

百度Comate插件安装使用手册

目录

- 0. 简介
- 1. 安装插件
 - 1.1 VSCode
 - 前置条件
 - 安装方法
 - 1.2 IntelliJ
 - 前置条件
 - 安装方法
- 2. 设置Key
 - 有关key的说明
- 3. 功能介绍
 - 3.1 单行推荐
 - 3.2 多行推荐
 - 3.3 行间提示
 - 3.4 注释生成代码
 - 3.5 增强生成代码
 - 3.6 生成单元测试
 - 3.7 代码生成doc注释
 - 3.8 生成代码
 - 3.9 代码解释
 - 3.10 生成行间注释
 - 3.11 长函数拆分
- 4. FAQ
 - 我应该怎么去使用这个插件
 - 我能够试用多久
 - 我觉得推荐的内容不准，怎么回事
 - 为什么推荐出来的内容有时候甚至无法编译
 - Comate与其它类似的产品比较起来怎么样
 - HumanEval和/或mbpp的评测结果怎么样
 - 是否会上传代码？上传哪些代码？
- 5. 插件下载

与内部版本不同，外部版本的插件在正式公布前，不考虑在插件市场上上线。两个版本的插件请使用文档附件下载对应IDE的插件。

0. 简介

百度Comate是一个利用大语言模型（LLM）实现的编码辅助工具。在编码过程中，Comate能够根据用户当前在编写的文件，推理出接下来可能的输入。当用户认为推理出的代码内容合适，可以使用快捷键tab采纳，采纳的内容就会被追加到光标后；如果认为内容不合适，则可以用ESC取消。

当前Comate支持的语言/框架如下：

1	分类	支持的语言
2	后端	C、C++、Java、Python、Go、PHP、Rust
3	前端	HTML、JavaScript、CSS、Less、swan、San、Vue、Scss、Sass、Stylus、pug、mermaid
4	App端	ObjectC、Kotlin、Swift
5	其它	Perl、Ruby、Shell、vhdl、GraphQL、Dockerfile

当前Comate支持的能力如下：

1	功能	描述	备注

2	单行、多行续写	根据上文内容，推理出接下来的一行、一段代码的内容，通常在实际开发中最常见的推荐，也是提效最明显的场景	
3	注释生成代码	根据特定的注释格式，生成一段代码。比较炫酷，但目前只支持JAVA、Python、CPP、Go等部分语言的函数注释	
4	生成单元测试代码	根据函数、方法的内容，生成相关的单元测试代码。支持Java、Python、Go、C/C++、JavaScript	
5	代码增加doc注释	给定一段代码，生成代码段的文档、函数注释	

1. 安装插件

1.1 VSCode

前置条件

- VSCode （vscode 1.70.x版本及以上）

安装方法

插件市场搜索 Baidu Comate，安装即可

1.2 IntelliJ

前置条件

- 最低版本：IDEA 2021.1.*

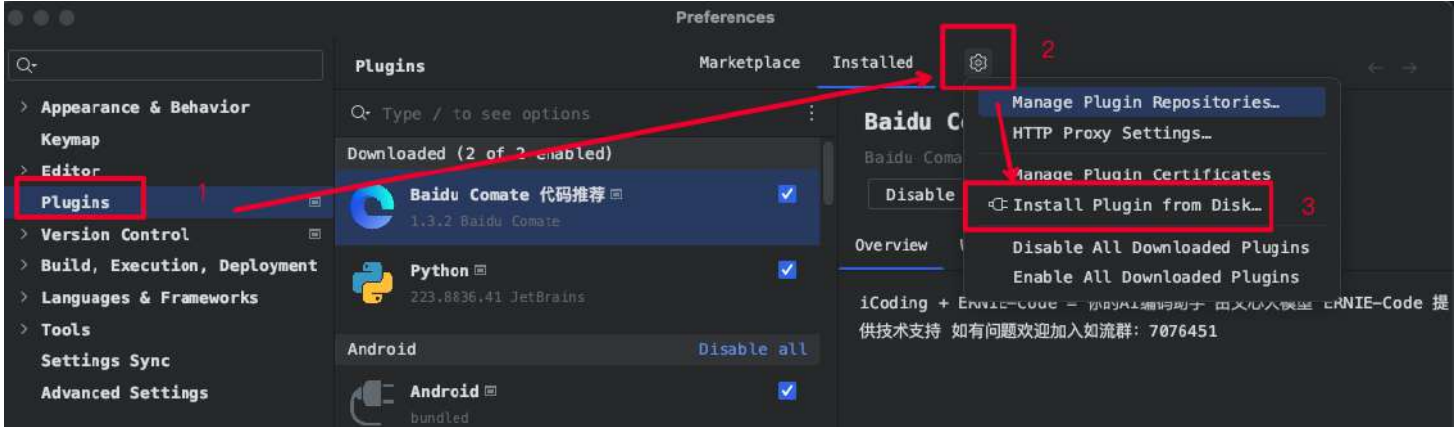
检查IntelliJ IDE的版本，以下图的IDEA为例，大版本为2022.2，所以Build版本对应下面插件下载的版本为222，同理如果大版本为2021.3，则对应的Build版本为213。

