address 193.23.244.244 was active",
  "/addresses/193.23.244.244/month/"
],
[
  "Show years when the IP address 193.23.244.244 was active",
  "/addresses/193.23.244.244/year/"
],
[
  "Show all information about the IP address 193.23.244.244",
  "/addresses/193.23.244.244/"
],
[
  "Search by IP addresses",
  "/addresses/"
]
]
}
```

Toreator Consensus Parser

Generated by Doxygen 1.8.16

1 Namespace Index	1
1.1 Namespace List	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	5
3.1 Class List	5
4 File Index	7
4.1 File List	7
5 Namespace Documentation	9
5.1 consensus_parser Namespace Reference	9
5.1.1 Function Documentation	9
5.1.1.1 add_geolite_data_to_onion_routers()	10
5.1.1.2 create_snapshot_from_consensus_file()	10
5.1.1.3 download_consensus_if_missing()	10
5.1.1.4 find_preprocessed_ip_address()	11
5.1.1.5 find_preprocessed_ip_address_time_filter()	12
5.1.1.6 fixup_missing_consensus()	12
5.1.1.7 get_consensus_fullpath()	13
5.1.1.8 get_ip_address_activity()	14
5.1.1.9 get_last_preprocessed_consensus()	14
5.1.1.10 main()	14
5.1.1.11 merge_subsequent_ors()	15
5.1.1.12 output_preprocessed_ip_addresses_time_filter()	15
5.1.1.13 parse_ors_from_file()	16
5.1.1.14 preprocess_directory_consensus()	16
5.1.1.15 preprocess_path_consensus()	17
5.1.1.16 process_args()	18
5.1.1.17 process_path_consensus()	18
5.1.1.18 update_path_consensus()	18
5.1.1.19 write_preprocessed()	19
5.1.1.20 write_preprocessed_from_consensus_file()	19
5.1.2 Variable Documentation	20
5.1.2.1 args	20
5.1.2.2 WAIT_FOR_CONSENSUS	20
5.2 dns Namespace Reference	20
5.2.1 Function Documentation	20
5.2.1.1 ip_addr_to_domain()	21
5.3 geolite2 Namespace Reference	21
5.3.1 Function Documentation	21
5.3.1.1 process_args()	21

5.3.2 Variable Documentation	22
5.3.2.1 args	22
5.3.2.2 gl	22
5.3.2.3 t	22
5.4 ipfiles Namespace Reference	22
5.4.1 Function Documentation	22
5.4.1.1 __find_addresses()	22
5.4.1.2 create_ip_filename()	23
5.4.1.3 get_ranges_addresses()	23
5.4.1.4 get_ranges_addresses_ipv4()	24
5.4.1.5 get_ranges_addresses_ipv6()	24
5.5 onion_router Namespace Reference	25
5.6 parametrizable_tc Namespace Reference	25
5.6.1 Function Documentation	25
5.6.1.1 parametrizable_test()	25
5.7 preprocess_geolite Namespace Reference	25
5.7.1 Function Documentation	26
5.7.1.1 main()	26
5.7.1.2 process_args()	26
5.7.1.3 process_file()	27
5.7.1.4 process_file4()	27
5.7.1.5 process_file6()	28
5.7.1.6 process_file_loc()	29
5.7.1.7 write_location()	29
5.7.1.8 write_network()	30
5.7.2 Variable Documentation	30
5.7.2.1 args	30
5.7.2.2 ASN_FIRST_LINE_CSV	30
5.7.2.3 CITY_FIRST_LINE_CSV	30
5.7.2.4 LOCATION_FIRST_LINE_CSV	30
5.8 profilerest Namespace Reference	31
5.8.1 Variable Documentation	31
5.8.1.1 debug	31
5.8.1.2 restrictions	31
5.8.1.3 sort_by	31
5.8.1.4 wsgi_app	31
5.9 rest Namespace Reference	31
5.9.1 Function Documentation	32
5.9.1.1 entry_point()	32
5.9.1.2 enum_addresses_generic()	32
5.9.1.3 evaluate_accept_header_convert_json()	33
5.9.1.4 htmlize()	33

5.9.1.5 process_route_addresses()	34
5.9.1.6 request_wants_json()	34
5.9.1.7 show_all_ip_address_information()	35
5.9.1.8 show_ip_address_active_months()	35
5.9.1.9 show_ip_address_active_years()	35
5.9.1.10 show_ip_address_activity_at_specific_time()	35
5.9.1.11 show_ip_address_activity_dates()	35
5.9.1.12 show_ip_address_activity_during_specific_month()	36
5.9.1.13 show_ip_address_activity_during_specific_year()	36
5.9.1.14 show_ip_address_activity_on_specific_date()	36
5.9.1.15 show_ip_address_activity_times()	36
5.9.1.16 show_known_address_ranges()	36
5.9.1.17 show_known_address_subranges()	37
5.9.2 Variable Documentation	37
5.9.2.1 app	37
5.9.2.2 GEOLITE_DIR	37
5.9.2.3 PREPORCESSED_DIR	37
5.10 serializable Namespace Reference	38
5.11 snapshot Namespace Reference	38
5.12 test Namespace Reference	38
5.12.1 Function Documentation	38
5.12.1.1 suite()	38
5.12.1.2 test()	39
5.12.2 Variable Documentation	40
5.12.2.1 verbosity	40
5.13 test_additional_info Namespace Reference	41
5.13.1 Function Documentation	41
5.13.1.1 suite()	41
5.13.1.2 test()	41
5.13.2 Variable Documentation	42
5.13.2.1 verbosity	42
5.14 test_cp Namespace Reference	42
5.14.1 Function Documentation	42
5.14.1.1 suite()	42
5.14.1.2 test()	43
5.14.2 Variable Documentation	43
5.14.2.1 CONSENSUS_PATH	43
5.14.2.2 PCVESELY_JSON	43
5.14.2.3 PREPROCESSED_DIR_CREATE	43
5.14.2.4 verbosity	43
5.15 test_files Namespace Reference	44
5.15.1 Function Documentation	44

5.15.1.1 suite()	44
5.15.1.2 test()	44
5.15.2 Variable Documentation	45
5.15.2.1 verbosity	45
5.16 test_geolite2 Namespace Reference	45
5.16.1 Function Documentation	45
5.16.1.1 suite()	45
5.16.1.2 test()	46
5.16.2 Variable Documentation	46
5.16.2.1 t20180605	46
5.16.2.2 t20180619	46
5.16.2.3 t20180626	46
5.16.2.4 t20180703	46
5.16.2.5 t20180709	47
5.16.2.6 verbosity	47
5.17 test_or Namespace Reference	47
5.17.1 Function Documentation	47
5.17.1.1 suite()	47
5.17.1.2 test()	48
5.17.2 Variable Documentation	48
5.17.2.1 verbosity	48
5.18 test_rest Namespace Reference	48
5.18.1 Function Documentation	49
5.18.1.1 suite()	49
5.18.1.2 test()	49
5.18.2 Variable Documentation	49
5.18.2.1 bs_failed	49
5.18.2.2 bs_required_text	50
5.18.2.3 flask_failed	50
5.18.2.4 rest	50
5.18.2.5 super_class	50
5.18.2.6 verbosity	50
5.19 test_serializable Namespace Reference	50
5.19.1 Function Documentation	51
5.19.1.1 suite()	51
5.19.1.2 test()	51
5.19.2 Variable Documentation	51
5.19.2.1 bs_failed	51
5.19.2.2 bs_required_text	52
5.19.2.3 verbosity	52
5.20 test_time Namespace Reference	52
5.20.1 Function Documentation	52

5.20.1.1 suite()	52
5.20.1.2 test()	53
5.20.2 Variable Documentation	53
5.20.2.1 verbosity	53
5.21 time_parser Namespace Reference	53
5.21.1 Variable Documentation	53
5.21.1.1 DAY_HAS_SECONDS	54
5.21.1.2 months	54
6 Class Documentation	55
6.1 geolite2.geolite2_accessor.db_info Class Reference	55
6.1.1 Constructor & Destructor Documentation	55
6.1.1.1 __init__()	56
6.1.2 Member Data Documentation	56
6.1.2.1 cache	56
6.1.2.2 dirs	56
6.1.2.3 last	56
6.2 test_rest.fake Class Reference	56
6.2.1 Member Function Documentation	57
6.2.1.1 __getattr__()	57
6.3 test_cp.fake_cp_args Class Reference	57
6.3.1 Constructor & Destructor Documentation	58
6.3.1.1 __init__()	58
6.3.2 Member Data Documentation	58
6.3.2.1 additional_info	58
6.3.2.2 consensus_path	58
6.3.2.3 date_prefix	58
6.3.2.4 fixup	59
6.3.2.5 geolite_dir	59
6.3.2.6 ipaddress	59
6.3.2.7 preprocessed_input	59
6.3.2.8 time	59
6.3.2.9 update	59
6.3.2.10 update_keeprunning	59
6.3.2.11 write_preprocessed	60
6.4 time_parser.FormatTimeWrapper Class Reference	60
6.4.1 Detailed Description	61
6.4.2 Constructor & Destructor Documentation	61
6.4.2.1 __init__()	62
6.4.3 Member Data Documentation	62
6.4.3.1 v	62
6.5 geolite2.geolite2_accessor Class Reference	62

6.5.1 Constructor & Destructor Documentation	63
6.5.1.1 <code>__init__()</code>	63
6.5.2 Member Function Documentation	63
6.5.2.1 <code>__get_blocks_matching_func()</code>	64
6.5.2.2 <code>__get_directories()</code>	64
6.5.2.3 <code>__get_geolite2_row()</code>	64
6.5.2.4 <code>__get_preprocessed_filename_ipaddr()</code>	65
6.5.2.5 <code>__get_preprocessed_filename_location()</code>	65
6.5.2.6 <code>create_ordered_dict_base()</code>	65
6.5.2.7 <code>get_data()</code>	66
6.5.2.8 <code>get_geolite2_as()</code>	66
6.5.2.9 <code>get_geolite2_geolocation()</code>	67
6.5.3 Member Data Documentation	67
6.5.3.1 <code>__geolite2_dbs</code>	67
6.5.3.2 <code>__geolite2_dir</code>	68
6.5.3.3 <code>__search_func</code>	68
6.6 metaclass Class Reference	68
6.7 onion_router.onion_router Class Reference	69
6.7.1 Detailed Description	71
6.7.2 Constructor & Destructor Documentation	71
6.7.2.1 <code>__init__()</code>	71
6.7.3 Member Function Documentation	71
6.7.3.1 <code>add_geolite_data()</code>	72
6.7.3.2 <code>add_network_status_consensus()</code>	72
6.7.3.3 <code>append_geolite_data()</code>	72
6.7.3.4 <code>append_lines()</code>	73
6.7.3.5 <code>get_allow_ports()</code>	73
6.7.3.6 <code>get_bandwidth()</code>	73
6.7.3.7 <code>get_digest()</code>	73
6.7.3.8 <code>get_dirport()</code>	74
6.7.3.9 <code>get_dns_reverse()</code>	74
6.7.3.10 <code>get_flags()</code>	74
6.7.3.11 <code>get_identity()</code>	74
6.7.3.12 <code>get_inconsensus_fresh_until()</code>	74
6.7.3.13 <code>get_inconsensus_fresh_until_unixtime()</code>	74
6.7.3.14 <code>get_inconsensus_val_after()</code>	75
6.7.3.15 <code>get_inconsensus_val_after_unixtime()</code>	75
6.7.3.16 <code>get_inconsensus_val_until()</code>	75
6.7.3.17 <code>get_inconsensus_val_until_unixtime()</code>	75
6.7.3.18 <code>get_ip()</code>	76
6.7.3.19 <code>get_ipv6()</code>	76
6.7.3.20 <code>get_json_dict()</code>	76

6.7.3.21	<code>get_maxmind_asn()</code>	76
6.7.3.22	<code>get_maxmind_geolocation()</code>	76
6.7.3.23	<code>get_nickname()</code>	77
6.7.3.24	<code>get_orport()</code>	77
6.7.3.25	<code>get_publication()</code>	77
6.7.3.26	<code>get_supported_proto()</code>	77
6.7.3.27	<code>get_version()</code>	77
6.7.3.28	<code>merge()</code>	78
6.7.3.29	<code>output_json()</code>	78
6.7.3.30	<code>parse_a()</code>	79
6.7.3.31	<code>parse_id()</code>	79
6.7.3.32	<code>parse_m()</code>	79
6.7.3.33	<code>parse_p()</code>	79
6.7.3.34	<code>parse_pr()</code>	79
6.7.3.35	<code>parse_r()</code>	79
6.7.3.36	<code>parse_s()</code>	80
6.7.3.37	<code>parse_v()</code>	80
6.7.3.38	<code>parse_w()</code>	80
6.7.3.39	<code>parse_x_dns_reverse()</code>	80
6.7.3.40	<code>parse_x_inconsensus_fresh_until()</code>	80
6.7.3.41	<code>parse_x_inconsensus_valid_after()</code>	80
6.7.3.42	<code>parse_x_inconsensus_valid_until()</code>	81
6.7.4	Member Data Documentation	81
6.7.4.1	<code>__all_lines</code>	81
6.7.4.2	<code>__allow_ports</code>	81
6.7.4.3	<code>__bandwidth</code>	81
6.7.4.4	<code>__digest</code>	81
6.7.4.5	<code>__dirport</code>	81
6.7.4.6	<code>__flags</code>	81
6.7.4.7	<code>__identity</code>	82
6.7.4.8	<code>__inconsensus_fresh_until</code>	82
6.7.4.9	<code>__inconsensus_fresh_until_unixtime</code>	82
6.7.4.10	<code>__inconsensus_val_after</code>	82
6.7.4.11	<code>__inconsensus_val_after_unixtime</code>	82
6.7.4.12	<code>__inconsensus_val_until</code>	82
6.7.4.13	<code>__inconsensus_val_until_unixtime</code>	82
6.7.4.14	<code>__ip</code>	82
6.7.4.15	<code>__ipv6</code>	83
6.7.4.16	<code>__nickname</code>	83
6.7.4.17	<code>__orport</code>	83
6.7.4.18	<code>__publication</code>	83
6.7.4.19	<code>__reverse_name</code>	83

6.7.4.20	<code>__supported_proto</code>	83
6.7.4.21	<code>__version</code>	83
6.7.4.22	<code>merge_attrs</code>	84
6.7.4.23	<code>request_additional</code>	84
6.8	<code>parametrizable_tc.parametrizable_tc_meta</code> Class Reference	84
6.8.1	Member Function Documentation	85
6.8.1.1	<code>__new__()</code>	85
6.9	<code>serializable.serializable_links</code> Class Reference	87
6.9.1	Detailed Description	88
6.9.2	Constructor & Destructor Documentation	88
6.9.2.1	<code>__init__()</code>	88
6.9.3	Member Function Documentation	89
6.9.3.1	<code>htmlize_lines()</code>	89
6.9.3.2	<code>jsonify()</code>	89
6.10	<code>serializable.serializable_list</code> Class Reference	90
6.10.1	Detailed Description	91
6.10.2	Constructor & Destructor Documentation	91
6.10.2.1	<code>__init__()</code>	91
6.10.3	Member Function Documentation	91
6.10.3.1	<code>_make_html_link()</code>	92
6.10.3.2	<code>append()</code>	92
6.10.3.3	<code>extend()</code>	92
6.10.3.4	<code>htmlize_lines()</code>	92
6.10.3.5	<code>jsonify()</code>	93
6.10.4	Member Data Documentation	93
6.10.4.1	<code>__class__</code>	93
6.10.4.2	<code>_wrapped</code>	93
6.11	<code>serializable.serializable_ors</code> Class Reference	94
6.11.1	Detailed Description	96
6.11.2	Constructor & Destructor Documentation	96
6.11.2.1	<code>__init__()</code>	96
6.11.3	Member Function Documentation	96
6.11.3.1	<code>__htmlize_or()</code>	96
6.11.3.2	<code>htmlize_lines()</code>	97
6.11.3.3	<code>jsonify()</code>	97
6.11.4	Member Data Documentation	97
6.11.4.1	<code>__addressbase</code>	97
6.11.4.2	<code>__datestr</code>	97
6.11.4.3	<code>__ipstr</code>	98
6.11.4.4	<code>__maxmind</code>	98
6.11.4.5	<code>__timestr</code>	98
6.12	<code>snapshot.Snapshot</code> Class Reference	98

6.12.1 Constructor & Destructor Documentation	99
6.12.1.1 <code>__init__()</code>	99
6.12.2 Member Function Documentation	99
6.12.2.1 <code>get_ipaddrs()</code>	99
6.12.2.2 <code>get_onion_router()</code>	99
6.12.2.3 <code>get_valid_after()</code>	100
6.12.2.4 <code>parse_fresh_until()</code>	100
6.12.2.5 <code>parse_valid_after()</code>	100
6.12.2.6 <code>parse_valid_until()</code>	100
6.12.3 Member Data Documentation	100
6.12.3.1 <code>__fresh_until</code>	100
6.12.3.2 <code>__ip</code>	100
6.12.3.3 <code>__obj_lines</code>	101
6.12.3.4 <code>__routers</code>	101
6.12.3.5 <code>__unparsed_lines</code>	101
6.12.3.6 <code>__valid_after</code>	101
6.12.3.7 <code>__valid_until</code>	101
6.13 <code>super_class</code> Class Reference	102
6.14 <code>test_additional_info.test_additional_info</code> Class Reference	103
6.14.1 Member Function Documentation	104
6.14.1.1 <code>test_dns()</code>	104
6.15 <code>test_cp.test_consensus_parser</code> Class Reference	105
6.15.1 Member Function Documentation	107
6.15.1.1 <code>check_pcvesely_merge()</code>	107
6.15.1.2 <code>check_pcvesely_or1()</code>	108
6.15.1.3 <code>check_pcvesely_or2()</code>	109
6.15.1.4 <code>check_pcvesely_ors()</code>	109
6.15.1.5 <code>check_preprocessed_consensus_descriptorstubs()</code>	110
6.15.1.6 <code>fix_pcvesely_ors_for_merge()</code>	111
6.15.1.7 <code>test_download_consensus_if_missing()</code>	111
6.15.1.8 <code>test_find_preprocessed_ip_address()</code>	111
6.15.1.9 <code>test_find_preprocessed_ip_address_time_filter_dateprefix()</code>	112
6.15.1.10 <code>test_find_preprocessed_ip_address_time_filter_None()</code>	112
6.15.1.11 <code>test_find_preprocessed_ip_address_time_filter_time()</code>	112
6.15.1.12 <code>test_fixup()</code>	112
6.15.1.13 <code>test_get_consensus_fullpath()</code>	113
6.15.1.14 <code>test_get_ip_address_activity()</code>	113
6.15.1.15 <code>test_get_last_preprocessed_consensus()</code>	113
6.15.1.16 <code>test_main_create()</code>	113
6.15.1.17 <code>test_main_findip_address()</code>	113
6.15.1.18 <code>test_main_update()</code>	114
6.15.1.19 <code>test_merge_subsequent_ors()</code>	114

6.15.1.20	test_merge_subsequent_ors_dns1()	114
6.15.1.21	test_merge_subsequent_ors_dns2()	115
6.15.1.22	test_merge_subsequent_ors_dns_both()	115
6.15.1.23	test_merge_subsequent_ors_failed()	115
6.15.1.24	test_parse_ors_from_file_2or()	116
6.16	test_files.test_files Class Reference	116
6.16.1	Member Function Documentation	117
6.16.1.1	scenario_abspath()	118
6.16.1.2	scenario_abspath_long()	118
6.16.1.3	scenario_addr()	119
6.16.1.4	scenario_create_ip_filename()	120
6.16.1.5	scenario_relpath()	120
6.16.1.6	scenario_relpath_long()	121
6.16.1.7	test_create_ip_filename_ipv4()	122
6.16.1.8	test_create_ip_filename_ipv6()	122
6.16.1.9	test_get_ranges_addresses_ipv4()	122
6.16.1.10	test_get_ranges_addresses_ipv6()	123
6.17	test_serializable.test_files Class Reference	124
6.17.1	Member Function Documentation	125
6.17.1.1	test_base_init()	125
6.17.1.2	test_htmlize_lines_links()	126
6.17.1.3	test_htmlize_ors()	126
6.17.1.4	test_jsonify_links()	126
6.17.1.5	test_jsonify_links_with_extend()	126
6.17.1.6	test_jsonify_ors()	126
6.18	test_geolite2.test_geolite2 Class Reference	127
6.18.1	Member Function Documentation	128
6.18.1.1	setUp()	129
6.18.1.2	test_empty()	129
6.18.1.3	test_geolite2_asn_ipv6()	129
6.18.1.4	test_geolite2_missing_data()	129
6.18.1.5	test_get_data_asn_firstrow_firstdb()	129
6.18.1.6	test_get_data_city_firstrow_firstdb()	129
6.18.1.7	test_get_data_city_noneu_database()	129
6.18.1.8	test_get_directories()	130
6.18.2	Member Data Documentation	130
6.18.2.1	db	130
6.18.2.2	maxDiff	130
6.19	test_or.test_or Class Reference	131
6.19.1	Member Function Documentation	133
6.19.1.1	setUp()	133
6.19.1.2	test_get_allow_ports()	133

6.19.1.3 test_get_bandwidth()	133
6.19.1.4 test_get_digest()	134
6.19.1.5 test_get_dirport()	134
6.19.1.6 test_get_flags()	134
6.19.1.7 test_get_identity()	134
6.19.1.8 test_get_inconsensus_fresh_until()	134
6.19.1.9 test_get_inconsensus_val_after()	134
6.19.1.10 test_get_inconsensus_val_until()	134
6.19.1.11 test_get_ip()	135
6.19.1.12 test_get_ipv6()	135
6.19.1.13 test_get_json_dict()	135
6.19.1.14 test_get_nickname()	135
6.19.1.15 test_get_orport()	135
6.19.1.16 test_get_publication()	135
6.19.1.17 test_get_supported_proto()	135
6.19.1.18 test_get_version()	136
6.19.1.19 test_output_json()	136
6.19.2 Member Data Documentation	136
6.19.2.1 __or	136
6.19.2.2 maxDiff	136
6.20 test_rest.test_rest Class Reference	137
6.20.1 Member Function Documentation	139
6.20.1.1 create_app()	139
6.20.1.2 make_request()	139
6.20.1.3 make_request_html()	140
6.20.1.4 make_request_json()	141
6.20.1.5 setUp()	142
6.20.1.6 test_htmlize_doctype()	143
6.20.1.7 test_htmlize_list_aka_links()	143
6.20.1.8 test_htmlize_title()	143
6.20.1.9 test_request_wants_json()	144
6.20.1.10 test_rest_redirect()	144
6.20.1.11 test_route_addresses()	145
6.20.1.12 test_route_addresses_date()	145
6.20.1.13 test_route_addresses_month()	145
6.20.1.14 test_route_addresses_specific()	146
6.20.1.15 test_route_addresses_specific_date()	146
6.20.1.16 test_route_addresses_specific_month()	146
6.20.1.17 test_route_addresses_specific_time()	147
6.20.1.18 test_route_addresses_specific_year()	147
6.20.1.19 test_route_addresses_year()	148
6.20.1.20 test_route_entry_point_html()	148

6.20.1.21 test_route_entry_point_json()	148
6.20.1.22 test_show_ip_address_activity_times_redirect()	149
6.20.2 Member Data Documentation	149
6.20.2.1 app	149
6.21 test_time.test_time_wrapper Class Reference	150
6.21.1 Member Function Documentation	151
6.21.1.1 test_format()	152
6.21.1.2 test_FormatTimeWrapper()	152
6.21.1.3 test_FormatTimeWrapper_get()	152
6.21.1.4 test_next_day()	152
6.21.1.5 test_next_month()	152
6.21.1.6 test_prev_day()	152
6.21.1.7 test_prev_month()	153
6.21.1.8 test_TimeWrapper()	153
6.21.1.9 test_TimeWrapper_get()	153
6.21.1.10 test_TorTimeWrapper()	153
6.21.1.11 test_TorTimeWrapper_get()	153
6.21.1.12 test_zero()	153
6.22 time_parser.TimeWrapper Class Reference	154
6.22.1 Detailed Description	155
6.22.2 Constructor & Destructor Documentation	155
6.22.2.1 __init__()	155
6.22.3 Member Data Documentation	156
6.22.3.1 v	156
6.23 time_parser.TimeWrapperBase Class Reference	156
6.23.1 Detailed Description	157
6.23.2 Member Function Documentation	157
6.23.2.1 __get_days_in_month()	157
6.23.2.2 __str__()	158
6.23.2.3 format()	158
6.23.2.4 get()	158
6.23.2.5 next_day()	158
6.23.2.6 next_month()	158
6.23.2.7 prev_day()	159
6.23.2.8 prev_month()	159
6.24 time_parser.TorTimeWrapper Class Reference	159
6.24.1 Detailed Description	160
6.24.2 Constructor & Destructor Documentation	160
6.24.2.1 __init__()	161
6.24.3 Member Data Documentation	161
6.24.3.1 v	161
6.25 type Class Reference	161

7 File Documentation	163
7.1 consensus_parser.py File Reference	163
7.2 dns.py File Reference	164
7.3 geolite2.py File Reference	164
7.4 ipfiles.py File Reference	164
7.5 onion_router.py File Reference	165
7.6 parametrizable_tc.py File Reference	165
7.7 preprocess_geolite.py File Reference	165
7.8 profilerest.py File Reference	166
7.9 rest.py File Reference	166
7.10 serializable.py File Reference	167
7.11 snapshot.py File Reference	167
7.12 test.py File Reference	167
7.13 test_additional_info.py File Reference	168
7.14 test_cp.py File Reference	168
7.15 test_files.py File Reference	169
7.16 test_geolite2.py File Reference	169
7.17 test_or.py File Reference	170
7.18 test_rest.py File Reference	170
7.19 test_serializable.py File Reference	171
7.20 test_time.py File Reference	171
7.21 time_parser.py File Reference	172
Index	173

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

consensus_parser	9
dns	20
geolite2	21
ipfiles	22
onion_router	25
parametrizable_tc	25
preprocess_geolite	25
profilerest	31
rest	31
serializable	38
snapshot	38
test	38
test_additional_info	41
test_cp	42
test_files	44
test_geolite2	45
test_or	47
test_rest	48
test_serializable	50
test_time	52
time_parser	53

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

geolite2.geolite2_accessor.db_info	55
test_rest.fake	56
test_cp.fake_cp_args	57
geolite2.geolite2_accessor	62
metaclass	68
test_additional_info.test_additional_info	103
test_cp.test_consensus_parser	105
test_files.test_files	116
test_geolite2.test_geolite2	127
test_rest.test_rest	137
test_serializable.test_files	124
test_time.test_time_wrapper	150
onion_router.onion_router	69
parametrizable_tc_meta	
test_additional_info.test_additional_info	103
test_cp.test_consensus_parser	105
test_files.test_files	116
test_geolite2.test_geolite2	127
test_rest.test_rest	137
test_serializable.test_files	124
test_time.test_time_wrapper	150
serializable.serializable_list	90
serializable.serializable_links	87
serializable.serializable_ors	94
snapshot.Snapshot	98
super_class	102
test_rest.test_rest	137
TestCase	
test_additional_info.test_additional_info	103
test_cp.test_consensus_parser	105
test_files.test_files	116
test_geolite2.test_geolite2	127
test_or.test_or	131
test_serializable.test_files	124

test_time.test_time_wrapper	150
time_parser.TimeWrapperBase	156
time_parser.FormatTimeWrapper	60
time_parser.TimeWrapper	154
time_parser.TorTimeWrapper	159
type	161
parametrizable_tc.parametrizable_tc_meta	84

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

geolite2.geolite2_accessor.db_info	55
test_rest.fake	56
test_cp.fake_cp_args	57
time_parser.FormatTimeWrapper	60
geolite2.geolite2_accessor	62
metaclass	68
onion_router.onion_router	69
parametrizable_tc.parametrizable_tc_meta	84
serializable.serializable_links	87
serializable.serializable_list	90
serializable.serializable_ors	94
snapshot.Snapshot	98
super_class	102
test_additional_info.test_additional_info	103
test_cp.test_consensus_parser	105
test_files.test_files	116
test_serializable.test_files	124
test_geolite2.test_geolite2	127
test_or.test_or	131
test_rest.test_rest	137
test_time.test_time_wrapper	150
time_parser.TimeWrapper	154
time_parser.TimeWrapperBase	156
time_parser.TorTimeWrapper	159
type	161

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

consensus_parser.py	163
dns.py	164
geolite2.py	164
ipfiles.py	164
onion_router.py	165
parametrizable_tc.py	165
preprocess_geolite.py	165
profilerest.py	166
rest.py	166
serializable.py	167
snapshot.py	167
test.py	167
test_additional_info.py	168
test_cp.py	168
test_files.py	169
test_geolite2.py	169
test_or.py	170
test_rest.py	170
test_serializable.py	171
test_time.py	171
time_parser.py	172

Chapter 5

Namespace Documentation

5.1 consensus_parser Namespace Reference

Functions

- def [create_snapshot_from_consensus_file](#) (fname)
- def [write_preprocessed_from_consensus_file](#) (fname, preprocess_abspath, last)
- def [preprocess_directory_consensus](#) (abspath, preprocess_abspath, last)
- def [preprocess_path_consensus](#) (abspath, preprocess_abspath, last)
- def [get_consensus_fullpath](#) (consensus_abspath, t)
- def [download_consensus_if_missing](#) (consensus_abspath, t)
- def [fixup_missing_consensus](#) (consensus_abspath, preprocess_abspath, t)
- def [get_last_preprocessed_consensus](#) (preprocess_abspath)
- def [process_path_consensus](#) (consensus_abspath, preprocess_abspath)
- def [update_path_consensus](#) (consensus_abspath, preprocess_abspath)
- def [write_preprocessed](#) (preprocess_abspath, consensus_snapshot)
- def [parse_ors_from_file](#) (fname)
- def [add_geolite_data_to_onion_routers](#) (ipaddr, ors, geolite_acc)
- def [find_preprocessed_ip_address](#) (ipaddr, preprocessed_dir, geolite_acc=None)
- def [find_preprocessed_ip_address_time_filter](#) (ipaddr, preprocessed_dir, time_search, date_prefix, geolite_acc)
- def [output_preprocessed_ip_addresses_time_filter](#) (ipaddr, preprocessed_dir, time_search, date_prefix, geolite_acc, outfile)
- def [merge_subsequent_ors](#) (ors)
- def [get_ip_address_activity](#) (ipaddr, preprocessed_dir, time_format="%Y-%m-%d")
- def [process_args](#) ()
- def [main](#) (args, stdout)

Variables

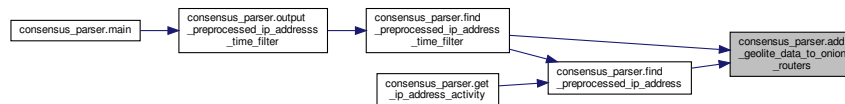
- int [WAIT_FOR_CONSENSUS](#) = 600
- def [args](#) = [process_args](#)()

5.1.1 Function Documentation

5.1.1.1 add_geolite_data_to_onion_routers()

```
def consensus_parser.add_geolite_data_to_onion_routers (
    ipaddr,
    ors,
    geolite_acc )
```

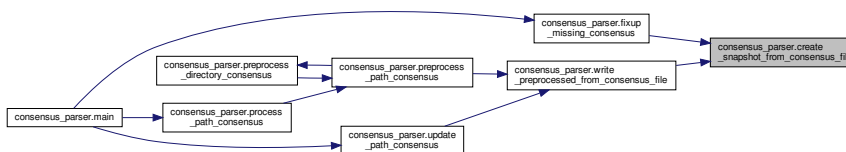
Here is the caller graph for this function:



5.1.1.2 create_snapshot_from_consensus_file()

```
def consensus_parser.create_snapshot_from_consensus_file (
    fname )
```

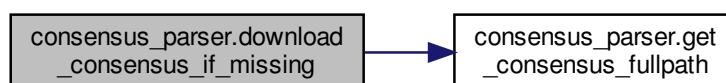
Here is the caller graph for this function:



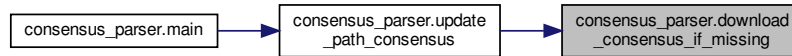
5.1.1.3 download_consensus_if_missing()

