



پروژه مبانی برنامه سازی

پاییز 99-00

دانشکده مهندسی کامپیوتر دانشگاه صنعتی شریف

فاز 1: تویتر (کلاینت)

مسئولین فاز یک :

سارا خسروی، امیر صدرا عبدالحی، محمدصادق مجیدی یزدی،

امیرمهدی کوششی، علیرضا بابازاده

فهرست مطالب

2.....	مقدمه
3.....	منوی برنامه
14.....	توکن
15.....	ارسال پیام به سرور
20.....	تازه سازی
22.....	راه اندازی سرور
23.....	نحوه اتصال و ارسال درخواست به سرور

1. مقدمه

توییت، اپلیکیشنی است که در آن افراد می‌توانند پیام‌های کوتاهی را در صفحه‌ی شخصی خود به اشتراک بگذارند و نظرات بقیه را درباره آن بفهمند و یا نظرات بقیه را ببینند و آنها را پسندند. در واقع توییت به اصطلاح یک میکرو بلاگ است. یک وبلاگ کوچک برای به اشتراک گذاری نظرات درباره وقایع روزمره یا خبرهایی از اتفاقات پیش آمده و ... که افراد به واسطه‌ی این نظرات با یکدیگر ارتباط برقرار می‌کنند.

برای آشنایی بیشتر با میکرو بلاگ از [اینجا](#) و [اینجا](#) استفاده کنید. سعی کنید قبل از ادامه این مستند، با این مفهوم آشنا شوید.

هدف این پروژه، طراحی میکرو بلاگی شبیه توییت است.

اگر تا حالا تجربه کار با اپلیکیشن توییت را نداشته‌اید، توصیه می‌کنیم از [این طریق](#) وارد توییت شوید و گزینه‌های مختلف آن را تست کنید. این کار برای درک اصطلاحات به کار برده شده در این پروژه به شما کمک می‌کند. هرچند این اصطلاحات تا حد امکان توضیح داده شده است.

توییت در بستری ارائه می‌شود که به آن معماری **کلاینت / سرور** می‌گویند.

برای توضیح این معماری به مثال جالب زیر دقت کنید:

یک مشتری را در نظر بگیرید که به رستورانی رفته است. این مشتری یک سری درخواست دارد که پاسخ به این درخواست ها از طرف آشپزخانه رستوران به فرد داده می‌شود. لذا مشتری این درخواست ها را به آشپزخانه می‌رساند و پاسخ را نیز دریافت می‌کند. در این صورت مشتری را که درخواست می‌دهد، **کلاینت** و آشپزخانه را که پاسخگوی نیاز مشتری است، **سرور** مینامیم.

حال به مثال جالب تر زیر توجه فرمایید:

در گوگل، وقتی شما به عنوان **کلاینت** چیزی را در کادر مربوطه می‌نویسید و دکمه search را می‌فشارید، در واقع یک درخواست برای **سرور** شرکت گوگل ارسال می‌کنید و گوگل نیز بعد از پیدا کردن پاسخ شما، نتایج را برایتان ارسال می‌کند. در این حالت، یک ارتباط بین شما (کلاینت) و گوگل (سرور) برقرار می‌شود.

توجهتان را به آخرین مثال نیز جلب می‌کنیم:

در اپلیکیشنی مانند توییت، زمانی که شما (کلاینت) توییتی (در ادامه با این مفهوم آشنا می‌شوید) را پست می‌کنید، شما یک درخواست به سرور توییت می‌فرستید که آن متن را برای شما در تایم لاین (با

این مفهوم نیز آشنا خواهید شد) قرار دهد، سرور توییتر نیز پس از بررسی درخواست شما، پیام را در تایم لاین قرار می دهد.

مثال های بالا مواردی برای درک مفاهیم کلاینت و سرور بود. برای درک بهتر به لینک های زیر مراجعه کنید. اگر باز هم کم و کاستی در درک این موضوعات داشتید، سرچ کنید!!

[لینک 1](#)

[لینک 2](#)

حال، پیاده سازی کلاینت توییتر در این فاز بر عهده شماست! در فاز های بعدی قسمت های دیگر توییتر را پیاده سازی خواهید کرد.

2. منوی برنامه

ابتدا باید این را بدانید که این پروژه، پروژه شماست، شما میتوانید هر طور که تمایل دارید و سلیقه ی شماست، آن را شخصی سازی کنید، اما باید ارتباط با سرور را حفظ کنید و پیامی که به سرور می فرستید مطابق توضیحات باشد.

برای هر کدام از دستورات، مورد پیشنهادی ساده ای برای فهم بهتر شما از آن دستور، تدارک دیده شده است؛ توجه داشته باشید که هیچ الزامی در استفاده از آن ندارید و حتی توصیه می کنیم که به طور دلخواه آن را طراحی کنید.

یک موضوع ساده اما قابل توجه: شما در این برنامه یک برنامه نویس هستید!! شما برنامه ای طراحی می کنید که برای یک سری افراد قابل استفاده باشد. مخاطب برنامه شما، همان کاربر است. به عبارتی شما برنامه را برای کاربر طراحی می کنید. این مسئله را در ضمن ساخت برنامه در نظر داشته باشید تا بتوانید پروژه بهتری ارائه دهید:).

1. منوی کاربری

همانطور که از نام آن پیداست، زمانی که کاربر، برنامه را باز می‌کند (در اینجا وقتی برنامه run می‌شود)، با آن روبه‌رو می‌شود. در این منو، کاربر با دو انتخاب مواجه است:

- ایجاد حساب کاربری (sign up)
- وارد شدن به حساب کاربری خود (log in)

اگر کاربر از قبل، یک حساب کاربری در برنامه شما داشته باشد، نیازی به ساخت حساب کاربری جدید نیست و می‌تواند به برنامه وارد شود (log in). اگر حساب کاربری نداشته باشد، لازم است که آن را ایجاد کند (sign up).

