APifox使用帮助

官方地址：https://apifox.com

官方帮助：https://docs.apifox.com

此帮助文档主要介绍如何简单介绍使用APifox调试接口，以及其常用功能。

1 安装APifox

安装过程非常简单，直接下一步、下一步就行，这里就不在截图了。

2 注册与登录

首次启动客户端，可以看到下图所示的效果

使用APifox需要注册一个账号，选择一种你喜欢的方式注册与登录即可。

注册完成后，进入 Apifox 主界面。如果是首次登录，你将需要选择你的角色和工作模式。请根据你的工作职责选择最合适的选项。

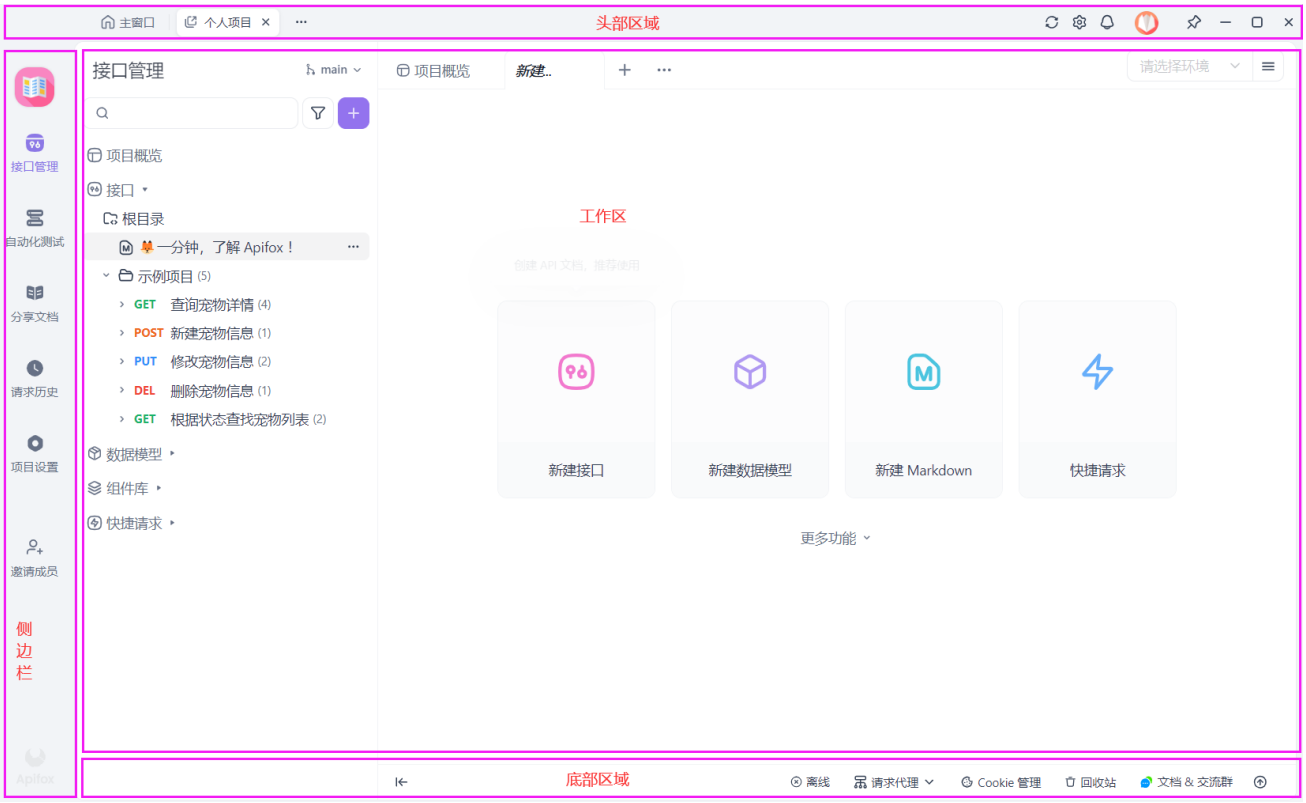
如果你处于学习阶段，暂时没有明确的角色，可以选择 “全栈工程师” 和 “API 设计优先” 模式。

点击 “开始” 按钮，即可开始使用 `Apifox`。

3 页面布局

登录进来后，你会默认拥有一个个人团队，并且系统会默认在个人团队创建一个个人项目，如下图所示

点击打开个人项目，简单介绍一下项目页面的布局



头部区域

头部区域包含以下功能模块：

- 主窗口
  - 进入个人主页，展示你加入的团队及其成员、项目、权限、订单、组织和 `API Hub` 等信息。
- 项目标签
  - 每个标签代表一个项目，点击即可切换；点击项目名称左侧按钮可在新窗口打开项目。
- 刷新
  - 重新加载当前项目数据，同时关闭所有打开的标签。
- 设置
  - 调整本地客户端的设置（如外观、网络代理等），这些设置仅影响当前客户端。
- 通知
  - 显示团队活动的提醒信息。
- 头像
  - 账号设置，可以设置基本账户信息、访问令牌 (Access tokens) 等。

侧边栏

侧边栏包含以下功能模块：

- 项目图标
  - 点击跳转到团队与项目。
- 接口管理
  - `Apifox` 的核心界面，你可以在这里 新建接口、发送接口请求、新建数据模型、编写 Markdown 等。

