

**راهنمای نصب و راه‌اندازی**

**سرویس نویسا لایو**

**Nevisa Live Setup**

**اردیبهشت 1404**

# نصب و راه‌اندازی

سامانه نویسا را می‌توان روی یک ماشین مجازی یا ماشین فیزیکی نصب و راه‌اندازی کرد. برای نصب سامانه روی ماشین مجازی از فایل OVF تحویل داده شده می‌توان استفاده کرد. برای نصب سامانه روی ماشین فیزیکی می‌توان از فایل img که تصویر پارتیشن هارد است استفاده کرد.

## حداقل سیستم مورد نیاز

حداقل سخت‌افزار لازم برای نصب سامانه نویسا به صورت جدول زیر است. این مشخصات برای حالت تک کاربره است. به ازای هر هسته پردازشی دو کاربر همزمان می‌توانند به صورت بلادرنگ از سامانه استفاده کنند.

|  |  |
| --- | --- |
| 2 GHz or faster with two or more cores on a compatible 64-bit processor | CPU |
| 16.0 GB | **RAM** |
| 100 GB SSD | **HDD** |

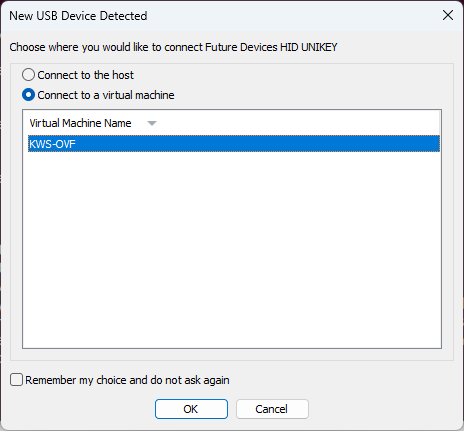
## نصب روی ماشین مجازی

در ادامه مراحل نصب از طریق برنامه VMware Workstation و سرور VMware ESXi توضیح داده می‌شود.

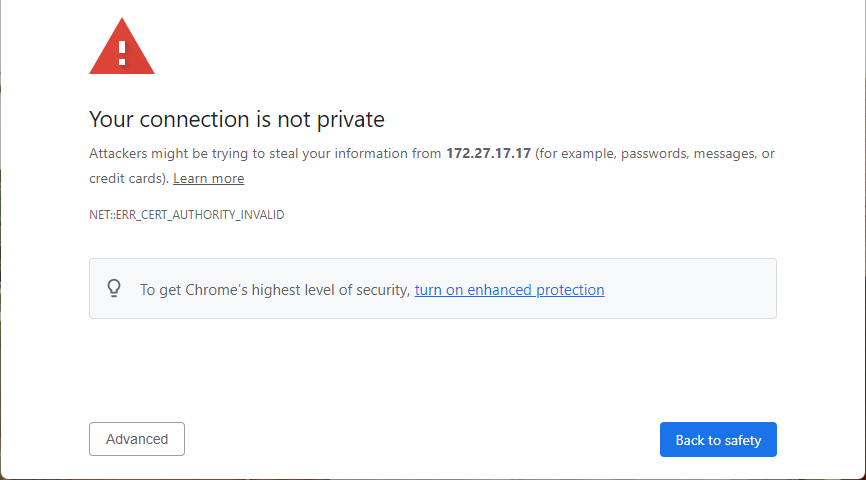
### نصب از طریق برنامه VMware Workstation

مراحل نصب سامانه نویسا از روی فایل OVF به صورت زیر است.

