博远信息技术社
Boyuan IT Club

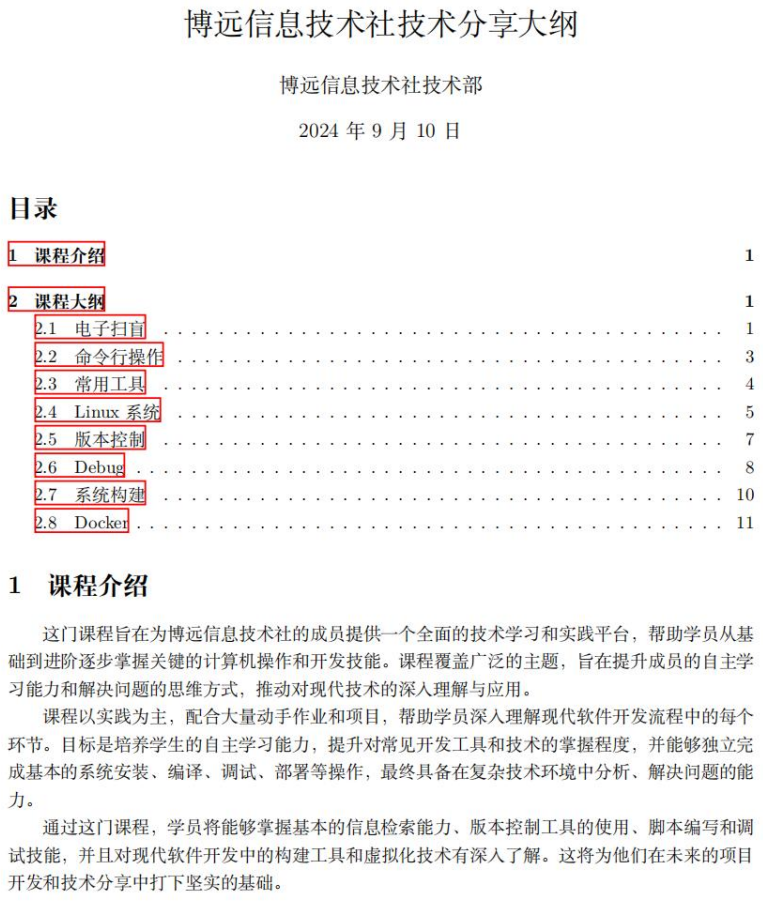
技术分享·课程大纲

Technique Sharing·Syllabus

2025年11月19日

活动介绍

技术分享



博远信息技术社技术分享大纲

博远信息技术社技术部

2024 年 9 月 10 日

目录

1 课程介绍

这门课程旨在为博远信息技术社的成员提供一个全面的技术学习和实践平台，帮助学员从基础到进阶逐步掌握关键的计算机操作和开发技能。课程覆盖广泛的主题，旨在提升成员的自主学习能力和解决问题的思维方式，推动对现代技术的深入理解与应用。

课程以实践为主，配合大量动手作业和项目，帮助学员深入理解现代软件开发流程中的每个环节。目标是培养学生的自主学习能力，提升对常见开发工具和技术的掌握程度，并能够独立完成基本的系统安装、编译、调试、部署等操作，最终具备在复杂技术环境中分析、解决问题的能力。

通过这门课程，学员将能够掌握基本的信息检索能力、版本控制工具的使用、脚本编写和调试技能，并且对现代软件开发中的构建工具和虚拟化技术有深入了解。这将为他们在未来的项目开发和技术分享中打下坚实的基础。

技术分享课程旨在为同学们提供一个全面的技术学习和实践平台，帮助学员从基础到进阶逐步掌握关键的计算机操作和开发技能。课程覆盖广泛的主题，旨在提升成员的自主学习能力和解决问题的思维方式，推动对现代技术的深入理解与应用。

课程以实践为主，配合大量动手作业和项目，帮助学员深入理解现代软件开发流程中的每个环节。目标是培养学生的自主学习能力，提升对常见开发工具和技术的掌握程度，并能够独立完成基本的系统安装、编译、调试、部署等操作，最终具备在复杂技术环境中分析、解决问题的能力。