- 自动化测试
    - 如果你需要批量发送请求 *（类似运行 Postman 的 Collection）* 或编排接口之间的数据关系和逻辑，可以在自动化测试中新建测试场景。基于测试场景，你还可以查看测试报告、运行性能测试、管理测试数据、集成 CI/CD 等。
- 分享文档
    - 一旦创建接口，可以在该模块中分享给其他同事，或者发布文档站。发布文档站可以自定义文档域名、调整页面结构以及自定义 URL 等。
- 请求历史
    - 在请求历史中可以查看并重新发送所有已发送的请求。
- 项目设置
    - 所有与当前项目相关的设置，包括基本设置、功能设置、通知设置、项目资源和项目数据的导入/导出。
- 邀请成员
    - 邀请其他用户加入当前项目。

工作区

工作区主要包含目录树和标签页

目录树

目录树从上到下包括以下功能模块：

- 分支与版本
  - 默认显示主分支，你可以切换到其它分支，或者新建分支、管理分支。
- 搜索与过滤
  - 你可以搜索和过滤接口。注意，目前只能搜索接口名称，无法搜索目录名称或用例名称。
- 新建
  - 你可以创建各种类型的元素，如接口、快捷请求、数据模型、Markdown 等。
- 项目概览
  - 项目的整体视图，包括项目统计、接口用例覆盖情况等。
- 接口
  - 以目录结构组织的接口和接口用例，Markdown 和 `WebSocket` 等也可以包含在其中。
- 数据模型
  - 以目录结构组织的数据模型。
- 组件库
  - 可复用的数据模型，可设置默认响应模板以及响应组件。

- 快捷请求
    - 以目录结构组织的接口请求，类似于 Postman 的 Collection。

标签页

点击目录树中的任何元素会打开一个标签页

标签页包括以下类型：

- 接口
- 数据模型
- 响应组件
- Markdown
- 目录
- 快捷请求
- `WebSocket`
- 项目概览
- 新建...

单击打开标签页时，标题显示为斜体，此时点击目录树其它元素会覆盖当前标签，适合浏览场景。

修改标签内容后，标题变为常规字体，此时点击目录树其它元素会打开新标签，适合编辑场景。

若不希望标签被覆盖，可双击打开标签，使其直接显示为常规字体。

标签页的右侧有两个按钮：

- **+**`新建`：可以创建各种类型的元素。
- **...**`更多`：你可以关闭所有标签、关闭当前标签或关闭除当前标签之外的所有标签。

底部区域

底部区域包含以下功能模块：

- 折叠/展开
    - 可以折叠或展开左侧目录树。
- 在线状态
    - 显示当前是否在线。如果离线，团队同步将出现问题。
- Agent
    - `Apifox Web` 独有的功能，允许你选择代理发送请求。
- Cookie 管理
    - Cookie 管理器，可以存储当前的 Cookies。
- 回收站
    - 删除的接口、用例、数据模型、响应组件、测试场景、定时任务等会进入回收站，30 天后自动删除。
- 文档 & 交流群
    - 可打开帮助文档或加入交流群。

