

ngx_queue.h Documentation :

Macros Defined :

```
#define ngx_queue_init(q) \
    (q)->prev = q; \
    (q)->next = q

#define ngx_queue_empty(h) \
    (h == (h)->prev)

#define ngx_queue_insert_head(h, x) \
    (x)->next = (h)->next; \
    (x)->next->prev = x; \
    (x)->prev = h; \
    (h)->next = x

#define ngx_queue_insert_after ngx_queue_insert_head

#define ngx_queue_insert_tail(h, x) \
    (x)->prev = (h)->prev; \
    (x)->prev->next = x; \
    (x)->next = h; \
    (h)->prev = x

#define ngx_queue_head(h) \
    (h)->next

#define ngx_queue_last(h) \
    (h)->prev

#define ngx_queue_sentinel(h) \
    (h)

#define ngx_queue_next(q) \
    (q)->next

#define ngx_queue_prev(q) \
    (q)->prev
```

```
#define ngx_queue_remove(x) \
    (x)->next->prev = (x)->prev; \
    (x)->prev->next = (x)->next; \
    (x)->prev = NULL; \
    (x)->next = NULL
```

```
#define ngx_queue_remove(x) \
    (x)->next->prev = (x)->prev; \
    (x)->prev->next = (x)->next
```

```
#define ngx_queue_split(h, q, n) \
    (n)->prev = (h)->prev; \
    (n)->prev->next = n; \
    (n)->next = q; \
    (h)->prev = (q)->prev; \
    (h)->prev->next = h; \
    (q)->prev = n;
```

```
#define ngx_queue_add(h, n) \
    (h)->prev->next = (n)->next; \
    (n)->next->prev = (h)->prev; \
    (h)->prev = (n)->prev; \
    (h)->prev->next = h;
```

```
#define ngx_queue_data(q, type, link) \
    (type *) ((u_char *) q - offsetof(type, link))
```

Type Definitions :

```
typedef struct ngx_queue_s ngx_queue_t;
```

Data Structures :

```
struct ngx_queue_s {
    ngx_queue_t *prev;
    ngx_queue_t *next;
}
```

Functions Defined :

```
ngx_queue_t *ngx_queue_middle( ngx_queue_t *queue  
                                )
```

```
void ngx_queue_sort( ngx_queue_t *queue,  
                     ngx_int_t (*cmp)(const ngx_queue_t *, const ngx_queue_t *)  
                     )
```

Include Dependency Graph :