```
def consensus_parser.download_consensus_if_missing (
    consensus_abspath,
    t )
```

Here is the call graph for this function:



Here is the caller graph for this function:

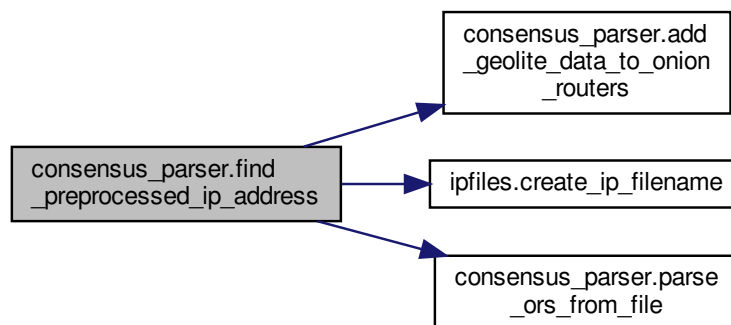


5.1.1.4 find_preprocessed_ip_address()

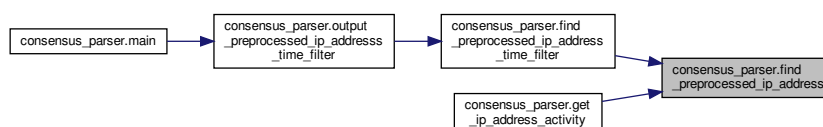
```

def consensus_parser.find_preprocessed_ip_address (
    ipaddr,
    preprocessed_dir,
    geolite_acc = None )
  
```

Here is the call graph for this function:



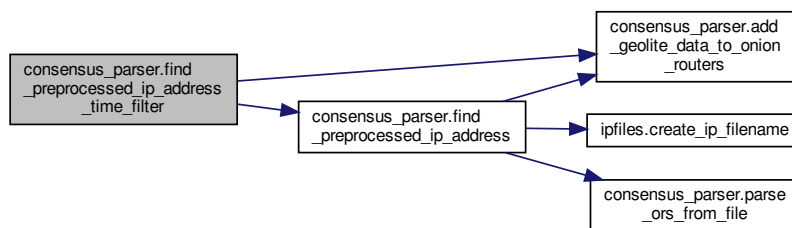
Here is the caller graph for this function:



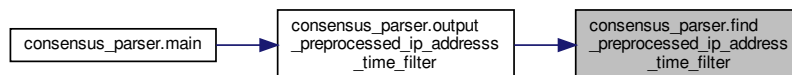
5.1.1.5 find_preprocessed_ip_address_time_filter()

```
def consensus_parser.find_preprocessed_ip_address_time_filter (
    ipaddr,
    preprocessed_dir,
    time_search,
    date_prefix,
    geolite_acc )
```

Here is the call graph for this function:



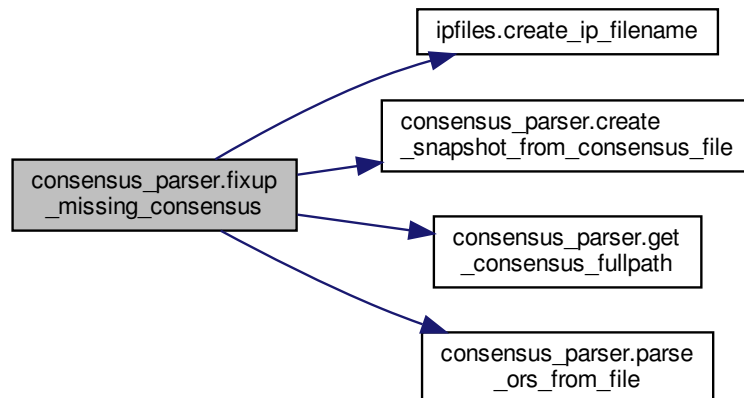
Here is the caller graph for this function:



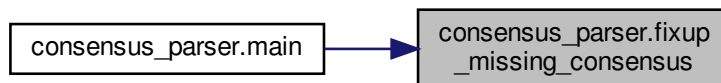
5.1.1.6 fixup_missing_consensus()

```
def consensus_parser.fixup_missing_consensus (
    consensus_abspath,
    preprocess_abspath,
    t )
```

Here is the call graph for this function:



Here is the caller graph for this function:

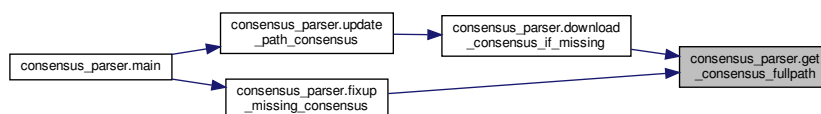


5.1.1.7 get_consensus_fullpath()

```

def consensus_parser.get_consensus_fullpath (
    consensus_abspath,
    t )
  
```

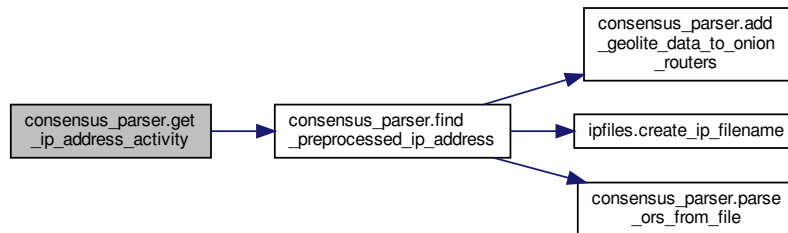
Here is the caller graph for this function:



5.1.1.8 get_ip_address_activity()

```
def consensus_parser.get_ip_address_activity (
    ipaddr,
    preprocessed_dir,
    time_format = "%Y-%m-%d" )
```

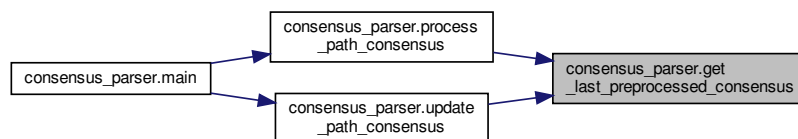
Here is the call graph for this function:



5.1.1.9 get_last_preprocessed_consensus()

```
def consensus_parser.get_last_preprocessed_consensus (
    preprocess_abspath )
```

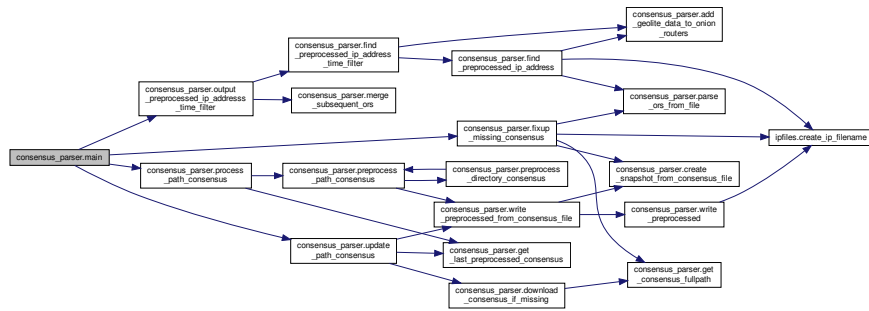
Here is the caller graph for this function:



5.1.1.10 main()

```
def consensus_parser.main (
    args,
    stdout )
```


Here is the call graph for this function:



5.1.1.11 merge_subsequent_ors()

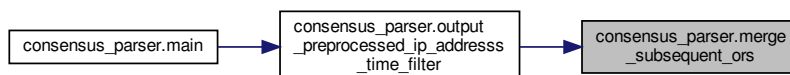
```
def consensus_parser.merge_subsequent_ors (
    ors )
```

Merges subsequent OR entries in given iterable.

Returns a new iterable with possibly merged ors.

Note that it merges subsequent entries only if they match in all properties including IP addresses. Only neighbour entries are examined.

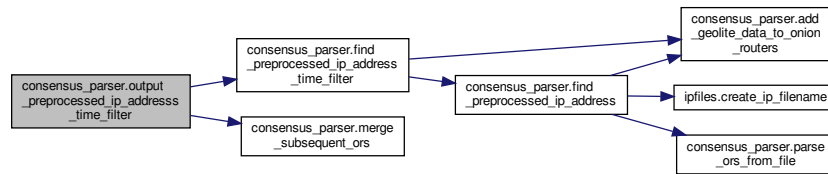
Here is the caller graph for this function:



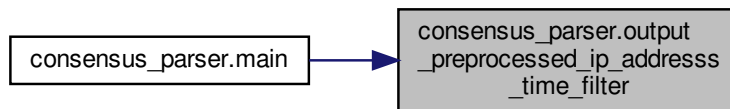
5.1.1.12 output_preprocessed_ip_addresses_time_filter()

```
def consensus_parser.output_preprocessed_ip_addresses_time_filter (
    ipaddr,
    preprocessed_dir,
    time_search,
    date_prefix,
    geolite_acc,
    outfile )
```

Here is the call graph for this function:



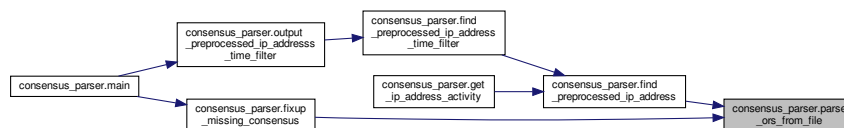
Here is the caller graph for this function:



5.1.1.13 parse_ors_from_file()

```
def consensus_parser.parse_ors_from_file (
    fname )
```

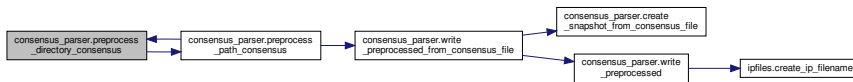
Here is the caller graph for this function:



5.1.1.14 preprocess_directory_consensus()

```
def consensus_parser.preprocess_directory_consensus (
    abspath,
    preprocess_abspath,
    last )
```

Here is the call graph for this function:



Here is the caller graph for this function:

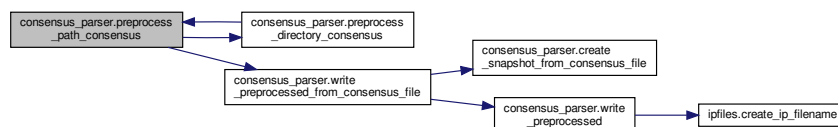


5.1.1.15 preprocess_path_consensus()

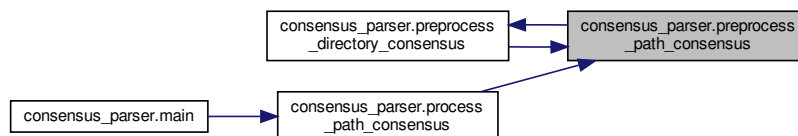
```

def consensus_parser.preprocess_path_consensus (
    abspath,
    preprocess_abspath,
    last )
  
```

Here is the call graph for this function:



Here is the caller graph for this function:



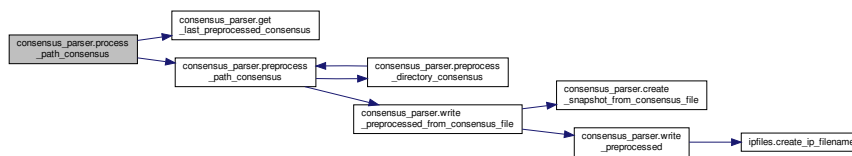
5.1.1.16 process_args()

```
def consensus_parser.process_args ( )
```

5.1.1.17 process_path_consensus()

```
def consensus_parser.process_path_consensus (
    consensus_abspath,
    preprocess_abspath )
```

Here is the call graph for this function:



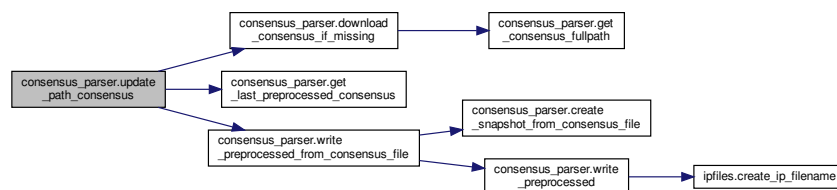
Here is the caller graph for this function:



5.1.1.18 update_path_consensus()

```
def consensus_parser.update_path_consensus (
    consensus_abspath,
    preprocess_abspath )
```

Here is the call graph for this function:



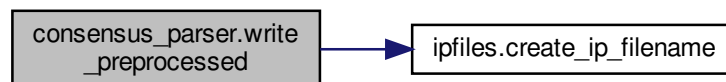
Here is the caller graph for this function:



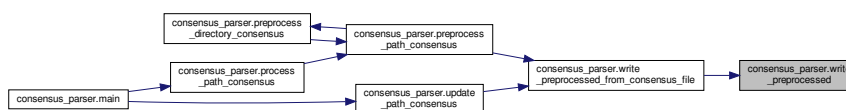
5.1.1.19 write_preprocessed()

```
def consensus_parser.write_preprocessed (
    preprocess_abspath,
    consensus_snapshot )
```

Here is the call graph for this function:



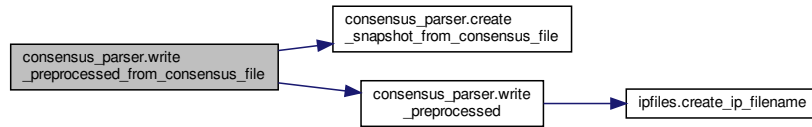
Here is the caller graph for this function:



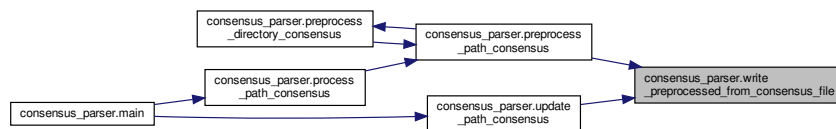
5.1.1.20 write_preprocessed_from_consensus_file()

```
def consensus_parser.write_preprocessed_from_consensus_file (
    fname,
    preprocess_abspath,
    last )
```

Here is the call graph for this function:



Here is the caller graph for this function:



5.1.2 Variable Documentation

5.1.2.1 args

```
def consensus_parser.args = process_args()
```

5.1.2.2 WAIT_FOR_CONSENSUS

```
int consensus_parser.WAIT_FOR_CONSENSUS = 600
```

5.2 dns Namespace Reference

Functions

- def [ip_addr_to_domain](#) (ip_addr)

5.2.1 Function Documentation

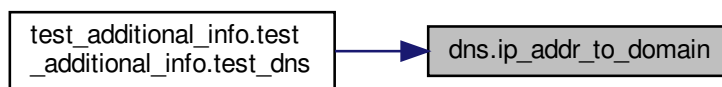
5.2.1.1 ip_addr_to_domain()

```
def dns.ip_addr_to_domain (
    ip_addr )
```

Converts IPv4/IPv6 address to its primary PTR name.

Note that in case of failed resolution, it returns None.

Here is the caller graph for this function:



5.3 geolite2 Namespace Reference

Classes

- class [geolite2_accessor](#)

Functions

- def [process_args](#) ()

Variables

- def `args` = [process_args](#)()
- `gl` = [geolite2_accessor](#)(args.geolite2dir)
- `t` = [TimeWrapper](#)(args.time).get()

5.3.1 Function Documentation

5.3.1.1 process_args()

```
def geolite2.process_args ( )
```

5.3.2 Variable Documentation

5.3.2.1 args

```
def geolite2.args = process_args()
```

5.3.2.2 gl

```
geolite2.gl = geolite2_accessor(args.geolite2dir)
```

5.3.2.3 t

```
geolite2.t = TimeWrapper(args.time).get()
```

5.4 ipfiles Namespace Reference

Functions

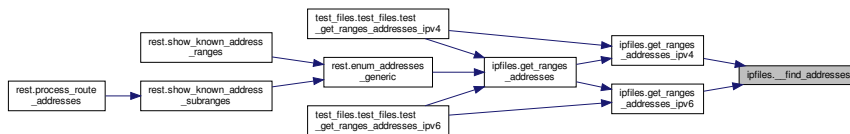
- def [create_ip_filename](#) (path, ipaddr, create_path=False)
- def [__find_addresses](#) (input_net, res_plen, p, prefix, remainder)
- def [get_ranges_addresses_ipv4](#) (path, network_addr, plen)
- def [get_ranges_addresses_ipv6](#) (path, network_addr, plen)
- def [get_ranges_addresses](#) (path, network_addr, plen)

5.4.1 Function Documentation

5.4.1.1 __find_addresses()

```
def ipfiles.__find_addresses (
    input_net,
    res_plen,
    p,
    prefix,
    remainder ) [private]
```

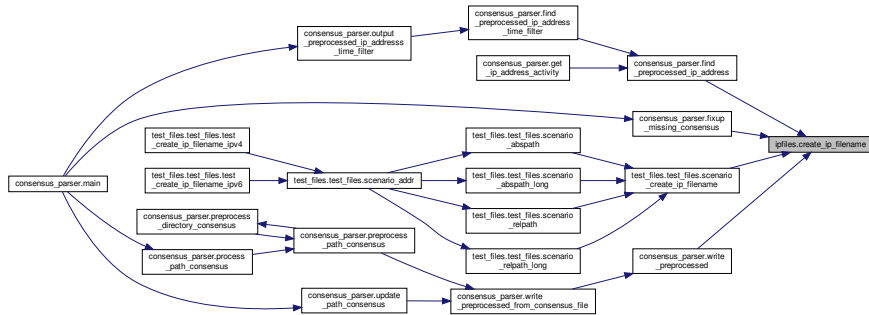
Here is the caller graph for this function:



5.4.1.2 create_ip_filename()

```
def ipfiles.create_ip_filename (
    path,
    ipaddr,
    create_path = False )
```

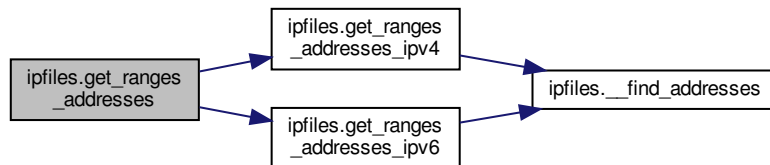
Here is the caller graph for this function:



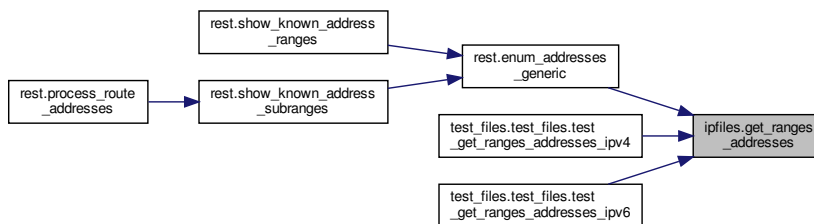
5.4.1.3 get_ranges_addresses()

```
def ipfiles.get_ranges_addresses (
    path,
    network_addr,
    plen )
```

Here is the call graph for this function:



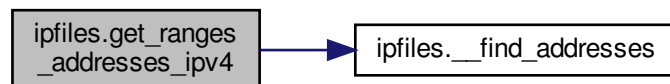
Here is the caller graph for this function:



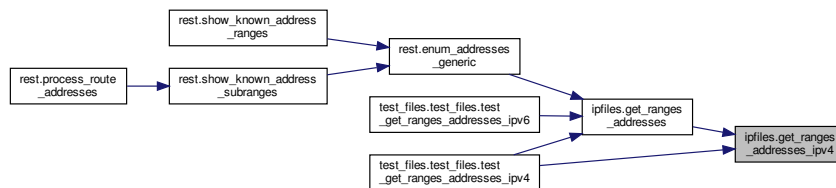
5.4.1.4 get_ranges_addresses_ipv4()

```
def ipfiles.get_ranges_addresses_ipv4 (
    path,
    network_addr,
    plen )
```

Here is the call graph for this function:



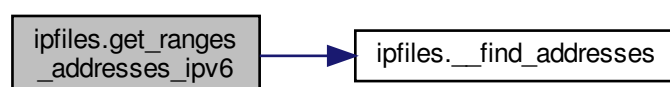
Here is the caller graph for this function:



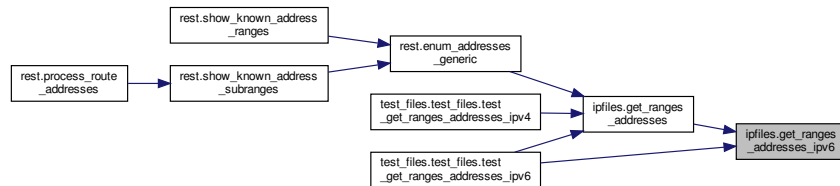
5.4.1.5 get_ranges_addresses_ipv6()

```
def ipfiles.get_ranges_addresses_ipv6 (
    path,
    network_addr,
    plen )
```

Here is the call graph for this function:



Here is the caller graph for this function:



5.5 onion_router Namespace Reference

Classes

- class [onion_router](#)

5.6 parametrizable_tc Namespace Reference

Classes

- class [parametrizable_tc_meta](#)

Functions

- def [parametrizable_test](#) (paramList)

5.6.1 Function Documentation

5.6.1.1 parametrizable_test()

```
def parametrizable_tc.parametrizable_test (
    paramList )
```

5.7 preprocess_geolite Namespace Reference

Functions

- def [write_network](#) (basedir, line, first_line_csv)
- def [write_location](#) (basedir, line)
- def [process_file](#) (fname, outdir, stdout, ipaddr_sep, plen_group_len, first_line_csv)
- def [process_file4](#) (fname, outdir, stdout, first_line_csv)
- def [process_file6](#) (fname, outdir, stdout, first_line_csv)
- def [process_file_loc](#) (fname, outdir, stdout)
- def [process_args](#) ()
- def [main](#) (args, stdout)

Variables

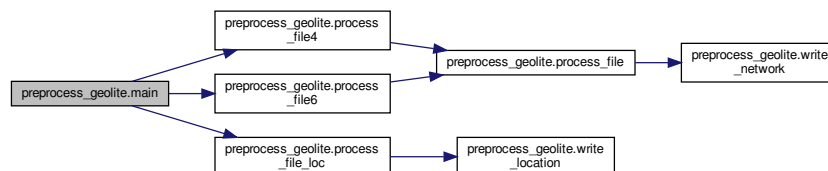
- string `ASN_FIRST_LINE_CSV` = "network,autonomous_system_number,autonomous_system_organization\n"
- string `CITY_FIRST_LINE_CSV` = "network,geoname_id,registered_country_geoname_id,represented_↔
country_geoname_id,is_anonymous_proxy,is_satellite_provider,postal_code,latitude,longitude,accuracy_↔
radius\n"
- string `LOCATION_FIRST_LINE_CSV` = "geoname_id,locale_code,continent_code,continent_name,country_↔
_iso_code,country_name,subdivision_1_iso_code,subdivision_1_name,subdivision_2_iso_code,subdivision_↔
_2_name,city_name,metro_code,time_zone\n"
- def `args` = `process_args()`

5.7.1 Function Documentation

5.7.1.1 main()

```
def preprocess_geolite.main (
    args,
    stdout )
```

Here is the call graph for this function:



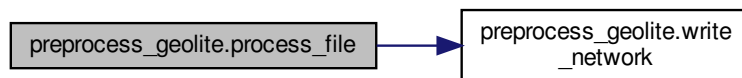
5.7.1.2 process_args()

```
def preprocess_geolite.process_args ( )
```

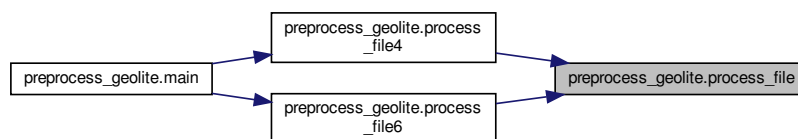
5.7.1.3 process_file()

```
def preprocess_geolite.process_file (
    fname,
    outdir,
    stdout,
    ipaddr_sep,
    plen_group_len,
    first_line_csv )
```

Here is the call graph for this function:



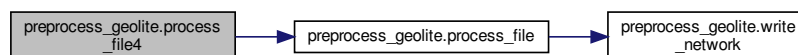
Here is the caller graph for this function:



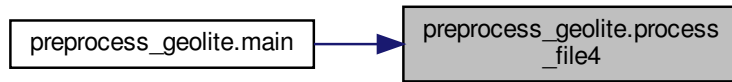
5.7.1.4 process_file4()

```
def preprocess_geolite.process_file4 (
    fname,
    outdir,
    stdout,
    first_line_csv )
```

Here is the call graph for this function:



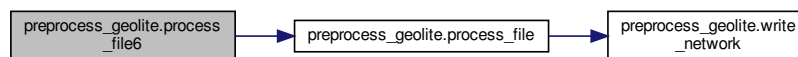
Here is the caller graph for this function:



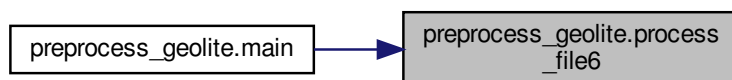
5.7.1.5 process_file6()

```
def preprocess_geolite.process_file6 (
    fname,
    outdir,
    stdout,
    first_line_csv )
```

Here is the call graph for this function:



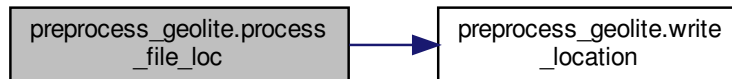
Here is the caller graph for this function:



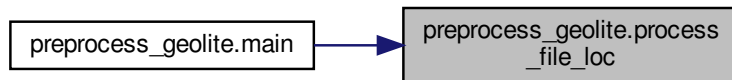
5.7.1.6 process_file_loc()

```
def preprocess_geolite.process_file_loc (  
    fname,  
    outdir,  
    stdout )
```

Here is the call graph for this function:



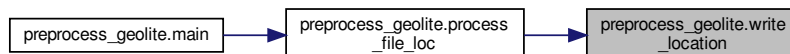
Here is the caller graph for this function:



5.7.1.7 write_location()

```
def preprocess_geolite.write_location (  
    basedir,  
    line )
```

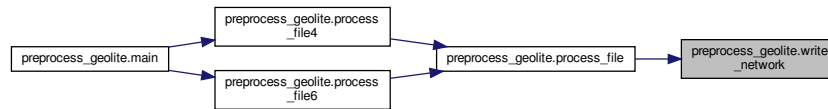
Here is the caller graph for this function:



5.7.1.8 write_network()

```
def preprocess_geolite.write_network (
    basedir,
    line,
    first_line_csv )
```

Here is the caller graph for this function:



5.7.2 Variable Documentation

5.7.2.1 args

```
def preprocess_geolite.args = process_args ()
```

5.7.2.2 ASN_FIRST_LINE_CSV

```
string preprocess_geolite.ASN_FIRST_LINE_CSV = "network,autonomous_system_number,autonomous_↔
system_organization↔"
```

5.7.2.3 CITY_FIRST_LINE_CSV

```
string preprocess_geolite.CITY_FIRST_LINE_CSV = "network,geoname_id,registered_country_↔
geoname_id,represented_country_geoname_id,is_anonymous_proxy,is_satellite_provider,postal_↔
code,latitude,longitude,accuracy_radius↔"
```

5.7.2.4 LOCATION_FIRST_LINE_CSV

```
string preprocess_geolite.LOCATION_FIRST_LINE_CSV = "geoname_id,locale_code,continent_code,continent_↔
_name,country_iso_code,country_name,subdivision_1_iso_code,subdivision_1_name,subdivision_2_↔
iso_code,subdivision_2_name,city_name,metro_code,time_zone↔"
```


5.8 profilerest Namespace Reference

Variables

- [wsgi_app](#)
- [sort_by](#)
- [restrictions](#)
- [debug](#)

5.8.1 Variable Documentation

5.8.1.1 debug

```
profilerest.debug
```

5.8.1.2 restrictions

```
profilerest.restrictions
```

5.8.1.3 sort_by

```
profilerest.sort_by
```

5.8.1.4 wsgi_app

```
profilerest.wsgi_app
```

5.9 rest Namespace Reference

Functions

- def [request_wants_json](#) ()
- def [htmlize](#) (result, related, func)
- def [evaluate_accept_header_convert_json](#) (func)
- def [entry_point](#) ()
- def [show_known_address_ranges](#) ()
- def [process_route_addresses](#) (address_detail)
- def [show_all_ip_address_information](#) (address)
- def [show_ip_address_activity_dates](#) (addr)
- def [show_ip_address_activity_on_specific_date](#) (addr, date)
- def [show_ip_address_activity_times](#) (addr)
- def [show_ip_address_activity_at_specific_time](#) (addr, time)
- def [show_ip_address_active_months](#) (addr)
- def [show_ip_address_activity_during_specific_month](#) (addr, month)
- def [show_ip_address_active_years](#) (addr)
- def [show_ip_address_activity_during_specific_year](#) (addr, year)
- def [enum_addresses_generic](#) (netaddr, plen)
- def [show_known_address_subranges](#) (netaddr, plen)

Variables

- `app` = flask.Flask(__name__)
- string `PREPORCESSED_DIR` = "/mnt/data/preprocessedip/"
- string `GEOLITE_DIR` = "/mnt/data/preprocessed_geolite/"

5.9.1 Function Documentation

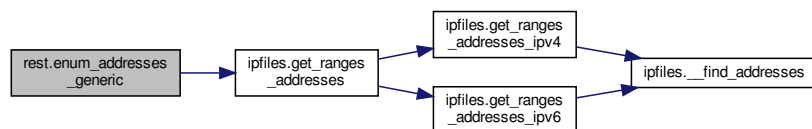
5.9.1.1 entry_point()

```
def rest.entry_point ( )
```

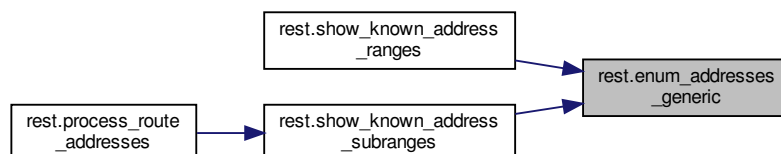
5.9.1.2 enum_addresses_generic()

```
def rest.enum_addresses_generic (
    netaddr,
    plen )
```

Here is the call graph for this function:



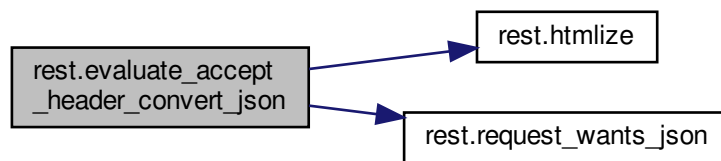
Here is the caller graph for this function:



5.9.1.3 evaluate_accept_header_convert_json()

```
def rest.evaluate_accept_header_convert_json (
    func )
```

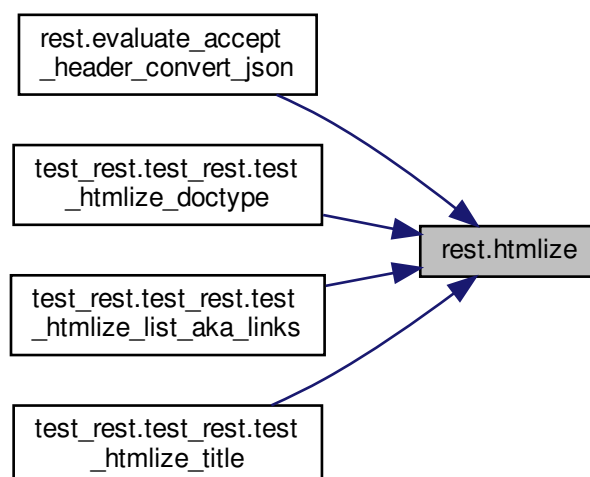
Here is the call graph for this function:



5.9.1.4 htmlize()

```
def rest.htmlize (
    result,
    related,
    func )
```

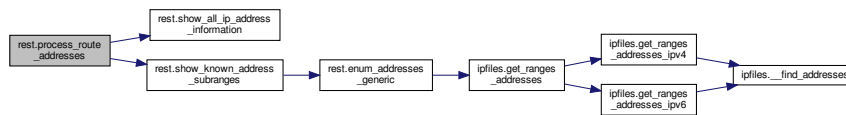
Here is the caller graph for this function:



5.9.1.5 process_route_addresses()

```
def rest.process_route_addresses (
    address_detail )
```

Here is the call graph for this function:



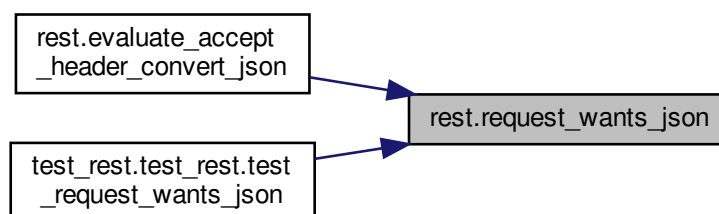
5.9.1.6 request_wants_json()

```
def rest.request_wants_json ( )
```

Check if we should return json

Copied from <http://flask.pocoo.org/snippets/45/>

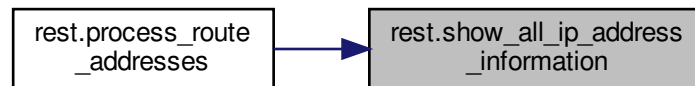
Here is the caller graph for this function:



5.9.1.7 show_all_ip_address_information()

```
def rest.show_all_ip_address_information (
    address )
```

Here is the caller graph for this function:



5.9.1.8 show_ip_address_active_months()

```
def rest.show_ip_address_active_months (
    addr )
```

5.9.1.9 show_ip_address_active_years()

```
def rest.show_ip_address_active_years (
    addr )
```

5.9.1.10 show_ip_address_activity_at_specific_time()

```
def rest.show_ip_address_activity_at_specific_time (
    addr,
    time )
```

5.9.1.11 show_ip_address_activity_dates()

```
def rest.show_ip_address_activity_dates (
    addr )
```

5.9.1.12 show_ip_address_activity_during_specific_month()