通常情况下，IDEA的版本类似2022.3.3这种格式，最后一位为小版本号无需关注，如上面的介绍，这个样例中对应的Build版本应为223。

以222版本为例，下载 v222_poc-comate-intellij-plugin-`{version}`.zip 文件，参照下面的安装方法安装和设置即可。

安装方法

Preferences -> plugins -> 右侧齿轮 -> Install plugin from disk... -> 选择插件文件。成功加载后需要重启IDE使用插件



2. 设置Key

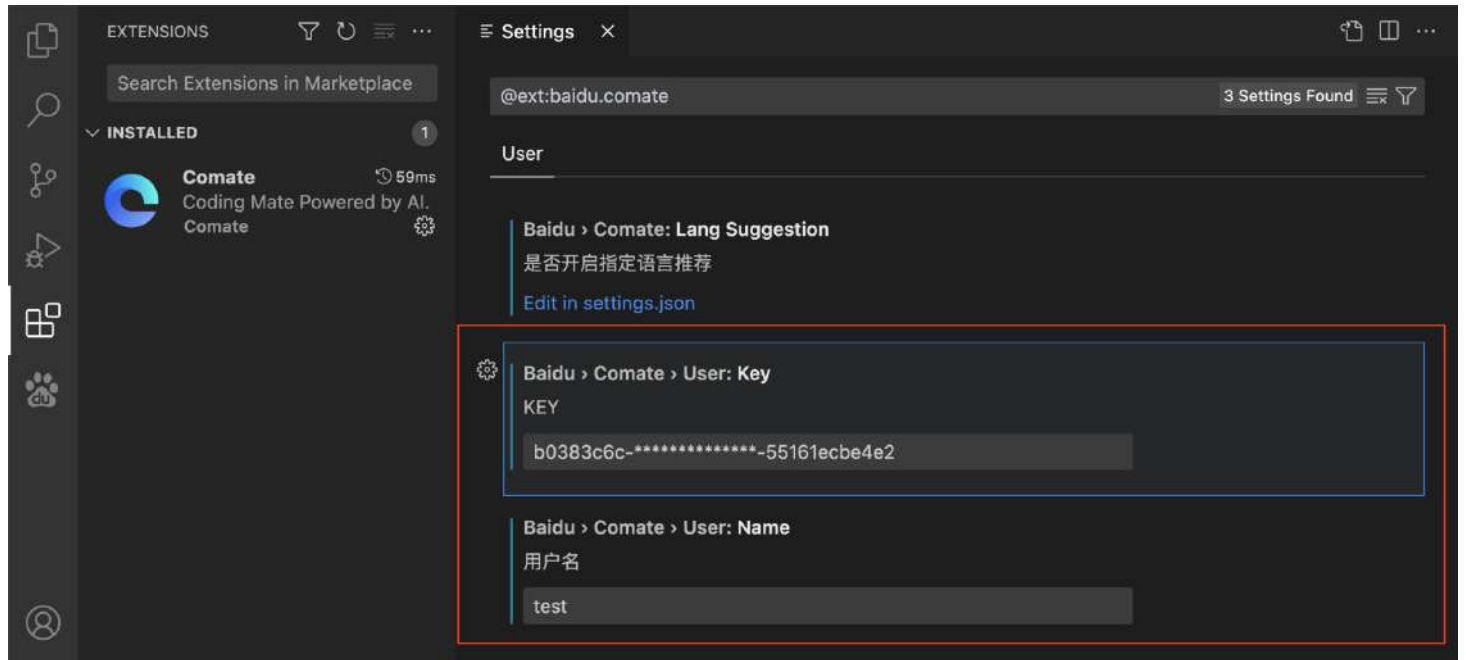
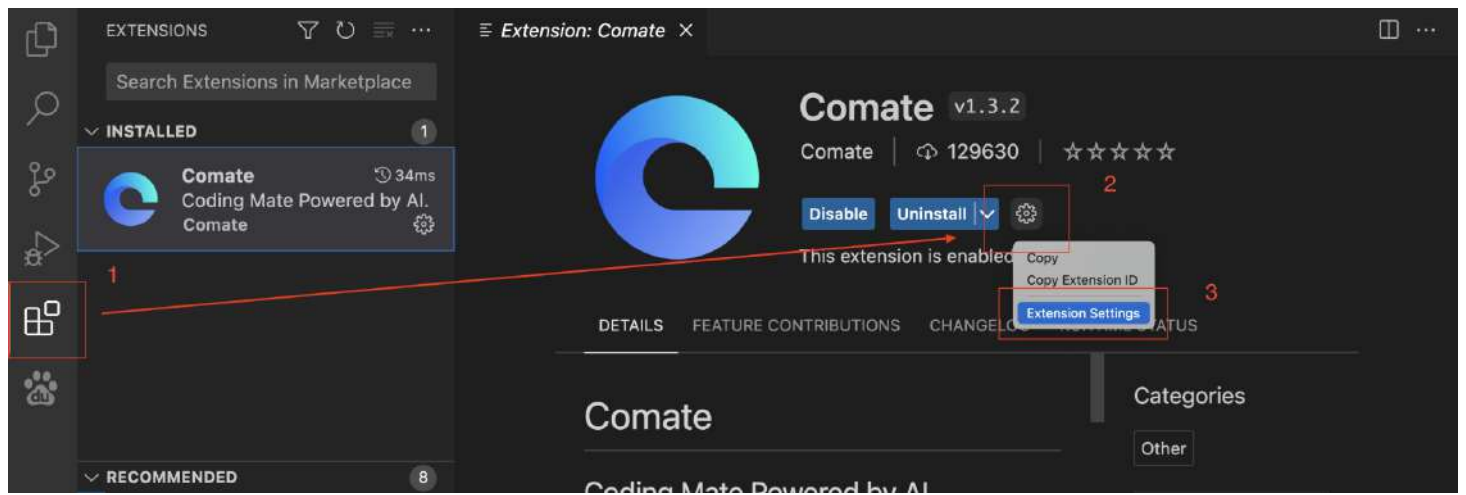
为了给用户提供更好的服务，当前的版本仅限邀测用户试用。系统会为每个邀测用户提供一个key。在插件成功安装好后，需要填入系统提供给用户的key。如果需要新的key，请与我们联系。