نمونه ای از منوی کاربری را در زیر مشاهده می‌کنید:

```
Register menu:
1.Sign up
2.Log in
```

1.1. ساخت حساب جدید (sign-up)

این مرحله، برای ساختن حساب کاربری جدید است.

کاربر ابتدا گزینه مربوط به ساخت حساب کاربری را وارد می‌کند:

(که در مثال پیشنهادی ما مورد اول بود؛ شما می‌توانید به شیوه خود عمل کنید.)

1

برای ساختن حساب کاربری، کاربر باید اطلاعاتی را در اختیار برنامه قرار دهد. لازم است که دو مولفه username و password توسط برنامه از کاربر دریافت شود.

(شکل زیر ادامه همان روند پیشنهادی است؛ ما ابتدا به کاربر گفتیم که username خود را وارد کند. بعد از آن، از کاربر مقدار password را دریافت کردیم. شما می‌توانید روش دیگری به کار ببرید.)

```
Username:
AmirMahdi
Password:
abc123
```

افراد در این برنامه باید username منحصر به فردی داشته باشند و هیچ دو کاربری، username یکسان ندارند. پس حالا که برنامه این دو مقدار را دریافت کرده است، لازم است بررسی کند که کاربر دیگری با این مشخصه وجود دارد یا نه. **این بررسی در این فاز به عهده شما نیست.** این کار را سرور برای

شما انجام خواهد داد. **سروری که در این فاز به آن نیاز دارید در اختیار شما قرار خواهد گرفت و شیوه استفاده از آن نیز در ادامه مستند برایتان شرح داده خواهد شد.** کاری که لازم است انجام دهید این است که این مشخصات را برای سرور ارسال کنید. در بخش **ارسال پیام به سرور**، به چگونگی این فرآیند خواهیم پرداخت. بعد از آنکه این مشخصات را برای سرور ارسال کردیم، سرور نیز به شما پاسخی ارسال خواهد کرد و وضعیت درخواست شما (بررسی موجود بودن چنین کاربری با مشخصات داده شده) را در این پیام بیان می‌کند. اگر کاربری از قبل با این نام وجود داشته باشد، سرور یک پیام خطا **برمیگرداند**، در غیر اینصورت، یک پیام اجرای موفق **برمیگرداند**. (چگونگی پیام های ارسال شده از طرف سرور نیز در همان بخش بیان شده است.)

اگر ساخت حساب با موفقیت انجام شود، کاربر باید با اکانت ساخته شده log in کند. اگر عملیات ساخت حساب با خطا روبه‌رو شود، حساب کاربری ایجاد نمی‌شود و برنامه به ابتدای کار برمی‌گردد و منتظر می‌ماند تا دوباره کاربر درخواست خود را وارد کند.

برای وارد شدن به حساب کاربری، همانند فرایند sign up، این بار کاربر به قسمت log in رفته و با وارد کردن username و password خود، وارد حساب کاربری می‌شود.

اگر log in با موفقیت انجام شود، سرور یک auth token **میسازد** (نگران نباشید، درباره auth token پایین تر توضیح مفصل داده شده است) و به کلاینت **برمیگرداند** و سپس کلاینت وارد منوی اصلی میشود. اگر نام کاربری وجود نداشت و یا رمز اشتباه بود، خطای مربوطه از طرف **سرور** برگردانده میشود و **کلاینت** باید در Register menu بماند. (در واقع کاربر باید از اول log in کند و اگر به خطایی نخورد وارد حساب خود میشود.)

2. منوی اصلی

اگر به **توییت** رفته باشید، هنگامی که شما وارد حساب کاربری خود شوید می‌توانید از امکانات آن استفاده کنید، حالا ما سعی کردیم در اینجا برخی از آن امکانات (یا در واقع همان منوها) را به شما شرح دهیم که بتوانید برای کاربر یک برنامه عالی بنویسید. همانطور که **میبینید** در صفحه اصلی 5 منو وجود دارد که به مرور به تشریح هر یک از آنها خواهیم پرداخت.

1. Timeline
2. Search
3. Tweet Profile
4. Personal area
5. Log out

2.1. تایم لاین (TimeLine)

تایم لاین قسمتی از توییتر است که در آن جا توییت هایی که افراد می زنند به نمایش در می آید. در واقع هنگامی که افراد توییتی را ارسال می کنند (نحوه ارسال توییت کامل توضیح داده شده است) این توییت ها در منویی به اسم TimeLine به افرادی که آن کاربر را دنبال می کنند به نمایش در خواهند آمد.

برای ورود به تایم لاین، ابتدا شماره آن را وارد میکنید.

(همانطور که دیدید در صفحه اصلی پیشنهادی ما منوی تایم لاین منوی اول بود به همین دلیل ابتدا شماره 1 را وارد میکنیم.)

1

سپس بعد وارد کردن شماره از طرف برنامه برای کاربر دو گزینه به نمایش در خواهد آمد:

```
1.send tweet
2.refresh
```

در واقع هنگامی که کاربر به تایم لاین رود در آنجا امکان دیدن توییت های بقیه افراد یا ارسال توییت از طرف خود کاربر را خواهد داشت. حال به شرح این 2 گزینه خواهیم پرداخت.

2.1.1. ارسال توییت

برای ارسال توییت، ابتدا عدد 1 را وارد می کنید (در منویی که میبینید ارسال توییت در اول منو قرار دارد که شما به دلخواه خود میتوانید جایگاه آن را تعیین کنید.) و سپس متن مورد نظرتان را می نویسید.

