

# تعریف پروژه پیام‌رسان

## مفاهیم کلی

### سرور مرکزی

سرور مرکزی روی پورت ۸۰۰۰ گوش می‌دهد. سرور مرکزی مسئول ایجاد فضای کار و همچنین معرفی فضای کارها به کلاینت‌ها است.

برنامه سرور مرکزی یک برنامه مستقل و همواره در حال اجرا است.

### میزبان

هر میزبان می‌تواند تعدادی فضای کار را در خود جا دهد. هر میزبان دارای یک آدرس (احتمالاً IP) و یک بازه پورت شامل حداکثر ۱۰۰۰ پورت و حداقل از پورت ۱۰۰۰۰ است.

هر میزبان یک برنامه مستقل جاوا است. میزبان از زمانی که اجرا می‌شود، خود را به سرور مرکزی معرفی می‌کند تا فضای کارها روی آن تشکیل شوند.

### فضای کار

هر فضای کار روی یک میزبان قرار دارد و روی یک پورت گوش می‌دهد. هر فضای کار یک نام یکتا دارد که یک رشته حداکثر ۶۰ کاراکتری از اعداد و حروف کوچک و بزرگ انگلیسی و \_ است.

فضای کارها روی میزبان‌ها اجرا می‌شوند و برنامه جاوای مستقلی ندارند.

### کلاینت

برنامه‌ای که کاربران نهایی با آن در تعامل هستند تا مدیریت فضای کارها، مدیریت گروه‌ها و گفتگوها را انجام دهند.

هر کلاینت یک برنامه مستقل جاوا است که در هر زمان می‌تواند راه‌اندازی یا متوقف شود.

## فاز اول: گپ شخصی

### میزبان

برای هر میزبان، برنامه میزبان را با پارامترهای آدرس و بازه IP اجرا می‌کنیم. همه میزبان‌ها همواره در حال اجرا هستند و هنگام اجرا این سناریو طی می‌شود:

۱- برنامه دستور زیر را به سرور مرکزی ارسال می‌کند:

```
create-host 127.0.0.1 10000 10999
```

که پارامترهای فوق به ترتیب آدرس و بازه پورت هستند.

۲- اگر ایرادی در پارامترها باشد (مثلاً پورت‌ها تکراری باشد)، سرور مرکزی پیام خطا می‌دهد:

ERROR Port in use by another host

ERROR Port number must be at least 10000

ERROR At most 1000 ports is allowed

...

۳- در صورت درست بودن پارامترها، به منظور اطمینان از اینکه میزبان درست گفته است، سرور مرکزی یکی از پورت‌های بازه گفته‌شده را بطور تصادفی اعلام می‌کند:

OK 10465

۴- میزبان، روی پورت مذکور گوش می‌دهد و به سرور مرکزی اعلام می‌کند:

check

۵- سرور مرکزی یک اتصال را به آدرس و پورت گفته‌شده باز می‌کند و یک کد ۱۰ رقمی عددی را به آن می‌فرستد و اتصال را می‌بندد:

OK 3212323321

۶- میزبان پس از دریافت کد، پورت مذکور را قطع و کد ۱۰ رقمی را به سرور مرکزی اعلام می‌کند:

۳۲۱۲۳۲۳۳۲۱

۷- اگر کد اشتباه باشد، سرور مرکزی خطا می‌دهد:

ERROR Invalid code

۸- اگر کد درست باشد، سرور مرکزی میزبان را ثبت کرده و پاسخ تأیید می‌دهد:

OK

۹- میزبان اتصال خود را با سرور مرکزی جهت ارتباط‌های بعدی حفظ می‌کند.

### کلاینت

به طور عادی، برای هر شخص یک برنامه کلاینت اجرا می‌شود که دو وظیفه را بر عهده دارد:

- بسته‌های دریافتی از فضای کار را نمایش می‌دهد

- دستورهای کاربر را دریافت کرده و به فضای کار می‌فرستد

کلاینت با پارامترهای phone\_number و password اجرا می‌شود.

دقت شود که در ادامه سند، معمولاً سناریوهای خطا بیان نشده‌اند ولی در پیاده‌سازی باید در نظر گرفته شود.

مهم: در سناریوهایی که مربوط به سرور مرکزی هستند، کلاینت همان موقع اتصال را برقرار کرده و وقتی کار تمام شد، آن را می‌بندد. ولی در سناریوهای مربوط به فضای کارها، کلاینت اتصال را باز نگه می‌دارد.

مهم: تقریباً در همه سناریوها کاربر باید از پیش لاگین شده باشد.

### سناریوی ثبت نام

۱- دستور ثبت نام را تایپ می‌کنیم:

register

۲- کلاینت، به سرور مرکزی متصل شده و دستور ثبت نام را به آن می‌فرستد.

register 09123456789 123456

۳- سرور مرکزی کاربر را ثبت کرده و تأیید می‌دهد:

OK

۴- کلاینت اتصال را می‌بندد.

### زیرسناریوی ورود در سرور مرکزی

این زیرسناریو معمولاً در ابتدای بقیه سناریوها اجرا می‌شود تا کاربر را به فضای کار بشناساند.

۱- کلاینت، به سرور مرکزی متصل شده و دستور ورود را به آن می‌فرستد:

login 09123456789 123456

۲- سرور مرکزی کاربر را لاگین کرده و تأیید می‌دهد:

OK

سناریوی ایجاد فضای کار

۱- دستور ایجاد فضای کار را تایپ می‌کنیم:

create-workspace company1

۲- کلاینت، در سرور مرکزی لاگین کرده و دستور فوق را به آن می‌فرستد.