设置好key后，返回编码界面，开始编码，即可成功获取到推荐的内容了。

其它几个字段含义及设置：

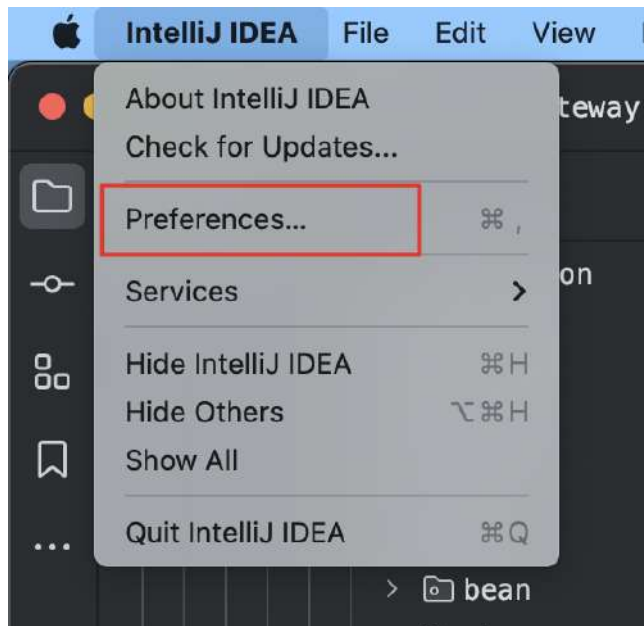
- Name：Comate会尝试获取当前登录的用户名，通过用户名来设置一些个性化推荐的配置，建议使用不和其他试用者冲突的名字已获得更好的使用体验

vscode 配置：Plugins -> 小齿轮 -> Extension Settings -> 输入 KEY、Name

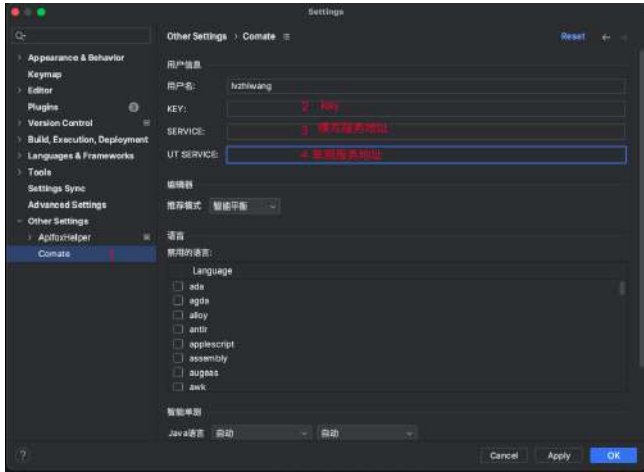


IntelliJ配置：Preferences -> Tools -> Comate -> 输入 KEY 、 Name以及Service

1. 菜单进入Preferences （或Settings）



2. 左侧找到Comate，点击后在右侧设置key



有关key的说明

为了控制整体服务的可用性，避免用户使用体验受到影响，服务端可能会监控每个key的请求次数。对于单个key请求过多的场景，推荐服务保留动态调整调用次数限制。超过单日请求次数上限的key可能无法继续使用推荐功能，但请放心，单日请求的上限设置一定是合理的，正常使用的用户不会受到限制。

3. 功能介绍

3.1 单行推荐

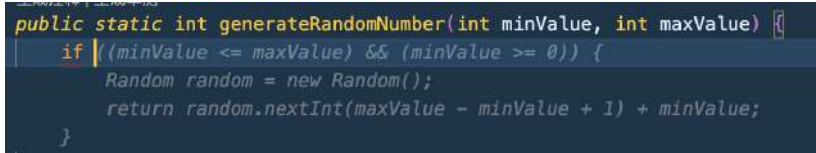
单行推荐是在续写代码场景中最直接、最高效的体现方式。在打开一个代码文件后，开始编码。在编码过程中稍微等待一下，即可看到Comate的推荐内容。如果觉得推荐内容不错，即可使用 `tab` 键采纳，否则可以使用 `ESC` 键取消推荐。



