

# Fin-AIGC产品简介与AI辅助编程使用分享

## 一、Fin-AIGC产品介绍

介绍部门内AIGC相关产品

## 二、AI辅助编程在日常工作的使用分享

分享个人日常工作中使用到的AI辅助工具

Abwu吴志隆 / FIT-支付金融平台与数据部

负责Fit ABTest、Polar智能推荐、AIGC相关产品等中台系统的前端开发

# Fin-AIGC产品简介与AI辅助编程使用分享

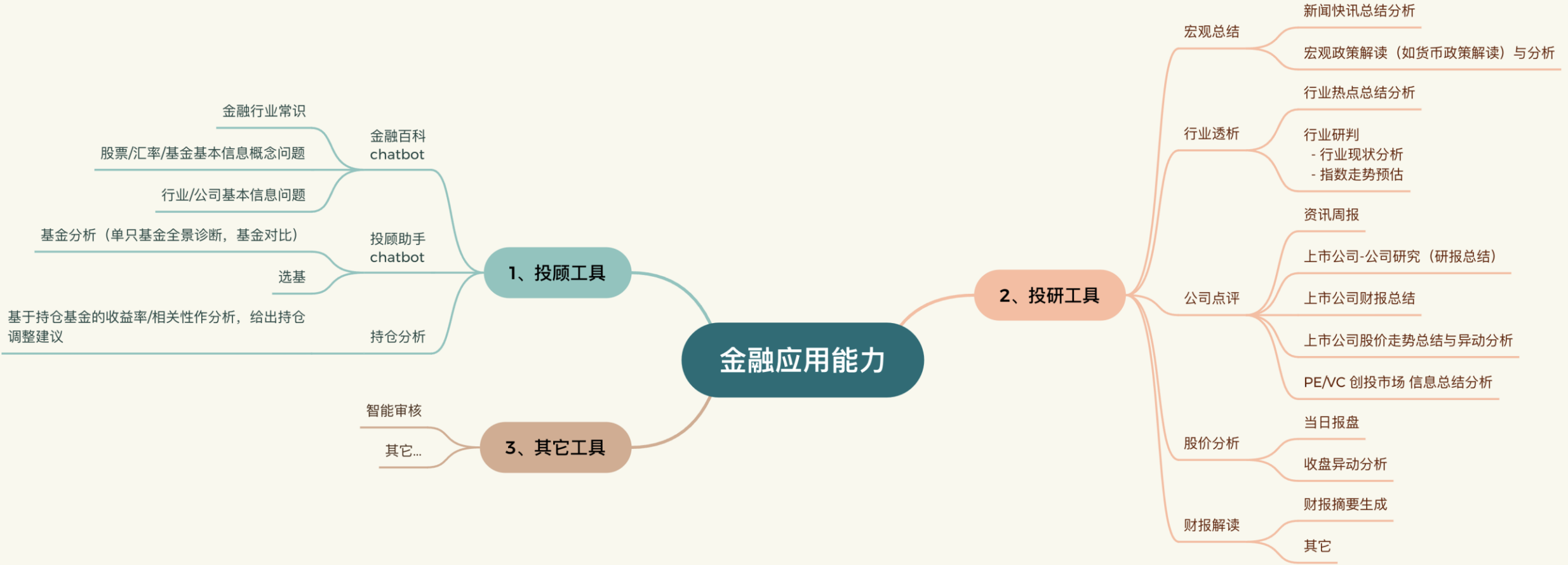
## 一、Fin-AIGC产品介绍

介绍部门内AIGC相关产品

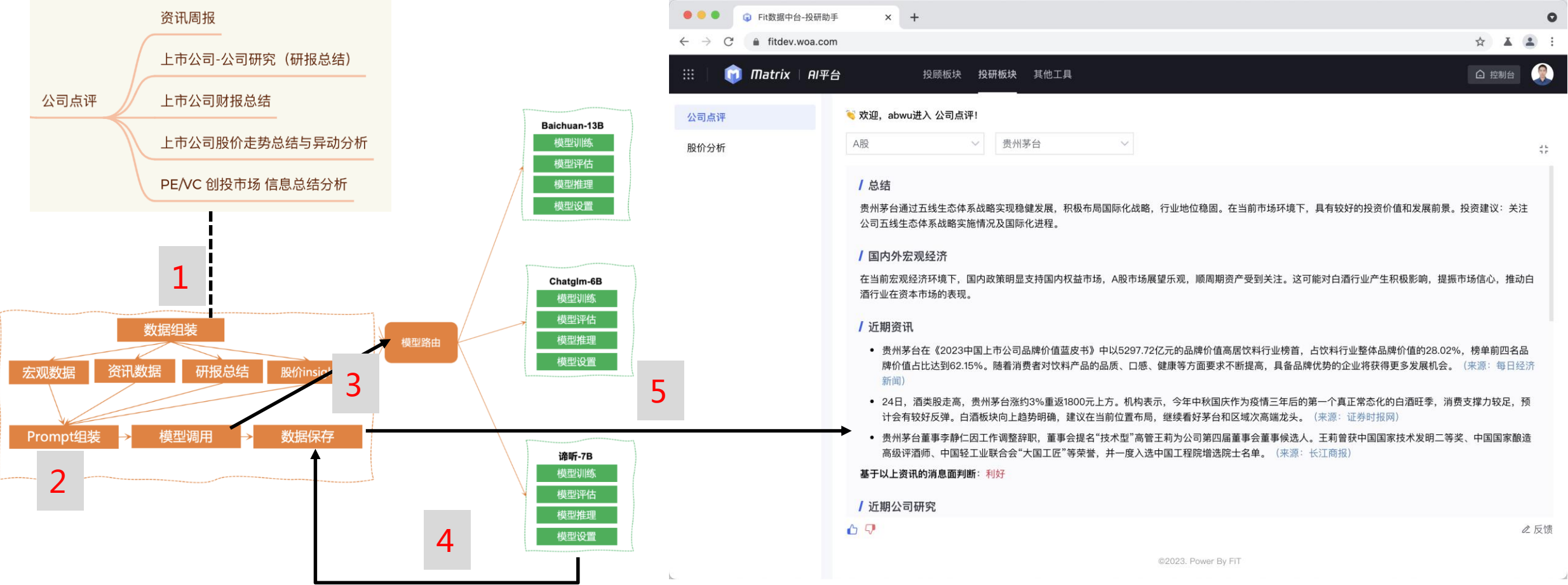
## 二、AI辅助编程在日常工作的使用分享

分享个人日常工作中使用到的AI辅助工具