4 基本概念

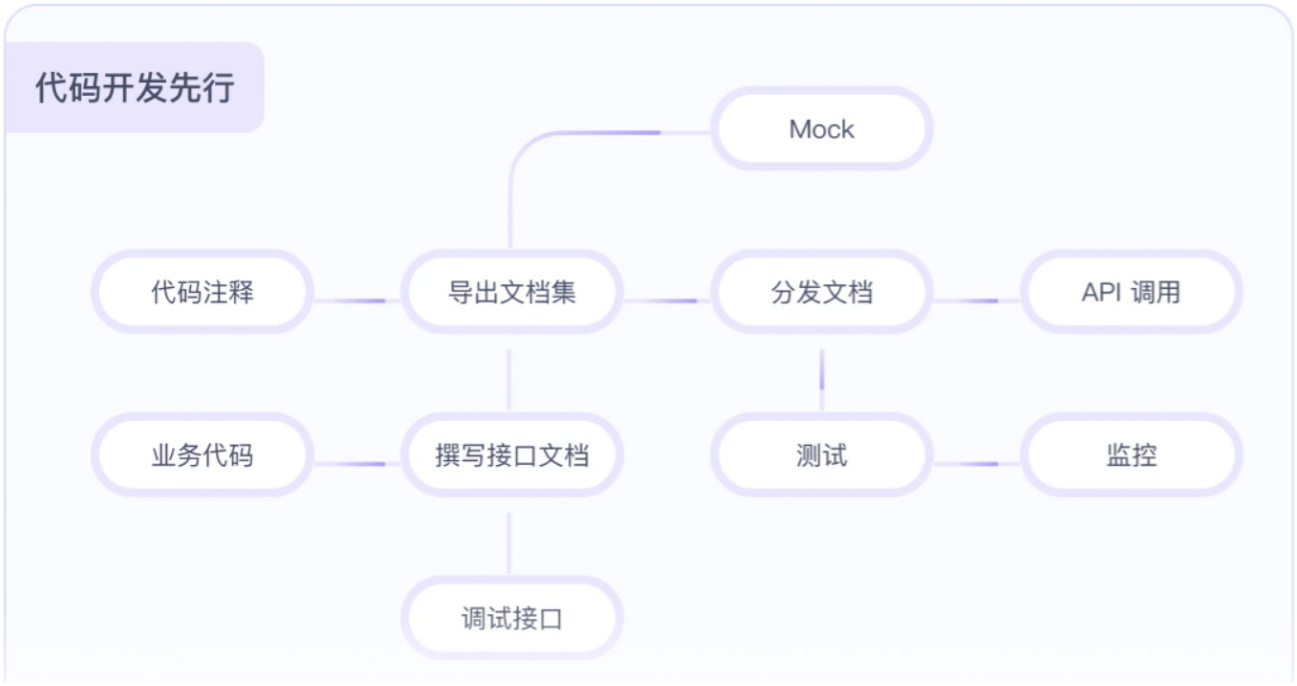
文档模式/调试模式

Apifox 的 API 调试有两种模式，可以接口标签页左下角切换，分别是：文档模式 和 调试模式。

这两种模式提供了类似的功能，但界面不同，以适应不同团队的工作流程。

- **文档模式**是 Apifox 推荐的模式，适合采用 API 设计优先的团队。在这个模式中，团队首先定义 API，随后根据 API 文档进行开发和测试。

- **调试模式**则特别适合那些没有事先定义 `API` 文档的团队。这样的团队通常集中在后端开发，先完成代码，然后再生成 `API` 文档进行测试工作。



如果需要调用其他人开发的 `API` 而没有文档，建议也使用调试模式。

接口

`Apifox` 是一个以 `API` *（接口）* 为核心的产品，这意味着所有操作都从定义 `API` 开始。

- 在 `Apifox` 的主界面中，接口是基本元素，它们以目录的形式进行分组。对于每个接口，你可以进行修改、预览、发送请求，或者将请求保存为接口用例。
- 这种结构与 Postman 大不相同，`Apifox` 更像是一种基于 OAS *（即 `OpenAPI Specification`，一种用于描述和定义 `RESTful API` 的标准化格式）* 的扩展——你可以直接调试和保存请求。
- 在 Postman 中，基本元素是 “请求”，而这些请求与 `API` 定义 *（即 `API specification`，是定义和描述 `API` 行为的关键文档或标准）* 本身是分离的。这意味着当 `API` 定义发生变化时，所有的请求和脚本都需要重新编写。
- 在 `Apifox` 中，所有的接口用例 *（对应 Postman 的请求）* 都是基于 `API` 定义 *（`API specification`）* 的。当 `API` 定义发生变化时，接口用例会同步变化，所有基于此的测试场景和 CI/CD 也可以自动或手动更新，非常适合开发团队在维护和更新 `API` 时使用。

快捷请求

在 `Apifox` 中，你也可以创建快捷请求。这些快捷请求不需要基于 `API` 定义进行设计，其方式与 Postman 请求相同。你还可以将成功的请求保存为接口文档。

测试场景

当你需要批量发送请求时 *（类似运行 Postman Collection）*，你可以使用自动化测试来编排测试场景。

一个测试场景包括一系列测试步骤。这些测试步骤可以从接口或接口用例中导入到测试场景，并可以在接口文档发生变化时自动或手动更新相应的参数。

测试场景还支持流程控制条件如 If、for、`forEach` 等。你可以在测试步骤间传递数据、动态生成请求参数。

基于测试场景，你还可以查看测试报告、运行性能测试、管理测试数据、集成 CI/CD 等。

环境

`Apifox` 中的环境与 Postman 中的环境类似，包含许多变量。切换环境时，可以使用相同环境变量的不同值。

然而，`Apifox` 的环境还包括另一个重要概念：服务。每个服务对应一个前置 URL。由于 `OpenAPI/Swagger` 中定义的接口都以 `/` 开头，它们可以被发送到不同的服务。在 `Apifox` 中，你无需在请求开头写 `{{Base_url}}` 来设置服务为变量。只需切换到对应的环境，所有请求都会自动发送到该环境中的相应服务。

项目

接口、快捷请求、数据模型、环境、测试场景等的集合构成了一个项目。在 Apifox 中，项目是协作的基本单位。

一个项目对应一个 API 定义 *(API specification)* 。你还可以将 OpenAPI/Swagger 导入项目，或将项目导出为 OpenAPI/Swagger。

与 Postman 相比，Apifox 项目大致相当于 Postman 的 Collection，而 Apifox 团队则类似于 Postman 的 Workspace。

