

重要类型定义

```
// nginx内存池的主结构体类型
struct ngx_pool_s {
    ngx_pool_data_t    d;    // 内存池的数据头
    size_t             max;  // 小块内存分配的最大值
    ngx_pool_t         *current; // 小块内存池入口指针
    ngx_chain_t        *chain;
    ngx_pool_large_t   *large; // 大块内存分配入口指针
    ngx_pool_cleanup_t *cleanup; // 清理函数handler的入口指针
    ngx_log_t          *log;
};
```

```
typedef struct ngx_pool_s      ngx_pool_t;
// 小块内存数据头信息
typedef struct {
    u_char                *last; // 可分配内存开始位置
    u_char                *end;  // 可分配内存末尾位置
    ngx_pool_t            *next; // 保存下一个内存池的地址
    ngx_uint_t            failed; // 记录当前内存池分配失败的次数
} ngx_pool_data_t;
```

```
typedef struct ngx_pool_large_s ngx_pool_large_t;
// 大块内存类型定义
struct ngx_pool_large_s {
    ngx_pool_large_t    *next; // 下一个大块内存
    void                *alloc; // 记录分配的大块内存的起始地址
};
```

```
typedef void (*ngx_pool_cleanup_pt)(void *data); // 清理回调函数的类型定义

typedef struct ngx_pool_cleanup_s ngx_pool_cleanup_t;
// 清理操作的类型定义，包括一个清理回调函数，传给回调函数的数据和下一个清理操作的地址
struct ngx_pool_cleanup_s {
    ngx_pool_cleanup_pt  handler; // 清理回调函数
    void                *data;    // 传递给回调函数的指针
    ngx_pool_cleanup_t  *next;    // 指向下一个清理操作
};
```

nginx内存池重要函数接口

```
ngx_pool_t *ngx_create_pool(size_t size, ngx_log_t *log); // 创建内存池
void ngx_destroy_pool(ngx_pool_t *pool); // 销毁内存池
void ngx_reset_pool(ngx_pool_t *pool); // 重置内存池
```

```
void *ngx_palloc(ngx_pool_t *pool, size_t size); // 内存分配函数，支持内存对齐
void *ngx_pnalloc(ngx_pool_t *pool, size_t size); // 内存分配函数，不支持内存对齐
void *ngx_pcalloc(ngx_pool_t *pool, size_t size); // 内存分配函数，支持内存初始化0
ngx_int_t ngx_pfree(ngx_pool_t *pool, void *p // 内存释放（大块内存）
```

```
ngx_pool_cleanup_t *ngx_pool_cleanup_add(ngx_pool_t *p, size_t size); // 添加清理
handler
```