```
def rest.show_ip_address_activity_during_specific_month (
    addr,
    month )
```

5.9.1.13 show_ip_address_activity_during_specific_year()

```
def rest.show_ip_address_activity_during_specific_year (
    addr,
    year )
```

5.9.1.14 show_ip_address_activity_on_specific_date()

```
def rest.show_ip_address_activity_on_specific_date (
    addr,
    date )
```

5.9.1.15 show_ip_address_activity_times()

```
def rest.show_ip_address_activity_times (
    addr )
```

5.9.1.16 show_known_address_ranges()

```
def rest.show_known_address_ranges ( )
```

Here is the call graph for this function:



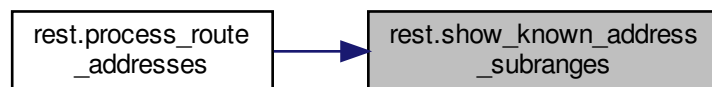
5.9.1.17 show_known_address_subranges()

```
def rest.show_known_address_subranges (
    netaddr,
    plen )
```

Here is the call graph for this function:



Here is the caller graph for this function:



5.9.2 Variable Documentation

5.9.2.1 app

```
rest.app = flask.Flask(__name__)
```

5.9.2.2 GEOLITE_DIR

```
string rest.GEOLITE_DIR = "/mnt/data/preprocessed_geolite/"
```

5.9.2.3 PREPORCESSED_DIR

```
string rest.PREPORCESSED_DIR = "/mnt/data/preprocessedip/"
```

5.10 serializable Namespace Reference

Classes

- class [serializable_links](#)
- class [serializable_list](#)
- class [serializable_ors](#)

5.11 snapshot Namespace Reference

Classes

- class [Snapshot](#)

5.12 test Namespace Reference

Functions

- def [suite](#) ()
- def [test](#) ([verbosity](#)=2, failfast=False)

Variables

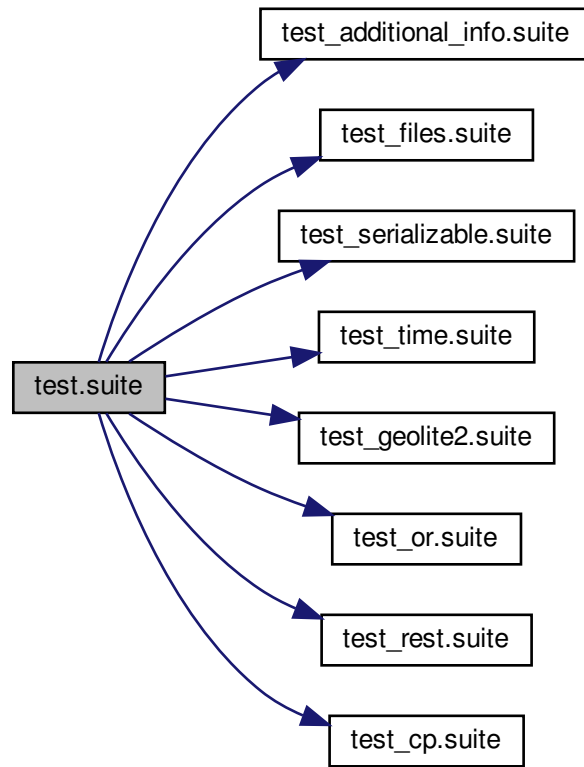
- [verbosity](#)

5.12.1 Function Documentation

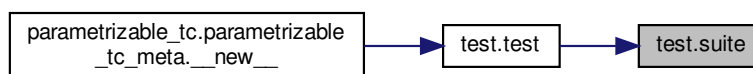
5.12.1.1 suite()

```
def test.suite ( )
```


Here is the call graph for this function:



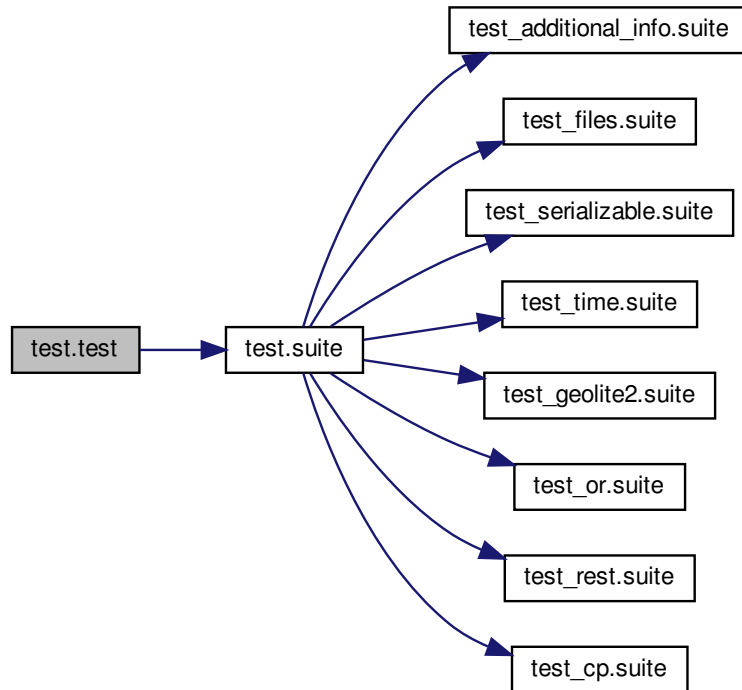
Here is the caller graph for this function:



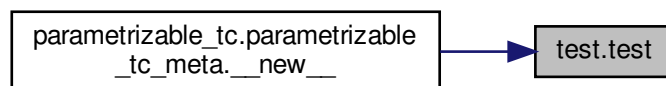
5.12.1.2 test()

```
def test.test (
    verbosity = 2,
    failfast = False )
```

Here is the call graph for this function:



Here is the caller graph for this function:



5.12.2 Variable Documentation

5.12.2.1 verbosity

`test.verbosity`

5.13 test_additional_info Namespace Reference

Classes

- class [test_additional_info](#)

Functions

- def [suite](#) ()
- def [test](#) (verbosity=2, failfast=False)

Variables

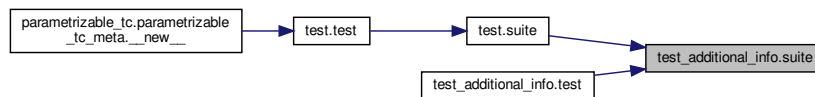
- [verbosity](#)

5.13.1 Function Documentation

5.13.1.1 suite()

```
def test_additional_info.suite ( )
```

Here is the caller graph for this function:



5.13.1.2 test()

```
def test_additional_info.test (
    verbosity = 2,
    failfast = False )
```

Here is the call graph for this function:



5.13.2 Variable Documentation

5.13.2.1 verbosity

```
test_additional_info.verbosity
```

5.14 test_cp Namespace Reference

Classes

- class [fake_cp_args](#)
- class [test_consensus_parser](#)

Functions

- def [suite](#) ()
- def [test](#) (verbosity=2, failfast=False)

Variables

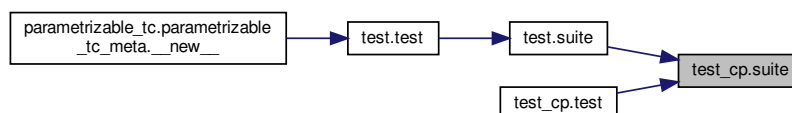
- string [PCVESELY_JSON](#)
- string [PREPROCESSED_DIR_CREATE](#) = "testfiles/preprocessedtestauto"
- string [CONSENSUS_PATH](#) = "testfiles/descriptorstubs"
- [verbosity](#)

5.14.1 Function Documentation

5.14.1.1 suite()

```
def test_cp.suite ( )
```

Here is the caller graph for this function:



5.14.1.2 test()

```
def test_cp.test (
    verbosity = 2,
    failfast = False )
```

Here is the call graph for this function:



5.14.2 Variable Documentation

5.14.2.1 CONSENSUS_PATH

```
string test_cp.CONSENSUS_PATH = "testfiles/descriptorstubs"
```

5.14.2.2 PCVESELY_JSON

```
string test_cp.PCVESELY_JSON
```

5.14.2.3 PREPROCESSED_DIR_CREATE

```
string test_cp.PREPROCESSED_DIR_CREATE = "testfiles/preprocessedtestauto"
```

5.14.2.4 verbosity

```
test_cp.verbosity
```

5.15 test_files Namespace Reference

Classes

- class [test_files](#)

Functions

- def [suite](#) ()
- def [test](#) (verbosity=2, failfast=False)

Variables

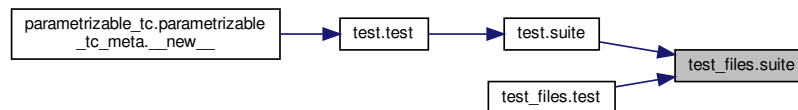
- [verbosity](#)

5.15.1 Function Documentation

5.15.1.1 suite()

```
def test_files.suite ( )
```

Here is the caller graph for this function:



5.15.1.2 test()

```
def test_files.test (
    verbosity = 2,
    failfast = False )
```

Here is the call graph for this function:



5.15.2 Variable Documentation

5.15.2.1 verbosity

```
test_files.verbosity
```

5.16 test_geolite2 Namespace Reference

Classes

- class [test_geolite2](#)

Functions

- def [suite](#) ()
- def [test](#) (verbosity=2, failfast=False)

Variables

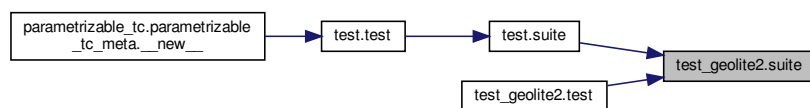
- [t20180605](#) = [FormatTimeWrapper](#)("20180605", "%Y%m%d").get()
- [t20180619](#) = [FormatTimeWrapper](#)("20180619", "%Y%m%d").get()
- [t20180626](#) = [FormatTimeWrapper](#)("20180626", "%Y%m%d").get()
- [t20180703](#) = [FormatTimeWrapper](#)("20180703", "%Y%m%d").get()
- [t20180709](#) = [FormatTimeWrapper](#)("20180709", "%Y%m%d").get()
- [verbosity](#)

5.16.1 Function Documentation

5.16.1.1 suite()

```
def test_geolite2.suite ( )
```

Here is the caller graph for this function:



5.16.1.2 test()

```
def test_geolite2.test (
    verbosity = 2,
    failfast = False )
```

Here is the call graph for this function:



5.16.2 Variable Documentation

5.16.2.1 t20180605

```
test_geolite2.t20180605 = FormatTimeWrapper("20180605", "%Y%m%d").get()
```

5.16.2.2 t20180619

```
test_geolite2.t20180619 = FormatTimeWrapper("20180619", "%Y%m%d").get()
```

5.16.2.3 t20180626

```
test_geolite2.t20180626 = FormatTimeWrapper("20180626", "%Y%m%d").get()
```

5.16.2.4 t20180703

```
test_geolite2.t20180703 = FormatTimeWrapper("20180703", "%Y%m%d").get()
```


5.16.2.5 t20180709

```
test_geolite2.t20180709 = FormatTimeWrapper("20180709", "%Y%m%d").get()
```

5.16.2.6 verbosity

```
test_geolite2.verbosity
```

5.17 test_or Namespace Reference

Classes

- class [test_or](#)

Functions

- def [suite](#) ()
- def [test](#) (verbosity=2, failfast=False)

Variables

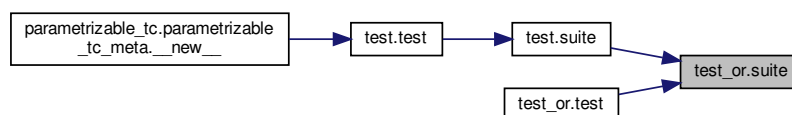
- [verbosity](#)

5.17.1 Function Documentation

5.17.1.1 suite()

```
def test_or.suite ( )
```

Here is the caller graph for this function:



5.17.1.2 test()

```
def test_or.test (
    verbosity = 2,
    failfast = False )
```

Here is the call graph for this function:



5.17.2 Variable Documentation

5.17.2.1 verbosity

```
test_or.verbosity
```

5.18 test_rest Namespace Reference

Classes

- class [fake](#)
- class [test_rest](#)

Functions

- def [suite](#) ()
- def [test](#) (verbosity=2, failfast=False)

Variables

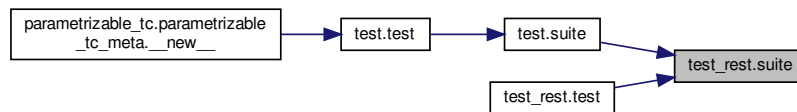
- bool [flask_failed](#) = False
- [super_class](#) = flask_testing.TestCase
- [rest](#)
- string [bs_required_text](#) = "BeautifulSoup not installed on this computer"
- bool [bs_failed](#) = False
- [verbosity](#)

5.18.1 Function Documentation

5.18.1.1 suite()

```
def test_rest.suite ( )
```

Here is the caller graph for this function:



5.18.1.2 test()

```
def test_rest.test (
    verbosity = 2,
    failfast = False )
```

Here is the call graph for this function:



5.18.2 Variable Documentation

5.18.2.1 bs_failed

```
bool test_rest.bs_failed = False
```

5.18.2.2 `bs_required_text`

```
string test_rest.bs_required_text = "BeautifulSoup not installed on this computer"
```

5.18.2.3 `flask_failed`

```
bool test_rest.flask_failed = False
```

5.18.2.4 `rest`

```
test_rest.rest
```

5.18.2.5 `super_class`

```
test_rest.super_class = flask_testing.TestCase
```

5.18.2.6 `verbosity`

```
test_rest.verbosity
```

5.19 `test_serializable` Namespace Reference

Classes

- class [test_files](#)

Functions

- def [suite](#) ()
- def [test](#) (verbosity=2, failfast=False)

Variables

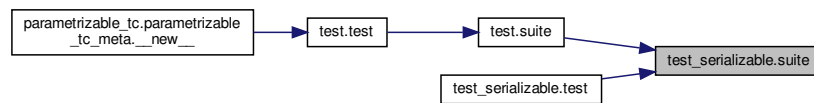
- string [bs_required_text](#) = "BeautifulSoup not installed on this computer"
- bool [bs_failed](#) = False
- [verbosity](#)

5.19.1 Function Documentation

5.19.1.1 suite()

```
def test_serializable.suite ( )
```

Here is the caller graph for this function:



5.19.1.2 test()

```
def test_serializable.test (
    verbosity = 2,
    failfast = False )
```

Here is the call graph for this function:



5.19.2 Variable Documentation

5.19.2.1 bs_failed

```
bool test_serializable.bs_failed = False
```

5.19.2.2 bs_required_text

```
string test_serializable.bs_required_text = "BeautifulSoup not installed on this computer"
```

5.19.2.3 verbosity

```
test_serializable.verbosity
```

5.20 test_time Namespace Reference

Classes

- class [test_time_wrapper](#)

Functions

- def [suite](#) ()
- def [test](#) (verbosity=2, failfast=False)

Variables

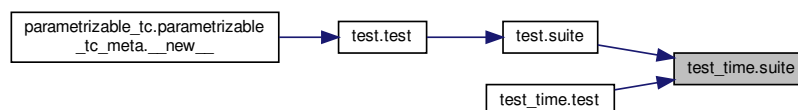
- [verbosity](#)

5.20.1 Function Documentation

5.20.1.1 suite()

```
def test_time.suite ( )
```

Here is the caller graph for this function:



5.20.1.2 test()

```
def test_time.test (
    verbosity = 2,
    failfast = False )
```

Here is the call graph for this function:



5.20.2 Variable Documentation

5.20.2.1 verbosity

```
test_time.verbosity
```

5.21 time_parser Namespace Reference

Classes

- class [FormatTimeWrapper](#)
- class [TimeWrapper](#)
- class [TimeWrapperBase](#)
- class [TorTimeWrapper](#)

Variables

- dictionary [months](#)
- int [DAY_HAS_SECONDS](#) = 24*60*60

5.21.1 Variable Documentation

5.21.1.1 DAY_HAS_SECONDS

```
int time_parser.DAY_HAS_SECONDS = 24*60*60
```

5.21.1.2 months

```
dictionary time_parser.months
```

Initial value:

```
1 = {  
2     "Jan": 1,  
3     "Feb": 2,  
4     "Mar": 3,  
5     "Apr": 4,  
6     "May": 5,  
7     "Jun": 6,  
8     "Jul": 7,  
9     "Aug": 8,  
10    "Sep": 9,  
11    "Oct": 10,  
12    "Nov": 11,  
13    "Dec": 12,  
14 }
```


Chapter 6

Class Documentation

6.1 geolite2.geolite2_accessor.db_info Class Reference

Collaboration diagram for geolite2.geolite2_accessor.db_info:

geolite2.geolite2_accessor.db_info
+ last + dirs + cache
+ __init__()

Public Member Functions

- def [__init__](#)(self)

Public Attributes

- [last](#)
- [dirs](#)
- [cache](#)

6.1.1 Constructor & Destructor Documentation

6.1.1.1 `__init__()`

```
def geolite2.geolite2_accessor.db_info.__init__ (
    self )
```

6.1.2 Member Data Documentation

6.1.2.1 `cache`

```
geolite2.geolite2_accessor.db_info.cache
```

6.1.2.2 `dirs`

```
geolite2.geolite2_accessor.db_info.dirs
```

6.1.2.3 `last`

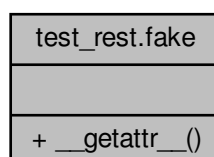
```
geolite2.geolite2_accessor.db_info.last
```

The documentation for this class was generated from the following file:

- [geolite2.py](#)

6.2 `test_rest.fake` Class Reference

Collaboration diagram for `test_rest.fake`:



Public Member Functions

- def [__getattr__](#) (self, attr)

6.2.1 Member Function Documentation

6.2.1.1 [__getattr__\(\)](#)

```
def test_rest.fake.__getattr__ (
    self,
    attr )
```

The documentation for this class was generated from the following file:

- [test_rest.py](#)

6.3 test_cp.fake_cp_args Class Reference

Collaboration diagram for test_cp.fake_cp_args:

test_cp.fake_cp_args
+ consensus_path
+ update
+ update_keeprunning
+ fixup
+ write_preprocessed
+ additional_info
+ ipaddress
+ preprocessed_input
+ geolite_dir
+ time
+ date_prefix
+ __init__()

Public Member Functions

- def [__init__](#) (self)

Public Attributes

- [consensus_path](#)
- [update](#)
- [update_keeprunning](#)
- [fixup](#)
- [write_preprocessed](#)
- [additional_info](#)
- [ipaddress](#)
- [preprocessed_input](#)
- [geolite_dir](#)
- [time](#)
- [date_prefix](#)

6.3.1 Constructor & Destructor Documentation

6.3.1.1 `__init__()`

```
def test_cp.fake_cp_args.__init__ (
    self )
```

6.3.2 Member Data Documentation

6.3.2.1 `additional_info`

```
test_cp.fake_cp_args.additional_info
```

6.3.2.2 `consensus_path`

```
test_cp.fake_cp_args.consensus_path
```

6.3.2.3 `date_prefix`

```
test_cp.fake_cp_args.date_prefix
```

6.3.2.4 fixup

`test_cp.fake_cp_args.fixup`

6.3.2.5 geolite_dir

`test_cp.fake_cp_args.geolite_dir`

6.3.2.6 ipaddress

`test_cp.fake_cp_args.ipaddress`

6.3.2.7 preprocessed_input

`test_cp.fake_cp_args.preprocessed_input`

6.3.2.8 time

`test_cp.fake_cp_args.time`

6.3.2.9 update

`test_cp.fake_cp_args.update`

6.3.2.10 update_keeprunning

`test_cp.fake_cp_args.update_keeprunning`

6.3.2.11 write_preprocessed

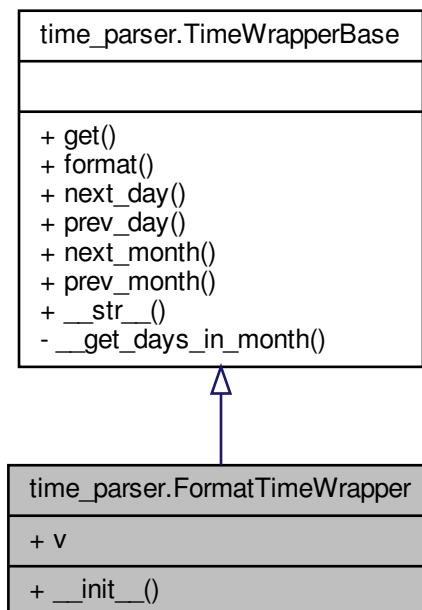
```
test_cp.fake_cp_args.write_preprocessed
```

The documentation for this class was generated from the following file:

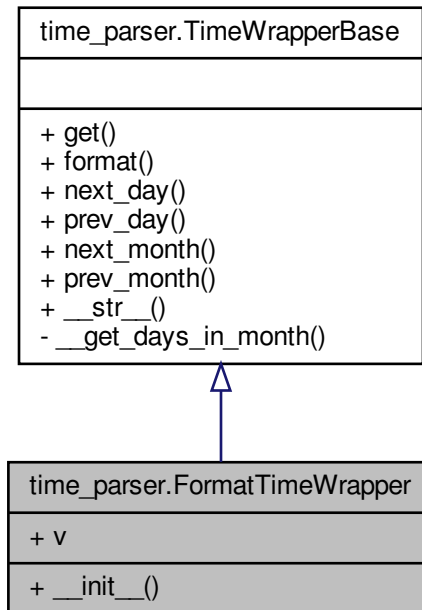
- [test_cp.py](#)

6.4 time_parser.FormatTimeWrapper Class Reference

Inheritance diagram for time_parser.FormatTimeWrapper:



Collaboration diagram for time_parser.FormatTimeWrapper:



Public Member Functions

- `def __init__(self, timestr, formatstr)`

Public Attributes

- `v`

6.4.1 Detailed Description

Converts timestamp with given format to Unix timestamp.

6.4.2 Constructor & Destructor Documentation

6.4.2.1 `__init__()`

```
def time_parser.FormatTimeWrapper.__init__ (
    self,
    timestr,
    formatstr )
```

Overloaded constructor

timestr string with the time data.
format specifies the format of the timestr (see Python time modul doc for details).

Note that the super constructor is not called at all.

6.4.3 Member Data Documentation

6.4.3.1 `v`

```
time_parser.FormatTimeWrapper.v
```

The documentation for this class was generated from the following file:

- [time_parser.py](#)

6.5 geolite2.geolite2_accessor Class Reference

Collaboration diagram for geolite2.geolite2_accessor:

geolite2.geolite2_accessor
<ul style="list-style-type: none"> - <code>__geolite2_dir</code> - <code>__geolite2_dbs</code> - <code>__search_func</code>
<ul style="list-style-type: none"> + <code>__init__()</code> + <code>get_data()</code> + <code>get_geolite2_geolocation()</code> + <code>get_geolite2_as()</code> + <code>create_ordered_dict_base()</code> - <code>__get_directories()</code> - <code>__get_preprocessed_filename_ipaddr()</code> - <code>__get_preprocessed_filename_location()</code> - <code>__get_geolite2_row()</code> - <code>__get_blocks_matching_func()</code>

Classes

- class [db_info](#)

Public Member Functions

- def [__init__](#) (self, path)
- def [get_data](#) (self, ipaddr, geolite2_db, start, end)
- def [get_geolite2_geolocation](#) (self, ip_addr, geolite2_dir)
- def [get_geolite2_as](#) (self, ip_addr, geolite2_dir)

Static Public Member Functions

- def [create_ordered_dict_base](#) ()

Private Member Functions

- def [__get_directories](#) (self, geolite2_db, start_time, end_time)

Static Private Member Functions

- def [__get_preprocessed_filename_ipaddr](#) (geolite2_dir, ip_addr)
- def [__get_preprocessed_filename_location](#) (geolite2_dir, geoname_id)
- def [__get_geolite2_row](#) (matching_func, fname)
- def [__get_blocks_matching_func](#) (ip_addr)

Private Attributes

- [__geolite2_dir](#)
- [__geolite2_dbs](#)
- [__search_func](#)

6.5.1 Constructor & Destructor Documentation

6.5.1.1 [__init__\(\)](#)

```
def geolite2.geolite2_accessor.__init__ (
    self,
    path )
```

6.5.2 Member Function Documentation

6.5.2.1 `__get_blocks_matching_func()`

```
def geolite2.geolite2_accessor.__get_blocks_matching_func (
    ip_addr ) [static], [private]
```

Returns matching function suitable for `get_geolite2_row()`.

6.5.2.2 `__get_directories()`

```
def geolite2.geolite2_accessor.__get_directories (
    self,
    geolite2_db,
    start_time,
    end_time ) [private]
```

Returns a list of two tuples containing directories with GeoLite2 CSVs.

Each item in the returned list is a two tuple: (1) the timestamp and (2) the subdirectory name.

`geolite2_db` Supported City or ASN.

`start_time` Unix timestamp specifying the start of the interval.

`end_time` Unix timestamp specifying the end of the interval.

Note that the function also returns one entry after the end of the interval. If there is a change comparing the last entry before `end_time`, it is possible that it happened during the interval

Here is the caller graph for this function:



6.5.2.3 `__get_geolite2_row()`

```
def geolite2.geolite2_accessor.__get_geolite2_row (
    matching_func,
    fname ) [static], [private]
```

Opens a CSV file and searches for a row matching the `input_str`.

`matching_func` Allows definition of matching, e.g. for performance reasons.

The function should expect one parameter - the first item

in the row and it should return True if the match is successful.

`geolite2_dir` The path to the geolite2 directory containing the CSVs.

`geolite2_type` The CSV name like City-Blocks-IPv4, City-Locations-en, etc.

Returns None if not successful.

Here is the caller graph for this function:



6.5.2.4 __get_preprocessed_filename_ipaddr()

```
def geolite2.geolite2_accessor.__get_preprocessed_filename_ipaddr (
    geolite2_dir,
    ip_addr ) [static], [private]
```

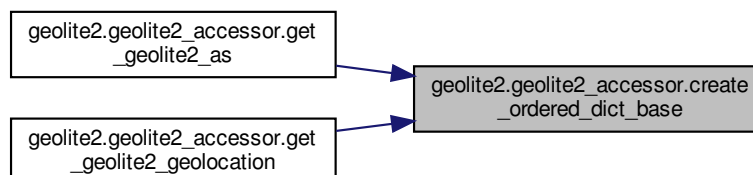
6.5.2.5 __get_preprocessed_filename_location()

```
def geolite2.geolite2_accessor.__get_preprocessed_filename_location (
    geolite2_dir,
    geoname_id ) [static], [private]
```

6.5.2.6 create_ordered_dict_base()

```
def geolite2.geolite2_accessor.create_ordered_dict_base ( ) [static]
```

Here is the caller graph for this function:



6.5.2.7 get_data()

```
def geolite2.geolite2_accessor.get_data (
    self,
    ipaddr,
    geolite2_db,
    start,
    end )
```

Searches for information about an IP address.

ipaddr The IP address to be searched for (IPv4/IPv6)
geolite2_db ASN or City
start, end Time frames during which the search is made.

Returns a list of two-tuples. Each tuple consists of a timestamp and a dictionary with known information. The list is empty if no information is available.

Note that the function also returns one entry after the end of the interval. If there is a change comparing the last entry before *end_time*, it is possible that it happened during the interval.

Here is the call graph for this function:



6.5.2.8 get_geolite2_as()

```
def geolite2.geolite2_accessor.get_geolite2_as (
    self,
    ip_addr,
    geolite2_dir )
```

Returns data about the AS of the IP address.

ip_addr The IP address to be searched for
geolite2_dir The directory with the GeoLite2 CSVs

Returns None if not successfull.

Here is the call graph for this function:



6.5.2.9 get_geolite2_geolocation()

```

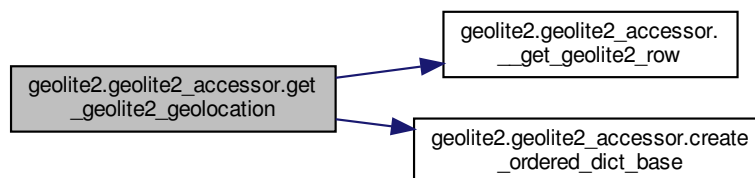
def geolite2.geolite2_accessor.get_geolite2_geolocation (
    self,
    ip_addr,
    geolite2_dir )
  
```

Returns geolocation data about the IP address

`ip_addr` The IP address to be searched for
`geolite2_dir` The directory with the GeoLite2 CSVs

Returns None if not successfull.

Here is the call graph for this function:



6.5.3 Member Data Documentation

6.5.3.1 __geolite2_dbs

`geolite2.geolite2_accessor.__geolite2_dbs` [private]

6.5.3.2 __geolite2_dir

geolite2.geolite2_accessor.__geolite2_dir [private]

6.5.3.3 __search_func

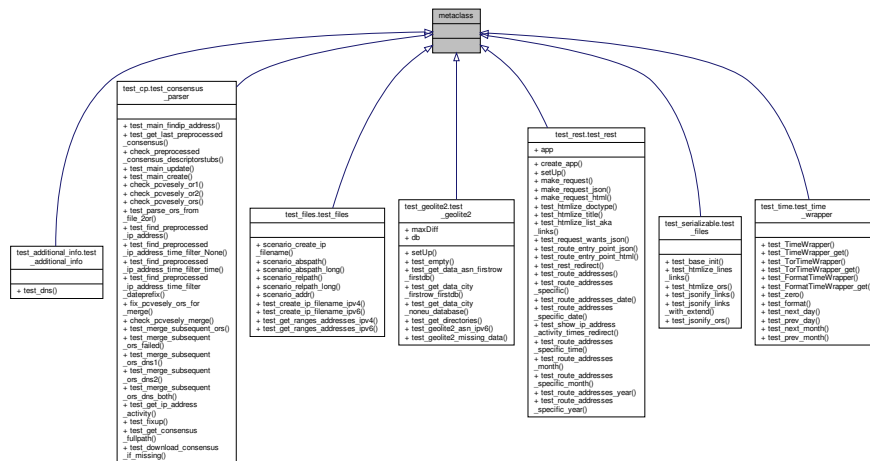
geolite2.geolite2_accessor.__search_func [private]

The documentation for this class was generated from the following file:

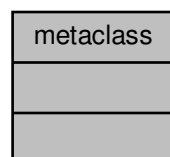
- [geolite2.py](#)

6.6 metaclass Class Reference

Inheritance diagram for metaclass:



Collaboration diagram for metaclass:

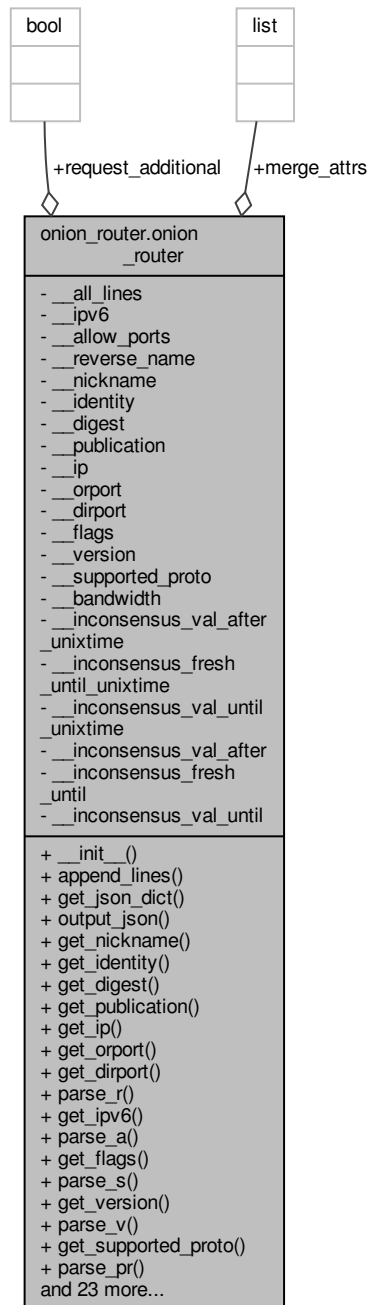


The documentation for this class was generated from the following file:

- [test_files.py](#)

6.7 onion_router.onion_router Class Reference

Collaboration diagram for onion_router.onion_router:



Public Member Functions

- def `__init__` (self, lines)
- def `append_lines` (self, write_file)

- def [get_json_dict](#) (self)
- def [output_json](#) (self, write_file)
- def [get_nickname](#) (self)
- def [get_identity](#) (self)
- def [get_digest](#) (self)
- def [get_publication](#) (self)
- def [get_ip](#) (self)
- def [get_orport](#) (self)
- def [get_dirport](#) (self)
- def [parse_r](#) (self, *args)
- def [get_ipv6](#) (self)
- def [parse_a](#) (self, *args)
- def [get_flags](#) (self)
- def [parse_s](#) (self, *args)
- def [get_version](#) (self)
- def [parse_v](#) (self, *args)
- def [get_supported_proto](#) (self)
- def [parse_pr](#) (self, *args)
- def [get_bandwidth](#) (self)
- def [parse_w](#) (self, *args)
- def [get_allow_ports](#) (self)
- def [parse_p](#) (self, *args)
- def [parse_m](#) (self, *args)
- def [parse_id](#) (self, *args)
- def [get_inconsensus_val_after](#) (self)
- def [get_inconsensus_val_after_unixtime](#) (self)
- def [get_inconsensus_fresh_until](#) (self)
- def [get_inconsensus_fresh_until_unixtime](#) (self)
- def [get_inconsensus_val_until](#) (self)
- def [get_inconsensus_val_until_unixtime](#) (self)
- def [add_network_status_consensus](#) (self, val_after, fresh_until, val_until)
- def [parse_x_inconsensus_valid_after](#) (self, *args)
- def [parse_x_inconsensus_fresh_until](#) (self, *args)
- def [parse_x_inconsensus_valid_until](#) (self, *args)
- def [get_dns_reverse](#) (self)
- def [parse_x_dns_reverse](#) (self, *args)
- def [merge](#) (self, other)
- def [get_maxmind_geolocation](#) (self)
- def [get_maxmind_asn](#) (self)
- def [add_geolite_data](#) (self, data, validity, geolite_type)
- def [append_geolite_data](#) (self, ipaddr, geolite2_access)

Static Public Attributes

- bool [request_additional](#) = False
- list [merge_attrs](#)

Private Attributes

- [__all_lines](#)
- [__ipv6](#)
- [__allow_ports](#)
- [__reverse_name](#)
- [__nickname](#)
- [__identity](#)
- [__digest](#)
- [__publication](#)
- [__ip](#)
- [__orport](#)
- [__dirport](#)
- [__flags](#)
- [__version](#)
- [__supported_proto](#)
- [__bandwidth](#)
- [__inconsensus_val_after_unixtime](#)
- [__inconsensus_fresh_until_unixtime](#)
- [__inconsensus_val_until_unixtime](#)
- [__inconsensus_val_after](#)
- [__inconsensus_fresh_until](#)
- [__inconsensus_val_until](#)

6.7.1 Detailed Description

This class represents information about a single onion router at a specific consensus.

6.7.2 Constructor & Destructor Documentation

6.7.2.1 `__init__()`