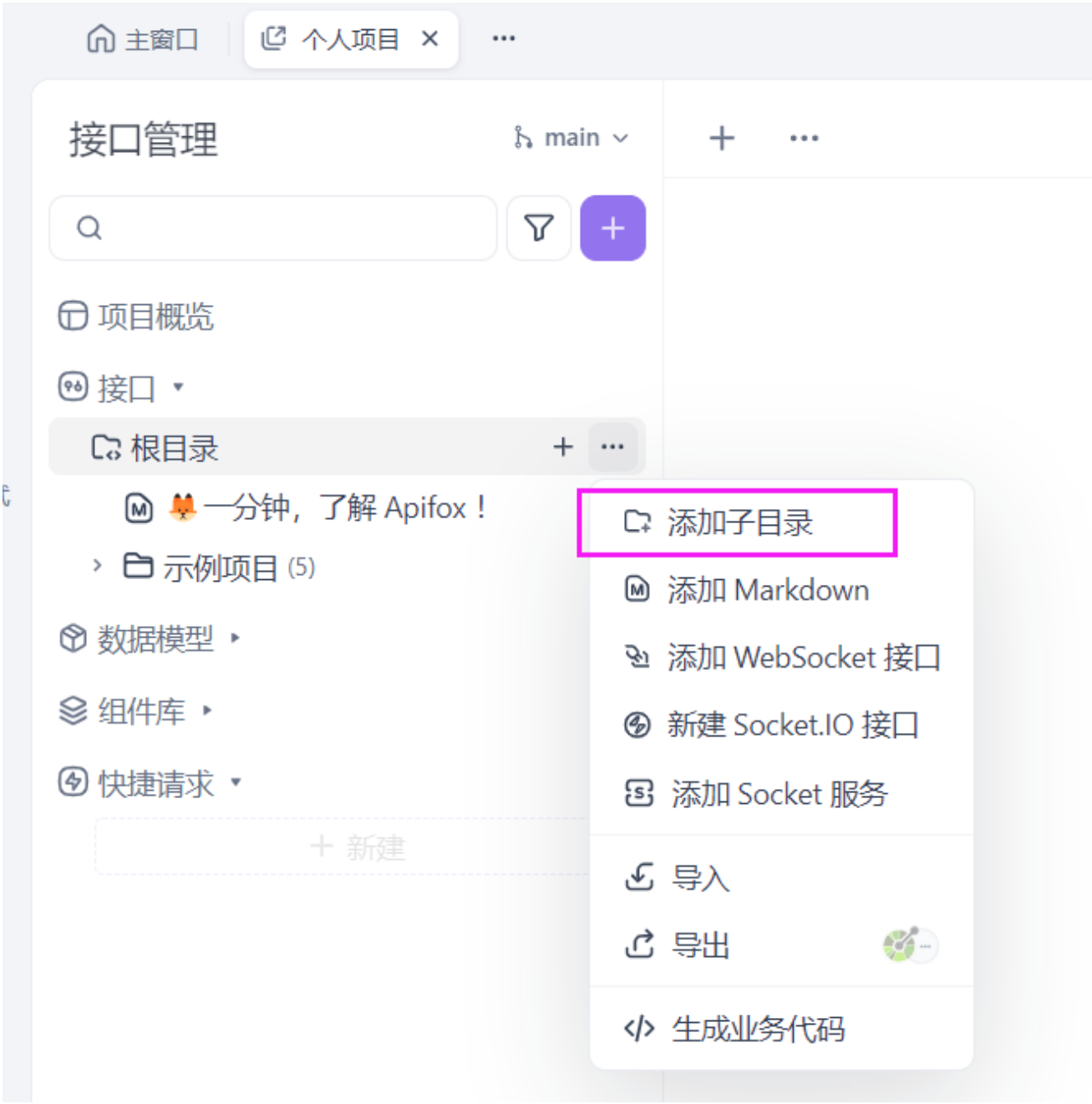
5 接口操作

Apifox 为 API 设计者提供了直观的界面来新建接口。下面来演示一个简单的分页查询示例

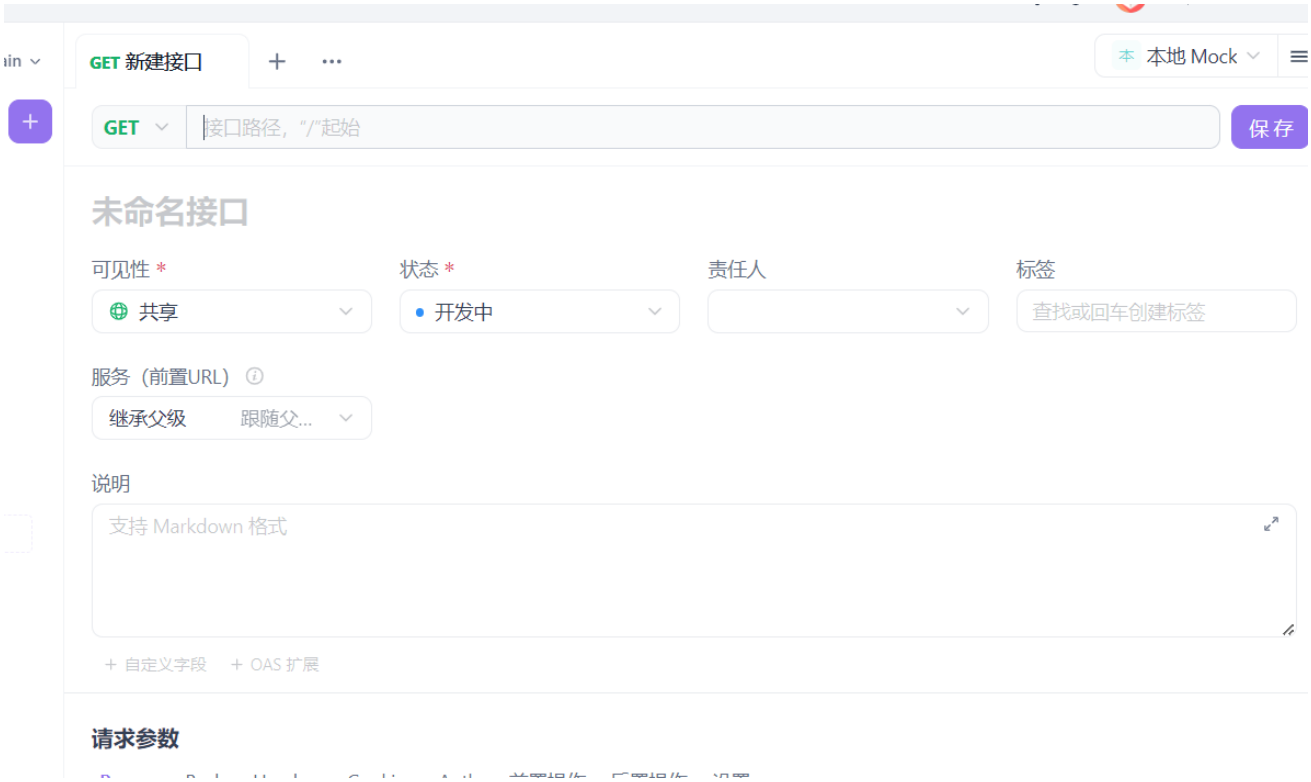
新建接口

新建接口有很多入口，选择一个你喜欢的方式都可以，如下图圈出来的地方都可以新建接口