如果是第一次使用Comate，Comate也会在推荐内容后展示出快捷键的提示。

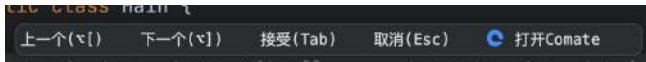
3.2 多行推荐

多行推荐与单行推荐的使用方式一致。在符合多条推荐的条件下（如for循环、if判断等），模型会优先计算一次多行推荐的逻辑。如果逻辑完整，则会展示出多行推荐的结果。否则还是按照单行推荐的逻辑来展示。



3.3 行间提示

在出现推荐内容后，鼠标移动到推荐内容上则会出现行间提示内容。行间提示主要用来使用更多能力，如多条推荐、触发其它动作等。



3.4 注释生成代码

注释生成代码是针对一段注释内容，生成相关的代码片段。适用于需求能够简单使用一句话描述清楚，或常见的算法片段的生成。

注意：由于算力限制，注释生成代码的耗时较长，请合理使用和评估。

```
public class Main {
    // 快速排序
    static int quickSort(int[] nums, int left, int right) {
        if (left < right) {
            int mid = (left + right) >> 1;
            quickSort(nums, left, mid - 1);
            quickSort(nums, mid + 1, right);
            swap(nums, left, mid, right);
        }
        return left;
    }

    public static void main(String[] args) {
    }
}
```

3.5 增强生成代码

如果在上面的模式中觉得生成的代码不够准确理解你的意思，可以使用这个功能调起更加精准的模型来理解和生成代码。

注意：由于算力限制，耗时较长，请合理使用和评估。

- 1. 触发方式同注释生成代码，待生成好后，出现“增强生成”按钮

```
public class Main {
    // 生成随机数
    random random = new Random();

    public static void main(String[] args) {
        System.out.println("The random number is " + random.nextInt(10));
    }
}
```

- 2. 点击“增强生成”，调出Comate面板
- 3. 等待返回

COMATE: 智能编码伙伴

2

Mr. Ryan

// 生成随机数

Comate

以下是Java语言生成随机数的完整方法：

java复制

import java.util.Random;

public class RandomNumberGenerator {
 public static void main(String[] args) {
 // 生成随机数
 Random random = new Random();
 int randomNumber = random.nextInt();
 System.out.println("随机数为: " + randomNumber);
 }
}

该方法使用Java内置的Random类来生成一个随机整数。调用nextInt()方法将返回一个介于0和Integer.MAX_VALUE之间的随机整数。

🔄 重新生成

你可能还需要：

UT 为选中的代码生成单元测试

📖 为选中的代码生成注释

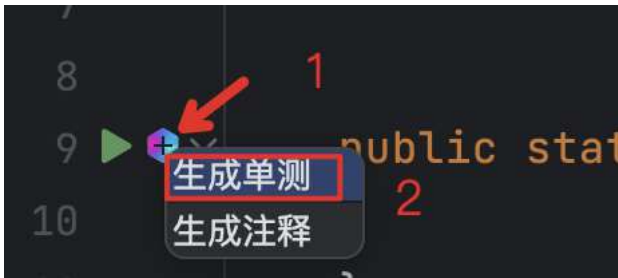
请输入描述来生成代码，可通过⌘+Enter换行

3.6 生成单元测试

生成单元测试，是针对一段函数（方法），生成其对应的单元测试代码。

触发方式：生成单元测试有三种方式：

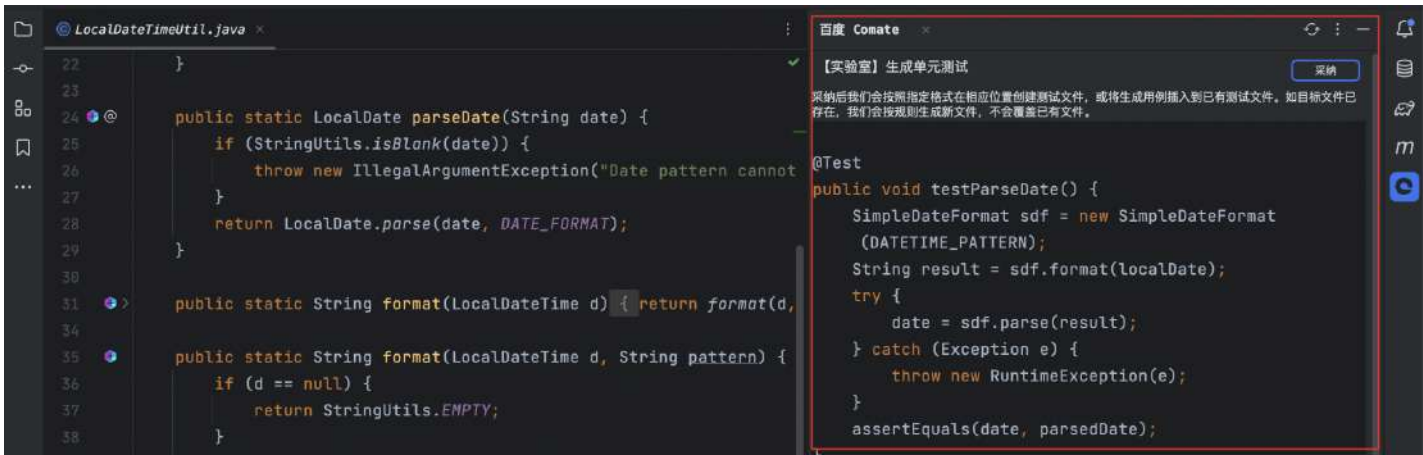
- 在函数（方法）签名头，点击Comate图标，选择“生成单测”