1

حال کاربر بعد از زدن گزینه 1، توییت خود را وارد میکند.

```
Hi! This is my first tweet on twitter!
```

سپس این توییت توسط برنامه به سرور ارسال خواهد شد. (با نحوه ارسال توییت به سرور آشنا خواهید شد عجله نکنید:)

و سپس بعد از فرستادن توییت توسط کاربر دوباره منوی تایم لاین ظاهر میشود.

```
1.send tweet
2.refresh
```

2.1.2. تازه سازی (Refresh)

با انتخاب این دستور و ارسال ریکوئست (درخواست) آن به سرور، سرور لیستی از توییت هایی که از زمان آخرین refresh شما تا الان توسط followers ها و following هاتون زده شده است را به شما به عنوان پاسخ برمیگرداند.

در واقع همانطور که گفته شد کاربر برای دیدن توییت سایر افراد، باید گزینه Refresh را انتخاب کند تا توییت ها برای وی به نمایش دربیاید. شما میتوانید توییت ها را هرگونه که دوست دارید به کاربر نشان دهید. (طرح پیشنهادی ما برای نشان دادن توییت ها بدین شکل است). در اینجا نمونه ای توییت ها در تایم لاین را مشاهده میکنید.

```
a.kousheshi 1298
Hi! This is my first tweet on twitter!
Likes:2 comments:0

cnn 29876
Iran's giant ocean-going ship
Likes:877 comments:457
```

همانطور که میبینید، ابتدا نام کاربری کسی آن توییت را زده میبینیم و سپس **شماره توییت** را مشاهده میکنیم (دقت کنید که هر توییت یک شماره مخصوص به خود دارد و شماره هیچ دو توییتی یکسان نیست. در واقع یعنی شماره هر توییت یکتا است.) این شماره برای آن است که اگر کاربر خواست توییتی را لایک کند یا برای آن کامنتی بگذارد، با استفاده از این شماره به راحتی کار خود را انجام دهد. سپس در خط بعد از آن متن توییت را مشاهده میکنید و در زیر آن نیز تعداد لایک ها و کامنت ها را مشاهده میکنید.

همانطور که گفته شد، شما هرگونه که بخواهید میتوانید این توییت ها را در تایم لاین به نمایش درآورید، اما دقت کنید که به گونه ای باشد که کاربر بتواند تعداد لایک ها و کامنت ها را ببیند و به راحتی توسط شماره هر توییت، برای آن توییت کامنتی بگذارد یا آن را لایک کند. (نحوه لایک کردن و کامنت گذاشتن نیز توضیح داده شده است.)

بدین گونه کاربر با زدن refresh می تواند به راحتی توییت های همه ی افرادی را که دنبال کرده است، یا توییت های همه ی افرادی که آن کاربر را دنبال کرده اند را ببیند.

هنگامی که کاربر گزینه refresh را زد و توییت ها برای وی ظاهر شد، کاربر میتواند توییتی را لایک کند یا برای آن کامنت بگذارد. شما باید به گونه ای برنامه بنویسید که این قابلیت را برای کاربر مهیا سازید، ما نیز به توضیح این کار با یک طرح پیشنهادی خواهیم پرداخت.

هنگامی که تمامی توییت ها ظاهر شد، شما باید دو قابلیت برای کاربر مهیا سازید:

1- کاربر بتواند یک توییت را انتخاب و آن را لایک کند یا کامنت بگذارد.

2- به منوی قبل بازگردد. (همان refresh و send tweet)

یک طرح پیشنهادی برای این کار ما به شما در زیر نمایش داده ایم، شما در عکس زیر توییت ها و در انتها نیز دو گزینه برای رفتن به منوی قبل یا لایک کردن و کامنت گذاشتن برای توییتی را میبینید.

(طرح پیشنهادی ما به شکل زیر است.)

```
a.kousheshi 1298
Hi! This is my first tweet on twitter!
Likes:2 comments:0

cnn 29876
Iranian Giant ocean-going ship
Likes:877 comments:457

1.Like or comment a tweet!
2.Back
```

حالا به لایک کردن و کامنت گذاشتن می پردازیم. برای لایک کردن یا کامنت گذاشتن طبق عکسی که در بالا آمده بود، ابتدا گزینه آن را انتخاب میکنیم. (در مثال پیشنهادی ما گزینه 1 بود.)

1

پس از زدن این گزینه، قابلیت های لایک کردن یا کامنت گذاشتن برای کاربر نشان داده میشود.

```
1.Like
2.Comment
3.Back
```

سپس کاربر با انتخاب هر گزینه و نوشتن **شماره تویییت** (همان شماره مخصوص به هر تویییت که در بالا توضیح داده شد)، می تواند آن را لایک کند یا برای آن تویییت کامنتی بگذارد.

نمونه ای را در زیر مشاهده کنید.

فرض کنید کاربر می خواهد یک تویییتی را لایک کند، طبق مثال پیشنهادی، ابتدا گزینه مربوط به لایک را انتخاب می کند (در مثال ما گزینه 1 برای لایک کردن است)

1

و سپس با نوشتن شماره تویییت، آن تویییت توسط کاربر لایک می شود. **دقت کنید که این عمل، یک درخواست به سرور نیز دارد.** یعنی ما بعد از این عملیات، یک درخواست به سرور میفرستیم و میگوییم لایک های تویییت مشخص شده را یک واحد افزایش بده. چگونگی آن را در بخش مربوطه خواهید دید. (حال طبق عکسی که در بالا داده شده بود برای نمایش تویییت ها در تایم لاین، همانگونه که دیدید تویییت مربوط به کاربری با یوزر نیم cnn، شماره تویییت آن 29876 بود، پس کاربر نیز با نوشتن این شماره تویییت، میتواند آن تویییت را لایک کند.)