顺便提一下，为了方便分模块或归类管理接口，你可以创建目录来管理你的接口，避免很混乱，通过下面的菜单可以创建目录

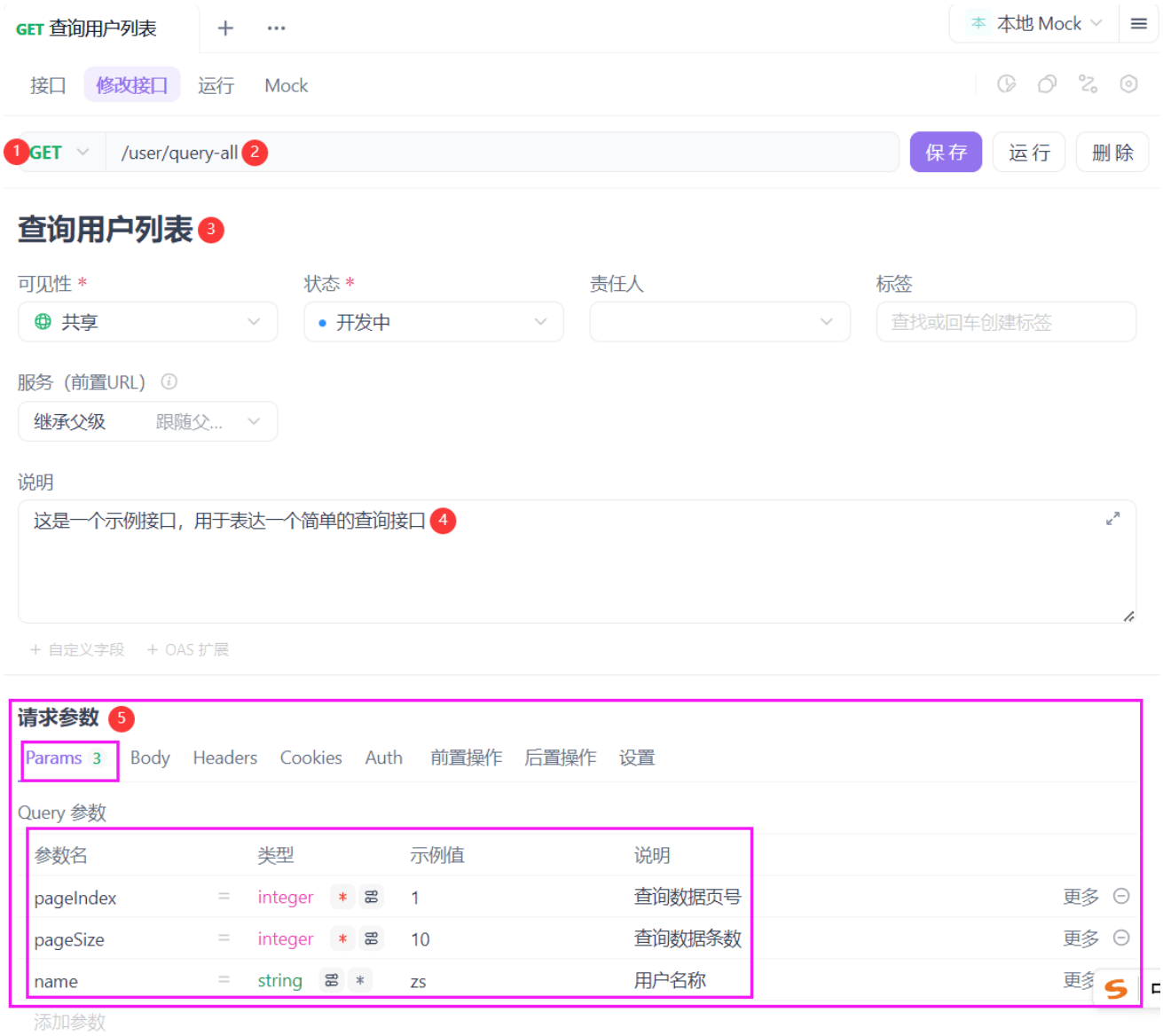


点击完新建接口后，会新开一个标签页来用于编辑新的接口的相关信息，当然在没有点击保存之前不会将接口保存到目录树的。

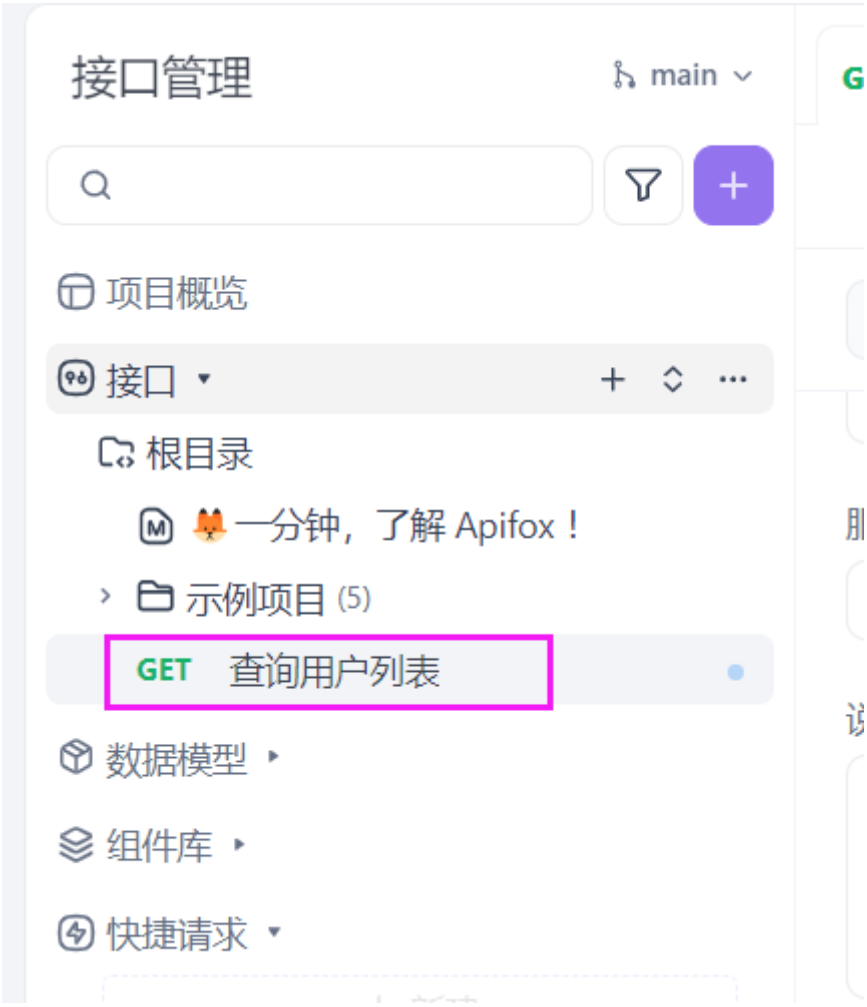


接下来我们来定义一下接口的定义请求信息，主要包括：

1. 请求方式：常用的请求方式（GET、POST、PUT、DELETE）
2. 请求地址：在接口路径栏中输入请求地址，请求地址/开头，如`/user/query-all`
3. 接口名称：给接口命名，这样我们才知道此接口是做什么业务的
4. 接口说明：可以给接口添加详细的描述信息
5. 请求参数：因为查询一般是GET请求，所以我们设置的是查询参数