# Fin-AIGC产品介绍



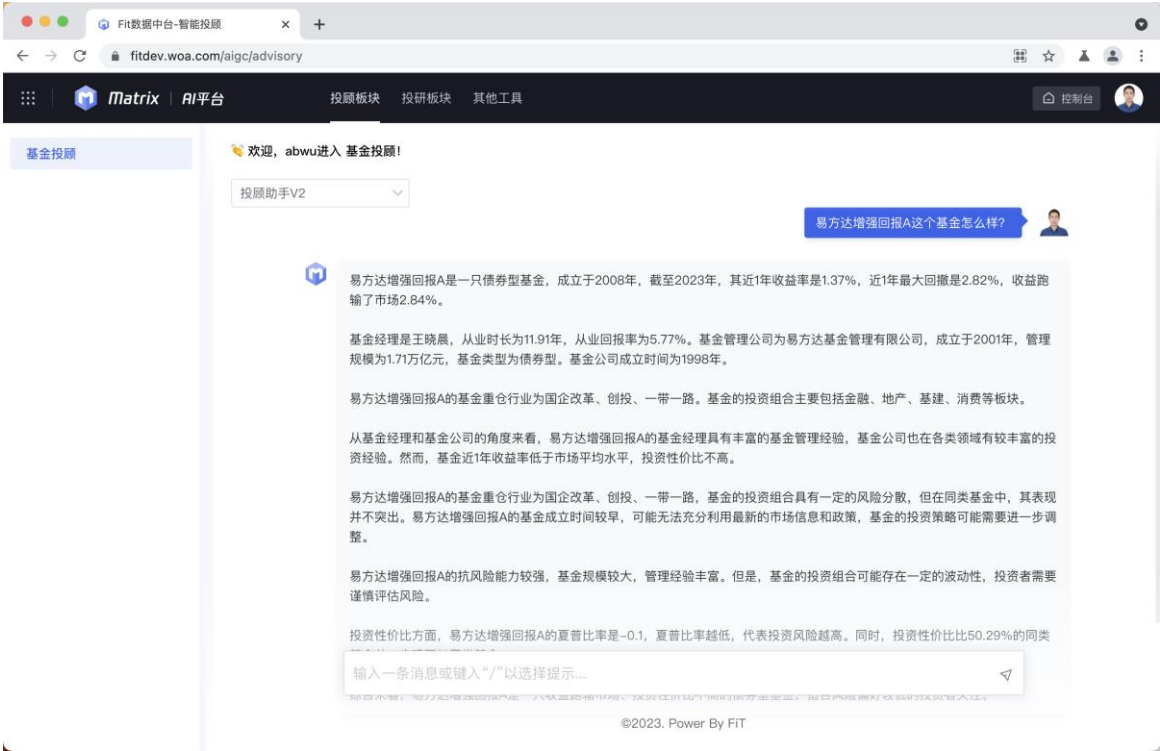
# Fin-AIGC产品介绍 - 公司点评



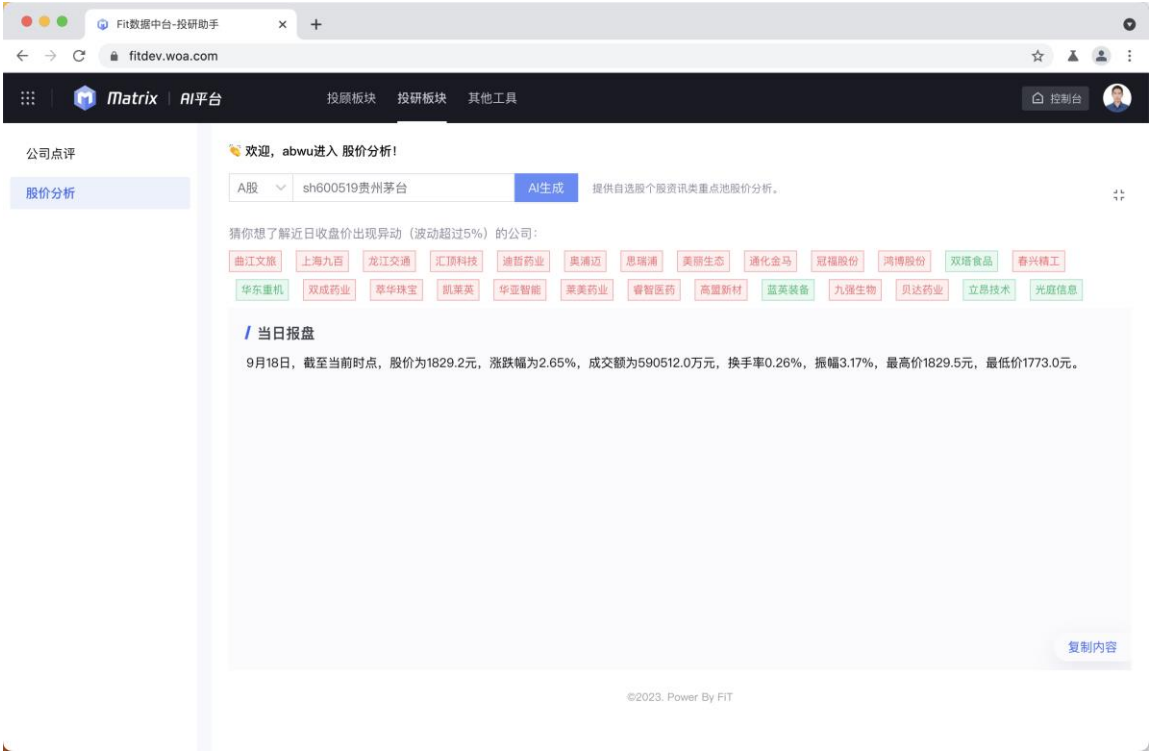
体验地址：  
<https://fit.woa.com/aigc/investment>