۳- سرور مرکزی یکی از میزبان‌ها را به طور تصادفی و همچنین یک پورت خالی آن را انتخاب کرده و دستور را همراه با شناسه کاربر سازنده به آن می‌فرستد:

create-workspace 10143 1001

۴- میزبان فضای کار را روی پورت تعیین‌شده ساخته و به سرور مرکزی تأیید می‌دهد:

OK

۵- سرور مرکزی آدرس و پورت فضای کار جدید را به کلاینت اعلام می‌کند:

OK 127.0.0.1 10143

۶- کلاینت اتصال خود را با سرور مرکزی می‌بندد.

سناریوی اتصال به فضای کار ایجادشده

۱- دستور اتصال به فضای کار را تایپ می‌کنیم:

connect-workspace company1

۲- کلاینت، در سرور مرکزی لاگین کرده و دستور فوق را به آن می‌فرستد.

۳- سرور مرکزی یک توکن موقت برای کاربر ایجاد کرده و آدرس و پورت فضای کار را همراه با توکن مذکور به کلاینت می‌فرستد:

OK 127.0.0.1 10143 fkla48fhhf

توکن موقت شامل ۱۰ کاراکتر از اعداد و حروف کوچک انگلیسی است و حداکثر ۵ دقیقه عمر دارد.

۴- کلاینت اتصال خود را با سرور مرکزی بسته و به فضای کار متصل می‌شود و درخواست اتصال می‌دهد:

connect fkla48fhhf

۵- فضای کار (در واقع، میزبان)، توکن را به سرور مرکزی می‌فرستد:

whois fkla48fhhf

۶- سرور مرکزی شناسه کاربر را به فضای کار می‌دهد:

OK 1001

۷- در صورتی که اولین اتصال کاربر به فضای کار باشد، فضای کار نام کاربری را از او می‌پرسد:

username?

نام کاربری باید یکتا باشد.

۸- در ادامه مورد قبل، نام کاربری خود را تایپ می‌کنیم و کلاینت، آن را به فضای کار می‌فرستد:

ahmad

۹- فضای کار به کلاینت تأیید می‌دهد:

OK

لازم به ذکر است کلاینت اتصال خود را به فضای کار حفظ می‌کند. همچنین کلاینت همزمان فقط به یک فضای کار متصل است.

### سناریوی قطع از فضای کار

۱- دستور قطع اتصال را تایپ می‌کنیم:

disconnect

۲- کلاینت اتصال خود را با فضای کار قطع می‌کند.

### سناریوی ارسال پیام متنی به شخص دیگر

(قبلاً به فضای کار متصل شده‌ایم)

۱- دستور ارسال پیام را با داشتن نام کاربری او تایپ می‌کنیم:

send-message saeed {"type": "text", "body": "Salam chetori?"}

قالب پیام شامل عبارت send-message سپس نام کاربری شخص مقابل و پس از آن یک JSON توصیف‌کننده پیام است.

۲- کلاینت دستور را به فضای کار می‌فرستد.

۳- فضای کار پیام را ذخیره کرده و seq آن را به فرستنده می‌دهد:

OK 1

مقدار seq یک عدد افزایشی بین این دو کاربر است.

۴- فضای کار پیام را به کلاینت(های) طرف مقابل می‌فرستد:

receive-message ahmad {"seq": 1, "from": "ahmad", "type": "text", "body": "Salam chetori?"}

۵- کلاینت طرف مقابل، همین عبارت را چاپ می‌کند.

### سناریوی دریافت لیست چتها

۱- دستور

get-chats

۲- پاسخ فضای کار:

OK [{"name": "saeed", "unread\_count": 2}, ...]

پاسخ یک JSON Array شامل چت‌های این کاربر با کاربران دیگر و تعداد پیام‌های خوانده‌نشده است.

### سناریوی دریافت گفتگو با کاربر دیگر

۱- دستور

get-messages saeed

۲- فضای کار، گفتگو را به عنوان خوانده‌شده علامت می‌زند و پاسخ می‌دهد:

OK [{"seq": 1, "from": "ahmad", "type": "text", "body": "Salam chetori?"}, ...]

پاسخ یک JSON Array شامل تمام پیام‌های ردوبدل شده بین این دو کاربر است.

## فاز دوم: ذخیره پیام‌ها

در صورتی که دستور shutdown را در سرور مرکزی یا میزبان تایپ کنیم، همه داده‌ها را در یک فایل با آدرس مشخص ذخیره کرده و بسته می‌شود.

در هنگام راه‌اندازی سرور مرکزی یا میزبان، اگر فایل مذکور وجود داشته باشد، آن را بارگیری می‌کند.

## فاز سوم: انواع بسته‌ها و پیام‌ها

امکان ویرایش پیام

امکان ارسال استیکر

## فاز چهارم: گروه

در این فاز، نوع جدیدی از گفتگو به نام گروه ایجاد می‌شود که دارای یک نام است. نام گروه‌ها یکتا بوده و با نام کاربری کاربران نیز اشتراک ندارد. دستورات فاز اول (مانند get-chats و get-messages و send-message) به نحوی اصلاح می‌شوند که بتوان ارسال گروهی نیز انجام داد.

ایجاد گروه

عضویت در گروه

افزودن عضو به گروه

این کار دو شرط دارد: اول اینکه کاربر عضو گروه باشد، دوم اینکه با کاربر دیگر دارای گفتگو باشد.