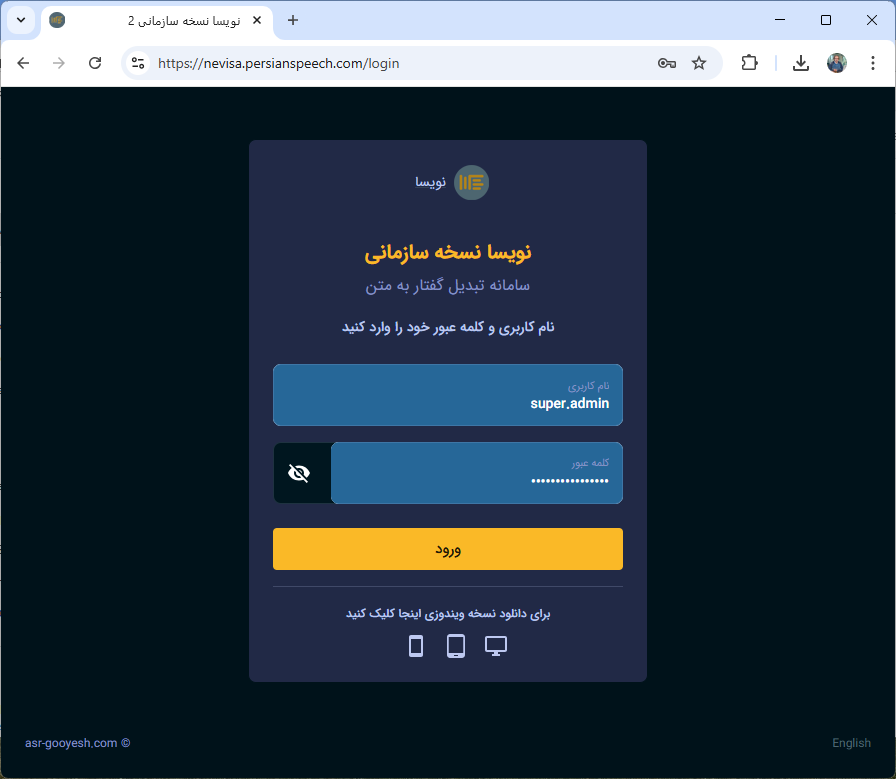
1. ابتدا برنامه VMware Workstation ترجیحاً نسخه 17 به بعد را نصب کنید.
2. در برنامه از طریق منوی فایل و گزینه Open فایل OVF را باز کنید.
3. بعد از ایجاد ماشین مجازی میزان RAM ماشین مجازی را تغییر دهید. (حداقل 7GB حافظه به ماشین مجازی اختصاص داده شود)
4. ماشین مجازی را روشن کنید.
5. قفل سخت‌افزاری را به ماشین مجازی متصل کنید.



1. با استفاده از نام کاربری admin1 و رمز عبور پیش‌فرض 123 وارد سیستم شوید. توصیه می‌شود پس از نصب سامانه رمز ورود را تغییر دهید. برای تغییر رمز از دستور passwd می‌توانید استفاده کنید.
2. در پنجره ترمینال دستور زیر را وارد کنید. با اجرای این دستور آدرس شبکه (IP) ماشین مجازی مشخص می‌شود.
3. یک صفحه مرورگر روی سیستمی در شبکه داخلی که آدرس IP ماشین مجازی را می‌بیند باز کنید و آدرس IP ماشین مجازی را وارد کنید. مرورگر مورد استفاده ترجیحاً Google Chrome یا Microsoft Edge باشد.
4. در صورتی که با پیام Your connection is not private مواجه شدید. گزینه Advanced و سپس Proceed to را بزنید. این پیام به دلیل استفاده از cert فایل‌های openssl نشان داده می‌شود. در صورتی که از cert فایل‌های مورد تایید مرورگر استفاده کنید این پیام نشان داده نمی‌شود.