# Fin-AIGC产品介绍

## 投顾组手（基金分析、基金选择）

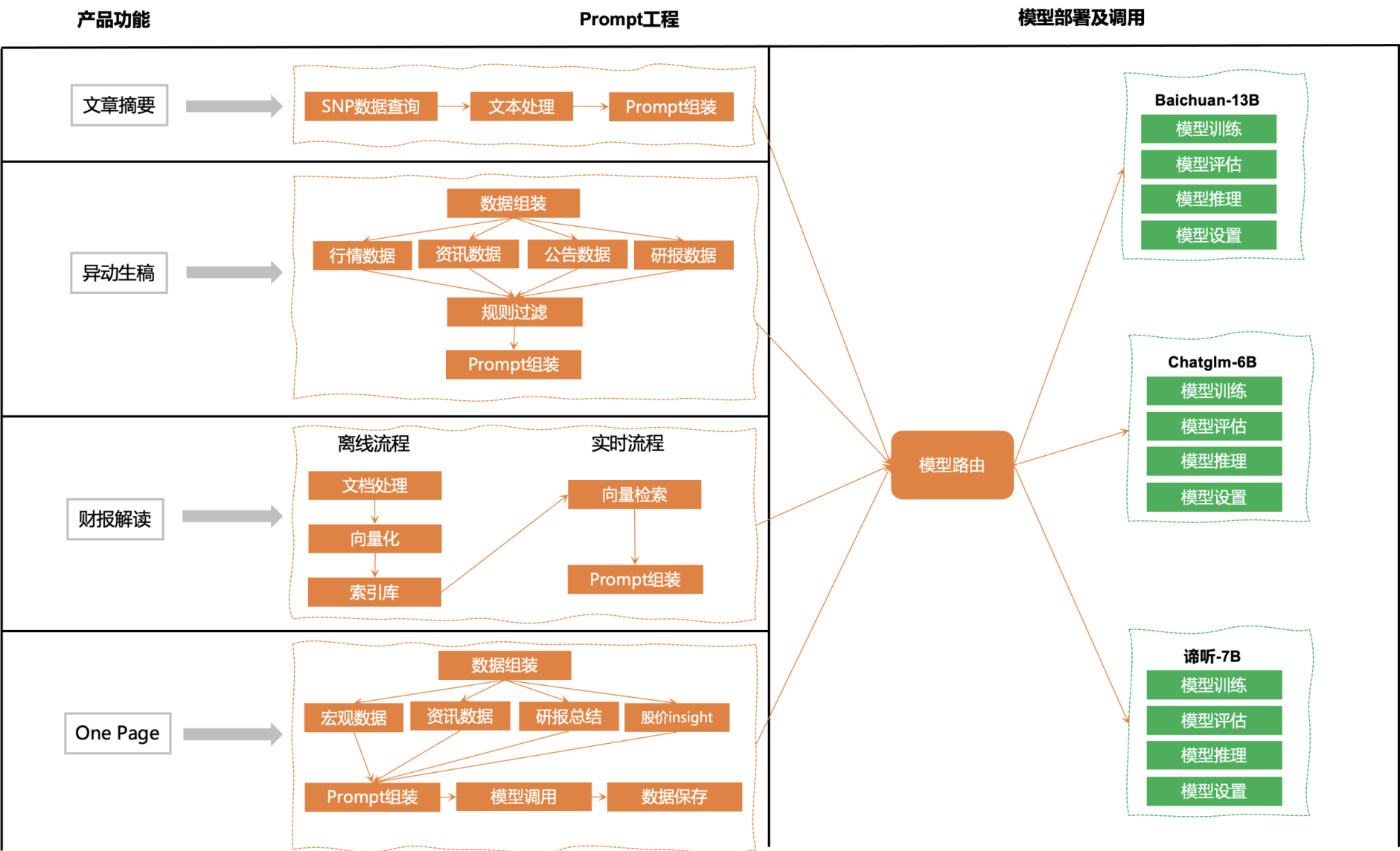


## 股价分析



体验地址：  
https://fit.woa.com/aigc/investment

# Fin-AIGC产品介绍



# Fin-AIGC产品简介与AI辅助编程使用分享

## 一、Fin-AIGC产品介绍

介绍部门内AIGC相关产品

## 二、AI辅助编程在日常工作的使用分享

分享个人日常工作中使用到的AI辅助工具 ( Github Copilot 和 ChatGPT )

# AI辅助编程使用分享

## 1、为什么要用？

- 效率、效率、效率

## 2、日常写代码中能帮我做哪些事情？

- 组件编写、代码优化、定位问题、用例生成、代码解释、生成备注

## 3、进阶的应用（落地的场景探索）

- 代码仓库理解 「chat to codebase」
- 设计稿 to 可用的代码 「design to code」

## 4、风险

- 数据安全和合规
- AI结果不一定完全准确



# AI辅助编程使用分享 -- 效率

- 1、获取关键信息效率更快、更直接（比Google/查文档更高效）
- 2、可以帮我处理重复的工作（带有上下文理解的推荐）

同样的问题询问 Google VS. ChatGPT

在设计vue组件中如何避免props drilling

图片

视频

购物

新闻

地图

图书

航班

财经

About 2,990 results (0.26 seconds)