- 在函数（方法）中，鼠标右键点击，选择“生成单测用例-Comate”，



- 使用快捷键“Ctrl + U”



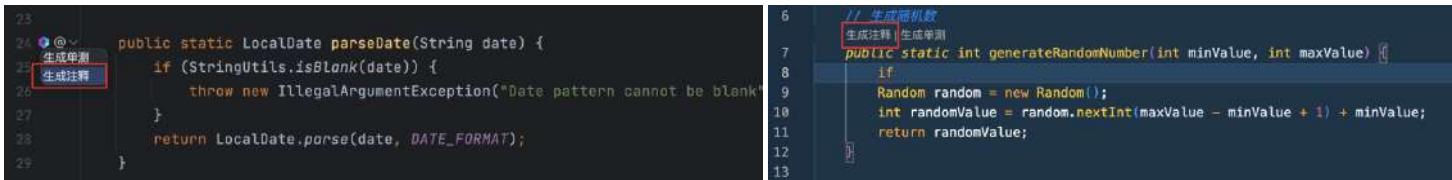
3.7 代码生成doc注释

生成doc注释，通过理解一个函数（方法）的内容，为该函数（方法）生成doc注释。

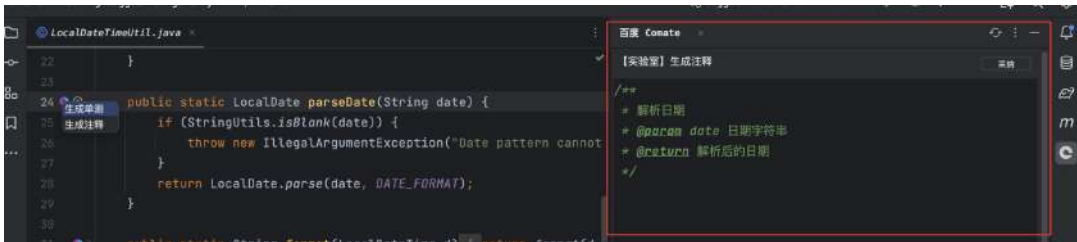
注意：由于算力限制，代码生成注释耗时较久，请合理使用和评估。

第一种触发方式：

