

ngx_inet.h Documentation :

Macros Defined :

```
#define NGX_INET_ADDRSTRLEN (sizeof("255.255.255.255") - 1)

#define NGX_INET6_ADDRSTRLEN \
    (sizeof("ffff:ffff:ffff:ffff:ffff:ffff:255.255.255.255") - 1)

#define NGX_UNIX_ADDRSTRLEN \
    (sizeof(struct sockaddr_un) - offsetof(struct sockaddr_un, sun_path))

#define NGX_SOCKADDR_STRLEN (sizeof("unix:") - 1 + NGX_UNIX_ADDRSTRLEN)
#define NGX_SOCKADDR_STRLEN (NGX_INET6_ADDRSTRLEN + sizeof("[]:65535") - 1)
#define NGX_SOCKADDRLEN sizeof(struct sockaddr_un)
#define NGX_SOCKADDRLEN 512
```

Data Structures :

```
typedef struct {
    in_addr_t    addr;
    in_addr_t    mask;
} ngx_in_cidr_t;

#if (NGX_HAVE_INET6)
typedef struct {
    struct in6_addr    addr;
    struct in6_addr    mask;
} ngx_in6_cidr_t;
#endif

typedef struct {
    ngx_uint_t    family;
    union {
        ngx_in_cidr_t    in;
        #if (NGX_HAVE_INET6)
            ngx_in6_cidr_t    in6;
        #endif
    } u;
} ngx_cidr_t;

typedef struct {
    struct sockaddr    *sockaddr;
    socklen_t          socklen;
    ngx_str_t          name;
} ngx_addr_t;
```

```
typedef struct {
    ngx_str_t    url;
    ngx_str_t    host;
    ngx_str_t    port_text;
    ngx_str_t    uri;
    in_port_t    port;
    in_port_t    default_port;
    int          family;
    unsigned      listen:1;
    unsigned      uri_part:1;
    unsigned      no_resolve:1;
    unsigned      one_addr:1; /* compatibility */
    unsigned      no_port:1;
    unsigned      wildcard:1;
    socklen_t     socklen;
    u_char        sockaddr[NGX_SOCKADDRLLEN];
    ngx_addr_t    *addrs;
    ngx_uint_t    naddrs;
    char          *err;
} ngx_url_t;
```

Functions Defined :

```
in_addr_t ngx_inet_addr( u_char *text,
                        size_t len
                        );
```

```
ngx_int_t ngx_inet6_addr( u_char *p,
                        size_t len,
                        u_char *addr
                        );
```

```
size_t ngx_inet6_ntop( u_char *p,
                    u_char *text,
                    size_t len
                    );
```

```
size_t ngx_sock_ntop( struct sockaddr *sa,
                    socklen_t socklen,
                    u_char *text,
                    size_t len,
                    ngx_uint_t port
                    );
```

```
size_t ngx_inet_ntop( int family,
                    void *addr,
```

```
    u_char *text,  
    size_t len  
);
```

```
ngx_int_t ngx_ptocidr( ngx_str_t *text,  
    ngx_cidr_t *cidr  
);
```

```
ngx_int_t ngx_parse_addr( ngx_pool_t *pool,  
    ngx_addr_t *addr,  
    u_char *text,  
    size_t len  
);
```

```
ngx_int_t ngx_parse_url( ngx_pool_t *pool,  
    ngx_url_t *u  
);
```

```
ngx_int_t ngx_inet_resolve_host( ngx_pool_t *pool,  
    ngx_url_t *u  
);
```

```
ngx_int_t ngx_cmp_sockaddr( struct sockaddr *sa1,  
    socklen_t slen1,  
    struct sockaddr *sa2,  
    socklen_t slen2,  
    ngx_uint_t cmp_port  
);
```

Include Dependency Graph :