29876

پس از این کار، تعداد لایک های آن تویییت، یک عدد افزایش میابد.

پس همانطور که دیدید، کاربر با انتخاب گزینه Like و سپس با وارد کردن شماره تویییتی که می خواهد آن را لایک کند، این کار را انجام داد. حالا به گذاشتن یک کامنت توسط کاربر برای تویییتی میپردازیم.

کاربر برای گذاشتن کامنت، ابتدا گزینه مربوط به آن را انتخاب می کند. (همانطور که دیدید طبق مثال ما گزینه مربوط به کامنت گذاشتن گزینه دوم بود.)

2

حالا با انتخاب کردن شماره تویییتی که کاربر می خواهد برای آن کامنت بگذارد، و سپس با وارد کردن متن کامنت، کاربر یک کامنت برای آن تویییت خواهد گذاشت، در زیر نحوه انجام پیشنهادی این کار را مشاهده کنید.

29876

Nice!

همانگونه که مشاهده می کنید، کاربر با وارد کردن شماره تویییت (در اینجا شماره 29876 مربوط به تویییت cnn است) و سپس در خط بعد از آن با نوشتن متن کامنت خود، یک کامنت برای آن تویییت می گذارد. اینجا هم مشابه لایک کردن، یک درخواست به سرور داریم.

دقت کنید که بعد از اینکه ما تویییتی را لایک کردیم یا کامنتی گذاشتیم به تعداد لایک یا کامنت آن تویییت اضافه خواهد شد.

با زدن گزینه Back نیز کاربر به همان منوی پیش تر می رود.

2.2. جست و جو (Search)

یکی از قابلیت هایی که توییتر دارد search است که شما قرار است این قابلیت را به صورت ساده پیاده سازی کنید.

توییتر و اکثر شبکه های اجتماعی search بسیار قوی و کارآمدی دارند به طوری که با وارد کردن حتی بخشی از نام کاربری شخص مورد نظر، لیستی از افراد با نام کاربری شبیه به آن را می آورد. اما ما از شما انتظار چنین چیزی را در ترم یک نداریم.

پس به صورت خیلی ساده با دریافت یک نام کاربری، تعیین میکنید که فردی با این نام کاربری در بین کاربرهای ثبت نام کرده، وجود دارد یا خیر؟

اگر چنین کاربری وجود داشت، مستقیماً باید وارد صفحه کاربری فرد بشود. در صورتی هم که جست و جو حاصلی در بر نداشت پیغام خطای مناسبی از سرور به شما بازگردانده میشود.

اگر کاربر مورد نظری را که سرچ کردید در توییتر حساب کاربری داشت، بلافاصله پس از سرچ کردن اسم کاربر به صفحه کاربر منتقل می شوید. در صفحه کاربر شما با موارد زیر سروکار دارید:

1. نام کاربری
2. بیوگرافی
3. تعداد following و followers
4. وضعیت دنبال کردن کاربر
5. تمام تویییت های کاربر

نمونه ای را در زیر میبینید:

ابتدا آیدی فرد مورد نظر را وارد میکنیم.

پس از اینکه نام کاربری را به سمت سرور ارسال کردید، پاسخی را مبنی بر اینکه این کاربر وجود دارد یا خیر، دریافت میکنید.

اگر فرد مورد نظر وجود داشت باید اطلاعاتی که ذکر شد را به نمایش در بیاورید.

می توانید چیداش اطلاعات را به شکل زیر در بیاورید. (هر نوعی از نمایش اطلاعات مورد قبول است)

```
Username: a.kousheshi
bio: something about me
followers: 11 followings: 10
unfollow

I like this day!
Likes: 2 Comments: 0
```

در اینجا با دیدن کلمه unfollow به این پی میبریم که ما کاربر را از قبل follow کرده بودیم، در اینجا با نوشتن کلمه unfollow میتوانیم از دنبال کردن کاربر دست بکشیم:

```
unfollow
```

پس از وارد کردن این دستور همچنان در پروفایل کاربر باقی می ماند و حتی نیازی به نمایش دوباره اطلاعات (برای اصلاح وضعیت دنبال کردن کاربر) نیست، ولی مسلماً از دفعات بعدی پس از وارد شدن به پروفایل کاربری این فرد، باید اطلاعات درست و اصلاح شده، نشان داده شود. (دقت کنید که شما باید برای اصلاح وضعیت این کاربر در سرور، درخواست مناسب را به سرور ارسال کنید که شیوه ارسال آن در کنار سایر درخواست ها شرح داده خواهد شد.)

در واقع اگر دوباره کاربر را سرچ کنیم خواهیم دید:

```
Username: a.kousheshi
bio: something about me
followers: 11 followings: 10
follow

I like this day!
Likes: 2 Comments: 0
```

و در اینجا با نوشتن کلمه follow میتوانیم کاربر را دنبال کنیم. (فرآیند دقیقاً شبیه به unfollow است که توضیح داده شده است)

2.3. تویییت پروفایل (Tweet profile)

در این بخش باید profile تویییت را پیاده سازی کنید.

اگر کاربر تویییت باشید، حتماً برایتان پیش آمده که از سر بیکاری، گاهی اوقات به سراغ تویییت های خود رفته و مشغول خواندن آنها می شوید. تویییت این قابلیت را در profile tweets قرار داده است. تویییت شما نیز باید همچین ویژگی داشته باشد ولی با تشکیلات کمتر!