- 鼠标移动到函数头部，选择Comate图标（IntelliJ）或函数上方的“生成注释”文字（VSCode）

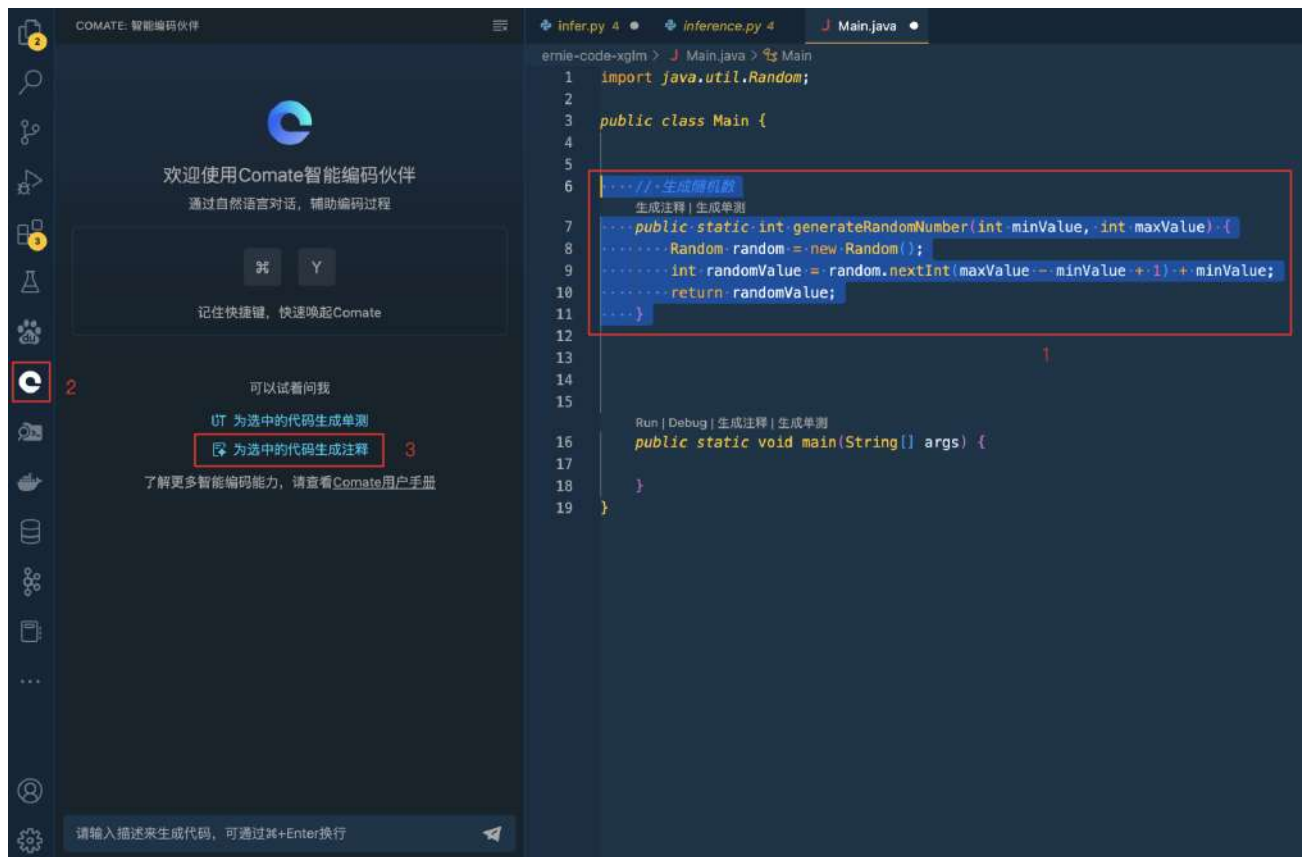


- 等待返回，可以使用“采纳”功能，将注释内容插入到函数头部

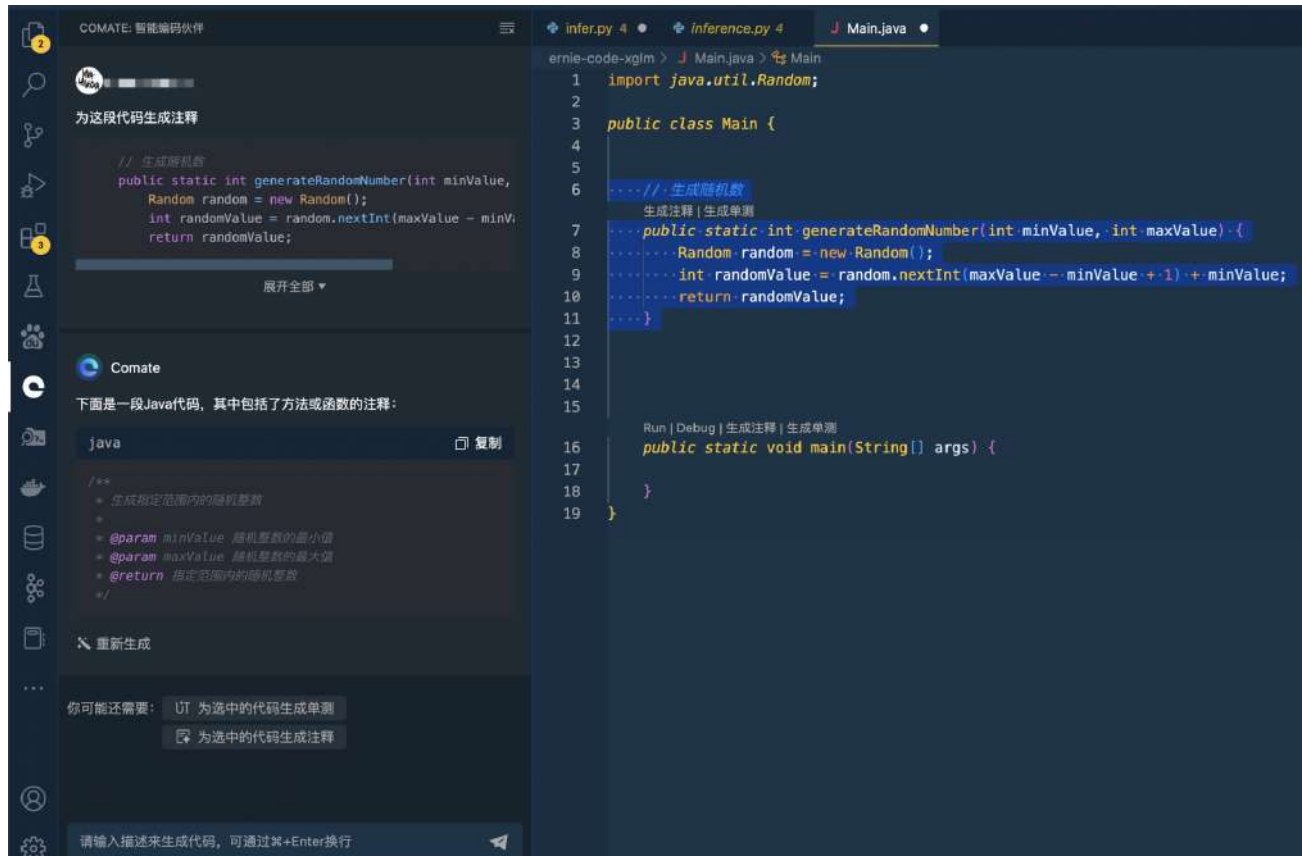


第二种触发方式：

- 选中一段代码后打开Comate面板
- 使用Comate面板，选择“为选中代码生成注释”