输入完成后记得保存，保存成功后可以在，目录树种看到你的接口了

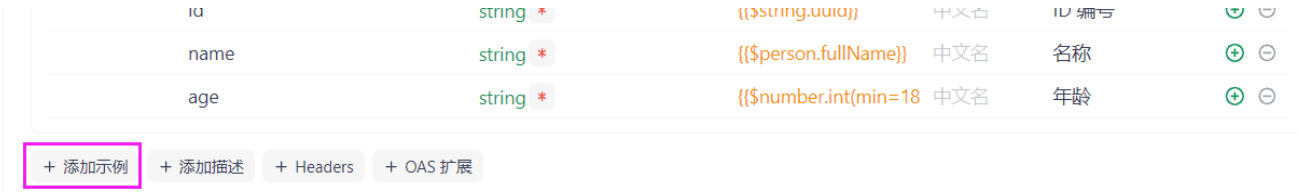


然后再来定义响应结构，滚动至 返回响应 部分，在 “成功(200)” 响应下，点击 “+”添加字段。

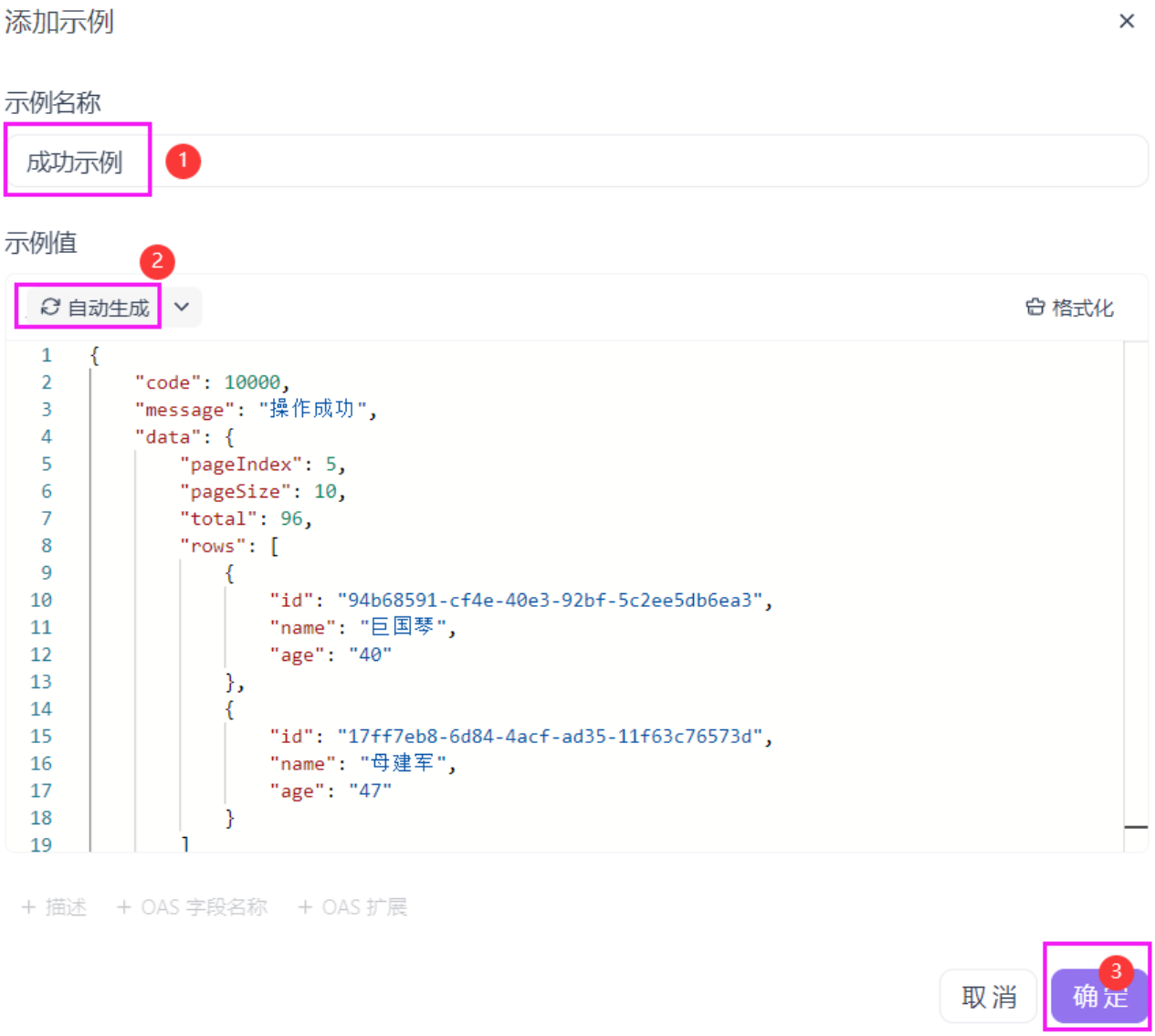
比如下面是我添加的响应结构示例

修改完记得随手保存一下。

有了响应结构可以使用它帮你生成响应示例，点击添加响应示例



然后再弹出界面生成响应示例



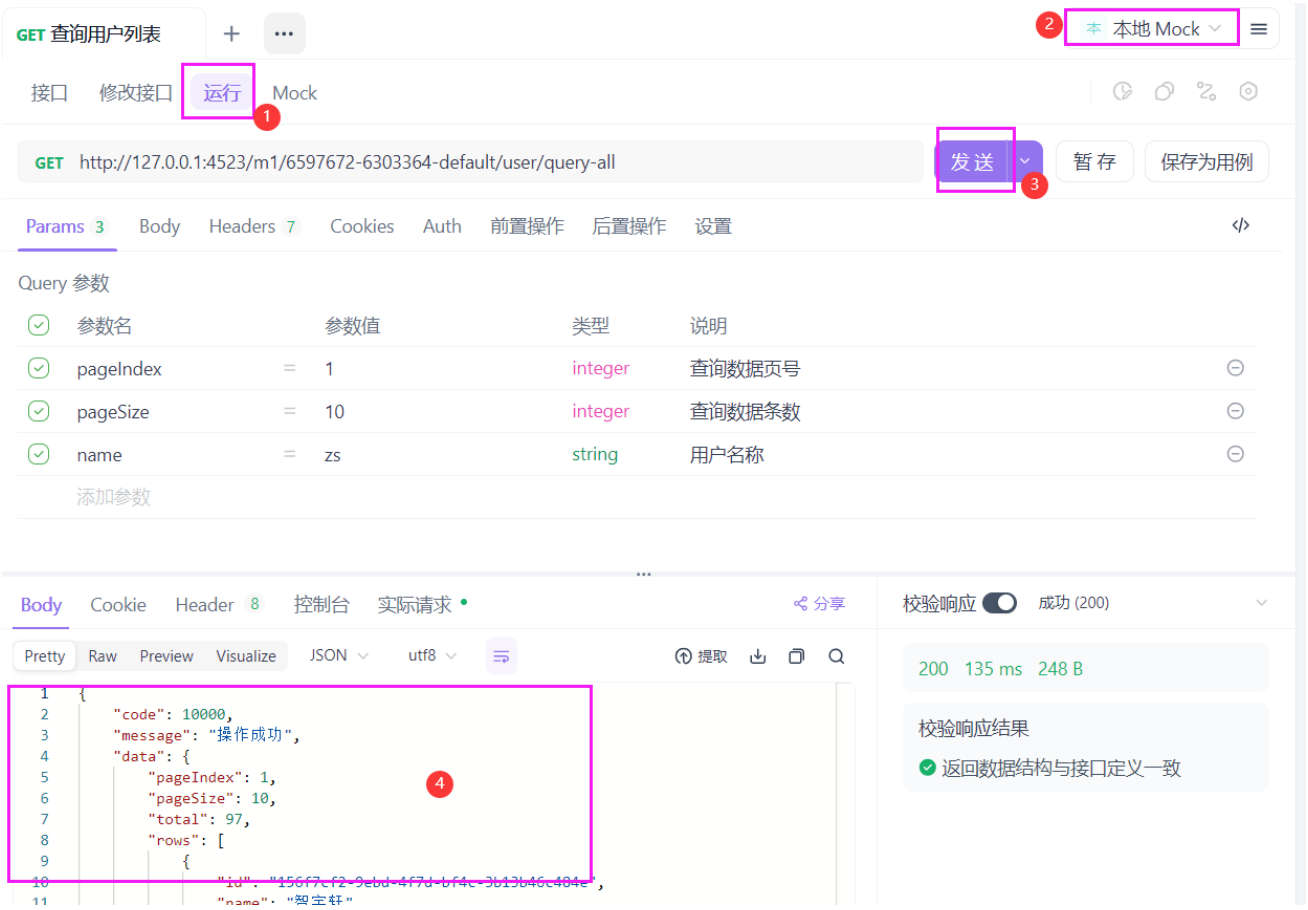
