



## توابع سرور

```
void myAddItemtoObject(char* response, char* part1, char* part1str, char* part2, char* part2str)
```

```
/*
```

این تابع پنج ورودی استرینگ میگیرد که ورودی اول رشته ای به صورت جیسون است و فرایند اضافه کردن آیتم ها روی آن انجام می شود و رشته ای به صورت جیسون با دو آیتم با نام هایی که در ورودی دوم و چهارم گرفته است و با `valueString`هایی که به ترتیب در ورودی های سوم و پنجم گرفته است خواهد ساخت و در آرگومان اول ذخیره می کند. استفاده از این تابع همانند استفاده دوبار از تابع `JSON_AddItemtoObject` می باشد.

خروجی ندارد

```
*/
```

```
void myCreateArray(char* array);
```

```
/*
```

این تابع یک رشته میگیرد که آن را برای ساخت آرایه ای به صورت جیسون شامل یک آرایه از استرینگ ها آماده می کند.

```
*/
```

```
void myAddItemtoArray(char* array, char* item);
```

```
/*
```

این تابع دو ورودی استرینگ میگیرد که ورودی اول آرایه ای به صورت جیسون است، ورودی دوم آیتمی از جیسون است که آن را به آرایه `array` که از قبل با تابع `myCreateArray` ساخته شده است اضافه میکند. استفاده از این تابع همانند استفاده از `JSON_AddItemToArray` می باشد.

خروجی ندارد

```
*/
```

```
void myArrayCreated(char* array);
```

```
/*
```

این تابع به فرایند ساخت آرایه `array` خاتمه می دهد

```
*/
```

```
void myAddArraytoObject(char* response, char* part1, char* part1str, char* part2, char* part2str);
```

```
/*
```

این تابع پنج ورودی استرینگ میگیرد که ورودی اول رشته ای به صورت جیسون است و فرایند اضافه کردن آیتم ها روی آن انجام می شود



و رشته ای به صورت جیسون با آیتمی با نامی که در ورودی دوم گرفته است و با `valueString` که در ورودی سوم گرفته است و آیتمی با نامی که در آرگومان چهارم گرفته است و رشته ای به صورت آرایه جیسون که در ورودی پنجم گرفته است می سازد. تفاوت این تابع و `myAddItemtoObject` در این است که آخرین آرگومان این تابع، رشته ای است که به صورت آرایه ای از جیسون است. استفاده از این تابع همانند استفاده دوبار از تابع `JSON_AddItemtoObject` می باشد.

خروجی ندارد.

\*/

```
void myGetItem(char* response, char* part1, char* part1str, char* part2, char* part2str);
```

/\*

این تابع پنج ورودی استرینگ میگیرد که ورودی اول رشته ای به صورت جیسون است که از سرور گرفته است، ورودی دوم و چهارم آیتمی از جیسون است که `valueString` مربوط به آنها را به ترتیب در رشته های سوم و پنجم تابع ذخیره میکند. استفاده از این تابع همانند استفاده دوبار از تابع `JSON_GetObjectItem` می باشد.

خروجی ندارد.

\*/

```
int findChannel(char key[MAX]);
```

/\*

این تابع اسم یک کانال را میگیرد و در کانال هایی که در پوشه `Resources\Channels` ذخیره شده است، جستجو می کند و اگر کانالی با این اسم پیدا کرد، ۱ و در غیر این صورت ۰ را خروجی می دهد.

\*/

```
int findOnlineUser(const char keyToken[25], char Name[100]);
```

/\*

این تابع یک توکن را میگیرد و در بین افراد آنلاین جستجو میکند (زیرا توکن صرفا مخصوص افراد آنلاین است) و اگر کاربر آنلاینی را با این توکن پیدا کرد، اسم او را در آرگومان دوم ذخیره میکند و یک خروجی میدهد و اگر چنین کاربری یافت نشد، ۰ خروجی میدهد.

\*/



```
int findActiveChannel(char nameOfChannel[100]);
```

```
/*
```

این تابع اسم یک کانال را میگیرد و در کانال های فعال (کانال هایی که حداقل یک عضو دارند) ، جستجو می کند و اگر کانالی با این اسم پیدا کرد، ۱ و در غیر این صورت ۰ را خروجی می دهد

```
*/
```

```
void createToken();
```

```
/*
```

این تابع یک رشته شامل ۲۵ کاراکتر از کاراکتر های حروف انگلیسی بزرگ و کوچک و اعداد می سازد و آن را در متغیر گلوبالی به اسم توکن ذخیره میکند.

```
*/
```

```
void create_server_socket();
```

```
/*
```

