AIStarter 项目管理平台

使用说明手册

文件状态:	文档作者	啊雄
【 √ 】草稿	参与者	阿东
【 】正式发布	当前版本	Latest Release 2.0.1 LTS
【 】正在修改	完成日期	2024 年 4 月 29 日

目录

一、前言

AIStarter 启动器是一款 AI 项目管理平台，旨在为用户提供一键下载、安装、启动和管理各种 AI 项目，同时支持用户通过脚本编辑创建自己的 AI 项目。

二、AIStarter 介绍

2.1、自动化

用户只需点击一下即可完成 AI 项目的环境检测、部署、安装和优化，从而轻松地下载、安装、启动各类 AI 项目，开始正常使用。

2.2、脚本编辑

- 用户可以通过脚本编辑在 AIStarter 上创建自己的 AI 项目，以满足其特定需求并扩展项目功能。这些简单的脚本语言涵盖了但不限于：编写文件、执行命令、安装库和其他应用程序、发布文件、下载文件、浏览网络以及管理项目等。
- AIStarter 的脚本功能使得几乎所有人都能在计算机上自动完成各种任务，无需人工干预。

2.3、更新与分享

AIStarter 支持项目的维护、更新和一键分享、下载功能，用户可以及时获取最新的项目更新，并轻松分享自己的项目给其他用户。

2.4、多平台支持

AIStarter 支持多平台，包括 Windows、Mac 和 Linux，确保用户无论使用哪种操作系统都能享受到便捷的 AI 项目管理与启动体验。

2.5、硬件要求

AI 项目的运行需要一定的硬件支持，以下是建议的最低硬件配置，以确保您的项

目能够顺利运行：

- GPU：8GB
- 内存：8GB
- CPU 核心数：8 核心

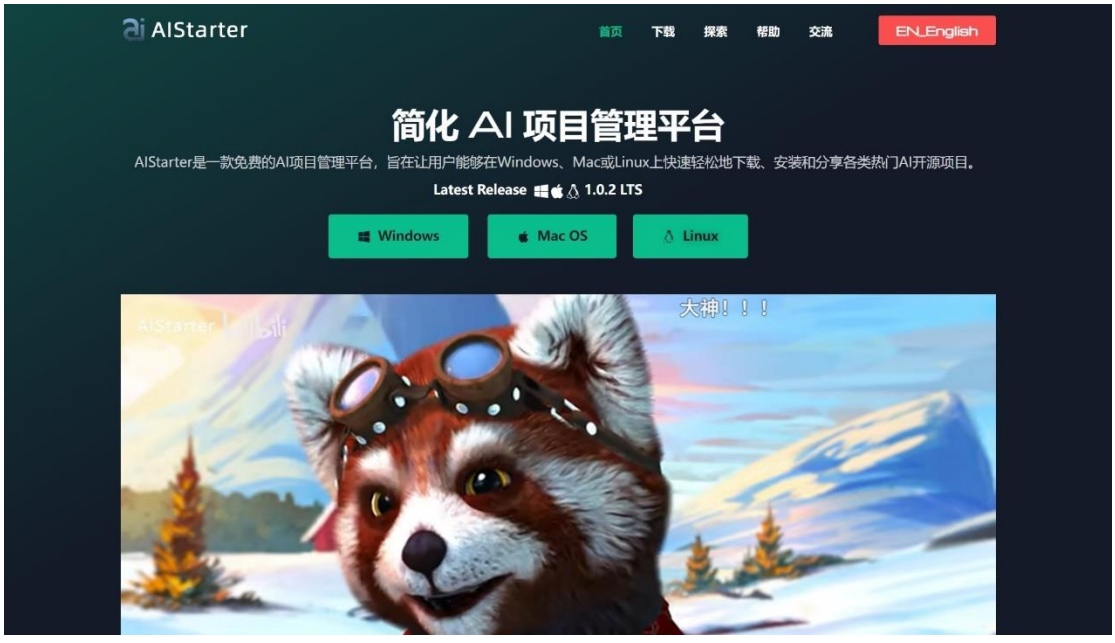
请注意，这些是建议的最低硬件配置。对于某些大型或计算密集型的 AI 项目，可能需要更高配置的硬件来提供更好的性能和稳定性。建议根据具体的 AI 项目需求和预算做出最终的硬件选择。

三、AIStarter 本体和 AI 项目下载、安装、卸载

3.1 AIStarter 本体下载

3.1.1 登录官网，下载 AIStarter，请根据您所使用的操作系统选择。

AIStarter 官网地址，**中文：**www.starter.top **| 英文：**www.starter.one



3.1.2 下载 AIStarter 本体后，双击按提示下一步下一步即可完成程序安装。

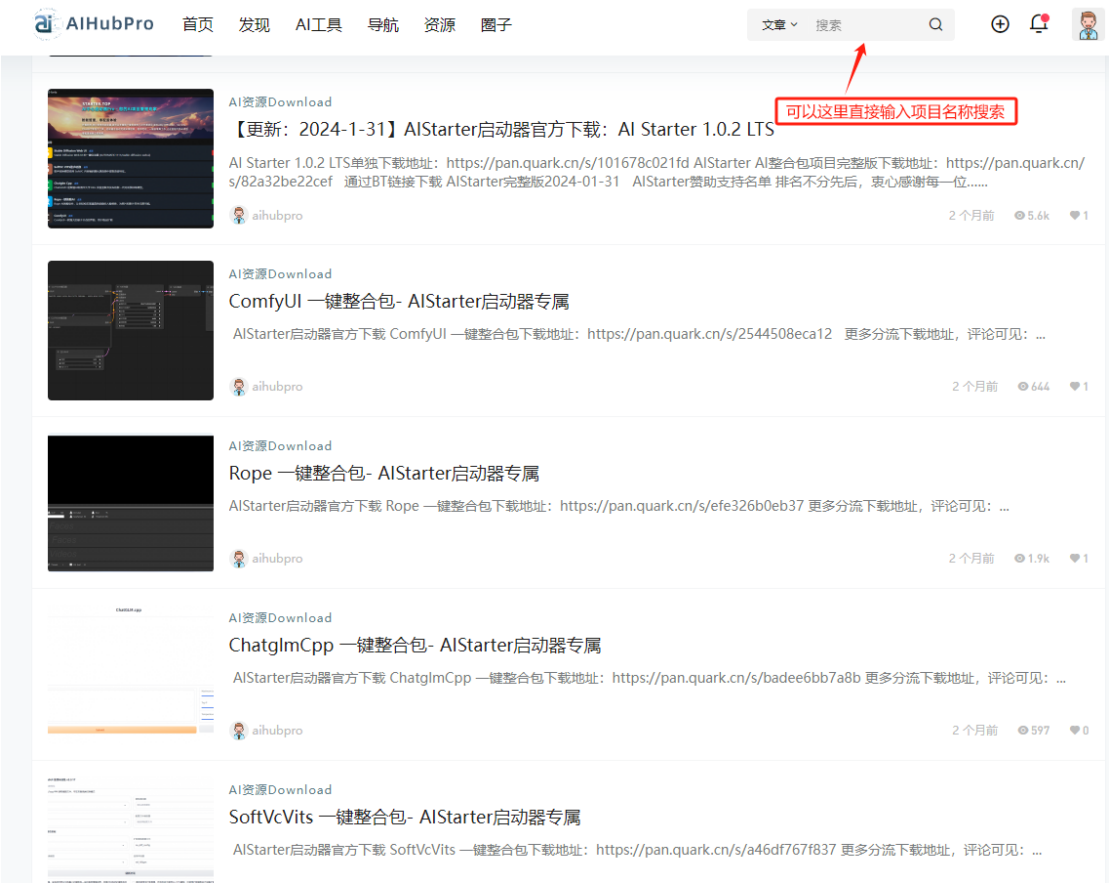
3.2 AI 项目下载

3.2.1 在 AIStarter 的市场界面上，用户可以直接点击所需 AI 项目的下载按钮，系统将自动完成 AI 项目的下载和安装。

3.2.2 分流下载即用户可以选择 AI 项目的离线下载方式。在官网下载页面，用户可以自行选择网盘、BT 等其他方式进行下载，下载完成后解压并将文件放置在 AIStarter 指定的目录中。