点击确定后可以看到响应示例，如下图所示

有了响应示例可以方便，前后端对接的时候看具体响应数据格式，保存后切换到接口预览视图，可以看到接口的完整信息

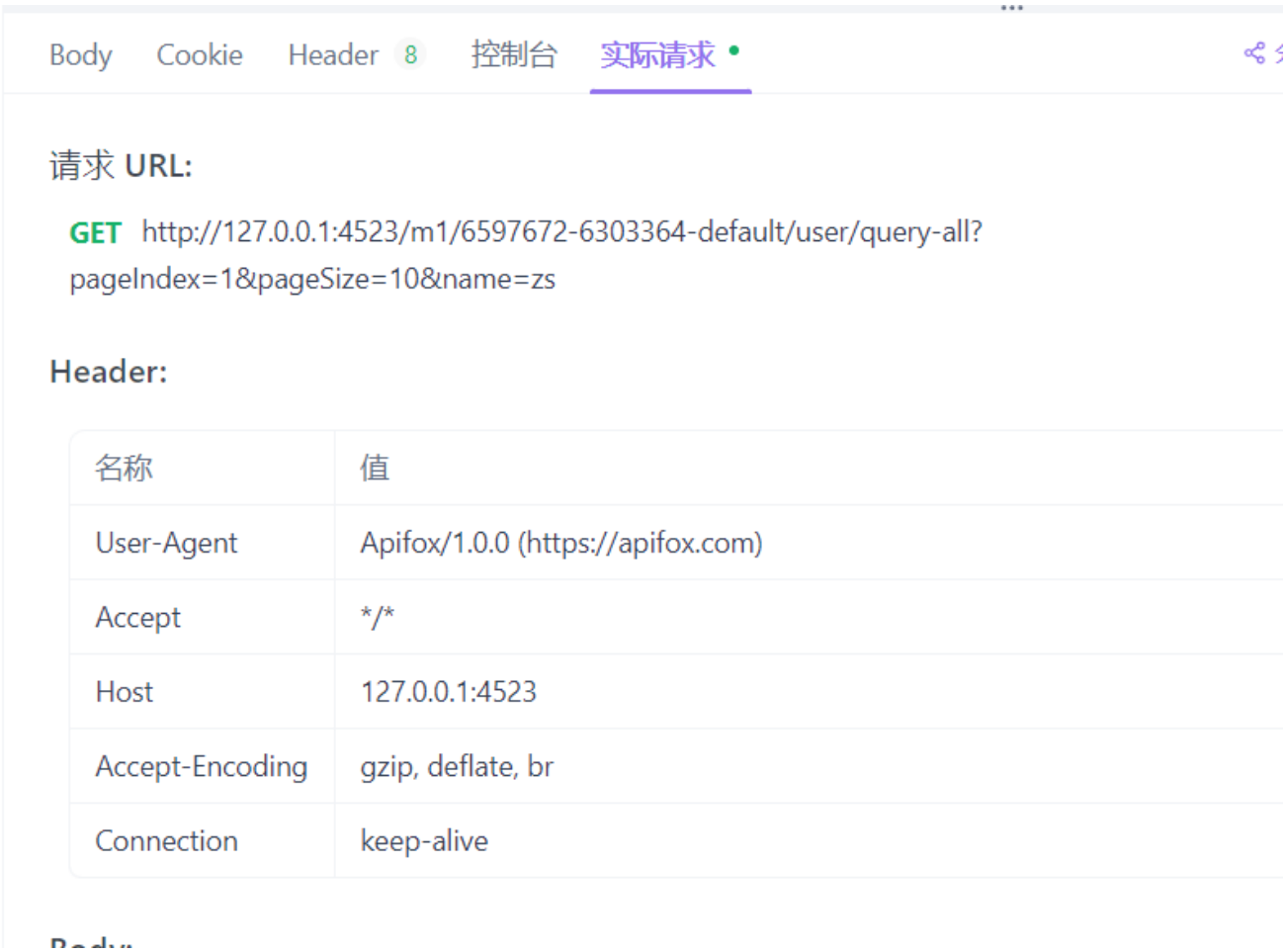
发送接口请求

当接口创建完成后，即可发起请求。

1. 在 **“文档模式”** 下，切换到 **“运行”** 标签页。
2. 在右上角选择环境为 **“本地 Mock”**。
3. 点击 **“发送”** 按钮以发送请求。
4. 你可以在**返回响应**区域查看请求的结果。



切换到 **“实际请求”** 以查看实际发送的请求。



更多内容点击访问官网，学习了解。