```
def onion_router.onion_router.__init__ (
    self,
    lines )
```

6.7.3 Member Function Documentation

6.7.3.1 add_geolite_data()

```
def onion_router.onion_router.add_geolite_data (
    self,
    data,
    validity,
    geolite_type )
```

Here is the caller graph for this function:



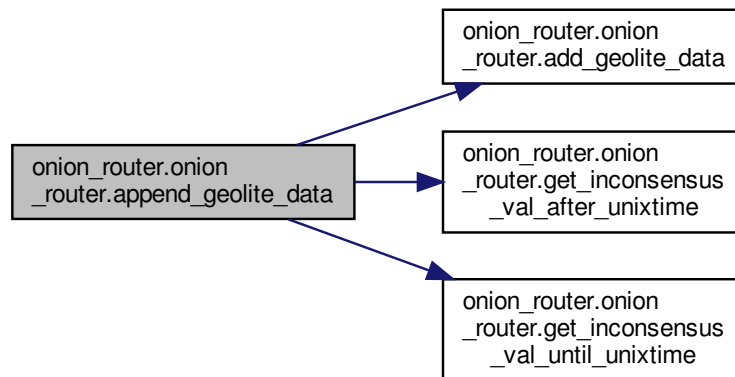
6.7.3.2 add_network_status_consensus()

```
def onion_router.onion_router.add_network_status_consensus (
    self,
    val_after,
    fresh_until,
    val_until )
```

6.7.3.3 append_geolite_data()

```
def onion_router.onion_router.append_geolite_data (
    self,
    ipaddr,
    geolite2_access )
```

Here is the call graph for this function:



6.7.3.4 append_lines()

```
def onion_router.onion_router.append_lines (
    self,
    write_file )
```

6.7.3.5 get_allow_ports()

```
def onion_router.onion_router.get_allow_ports (
    self )
```

6.7.3.6 get_bandwidth()

```
def onion_router.onion_router.get_bandwidth (
    self )
```

6.7.3.7 get_digest()

```
def onion_router.onion_router.get_digest (
    self )
```

6.7.3.8 get_dirport()

```
def onion_router.onion_router.get_dirport (
    self )
```

6.7.3.9 get_dns_reverse()

```
def onion_router.onion_router.get_dns_reverse (
    self )
```

6.7.3.10 get_flags()

```
def onion_router.onion_router.get_flags (
    self )
```

6.7.3.11 get_identity()

```
def onion_router.onion_router.get_identity (
    self )
```

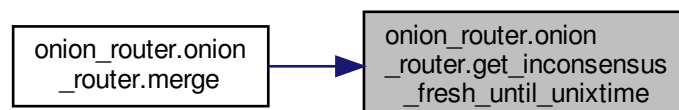
6.7.3.12 get_inconsensus_fresh_until()

```
def onion_router.onion_router.get_inconsensus_fresh_until (
    self )
```

6.7.3.13 get_inconsensus_fresh_until_unixtime()

```
def onion_router.onion_router.get_inconsensus_fresh_until_unixtime (
    self )
```

Here is the caller graph for this function:



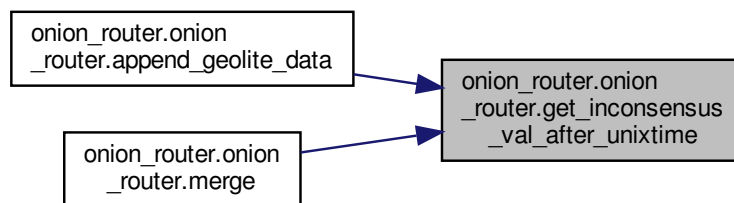
6.7.3.14 get_inconsensus_val_after()

```
def onion_router.onion_router.get_inconsensus_val_after (
    self )
```

6.7.3.15 get_inconsensus_val_after_unixtime()

```
def onion_router.onion_router.get_inconsensus_val_after_unixtime (
    self )
```

Here is the caller graph for this function:



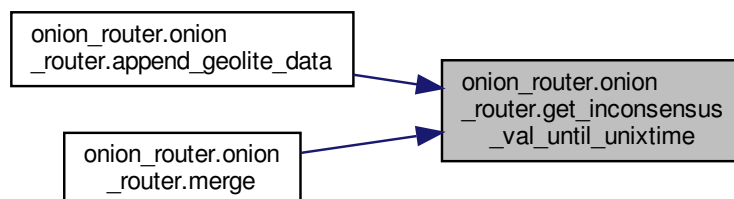
6.7.3.16 get_inconsensus_val_until()

```
def onion_router.onion_router.get_inconsensus_val_until (
    self )
```

6.7.3.17 get_inconsensus_val_until_unixtime()

```
def onion_router.onion_router.get_inconsensus_val_until_unixtime (
    self )
```

Here is the caller graph for this function:



6.7.3.18 get_ip()

```
def onion_router.onion_router.get_ip (
    self )
```

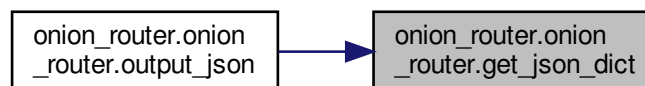
6.7.3.19 get_ipv6()

```
def onion_router.onion_router.get_ipv6 (
    self )
```

6.7.3.20 get_json_dict()

```
def onion_router.onion_router.get_json_dict (
    self )
```

Here is the caller graph for this function:



6.7.3.21 get_maxmind_asn()

```
def onion_router.onion_router.get_maxmind_asn (
    self )
```

6.7.3.22 get_maxmind_geolocation()

```
def onion_router.onion_router.get_maxmind_geolocation (
    self )
```

6.7.3.23 get_nickname()

```
def onion_router.onion_router.get_nickname (
    self )
```

6.7.3.24 get_orport()

```
def onion_router.onion_router.get_orport (
    self )
```

6.7.3.25 get_publication()

```
def onion_router.onion_router.get_publication (
    self )
```

6.7.3.26 get_supported_proto()

```
def onion_router.onion_router.get_supported_proto (
    self )
```

6.7.3.27 get_version()

```
def onion_router.onion_router.get_version (
    self )
```

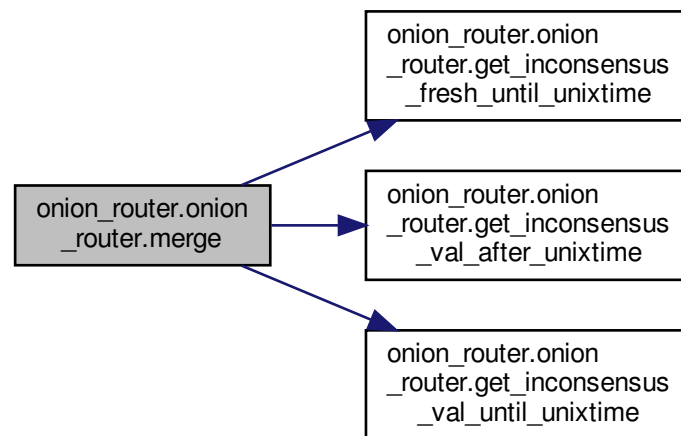
6.7.3.28 merge()

```
def onion_router.onion_router.merge (
    self,
    other )
```

Merges information from other to self if allowed.

Only entries with the same attributes can be merged, so only `inconsensus` and `DNS reverse query time` may differ. Returns `true` if the original was updated,

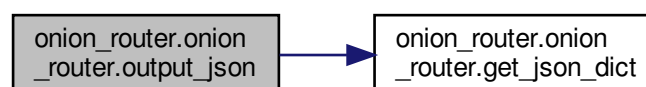
Here is the call graph for this function:



6.7.3.29 output_json()

```
def onion_router.onion_router.output_json (
    self,
    write_file )
```

Here is the call graph for this function:



6.7.3.30 parse_a()

```
def onion_router.onion_router.parse_a (
    self,
    * args )
```

6.7.3.31 parse_id()

```
def onion_router.onion_router.parse_id (
    self,
    * args )
```

6.7.3.32 parse_m()

```
def onion_router.onion_router.parse_m (
    self,
    * args )
```

6.7.3.33 parse_p()

```
def onion_router.onion_router.parse_p (
    self,
    * args )
```

6.7.3.34 parse_pr()

```
def onion_router.onion_router.parse_pr (
    self,
    * args )
```

6.7.3.35 parse_r()

```
def onion_router.onion_router.parse_r (
    self,
    * args )
```

6.7.3.36 parse_s()

```
def onion_router.onion_router.parse_s (
    self,
    * args )
```

6.7.3.37 parse_v()

```
def onion_router.onion_router.parse_v (
    self,
    * args )
```

6.7.3.38 parse_w()

```
def onion_router.onion_router.parse_w (
    self,
    * args )
```

6.7.3.39 parse_x_dns_reverse()

```
def onion_router.onion_router.parse_x_dns_reverse (
    self,
    * args )
```

6.7.3.40 parse_x_inconsensus_fresh_until()

```
def onion_router.onion_router.parse_x_inconsensus_fresh_until (
    self,
    * args )
```

6.7.3.41 parse_x_inconsensus_valid_after()

```
def onion_router.onion_router.parse_x_inconsensus_valid_after (
    self,
    * args )
```

6.7.3.42 parse_x_inconsensus_valid_until()

```
def onion_router.onion_router.parse_x_inconsensus_valid_until (
    self,
    * args )
```

6.7.4 Member Data Documentation

6.7.4.1 __all_lines

onion_router.onion_router.__all_lines [private]

6.7.4.2 __allow_ports

onion_router.onion_router.__allow_ports [private]

6.7.4.3 __bandwidth

onion_router.onion_router.__bandwidth [private]

6.7.4.4 __digest

onion_router.onion_router.__digest [private]

6.7.4.5 __dirport

onion_router.onion_router.__dirport [private]

6.7.4.6 __flags

onion_router.onion_router.__flags [private]

6.7.4.7 `__identity`

`onion_router.onion_router.__identity` [private]

6.7.4.8 `__inconsensus_fresh_until`

`onion_router.onion_router.__inconsensus_fresh_until` [private]

6.7.4.9 `__inconsensus_fresh_until_unixtime`

`onion_router.onion_router.__inconsensus_fresh_until_unixtime` [private]

6.7.4.10 `__inconsensus_val_after`

`onion_router.onion_router.__inconsensus_val_after` [private]

6.7.4.11 `__inconsensus_val_after_unixtime`

`onion_router.onion_router.__inconsensus_val_after_unixtime` [private]

6.7.4.12 `__inconsensus_val_until`

`onion_router.onion_router.__inconsensus_val_until` [private]

6.7.4.13 `__inconsensus_val_until_unixtime`

`onion_router.onion_router.__inconsensus_val_until_unixtime` [private]

6.7.4.14 `__ip`

`onion_router.onion_router.__ip` [private]

6.7.4.15 `__ipv6`

`onion_router.onion_router.__ipv6` [private]

6.7.4.16 `__nickname`

`onion_router.onion_router.__nickname` [private]

6.7.4.17 `__orport`

`onion_router.onion_router.__orport` [private]

6.7.4.18 `__publication`

`onion_router.onion_router.__publication` [private]

6.7.4.19 `__reverse_name`

`onion_router.onion_router.__reverse_name` [private]

6.7.4.20 `__supported_proto`

`onion_router.onion_router.__supported_proto` [private]

6.7.4.21 `__version`

`onion_router.onion_router.__version` [private]

6.7.4.22 merge_attrs

```
list onion_router.onion_router.merge_attrs [static]
```

Initial value:

```
= [{"_onion_router__" + attr for attr in ["nickname", "identity", \
    "digest", "publication", "ip", "orport", "dirport", "ipv6", "flags", \
    "version", "supported_proto", "bandwidth", "allow_ports"]}]
```

6.7.4.23 request_additional

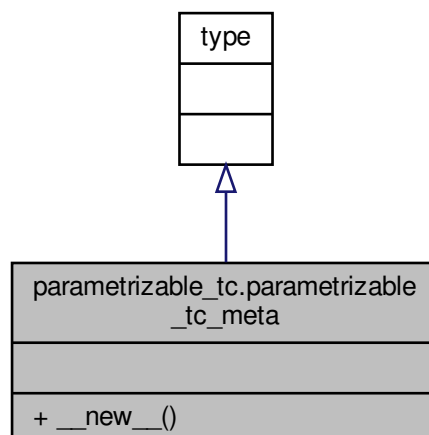
```
bool onion_router.onion_router.request_additional = False [static]
```

The documentation for this class was generated from the following file:

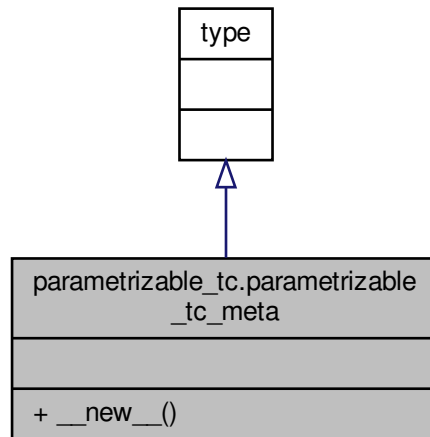
- [onion_router.py](#)

6.8 parametrizable_tc.parametrizable_tc_meta Class Reference

Inheritance diagram for parametrizable_tc.parametrizable_tc_meta:



Collaboration diagram for parametrizable_tc.parametrizable_tc_meta:



Public Member Functions

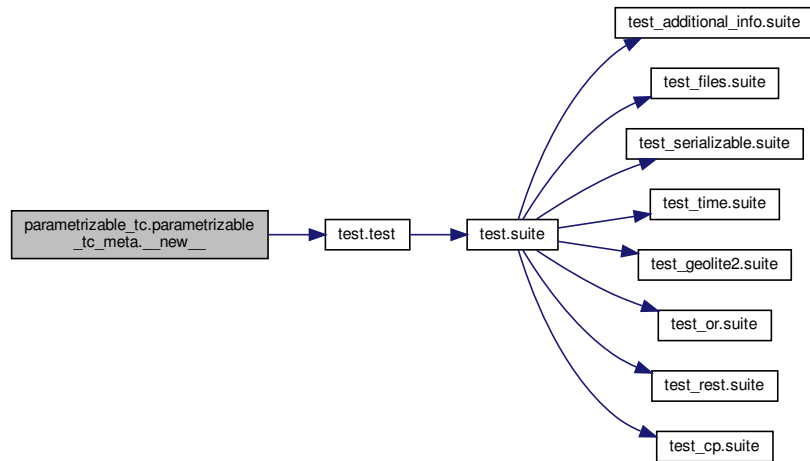
- `def __new__(cls, name, bases, bodydict)`

6.8.1 Member Function Documentation

6.8.1.1 `__new__()`

```
def parametrizable_tc.parametrizable_tc_meta.__new__(
    cls,
    name,
    bases,
    bodydict )
```

Here is the call graph for this function:

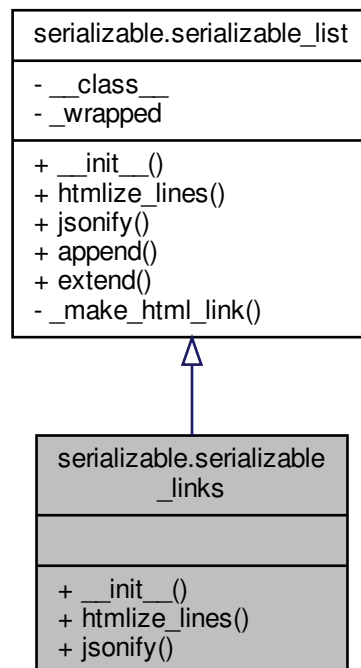


The documentation for this class was generated from the following file:

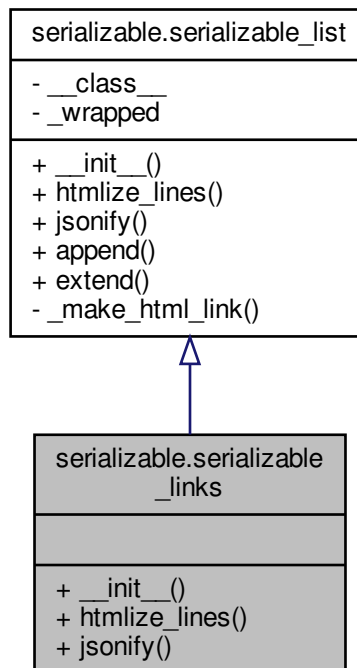
- [parametrizable_tc.py](#)

6.9 serializable.serializable_links Class Reference

Inheritance diagram for serializable.serializable_links:



Collaboration diagram for `serializable.serializable_links`:



Public Member Functions

- `def __init__(self, lst)`
- `def htmlize_lines(self)`
- `def jsonify(self)`

6.9.1 Detailed Description

Provides functionality to HTMLize and JSONify links.

Stores a list of two-tuples - (Link description, URI).

6.9.2 Constructor & Destructor Documentation

6.9.2.1 __init__()

```
def serializable.serializable_links.__init__(
    self,
    lst )
```

Reimplemented from [serializable.serializable_list](#).

6.9.3 Member Function Documentation

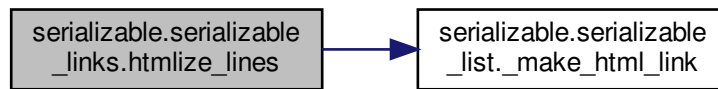
6.9.3.1 htmlize_lines()

```
def serializable.serializable_links.htmlize_lines (  
    self )
```

Returns a list of lines that represents the iterable.

Reimplemented from [serializable.serializable_list](#).

Here is the call graph for this function:



6.9.3.2 jsonify()

```
def serializable.serializable_links.jsonify (  
    self )
```

Returns an object to be JSONified.

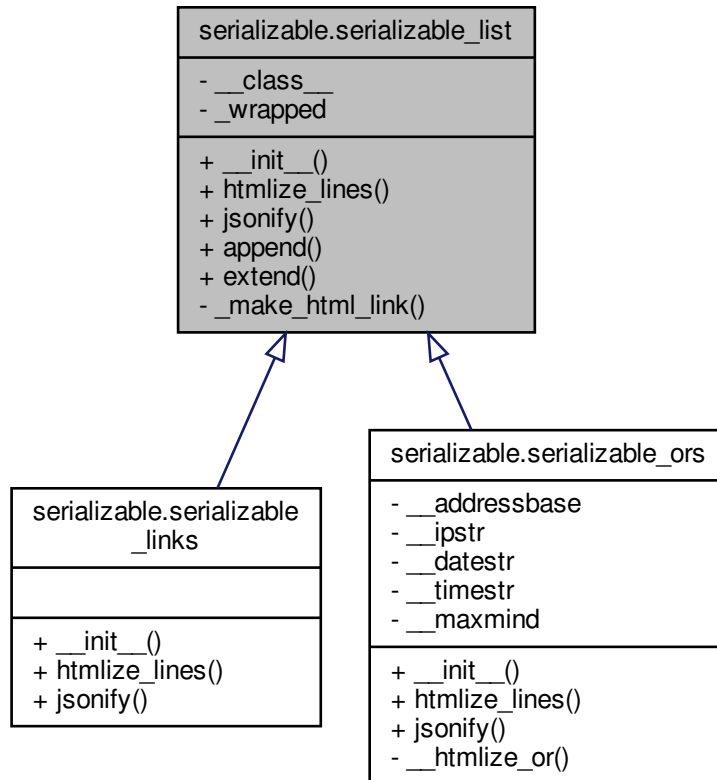
Reimplemented from [serializable.serializable_list](#).

The documentation for this class was generated from the following file:

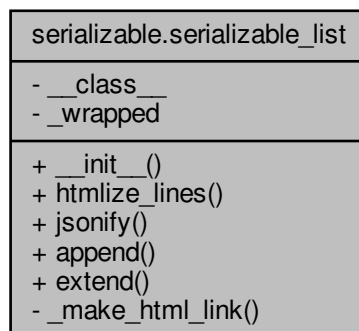
- [serializable.py](#)

6.10 serializable.serializable_list Class Reference

Inheritance diagram for serializable.serializable_list:



Collaboration diagram for serializable.serializable_list:



Public Member Functions

- def [__init__](#) (self, lst)
- def [htmlize_lines](#) (self)
- def [jsonify](#) (self)
- def [append](#) (self, item)
- def [extend](#) (self, other)

Static Private Member Functions

- def [_make_html_link](#) (url, desc)

Private Attributes

- [__class__](#)
- [_wrapped](#)

6.10.1 Detailed Description

This is a base class intended to be subclassed by specialized serializers.

6.10.2 Constructor & Destructor Documentation

6.10.2.1 [__init__\(\)](#)

```
def serializable.serializable_list.__init__ (
    self,
    lst )
```

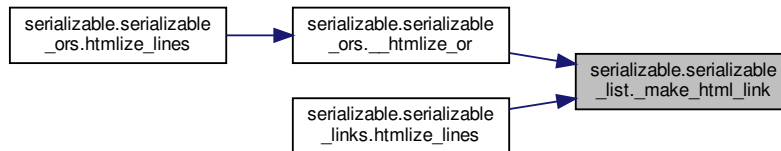
Reimplemented in [serializable.serializable_links](#).

6.10.3 Member Function Documentation

6.10.3.1 `_make_html_link()`

```
def serializable.serializable_list._make_html_link (
    url,
    desc ) [static], [private]
```

Here is the caller graph for this function:



6.10.3.2 `append()`

```
def serializable.serializable_list.append (
    self,
    item )
```

6.10.3.3 `extend()`

```
def serializable.serializable_list.extend (
    self,
    other )
```

6.10.3.4 `htmlize_lines()`

```
def serializable.serializable_list.htmlize_lines (
    self )
```

Returns a list of lines that represents the iterable.

Reimplemented in [serializable.serializable_links](#), and [serializable.serializable_ors](#).

6.10.3.5 jsonify()

```
def serializable.serializable_list.jsonify (
    self )
```

Returns an object to be JSONified.

Reimplemented in [serializable.serializable_links](#), and [serializable.serializable_ors](#).

6.10.4 Member Data Documentation

6.10.4.1 __class__

`serializable.serializable_list.__class__` [private]

6.10.4.2 _wrapped

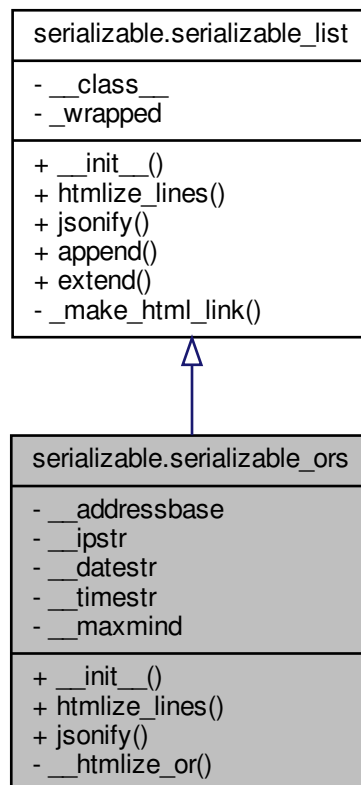
`serializable.serializable_list._wrapped` [private]

The documentation for this class was generated from the following file:

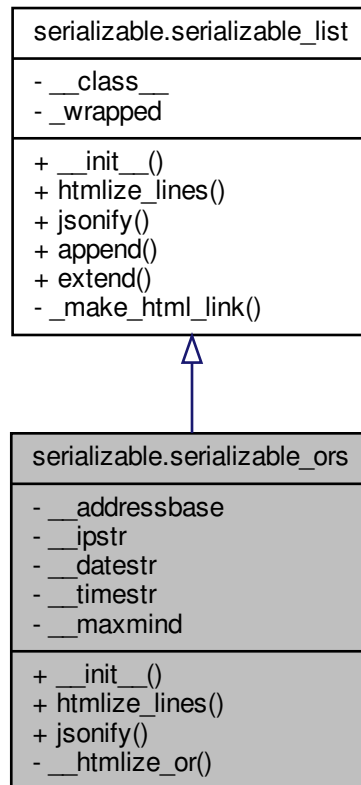
- [serializable.py](#)

6.11 serializable.serializable_ors Class Reference

Inheritance diagram for serializable.serializable_ors:



Collaboration diagram for serializable.serializable_ors:



Public Member Functions

- def `__init__` (self, lst, addressbase, ipstr, datestr, timestr)
- def `htmlize_lines` (self)
- def `jsonify` (self)

Private Member Functions

- def `__htmlize_or` (self, r, res)

Private Attributes

- `__addressbase`
- `__ipstr`
- `__datestr`
- `__timestr`
- `__maxmind`

6.11.1 Detailed Description

Provides functionality to HTMLize and JSONify ORs.

6.11.2 Constructor & Destructor Documentation

6.11.2.1 `__init__()`

```
def serializable.serializable_ors.__init__ (
    self,
    lst,
    addressbase,
    ipstr,
    datestr,
    timestr )
```

6.11.3 Member Function Documentation

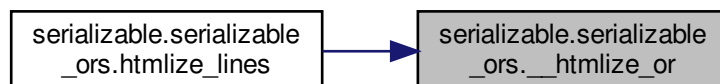
6.11.3.1 `__htmlize_or()`

```
def serializable.serializable_ors.__htmlize_or (
    self,
    r,
    res ) [private]
```

Here is the call graph for this function:



Here is the caller graph for this function:



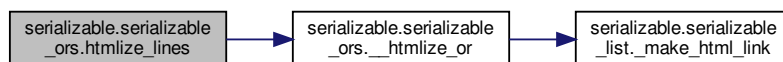
6.11.3.2 htmlize_lines()

```
def serializable.serializable_ors.htmlize_lines (
    self )
```

Returns a list of lines that represents the iterable.

Reimplemented from [serializable.serializable_list](#).

Here is the call graph for this function:



6.11.3.3 jsonify()

```
def serializable.serializable_ors.jsonify (
    self )
```

Returns an object to be JSONified.

Reimplemented from [serializable.serializable_list](#).

6.11.4 Member Data Documentation

6.11.4.1 __addressbase

```
serializable.serializable_ors.__addressbase [private]
```

6.11.4.2 __datestr

```
serializable.serializable_ors.__datestr [private]
```

6.11.4.3 `__ipstr`

`serializable.serializable_ors.__ipstr` [private]

6.11.4.4 `__maxmind`

`serializable.serializable_ors.__maxmind` [private]

6.11.4.5 `__timestr`

`serializable.serializable_ors.__timestr` [private]

The documentation for this class was generated from the following file:

- [serializable.py](#)

6.12 `snapshot.Snapshot` Class Reference

Collaboration diagram for `snapshot.Snapshot`:

<code>snapshot.Snapshot</code>
<ul style="list-style-type: none">- <code>__obj_lines</code>- <code>__routers</code>- <code>__unparsed_lines</code>- <code>__ip</code>- <code>__valid_after</code>- <code>__fresh_until</code>- <code>__valid_until</code>
<ul style="list-style-type: none">+ <code>__init__()</code>+ <code>get_onion_router()</code>+ <code>get_ipaddrs()</code>+ <code>get_valid_after()</code>+ <code>parse_valid_after()</code>+ <code>parse_fresh_until()</code>+ <code>parse_valid_until()</code>

Public Member Functions

- def `__init__` (self, obj_lines, orouters)
- def `get_onion_router` (self, ipaddr)
- def `get_ipaddrs` (self)
- def `get_valid_after` (self)
- def `parse_valid_after` (self, date, time)
- def `parse_fresh_until` (self, date, time)
- def `parse_valid_until` (self, date, time)

Private Attributes

- `__obj_lines`
- `__routers`
- `__unparsed_lines`
- `__ip`
- `__valid_after`
- `__fresh_until`
- `__valid_until`

6.12.1 Constructor & Destructor Documentation

6.12.1.1 `__init__()`

```
def snapshot.Snapshot.__init__ (
    self,
    obj_lines,
    orouters )
```

6.12.2 Member Function Documentation

6.12.2.1 `get_ipaddrs()`

```
def snapshot.Snapshot.get_ipaddrs (
    self )
```

6.12.2.2 `get_onion_router()`

```
def snapshot.Snapshot.get_onion_router (
    self,
    ipaddr )
```

6.12.2.3 `get_valid_after()`

```
def snapshot.Snapshot.get_valid_after (
    self )
```

6.12.2.4 `parse_fresh_until()`

```
def snapshot.Snapshot.parse_fresh_until (
    self,
    date,
    time )
```

6.12.2.5 `parse_valid_after()`

```
def snapshot.Snapshot.parse_valid_after (
    self,
    date,
    time )
```

6.12.2.6 `parse_valid_until()`

```
def snapshot.Snapshot.parse_valid_until (
    self,
    date,
    time )
```

6.12.3 Member Data Documentation

6.12.3.1 `__fresh_until`

```
snapshot.Snapshot.__fresh_until [private]
```

6.12.3.2 `__ip`

```
snapshot.Snapshot.__ip [private]
```

6.12.3.3 `__obj_lines`

`snapshot.Snapshot.__obj_lines` [private]

6.12.3.4 `__routers`

`snapshot.Snapshot.__routers` [private]

6.12.3.5 `__unparsed_lines`

`snapshot.Snapshot.__unparsed_lines` [private]

6.12.3.6 `__valid_after`

`snapshot.Snapshot.__valid_after` [private]

6.12.3.7 `__valid_until`

`snapshot.Snapshot.__valid_until` [private]

The documentation for this class was generated from the following file:

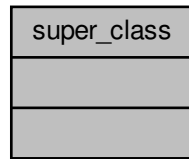
- [snapshot.py](#)

6.13 super_class Class Reference

Inheritance diagram for super_class:



Collaboration diagram for super_class:

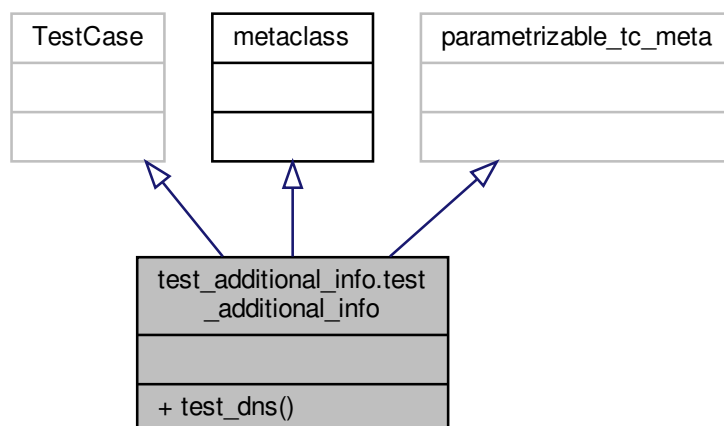


The documentation for this class was generated from the following file:

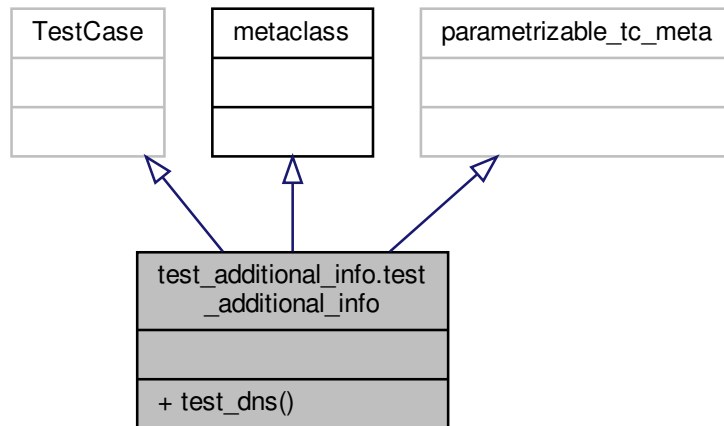
- [test_rest.py](#)

6.14 test_additional_info.test_additional_info Class Reference

Inheritance diagram for test_additional_info.test_additional_info:



Collaboration diagram for `test_additional_info.test_additional_info`:



Public Member Functions

- def `test_dns` (self, ip_addr, expected)

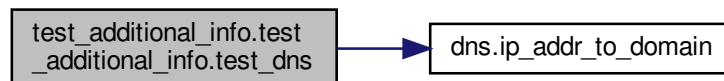
6.14.1 Member Function Documentation

6.14.1.1 test_dns()