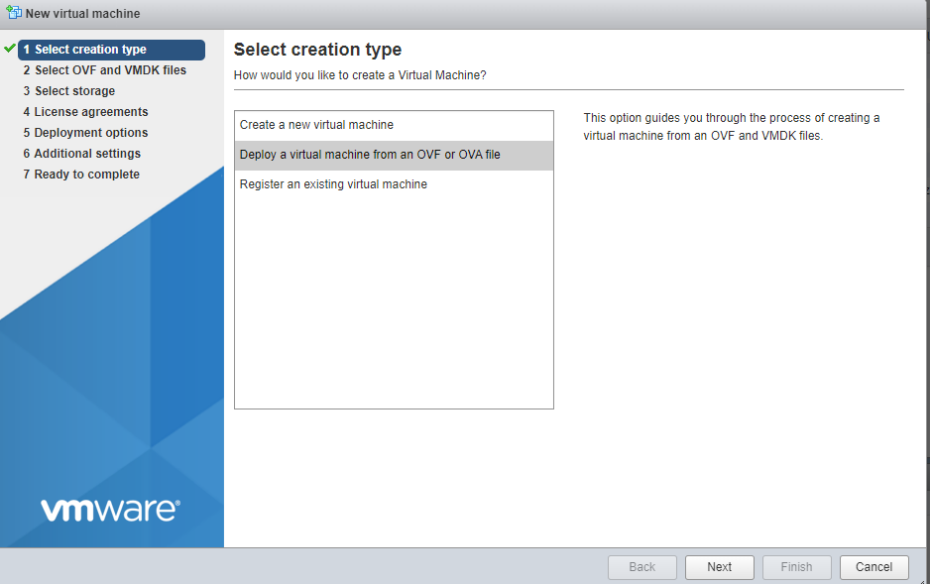
1. نام کاربری مدیر سامانه برای ورود super.admin و رمز پیش فرض مدیر سامانه 123 است. توصیه می‌شود پس از ورود به سامانه، از بخش مدیریت کاربران رمز مدیر را تغییر دهید و برای استفاده دیگر کاربران، حساب کاربری جدید تعریف کنید. توضیحات مربوط به نحوه استفاده از سامانه در راهنمای کاربران آمده است.