AIStarter 启动器下载地址：https://www.starter.top/download/

AI 项目下载地址：https://www.aihubpro.cn/collection/aistarter



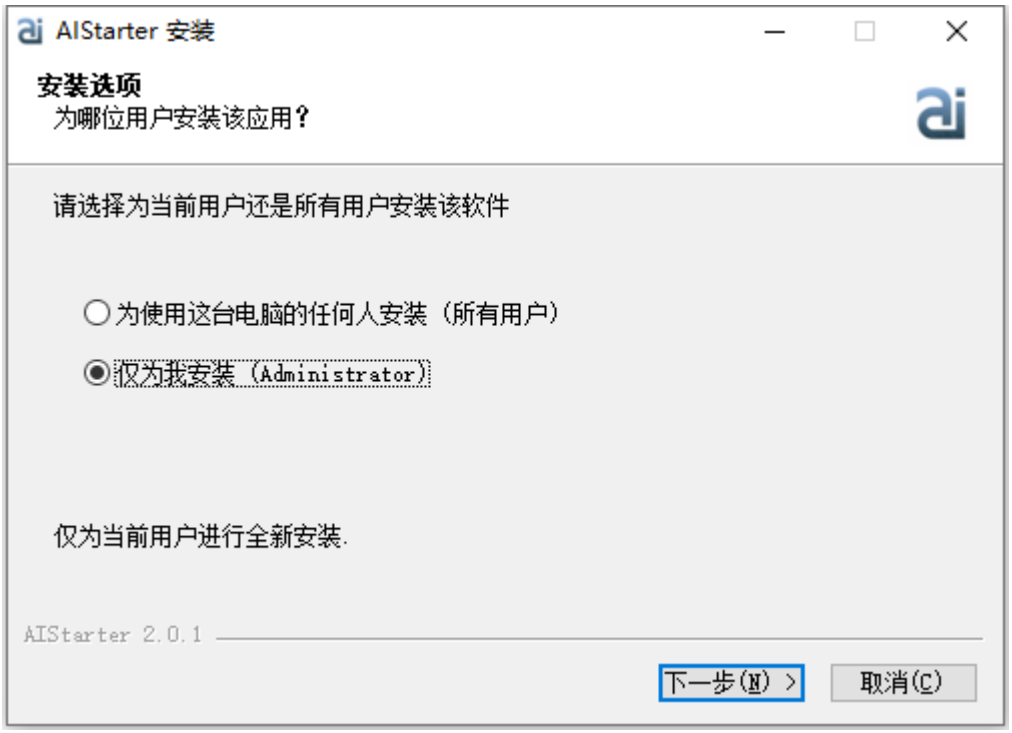
3.3 程序安装

3.3.1 AIStarter 安装

- 双击 AIStarter 程序安装包，如下图，根据提示下一步下一步完成安装。

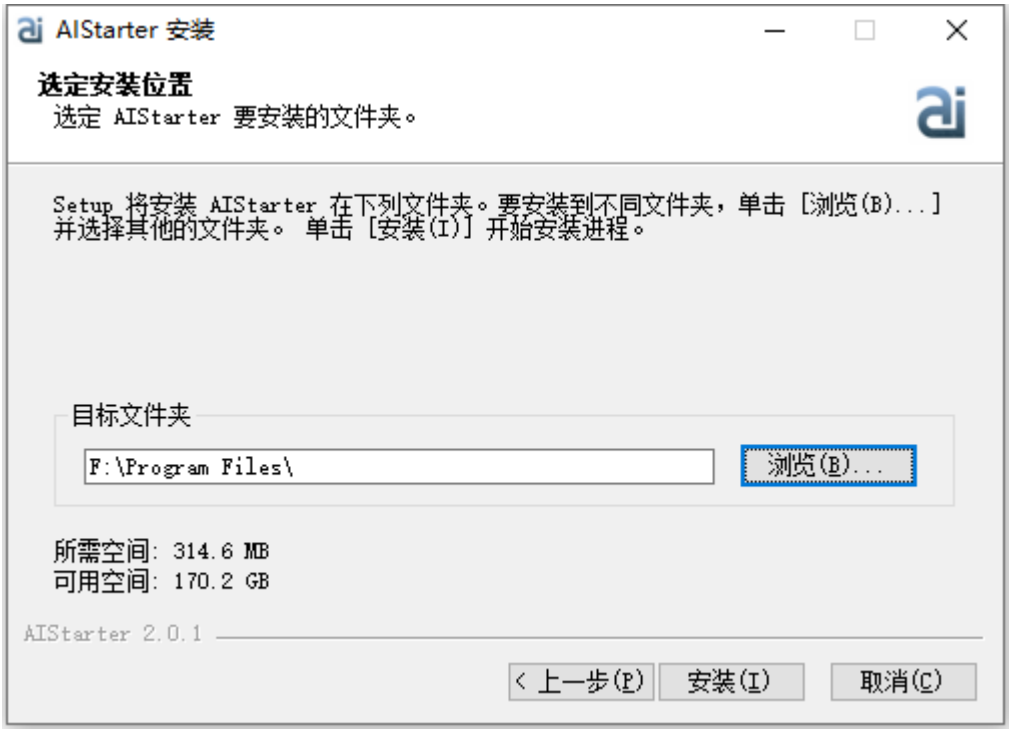
第一步：双击 AIStarter 安装程序

第二步：根据用户需求选择，点击下一步



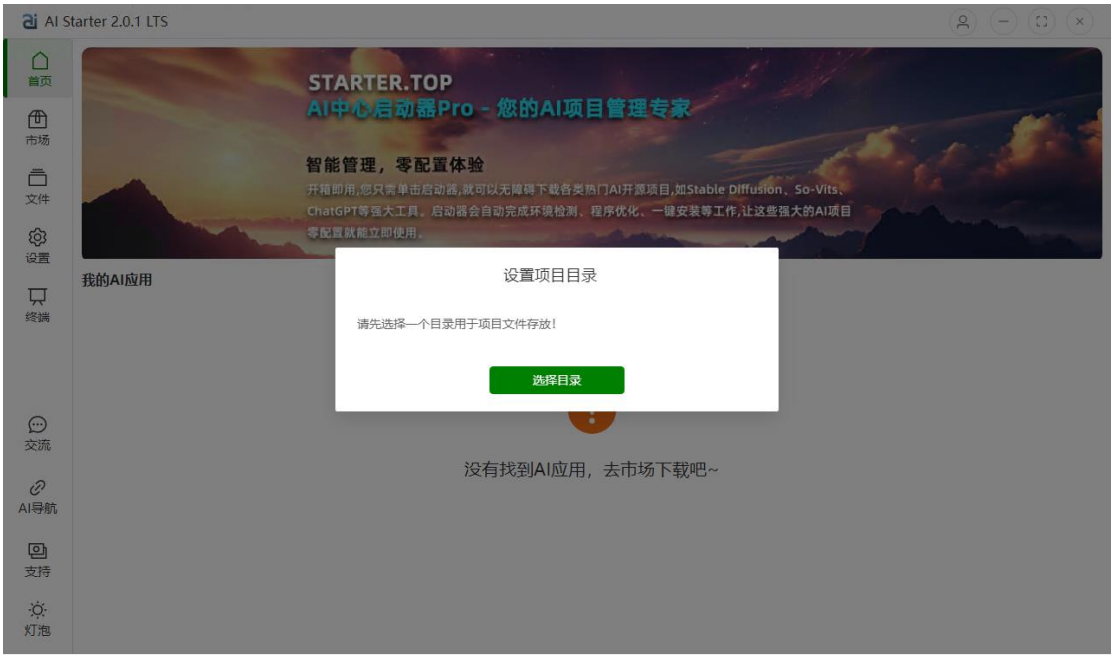
第三步：选定安装位置，通常默认即可

（路径不要使用中文字符或特殊字符，使用英文命名）



第四步：安装完成

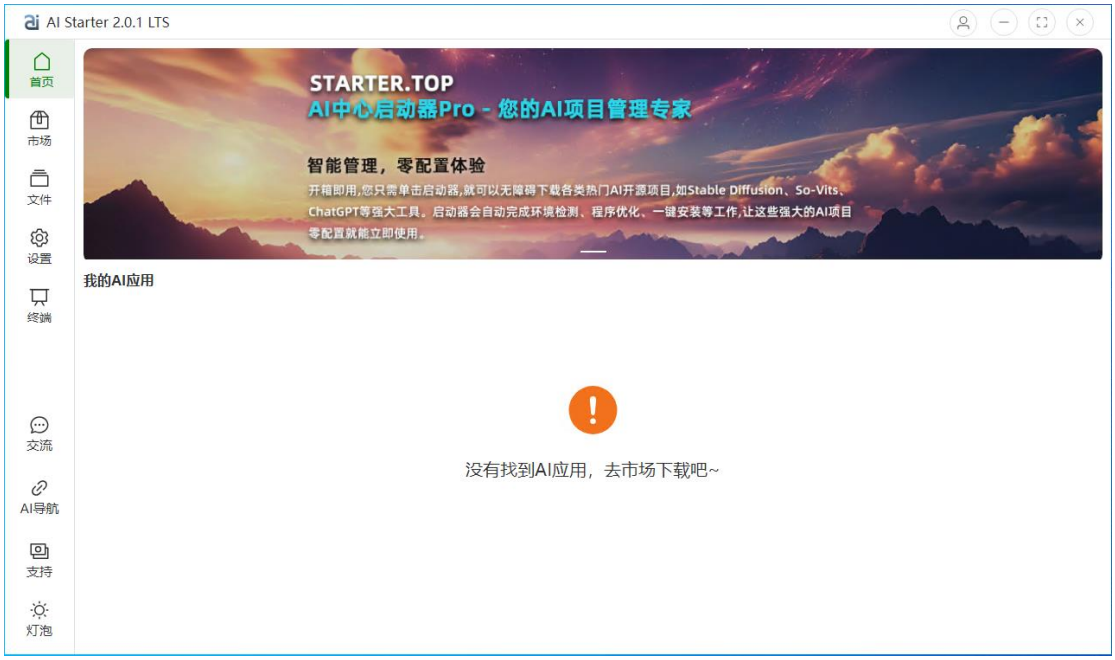
- 安装完成后，首次启动需要设置项目根目录，请注意，根目录路径不要使用中文字符或特殊字符，使用英文命名。