این تابع برای اتصال کلاینت ها به سرور، سوکت را می سازد

```
*/
```

```
void myaccept();
```

```
/*
```

این تابع پس از درخواست کلاینت ها برای اتصال به سوکت ساخته شده توسط سرور، درخواست انها را قبول میکند. با توجه به اینکه پس از انجام هر درخواست کلاینت ها، این تابع در سرور اجرا می شود، پس سرور می تواند به صورت state less تعداد زیادی کلاینت را هندل کند

```
*/
```

```
void listening();
```

```
/*
```

این تابع ابتدا درخواست کلاینت را دریافت میکند و با توجه به درخواست، بررسی درخواست را به تابع های دیگر محول میکند

```
*/
```



```
void reggister();
```

```
/*
```

اگر درخواست کلاینت ثبت نام کاربر جدید باشد، تابع **listening** بررسی درخواست را به این تابع محول می کند. این تابع در دست بودن نام کاربری را بررسی می کند و اگر در دسترس بود، کاربر را ثبت نام می کند و فولدری برای این کاربر در **Resources\Users** شامل یوزر نیم و پسورد کاربر میسازد و پیام موفقیت را به کلاینت ارسال میکند و در غیر این صورت خطای مناسب را به سرور ارسال می کند.

```
*/
```

```
void login();
```

```
/*
```

اگر درخواست کلاینت لاگین کاربر باشد، تابع **listening** بررسی درخواست را به این تابع محول می کند. این تابع درستی یوزرنیم و پسورد را بررسی می کند و اگر مشکلی نبود، توکن ایجاد کرده و آن را به کلاینت می دهد و در غیر این صورت خطای لازم را ارسال میکند.

```
*/
```

```
void create_channel();
```

```
/*
```

اگر درخواست کلاینت ایجاد کانال جدید باشد، این تابع اجرا می شود. اگر اسم کانال مورد نظر کاربر تکراری بود و یا توکنی که چنین درخواستی داده بود در سرور وجود نداشت، پیغام خطای مناسب ارسال میکند و در غیر این صورت فایلی شامل اسم کانال و اولین پیام که

... created the channel

در مسیر **Resources\Channels** ایجاد می کند. و اسم کانال کاربر به اسم کانال تغییر میکند و اسم کاربر به اسم اعضای کانال اضافه می شود

و پیام موفقیت را به کلاینت ارسال میکند

```
*/
```

```
void join_channel();
```

```
/*
```

اگر درخواست کلاینت عضو شدن به کانال های موجود باشد، این تابع اجرا می شود. اگر کانالی با اسم مورد نظر کاربر وجود نداشت بود و یا توکنی که چنین درخواستی داده بود در سرور وجود نداشت، پیغام خطای مناسب ارسال میکند و در غیر این صورت در فایل مربوط به کانال، پیام

... joined the channel

در مسیر **Resources\Channels** ایجاد می کند. و اسم کانال کاربر به اسم کانال تغییر میکند و اسم کاربر به اسم اعضای کانال اضافه می شود و پیام

موفقیت را به کلاینت ارسال میکند

```
*/
```



```
void logout();
```

```
/*
```

اگر درخواست کلاینت خروج از حساب کاربری خود باشد، این تابع اجرا می شود. اگر کاربر در کانالی عضو نبود و یا توکنی که چنین درخواستی داده بود در سرور وجود نداشت، پیغام خطای مناسب ارسال میکند و در غیر این صورت توکن کاربر را از بین می برد و پیام موفقیت را به کلاینت ارسال میکند

```
*/
```

```
void send_message();
```

```
/*
```

اگر درخواست کلاینت ارسال پیام در کانال باشد، این تابع اجرا می شود. اگر کاربر در کانالی عضو نبود و یا توکنی که چنین درخواستی داده بود در سرور وجود نداشت، پیغام خطای مناسب ارسال میکند و در غیر این صورت در فایل مربوط به کانال، در مسیر Resources\Channels

پیام مورد نظر ذخیره می شود و پیام موفقیت را به کلاینت ارسال میکند.

```
*/
```

```
void refresh();
```

```
/*
```

اگر درخواست کلاینت دیدن پیام های جدید کانال باشد، این تابع اجرا می شود. اگر کاربر در کانالی عضو نبود و یا توکنی که چنین درخواستی داده بود در سرور وجود نداشت، پیغام خطای مناسب ارسال میکند و در غیر این صورت از فایل مربوط به کانال، در مسیر Resources\Channels

پیام های جدید به کلاینت ارسال میشوند و آخرین خطی که از پیام ها خوانده شده برای کاربر تغییر میکند

```
*/
```

```
void leave();
```

```
/*
```