稀土掘金  
https://juejin.cn › 前端  
props drilling(props钻取)  
Dec 15, 2021 — props drilling 是数据以只是传递的组件层级过深,而中间层组件

CSDN博客  
https://blog.csdn.net › details  
Vue 依赖注入使用Provide 和  
May 27, 2022 — 使用provide 和inject 的结构中一路传递prop,即使许多组件之间

CSDN博客  
https://blog.csdn.net › details  
Vue.js 3.x 中跨层级组件如何  
May 5, 2022 — 在 Vue.js 中,跨层级组件的数据传递给子组件: prop-drilling

GPT-4

BY

在设计vue组件中如何避免props drilling

避免props drilling (属性钻取) 是一个在Vue组件设计中非常重要的考虑因素。这种现象通常发生在多层嵌套的组件树中,某些属性需要从最顶层一直传递到底层组件。这不仅使得代码难以维护,还增加了组件之间的耦合性。

下面是一些方法来避免这种情况:

使用Provide/Inject

Vue提供了`provide`和`inject`这两个API,使得祖先组件能够向其所有子孙组件提供一个属性或方法,无需通过props一层一层地传递。

javascript

Copy code

```
// 祖先组件
export default {
  provide() {
    return {
      someData: this.someData
    };
  },
  data() {
    return {
      someData: 'I am some data'
    };
  }
};
```

而且 GPT 可以做得更好，提供上下文追问

GPT-4

开发背景,我推测你可能会更倾向于使用Vuex或者Provide/Inject方式。

BY

通过 Provide/Inject 方式,提供一个实际应用的例子

当然,这里是一个使用`provide`和`inject`的Vue3应用示例。在这个例子中,我们有一个祖先组件`Parent`,它提供一个用户名称(`userName`)给两个子孙组件`Child`和`GrandChild`。

Parent Component (祖先组件)

javascript

Copy code

```
// Parent.vue
<template>
  <div>
    <p>I am the Parent. My name is: {{ userName }}</p>
    <Child></Child>
  </div>
</template>

<script setup>
import { ref } from 'vue';
import Child from './Child.vue';
```

# AI辅助编程使用分享 -- 效率

- 1、获取关键信息效率更快、更直接（比Google/查文档更高效）
- 2、可以帮我处理重复的工作（带有上下文理解的推荐）

会根据上下文推出你想要的内容

```
3 export const useUserInfo = defineStore('userInfo', {
4   state: () => {
5     return {
6       username: '',
7       userId: '',
8     };
9   },
10  actions: {
11    setUsername(name) {
12      this.username = name;
13    },
14    setUserId(id) {
15      this.userId = id;
16    },
17  },
18 });
```

你可以ChatToCode，叫 AI帮你在指定模块内新增内容

```
3 export const useUserInfo = defineStore('userInfo', {
4   state: () => {
5     return {
6       username: '',
7       userId: '',
8       userChName: '',
9     };
10  },
11  actions: {
12    setUsername(name) {
13      this.username = name;
14    },
15    setUserId(id) {
16      this.userId = id;
17    },
18    setUserChName(name) {
19      this.userChName = name;
20    },
21  },
22 });
```

add userId and userChName state and actions

Accept Discard ↻

Changed 8 lines

# AI辅助编程使用分享 -- 日常工作中可以帮你做哪些事情？

写业务逻辑、组件编写、代码优化、定位问题、用例生成、代码解释、备注生成、等等

## 代码优化

find bugs and refactor the following code, should be more defensive, maintainable, readable, easy for unit testing, and best practices

```
requestInterceptorsResolve: config => {
  return new Promise(resolve => {
    if (config.tenantId) {
      if (!store.state.menus.fbusiness) {
        const timer = setInterval(() => {
          if (store.state.menus.fbusiness) {
            config.headers['X-Tenant-Id'] = store.state.menus.fbusiness;
            clearInterval(timer);
            resolve();
          }
        }, 500);
      } else {
        config.headers['X-Tenant-Id'] = store.state.menus.fbusiness;
        resolve();
      }
    } else {
      resolve();
    }
  });
}
```

