ngx_array.h Documentation:

// Function to create a new array by allocating memory from pool

ngx_array_t* ngx_array_create(ngx _pool_t* p,   ->Address of pool used to allocate memory for array elements

ngx_unit_t   n,   ->No of elements for which memory is allocated

size_t   size ->size of each element of array (in bytes)

)

//Function to destroy the array to free used memory by returning it to memory pool

void ngx_array_destroy(ngx_array_t * a   ->Array passed as parameter will be destroyed and

)   memory will be returned back to the pool

//Function to push a new element to end of the array

void* ngx_array_push (ngx_array_t * a   -> Array to which elements will be added

)

//Function to push 'N' elements to the end of array

void* ngx_array_push_n(ngx_array_t* a,   -> Array to which elements will be added

ngx_uint_t   n   -> Number of elements to be allocated

)

Data Structures Defined:

struct ngx_array_t {

        void *   elts,   -> Pointer to elements of the array

        ngx_uint_t   nelts,   -> Number of elements in the array

        size_t   size,   -> Size of each element (in bytes)

        ngx_unit_t   nalloc, -> Number of elements for which memory is allocated

        ngx_pool_t * pool   -> Pointer to the address pool from which memory will be allocated to the array

}

Macros Defined:

No macros defined

ngx_array.h → ngx_core.h

ngx_errno.h
ngx_atomic.h
ngx_thread.h
ngx_rbtree.h
ngx_time.h
ngx_socket.h
ngx_string.h
ngx_files.h
ngx_shmen.h
ngx_process.h
ngx_user.h
ngx_parse.h
ngx_log.h
ngx_alloc.h
ngx_palloc.h
ngx_buf.h
ngx_queue.h
ngx_list.h
ngx_hash.h
ngx_file.h
ngx_crc.h
ngx_crc32.h
ngx_murmurhash.h
ngx_radix_tree.h
ngx_times.h
ngx_shmtx.h
ngx_slab.h
ngx_inet.h
ngx_cycle.h
ngx_resolver.h
ngx_process_cycle.h
ngx_conf_file.h
ngx_open_file_cache.h
ngx_os.h

ngx_config.h

ngx_auto_headers.h
ngx_posix_config.h

sys/types.h
sys/time.h
stdarg.h
stddef.h
stdio.h
stdlib.h
errno.h
string.h
signal.h
pwd.h