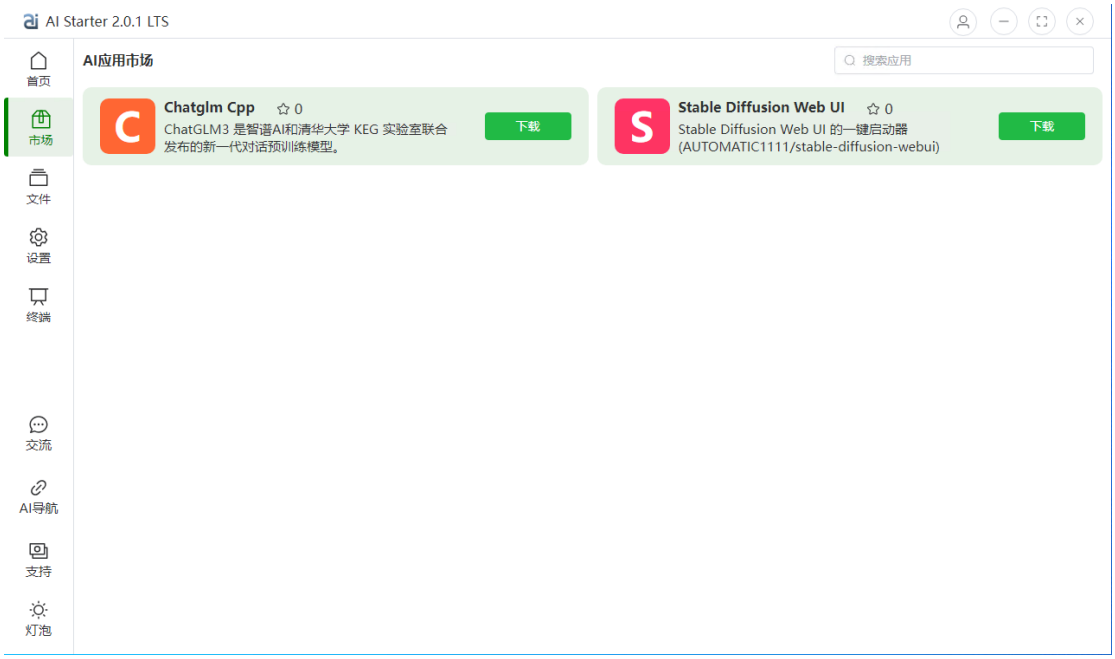
由于 AI 项目通常占用较大空间，请尽量选择足够的硬盘空间作为根目录。

3.3.2 AI 项目安装

- 首次启动软件界面



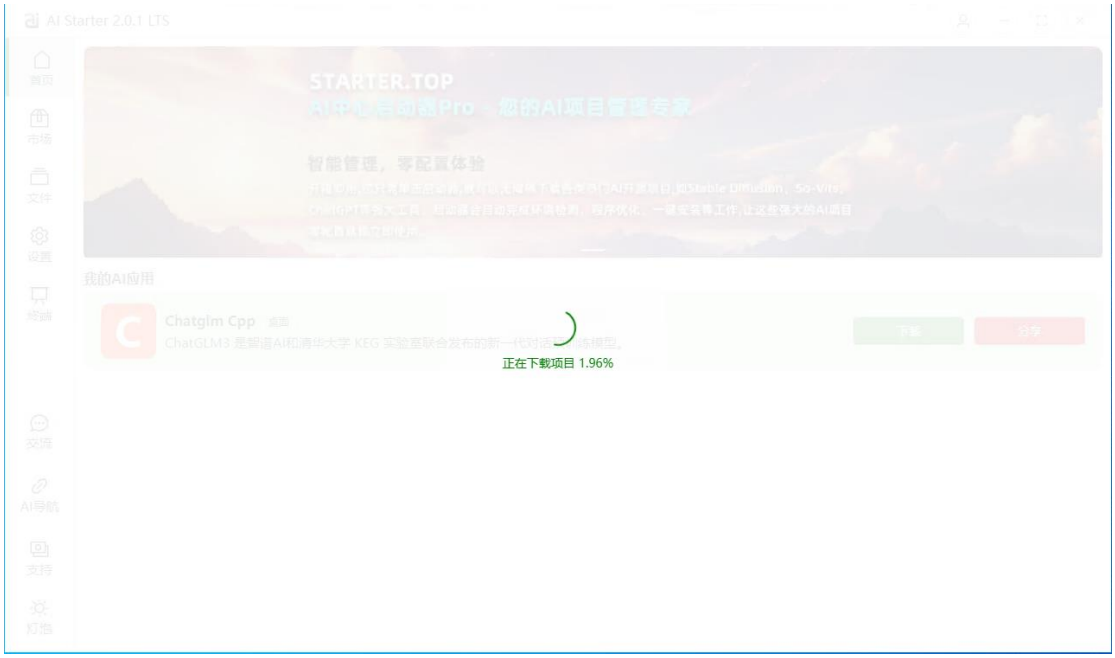
- 打开软件点击市场-AI 应用市场



- 点击下载所需的 AI 项目

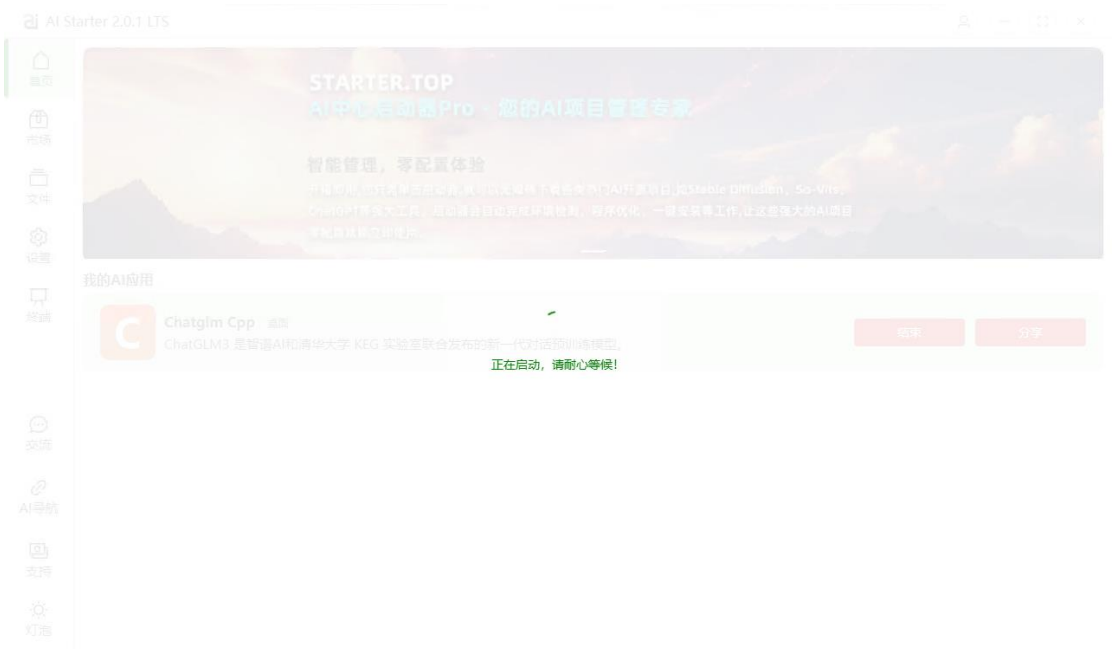
- 点击首页-点击下载 AI 项目

- 点击首页-下载-安装 AI 项目

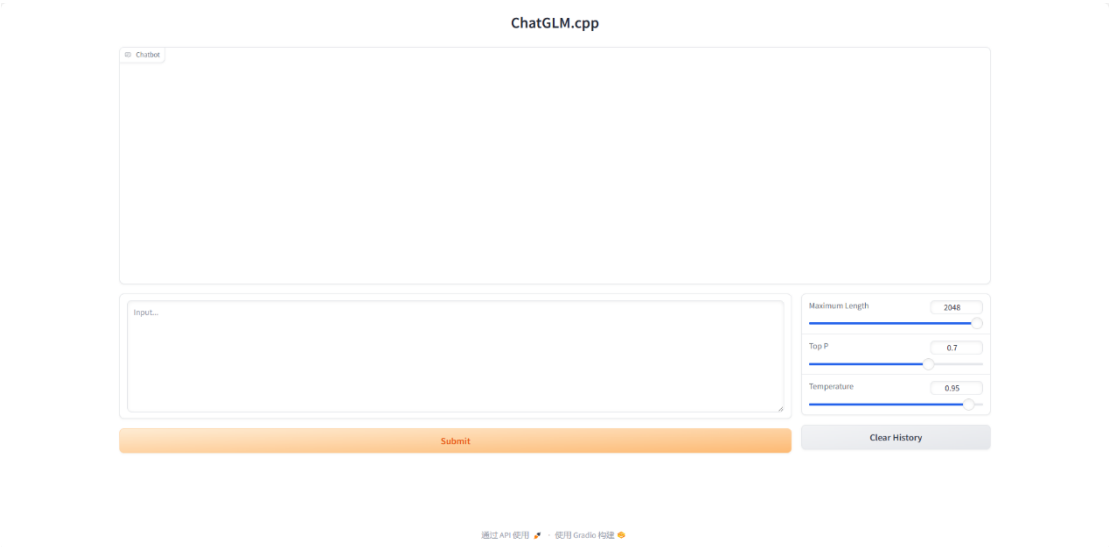


- 点击首页-启动 AI 项目


AI Starter 2.0.1 LTS
首页
市场
文件
设置
终端
交流
AI导航
支持
灯泡
STARTER.TOP
AI中心启动器Pro - 您的AI项目管理专家
智能管理，零配置体验
开箱即用,您只需单击启动器,就可以无障碍下载各类热门AI开源项目,如Stable Diffusion、So-Vits、
ChatGPT等强大工具。启动器会自动完成环境检测、程序优化、一键安装等工作,让这些强大的AI项目
零配置就能立即使用。
我的AI应用
Chatglm Cpp
桌面
ChatGLM3 是智谱AI和清华大学 KEG 实验室联合发布的新一代对话预训练模型。
启动
分享

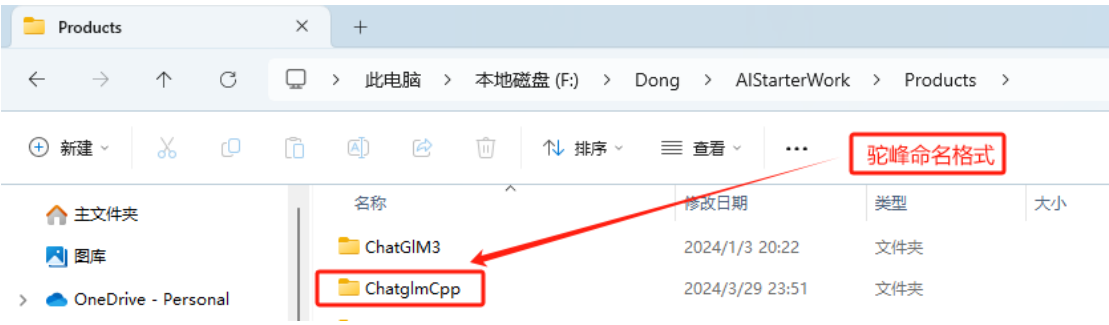

AI Starter 2.0.1 LTS
STARTER.TOP
我的AI应用
Chatglm Cpp
正在启动，请耐心等候!