```

def test_additional_info.test_additional_info.test_dns (
    self,
    ip_addr,
    expected )
  
```

Here is the call graph for this function:

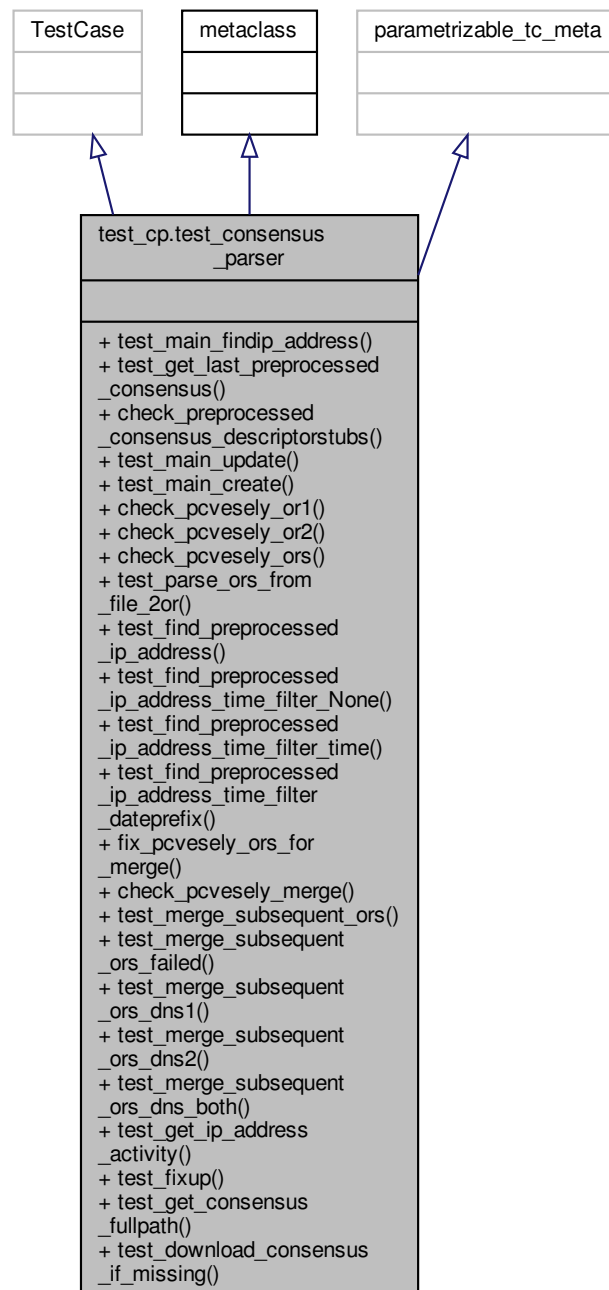


The documentation for this class was generated from the following file:

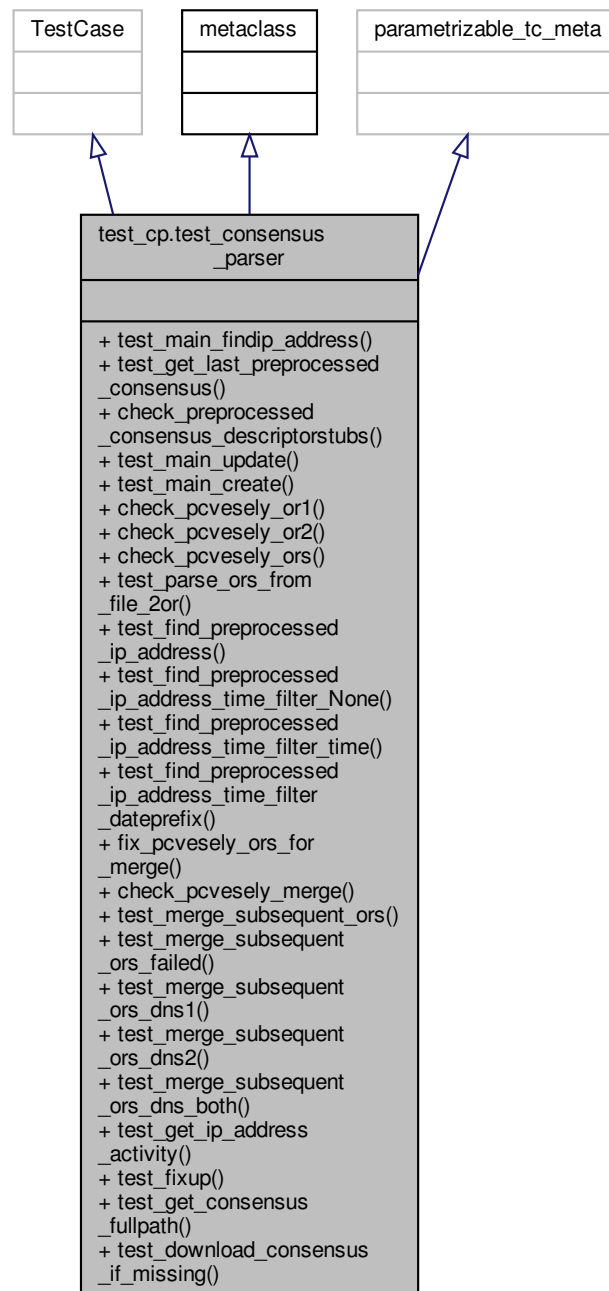
- [test_additional_info.py](#)

6.15 test_cp.test_consensus_parser Class Reference

Inheritance diagram for test_cp.test_consensus_parser:



Collaboration diagram for `test_cp.test_consensus_parser`:



Public Member Functions

- def [test_main_findip_address](#) (self, t, date, expected)
- def [test_get_last_preprocessed_consensus](#) (self)
- def [check_preprocessed_consensus_descriptorstubs](#) (self)
- def [test_main_update](#) (self)
- def [test_main_create](#) (self)

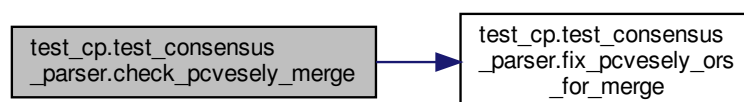
- def [check_pcvesely_or1](#) (self, orouter)
- def [check_pcvesely_or2](#) (self, orouter)
- def [check_pcvesely_ors](#) (self, ors)
- def [test_parse_ors_from_file_2or](#) (self)
- def [test_find_preprocessed_ip_address](#) (self)
- def [test_find_preprocessed_ip_address_time_filter_None](#) (self)
- def [test_find_preprocessed_ip_address_time_filter_time](#) (self)
- def [test_find_preprocessed_ip_address_time_filter_dateprefix](#) (self, date_prefix, checkfunc)
- def [fix_pcvesely_ors_for_merge](#) (self, ors)
- def [check_pcvesely_merge](#) (self, ors)
- def [test_merge_subsequent_ors](#) (self)
- def [test_merge_subsequent_ors_failed](#) (self)
- def [test_merge_subsequent_ors_dns1](#) (self)
- def [test_merge_subsequent_ors_dns2](#) (self)
- def [test_merge_subsequent_ors_dns_both](#) (self)
- def [test_get_ip_address_activity](#) (self)
- def [test_fixup](#) (self)
- def [test_get_consensus_fullpath](#) (self, abspath, t, directory, fname)
- def [test_download_consensus_if_missing](#) (self)

6.15.1 Member Function Documentation

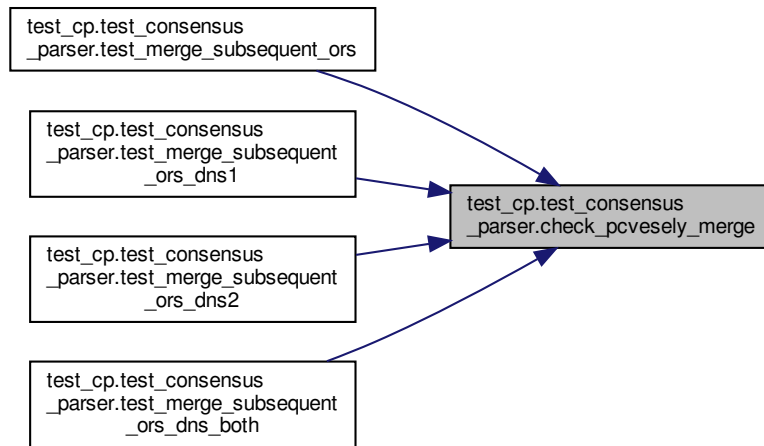
6.15.1.1 [check_pcvesely_merge\(\)](#)

```
def test_cp.test_consensus_parser.check_pcvesely_merge (  
    self,  
    ors )
```

Here is the call graph for this function:



Here is the caller graph for this function:

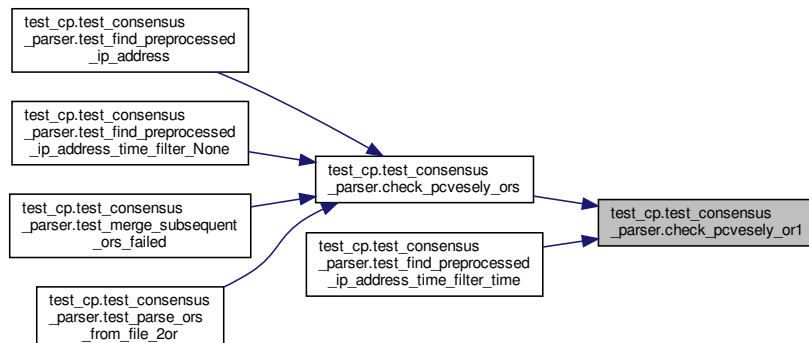


6.15.1.2 check_pcvesely_or1()

```

def test_cp.test_consensus_parser.check_pcvesely_or1 (
    self,
    router )
  
```

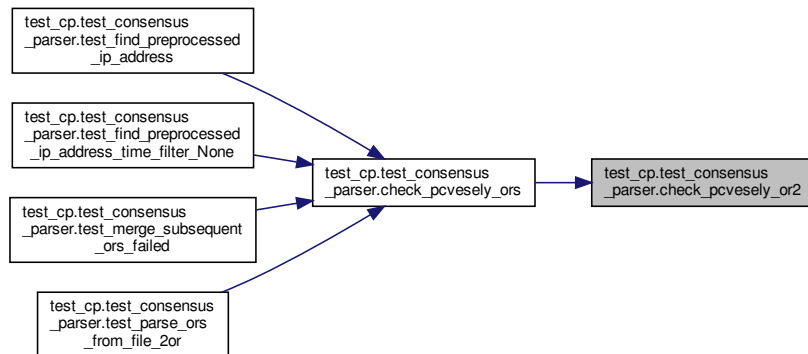
Here is the caller graph for this function:



6.15.1.3 check_pcvesely_or2()

```
def test_cp.test_consensus_parser.check_pcvesely_or2 (
    self,
    orouter )
```

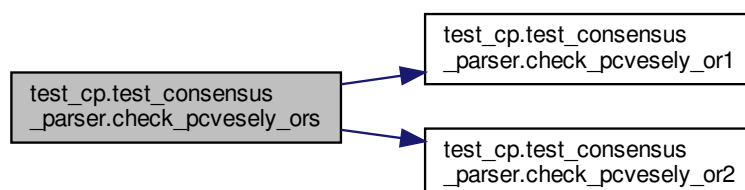
Here is the caller graph for this function:



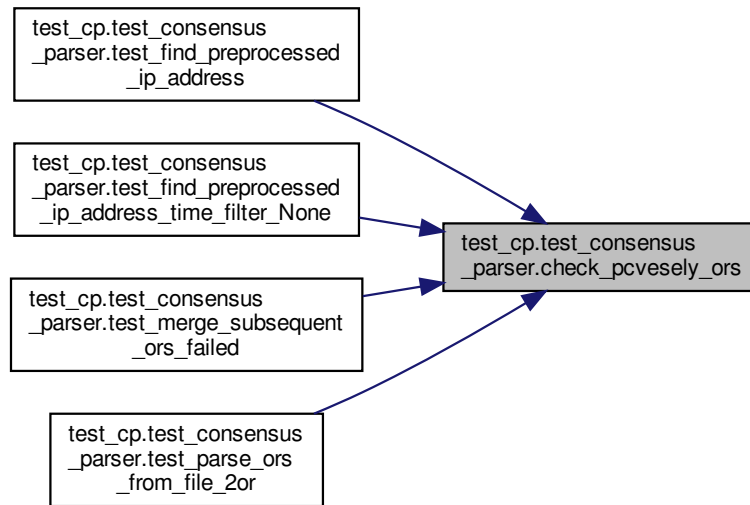
6.15.1.4 check_pcvesely_ors()

```
def test_cp.test_consensus_parser.check_pcvesely_ors (
    self,
    ors )
```

Here is the call graph for this function:



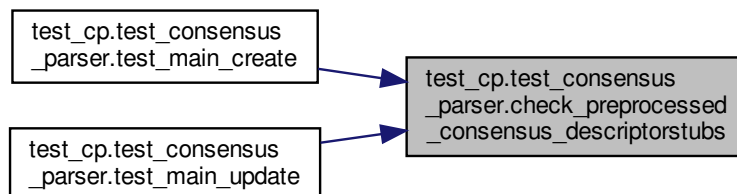
Here is the caller graph for this function:



6.15.1.5 check_preprocessed_consensus_descriptorstubs()

```
def test_cp.test_consensus_parser.check_preprocessed_consensus_descriptorstubs (
    self )
```

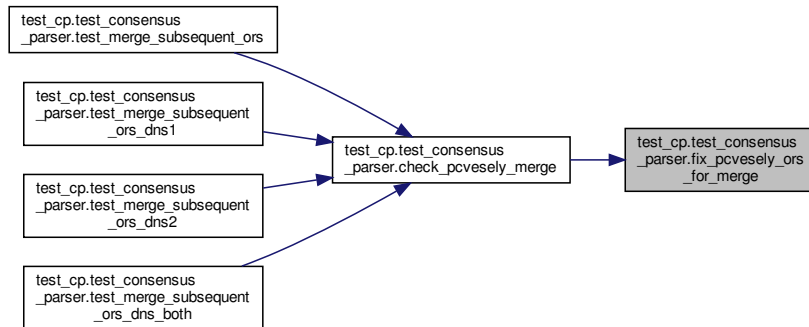
Here is the caller graph for this function:



6.15.1.6 fix_pcvesely_ors_for_merge()

```
def test_cp.test_consensus_parser.fix_pcvesely_ors_for_merge (
    self,
    ors )
```

Here is the caller graph for this function:



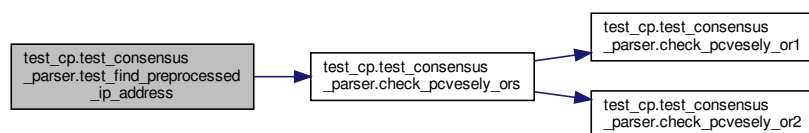
6.15.1.7 test_download_consensus_if_missing()

```
def test_cp.test_consensus_parser.test_download_consensus_if_missing (
    self )
```

6.15.1.8 test_find_preprocessed_ip_address()

```
def test_cp.test_consensus_parser.test_find_preprocessed_ip_address (
    self )
```

Here is the call graph for this function:



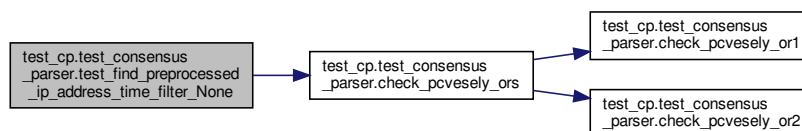
6.15.1.9 test_find_preprocessed_ip_address_time_filter_dateprefix()

```
def test_cp.test_consensus_parser.test_find_preprocessed_ip_address_time_filter_dateprefix (
    self,
    date_prefix,
    checkfunc )
```

6.15.1.10 test_find_preprocessed_ip_address_time_filter_None()

```
def test_cp.test_consensus_parser.test_find_preprocessed_ip_address_time_filter_None (
    self )
```

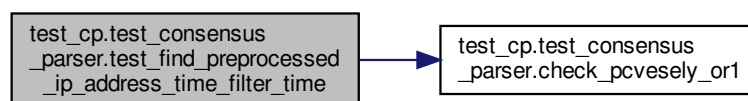
Here is the call graph for this function:



6.15.1.11 test_find_preprocessed_ip_address_time_filter_time()

```
def test_cp.test_consensus_parser.test_find_preprocessed_ip_address_time_filter_time (
    self )
```

Here is the call graph for this function:



6.15.1.12 test_fixup()

```
def test_cp.test_consensus_parser.test_fixup (
    self )
```

6.15.1.13 test_get_consensus_fullpath()

```
def test_cp.test_consensus_parser.test_get_consensus_fullpath (
    self,
    abspath,
    t,
    directory,
    fname )
```

6.15.1.14 test_get_ip_address_activity()

```
def test_cp.test_consensus_parser.test_get_ip_address_activity (
    self )
```

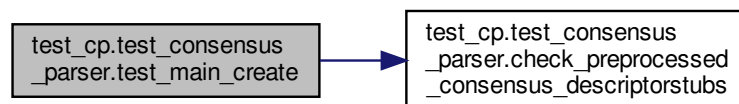
6.15.1.15 test_get_last_preprocessed_consensus()

```
def test_cp.test_consensus_parser.test_get_last_preprocessed_consensus (
    self )
```

6.15.1.16 test_main_create()

```
def test_cp.test_consensus_parser.test_main_create (
    self )
```

Here is the call graph for this function:



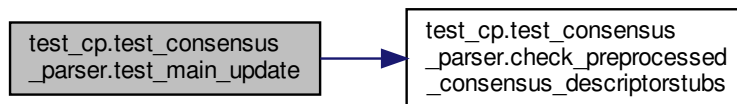
6.15.1.17 test_main_findip_address()

```
def test_cp.test_consensus_parser.test_main_findip_address (
    self,
    t,
    date,
    expected )
```

6.15.1.18 test_main_update()

```
def test_cp.test_consensus_parser.test_main_update (
    self )
```

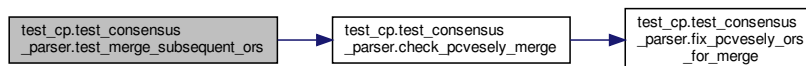
Here is the call graph for this function:



6.15.1.19 test_merge_subsequent_ors()

```
def test_cp.test_consensus_parser.test_merge_subsequent_ors (
    self )
```

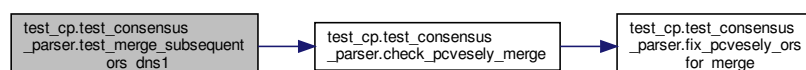
Here is the call graph for this function:



6.15.1.20 test_merge_subsequent_ors_dns1()

```
def test_cp.test_consensus_parser.test_merge_subsequent_ors_dns1 (
    self )
```

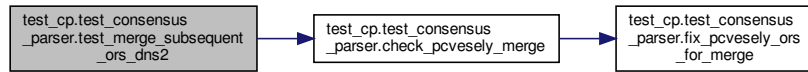
Here is the call graph for this function:



6.15.1.21 test_merge_subsequent_ors_dns2()

```
def test_cp.test_consensus_parser.test_merge_subsequent_ors_dns2 (
    self )
```

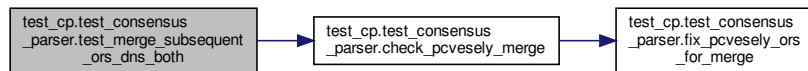
Here is the call graph for this function:



6.15.1.22 test_merge_subsequent_ors_dns_both()

```
def test_cp.test_consensus_parser.test_merge_subsequent_ors_dns_both (
    self )
```

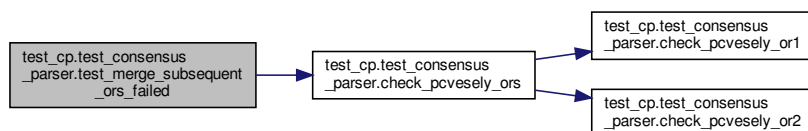
Here is the call graph for this function:



6.15.1.23 test_merge_subsequent_ors_failed()

```
def test_cp.test_consensus_parser.test_merge_subsequent_ors_failed (
    self )
```

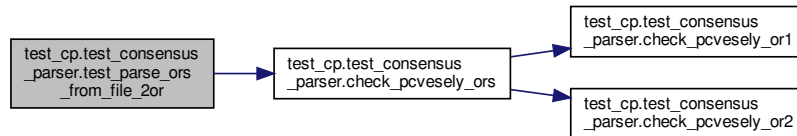
Here is the call graph for this function:



6.15.1.24 test_parse_ors_from_file_2or()

```
def test_cp.test_consensus_parser.test_parse_ors_from_file_2or (
    self )
```

Here is the call graph for this function:

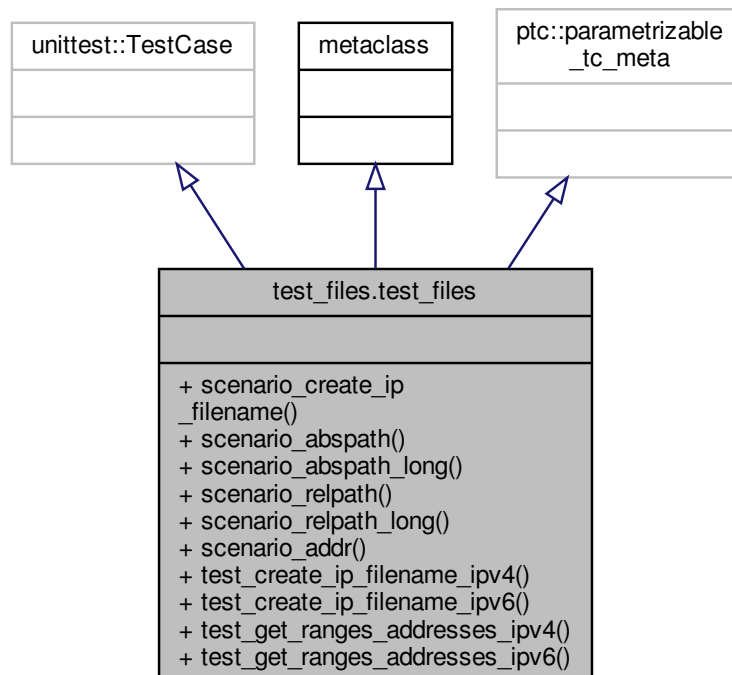


The documentation for this class was generated from the following file:

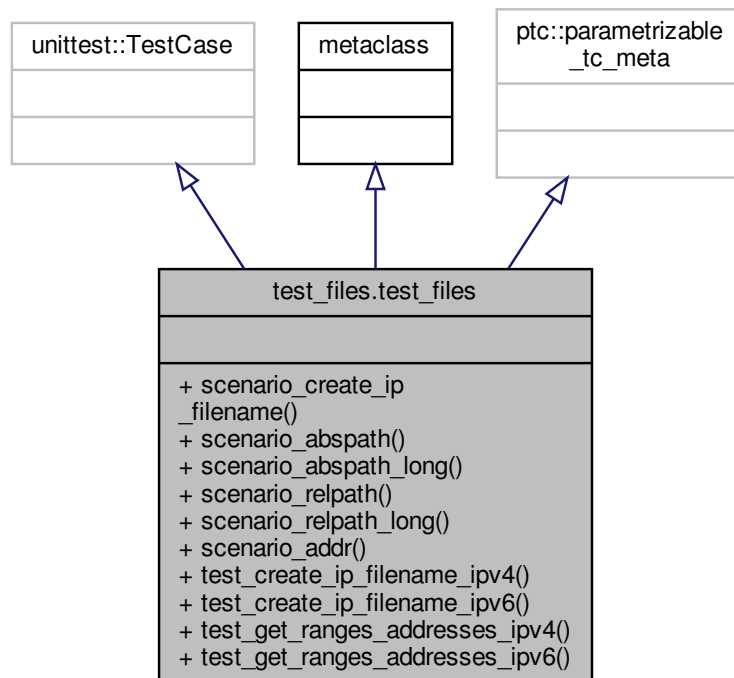
- [test_cp.py](#)

6.16 test_files.test_files Class Reference

Inheritance diagram for `test_files.test_files`:



Collaboration diagram for test_files.test_files:



Public Member Functions

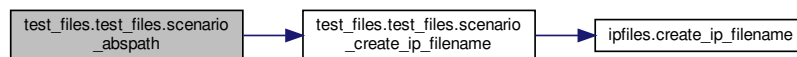
- def [scenario_create_ip_filename](#) (self, path_no_slash, d1, d2, ipaddr)
- def [scenario_abspath](#) (self, d1, d2, ipaddr)
- def [scenario_abspath_long](#) (self, d1, d2, ipaddr)
- def [scenario_relpath](#) (self, d1, d2, ipaddr)
- def [scenario_relpath_long](#) (self, d1, d2, ipaddr)
- def [scenario_addr](#) (self, d1, d2, ipaddr)
- def [test_create_ip_filename_ipv4](#) (self, ipaddr)
- def [test_create_ip_filename_ipv6](#) (self, *args)
- def [test_get_ranges_addresses_ipv4](#) (self, network_addr, plen, expected)
- def [test_get_ranges_addresses_ipv6](#) (self, network_addr, plen, expected)

6.16.1 Member Function Documentation

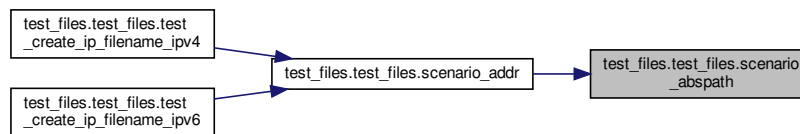
6.16.1.1 scenario_abspath()

```
def test_files.test_files.scenario_abspath (
    self,
    d1,
    d2,
    ipaddr )
```

Here is the call graph for this function:



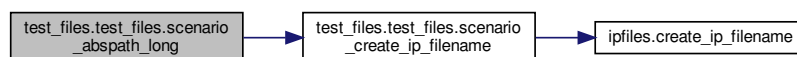
Here is the caller graph for this function:



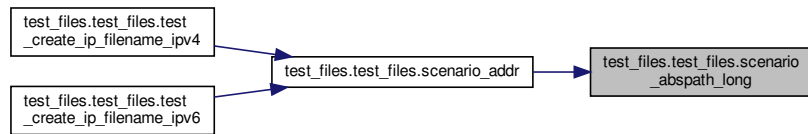
6.16.1.2 scenario_abspath_long()

```
def test_files.test_files.scenario_abspath_long (
    self,
    d1,
    d2,
    ipaddr )
```

Here is the call graph for this function:



Here is the caller graph for this function:

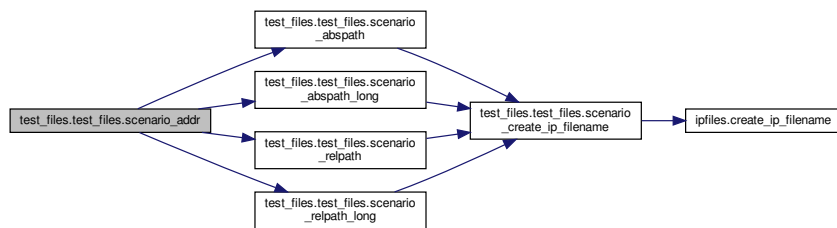


6.16.1.3 scenario_addr()

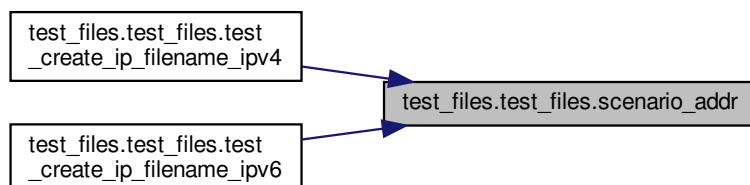
```

def test_files.test_files.scenario_addr (
    self,
    d1,
    d2,
    ipaddr )
  
```

Here is the call graph for this function:



Here is the caller graph for this function:



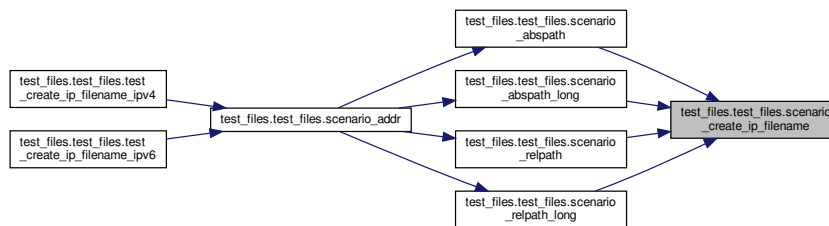
6.16.1.4 scenario_create_ip_filename()

```
def test_files.test_files.scenario_create_ip_filename (
    self,
    path_no_slash,
    d1,
    d2,
    ipaddr )
```

Here is the call graph for this function:



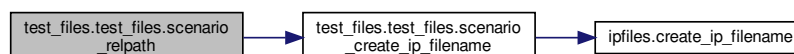
Here is the caller graph for this function:



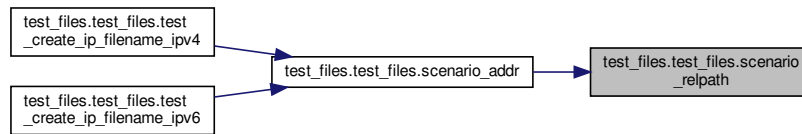
6.16.1.5 scenario_relpath()

```
def test_files.test_files.scenario_relpath (
    self,
    d1,
    d2,
    ipaddr )
```

Here is the call graph for this function:



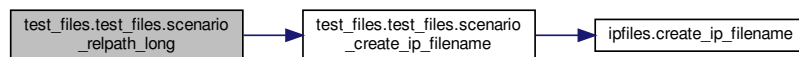
Here is the caller graph for this function:



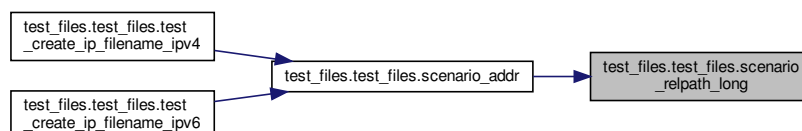
6.16.1.6 scenario_relpath_long()

```
def test_files.test_files.scenario_relpath_long (
    self,
    d1,
    d2,
    ipaddr )
```

Here is the call graph for this function:



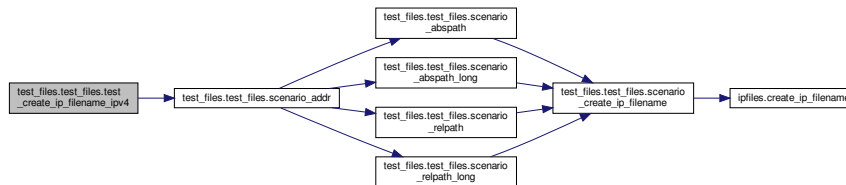
Here is the caller graph for this function:



6.16.1.7 test_create_ip_filename_ipv4()

```
def test_files.test_files.test_create_ip_filename_ipv4 (
    self,
    ipaddr )
```

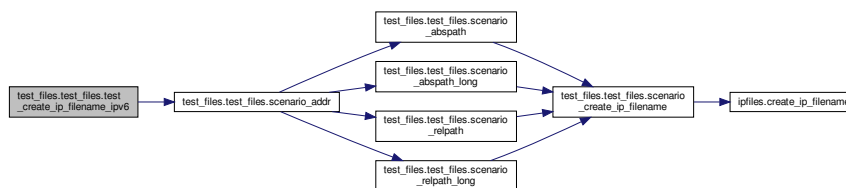
Here is the call graph for this function:



6.16.1.8 test_create_ip_filename_ipv6()