问题1: 内嵌层级太深、可读性差

问题2: X-Tenant-Id 会被重复设置

问题3??: 在主函数里面使用 setInterval/clearInterval



```
// 辅助函数，用于等待某个条件得到满足
const waitFor = (conditionFunc, interval, timeout) => {
  return new Promise((resolve, reject) => {
    const timer = setInterval(() => {
      if (conditionFunc()) {
        clearInterval(timer);
        resolve();
      }
    }, interval);

    setTimeout(() => {
      clearInterval(timer);
      reject(new Error('在规定时间内条件未满足。'));
    }, timeout);
  });
};

const requestInterceptorsResolve = async (config) => {
  // 防御性编程: 检查 config 对象及其必要属性是否存在
  if (!config || !config.headers) {
    throw new Error('配置对象无效。');
  }

  const tenantId = config.tenantId;
  const fbusiness = store?.state?.menus?.fbusiness;

  if (tenantId) {
    try {
      // 使用辅助函数使代码更易维护和单元测试
      await waitFor(
        () => store?.state?.menus?.fbusiness !== undefined,
        500,
        5000 // 考虑一个超时，例如 5 秒
      );
    } catch (error) {
      // 记录错误并相应处理
      console.error(error);
      throw new Error('在规定时间内无法获取 fbusiness。');
    }

    config.headers['X-Tenant-Id'] = store.state.menus.fbusiness;
  }

  return config;
}
```

关键说明:

- 1. 辅助函数: 添加了 'waitFor' 辅助函数，以便在规定的时间内检查某个条件。这使代码更易维护，并方便单元测试。
- 2. Async/Await: 把 'requestInterceptorsResolve' 函数设为 'async'，以便更容易理解。
- 3. 防御性编程: 加入了检查 'config' 对象及其 'headers' 属性的有效性，以避免运行时错误。
- 4. 超时: 在 'waitFor' 函数中添加了超时，以确保它不会无限期地运行，如果条件永远不满足。
- 5. 错误处理: 加入了错误处理来更优雅地捕获和处理错误。
- 6. 代码简洁: 通过重构去掉了重复的代码行。
- 7. 可选链: 使用可选链 ('?.') 访问深层嵌套的对象属性。这样可以防止运行时错误，如果其中一个属性是 'undefined' 或 'null'。
- 8. 单一责任: 代码的每一部分都有单一的责任，使其更易读和维护。

这样应该会让代码更易维护、防御性更强，以及更易进行单元测试。

通常你给的限定词越多，它越能根据你的需求做出优化

注意：避免过度优化！

# AI辅助编程使用分享 -- 日常工作中可以帮你做哪些事情？

写业务逻辑、**组件编写**、代码优化、定位问题、用例生成、代码解释、生成备注、等等

**组件编写** ( AI平台70%的代码都是AI生产 )



第一步：需要你把组件拆分



extra-star.vue

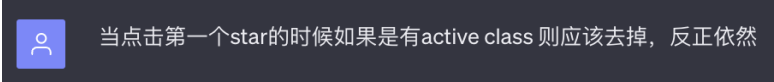
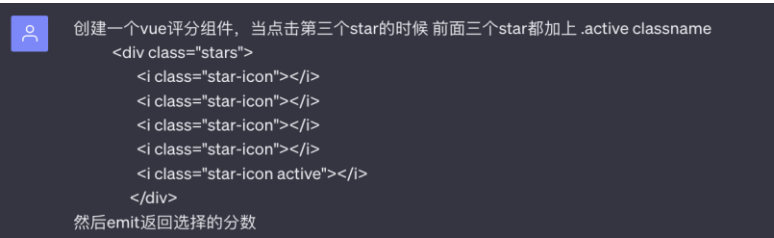
内容无出处 消息面情感判断有误

请输入

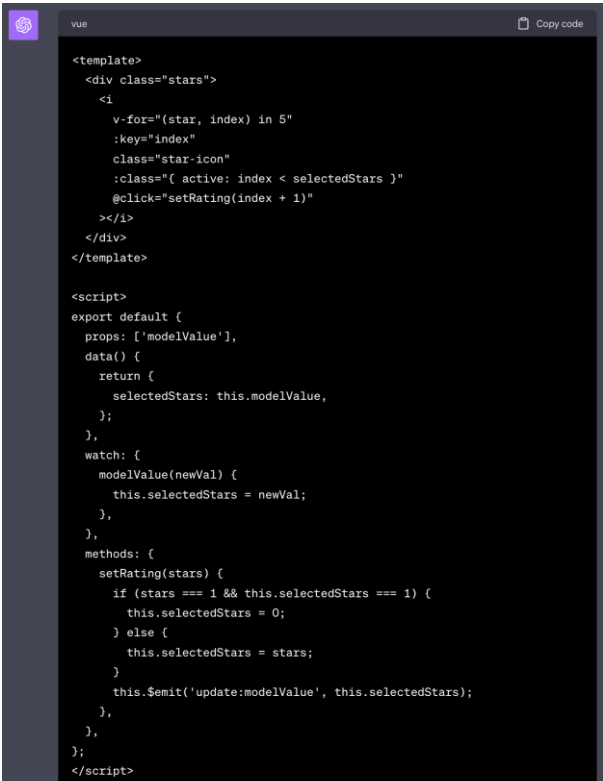
feedback.vue



第二步：把你想要的功能描述给GPT ( 你描述得多清楚，GPT就可以帮你实现得多准确 )