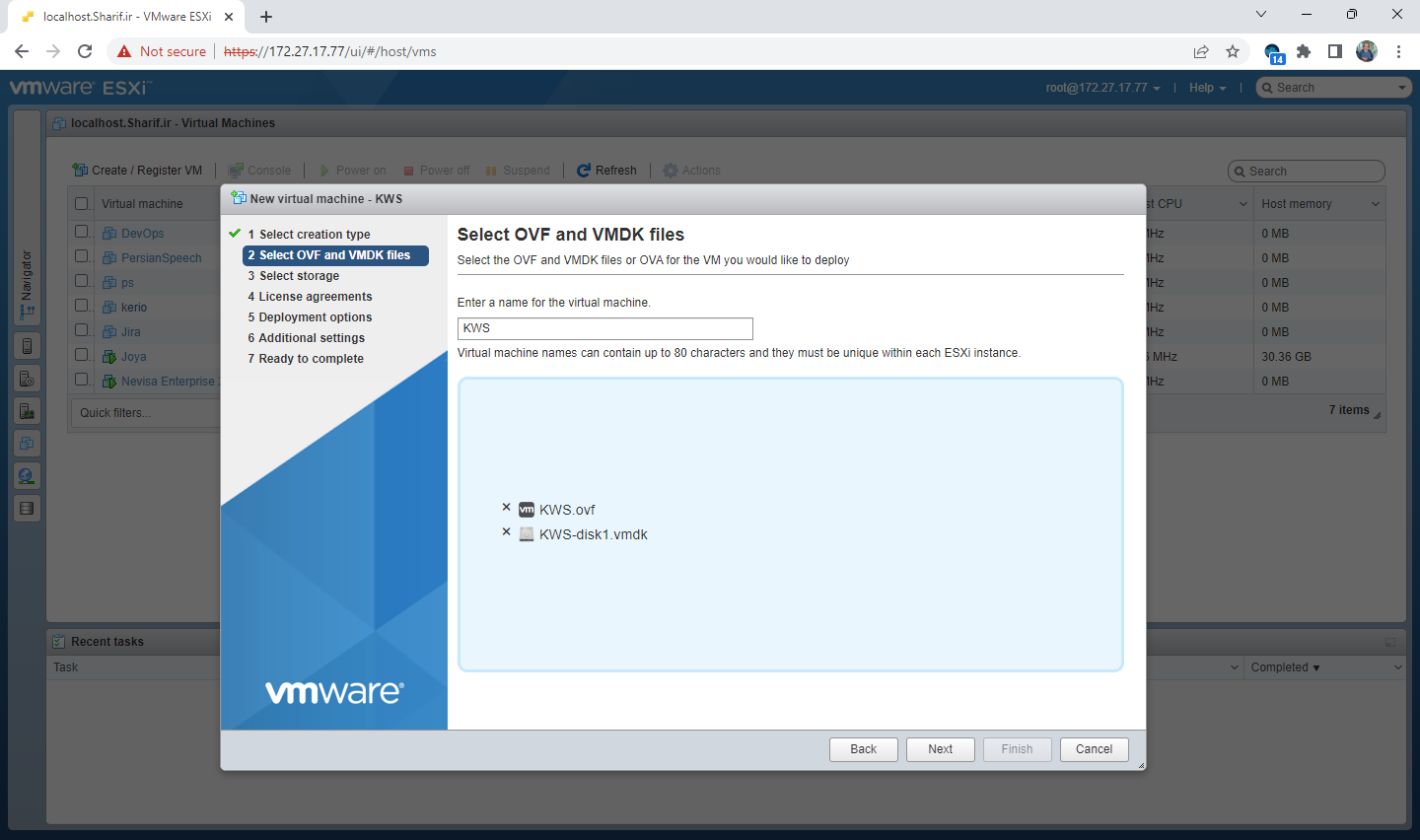
### نصب روی سرور ESXi

مراحل نصب سامانه نویسا روی سرور ESXi نسخه 6.5 به بعد به صورت زیر است.

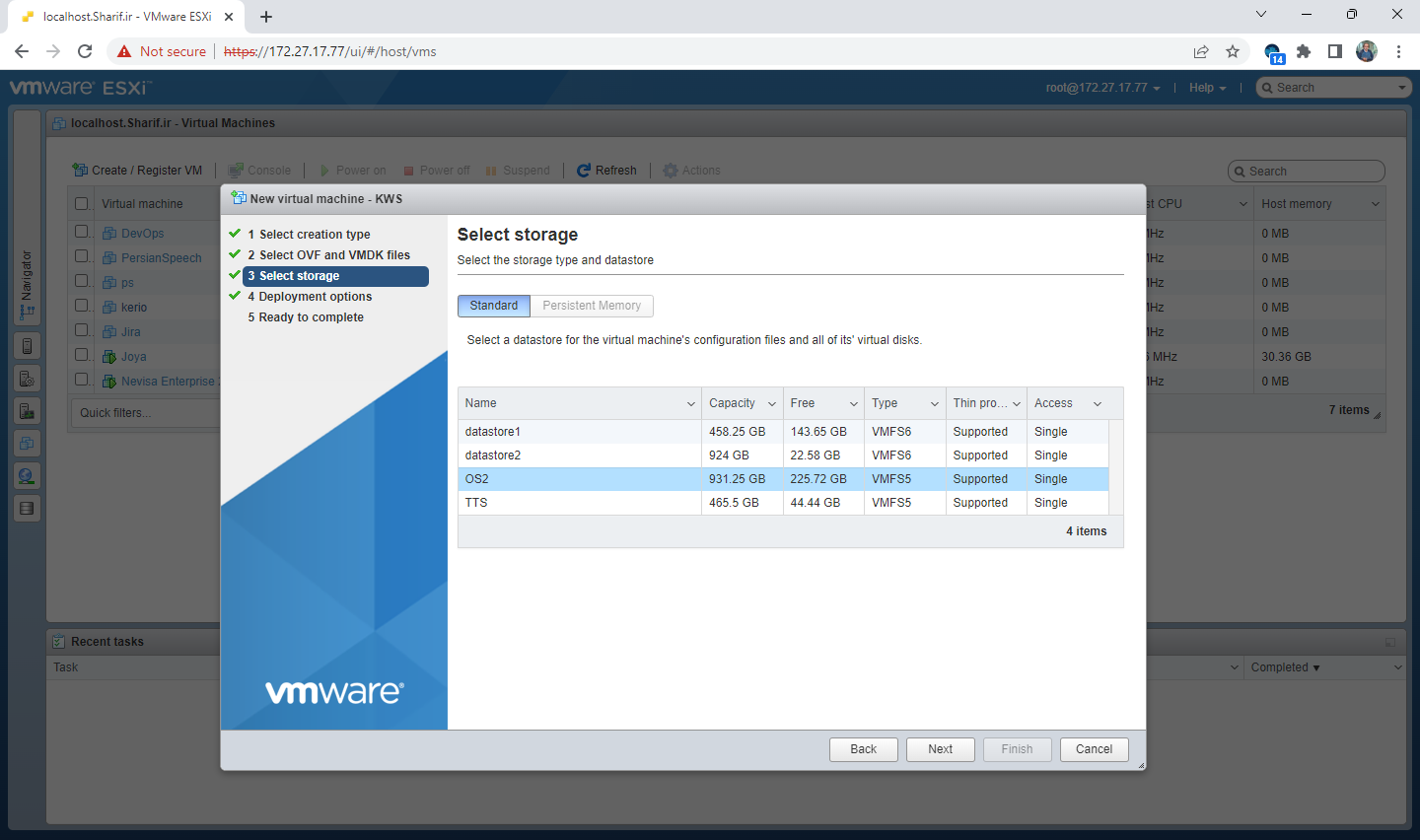
1. وارد کنسول ESXi شوید.
2. از قسمت Virtual Machines گزینه Create/Register VM را انتخاب کنید.
3. در پنجره باز شده گزینه Deploy a virtual machine from an OVF or OVA file را انتخاب کنید و دکمه Next را بزنید.