```
def test_files.test_files.test_create_ip_filename_ipv6 (
    self,
    * args )
```

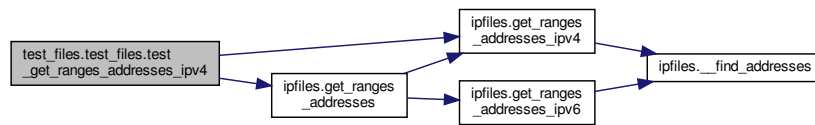
Here is the call graph for this function:



6.16.1.9 test_get_ranges_addresses_ipv4()

```
def test_files.test_files.test_get_ranges_addresses_ipv4 (
    self,
    network_addr,
    plen,
    expected )
```

Here is the call graph for this function:

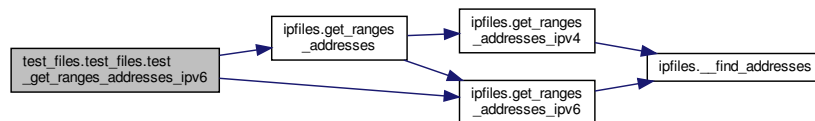


6.16.1.10 test_get_ranges_addresses_ipv6()

```

def test_files.test_files.test_get_ranges_addresses_ipv6 (
    self,
    network_addr,
    plen,
    expected )
  
```

Here is the call graph for this function:

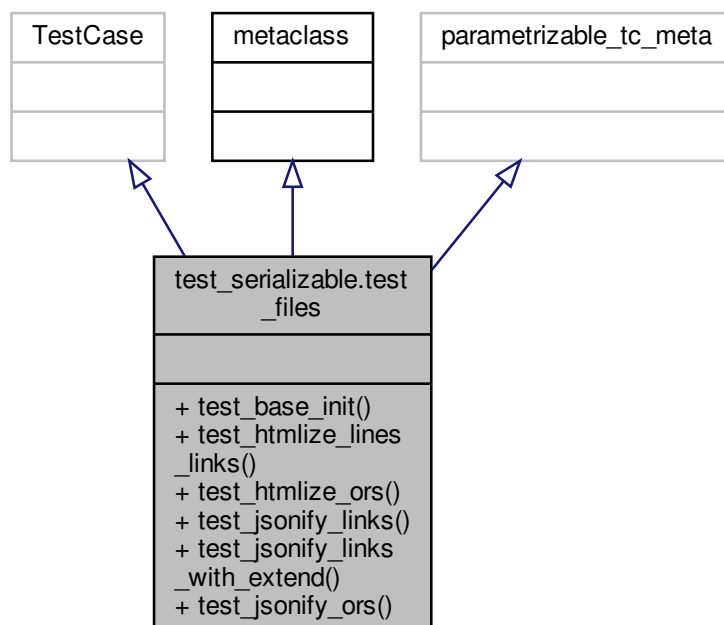


The documentation for this class was generated from the following file:

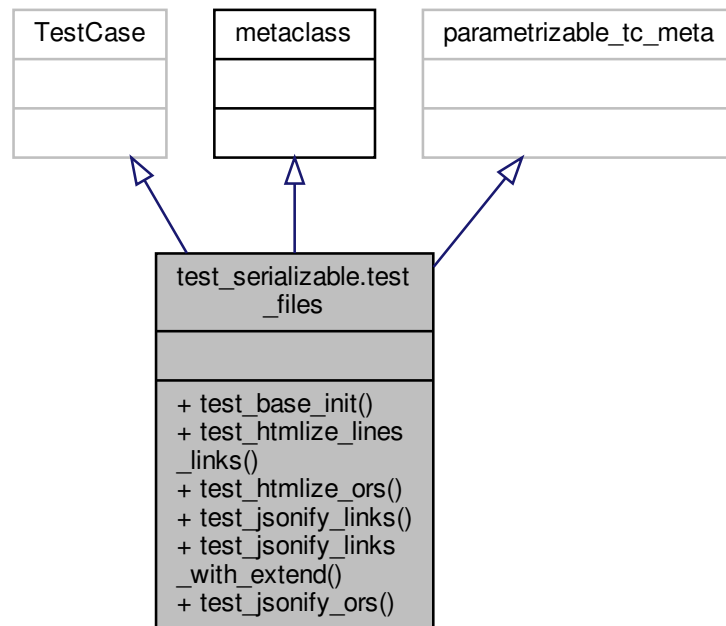
- [test_files.py](#)

6.17 test_serializable.test_files Class Reference

Inheritance diagram for test_serializable.test_files:



Collaboration diagram for test_serializable.test_files:



Public Member Functions

- def [test_base_init](#) (self)
- def [test_htmlize_links_links](#) (self)
- def [test_htmlize_ors](#) (self, path, digests)
- def [test_jsonify_links](#) (self)
- def [test_jsonify_links_with_extend](#) (self)
- def [test_jsonify_ors](#) (self, path, digests)

6.17.1 Member Function Documentation

6.17.1.1 test_base_init()

```
def test_serializable.test_files.test_base_init (
    self )
```

6.17.1.2 test_htmlize_lines_links()

```
def test_serializable.test_files.test_htmlize_lines_links (
    self )
```

6.17.1.3 test_htmlize_ors()

```
def test_serializable.test_files.test_htmlize_ors (
    self,
    path,
    digests )
```

6.17.1.4 test_jsonify_links()

```
def test_serializable.test_files.test_jsonify_links (
    self )
```

6.17.1.5 test_jsonify_links_with_extend()

```
def test_serializable.test_files.test_jsonify_links_with_extend (
    self )
```

6.17.1.6 test_jsonify_ors()

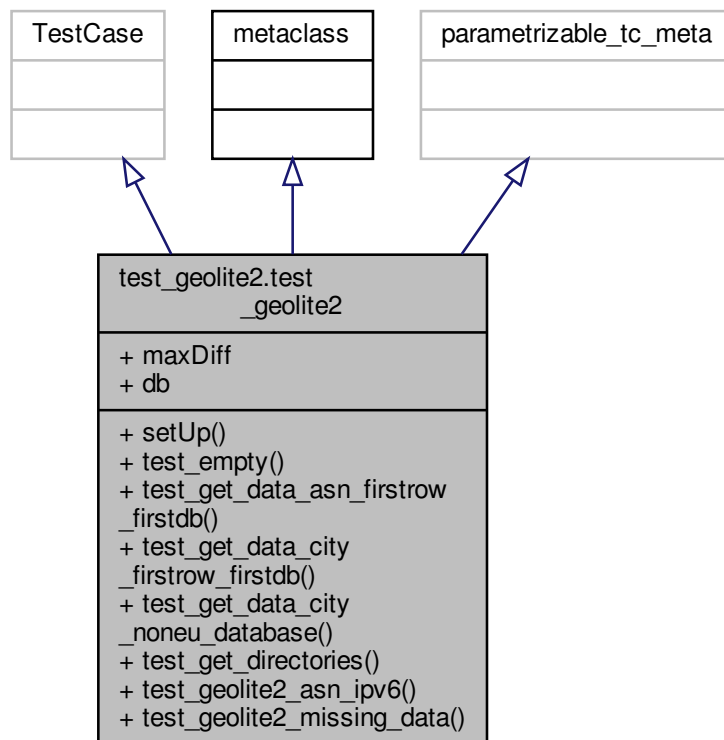
```
def test_serializable.test_files.test_jsonify_ors (
    self,
    path,
    digests )
```

The documentation for this class was generated from the following file:

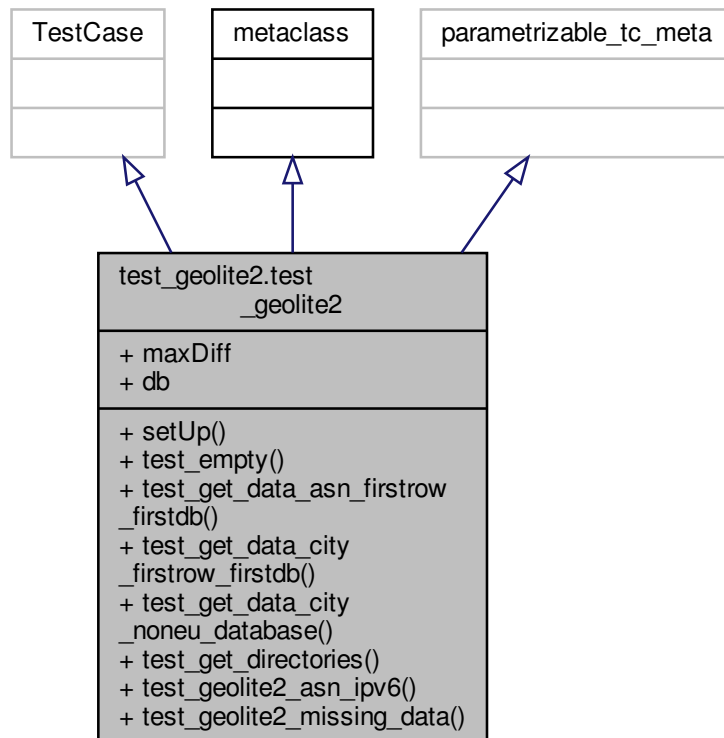
- [test_serializable.py](#)

6.18 test_geolite2.test_geolite2 Class Reference

Inheritance diagram for test_geolite2.test_geolite2:



Collaboration diagram for `test_geolite2.test_geolite2`:



Public Member Functions

- def `setUp` (self)
- def `test_empty` (self)
- def `test_get_data_asn_firstrow_firstdb` (self)
- def `test_get_data_city_firstrow_firstdb` (self)
- def `test_get_data_city_noneu_database` (self)
- def `test_get_directories` (self, geolite2_db, start, end, expected)
- def `test_geolite2_asn_ipv6` (self)
- def `test_geolite2_missing_data` (self)

Public Attributes

- `maxDiff`
- `db`

6.18.1 Member Function Documentation

6.18.1.1 setUp()

```
def test_geolite2.test_geolite2.setUp (
    self )
```

6.18.1.2 test_empty()

```
def test_geolite2.test_geolite2.test_empty (
    self )
```

6.18.1.3 test_geolite2_asn_ipv6()

```
def test_geolite2.test_geolite2.test_geolite2_asn_ipv6 (
    self )
```

6.18.1.4 test_geolite2_missing_data()

```
def test_geolite2.test_geolite2.test_geolite2_missing_data (
    self )
```

6.18.1.5 test_get_data_asn_firstrow_firstdb()

```
def test_geolite2.test_geolite2.test_get_data_asn_firstrow_firstdb (
    self )
```

6.18.1.6 test_get_data_city_firstrow_firstdb()

```
def test_geolite2.test_geolite2.test_get_data_city_firstrow_firstdb (
    self )
```

6.18.1.7 test_get_data_city_noneu_database()

```
def test_geolite2.test_geolite2.test_get_data_city_noneu_database (
    self )
```

6.18.1.8 test_get_directories()

```
def test_geolite2.test_geolite2.test_get_directories (
    self,
    geolite2_db,
    start,
    end,
    expected )
```

6.18.2 Member Data Documentation

6.18.2.1 db

test_geolite2.test_geolite2.db

6.18.2.2 maxDiff

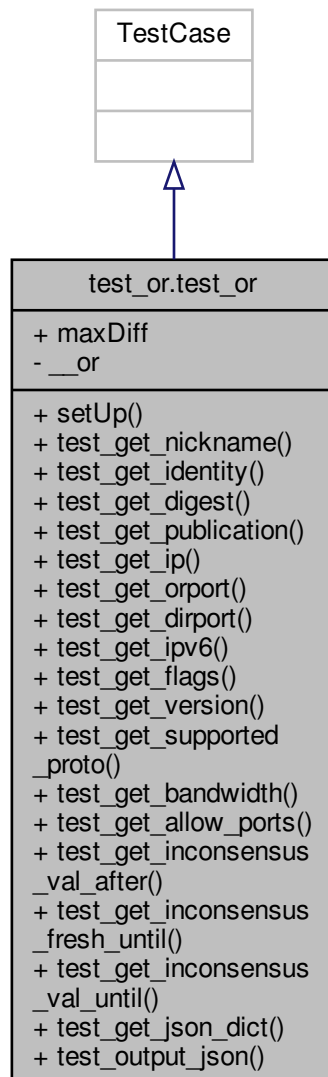
test_geolite2.test_geolite2.maxDiff

The documentation for this class was generated from the following file:

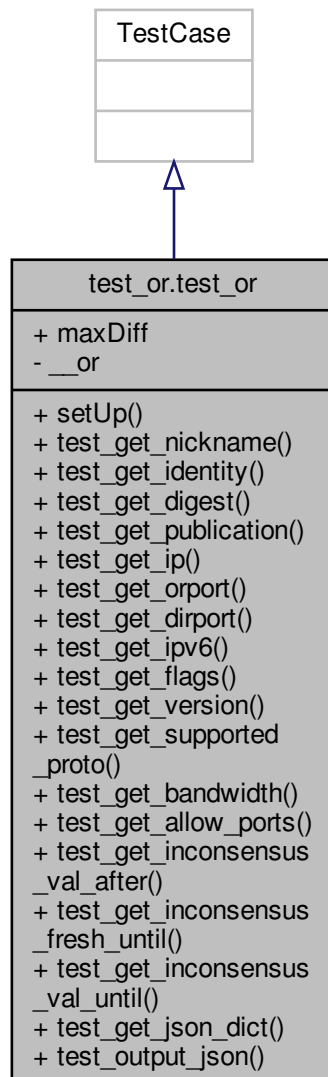
- [test_geolite2.py](#)

6.19 test_or.test_or Class Reference

Inheritance diagram for test_or.test_or:



Collaboration diagram for test_or.test_or:



Public Member Functions

- def [setUp](#) (self)
- def [test_get_nickname](#) (self)
- def [test_get_identity](#) (self)
- def [test_get_digest](#) (self)
- def [test_get_publication](#) (self)
- def [test_get_ip](#) (self)
- def [test_get_orport](#) (self)
- def [test_get_dirport](#) (self)
- def [test_get_ipv6](#) (self)
- def [test_get_flags](#) (self)

- def [test_get_version](#) (self)
- def [test_get_supported_proto](#) (self)
- def [test_get_bandwidth](#) (self)
- def [test_get_allow_ports](#) (self)
- def [test_get_inconsensus_val_after](#) (self)
- def [test_get_inconsensus_fresh_until](#) (self)
- def [test_get_inconsensus_val_until](#) (self)
- def [test_get_json_dict](#) (self)
- def [test_output_json](#) (self)

Public Attributes

- [maxDiff](#)

Private Attributes

- [__or](#)

6.19.1 Member Function Documentation

6.19.1.1 setUp()

```
def test_or.test_or.setUp (  
    self )
```

6.19.1.2 test_get_allow_ports()

```
def test_or.test_or.test_get_allow_ports (  
    self )
```

6.19.1.3 test_get_bandwidth()

```
def test_or.test_or.test_get_bandwidth (  
    self )
```

6.19.1.4 test_get_digest()

```
def test_or.test_or.test_get_digest (
    self )
```

6.19.1.5 test_get_dirport()

```
def test_or.test_or.test_get_dirport (
    self )
```

6.19.1.6 test_get_flags()

```
def test_or.test_or.test_get_flags (
    self )
```

6.19.1.7 test_get_identity()

```
def test_or.test_or.test_get_identity (
    self )
```

6.19.1.8 test_get_inconsensus_fresh_until()

```
def test_or.test_or.test_get_inconsensus_fresh_until (
    self )
```

6.19.1.9 test_get_inconsensus_val_after()

```
def test_or.test_or.test_get_inconsensus_val_after (
    self )
```

6.19.1.10 test_get_inconsensus_val_until()

```
def test_or.test_or.test_get_inconsensus_val_until (
    self )
```


6.19.1.11 test_get_ip()

```
def test_or.test_or.test_get_ip (
    self )
```

6.19.1.12 test_get_ipv6()

```
def test_or.test_or.test_get_ipv6 (
    self )
```

6.19.1.13 test_get_json_dict()

```
def test_or.test_or.test_get_json_dict (
    self )
```

6.19.1.14 test_get_nickname()

```
def test_or.test_or.test_get_nickname (
    self )
```

6.19.1.15 test_get_orport()

```
def test_or.test_or.test_get_orport (
    self )
```

6.19.1.16 test_get_publication()

```
def test_or.test_or.test_get_publication (
    self )
```

6.19.1.17 test_get_supported_proto()

```
def test_or.test_or.test_get_supported_proto (
    self )
```

6.19.1.18 test_get_version()

```
def test_or.test_or.test_get_version (
    self )
```

6.19.1.19 test_output_json()

```
def test_or.test_or.test_output_json (
    self )
```

6.19.2 Member Data Documentation

6.19.2.1 __or

```
test_or.test_or.__or [private]
```

6.19.2.2 maxDiff

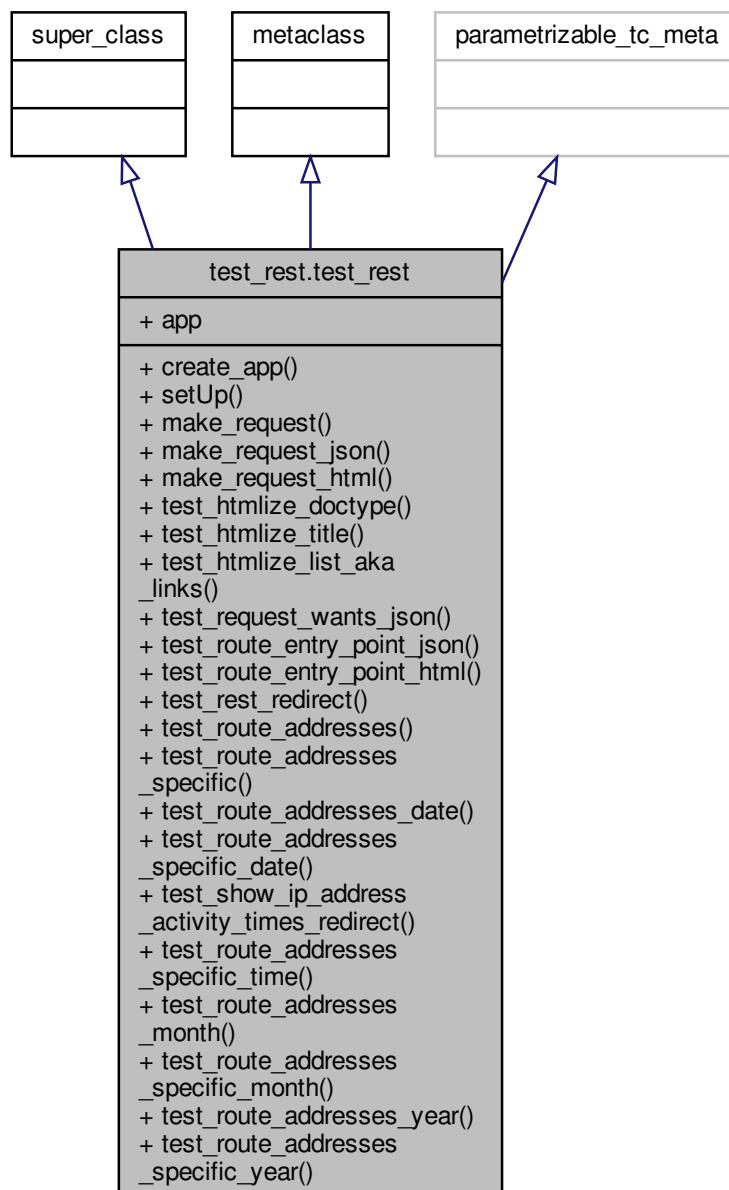
```
test_or.test_or.maxDiff
```

The documentation for this class was generated from the following file:

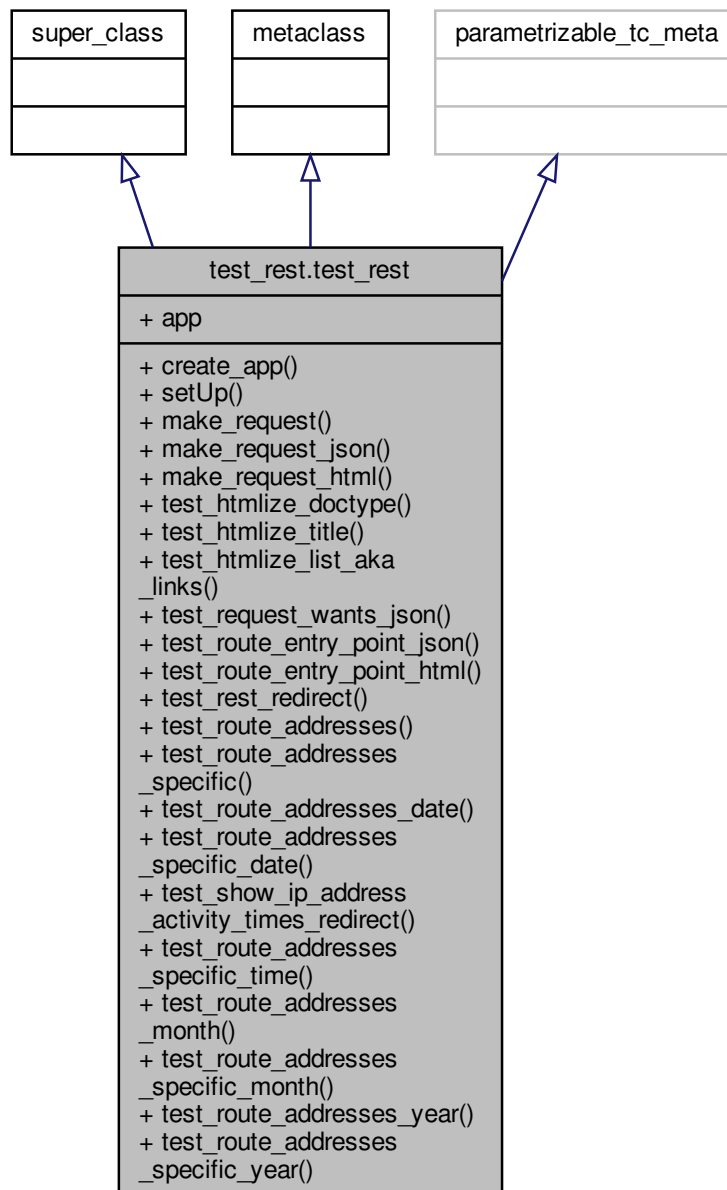
- [test_or.py](#)

6.20 test_rest.test_rest Class Reference

Inheritance diagram for test_rest.test_rest:



Collaboration diagram for `test_rest.test_rest`:



Public Member Functions

- def `create_app` (self)
- def `setUp` (self)
- def `make_request` (self, route, headers=[])
- def `make_request_json` (self, route, headers=[])
- def `make_request_html` (self, route, headers=[])
- def `test_htmlize_doctype` (self)
- def `test_htmlize_title` (self, func, expected_subtitle)

- def [test_htmlize_list_aka_links](#) (self)
- def [test_request_wants_json](#) (self, accept_content, expected)
- def [test_route_entry_point_json](#) (self)
- def [test_route_entry_point_html](#) (self)
- def [test_rest_redirect](#) (self)
- def [test_route_addresses](#) (self, endpoint, expected_range, expected_endpoint)
- def [test_route_addresses_specific](#) (self, addr, records, digest0)
- def [test_route_addresses_date](#) (self, addr, dates)
- def [test_route_addresses_specific_date](#) (self, addr, date, prev_date, next_date, items, digest)
- def [test_show_ip_address_activity_times_redirect](#) (self)
- def [test_route_addresses_specific_time](#) (self, addr, time, date, items, digest)
- def [test_route_addresses_month](#) (self, addr, months)
- def [test_route_addresses_specific_month](#) (self, addr, month, prev_month, next_month, items, digest)
- def [test_route_addresses_year](#) (self, addr, years)
- def [test_route_addresses_specific_year](#) (self, addr, year, prev_year, next_year, items, digest)

Public Attributes

- [app](#)

6.20.1 Member Function Documentation

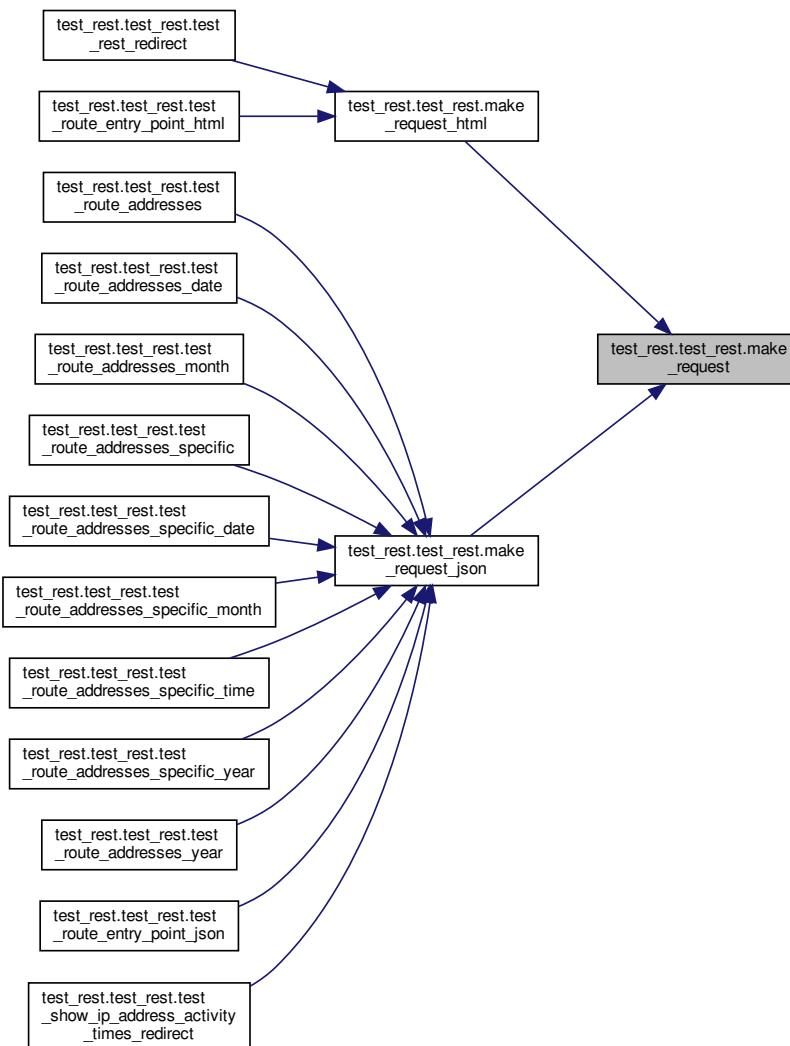
6.20.1.1 create_app()

```
def test_rest.test_rest.create_app (  
    self )
```

6.20.1.2 make_request()

```
def test_rest.test_rest.make_request (  
    self,  
    route,  
    headers = [] )
```

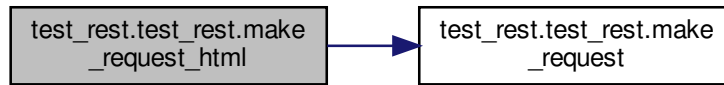
Here is the caller graph for this function:



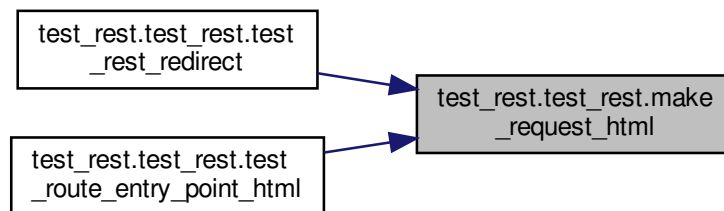
6.20.1.3 make_request_html()

```
def test_rest.test_rest.make_request_html (
    self,
    route,
    headers = [] )
```

Here is the call graph for this function:



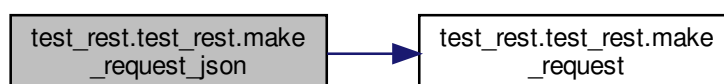
Here is the caller graph for this function:



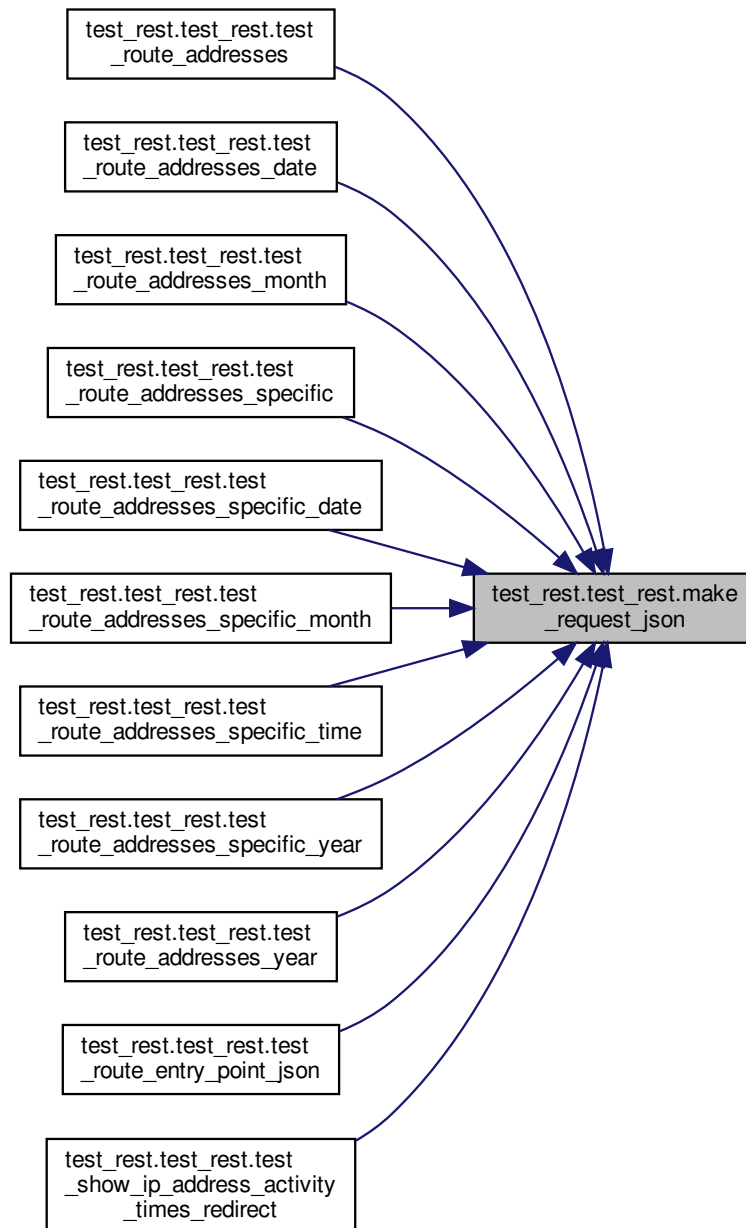
6.20.1.4 make_request_json()

```
def test_rest.test_rest.make_request_json (
    self,
    route,
    headers = [] )
```

Here is the call graph for this function:



Here is the caller graph for this function:



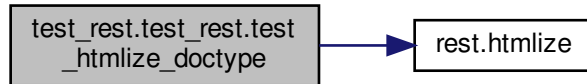
6.20.1.5 setUp()

```
def test_rest.test_rest.setUp (  
    self )
```


6.20.1.6 test_htmlize_doctype()

```
def test_rest.test_rest.test_htmlize_doctype (
    self )
```

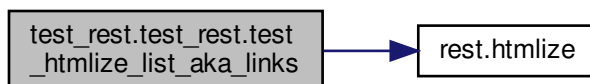
Here is the call graph for this function:



6.20.1.7 test_htmlize_list_aka_links()

```
def test_rest.test_rest.test_htmlize_list_aka_links (
    self )
```

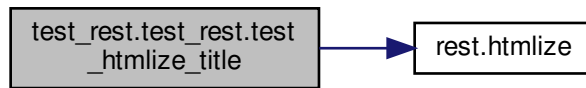
Here is the call graph for this function:



6.20.1.8 test_htmlize_title()

```
def test_rest.test_rest.test_htmlize_title (
    self,
    func,
    expected_subtitle )
```

Here is the call graph for this function:



6.20.1.9 test_request_wants_json()

```
def test_rest.test_rest.test_request_wants_json (
    self,
    accept_content,
    expected )
```

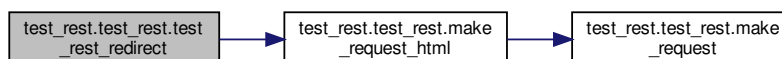
Here is the call graph for this function:



6.20.1.10 test_rest_redirect()

```
def test_rest.test_rest.test_rest_redirect (
    self )
```

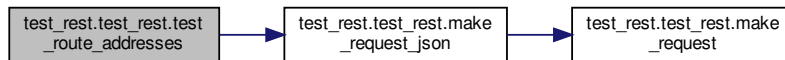
Here is the call graph for this function:



6.20.1.11 test_route_addresses()

```
def test_rest.test_rest.test_route_addresses (
    self,
    endpoint,
    expected_range,
    expected_endpoint )
```

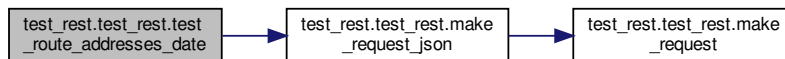
Here is the call graph for this function:



6.20.1.12 test_route_addresses_date()