قسمت تویییت پروفایل شما باید ترکیبی از بخش های مختلف profile در تویییت باشد به طوری که این صفحه باید شامل:

1. نام کاربری
2. بیوگرافی
3. تعداد followers و following
4. تمام تویییت های کاربر باشد. ترتیب نمایش این اطلاعات را برای زیبایی و خوانایی صفحه رعایت کنید.

نمونه ای از تویییت پروفایل:

```
Username: AmirMahdi
bio: something about me
followers: 15   followings: 12

I like this day!
Likes: 1 Comments: 2
```

2.4. صفحه شخصی (Personal area)

هنگامی که کاربر به صفحه شخصی وارد میشود، منویی مانند تصویر زیر به روی او باز خواهد شد: (ترتیب نمایش منوها کاملاً به دلخواه شماست)

```
1.set bio
2.change password
```

2.4.1. ست کردن بیوگرافی

در این قسمت کاربر باید بتواند بیوگرافی خود را ست کند و توجه داشته باشید که قابلیت تعویض بیو برای کاربر وجود ندارد و فقط می تواند یک بار برای همیشه آن را ست کند. پیاده سازی این قسمت و چاپ پیغام های خطا (در صورت نیاز) به دلخواه خودتان است.

با توجه به منوی نشان داده شده، با وارد کردن گزینه 1 باید بتوانید بیوگرافی خود را ست کنید:

1

پس از آن اگر قبلا برای خود بیوگرافی قرار داده اید، باید پیغام خطایی به دلخواه خود مبنی بر آن چاپ کنید.

در غیر این صورت بیوی مورد نظر را وارد میکنیم.

```
nothing about me
```

و سپس به منوی قبل باز می گردید.

```
1.set bio
2.change password
```

2.4.2. تغییر رمز عبور (امتیازی)

حتماً برایتان پیش آمده که گاهی امنیت خود را در شبکه های اجتماعی، بر باد هوا دیده و تصمیم به تعویض رمز عبور خود کرده اید تا حداقل های امنیت را رعایت کرده باشید.

پس در توییت شما هم هر بار که کاربری احساس ترس کرد، باید بتواند رمز عبور خود را تغییر دهد.

و لازم به گفتن نیست که لازمه تعویض رمز عبور، دانستن رمز عبور قبلی است. به این صورت که شما دو ورودی از کاربر می گیرید، یکی رمز عبور قبلی و دیگری رمز عبور جدید که اگر رشته وارد شده تحت عنوان رمز عبور قبلی صحت داشت، آنگاه می توانید رمز عبور را تغییر دهید، وگرنه به دلخواه یک پیغام خطا چاپ کنید.

با توجه به منوی مثال زده شده، عدد 2 را وارد میکنیم:

2

و سپس در خط اول رمز فعلی و در خط بعدی رمز جدید را وارد میکنیم:

```
abc123
amirmahdi1234
```

پس از آن که رمز عبور قبلی را به سمت سرور فرستادید، اگر که رمز وارد شده توسط کاربر صحت نداشته باشد، از سرور پیغام خطا دریافت می کنید که باید خطایی مبنی بر آن را به کاربر نمایش دهید. در غیر این صورت می توانید رمز کاربر را تغییر دهید.

3. توکن (Auth Token)

توکن چیست؟ توکن یک عبارت یا یک رمز برای انحصار ارتباط بین کلاینت و سرور است. در واقع برای اینکه سرور بفهمد شما همانی هستید که باید باشید، از شما توکن میخواهد.

به عنوان مثال، اگر دانشگاه را سروری بگیرید که به ریکوئست های شما پاسخ میدهد، برای اینکه بفهمد شما کی هستید و کار های شما را درست انجام بدهد یا به عبارت دیگر شما را درست شناسایی کند، از شما، شماره دانشجویی میخواهد، در واقع شماره دانشجویی شما همان Token است که برای هر درخواستی از دانشگاه دارید باید به آن ارائه دهید تا پاسخ درخواست خود را دریافت کنید.

از آنجایی که قرار است شما پیوسته به سرور ریکوئست دهید، لذا لازم است که سرور شما را به ازای هر ریکوئست تایید کند و بتواند برایتان پاسخ را ارسال کند. پس این ارتباط نیازمند این است که چیزی بین شما و سرور به صورت منحصر به فرد باشد. چنین چیزی را **توکن** یا **authentication token** می نامیم.

توکن پس از لاگین برای شما ارسال میشود و باید آن را جایی ذخیره کنید و در هر درخواست برای سرور ارسال کنید. پیاده سازی شما باید به گونه ای باشد که پس از هر بار login توکن را عوض کند، زیرا توکن شما در سرور عوض شده و سرور توکن جدیدی را برای شما ارسال میکند.

همانطور که پیش تر نیز به آن اشاره شد، در فاز اول فقط پیاده سازی کلاینت بر عهده شماست. حال برای آزمایش کلاینت خود، از اتصال به سرور تا ارسال درخواست به آن و دریافت پاسخ از آن، سروری متناسب با فرمت ریکوئست های این پروژه در اختیارتان قرار میدهیم. برای راه اندازی و استفاده از آن، این بخش را دنبال کنید.

4. ارسال پیام به سرور

همانطور که در بالا دیدید، هر کاری که برنامه شما برای کاربر انجام میدهد، مانند جست و جو، لایک کردن و کامنت گذاشتن، دنبال کردن افراد، ست کردن بیوگرافی و ...، برنامه شما این کارها را با فرستادن درخواست به سرور انجام می دهد. حال برای این که با سرور ارتباط برقرار کنید و درخواست های خود را به آن بفرستید، باید پیام هایتان را در فرمت های خاصی باشند. **(این فرمت لازم است به همانگونه که اینجا شرح می دهیم پیاده سازی شود و نباید تغییری در این فرمت لحاظ کنید.)**

هر درخواستی که شما از کاربر می گیرید، باید آن را به یک فرمت خاص تبدیل کنید.