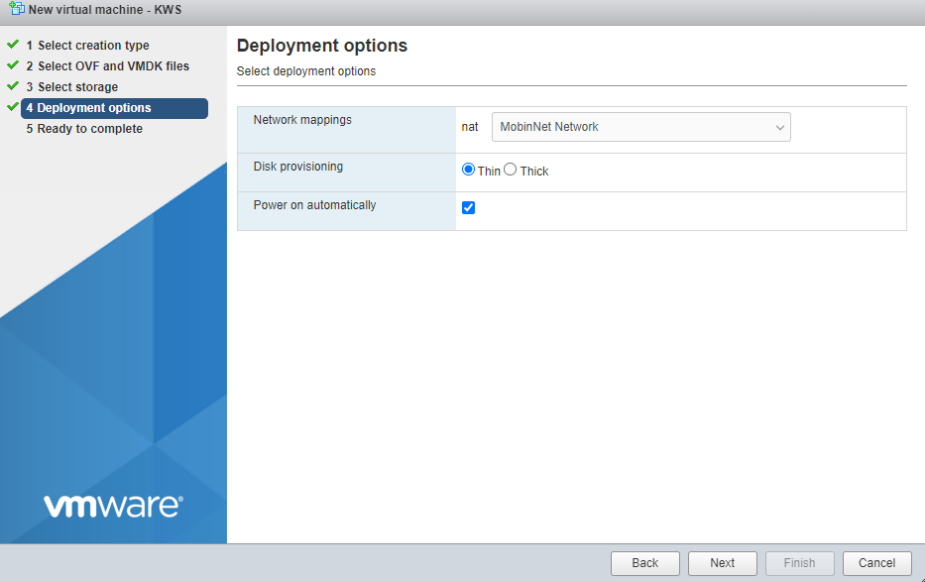
1. در پنجره بعدی نام ماشین مجازی را مشخص کرده و فایل ovf و vmdk را در ناحیه مشخص شده در پنجره بکشید و بیاندازید.



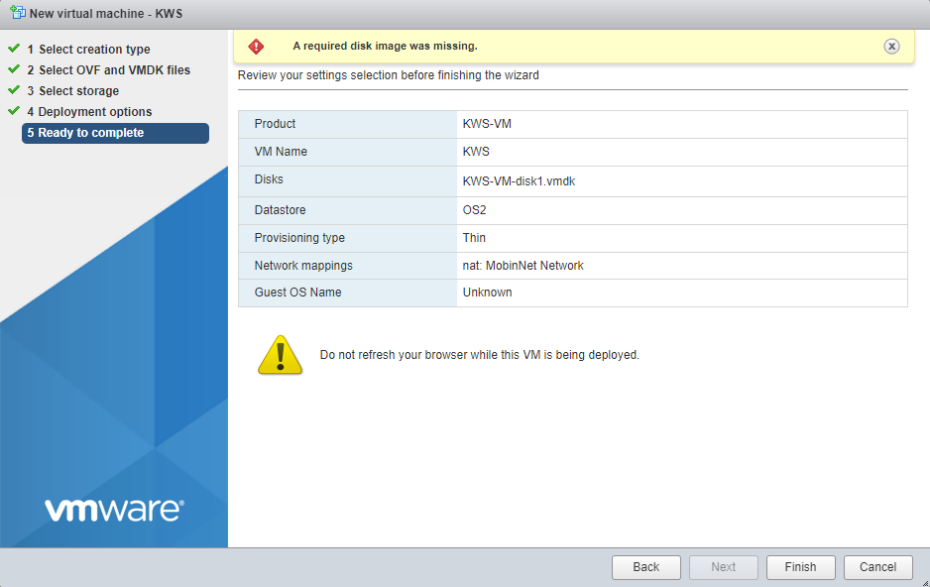
1. در پنجره بعدی datastore برای ایجاد ماشین مجازی را انتخاب کنید. حداقل فضای خالی روی datastore باید 100GB باشد.