- AIStarter 打开所启动的 AI 项目，即可正常使用



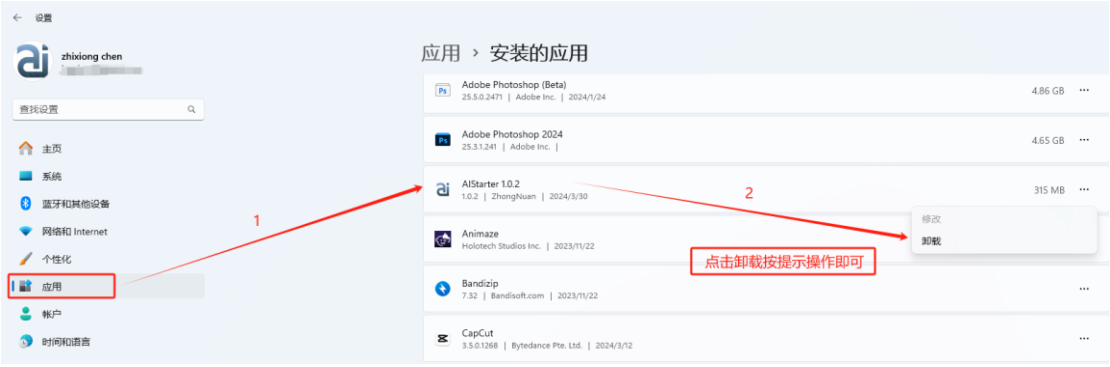
3.3.3 AI 项目离线下载

- 您可以选择网盘、BT 等其他方式进行下载。下载完成后，请解压文件，并将其放置在"Products"文件夹中。请注意，解压后的文件名必须采用驼峰命名格式，并且不支持文件夹内嵌套或修改文件名。

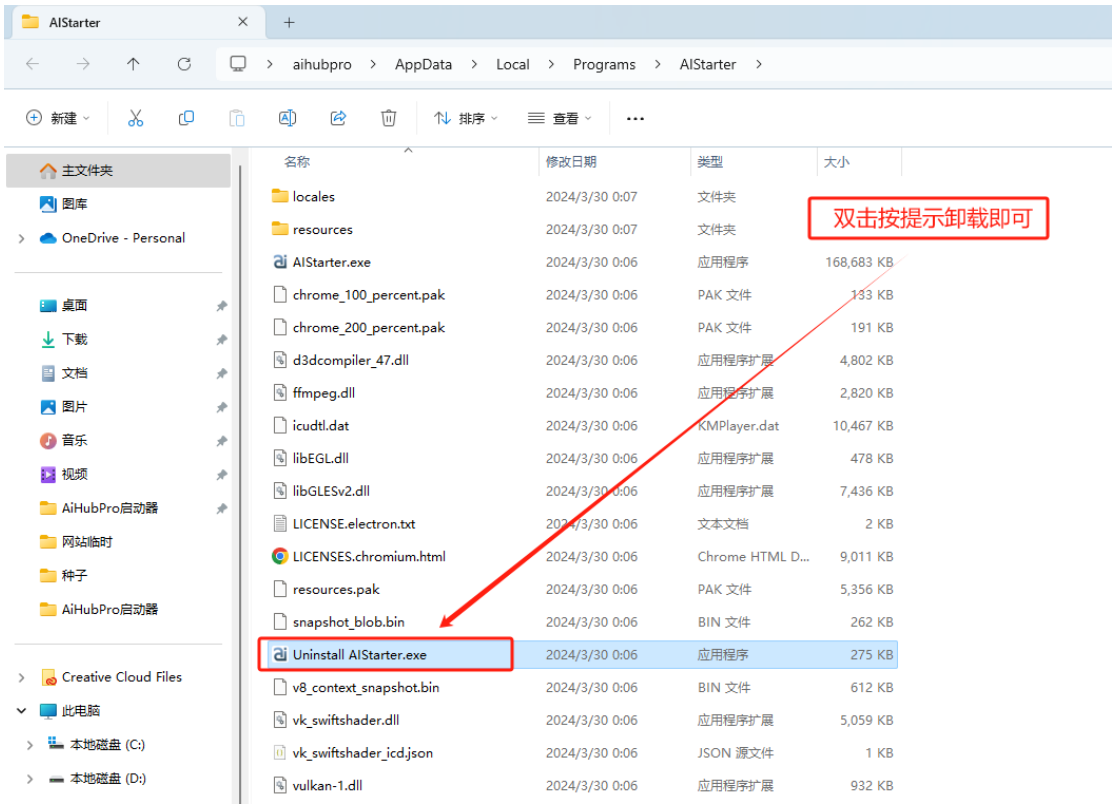


3.4 程序卸载（您可以按以下任一方式卸载程序）

- 开始菜单卸载流程：设置-应用-安装的应用-选择 AIStarter 程序-卸载

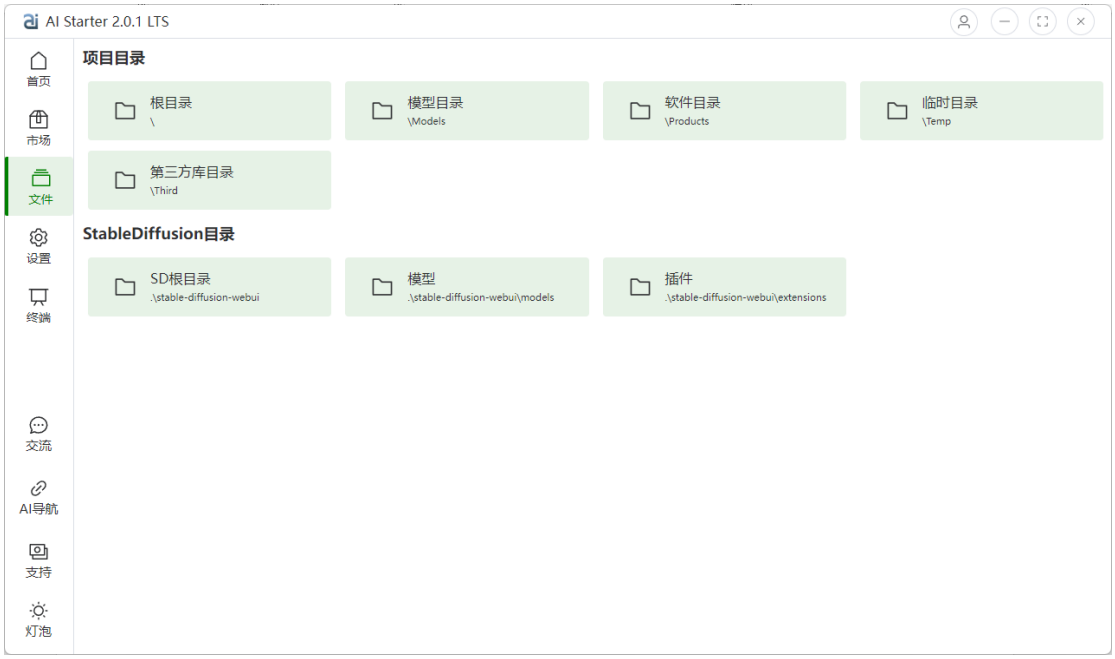


- 右键 AIStarter 打开文件所在的位置，双击 Uninstall AIStarter.exe 卸载。



四、文件（插件安装和模型安装）

程序本体默认目录和 AI 项目目录如下：



4.1 程序本体目录

在程序本体目录下，包含以下几个子目录：

根目录：程序的主要目录，包含程序的执行文件和主要配置文件。

模型目录：存放程序本体的模型文件目录。

软件目录：软件目录是存放已下载安装好的 AI 项目的位置，其中包括项目的执行文件、主要配置文件以及所依赖的库文件和运行时环境。

临时目录：用于存放临时文件的目录，例如临时生成的中间文件、缓存文件等。

第三方库目录：存放程序所依赖的各种第三方库文件，包括 Python 库、Git、Torch、Xformers、CuDNN 运行环境等。

分享目录：用于存放用户分享打包的文件或项目的目录，方便用户之间进行文件或项目的分享下载。

4.2 AI 项目文件目录

根目录：AI 项目的主要目录，包含项目的执行文件和主要配置文件。

模型目录：存放 AI 项目所使用的模型文件，包括训练好的模型和模型参数等。

插件目录：存放 AI 项目所使用的插件文件，用于扩展项目的功能和特性。

脚本目录：用于存放用户打包分享的脚本文件，经过审核后的脚本会显示在市场的 AI 应用市场中。这些脚本可以包括各种功能性脚本，例如安装脚本、环境配置脚本、工作流程脚本等。通过将脚本存放在这个目录下并进行分享，用户可以方便地共享自己的脚本，并让其他用户从中受益。

以上为默认设置，同时，AIStarter 还支持用户自定义添加文件路径。AI 项目相关的插件安装和模型安装，请参考项目官方说明。通常情况下，大多数主流项目将插件或模型放置到对应的文件夹中，然后重新启动即可完成安装。

五、设置

5.1 软件设置功能和选项

5.1.1 语言设置：

- 支持中文和英文语言设置：您可以选择界面显示的语言为中文或英文。
- 用户编辑语言文件：如果您希望使用其他语言，您可以编辑语言文件，路径为：[语言文件路径]。编辑完成后，重新启动软件即可选择相应的语言设置。

5.1.2 显卡 GPU 设置：配置显卡的相关选项，包括性能和功耗等设置。

5.1.3 CPU 设置：调整 CPU 的性能模式和核心分配等设置。

5.1.4 项目根目录：指定 AI 项目的根目录，用于存放项目文件和相关数据。

5.1.5 开发者选项：提供高级设置和开发者工具，适用于开发人员调试和测试。

5.1.6 API 设置：配置 API 接口的相关选项，包括调用限制和安全设置等。

- API 接口调用：使用 API 接口进行数据访问和交互。

5.1.7 网络设置：配置网络连接的相关选项，包括代理和 VPN 设置。

- 网络代理：配置代理服务器以访问互联网。
- VPN 设置：配置虚拟专用网络以进行安全连接。

5.1.8 环境检测：环境检测功能用于确保系统正常运行，包括两个方面的检测：

1. 系统环境检测：

- 包含常用的 Python、Git、Torch、Xformers、CuDNN 等软件的版本检测。
- 确保这些软件的安装和版本符合系统要求，以保证后续功能的正常运行。