3. 等待返回



3.8 生成代码

根据一段通过自然语言的描述，生成一段代码。

注意：由于算力限制，生成代码耗时较长，请合理使用和评估。

使用方式：

1. 如果使用VSCode，按下Cmd + Y（Windows使用Ctrl + Y）呼出Comate面板， 如果使用IntelliJ，使用Cmd + Shift + Y（Windows使用Ctrl + Shift + Y）或使用侧边栏按钮打开Comate面板



2. 在面板下面的输入框中输入问题



3. 等待返回

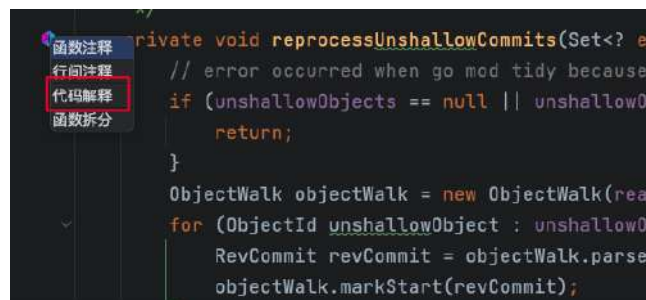


3.9 代码解释

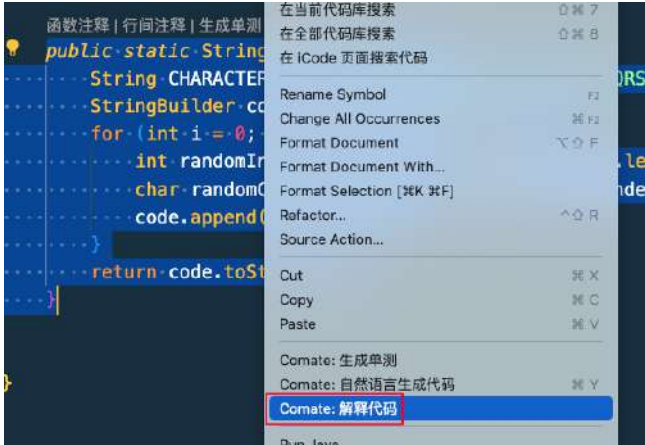
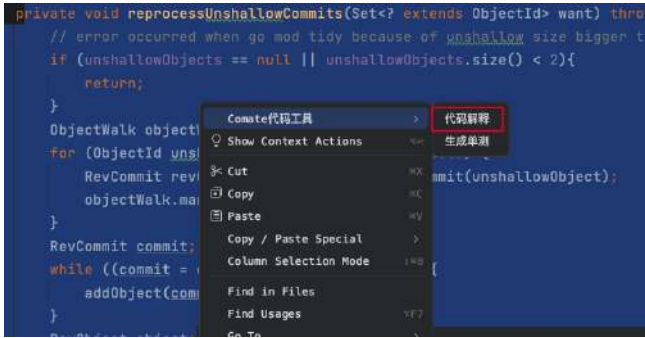
代码解释主要用于给定一段代码，通过大模型对于代码的理解，输出人类容易理解的解释。能够帮助开发者快速理解已有的代码。