```
def test_rest.test_rest.test_route_addresses_date (
    self,
    addr,
    dates )
```

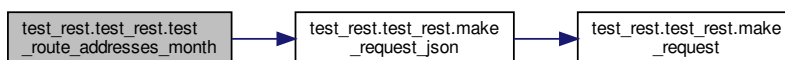
Here is the call graph for this function:



6.20.1.13 test_route_addresses_month()

```
def test_rest.test_rest.test_route_addresses_month (
    self,
    addr,
    months )
```

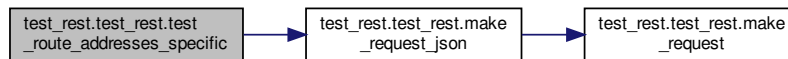
Here is the call graph for this function:



6.20.1.14 test_route_addresses_specific()

```
def test_rest.test_rest.test_route_addresses_specific (
    self,
    addr,
    records,
    digest0 )
```

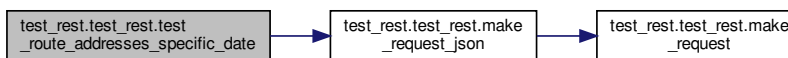
Here is the call graph for this function:



6.20.1.15 test_route_addresses_specific_date()

```
def test_rest.test_rest.test_route_addresses_specific_date (
    self,
    addr,
    date,
    prev_date,
    next_date,
    items,
    digest )
```

Here is the call graph for this function:

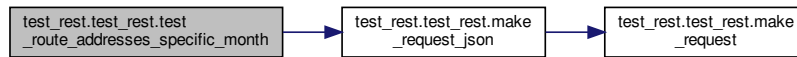


6.20.1.16 test_route_addresses_specific_month()

```
def test_rest.test_rest.test_route_addresses_specific_month (
    self,
    addr,
    month,
    prev_month,
    next_month,
```

```
    items,  
    digest )
```

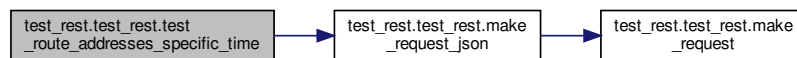
Here is the call graph for this function:



6.20.1.17 test_route_addresses_specific_time()

```
def test_rest.test_rest.test_route_addresses_specific_time (  
    self,  
    addr,  
    time,  
    date,  
    items,  
    digest )
```

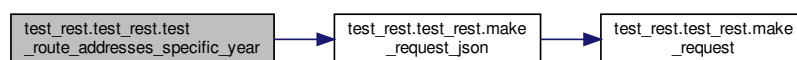
Here is the call graph for this function:



6.20.1.18 test_route_addresses_specific_year()

```
def test_rest.test_rest.test_route_addresses_specific_year (  
    self,  
    addr,  
    year,  
    prev_year,  
    next_year,  
    items,  
    digest )
```

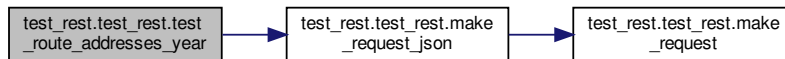
Here is the call graph for this function:



6.20.1.19 test_route_addresses_year()

```
def test_rest.test_rest.test_route_addresses_year (
    self,
    addr,
    years )
```

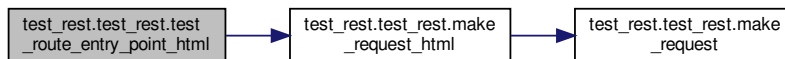
Here is the call graph for this function:



6.20.1.20 test_route_entry_point_html()

```
def test_rest.test_rest.test_route_entry_point_html (
    self )
```

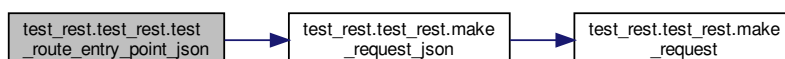
Here is the call graph for this function:



6.20.1.21 test_route_entry_point_json()

```
def test_rest.test_rest.test_route_entry_point_json (
    self )
```

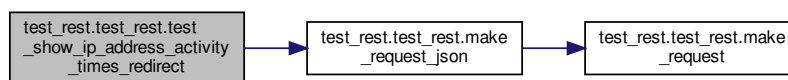
Here is the call graph for this function:



6.20.1.22 test_show_ip_address_activity_times_redirect()

```
def test_rest.test_rest.test_show_ip_address_activity_times_redirect (
    self )
```

Here is the call graph for this function:



6.20.2 Member Data Documentation

6.20.2.1 app

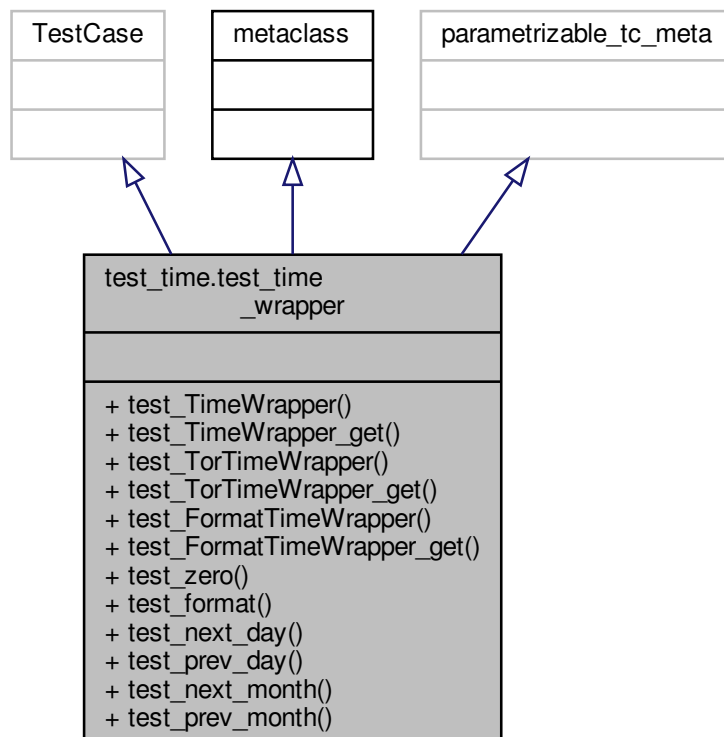
```
test_rest.test_rest.app
```

The documentation for this class was generated from the following file:

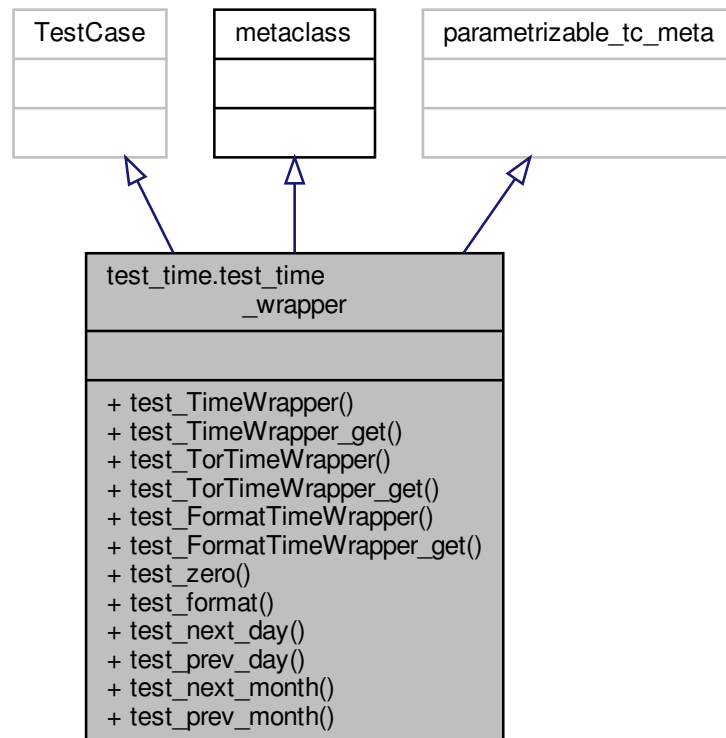
- [test_rest.py](#)

6.21 test_time.test_time_wrapper Class Reference

Inheritance diagram for test_time.test_time_wrapper:



Collaboration diagram for test_time.test_time_wrapper:



Public Member Functions

- def `test_TimeWrapper` (self, valuein, expected)
- def `test_TimeWrapper_get` (self)
- def `test_TorTimeWrapper` (self, datein, timein, expected)
- def `test_TorTimeWrapper_get` (self)
- def `test_FormatTimeWrapper` (self, timein, formatin, expected)
- def `test_FormatTimeWrapper_get` (self)
- def `test_zero` (self, cls, args)
- def `test_format` (self, timein, reqformat, expected)
- def `test_next_day` (self, timein, expected_date)
- def `test_prev_day` (self, timein, expected_date)
- def `test_next_month` (self, timein, expected_date)
- def `test_prev_month` (self, timein, expected_date)

6.21.1 Member Function Documentation

6.21.1.1 test_format()

```
def test_time.test_time_wrapper.test_format (
    self,
    timein,
    reqformat,
    expected )
```

6.21.1.2 test_FormatTimeWrapper()

```
def test_time.test_time_wrapper.test_FormatTimeWrapper (
    self,
    timein,
    formatin,
    expected )
```

6.21.1.3 test_FormatTimeWrapper_get()

```
def test_time.test_time_wrapper.test_FormatTimeWrapper_get (
    self )
```

6.21.1.4 test_next_day()

```
def test_time.test_time_wrapper.test_next_day (
    self,
    timein,
    expected_date )
```

6.21.1.5 test_next_month()

```
def test_time.test_time_wrapper.test_next_month (
    self,
    timein,
    expected_date )
```

6.21.1.6 test_prev_day()

```
def test_time.test_time_wrapper.test_prev_day (
    self,
    timein,
    expected_date )
```

6.21.1.7 test_prev_month()

```
def test_time.test_time_wrapper.test_prev_month (
    self,
    timein,
    expected_date )
```

6.21.1.8 test_TimeWrapper()

```
def test_time.test_time_wrapper.test_TimeWrapper (
    self,
    valuein,
    expected )
```

6.21.1.9 test_TimeWrapper_get()

```
def test_time.test_time_wrapper.test_TimeWrapper_get (
    self )
```

6.21.1.10 test_TorTimeWrapper()

```
def test_time.test_time_wrapper.test_TorTimeWrapper (
    self,
    datein,
    timein,
    expected )
```

6.21.1.11 test_TorTimeWrapper_get()

```
def test_time.test_time_wrapper.test_TorTimeWrapper_get (
    self )
```

6.21.1.12 test_zero()

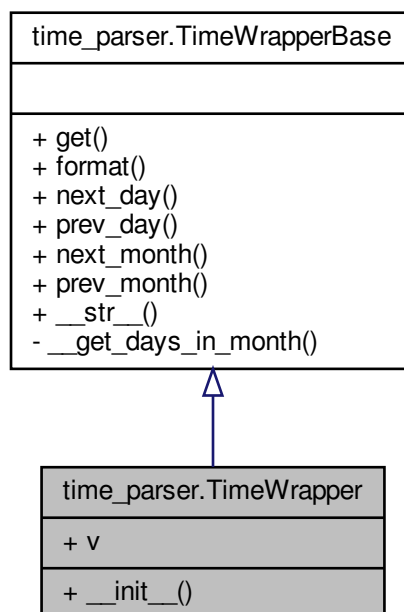
```
def test_time.test_time_wrapper.test_zero (
    self,
    cls,
    args )
```

The documentation for this class was generated from the following file:

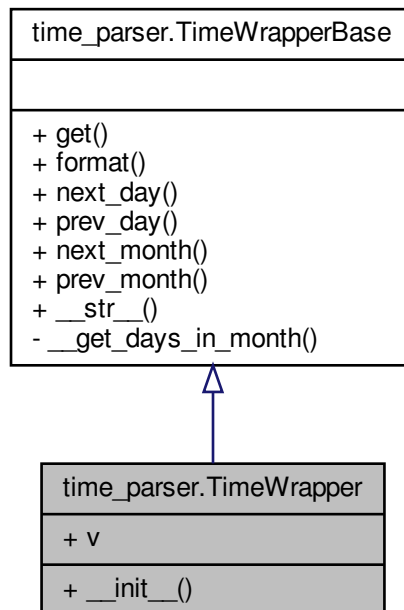
- [test_time.py](#)

6.22 time_parser.TimeWrapper Class Reference

Inheritance diagram for time_parser.TimeWrapper:



Collaboration diagram for time_parser.TimeWrapper:



Public Member Functions

- `def __init__(self, s)`

Public Attributes

- `v`

6.22.1 Detailed Description

Converts various string representation of time to Unix timestamp.

6.22.2 Constructor & Destructor Documentation

6.22.2.1 `__init__()`

```
def time_parser.TimeWrapper.__init__(
    self,
    s )
```

6.22.3 Member Data Documentation

6.22.3.1 v

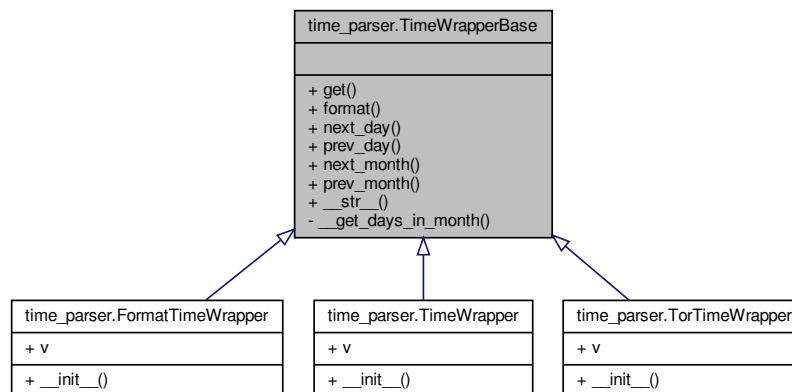
`time_parser.TimeWrapper.v`

The documentation for this class was generated from the following file:

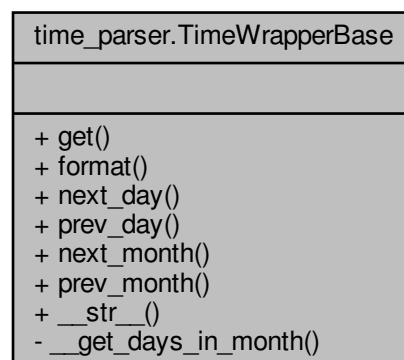
- [time_parser.py](#)

6.23 time_parser.TimeWrapperBase Class Reference

Inheritance diagram for `time_parser.TimeWrapperBase`:



Collaboration diagram for `time_parser.TimeWrapperBase`:



Public Member Functions

- def `get` (self)
- def `format` (self, fmt)
- def `next_day` (self)
- def `prev_day` (self)
- def `next_month` (self)
- def `prev_month` (self)
- def `__str__` (self)

Static Private Member Functions

- def `__get_days_in_month` (year, month)

6.23.1 Detailed Description

Base abstract class.

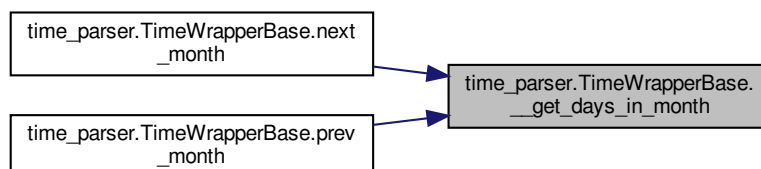
Expects that subclasses create `v` that stores time as Unix timestamp.

6.23.2 Member Function Documentation

6.23.2.1 `__get_days_in_month()`

```
def time_parser.TimeWrapperBase.__get_days_in_month (  
    year,  
    month ) [static], [private]
```

Here is the caller graph for this function:



6.23.2.2 `__str__()`

```
def time_parser.TimeWrapperBase.__str__ (
    self )
```

6.23.2.3 `format()`

```
def time_parser.TimeWrapperBase.format (
    self,
    fmt )
```

6.23.2.4 `get()`

```
def time_parser.TimeWrapperBase.get (
    self )
```

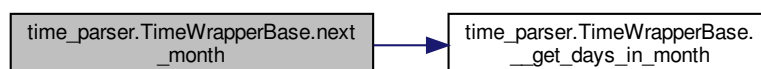
6.23.2.5 `next_day()`

```
def time_parser.TimeWrapperBase.next_day (
    self )
```

6.23.2.6 `next_month()`

```
def time_parser.TimeWrapperBase.next_month (
    self )
```

Here is the call graph for this function:



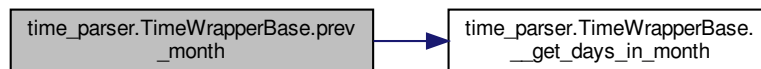
6.23.2.7 prev_day()

```
def time_parser.TimeWrapperBase.prev_day (
    self )
```

6.23.2.8 prev_month()

```
def time_parser.TimeWrapperBase.prev_month (
    self )
```

Here is the call graph for this function:

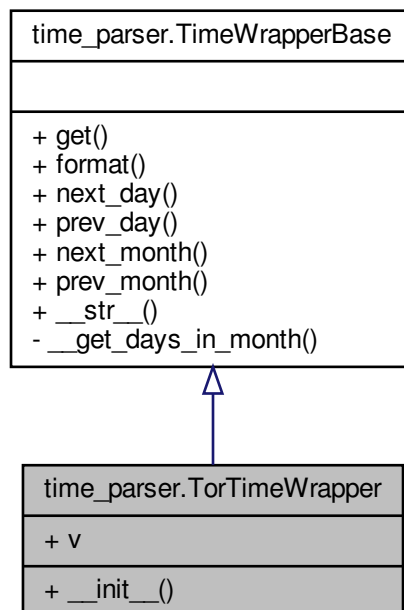


The documentation for this class was generated from the following file:

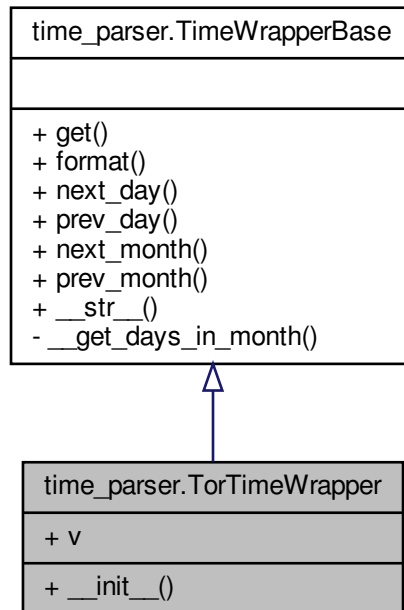
- [time_parser.py](#)

6.24 time_parser.TorTimeWrapper Class Reference

Inheritance diagram for `time_parser.TorTimeWrapper`:



Collaboration diagram for `time_parser.TorTimeWrapper`:



Public Member Functions

- `def __init__(self, datestr, timestr)`

Public Attributes

- `v`

6.24.1 Detailed Description

Converts Tor timestamp to Unix timestamp.

6.24.2 Constructor & Destructor Documentation

6.24.2.1 `__init__()`