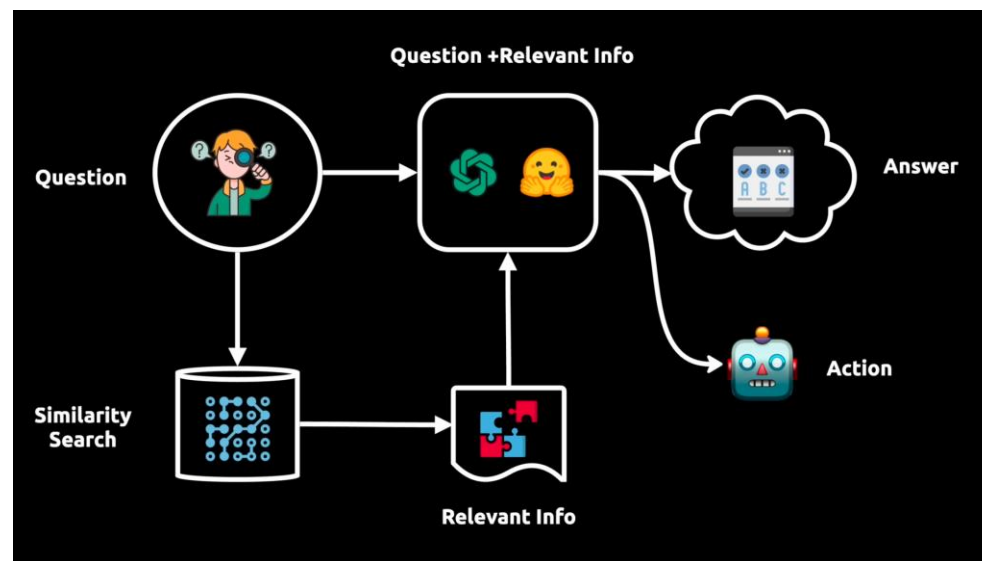
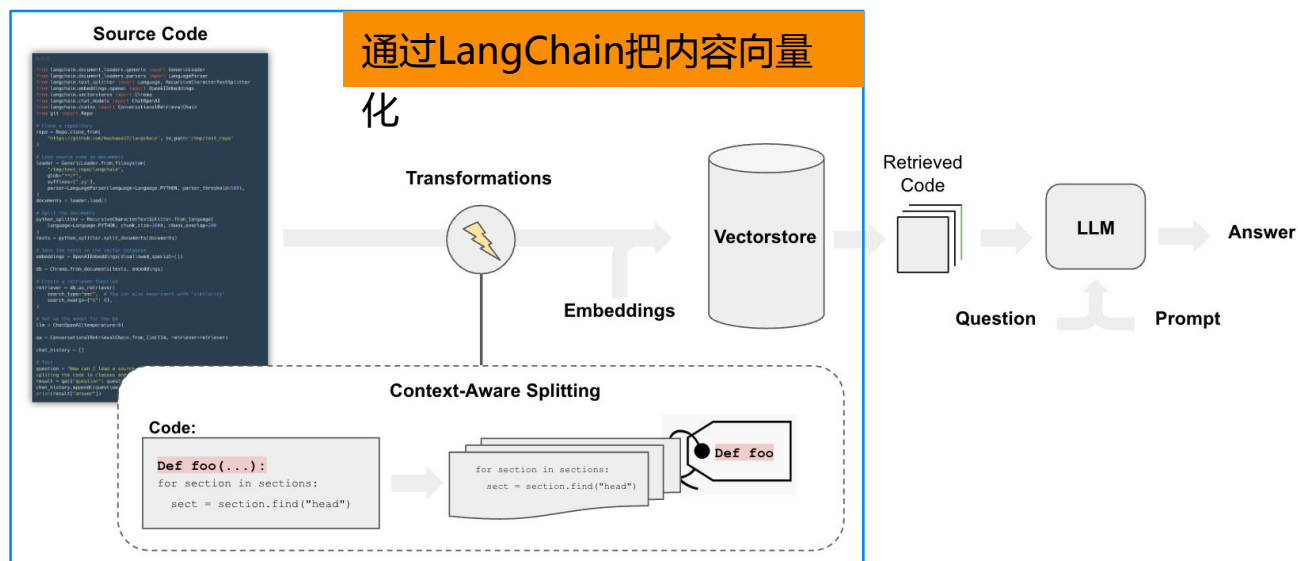
得到的结果有问题，调整你的



