ngx_string.h Documentation :

Macros Defined :

```
#define ngx_string(str)     { sizeof(str) - 1, (u_char *) str }
#define ngx_null_string     { 0, NULL }

#define ngx_str_set(str, text)                                               \
    (str)->len = sizeof(text) - 1; (str)->data = (u_char *) text

#define ngx_str_null(str)   (str)->len = 0; (str)->data = NULL
#define ngx_tolower(c)      (u_char) ((c >= 'A' && c <= 'Z') ? (c | 0x20) : c)
#define ngx_toupper(c)      (u_char) ((c >= 'a' && c <= 'z') ? (c & ~0x20) : c)
#define ngx_strncmp(s1, s2, n)  strncmp((const char *) s1, (const char *) s2, n)
#define ngx_strcmp(s1, s2)  strcmp((const char *) s1, (const char *) s2)
#define ngx_strstr(s1, s2)  strstr((const char *) s1, (const char *) s2)
#define ngx_strlen(s)       strlen((const char *) s)
#define ngx_strchr(s1, c)   strchr((const char *) s1, (int) c)
#define ngx_memzero(buf, n)       (void) memset(buf, 0, n)
#define ngx_memset(buf, c, n)     (void) memset(buf, c, n)
#define NGX_ESCAPE_URI            0
#define NGX_ESCAPE_ARGS           1
#define NGX_ESCAPE_URI_COMPONENT  2
#define NGX_ESCAPE_HTML           3
#define NGX_ESCAPE_REFRESH        4
#define NGX_ESCAPE_MEMCACHED      5
#define NGX_ESCAPE_MAIL_AUTH      6
#define NGX_UNESCAPE_URI       1
#define NGX_UNESCAPE_REDIRECT  2
#define ngx_base64_encoded_length(len)  (((len + 2) / 3) * 4)
#define ngx_base64_decoded_length(len)  (((len + 3) / 4) * 3)
#define ngx_qsort             qsort

#define ngx_value_helper(n)   #n
#define ngx_value(n)          ngx_value_helper(n)
```

Data Structures :

```
typedef struct {
    size_t      len;
    u_char     *data;
} ngx_str_t;


typedef struct {
    ngx_str_t   key;
    ngx_str_t   value;
} ngx_keyval_t;


typedef struct {
    unsigned    len:28;
    unsigned    valid:1;
    unsigned    no_cacheable:1;
    unsigned    not_found:1;
    unsigned    escape:1;
    u_char     *data;
} ngx_variable_value_t;
```

```
typedef struct {
    ngx_rbtree_node_t          node;
    ngx_str_t                  str;
} ngx_str_node_t;
```

Functions Defined :

```
u_char *ngx_cpystrn(u_char *dst, u_char *src, size_t n);

u_char *ngx_pstrdup(ngx_pool_t *pool, ngx_str_t *src);

u_char * ngx_cdecl ngx_sprintf(u_char *buf, const char *fmt, ...);

u_char * ngx_cdecl ngx_snprintf(u_char *buf, size_t max, const char *fmt, ...);

u_char * ngx_cdecl ngx_slprintf(u_char *buf, u_char *last, const char *fmt, ...);

u_char *ngx_vslprintf(u_char *buf, u_char *last, const char *fmt, va_list args);

ngx_int_t ngx_strcasecmp(u_char *s1, u_char *s2);

ngx_int_t ngx_strncasecmp(u_char *s1, u_char *s2, size_t n);

u_char *ngx_strnstr(u_char *s1, char *s2, size_t n);

u_char *ngx_strstrn(u_char *s1, char *s2, size_t n);

u_char *ngx_strcasestrn(u_char *s1, char *s2, size_t n);

u_char *ngx_strlcasestrn(u_char *s1, u_char *last, u_char *s2, size_t n);

ngx_int_t ngx_rstrncmp(u_char *s1, u_char *s2, size_t n);

ngx_int_t ngx_rstrncasecmp(u_char *s1, u_char *s2, size_t n);

ngx_int_t ngx_memn2cmp(u_char *s1, u_char *s2, size_t n1, size_t n2);

ngx_int_t ngx_dns_strcmp(u_char *s1, u_char *s2);

ngx_int_t ngx_filename_cmp(u_char *s1, u_char *s2, size_t n);

ngx_int_t ngx_atoi(u_char *line, size_t n);

ngx_int_t ngx_atofp(u_char *line, size_t n, size_t point);

ssize_t ngx_atosz(u_char *line, size_t n);

off_t ngx_atoof(u_char *line, size_t n);

time_t ngx_atotm(u_char *line, size_t n);

ngx_int_t ngx_hextoi(u_char *line, size_t n);

u_char *ngx_hex_dump(u_char *dst, u_char *src, size_t len);
```

```
void ngx_encode_base64(ngx_str_t *dst, ngx_str_t *src);

void ngx_encode_base64url(ngx_str_t *dst, ngx_str_t *src);

ngx_int_t ngx_decode_base64(ngx_str_t *dst, ngx_str_t *src);

ngx_int_t ngx_decode_base64url(ngx_str_t *dst, ngx_str_t *src);

uint32_t ngx_utf8_decode(u_char **p, size_t n);

size_t ngx_utf8_length(u_char *p, size_t n);

u_char *ngx_utf8_cpystrn(u_char *dst, u_char *src, size_t n, size_t len);

uintptr_t ngx_escape_uri(u_char *dst, u_char *src, size_t size, ngx_uint_t type);

void ngx_unescape_uri(u_char **dst, u_char **src, size_t size, ngx_uint_t type);

uintptr_t ngx_escape_html(u_char *dst, u_char *src, size_t size);

void ngx_str_rbtree_insert_value(ngx_rbtree_node_t *temp, ngx_rbtree_node_t *node, ngx_rbtree_node_t
*sentinel);

ngx_str_node_t *ngx_str_rbtree_lookup(ngx_rbtree_t *rbtree, ngx_str_t *name, uint32_t hash);

void ngx_sort(void *base, size_t n, size_t size, ngx_int_t (*cmp)(const void *, const void *));

void ngx_strlow(u_char *dst, u_char *src, size_t n);
```

Include Dependency Graph :