```
def time_parser.TorTimeWrapper.__init__ (
    self,
    datestr,
    timestr )
```

Overloaded constructor

The `datestr` string is expected in format `%Y-%m-%d`.
The `timestr` string is expected in format `%H:%M:%S`.

Note that the super constructor is not called at all.

6.24.3 Member Data Documentation

6.24.3.1 `v`

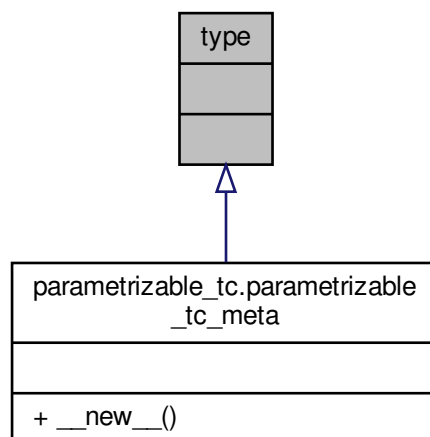
```
time_parser.TorTimeWrapper.v
```

The documentation for this class was generated from the following file:

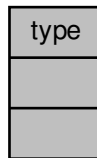
- [time_parser.py](#)

6.25 type Class Reference

Inheritance diagram for `type`:



Collaboration diagram for type:



The documentation for this class was generated from the following file:

- [parametrizable_tc.py](#)

Chapter 7

File Documentation

7.1 consensus_parser.py File Reference

Namespaces

- [consensus_parser](#)

Functions

- def [consensus_parser.create_snapshot_from_consensus_file](#) (fname)
- def [consensus_parser.write_preprocessed_from_consensus_file](#) (fname, preprocess_abspath, last)
- def [consensus_parser.preprocess_directory_consensus](#) (abspath, preprocess_abspath, last)
- def [consensus_parser.preprocess_path_consensus](#) (abspath, preprocess_abspath, last)
- def [consensus_parser.get_consensus_fullpath](#) (consensus_abspath, t)
- def [consensus_parser.download_consensus_if_missing](#) (consensus_abspath, t)
- def [consensus_parser.fixup_missing_consensus](#) (consensus_abspath, preprocess_abspath, t)
- def [consensus_parser.get_last_preprocessed_consensus](#) (preprocess_abspath)
- def [consensus_parser.process_path_consensus](#) (consensus_abspath, preprocess_abspath)
- def [consensus_parser.update_path_consensus](#) (consensus_abspath, preprocess_abspath)
- def [consensus_parser.write_preprocessed](#) (preprocess_abspath, consensus_snapshot)
- def [consensus_parser.parse_ors_from_file](#) (fname)
- def [consensus_parser.add_geolite_data_to_onion_routers](#) (ipaddr, ors, geolite_acc)
- def [consensus_parser.find_preprocessed_ip_address](#) (ipaddr, preprocessed_dir, geolite_acc=None)
- def [consensus_parser.find_preprocessed_ip_address_time_filter](#) (ipaddr, preprocessed_dir, time_search, date_prefix, geolite_acc)
- def [consensus_parser.output_preprocessed_ip_addresses_time_filter](#) (ipaddr, preprocessed_dir, time_search, date_prefix, geolite_acc, outfile)
- def [consensus_parser.merge_subsequent_ors](#) (ors)
- def [consensus_parser.get_ip_address_activity](#) (ipaddr, preprocessed_dir, time_format="%Y-%m-%d")
- def [consensus_parser.process_args](#) ()
- def [consensus_parser.main](#) (args, stdout)

Variables

- int [consensus_parser.WAIT_FOR_CONSENSUS](#) = 600
- def [consensus_parser.args](#) = process_args()

7.2 dns.py File Reference

Namespaces

- [dns](#)

Functions

- def [dns.ip_addr_to_domain](#) (ip_addr)

7.3 geolite2.py File Reference

Classes

- class [geolite2.geolite2_accessor](#)
- class [geolite2.geolite2_accessor.db_info](#)

Namespaces

- [geolite2](#)

Functions

- def [geolite2.process_args](#) ()

Variables

- def [geolite2.args](#) = process_args()
- [geolite2.gl](#) = geolite2_accessor(args.geolite2dir)
- [geolite2.t](#) = TimeWrapper(args.time).get()

7.4 ipfiles.py File Reference

Namespaces

- [ipfiles](#)

Functions

- def [ipfiles.create_ip_filename](#) (path, ipaddr, create_path=False)
- def [ipfiles.__find_addresses](#) (input_net, res_plen, p, prefix, remainder)
- def [ipfiles.get_ranges_addresses_ipv4](#) (path, network_addr, plen)
- def [ipfiles.get_ranges_addresses_ipv6](#) (path, network_addr, plen)
- def [ipfiles.get_ranges_addresses](#) (path, network_addr, plen)

7.5 onion_router.py File Reference

Classes

- class [onion_router.onion_router](#)

Namespaces

- [onion_router](#)

7.6 parametrizable_tc.py File Reference

Classes

- class [parametrizable_tc.parametrizable_tc_meta](#)

Namespaces

- [parametrizable_tc](#)

Functions

- def [parametrizable_tc.parametrizable_test](#) (paramList)

7.7 preprocess_geolite.py File Reference

Namespaces

- [preprocess_geolite](#)

Functions

- def [preprocess_geolite.write_network](#) (basedir, line, first_line_csv)
- def [preprocess_geolite.write_location](#) (basedir, line)
- def [preprocess_geolite.process_file](#) (fname, outdir, stdout, ipaddr_sep, plen_group_len, first_line_csv)
- def [preprocess_geolite.process_file4](#) (fname, outdir, stdout, first_line_csv)
- def [preprocess_geolite.process_file6](#) (fname, outdir, stdout, first_line_csv)
- def [preprocess_geolite.process_file_loc](#) (fname, outdir, stdout)
- def [preprocess_geolite.process_args](#) ()
- def [preprocess_geolite.main](#) (args, stdout)

Variables

- string `preprocess_geolite.ASN_FIRST_LINE_CSV` = "network,autonomous_system_number,autonomous↵_system_organization↵\n"
- string `preprocess_geolite.CITY_FIRST_LINE_CSV` = "network,geoname_id,registered_country_geoname↵_id,represented_country_geoname_id,is_anonymous_proxy,is_satellite_provider,postal_code,latitude,longitude,accuracy↵_radius↵\n"
- string `preprocess_geolite.LOCATION_FIRST_LINE_CSV` = "geoname_id,locale_code,continent_↵code,continent_name,country_iso_code,country_name,subdivision_1_iso_code,subdivision_1_name,subdivision↵_2_iso_code,subdivision_2_name,city_name,metro_code,time_zone↵\n"
- def `preprocess_geolite.args` = process_args()

7.8 profilerest.py File Reference

Namespaces

- `profilerest`

Variables

- `profilerest.wsgi_app`
- `profilerest.sort_by`
- `profilerest.restrictions`
- `profilerest.debug`

7.9 rest.py File Reference

Namespaces

- `rest`

Functions

- def `rest.request_wants_json` ()
- def `rest.htmlize` (result, related, func)
- def `rest.evaluate_accept_header_convert_json` (func)
- def `rest.entry_point` ()
- def `rest.show_known_address_ranges` ()
- def `rest.process_route_addresses` (address_detail)
- def `rest.show_all_ip_address_information` (address)
- def `rest.show_ip_address_activity_dates` (addr)
- def `rest.show_ip_address_activity_on_specific_date` (addr, date)
- def `rest.show_ip_address_activity_times` (addr)
- def `rest.show_ip_address_activity_at_specific_time` (addr, time)
- def `rest.show_ip_address_active_months` (addr)
- def `rest.show_ip_address_activity_during_specific_month` (addr, month)
- def `rest.show_ip_address_active_years` (addr)
- def `rest.show_ip_address_activity_during_specific_year` (addr, year)
- def `rest.enum_addresses_generic` (netaddr, plen)
- def `rest.show_known_address_subranges` (netaddr, plen)

Variables

- `rest.app` = `flask.Flask(__name__)`
- string `rest.PREPORCESSED_DIR` = `"/mnt/data/preprocessedip/"`
- string `rest.GEOLITE_DIR` = `"/mnt/data/preprocessed_geolite/"`

7.10 serializable.py File Reference

Classes

- class `serializable.serializable_list`
- class `serializable.serializable_ors`
- class `serializable.serializable_links`

Namespaces

- `serializable`

7.11 snapshot.py File Reference

Classes

- class `snapshot.Snapshot`

Namespaces

- `snapshot`

7.12 test.py File Reference

Namespaces

- `test`

Functions

- def `test.suite` ()
- def `test.test` (verbosity=2, failfast=False)

Variables

- `test.verbosity`

7.13 test_additional_info.py File Reference

Classes

- class [test_additional_info.test_additional_info](#)

Namespaces

- [test_additional_info](#)

Functions

- def [test_additional_info.suite](#) ()
- def [test_additional_info.test](#) (verbosity=2, failfast=False)

Variables

- [test_additional_info.verbosity](#)

7.14 test_cp.py File Reference

Classes

- class [test_cp.fake_cp_args](#)
- class [test_cp.test_consensus_parser](#)

Namespaces

- [test_cp](#)

Functions

- def [test_cp.suite](#) ()
- def [test_cp.test](#) (verbosity=2, failfast=False)

Variables

- string [test_cp.PCVESELY_JSON](#)
- string [test_cp.PREPROCESSED_DIR_CREATE](#) = "testfiles/preprocessedtestauto"
- string [test_cp.CONSENSUS_PATH](#) = "testfiles/descriptorstubs"
- [test_cp.verbosity](#)

7.15 test_files.py File Reference

Classes

- class [test_files.test_files](#)

Namespaces

- [test_files](#)

Functions

- def [test_files.suite](#) ()
- def [test_files.test](#) (verbosity=2, failfast=False)

Variables

- [test_files.verbosity](#)

7.16 test_geolite2.py File Reference

Classes

- class [test_geolite2.test_geolite2](#)

Namespaces

- [test_geolite2](#)

Functions

- def [test_geolite2.suite](#) ()
- def [test_geolite2.test](#) (verbosity=2, failfast=False)

Variables

- [test_geolite2.t20180605](#) = FormatTimeWrapper("20180605", "%Y%m%d").get()
- [test_geolite2.t20180619](#) = FormatTimeWrapper("20180619", "%Y%m%d").get()
- [test_geolite2.t20180626](#) = FormatTimeWrapper("20180626", "%Y%m%d").get()
- [test_geolite2.t20180703](#) = FormatTimeWrapper("20180703", "%Y%m%d").get()
- [test_geolite2.t20180709](#) = FormatTimeWrapper("20180709", "%Y%m%d").get()
- [test_geolite2.verbosity](#)

7.17 test_or.py File Reference

Classes

- class [test_or.test_or](#)

Namespaces

- [test_or](#)

Functions

- def [test_or.suite](#) ()
- def [test_or.test](#) (verbosity=2, failfast=False)

Variables

- [test_or.verbosity](#)

7.18 test_rest.py File Reference

Classes

- class [test_rest.fake](#)
- class [test_rest.test_rest](#)

Namespaces

- [test_rest](#)

Functions

- def [test_rest.suite](#) ()
- def [test_rest.test](#) (verbosity=2, failfast=False)

Variables

- bool [test_rest.flask_failed](#) = False
- [test_rest.super_class](#) = flask_testing.TestCase
- [test_rest.rest](#)
- string [test_rest.bs_required_text](#) = "BeautifulSoup not installed on this computer"
- bool [test_rest.bs_failed](#) = False
- [test_rest.verbosity](#)

7.19 test_serializable.py File Reference

Classes

- class [test_serializable.test_files](#)

Namespaces

- [test_serializable](#)

Functions

- def [test_serializable.suite](#) ()
- def [test_serializable.test](#) (verbosity=2, failfast=False)

Variables

- string [test_serializable.bs_required_text](#) = "BeautifulSoup not installed on this computer"
- bool [test_serializable.bs_failed](#) = False
- [test_serializable.verbosity](#)

7.20 test_time.py File Reference

Classes

- class [test_time.test_time_wrapper](#)

Namespaces

- [test_time](#)

Functions

- def [test_time.suite](#) ()
- def [test_time.test](#) (verbosity=2, failfast=False)

Variables

- [test_time.verbosity](#)

7.21 time_parser.py File Reference

Classes

- class [time_parser.TimeWrapperBase](#)
- class [time_parser.TimeWrapper](#)
- class [time_parser.TorTimeWrapper](#)
- class [time_parser.FormatTimeWrapper](#)

Namespaces

- [time_parser](#)

Variables

- dictionary [time_parser.months](#)
- int [time_parser.DAY_HAS_SECONDS](#) = 24*60*60

Index

- __addressbase
 - serializable.serializable_ors, 97
- __all_lines
 - onion_router.onion_router, 81
- __allow_ports
 - onion_router.onion_router, 81
- __bandwidth
 - onion_router.onion_router, 81
- __class__
 - serializable.serializable_list, 93
- __datestr
 - serializable.serializable_ors, 97
- __digest
 - onion_router.onion_router, 81
- __dirport
 - onion_router.onion_router, 81
- __find_addresses
 - ipfiles, 22
- __flags
 - onion_router.onion_router, 81
- __fresh_until
 - snapshot.Snapshot, 100
- __geolite2_dbs
 - geolite2.geolite2_accessor, 67
- __geolite2_dir
 - geolite2.geolite2_accessor, 67
- __get_blocks_matching_func
 - geolite2.geolite2_accessor, 63
- __get_days_in_month
 - time_parser.TimeWrapperBase, 157
- __get_directories
 - geolite2.geolite2_accessor, 64
- __get_geolite2_row
 - geolite2.geolite2_accessor, 64
- __get_preprocessed_filename_ipaddr
 - geolite2.geolite2_accessor, 65
- __get_preprocessed_filename_location
 - geolite2.geolite2_accessor, 65
- __getattr__
 - test_rest.fake, 57
- __htmlize_or
 - serializable.serializable_ors, 96
- __identity
 - onion_router.onion_router, 81
- __inconsensus_fresh_until
 - onion_router.onion_router, 82
- __inconsensus_fresh_until_unixtime
 - onion_router.onion_router, 82
- __inconsensus_val_after
 - onion_router.onion_router, 82
- __inconsensus_val_after_unixtime
 - onion_router.onion_router, 82
- __inconsensus_val_until
 - onion_router.onion_router, 82
- __inconsensus_val_until_unixtime
 - onion_router.onion_router, 82
- __init__
 - geolite2.geolite2_accessor, 63
 - geolite2.geolite2_accessor.db_info, 55
 - onion_router.onion_router, 71
 - serializable.serializable_links, 88
 - serializable.serializable_list, 91
 - serializable.serializable_ors, 96
 - snapshot.Snapshot, 99
 - test_cp.fake_cp_args, 58
 - time_parser.FormatTimeWrapper, 61
 - time_parser.TimeWrapper, 155
 - time_parser.TorTimeWrapper, 160
- __ip
 - onion_router.onion_router, 82
 - snapshot.Snapshot, 100
- __ipstr
 - serializable.serializable_ors, 97
- __ipv6
 - onion_router.onion_router, 82
- __maxmind
 - serializable.serializable_ors, 98
- __new__
 - parametrizable_tc.parametrizable_tc_meta, 85
- __nickname
 - onion_router.onion_router, 83
- __obj_lines
 - snapshot.Snapshot, 100
- __or
 - test_or.test_or, 136
- __orport
 - onion_router.onion_router, 83
- __publication
 - onion_router.onion_router, 83
- __reverse_name
 - onion_router.onion_router, 83
- __routers
 - snapshot.Snapshot, 101
- __search_func
 - geolite2.geolite2_accessor, 68
- __str__
 - time_parser.TimeWrapperBase, 157
- __supported_proto

- onion_router.onion_router, 83
- __timestr
 - serializable.serializable_ors, 98
- __unparsed_lines
 - snapshot.Snapshot, 101
- __valid_after
 - snapshot.Snapshot, 101
- __valid_until
 - snapshot.Snapshot, 101
- __version
 - onion_router.onion_router, 83
- _make_html_link
 - serializable.serializable_list, 91
- _wrapped
 - serializable.serializable_list, 93
- add_geolite_data
 - onion_router.onion_router, 71
- add_geolite_data_to_onion_routers
 - consensus_parser, 9
- add_network_status_consensus
 - onion_router.onion_router, 72
- additional_info
 - test_cp.fake_cp_args, 58
- app
 - rest, 37
 - test_rest.test_rest, 149
- append
 - serializable.serializable_list, 92
- append_geolite_data
 - onion_router.onion_router, 72
- append_lines
 - onion_router.onion_router, 73
- args
 - consensus_parser, 20
 - geolite2, 22
 - preprocess_geolite, 30
- ASN_FIRST_LINE_CSV
 - preprocess_geolite, 30
- bs_failed
 - test_rest, 49
 - test_serializable, 51
- bs_required_text
 - test_rest, 49
 - test_serializable, 51
- cache
 - geolite2.geolite2_accessor.db_info, 56
- check_pcvesely_merge
 - test_cp.test_consensus_parser, 107
- check_pcvesely_or1
 - test_cp.test_consensus_parser, 108
- check_pcvesely_or2
 - test_cp.test_consensus_parser, 108
- check_pcvesely_ors
 - test_cp.test_consensus_parser, 109
- check_preprocessed_consensus_descriptorstubs
 - test_cp.test_consensus_parser, 110
- CITY_FIRST_LINE_CSV
 - preprocess_geolite, 30
- consensus_parser, 9
 - add_geolite_data_to_onion_routers, 9
 - args, 20
 - create_snapshot_from_consensus_file, 10
 - download_consensus_if_missing, 10
 - find_preprocessed_ip_address, 11
 - find_preprocessed_ip_address_time_filter, 11
 - fixup_missing_consensus, 12
 - get_consensus_fullpath, 13
 - get_ip_address_activity, 13
 - get_last_preprocessed_consensus, 14
 - main, 14
 - merge_subsequent_ors, 15
 - output_preprocessed_ip_addresses_time_filter, 15
 - parse_ors_from_file, 16
 - preprocess_directory_consensus, 16
 - preprocess_path_consensus, 17
 - process_args, 17
 - process_path_consensus, 18
 - update_path_consensus, 18
 - WAIT_FOR_CONSENSUS, 20
 - write_preprocessed, 19
 - write_preprocessed_from_consensus_file, 19
- consensus_parser.py, 163
- CONSENSUS_PATH
 - test_cp, 43
- consensus_path
 - test_cp.fake_cp_args, 58
- create_app
 - test_rest.test_rest, 139
- create_ip_filename
 - ipfiles, 22
- create_ordered_dict_base
 - geolite2.geolite2_accessor, 65
- create_snapshot_from_consensus_file
 - consensus_parser, 10
- date_prefix
 - test_cp.fake_cp_args, 58
- DAY_HAS_SECONDS
 - time_parser, 53
- db
 - test_geolite2.test_geolite2, 130
- debug
 - profilerest, 31
- dirs
 - geolite2.geolite2_accessor.db_info, 56
- dns, 20
 - ip_addr_to_domain, 20
- dns.py, 164
- download_consensus_if_missing
 - consensus_parser, 10
- entry_point
 - rest, 32
- enum_addresses_generic
 - rest, 32

- evaluate_accept_header_convert_json
 - rest, [32](#)
- extend
 - serializable.serializable_list, [92](#)
- find_preprocessed_ip_address
 - consensus_parser, [11](#)
- find_preprocessed_ip_address_time_filter
 - consensus_parser, [11](#)
- fix_pcvesely_orc_for_merge
 - test_cp.test_consensus_parser, [110](#)
- fixup
 - test_cp.fake_cp_args, [58](#)
- fixup_missing_consensus
 - consensus_parser, [12](#)
- flask_failed
 - test_rest, [50](#)
- format
 - time_parser.TimeWrapperBase, [158](#)
- geolite2, [21](#)
 - args, [22](#)
 - gl, [22](#)
 - process_args, [21](#)
 - t, [22](#)
- geolite2.geolite2_accessor, [62](#)
 - __geolite2_db, [67](#)
 - __geolite2_dir, [67](#)
 - __get_blocks_matching_func, [63](#)
 - __get_directories, [64](#)
 - __get_geolite2_row, [64](#)
 - __get_preprocessed_filename_ipaddr, [65](#)
 - __get_preprocessed_filename_location, [65](#)
 - __init__, [63](#)
 - __search_func, [68](#)
 - create_ordered_dict_base, [65](#)
 - get_data, [65](#)
 - get_geolite2_as, [66](#)
 - get_geolite2_geolocation, [67](#)
- geolite2.geolite2_accessor.db_info, [55](#)
 - __init__, [55](#)
 - cache, [56](#)
 - dirs, [56](#)
 - last, [56](#)
- geolite2.py, [164](#)
- GEOLITE_DIR
 - rest, [37](#)
- geolite_dir
 - test_cp.fake_cp_args, [59](#)
- get
 - time_parser.TimeWrapperBase, [158](#)
- get_allow_ports
 - onion_router.onion_router, [73](#)
- get_bandwidth
 - onion_router.onion_router, [73](#)
- get_consensus_fullpath
 - consensus_parser, [13](#)
- get_data
 - geolite2.geolite2_accessor, [65](#)
 - get_digest
 - onion_router.onion_router, [73](#)
 - get_dirport
 - onion_router.onion_router, [73](#)
 - get_dns_reverse
 - onion_router.onion_router, [74](#)
 - get_flags
 - onion_router.onion_router, [74](#)
 - get_geolite2_as
 - geolite2.geolite2_accessor, [66](#)
 - get_geolite2_geolocation
 - geolite2.geolite2_accessor, [67](#)
 - get_identity
 - onion_router.onion_router, [74](#)
 - get_inconsensus_fresh_until
 - onion_router.onion_router, [74](#)
 - get_inconsensus_fresh_until_unixtime
 - onion_router.onion_router, [74](#)
 - get_inconsensus_val_after
 - onion_router.onion_router, [74](#)
 - get_inconsensus_val_after_unixtime
 - onion_router.onion_router, [75](#)
 - get_inconsensus_val_until
 - onion_router.onion_router, [75](#)
 - get_inconsensus_val_until_unixtime
 - onion_router.onion_router, [75](#)
 - get_ip
 - onion_router.onion_router, [76](#)
 - get_ip_address_activity
 - consensus_parser, [13](#)
 - get_ipaddrs
 - snapshot.Snapshot, [99](#)
 - get_ipv6
 - onion_router.onion_router, [76](#)
 - get_json_dict
 - onion_router.onion_router, [76](#)
 - get_last_preprocessed_consensus
 - consensus_parser, [14](#)
 - get_maxmind_asn
 - onion_router.onion_router, [76](#)
 - get_maxmind_geolocation
 - onion_router.onion_router, [76](#)
 - get_nickname
 - onion_router.onion_router, [76](#)
 - get_onion_router
 - snapshot.Snapshot, [99](#)
 - get_orport
 - onion_router.onion_router, [77](#)
 - get_publication
 - onion_router.onion_router, [77](#)
 - get_ranges_addresses
 - ipfiles, [23](#)
 - get_ranges_addresses_ipv4
 - ipfiles, [24](#)
 - get_ranges_addresses_ipv6
 - ipfiles, [24](#)
 - get_supported_proto
 - onion_router.onion_router, [77](#)

- get_valid_after
 - snapshot.Snapshot, 99
- get_version
 - onion_router.onion_router, 77
- gl
 - geolite2, 22
- htmlize
 - rest, 33
- htmlize_lines
 - serializable.serializable_links, 89
 - serializable.serializable_list, 92
 - serializable.serializable_ors, 96
- ip_addr_to_domain
 - dns, 20
- ipaddress
 - test_cp.fake_cp_args, 59
- ipfiles, 22
 - __find_addresses, 22
 - create_ip_filename, 22
 - get_ranges_addresses, 23
 - get_ranges_addresses_ipv4, 24
 - get_ranges_addresses_ipv6, 24
- ipfiles.py, 164
- jsonify
 - serializable.serializable_links, 89
 - serializable.serializable_list, 92
 - serializable.serializable_ors, 97
- last
 - geolite2.geolite2_accessor.db_info, 56
- LOCATION_FIRST_LINE_CSV
 - preprocess_geolite, 30
- main
 - consensus_parser, 14
 - preprocess_geolite, 26
- make_request
 - test_rest.test_rest, 139
- make_request_html
 - test_rest.test_rest, 140
- make_request_json
 - test_rest.test_rest, 141
- maxDiff
 - test_geolite2.test_geolite2, 130
 - test_or.test_or, 136
- merge
 - onion_router.onion_router, 77
- merge_attrs
 - onion_router.onion_router, 83
- merge_subsequent_ors
 - consensus_parser, 15
- metaclass, 68
- months
 - time_parser, 54
- next_day
 - time_parser.TimeWrapperBase, 158
- next_month
 - time_parser.TimeWrapperBase, 158
- onion_router, 25
- onion_router.onion_router, 69
 - __all_lines, 81
 - __allow_ports, 81
 - __bandwidth, 81
 - __digest, 81
 - __dirport, 81
 - __flags, 81
 - __identity, 81
 - __inconsensus_fresh_until, 82
 - __inconsensus_fresh_until_unixtime, 82
 - __inconsensus_val_after, 82
 - __inconsensus_val_after_unixtime, 82
 - __inconsensus_val_until, 82
 - __inconsensus_val_until_unixtime, 82
 - __init__, 71
 - __ip, 82
 - __ipv6, 82
 - __nickname, 83
 - __orport, 83
 - __publication, 83
 - __reverse_name, 83
 - __supported_proto, 83
 - __version, 83
 - add_geolite_data, 71
 - add_network_status_consensus, 72
 - append_geolite_data, 72
 - append_lines, 73
 - get_allow_ports, 73
 - get_bandwidth, 73
 - get_digest, 73
 - get_dirport, 73
 - get_dns_reverse, 74
 - get_flags, 74
 - get_identity, 74
 - get_inconsensus_fresh_until, 74
 - get_inconsensus_fresh_until_unixtime, 74
 - get_inconsensus_val_after, 74
 - get_inconsensus_val_after_unixtime, 75
 - get_inconsensus_val_until, 75
 - get_inconsensus_val_until_unixtime, 75
 - get_ip, 76
 - get_ipv6, 76
 - get_json_dict, 76
 - get_maxmind_asn, 76
 - get_maxmind_geolocation, 76
 - get_nickname, 76
 - get_orport, 77
 - get_publication, 77
 - get_supported_proto, 77
 - get_version, 77
 - merge, 77
 - merge_attrs, 83
 - output_json, 78
 - parse_a, 78
 - parse_id, 79

- parse_m, 79
- parse_p, 79
- parse_pr, 79
- parse_r, 79
- parse_s, 79
- parse_v, 80
- parse_w, 80
- parse_x_dns_reverse, 80
- parse_x_inconsensus_fresh_until, 80
- parse_x_inconsensus_valid_after, 80
- parse_x_inconsensus_valid_until, 80
- request_additional, 84
- onion_router.py, 165
- output_json
 - onion_router.onion_router, 78
- output_preprocessed_ip_addresses_time_filter
 - consensus_parser, 15
- parametrizable_tc, 25
 - parametrizable_test, 25
- parametrizable_tc.parametrizable_tc_meta, 84
 - __new__, 85
- parametrizable_tc.py, 165
- parametrizable_test
 - parametrizable_tc, 25
- parse_a
 - onion_router.onion_router, 78
- parse_fresh_until
 - snapshot.Snapshot, 100
- parse_id
 - onion_router.onion_router, 79
- parse_m
 - onion_router.onion_router, 79
- parse_ors_from_file
 - consensus_parser, 16
- parse_p
 - onion_router.onion_router, 79
- parse_pr
 - onion_router.onion_router, 79
- parse_r
 - onion_router.onion_router, 79
- parse_s
 - onion_router.onion_router, 79
- parse_v
 - onion_router.onion_router, 80
- parse_valid_after
 - snapshot.Snapshot, 100
- parse_valid_until
 - snapshot.Snapshot, 100
- parse_w
 - onion_router.onion_router, 80
- parse_x_dns_reverse
 - onion_router.onion_router, 80
- parse_x_inconsensus_fresh_until
 - onion_router.onion_router, 80
- parse_x_inconsensus_valid_after
 - onion_router.onion_router, 80
- parse_x_inconsensus_valid_until
 - onion_router.onion_router, 80
- PCVESELY_JSON
 - test_cp, 43
- PREPROCESSED_DIR
 - rest, 37
- preprocess_directory_consensus
 - consensus_parser, 16
- preprocess_geolite, 25
 - args, 30
 - ASN_FIRST_LINE_CSV, 30
 - CITY_FIRST_LINE_CSV, 30
 - LOCATION_FIRST_LINE_CSV, 30
 - main, 26
 - process_args, 26
 - process_file, 26
 - process_file4, 27
 - process_file6, 28
 - process_file_loc, 28
 - write_location, 29
 - write_network, 29
- preprocess_geolite.py, 165
- preprocess_path_consensus
 - consensus_parser, 17
- PREPROCESSED_DIR_CREATE
 - test_cp, 43
- preprocessed_input
 - test_cp.fake_cp_args, 59
- prev_day
 - time_parser.TimeWrapperBase, 158
- prev_month
 - time_parser.TimeWrapperBase, 159
- process_args
 - consensus_parser, 17
 - geolite2, 21
 - preprocess_geolite, 26
- process_file
 - preprocess_geolite, 26
- process_file4
 - preprocess_geolite, 27
- process_file6
 - preprocess_geolite, 28
- process_file_loc
 - preprocess_geolite, 28
- process_path_consensus
 - consensus_parser, 18
- process_route_addresses
 - rest, 33
- profilerest, 31
 - debug, 31
 - restrictions, 31
 - sort_by, 31
 - wsgi_app, 31
- profilerest.py, 166
- request_additional
 - onion_router.onion_router, 84
- request_wants_json
 - rest, 34
- rest, 31
 - app, 37

- entry_point, 32
- enum_addresses_generic, 32
- evaluate_accept_header_convert_json, 32
- GEOLITE_DIR, 37
- htmlize, 33
- PREPORCESSED_DIR, 37
- process_route_addresses, 33
- request_wants_json, 34
- show_all_ip_address_information, 34
- show_ip_address_active_months, 35
- show_ip_address_active_years, 35
- show_ip_address_activity_at_specific_time, 35
- show_ip_address_activity_dates, 35
- show_ip_address_activity_during_specific_month, 35
- show_ip_address_activity_during_specific_year, 36
- show_ip_address_activity_on_specific_date, 36
- show_ip_address_activity_times, 36
- show_known_address_ranges, 36
- show_known_address_subranges, 36
- test_rest, 50
- rest.py, 166
- restrictions
 - profilerest, 31
- scenario_abspath
 - test_files.test_files, 117
- scenario_abspath_long
 - test_files.test_files, 118
- scenario_addr
 - test_files.test_files, 119
- scenario_create_ip_filename
 - test_files.test_files, 119
- scenario_relpath
 - test_files.test_files, 120
- scenario_relpath_long
 - test_files.test_files, 121
- serializable, 38
- serializable.py, 167
- serializable.serializable_links, 87
 - __init__, 88
 - htmlize_lines, 89
 - jsonify, 89
- serializable.serializable_list, 90
 - __class__, 93
 - __init__, 91
 - _make_html_link, 91
 - _wrapped, 93
 - append, 92
 - extend, 92
 - htmlize_lines, 92
 - jsonify, 92
- serializable.serializable_ors, 94
 - __addressbase, 97
 - __datestr, 97
 - __htmlize_or, 96
 - __init__, 96
 - __ipstr, 97
 - __maxmind, 98
 - __timestr, 98
 - htmlize_lines, 96
 - jsonify, 97
- setUp
 - test_geolite2.test_geolite2, 128
 - test_or.test_or, 133
 - test_rest.test_rest, 142
- show_all_ip_address_information
 - rest, 34
- show_ip_address_active_months
 - rest, 35
- show_ip_address_active_years
 - rest, 35
- show_ip_address_activity_at_specific_time
 - rest, 35
- show_ip_address_activity_dates
 - rest, 35
- show_ip_address_activity_during_specific_month
 - rest, 35
- show_ip_address_activity_during_specific_year
 - rest, 36
- show_ip_address_activity_on_specific_date
 - rest, 36
- show_ip_address_activity_times
 - rest, 36
- show_known_address_ranges
 - rest, 36
- show_known_address_subranges
 - rest, 36
- snapshot, 38
- snapshot.py, 167
- snapshot.Snapshot, 98
 - __fresh_until, 100
 - __init__, 99
 - __ip, 100
 - __obj_lines, 100
 - __routers, 101
 - __unparsed_lines, 101
 - __valid_after, 101
 - __valid_until, 101
 - get_ipaddrs, 99
 - get_onion_router, 99
 - get_valid_after, 99
 - parse_fresh_until, 100
 - parse_valid_after, 100
 - parse_valid_until, 100
- sort_by
 - profilerest, 31
- suite
 - test, 38
 - test_additional_info, 41
 - test_cp, 42
 - test_files, 44
 - test_geolite2, 45
 - test_or, 47
 - test_rest, 49
 - test_serializable, 51

- test_time, 52
- super_class, 102
- test_rest, 50
- t
 - geolite2, 22
- t20180605
 - test_geolite2, 46
- t20180619
 - test_geolite2, 46
- t20180626
 - test_geolite2, 46
- t20180703
 - test_geolite2, 46
- t20180709
 - test_geolite2, 46
- test, 38
 - suite, 38
 - test, 39
 - test_additional_info, 41
 - test_cp, 42
 - test_files, 44
 - test_geolite2, 45
 - test_or, 47
 - test_rest, 49
 - test_serializable, 51
 - test_time, 52
 - verbosity, 40
- test.py, 167
- test_additional_info, 41
 - suite, 41
 - test, 41
 - verbosity, 42
- test_additional_info.py, 168
- test_additional_info.test_additional_info, 103
 - test_dns, 104
- test_base_init
 - test_serializable.test_files, 125
- test_cp, 42
 - CONSENSUS_PATH, 43
 - PCVESELY_JSON, 43
 - PREPROCESSED_DIR_CREATE, 43
 - suite, 42
 - test, 42
 - verbosity, 43
- test_cp.fake_cp_args, 57
 - __init__, 58
 - additional_info, 58
 - consensus_path, 58
 - date_prefix, 58
 - fixup, 58
 - geolite_dir, 59
 - ipaddress, 59
 - preprocessed_input, 59
 - time, 59
 - update, 59
 - update_keeprunning, 59
 - write_preprocessed, 59
- test_cp.py, 168
- test_cp.test_consensus_parser, 105
 - check_pcvesely_merge, 107
 - check_pcvesely_or1, 108
 - check_pcvesely_or2, 108
 - check_pcvesely_ors, 109
 - check_preprocessed_consensus_descriptorstubs, 110
 - fix_pcvesely_ors_for_merge, 110
 - test_download_consensus_if_missing, 111
 - test_find_preprocessed_ip_address, 111
 - test_find_preprocessed_ip_address_time_filter_dateprefix, 111
 - test_find_preprocessed_ip_address_time_filter_None, 112
 - test_find_preprocessed_ip_address_time_filter_time, 112
 - test_fixup, 112
 - test_get_consensus_fullpath, 112
 - test_get_ip_address_activity, 113
 - test_get_last_preprocessed_consensus, 113
 - test_main_create, 113
 - test_main_findip_address, 113
 - test_main_update, 113
 - test_merge_subsequent_ors, 114
 - test_merge_subsequent_ors_dns1, 114
 - test_merge_subsequent_ors_dns2, 114
 - test_merge_subsequent_ors_dns_both, 115
 - test_merge_subsequent_ors_failed, 115
 - test_parse_ors_from_file_2or, 115
- test_create_ip_filename_ipv4
 - test_files.test_files, 121
- test_create_ip_filename_ipv6
 - test_files.test_files, 122
- test_dns
 - test_additional_info.test_additional_info, 104
- test_download_consensus_if_missing
 - test_cp.test_consensus_parser, 111
- test_empty
 - test_geolite2.test_geolite2, 129
- test_files, 44
 - suite, 44
 - test, 44
 - verbosity, 45
- test_files.py, 169
- test_files.test_files, 116
 - scenario_abspath, 117
 - scenario_abspath_long, 118
 - scenario_addr, 119
 - scenario_create_ip_filename, 119
 - scenario_relpath, 120
 - scenario_relpath_long, 121
 - test_create_ip_filename_ipv4, 121
 - test_create_ip_filename_ipv6, 122
 - test_get_ranges_addresses_ipv4, 122
 - test_get_ranges_addresses_ipv6, 123
- test_find_preprocessed_ip_address
 - test_cp.test_consensus_parser, 111
- test_find_preprocessed_ip_address_time_filter_dateprefix

- test_cp.test_consensus_parser, 111
- test_find_preprocessed_ip_address_time_filter_None
 - test_cp.test_consensus_parser, 112
- test_find_preprocessed_ip_address_time_filter_time
 - test_cp.test_consensus_parser, 112
- test_fixup
 - test_cp.test_consensus_parser, 112
- test_format
 - test_time.test_time_wrapper, 151
- test_FormatTimeWrapper
 - test_time.test_time_wrapper, 152
- test_FormatTimeWrapper_get
 - test_time.test_time_wrapper, 152
- test_geolite2, 45
 - suite, 45
 - t20180605, 46
 - t20180619, 46
 - t20180626, 46
 - t20180703, 46
 - t20180709, 46
 - test, 45
 - verbosity, 47
- test_geolite2.py, 169
- test_geolite2.test_geolite2, 127
 - db, 130
 - maxDiff, 130
 - setUp, 128
 - test_empty, 129
 - test_geolite2_asn_ipv6, 129
 - test_geolite2_missing_data, 129
 - test_get_data_asn_firstrow_firstdb, 129
 - test_get_data_city_firstrow_firstdb, 129
 - test_get_data_city_noneu_database, 129
 - test_get_directories, 129
- test_geolite2_asn_ipv6
 - test_geolite2.test_geolite2, 129
- test_geolite2_missing_data
 - test_geolite2.test_geolite2, 129
- test_get_allow_ports
 - test_or.test_or, 133
- test_get_bandwidth
 - test_or.test_or, 133
- test_get_consensus_fullpath
 - test_cp.test_consensus_parser, 112
- test_get_data_asn_firstrow_firstdb
 - test_geolite2.test_geolite2, 129
- test_get_data_city_firstrow_firstdb
 - test_geolite2.test_geolite2, 129
- test_get_data_city_noneu_database
 - test_geolite2.test_geolite2, 129
- test_get_digest
 - test_or.test_or, 133
- test_get_directories
 - test_geolite2.test_geolite2, 129
- test_get_dirport
 - test_or.test_or, 134
- test_get_flags
 - test_or.test_or, 134
- test_get_identity
 - test_or.test_or, 134
- test_get_inconsensus_fresh_until
 - test_or.test_or, 134
- test_get_inconsensus_val_after
 - test_or.test_or, 134
- test_get_inconsensus_val_until
 - test_or.test_or, 134
- test_get_ip
 - test_or.test_or, 134
- test_get_ip_address_activity
 - test_cp.test_consensus_parser, 113
- test_get_ipv6
 - test_or.test_or, 135
- test_get_json_dict
 - test_or.test_or, 135
- test_get_last_preprocessed_consensus
 - test_cp.test_consensus_parser, 113
- test_get_nickname
 - test_or.test_or, 135
- test_get_orport
 - test_or.test_or, 135
- test_get_publication
 - test_or.test_or, 135
- test_get_ranges_addresses_ipv4
 - test_files.test_files, 122
- test_get_ranges_addresses_ipv6
 - test_files.test_files, 123
- test_get_supported_proto
 - test_or.test_or, 135
- test_get_version
 - test_or.test_or, 135
- test_htmlize_doctype
 - test_rest.test_rest, 142
- test_htmlize_lines_links
 - test_serializable.test_files, 125
- test_htmlize_list_aka_links
 - test_rest.test_rest, 143
- test_htmlize_ors
 - test_serializable.test_files, 126
- test_htmlize_title
 - test_rest.test_rest, 143
- test_jsonify_links
 - test_serializable.test_files, 126
- test_jsonify_links_with_extend
 - test_serializable.test_files, 126
- test_jsonify_ors
 - test_serializable.test_files, 126
- test_main_create
 - test_cp.test_consensus_parser, 113
- test_main_findip_address
 - test_cp.test_consensus_parser, 113
- test_main_update
 - test_cp.test_consensus_parser, 113
- test_merge_subsequent_ors
 - test_cp.test_consensus_parser, 114
- test_merge_subsequent_ors_dns1
 - test_cp.test_consensus_parser, 114

- test_merge_subsequent_ors_dns2
 - test_cp.test_consensus_parser, 114
- test_merge_subsequent_ors_dns_both
 - test_cp.test_consensus_parser, 115
- test_merge_subsequent_ors_failed
 - test_cp.test_consensus_parser, 115
- test_next_day
 - test_time.test_time_wrapper, 152
- test_next_month
 - test_time.test_time_wrapper, 152
- test_or, 47
 - suite, 47
 - test, 47
 - verbosity, 48
- test_or.py, 170
- test_or.test_or, 131
 - __or, 136
 - maxDiff, 136
 - setUp, 133
 - test_get_allow_ports, 133
 - test_get_bandwidth, 133
 - test_get_digest, 133
 - test_get_dirport, 134
 - test_get_flags, 134
 - test_get_identity, 134
 - test_get_inconsensus_fresh_until, 134
 - test_get_inconsensus_val_after, 134
 - test_get_inconsensus_val_until, 134
 - test_get_ip, 134
 - test_get_ipv6, 135
 - test_get_json_dict, 135
 - test_get_nickname, 135
 - test_get_orport, 135
 - test_get_publication, 135
 - test_get_supported_proto, 135
 - test_get_version, 135
 - test_output_json, 136
- test_output_json
 - test_or.test_or, 136
- test_parse_ors_from_file_2or
 - test_cp.test_consensus_parser, 115
- test_prev_day
 - test_time.test_time_wrapper, 152
- test_prev_month
 - test_time.test_time_wrapper, 152
- test_request_wants_json
 - test_rest.test_rest, 144
- test_rest, 48
 - bs_failed, 49
 - bs_required_text, 49
 - flask_failed, 50
 - rest, 50
 - suite, 49
 - super_class, 50
 - test, 49
 - verbosity, 50
- test_rest.fake, 56
 - __getattr__, 57
- test_rest.py, 170
- test_rest.test_rest, 137
 - app, 149
 - create_app, 139
 - make_request, 139
 - make_request_html, 140
 - make_request_json, 141
 - setUp, 142
 - test_htmlize_doctype, 142
 - test_htmlize_list_aka_links, 143
 - test_htmlize_title, 143
 - test_request_wants_json, 144
 - test_rest_redirect, 144
 - test_route_addresses, 144
 - test_route_addresses_date, 145
 - test_route_addresses_month, 145
 - test_route_addresses_specific, 145
 - test_route_addresses_specific_date, 146
 - test_route_addresses_specific_month, 146
 - test_route_addresses_specific_time, 147
 - test_route_addresses_specific_year, 147
 - test_route_addresses_year, 147
 - test_route_entry_point_html, 148
 - test_route_entry_point_json, 148
 - test_show_ip_address_activity_times_redirect, 148
- test_rest_redirect
 - test_rest.test_rest, 144
- test_route_addresses
 - test_rest.test_rest, 144
- test_route_addresses_date
 - test_rest.test_rest, 145
- test_route_addresses_month
 - test_rest.test_rest, 145
- test_route_addresses_specific
 - test_rest.test_rest, 145
- test_route_addresses_specific_date
 - test_rest.test_rest, 146
- test_route_addresses_specific_month
 - test_rest.test_rest, 146
- test_route_addresses_specific_time
 - test_rest.test_rest, 147
- test_route_addresses_specific_year
 - test_rest.test_rest, 147
- test_route_addresses_year
 - test_rest.test_rest, 147
- test_route_entry_point_html
 - test_rest.test_rest, 148
- test_route_entry_point_json
 - test_rest.test_rest, 148
- test_serializable, 50
 - bs_failed, 51
 - bs_required_text, 51
 - suite, 51
 - test, 51
 - verbosity, 52
- test_serializable.py, 171
- test_serializable.test_files, 124

- test_base_init, 125
- test_htmlize_lines_links, 125
- test_htmlize_ors, 126
- test_jsonify_links, 126
- test_jsonify_links_with_extend, 126
- test_jsonify_ors, 126
- test_show_ip_address_activity_times_redirect
 - test_rest.test_rest, 148
- test_time, 52
 - suite, 52
 - test, 52
 - verbosity, 53
- test_time.py, 171
- test_time.test_time_wrapper, 150
 - test_format, 151
 - test_FormatTimeWrapper, 152
 - test_FormatTimeWrapper_get, 152
 - test_next_day, 152
 - test_next_month, 152
 - test_prev_day, 152
 - test_prev_month, 152
 - test_TimeWrapper, 153
 - test_TimeWrapper_get, 153
 - test_TorTimeWrapper, 153
 - test_TorTimeWrapper_get, 153
 - test_zero, 153
- test_TimeWrapper
 - test_time.test_time_wrapper, 153
- test_TimeWrapper_get
 - test_time.test_time_wrapper, 153
- test_TorTimeWrapper
 - test_time.test_time_wrapper, 153
- test_TorTimeWrapper_get
 - test_time.test_time_wrapper, 153
- test_zero
 - test_time.test_time_wrapper, 153
- time
 - test_cp.fake_cp_args, 59
- time_parser, 53
 - DAY_HAS_SECONDS, 53
 - months, 54
- time_parser.FormatTimeWrapper, 60
 - __init__, 61
 - v, 62
- time_parser.py, 172
- time_parser.TimeWrapper, 154
 - __init__, 155
 - v, 156
- time_parser.TimeWrapperBase, 156
 - __get_days_in_month, 157
 - __str__, 157
 - format, 158
 - get, 158
 - next_day, 158
 - next_month, 158
 - prev_day, 158
 - prev_month, 159
- time_parser.TorTimeWrapper, 159
 - __init__, 160
 - v, 161
- type, 161
- update
 - test_cp.fake_cp_args, 59
- update_keeprunning
 - test_cp.fake_cp_args, 59
- update_path_consensus
 - consensus_parser, 18
- v
 - time_parser.FormatTimeWrapper, 62
 - time_parser.TimeWrapper, 156
 - time_parser.TorTimeWrapper, 161
- verbosity
 - test, 40
 - test_additional_info, 42
 - test_cp, 43
 - test_files, 45
 - test_geolite2, 47
 - test_or, 48
 - test_rest, 50
 - test_serializable, 52
 - test_time, 53
- WAIT_FOR_CONSENSUS
 - consensus_parser, 20
- write_location
 - preprocess_geolite, 29
- write_network
 - preprocess_geolite, 29
- write_preprocessed
 - consensus_parser, 19
 - test_cp.fake_cp_args, 59
- write_preprocessed_from_consensus_file
 - consensus_parser, 19
- wsgi_app
 - profilerest, 31