اگر درخواست کلاینت خروج از کانال باشد، این تابع اجرا می شود. اگر کاربر در کانالی عضو نبود و یا توکنی که چنین درخواستی داده بود در سرور وجود نداشت، پیغام خطای مناسب ارسال میکند و در غیر این صورت در فایل مربوط به کانال، پیام

... left the channel

در مسیر Resources\Channels ایجاد می کند. و کانال مربوط به کاربر حذف میشود و کاربر از اعضای کانال حذف می شود و پیام موفقیت را به کلاینت ارسال میکند

```
*/
```



```
void channel_members();
```

```
/*
```

اگر درخواست کلاینت مشاهده اعضای کانال باشد، این تابع اجرا می شود. اگر کاربر در کانالی عضو نبود و یا توکنی که چنین درخواستی داده بود در سرور وجود نداشت، پیام خطای مناسب ارسال میکند و در غیر این صورت لیست اعضای کانال به کلاینت ارسال می شود

```
*/
```

```
void search_members();
```

```
/*
```

اگر درخواست کلاینت سرچ در اعضای کانال باشد، این تابع اجرا می شود. اگر کاربر در کانالی عضو نبود و یا توکنی که چنین درخواستی داده بود در سرور وجود نداشت، پیام خطای مناسب ارسال میکند و در غیر این صورت به کلاینت فرستاده می شود که کاربر مورد نظر در این کانال وجود دارد یا خیر.

```
*/
```

```
void search_messages();
```

```
/*
```

اگر درخواست کلاینت سرچ یک واژه در پیام های کانال باشد، این تابع اجرا می شود. اگر کاربر در کانالی عضو نبود و یا توکنی که چنین درخواستی داده بود در سرور وجود نداشت، پیام خطای مناسب ارسال میکند و در غیر این صورت. پیام هایی که شامل این کلمه هستند به کلاینت فرستاده می شود یا این که فرستاده می شود که پیامی که شامل این کلمه باشد یافت نشد.

```
*/
```



## توابع کلاینت

```
void myGetObjectItem(char* response, char* part1, char* part1str, char* part2, char* part2str);
```

```
/*
```

این تابع پنج ورودی استرینگ میگیرد که ورودی اول رشته ای به صورت جیسون است که از سرور گرفته است، ورودی دوم و چهارم آیتمی از جیسون است که valueString مربوط به آنها را به ترتیب در رشته های سوم و پنجم تابع ذخیره میکند. استفاده از این تابع همانند استفاده دوبار از تابع

cJSON\_GetObjectItem می باشد. خروجی ندارد.

```
*/
```

```
void myGetArrayItems(char* response, char* part1, char* part1str, char* part2, char* part2str);
```

```
/*
```

این تابع پنج ورودی استرینگ میگیرد که ورودی اول رشته ای به صورت جیسون است که از سرور گرفته است، ورودی دوم و چهارم آیتمی از جیسون است که valueString مربوط به آنها را به ترتیب در رشته های سوم و پنجم تابع ذخیره میکند. تفاوت این تابع و تابع قبل در این است که ورودی پنجم یک آرایه به صورت جیسون است. استفاده از این تابع همانند دو بار استفاده از تابع cJSON\_GetObjectItem می باشد. خروجی ندارد.

```
*/
```

```
void myPrintArrayItems(char* response);
```

```
/*
```

این تابع یک رشته به صورت جیسون را میگیرد و اعضای آیتم دوم آن که یک آرایه ای از جیسون ها است را به صورت sender: content پرینت می کند. از این تابع در رفرش و برای پرینت پیام های جدید استفاده می شود خروجی ندارد.

```
*/
```

```
void setting();
```

```
/*
```

این تابع هنگامی که کاربر قصد تغییر تنظیمات چت اپلیکیشن خود را داشته باشد صدا زده میشود. این تابع ۲ گزینه شامل رنگ نوشته ها و رنگ پس زمینه دارد که هر کدام از آنها شامل ۱۶ رنگ می باشند

```
*/
```



```
int socket_creator();
```

```
/*
```

این تابع برای اتصال به سرور، سوکت را می سازد

```
*/
```

```
void menu();
```

```
/*
```

بعد از لاگین شدن این تابع صدا زده میشود و امکان ایجاد کانال، عضو شدن در دیگر کانال ها، و لاگ اوت را به ما میدهد

```
*/
```

```
void loginClient();
```

```
/*
```

اگر کاربر درخواست لاگین داشته باشد، این تابع صدا زده می شود و یوزرنیم و پسورد کاربر را میگیرد و این درخواست را به سرور می فرستد و بر اساس پاسخ سرور تصمیم گرفته می شود که خطا نمایش داده شود و یا کاربر را به منو منتقل کند.

```
*/
```

```
void registerClient();
```

```
/*
```