通过这门课程，学员将能够掌握基本的信息检索能力、版本控制工具的使用、脚本编写和调试技能，并且对现代软件开发中的构建工具和虚拟化技术有深入了解。这将为他们在未来的项目开发和技术分享中打下坚实的基础。

活动介绍

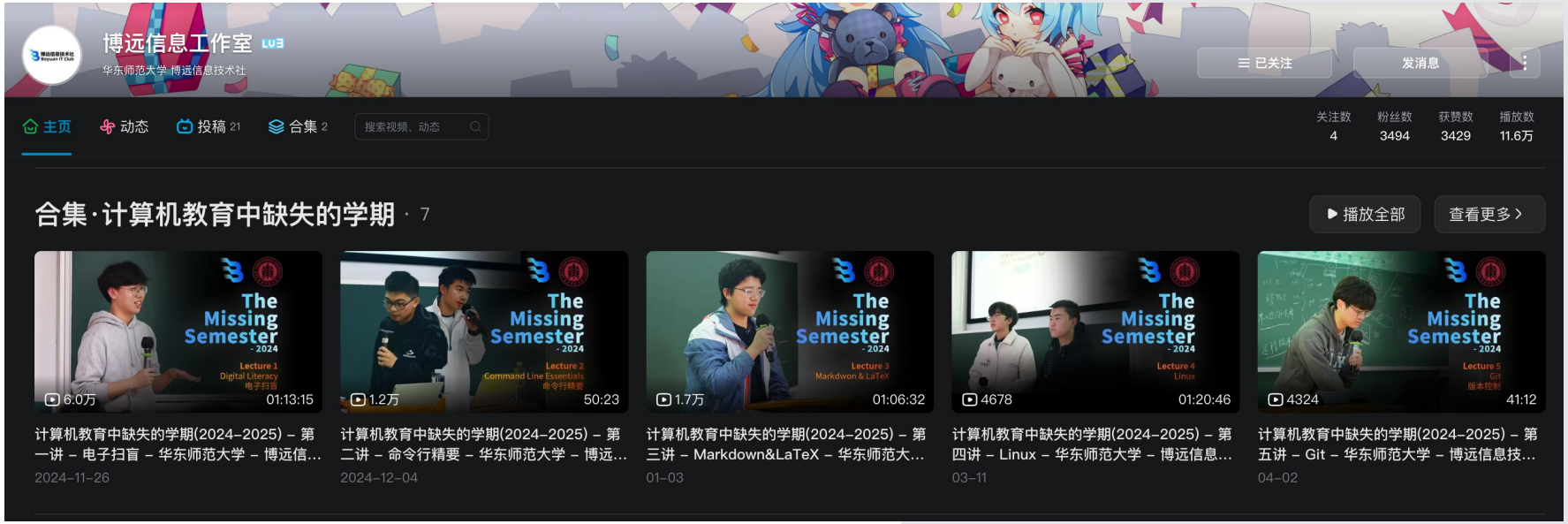
技术分享

在2024年的技术分享中，社团尝试引入更系统的学习思路，参考MIT的"Missing Semester"课程，聚焦于“计算机教育中缺失的学期”这一理念。分享内容涵盖了电子扫描、Git、Docker等基础但实际应用广泛的技术工具与流程，既贴近成员的学习需求，也兼顾了技能的系统性和实用性。

这一年度的分享获得了较好的反响，成员普遍认为内容实用性强、易于上手，同时也拓展了对计算机基础工具的认知。

技术分享

bilibili账号：**博远信息工作室**
开设的《计算机教育中缺失的学期》累计获得10W+播放量



活动介绍

标题：AI 对话应用的基础构建

1. 理解与调用大模型接口（HTTP 基础）
2. 构建面向对象的对话程序（上下文与输入管理）
3. 提示词工程与人机交互设计
4. 语音识别与语音合成（ASR 与 TTS）

标题：AI 能力扩展与智能增强

5. 多模态输入处理：附件、OCR 与动态变量
6. 知识增强对话系统（RAG 技术）
7. 智能体与工具调用（Function Calling / MCP）
8. 模型微调与个性化定制