2. 硬件设备检测：

- 检测系统中的硬件设备，如 GPU、CPU 等。
- 确保硬件设备的正常连接和功能，以满足后续运算任务的要求。

通过环境检测功能，用户可以快速了解系统的运行环境是否符合要求，从而及时调整和处理可能出现的问题，确保系统正常运行。

5.1.9 版本信息：显示当前系统和应用程序的版本信息。

5.2 AI 项目设置

5.2.1 显卡或 GPU 选择：选择用于运行 AI 项目的显卡或 GPU。

5.2.2 启用 API：启用 AI 项目中使用的 API 接口，以实现数据交互和功能扩展。

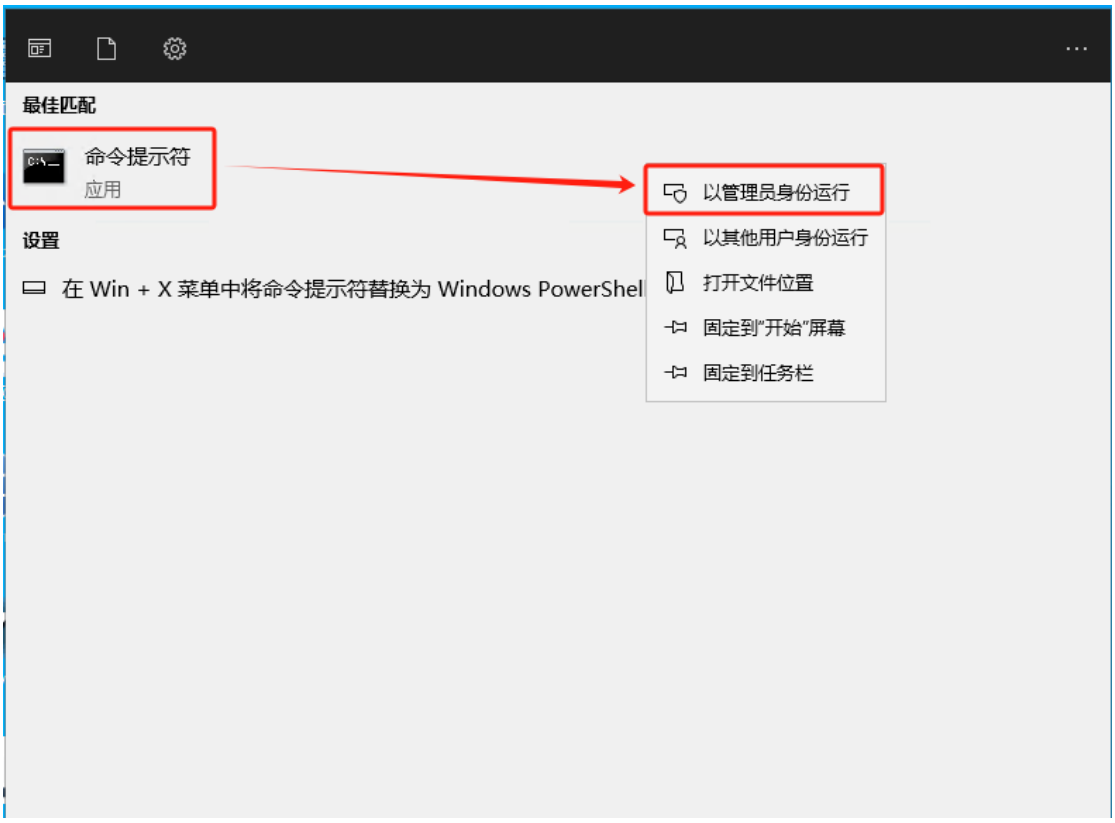
六、终端

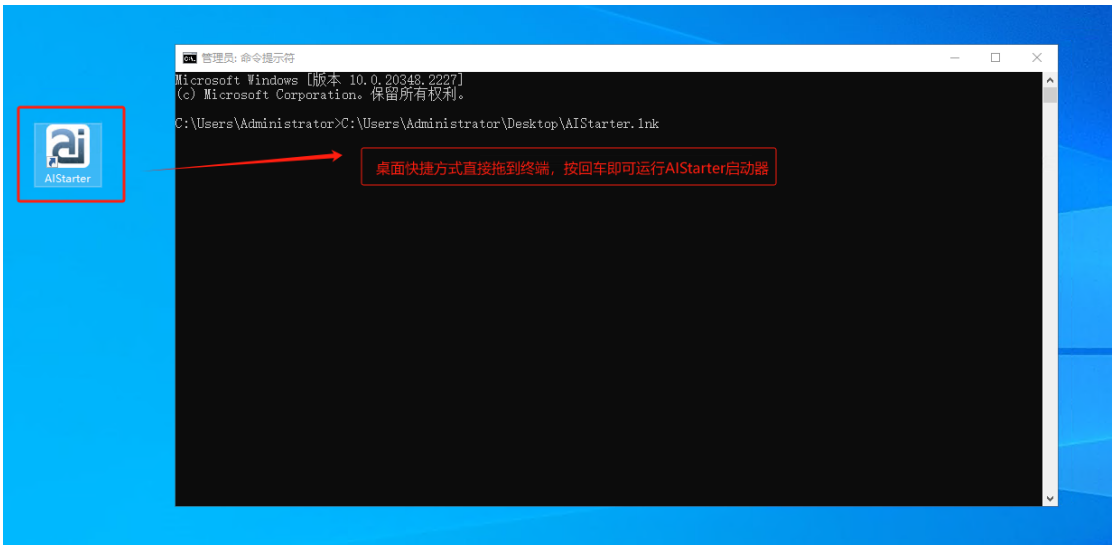
- 在 AIStarter 软件上，您可以通过终端来查看软件的运行日志，这对于出现问题时进行

排查非常有用。

- 通过终端，您可以实时监视软件的运行情况，了解其执行过程中是否出现了任何错误或异常。您可以使用命令来查看最新的日志信息，或者过滤特定关键字，以便更快地定位和解决问题。

- 您还可以通过直接右键管理员打开系统自动的终端，将 AIStarter 程序直接拖到终端上运行，这样您可以查看更详细的运行日志。通常，这种操作适用于开发人员或者需要进行软件打包分享的用户。

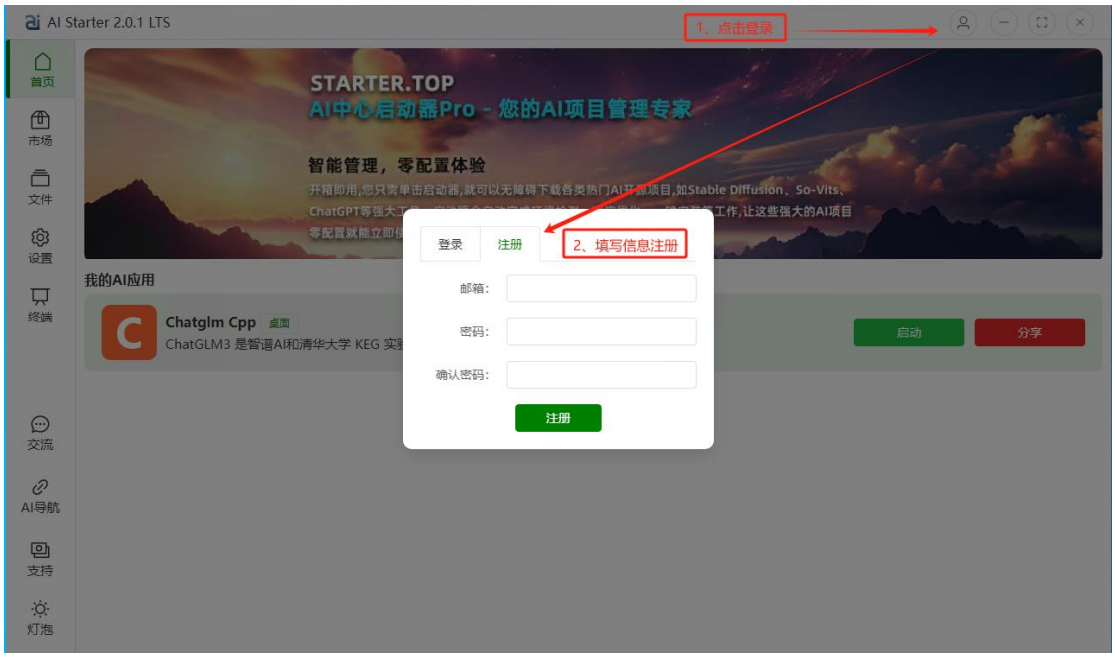




- 通过这种方式，您可以获得更加详尽和全面的运行日志信息，有助于开发人员快速定位和解决潜在的问题，并确保软件的稳定性和可靠性。

七、用户管理

7.1 用户登录-注册



7.2 用户密码

- "点击'更改密码'或'忘记密码'，输入注册邮箱后，系统将向您的注册邮箱发送包含密码重置链接的邮件。"

7.3 用户中心

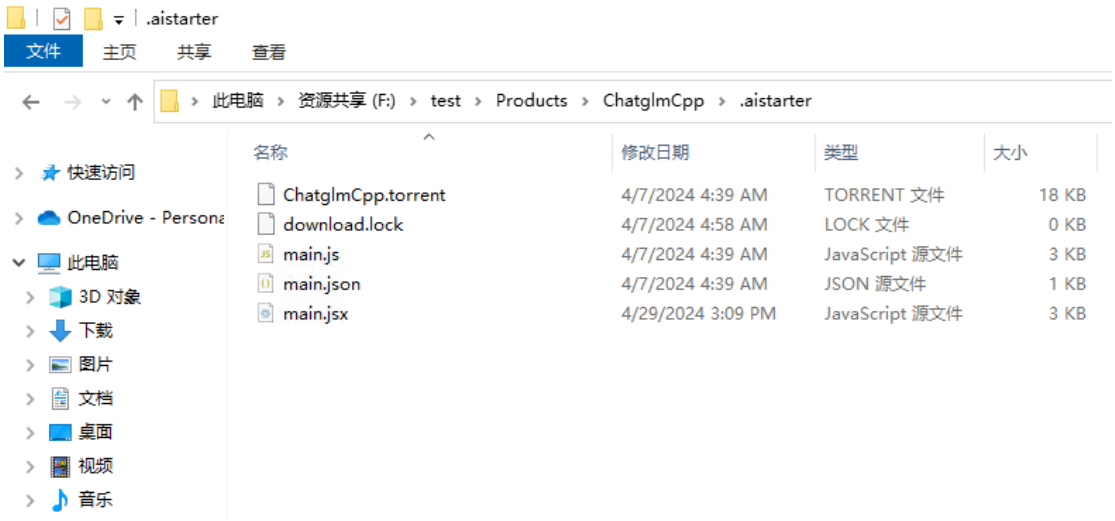
- 用户中心为用户提供了全面的功能，包括新项目和更新提醒、常规消息通知、项目收藏、点赞和评论等功能。用户可以与其他用户互动交流，同时记录下载历史，并支持用户对自己的项目进行打包分享、更新和删除管理。