1. در پنجره بعدی تنظیمات شبکه را انتخاب کنید.



1. در مرحله آخر دکمه Finish را بزنید و منتظر بمانید که ماشین مجازی ایجاد شود.



1. ادامه کار را مطابق مرحله سوم به بعد نصب VMware Workstation ادامه دهید.

# اطلاعات تخصصی برای برنامه نویسان

مستندات مربوط به سرویس نویسا همراه با نمونه کدهای اتصال در گیت هاب نویسا قرار دارد:

<https://github.com/nevisa-team/nevisa-v2-docs>

## لیست سرویس‌های اپلیکیشن:

**1. api-server**

* این سرویس به درخواست‌های تمامی کاربران به صورت همزمان پاسخ می‌دهد
* مستندات Swagger نحوه استفاده از API نیز موجود است. https://<ServerAddress>/api/docs
* مدیریت کاربران و اتصال آنها از طریق Websocket (Socket.io) به عهده این سرویس است

**2. asr-server**

* تبدیل گفتار به متن به زبان فارسی و استخراج کلمات کلیدی به عهده این سرویس می‌باشد

**3. queue-server**

* مدیریت پردازش فایل‌های هر کاربر و تشکیل صف انتظار برای هر کاربر به عهده این سرویس می‌باشد

## مدیریت سرویس‌ها با PM2

تمامی سرویس‌های بالا با استفاده از PM2 مدیریت می‌شود. مستندات کامل استفاده از PM2 در این [لینک](https://pm2.keymetrics.io/docs/usage/quick-start/) موجود است.

تمامی دستورات PM2 حتماً باید با sudo اجرا شود.

* پس اجرای این دستور لیست تمامی سرویس‌ها به همراه جزئیات نمایش داده می‌شود

sudo pm2 ls

| **id** | **name** | **mode** | **↻** | **status** | **cpu** | **memory** |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | api-server | cluster | 8 | online | 0% | 91.7mb |
| 1 | asr-server | fork | 754 | online | 0% | 3.9mb |
| 2 | queue-server | cluster | 2 | online | 0% | 70.7mb |

* پس از اجرای دستور سرویس مورد نظر / همه سرویس‌ها راه اندازی خواهد شد

sudo pm2 start <Service Name>

* پس از اجرای دستور سرویس مورد نظر متوقف خواهد شد

sudo pm2 stop <Service Name>

* پس از اجرای دستور سرویس مورد نظر راه اندازی خواهد شد

sudo pm2 restart <Service Name>

* پس از اجرای دستور سرویس مورد نظر راه اندازی خواهد شد

sudo pm2 delete <Service Name>

* پس از اجرای دستور وضعیت فعلی سرویس‌ها ذخیره خواهد شد. پس از راه اندازی مجدد سیستم عامل، سرویس‌ها با آخرین وضعیت ذخیره شده اجرا می‌شوند

sudo pm2 save

## اتصال با Socket.io

### نحوه اتصال به socket در Javascript:

با استفاده از دستور زیر پکیج socket.io رانصب کنید

npm install socket.io-client

import { io } from "socket.io-client";

const socket = io("SERVER\_IP\_ADDRESS\_OR\_URL", { extraHeaders: { token: token, platform: "mobile-app" || "browser" }})

توکن: از سرویس Swagger استفاده کنید و در قسمت Authentication درخواست Login را ارسال کنید. در جواب درخواست، توکن برای شما ارسال خواهد شد.