این فرمت خاص، یک ریکوئست قابل فهم برای سرور است که میتوانید آن را برای سرور ارسال کنید. در واقع این ریکوئست ها، استاندارد هایی است که بین کلاینت و سرور قرار داده شده است. این ریکوئست ها در واقع یک رشته (string) هستند و شما این رشته ها را به سرور میفرستید. ریکوئست ها به فرمت های زیر تعریف شده اند:

دقت کنید: در پایان هر درخواست یک "\n" بگذارید.

4.1. ثبت کاربر جدید

"signup <username>, <password>"

این فرمت پیامی است که باید برای ثبت کاربر جدید به سرور ارسال کنید. یعنی هنگامی که در Register menu کاربر اقدام به ساخت یک حساب کاربری کرد، شما پس از دریافت username و password از کاربر، این پیامی به این فرمت به سرور ارسال می کنید.

مثال:

signup AmirMahdi, abc123

دقت کنید: بعد از signup و username یک فاصله است، بعد از username بدون هیچ فاصله ای یک کاما می آید (,) و سپس بعد از یک فاصله، password می آید.

پاسخ سرور:

```
{"type": "Successful", "message": ""}
```


این پیام به این معنی است که حساب کاربری با موفقیت ساخته شده است، و کاربر میتواند Log in کند. اما اگر این Username از قبل توسط یک فرد دیگری استفاده می شده است، سرور برای شما یک پیغام خطا می فرستد و شما آن پیغام را به کاربر نمایش می دهید.

پاسخ سرور به هنگام خطا:

```
{"type":"Error","message":"This username is already taken."}
```

در اینجا خطایی داریم که می گوید این نام کاربری از قبل مورد استفاده واقع شده است.

4.2. ورود کاربر به حساب خود

"login <username>, <password>"

اگر این ارسال خطایی نداشته باشد، سرور برای شما یک پیام ارسال میکند که حاوی یک توکن است. این توکن برای اطمینان سرور است که شما همانی هستید که باید باشید. پس لازم است در تمام درخواست های بعدی، توکن را نیز به عنوان بخشی از ریکوئست طبق فرمت گفته شده برای سرور ارسال کنید.

مثال:

login AmirMahdi, abc123

دقت کنید: بعد از signup و username یک فاصله است، بعد از username بدون هیچ فاصله ای یک کاما می آید (,) و سپس بعد از یک فاصله، password می آید.

اگر رمز عبور و یا نام کاربری درست باشد، سرور برای شما یک توکن میفرستد که برای آن کاربر است، در غیر این صورت سرور یک پیغام خطا به شما ارسال می کند.

پاسخ سرور:

```
{"type":"Token","message":"jmgA6quS71Pn9kbfcjuUs0oo5LSGM32g"}
```

عبارت **jmgA6quS71Pn9kbfcjuUs0oo5LSGM32g** توکن شماست.

پاسخ سرور هنگام خطا:

```
{"type":"Error","message":"Incorrect password."}
```

به عنوان مثال اینجا خطای غلط بودن رمز عبور را داریم.

4.3. ارسال توییت

"send tweet <Token>, <tweet>"

مثال:

send tweet jmgA6quS71Pn9kbfcjuUs0oo5ISGM32g, Hi! This is my first tweet on twitter!

دقت کنید: بعد از send tweet و Token یک فاصله است، بعد از Token بدون هیچ فاصله ای یک کاما می آید (,) و سپس بعد از یک فاصله، tweet می آید.

4.4. تازه سازی (Refresh)

"refresh <Token>"

مثال:

refresh jmgA6quS71Pn9kbfcjuUs0oo5ISGM32g

پاسخ سرور برای درخواست **تازه سازی** در یک بخش جداگانه مفصل توضیح داده شده است!

4.5. لایک کردن

"like <Token>, <tweet_id>"

همانطور که گفته شد، هر **توییت** یک آیدی منحصر به فرد دارد که در اینجا همان **tweet_id** است که مشاهده میکنید.

دقت کنید: بعد از like و Token یک فاصله است، بعد از Token بدون هیچ فاصله ای یک کاما می آید (,) و سپس بعد از یک فاصله، tweet_id می آید.

مثال:

like jmgA6quS71Pn9kbfcjuUs0oo5ISGM32g, 1298

4.6. کامنت گذاشتن

"comment <Token>, <tweet_id>, <message>"

مثال:

comment jmgA6quS71Pn9kbfcjuUs0oo5ISGM32g, 1298, nice!

دقت کنید: بعد از comment و Token یک فاصله است، بعد از Token بدون هیچ فاصله ای یک کاما می آید (,) و سپس بعد از یک فاصله، tweet_id، بعد از tweet_id بدون هیچ فاصله ای یک کاما (,) می آید و سپس بعد از یک فاصله، message می آید.

4.7. جست و جو

"search <Token>, <username>"

مثال:

search jmgA6quS71Pn9kbfcjuUs0oo5ISGM32g, a.kousheshi

دقت کنید: بعد از search و Token یک فاصله است، بعد از Token بدون هیچ فاصله ای یک کاما می آید (,) و سپس بعد از یک فاصله، username می آید.

پاسخ سرور:

```
{"type":"Profile", "message":{"username":"a.kousheshi","bio":"something about me", "numberOfFollowers":1,"numberOfFollowings":0,"followStatus":"Followed","allTweets":[]}}
```

دقت کنید که در اینجا این کاربر هیچ تویییتی ندارد.