# AI辅助编程使用分享 -- 进阶的应用探索 ( Chat To Codebase )

- 受到token的长度限制 ( 目前大多数LLM能理解的内容长度是有限的 )
- 数据安全&合规

方案：通过 LangChain 来实现大文本的阅读 ( 业界的解决方案，常见的应用例如：ChatPDF )



- 1、通过 LangChain 把大内容向量化 ( 存入向量数据库 )
- 2、当用户提问时
  - a) 去向量数据库查找相关联的内容；
  - b) 同时把用户的问题 ( Prompt拼装 ) ，一起传给LLM ( ChatGPT、Llama2 )

# AI辅助编程使用分享 -- 进阶的应用探索 ( Chat To Codebase )

## LangChain + ChatGPT3.5 API

### 1、把仓库源码向量化 ( Transformation + Embedding )

```
Updated summary for src/utils/clipboard.js
Processing file: src/utils/format.js
Updated summary for src/utils/format.js
Processing file: src/utils/http-client.js
Updated summary for src/utils/http-client.js
Processing file: src/utils/request.js
Updated summary for src/utils/request.js
Processing file: src/views/advisory/apis/chat.js
Updated summary for src/views/advisory/apis/chat.js
Processing file: src/views/assistant/api/article.js
Updated summary for src/views/assistant/api/article.js
Processing file: src/views/assistant/data/assistant.js
Updated summary for src/views/assistant/data/assistant.js
Processing file: src/views/nlp/config/api-map.js
Updated summary for src/views/nlp/config/api-map.js
Processing file: src/views/nlp/data/index.js
```

### 2、执行任务："add 502 vue route, and update 502.vue data property 'message' to 'new 502

```
message"
fixing the gap in summarizing? ... no
Please enter your TASK (multiline supported): ... add 502 vue route, and update 502.vue 'message' to 'new 502 message again'
Task: add 502 vue route, and update 502.vue 'message' to 'new 502 message again'
Tokens in Summaries: 1452
Split summaries into 1 chunks of up to 2000; tokens each. (an agent would run for each)
Agent getRelevantFiles is running.
[
  {
    path: 'src/router/index.js',
    reason: 'The router configuration file is responsible for defining the routes in the application.',
    task: 'Add a new route for the '502.vue' component.',
    exists: true
  },
  {
    path: 'src/views/errors/502.vue',
    reason: 'The '502.vue' file is the Vue component for the 502 Bad Gateway error page.',
    task: 'Update the 'message' property in the '502.vue' component.',
    exists: true
  }
]
```

找出相关联内容

### 3、模型做出回答 ( 工具更新对应的文件 )

```
diff --git a/src/router/index.js b/src/router/index.js
index 64f9f00..77ccf91 100644
--- a/src/router/index.js
+++ b/src/router/index.js
@@ -93,6 +93,11 @@ const routes = [
   path: '/:pathMatch(.*)*',
   redirect: '/404',
 },
+ {
+   path: '/502',
+   name: '502',
+   component: () => import(/* webpackChunkName: "502" */ '../views/502.vue'),
+ },
];

// const router = createRouter({
diff --git a/src/views/502.vue b/src/views/502.vue
index 543a060..f155e25 100644
--- a/src/views/502.vue
+++ b/src/views/502.vue
@@ -1,15 +1,23 @@
<template>
  <div>
    <h1>502 Error</h1>
    <p>new 502 message</p>
    <div class="error-page">
      <h1>502 Bad Gateway</h1>
      <p>{{ message }}</p>
    </div>
  </div>
</template>

<script>
export default {
  name: '502',
+ data() {
+   return {
+     message: 'new 502 message',
+   };
+ },
+};
</script>

<style scoped>
</style>
\ No newline at end of file
+ .error-page {
+   text-align: center;
+ }
+</style>
(END)
```

# AI辅助编程使用分享 -- 风险

- 数据安全&合规
- AI的结果不符合你的预期

**QA & Thanks**