NAME
    ldns_dnssec_data_chain, ldns_dnssec_data_chain_struct, ldns_dnssec_trust_tree - data structures for
    validation chains

SYNOPSIS
    #include <stdint.h>
    #include <stdbool.h>

    #include <ldns/ldns.h>

     ldns_dnssec_data_chain_struct();

DESCRIPTION
    ldns_dnssec_data_chain
          Chain structure that contains all DNSSEC data needed to
          verify an rrset
          struct ldns_dnssec_data_chain_struct
          {
                  ldns_rr_list *rrset;
                  ldns_rr_list *signatures;
                  ldns_rr_type parent_type;
                  ldns_dnssec_data_chain *parent;
                  ldns_pkt_rcode packet_rcode;
                  ldns_rr_type packet_qtype;
                  bool packet_nodata;
          };

          typedef struct ldns_dnssec_data_chain_struct ldns_dnssec_data_chain;

    ldns_dnssec_data_chain_struct()

    ldns_dnssec_trust_tree
          Tree structure that contains the relation of DNSSEC data,
          and their cryptographic status.

          This tree is derived from a data_chain, and can be used
          to look whether there is a connection between an RRSET
          and a trusted key. The tree only contains pointers to the
          data_chain, and therefore one should *never* free() the

data_chain when there is still a trust tree derived from
that chain.

Example tree:

```
  key   key   key
   \   |   /
    \  |  /
     \ | /
       ds
       |
      key
       |
      key
       |
       rr
```

For each signature there is a parent; if the parent
pointer is null, it couldn't be found and there was no
denial; otherwise is a tree which contains either a
DNSKEY, a DS, or a NSEC rr
struct ldns_dnssec_trust_tree_struct
{

```
        ldns_rr *rr;
        /* the complete rrset this rr was in */
        ldns_rr_list *rrset;
        ldns_dnssec_trust_tree *parents[LDNS_DNSSEC_TRUST_TREE_MAX_PARENTS];
        ldns_status parent_status[LDNS_DNSSEC_TRUST_TREE_MAX_PARENTS];
        /** for debugging, add signatures too (you might want
           those if they contain errors) */
        ldns_rr *parent_signature[LDNS_DNSSEC_TRUST_TREE_MAX_PARENTS];
        size_t parent_count;
};
```

typedef struct ldns_dnssec_trust_tree_struct ldns_dnssec_trust_tree;

AUTHOR
The ldns team at NLnet Labs.

REPORTING BUGS

Please report bugs to ldns-team@nlnetlabs.nl or in our bugzilla at
http://www.nlnetlabs.nl/bugs/index.html

COPYRIGHT

Copyright (c) 2004 - 2006 NLnet Labs.

Licensed under the BSD License. There is NO warranty; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE.

SEE ALSO

ldns_dnssec_data_chain_new, *ldns_dnssec_trust_tree_new*, *ldns_dnssec_verify_denial*. And **perldoc
Net::DNS**, **RFC1034**, **RFC1035**, **RFC4033**, **RFC4034** and **RFC4035**.

REMARKS

This manpage was automatically generated from the ldns source code.