4.8. فالو و آنفالو (Follow and Unfollow)

"follow <Token>, <username>"

مثال:

Follow jmgA6quS71Pn9kbfcjuUs0oo5ISGM32g, a.kousheshi

"unfollow <Token>, <username>"

مثال:

unfollow jmgA6quS71Pn9kbfcjuUs0oo5ISGM32g, a.kousheshi

4.9. ست کردن بیوگرافی

"set bio <Token>, <bio>"

مثال:

set bio jmgA6quS71Pn9kbfcjuUs0oo5ISGM32g, nothing about me

دقت کنید: بعد از set bio و Token یک فاصله است، بعد از Token بدون هیچ فاصله ای یک کاما می آید (,) و سپس بعد از یک فاصله، bio می آید.

4.10. خارج شدن از حساب کاربری (Log out)

"logout <Token>"

مثال:

logout jmgA6quS71Pn9kbfcjuUs0oo5ISGM32g

4.11. توییت پروفایل

"profile <Token>"

پاسخ سرور این ریکوئست نیز مانند پاسخ سرچ کردن است.

4.12. عوض کردن رمز عبور (امتیازی)

"Change password <Token>, <current_pass>, <new_pass>"

مثال:

Change password jmgA6quS71Pn9kbfcjuUs0oo5ISGM32g, abc123,
AmirMahdi1234

در اینجا نیز اگر رمز عبور فعلی کاربر درست نباشد **سرور** یک پیغام خطا خواهد فرستاد.

4.13. دیدن کامنت ها (امتیازی)

برای دیدن کامنت ها نیازی به ریکوئست جداگانه نیست.

همانطور که برای لایک کردن یا کامنت گذاشتن توضیح داده شده بود، شما میتوانید یک گزینه سومی نیز بزرارید به عنوان see comments، با رفتن به این بخش کاربر میتواند همه کامنت هایی که کاربران برای یک توییت گذاشته اند را ببیند، بعد از خواندن بخش **تازه سازی** کامل متوجه خواهید شد که کامنت ها به چه صورت برای هر توییت خواهد آمد.

5. تازه سازی (Refresh)

این ویژگی کاربر را قادر می سازد تا توییت هایی که followers یا followings های کاربر ارسال میکنند را بتواند ببیند. حالا همانطور که گفته شده بود توضیحات مربوط به تازه سازی را در این جا خواهید خواند.

بعد از اینکه شما ریکوئست این دستور را به سرور ارسال کردید، از طرف سرور به شما پاسخی ارسال میشود که لیست توییت هایی است که شما باید در تایم لاین خود نشان دهید. دقت کنید که شما میتوانید این توییت ها را به هر فرمتی که میخواهید در کنسول چاپ کنید، نمونه ای از پیاده سازی را نیز در قسمت تازه سازی منوی برنامه مشاهده کردید.

پیام کلاینت به سرور:

refresh jmgA6quS71Pn9kbfcjuUs0oo5ISGM32g

پاسخ سرور:

```
{
  "type": "List",
  "message": [
    {
      "id": 1,
      "author": "user",
      "content": "hello this is my first tweet.",
      "comments": {
        "sadegh": "another comment",
        "amir": "this is a comment"
      },
      "likes": 1
    },
    {
      "id": 2,
      "author": "user",
      "content": "this is another tweet.",
      "comments": {},
      "likes": 0
    }
  ]
}
```

نمونه ای از نمایش توییت ها:

```
user    1
hello! this is my first tweet.
Likes:2 Comments:2

user    2
this is another tweet.
Likes:0 Comments:0

1. Like or comment a tweet
2. Back
```

در واقع شما از پاسخی که سرور به شما داد، می توانید نام کاربری فردی که توییت زده است، متن توییت، شماره توییت، تعداد لایک ها و کامنت های توییت و تمامی کامنت های هر توییت را نیز مشاهده کنید.

برای پیدا کردن تعداد کامنت ها نیز، شما باید از همان پاسخ سرور، تعداد کامنت ها را پیدا کنید. همانطور که برای دیدن کامنت ها توضیحات داده شده بود، طبق عکسی که مشاهده می کنید، برای هر توییت، کامنت های آن نیز آمده است، برای پیاده سازی این بخش، شما باید کامنت های هر توییت را در جایی ذخیره کنید، و هنگامی که کاربر خواست کامنت های آن توییت را مشاهده کند، شما به کاربر نشان دهید. (همانطور که گفته شد زدن این بخش امتیازی است).

برای مثال:

```
user    1
hello! this is my first tweet.
Likes:2 Comments:2
```

```
user    2
this is another tweet.
Likes:0 Comments:0
```

```
1.Like or comment a tweet
2.Back
```

برای لایک کردن یا کامنت گذاشتن برای توییتی می‌توانیم ابتدا گزینه 1 را زده.

1

```
1.Like
2.Comment
3.See comments
4.Back
```

با زدن گزینه مربوط به دیدن کامنت ها، و سپس وارد کردن شماره آن توییت، کامنت های آن توییت به نمایش در خواهد آمد.

ابتدا گزینه مورد نظر را انتخاب می‌کنیم.

3

و سپس شماره توییت را وارد می‌کنیم.

1

حال کامنت ها به نمایش در خواهند آمد.

```
sadegh: another comment
amir: this is a comment
```

در اینجا نیز کامنت را مشاهده میکنید. (میتواند چند تا کامنت نیز باشد).