八、脚本管理

8.1 脚本文件路径

脚本文件位于各个独立项目的目录中，具体路径为`.aistarter`文件夹内。

示例：



8.2 脚本说明

8.2.1、脚本概述

1、init:async function(){}

初始化方法，软件进入加载脚本时会调用一次（必要）

2、install:async function(){}

安装程序方法，点击安装按钮会调用此方法（必要）

3、download:async function(){}

下载实现，点击下载的时候会调用此方法，通常调用软件自带的 bt 下载方法（必要）

4、run:async function(){}

运行软件方法，点击运行按钮调用（必要）

5、exit:async function(){}

退出项目方法，点击项目退出的时候调用（必要）

6、onAppExit:async function(){}

软件退出后会调用此方法

7、isRunning:async function(){}

返回 AI 项目是否正在运行中，软件使用这个接口判断项目是否真正运行（必要）

8、onSettingChange:async function(settingKey, settingValue,

settingClass){}

当软件设置改变时会回调这个方法，回传的三个参数对应 addSetting 里面配置的值

参数：

settingKey 设置选项的键值

settingValue 设置选项的值

settingClass 配置所属的类别

9、addSetting:async function(){}

给项目添加设置选项，返回设置配置相关的结果

返回参数（json 对象）：

className 设置选项所属的类名

title 设置的标题

configList 配置的列表（json 对象数组）

key 设置选项的键值

label 设置标题

desc 设置的详细描述（可选）

type 设置的类型，支持（switch:开关、dir:目录选择、 select:下拉菜单）

default 默认值，没有设置时的初始值

8.2.2、脚本 API

1、async zn.markPluginInstalled(pluginName) {}

标记项目已经安装成功，当安装脚本已经正确安装完成后调用这个方法，通知软件标记改项目已经安装成功。

参数：

pluginName 脚本自身的名称，通常传 this.pluginName 即可

2、async zn.markPluginDownloaded(pluginName) {}

同上标记项目是否已经下载并且解压成功。注：调用 zn.downloadProjectTorrent 方法不需要在标记下载成功，该方法已经实现。

3、async zn.pluginLog(log){}

安装的时候显示安装日志（全屏菊花带文本）

4、async zn.isPluginInstalled(pluginName){}

判断项目是否已经安装成功，当调用 zn.markPluginInstalled 标记成功后返回 true

5、async zn.isProjectDownloaded(pluginName){}

判断项目是否已经下载解压完成，当调用 zn.markPluginDownloaded 标记成功后返回 true

6、async zn.addDirToClass(className, dirInfo){}

在软件的目录选项卡添加跳转目录

参数：

className 目录的类别，通常用 this.pluginName 即可

dirInfo 目录信息（json 结构）

icon：目录的图标

viewName：目录显示的名称

viewPath：显示的路径，只是显示实际的路径太长了

dirPath：目录实际跳转的路径，相对于软件工作目录或者绝对路径

7、async zn.execute (command, args, options = {}, callbacks = {}) {}

创建子进程执行命令

参数：

command 执行命令

args 命令参数数组

options 包含 cwd:执行目录、env:执行的环境变量

callbacks 回调，包含标准输出 stdout(std)、stderr(err)、created(childProcess)、close(code)

8、async zn.killProcessTree(pid){}

杀掉进程

参数：

pid 进程 ID，使用 zn.execute 的回调 created 返回的 childProcess 参数

9、async zn. showLoading(isShow, text, autoHideTime){}

在软件中显示全屏 Loading

参数：

isShow 是否显示,true 为显示

text 显示的文本

autoHideTime (可选)自动隐藏的时间，毫秒

10、async zn.getGupInfo(){}

获取系统 GUP 信息，包含显存、显卡型号等信息。可通过 console.log 打印日志查看

11、async zn. getNvidiaSmi (){}

获取系统 nvidia-smi 命令的信息，用于判断 CUDA 版本

12、async zn.getSettingValue(keyName, keyClass){}

获取软件设置值，由 addSetting 方法定义的值

参数：

keyName 设置的键值

keyClass 设置的类别

13、async zn.setSettingValue(keyName, keyVal, keyClass){}

设置软件配置的值，参数同上

参数：

keyVal 配置的值

14、async zn.showMessage(text){}

在软件中弹出消息提示

参数：

text 提示消息的内容

15、async zn.zn.terminalMessage(text){}

在软件中的终端打印文本

参数:

text 打印的文本

16、async zn.downloadProjectTorrent(pluginName){}

BT 下载 AI 项目，使用软件自带打包调用此方法

参数:

pluginName 脚本自身的名称，通常传 this.pluginName 即可

17、async zn.changeBtnState(pluginName, state){}

改变项目按钮的状态

参数:

state 按钮状态，包含 run:启动按钮状态、exit:结束按钮状态

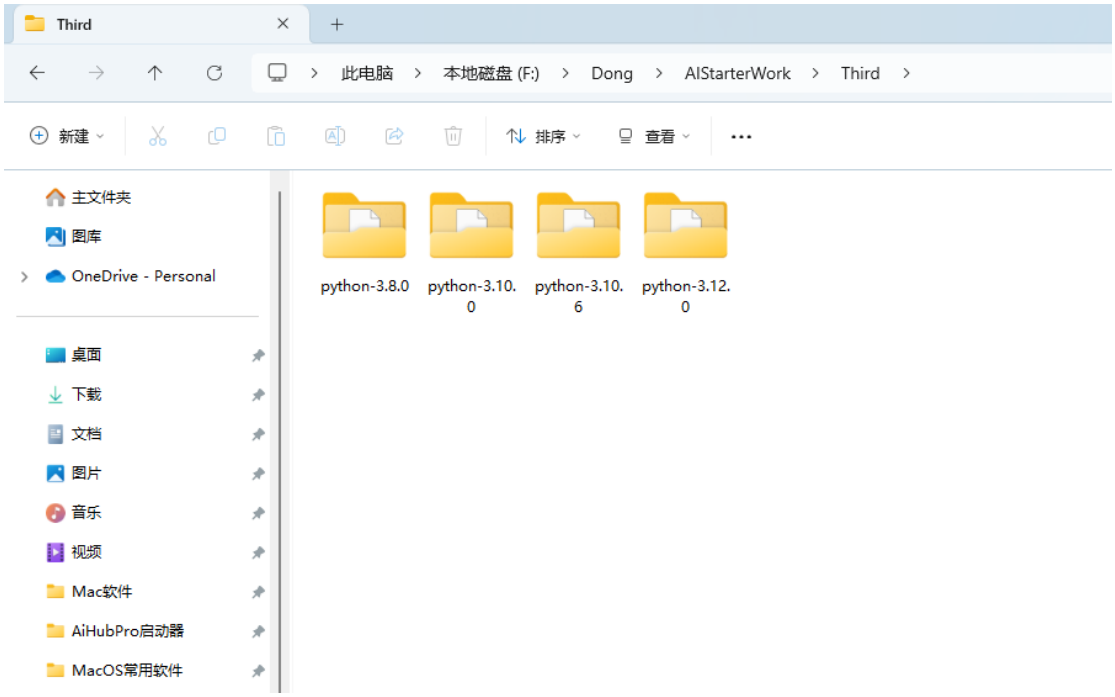
8.3 脚本下载

九、打包分享

9.1 打包软件说明

AIStarter 启动器集成了 AI 项目所需的常规环境，同时支持自定义增加环境，具体集成的环境可打开以下路径，环境存放在 AIStarter 项目根目录的`Third`文件夹中，同时也支持用户自定义目录。

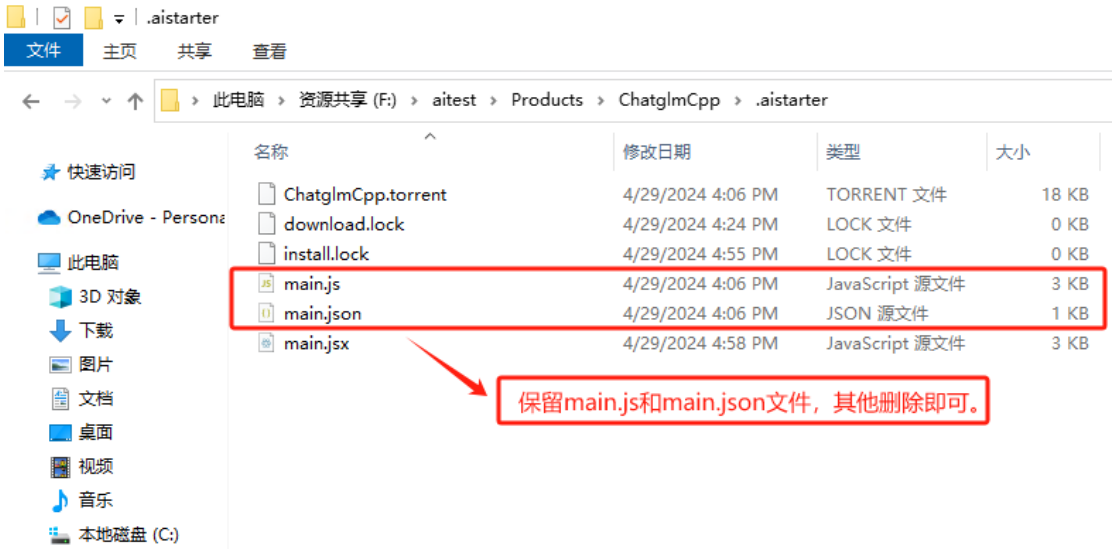
示例：



9.2 打包示例

9.2.1 脚本打包的步骤如下：

1. 在 Products 目录下新建一个目录，目录名应采用**驼峰命名**，只支持大小写英文、数字和下划线，不支持中文和特殊字符。
2. 将 AI 项目拷贝到新建的目录中。
3. 在该目录下创建.aistarter 目录，并在其中新建 main.js 脚本文件和 main.json 描述文件（您可以直接复制已有的文件）。



9.2.2 main.json 示例


```
{
   "name": "ComfyUI 铁锅炖迷你版",
   "description": "ComfyUI 铁锅炖迷你版一款强大的基于节点的界面，用于稳定扩散",
   "version": "2024-04-27",
   "install_dir": "BlenderComfyUIMini",
   "author": " aihubpro@foxmail.com ",
   "platforms": [
      "win"
   ],
   "library": [],
   "compress_list": [
      "Blender_ComfyUI_Mini",
```

```
        "Workflow"
    ],
    "project_zip_size": 5513058
}
```