### نحوه ارسال و دریافت پیام به سوکت:

socket.emit("Your Event Name", data)

socket.on("Your Event Listener Name", (response) => {

console.log(response); //socket response

})

### لیست رویدادهای سوکت (Events):

1. get-online-users

* پس از ارسال پیام به این event، لیست کاربران آنلاین به یک event listener به نام online-users ارسال می‌شود
* این event فقط از طریق کاربرانی که مدیر هستند قابل ارسال است در غیر اینصورت پاسخی دریافت نمی‌کنید

1. disconnect-user

* پس از ارسال این event کاربر مورد نظر از حساب کاربری خود خارج می‌شود و پیام مربوطه به کاربر نمایش داده می‌شود
* این event فقط از طریق کاربرانی که مدیر هستند قابل ارسال است در غیر اینصورت پاسخی دریافت نمی‌کنید
* ارسال داده حتماً باید به صورت JSON باشد مثال {id: "user id", deleteUser: true/false}
* درصورتی که کاربر مورد نظر حذف کرده باشید باید مقدار deleteUser را true و صورتی که حساب کاربری شخص مورد نظر را مسدود کرده‌اید مقدار را false قرار دهید

1. get-system-monitor

* پس از ارسال پیام به این event، لیست کاربران آنلاین به یک event listener به نام system-monitor ارسال می‌شود
* این event فقط از طریق کاربرانی که مدیر هستند قابل ارسال است در غیر اینصورت پاسخی دریافت نمی‌کنید

1. start-microphone

* با ارسال این event سرور، برای دریافت chunk های صوتی آماده می‌شود
* پس از ارسال، سرور، قفل سخت افزاری را بررسی می‌کند. این مرحله ممکن از زمانبر باشد.
* نتیجه بررسی قفل سخت افزاری در یک event listener به نام start-microphone گزارش خواهد شد

1. microphone-blob

* پس از ارسال event مربوط به آماده سازی سرور (start-microphone) و دریافت گزارش بررسی قفل سخت افزاری درصورتی که گزارش lockChecked: true بود می‌توانید chunk های صوتی را ارسال کنید
* برای بهبود در سرعت دریافت نتیجه chunk های صوتی را هر 500 میلی ثانیه ارسال کنید
* نتیجه در یک event listener به نام result برای شما ارسال خواهد شد

1. stop-microphone

* پس از ارسال event پردازش شما میکروفونی متوقف خواهد شد

1. set-text-to-number

* با ارسال این event، سرویس تبدیل اعداد فعال/غیرفعال خواهد شد
* ارسال داده حتماً باید به صورت JSON باشد مثال { textToNumberStatus: true / false}
* درصورت بروز خطا، پیام خطا در یک event listener به نام message گزارش خواهد شد

1. set-punctuations

* با ارسال این event، سرویس درج خودگار علائم نگارشی فعال/غیرفعال خواهد شد
* ارسال داده حتماً باید به صورت JSON باشد مثال { punctuationStatus: true / false}
* درصورت بروز خطا، پیام خطا در یک event listener به نام message گزارش خواهد شد

1. start-file-process

* پس از ارسال فایل با استفاده از API، اطلاعات فایل یا فایل‌های ارسال شده را به صورت JSON Array در این event ارسال کنید
* پس از ارسال این event قفل سخت افزاری بررسی و نتیجه در یک event listener به نام start -file-process برای شما ارسال خواهد شد
* درصورت بروز خطا، اطلاعات خطا در یک event listener به نام message برای شما ارسال خواهد شد
* درصورت عدم بروز خطا، پردازش فایل آغاز خواهد شد و آخرین وضعیت فایل مورد نظر در یک event listener به نام queue-report برای شما ارسال خواهد شد

1. stop-file-process

* با ارسال شناسه فایل در حال پردازش، عملیات پردازش فایل مورد نظر متوقف خواهد شد
* شناسه فایل حتماً باید به صورت JSON ارسال شود مثال: {"id: "file id}