代码解释的入口有以下两种：

- 函数 / 方法开头处的图标（IntelliJ）、或函数标题上方（VSCode）



- 选定一段代码 -> 鼠标右键 -> Comate代码工具 -> 代码解释(IntelliJ)、选定一段代码 -> 鼠标右键 -> Comate：解释代码



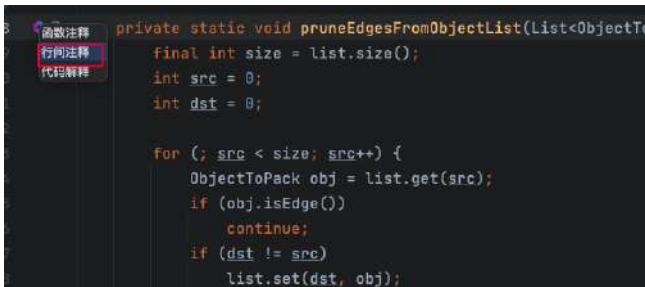
使用解释功能后，会针对代码内容做概括解释。解释的内容展现在侧边栏中。



3.10 生成行间注释

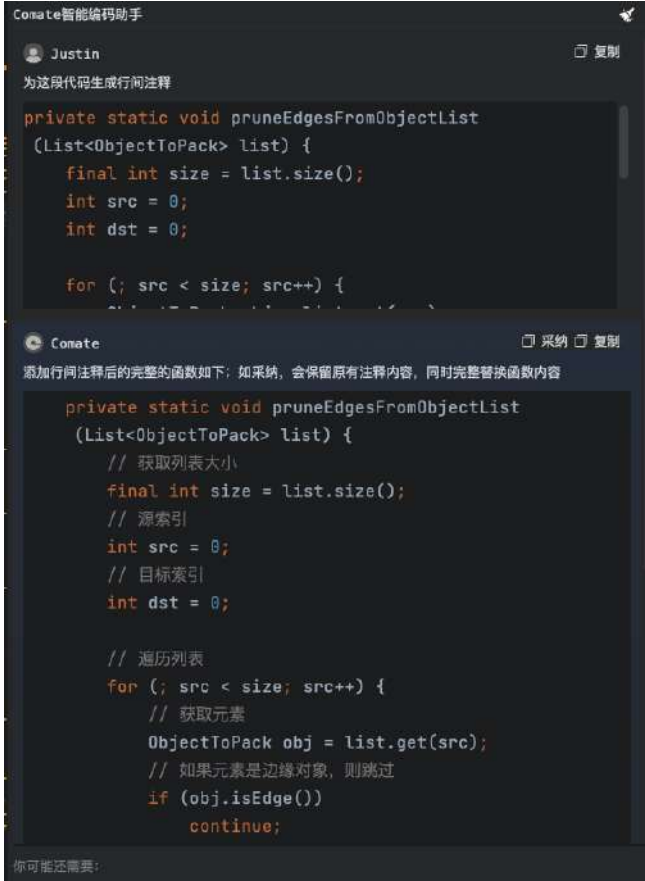
生成行间注释的功能，是针对一段已有的代码，按照代码内容，按段落生成行间注释的功能。

当前功能只支持从函数 / 方法开头处的图标（IntelliJ）、或函数标题上方（VSCode）触发。





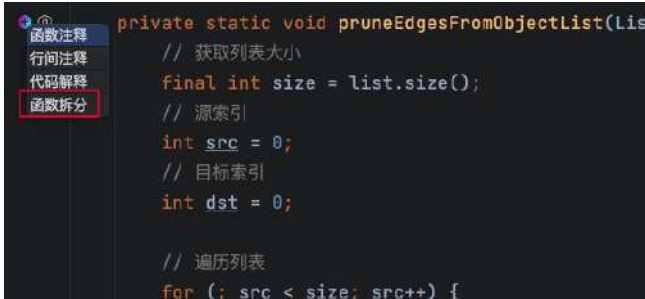
生成成功后，会在侧边栏中展示出针对代码片段生成的行间注释内容。可以对结果采纳或复制，采纳后，新增加的函数内容会覆盖到代码编辑区，也可以手工复制部分片段。



3.11 长函数拆分

长函数拆分的功能主要用于将一个较长片段的函数，拆分成多个短函数。

当前功能只支持从函数 / 方法开头处的图标（IntelliJ）、或函数标题上方（VSCode）触发。





生成功完成后，会在侧边栏中展示出针对代码片段生成的行间注释内容。可以对结果采纳或复制，采纳后，新增加的函数内容会覆盖到代码编辑区，也可以手工复制部分片段。

4. FAQ

我应该怎么去使用这个插件

真正的打开一个实际开发的项目，去编写代码就好。随着输入的内容越多，大模型的理解越深刻，推荐的内容也就越精准和有效了。

我能够试用多久

每个key的有效期是30天。

我觉得推荐的内容不准，怎么回事

首先大模型类产品本身的输出内容就存在一定的不确定性，其次模型自身对于上文的理解也千差万别。因此如果发现推荐内容不准的情况，我们的建议是继续写下去，随着Comate理解代码的意图越来越多，则会推荐出准确的内容了。

为什么推荐出来的内容有时候甚至无法编译

如之前所说，大模型输出内容本身具备不确定性。Comate可能识别了当前的内容在做某个事情。而基于它自身的学习，达成这个事情的最多、最佳实现是它推荐出的内容类似。Comate本身不会像IDE一样，根据当前项目自身的结构和文件来推荐合适的内容（当然这是Comate后续想要覆盖的方向）。

Comate与其它类似的产品比较起来怎么样

因为模型本身多样性比较多，实际上无法客观的介绍哪个效果更好，不如先用起来，只有用了才知道。真的借助Comate写一个功能，就会体会到在过程中的效果了。

HumanEval和/或mbpp的评测结果怎么样

我们确实也对模型做过HumanEval和其它评测集的各种测试，在早期我们的模型效果在同规模模型中领先，甚至超过部分更大规模的模型。但是随着公开的模型越来越多，也有越来越多的模型专门针对HumanEval等公开评测集做优化，导致评测集严重失真，甚至有2B参数模型HumanEval评分好于13B模型的情况发生。因此我们认为目前所有公开的评测集得分均不具备任何参考性。

为了能够量化模型的效果，我们内部本着公平和随机的原则，构建了一个规模更大，覆盖范围更广的评测集和评测体系。在我们内部的评测结果中，可以确定的是我们的模型评测结果都处于领先地位。

是否会上代码？上传哪些代码？

测试过程中需要利用正在编写的文件中的上文，来推理出接下来的下文。因此确实会需要通过网络将正在编写的代码文件内容传输给模型推理。我们承诺不使用这部分上传的代码做任何积累、分析、加工等后续动作。

但是在私有化部署的情况下，模型和服务都部署在客户私域内，数据不会流出，可以放心使用。

5. 插件下载

VSCode：

<https://marketplace.visualstudio.com/items?itemName=BaiduComate.comate>

Intellij：

敬请期待。