9.2.3 main.js 示例

```
{
    //执行初始化
    init:async function(){
        let pluginPath = this.pluginPath; //AI 软件所在目录

        console.log('Plugin Inited:', this.pluginName);

        this.runInstance = null

        this.addDir();

    },
    //执行安装
    install:async function(){

        // 标记安装成功（安装完成都要标记它，用于判断是否安装成功）
```

```
        await zn.markPluginInstalled(this.pluginName);


        this.addDir();


},

//执行下载
download:async function(){
        console.log("plugin download!!");
        await zn.downloadProjectTorrent(this.pluginName);
},

//运行
run:async function(){
        if(this.runInstance){
                console.log(this.runInstance.pid)
                //this.runInstance.stdin.write("stop")
                //this.runInstance.stdin.end();


                zn.killProcessTree(this.runInstance.pid);
```

```
            //this.runInstance.kill('SIGKILL');


            this.runInstance = null
            return;
        }

        zn.showLoading(true, "正在启动，启动时间比较长请耐心等候！");

        let runMode = await zn.getSettingValue('RunMode',
"BlenderComfyUI");

        let useCpu = "";
        if(runMode != "gpu"){
            useCpu = " --cpu"
        }

        //回调
        let callback = {
            created:(process) => {
                this.runInstance = process
            },
```

```
            stdout:(data) => {
                let stdoutStr = data.toString();
                if(stdoutStr.includes("Starting")){
                    zn.showLoading(false);
                }
            }
        }

        const options = {
            cwd: `${this.pluginPath}\\Blender_ComfyUI_Mini`
        };

        try {
            await zn.execute('cmd.exe', ['/c', `.\\python_embeded\\python.exe -u -s ComfyUI\\main.py${useCpu} --windows-standalone-build`], options, callback);
        } catch (error) {
            console.error(error.message);
            if(this.runInstance){
                zn.showMessage("启动失败请查看终端日志！");
                zn.changeBtnState(this.pluginName, "run");
```

```
                this.runInstance = null;
            }

        }

        zn.showLoading(false);

    },

    //点击退出
    exit:async function(){
        if(this.runInstance){
            let pid = this.runInstance.pid;
            this.runInstance = null
            zn.killProcessTree(pid);
        }
    },

    //软件退出时调用
    onAppExit:async function(){
        this.exit();
```

```
    },

    //是否正在运行
    isRunning:async function(){
        return this.runInstance != null;
    },

    //添加目录
    addDir:async function(){
        let isInstalled = await zn.isPluginInstalled(this.pluginName);
        if(isInstalled){
            zn.addDirToClass(this.pluginName, {icon:"Folder", viewName:"ComfyUI 根 目 录 ", viewPath:`..\\ComfyUI`, dirPath:`\\Products\\${this.pluginName}\\Blender_ComfyUI_Mini\\ComfyUI`});
            zn.addDirToClass(this.pluginName, {icon:"Folder", viewName:" 模 型 目 录 ", viewPath:`..\\ComfyUI\\models`, dirPath:`\\Products\\${this.pluginName}\\Blender_ComfyUI_Mini\\ComfyUI\\models`});
        }
    },
```

```
//设置菜单
addSetting:async function(){
    const settings = {className:"BlenderComfyUI", title:" 铁 锅 炖 ComfyUI 设置", configList:[
        {
          key: 'RunMode',
          label:'启动模式',
          type: 'select',
          options: [
            { label: 'CPU', value: 'cpu' },
            { label: 'GPU', value: 'gpu' },
            // ...其他 GPU 选项
          ],
          default: 'cpu'
        }
    ]};

    return settings;
},
```

```
    //设置值改变回调
    onSettingChange:async function(settingKey, settingValue, settingClass){
        //console.log("onSettingChange::" + settingKey + "_" + settingValue + "_" + settingClass);

    }

}
```

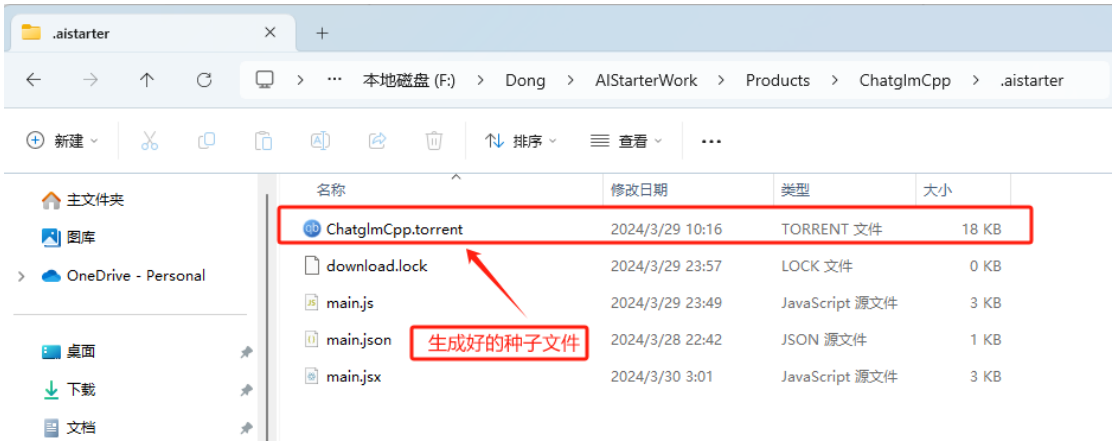
十、BT 做种

10.1 AIStarter 做种，需登录账号操作：点击分享后，AIStarter 启动器将自动创建脚本并压缩 AI 项目，生成 BT 种子文件。要创建和分享您的打包 AI 整合包，网络必须支持 IPv6，并且请保持 AIStarter 启动器处于运行状态。

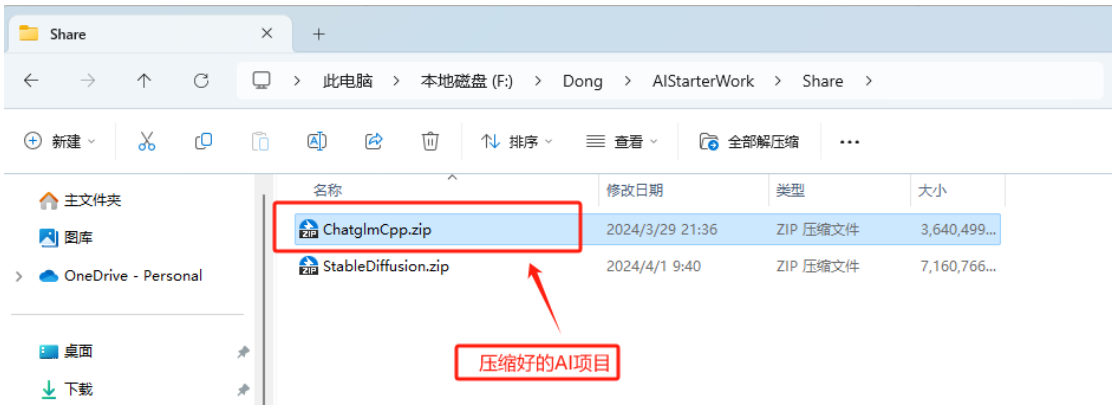
10.2 AIStarter 转种：当前网络环境不支持 IPv6，因此需要将原本在不支持 IPv6 的网络中的种子转移到支持 IPv6 的电脑上继续做种。具体操作步骤如下：

第一台电脑（不支持 IPv6）：

- 找到存放种子文件和压缩好的 AI 项目文件夹位置，路径如下所示：
  - 种子文件：`\AIStarterWork\Products\ChatglmCpp\.aistarter`



- AI 项目：`\AIStarterWork\Share`



说明：

1. `\.aistarter` 和 `\Share` 是种子文件和 AI 项目的相对路径，表示它们的路径。

2. 当使用这个相对路径时，假定种子文件和 AI 项目位于当前工作目录的根目录下。

3. 请确保在使用相对路径时，当前工作目录是正确的，以便程序能够正确找到种子文件夹。

4. 复制 AIStarter 生成的种子文件和 AI 项目。

第二台电脑（支持 IPv6）：

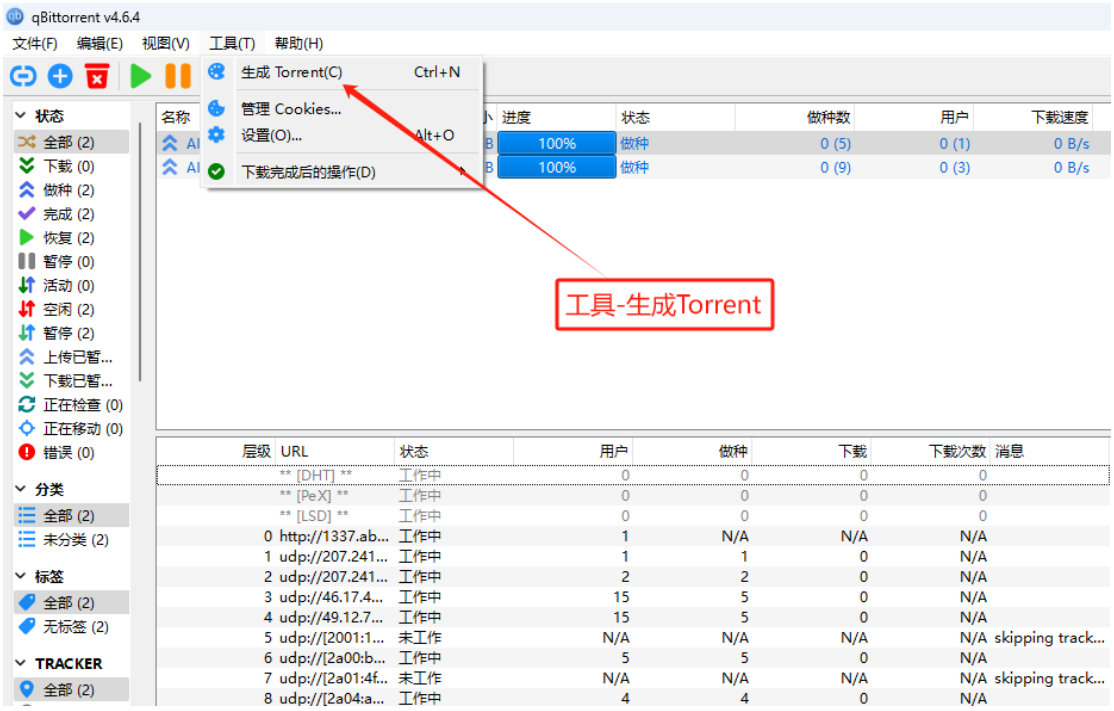
- 安装 AIStarter 启动器软件。

- 将从第一台电脑复制的种子文件和 AI 项目粘贴到第二台电脑上对应的 .aistarter 和 Share 文件夹位置。
- 保持 AIStarter 启动器运行，这样就可以正常进行种子做种。