اگر کاربر درخواست ثبت نام داشته باشد، این تابع صدا زده می شود و یوزرنیم و پسورد مورد نظر کاربر را میگیرد و این درخواست را به سرور می فرستد و بر اساس پاسخ سرور تصمیم گرفته می شود که خطا نمایش داده شود و یا کاربر را به تابع main برای لاگین منتقل کند.

```
*/
```

```
void creatChannel();
```

```
/*
```

اگر کاربر درخواست ایجاد کانال داشته باشد، این تابع صدا زده می شود و اسم مورد نظر کاربر را برای کانال میگیرد و این درخواست را به سرور می فرستد و بر اساس پاسخ سرور تصمیم گرفته می شود که خطا نمایش داده شود و یا کاربر را به منوی چت در کانال منتقل کند.

```
*/
```





```
void join_channel();
```

```
/*
```

اگر کاربر درخواست عضو کانال موجود را داشته باشد، این تابع صدا زده می شود و اسم کانال میگیرد و این درخواست را به سرور می فرستد و بر اساس پاسخ سرور تصمیم گرفته می شود که خطا نمایش داده شود و یا کاربر را به منوی چت در کانال منتقل کند.

```
*/
```

```
void logout();
```

```
/*
```

اگر کاربر درخواست خارج شدن از حساب خود را داشته باشد، این تابع صدا زده می شود و این درخواست را به سرور می فرستد و بر اساس پاسخ سرور تصمیم گرفته می شود که خطا نمایش داده شود و یا کاربر را به تابع main برگرداند.

```
*/
```

```
void chatMenu();
```

```
/*
```

بعد از ساختن کانال یا عضو کانال شدن، این تابع صدا زده می شود و به کاربر امکان ارسال پیام در گروه، دیدن پیام های جدید، دیدن اعضای کانال، جستجوی کلمه ای در پیام ها، جستجوی کاربری با یوزر نیم او برای اینکه بداند در کانال حضور دارد یا خیر، و امکان ترک کردن کانال و بازگشت به منو را میدهد.

```
*/
```

```
void sendMessage();
```

```
/*
```

اگر کاربر عضو کانالی باشد، و درخواست ارسال پیام در کانال را داشته باشد، این تابع صدا زده می شود و این درخواست را به سرور می فرستد و بر اساس پاسخ سرور تصمیم گرفته می شود که خطا نمایش داده شود و یا خیر.

```
*/
```

```
void refresh();
```

```
/*
```

اگر کاربر عضو کانالی باشد، و درخواست دیدن پیام های ناخوانده در کانال را داشته باشد، این تابع صدا زده می شود و این درخواست را به سرور می فرستد و بر اساس پاسخ سرور تصمیم گرفته می شود که خطا نمایش داده شود و یا پیام های جدید را به کاربر نشان بدهد(آنها را از سرور دریافت کرده است).

```
*/
```



```
void channelMembers();
```

```
/*
```

اگر کاربر عضو کانالی باشد، و درخواست دیدن اعضای کانال را داشته باشد، این تابع صدا زده می شود و این درخواست را به سرور می فرستد و بر اساس پاسخ سرور تصمیم گرفته می شود که خطا نمایش داده شود و یا اعضای کانال را به کاربر نشان بدهد(آنها را از سرور دریافت کرده است!).

```
*/
```

```
void leave();
```

```
/*
```

اگر کاربر عضو کانالی باشد، و درخواست ترک کانال را داشته باشد، این تابع صدا زده می شود و این درخواست را به سرور می فرستد و بر اساس پاسخ سرور تصمیم گرفته می شود که خطا نمایش داده شود و یا کانال را ترک کند و به menu برگردد.

```
*/
```

```
void search_in_messages();
```

```
/*
```

اگر کاربر عضو کانالی باشد، و درخواست جستجو برای کلمه ای خاص در پیام های کانال را داشته باشد، این تابع صدا زده می شود و این درخواست را به سرور می فرستد و بر اساس پاسخ سرور تصمیم گرفته می شود که خطا نمایش داده شود و یا پیام های حاوی کلمه مورد نظر را به کاربر نشان بدهد.

```
*/
```

```
void search_in_members();
```

```
/*
```

اگر کاربر عضو کانالی باشد، و درخواست جستجو برای کاربری خاص در اعضای کانال را داشته باشد، این تابع صدا زده می شود و این درخواست را به سرور می فرستد و بر اساس پاسخ سرور تصمیم گرفته می شود

که خطا نمایش داده شود و یا اینکه بگوید که کاربر مورد نظر عضو کانال هست یا نه..

```
*/
```