6. راه اندازی سرور

سرور این برنامه شامل یک فایل `server.jar` می باشد که راه اندازی سرور از طریق این فایل امکان پذیر است. اگانت هایی که می سازید و توییت هایی که می زنید، در پوشه ای به نام `Resources` ذخیره می شوند. این پوشه قبل از راه اندازی سرور وجود خارجی ندارد و پس از اولین راه اندازی توسط سرور ایجاد می شود.

برای اجرای سرور روی لپتاپ خود، نیاز به ابزارهای توسعه جاوا (java) دارید.

ابتدا از طریق [این لینک](#) و از جدول `Java SE Development Kit 11.0.9`، نسخه مربوط به سیستم عامل خود را دریافت و سپس نصب کنید. ممکن است برای دریافت این ابزار نیاز به اتصال به فیلتر شکن داشته باشید! می توانید آموزش ویدئویی نصب جاوا برای سیستم عامل های `Windows` و `Mac` را به ترتیب از [این جا](#) و [این جا](#) ببینید. همچنین کاربران لینوکس می توانند از طریق ترمینال لینوکس و با اجرای دستور `sudo apt-get install openjdk-11-jre`، بدون نیاز به رفتن به سایت و دانلود فایل، ابزار توسعه جاوا را روی سیستم خود داشته باشند. (اگر از توزیع هایی از لینوکس نظیر `Monjro` استفاده می کنید که `apt` ندارند، باید از `packaging tool` مخصوص به آن توزیع استفاده کنید)

بعد از نصب `Java` کاربران لینوکس و مک در `Terminal` و کاربران ویندوز در `Command Prompt` خود می توانند با اجرای `java -version`، از موفقیت آمیز بودن مراحل نصب اطمینان حاصل کنند.

حال با نصب شدن `Java` بر روی سیستم خود، می توانید از سرور استفاده کنید.

```
sadegh@sadegh-ThinkPad-E580:~/Programming/Server$ java -version
openjdk version "11.0.9.1" 2020-11-04
OpenJDK Runtime Environment (build 11.0.9.1+1-Ubuntu-0ubuntu1.20.04)
OpenJDK 64-Bit Server VM (build 11.0.9.1+1-Ubuntu-0ubuntu1.20.04, mixed mode, sharing)
```

برای فعال سازی سرور، ابتدا به پوشه ای که فایل سرور را در آن قرار داده اید بروید. کاربران ویندوز می توانند با دابل کلیک روی فایل `server.jar` سرور را بالا بیاورند.

در همه سیستم عامل ها از جمله ویندوز، می توان با باز کردن `Terminal` و `Command Prompt` در آدرسی که سرور در آن قرار دارد، با اجرای دستور `java -jar server.jar`، سرور را اجرا کرد.

```
sadegh@sadegh-ThinkPad-E580:~/Programming/Server$ java -jar server.jar
Info | 2020/12/23 00:49:58 | Initializing...
Info | 2020/12/23 00:50:00 | Initialized.
Info | 2020/12/23 00:50:00 | Listening on port 12345
```

برای باز کردن Terminal در مکان مورد نظر برای سیستم عامل های Mac و Linux ، با File Explorer به پوشه ی سرور بروید و با راست کلیک روی پوشه و انتخاب گزینه ی open in terminal ، ترمینال را در آن پوشه باز کنید. برای باز کردن Command Prompt در

سیستم عامل ویندوز، ابتدا به پوشه ای که سرور در آن قرار دارد رفته و سپس در نوار بالای پنجره که آدرس پوشه فعلی نوشته شده است، عبارت cmd را تایپ کرده و کلید enter را بزنید. همچنین با cd کردن از آدرس root در Terminal و یا Command Prompt نیز می توانید به پوشه مورد نظر دسترسی پیدا کنید.

توجه کنید بعد از اولین راه اندازی سرور، پوشه ای به نام Resources در کنار فایل server.jar وجود می آید. این پوشه دیتابیس سرور است. به هیچ وجه آن را پاک نکنید زیرا باعث از دست رفتن اطلاعات ذخیره شده می شود.

7. نحوه اتصال و ارسال درخواست به سرور

ابتدا کلاینت اتصال سوکت خود با سرور را برقرار میکند، سپس درخواست خود را برای سرور ارسال میکند، بعد اتصال قطع می شود. یعنی کلاینت برای ارسال درخواست دیگر، باید دوباره ارتباط برقرار کند.

توجه کنید اگر کلاینتی به سرور متصل شود و درخواستی برای آن ارسال نکند، بعد از چند ثانیه سرور خود ارتباط را قطع می کند.

```
F:\Sadegh Majidi>java -jar server.jar
Info | 2020/12/23 01:27:14 | Initializing...
Info | 2020/12/23 01:27:14 | Initialized.
Info | 2020/12/23 01:27:14 | Listening on port 12345
Error | 2020/12/23 01:27:29 | java.net.SocketTimeoutException: Read timed out
    at java.net.SocketInputStream.socketRead0(Native Method)
    at java.net.SocketInputStream.socketRead(Unknown Source)
    at java.net.SocketInputStream.read(Unknown Source)
    at java.net.SocketInputStream.read(Unknown Source)
    at sun.nio.cs.StreamDecoder.readBytes(Unknown Source)
    at sun.nio.cs.StreamDecoder.implRead(Unknown Source)
    at sun.nio.cs.StreamDecoder.read(Unknown Source)
    at java.io.InputStreamReader.read(Unknown Source)
    at java.io.BufferedReader.fill(Unknown Source)
    at java.io.BufferedReader.readLine(Unknown Source)
    at java.io.BufferedReader.readLine(Unknown Source)
    at server.Server.Run(Server.java:45)
    at Main.main(Main.java:11)
```