通过以上步骤，你就能够将种子从不支持 IPv6 的电脑转移到支持 IPv6 的电脑上，确保做种的连续性和稳定性。

10.3 您也可以通过第三方 BT 下载工具做种，以 qBittorrent 为例，做种步骤如下：

第一步：打开 qBittorrent 软件，点击工具-选择生成 Torrent



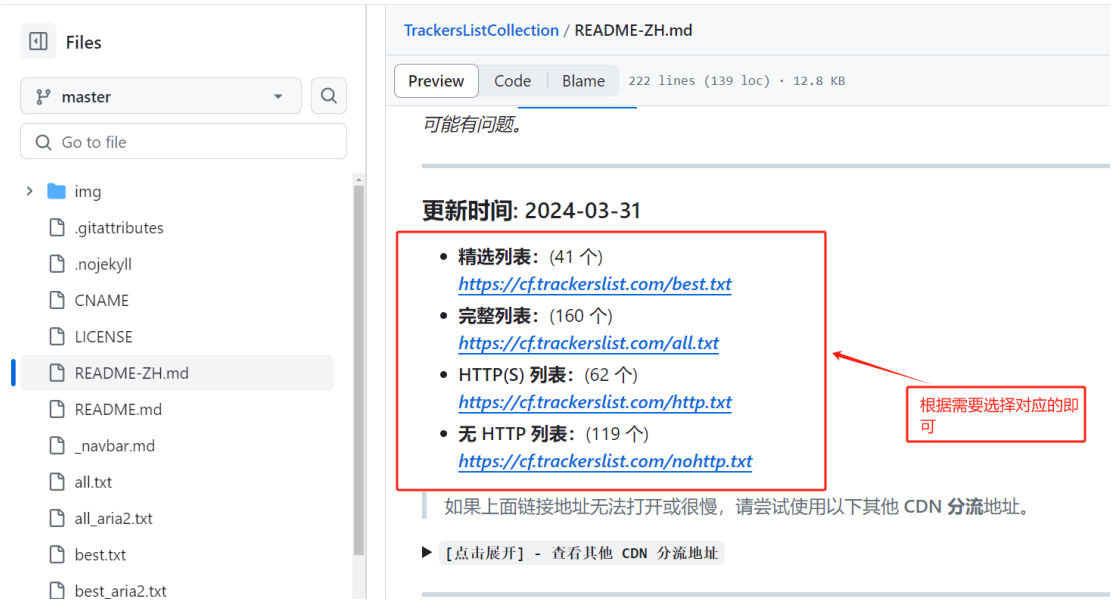
第二步：

1、选择要做种的文件或文件夹；

2、勾选完成后开始做种；

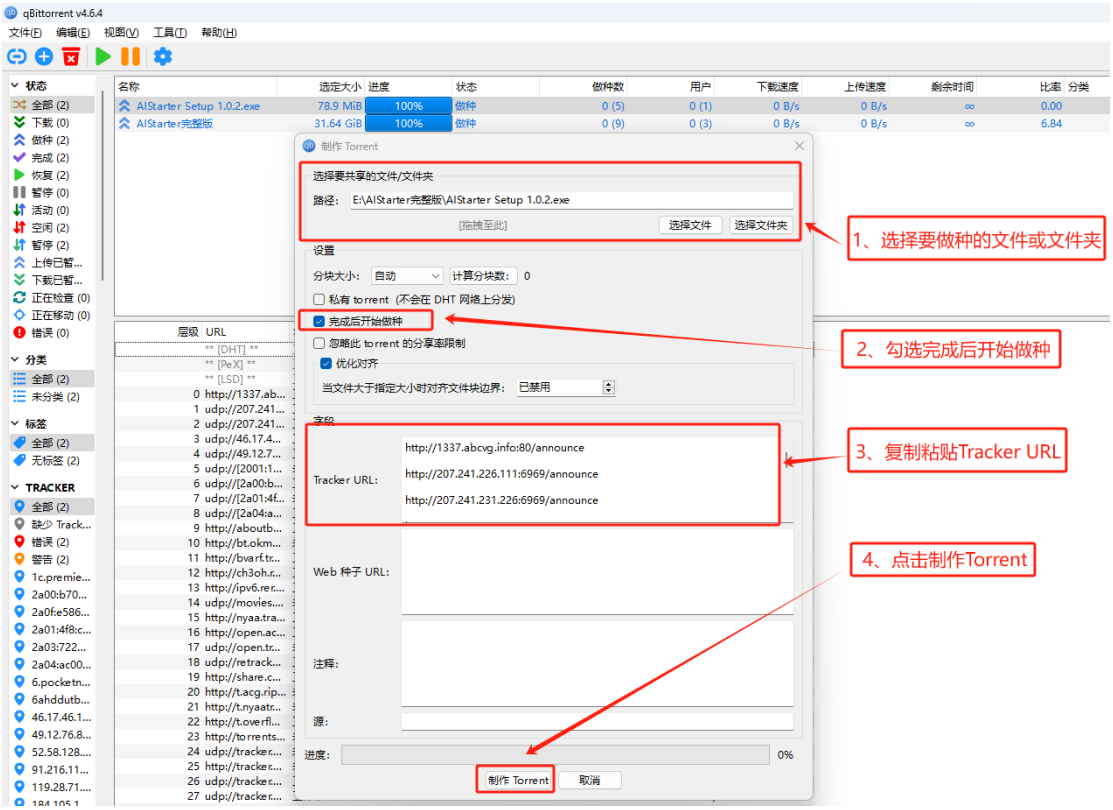
3、复制粘贴 Tracker URL；

这里有每天更新的 BitTorrent Tracker 列表：



https://github.com/XIU2/TrackersListCollection/blob/master/README-ZH.md

4、制作 Torrnet；



5、完成做种：一旦您的种子文件制作完毕，即可发布。请确保您的电脑网络支持 IPv4 公网或 IPv6 网络，并在每次开机时运行 qBittorrent 软件，以确保种子下载

的顺利进行。

十一、常见问题

11.1 AIStarter 启动器怎么获取？是否需要付费？

答：AIStarter 启动器是免费下载和使用的软件，并未授权任何商家进行收费。我们强烈建议您通过正规途径获取启动器，切勿通过其他渠道支付费用。如果您通过其他渠道支付费用获得了本软件，请立即退款并投诉相应商家。

11.2 安装点击一直转圈圈怎么办？

答：在正常情况下安装时通常不会出现这个问题，但如果您选择离线下载安装，可以参考 3.3.3 节的 AI 项目离线下载说明。

11.3 提示安装 cuda 怎么办？

答：当安装过程中出现提示安装 CUDA 时，这意味着您的系统需要 CUDA 支持。N 卡的用户，请根据自己显卡版本下载相应的 CUDA。您可以参考以下相关教程：

1、windows10 版本安装 CUDA 图文教程

2、CMD 窗口命令安装 CUDA | Windows 安装教程

请注意，安装 CUDA 可能需要管理员权限，并且可能需要重新启动系统才能生效。

11.4 AI 应用市场点击下载没反应怎么办？

答：通过 AIStarter 下载或分享的 AI 项目，您的电脑网络必须支持 IPv6 网络。如果您的网络不支持 IPv6，您可以参考第 3.3.3 节的 AI 项目离线下载说明进行操作。

十二、交流与支持

12.1 点击“交流”按钮，您将直接进入 AIStarter 的交流社区，您可以在这里向我们提供反馈和建议。此外，您还可以通过官方公布的 QQ 群、微信群、Telegram、Twitter、Discord 等即时线上交流工具加入 AIStarter 大家庭，参与讨论和交流。

12.2 点击“支持”可以直接赞助作者一杯咖啡，您也可以点击更多支持方案。AIStarter 启动器的持续发展和提升离不开您的支持和捐助。作为一款免费平台，我们依赖用户的慷慨捐款来推动产品的开发和提供更好的用户支持。

12.3 如果您对 AIStarter 启动器感到满意并且认为它对您有所帮助，我们诚挚地邀请您考虑捐赠，了解更多赞助方式。您的捐款不仅将激励我们持续改进产品，还将有助于确保我们能够为您提供更出色的服务。

感谢您对 AIStarter 启动器的支持！我们将竭尽全力确保您的捐赠产生最大的价值，并使我们能够不断改进和完善我们的平台，以满足您的需求和期望。

十三、用户协议

13.1 感谢您选择使用 AIStarter 启动器。启动器的 AI 项目整合包含了基于 GitHub 上开源的 AI 项目，同时也包括用户创建并分享的 AI 整合包。热门的 AI 项目包括 AI 绘画、AI 视频、AI 换脸、AI 对话、StableDiffusion、ComfyUI、SoftVcVits 等。我们不断更新这些整合包，以提供更多的 AI 项目选择。我们的目标是为 AIGC 技术学习提供一个便捷的算法运行环境，让我们共同努力支持 AI 开源项目的发展。为确保您的权益和顺利使用本软件，我们特此发布以下用户协议，请您仔细阅读：

您同意严格遵守法律法规，不得使用本软件从事任何违法活动，包括但不限于：

* 反对宪法所规定的基本原则；
* 危害国家安全，泄露国家秘密，颠覆国家政权，破坏国家统一；
* 损害国家荣誉和利益；
* 煽动民族仇恨、民族歧视，破坏民族团结；
* 破坏国家宗教政策，宣扬邪教和封建迷信；
* 散布谣言，扰乱社会秩序，破坏社会稳定；
* 散布淫秽、色情、赌博、暴力、凶杀、恐怖或教唆犯罪；
* 侮辱或诽谤他人，侵害他人合法权益；
* 实施任何违背“七条底线”的行为；
* 含有法律、行政法规禁止的其他内容。

十四、免责声明

14.1 使用条款

- 使用 AIStarter 启动器即代表您已详细阅读并同意以下使用条款。任何由使用 AIStarter 启动器产生的数据的产生、收集、处理、使用等相关事项的违法违规行为均由您自行承担责任，与 AIStarter 启动器无关。所有使用 AIStarter 启动器的行为和后果均由用户个人承担。

14.2 解释权和修改权

- AIStarter 作者保留最终的解释权和修改权，以最新的修改为准。使用者使用 AIStarter 启动器即视为同意并接受 AIStarter 作者对用户协议的解释和修改。

14.3 法律责任

- 违反本使用条款可能导致 AIStarter 启动器使用权被终止，并可能承担相应的法律责任。通过使用 AIStarter 启动器即代表您已阅读、理解并同意本使用条款。

感谢您的理解与合作。