技术分享

进入2025年，新一届的技术分享计划在总结往年经验的基础上，明确了技术前沿和社团成员兴趣的结合点。今年的分享重点将聚焦于人工智能相关领域，包括AI调用、RAG、MCP、微调等前沿技术。计划通过系统化、循序渐进的课程设计，使同学们不仅能够了解AI技术的理论基础和实际应用场景，还能够动手实践、提升解决实际问题的能力。通过这样的安排，社团希望打造更有深度、更具连续性的技术分享体系，让同学们在兴趣驱动的基础上，获得真正可落地的技能和前沿技术视野。

理解与调用大模型接口（HTTP 基础）

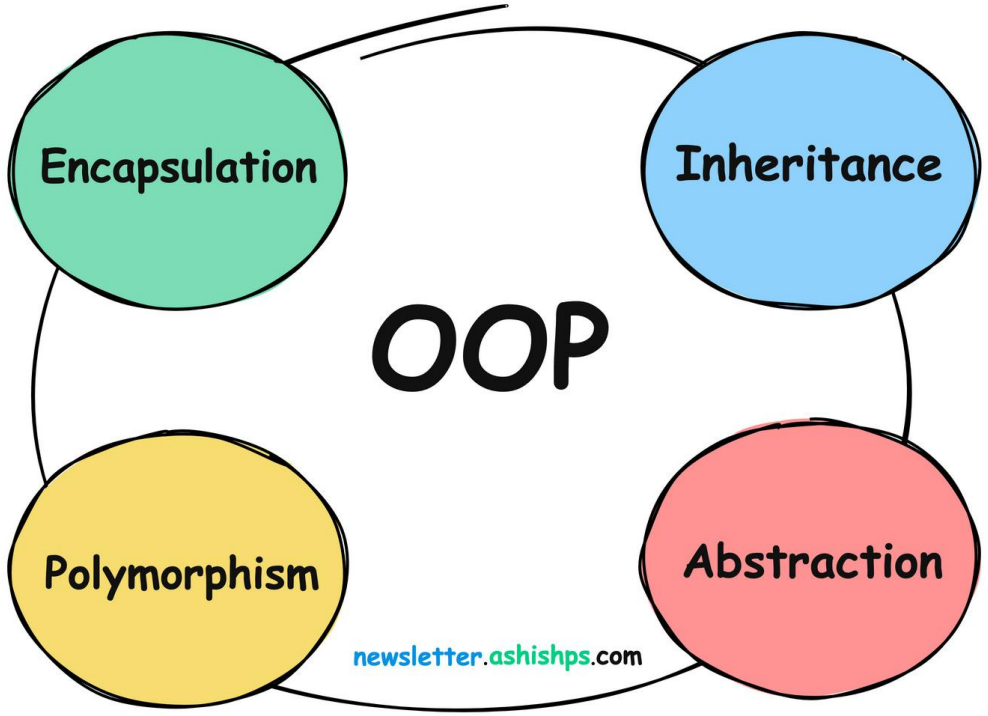
学习目标

本板块是Web开发和AI应用的通信基础——解决"前端和后端如何交换数据"的核心问题。通过学习HTTP请求与响应机制，掌握"如何发送数据给服务器"和"如何接收服务器返回的内容"，特别是现代AI应用中常用的流式传输技术（AI中的“打字机”效果），为开发实时交互的Web应用打下坚实基础。

构建面向对象的对话程序（上下文与输入管理）

学习目标

通过本课程的学习，应能够掌握 Python 面向对象编程的基本思想与语法，包括继承、封装和多态的应用；理解面向对象相对于面向过程编程的优势，并能够在实际问题中进行应用。同时，学生能够理解对话系统上下文管理的重要性，掌握利用 Python 数据结构、文本文件或 SQLite 数据库实现上下文存储与管理的方法，能够设计并实现一个具备多轮对话能力的简单对话程序。



提示词工程与人机交互设计

学习目标

学生通过本课程学习，应能够设计并实现具备“开场白”和“问题建议”功能的 AI 对话系统，包括：明确 AI 身份与能力边界、规范输入输出格式、构建结构化建议输出（Suggestions Schema）；掌握提示词工程的分层设计（指令层、上下文层、控制层）及模板化管理；

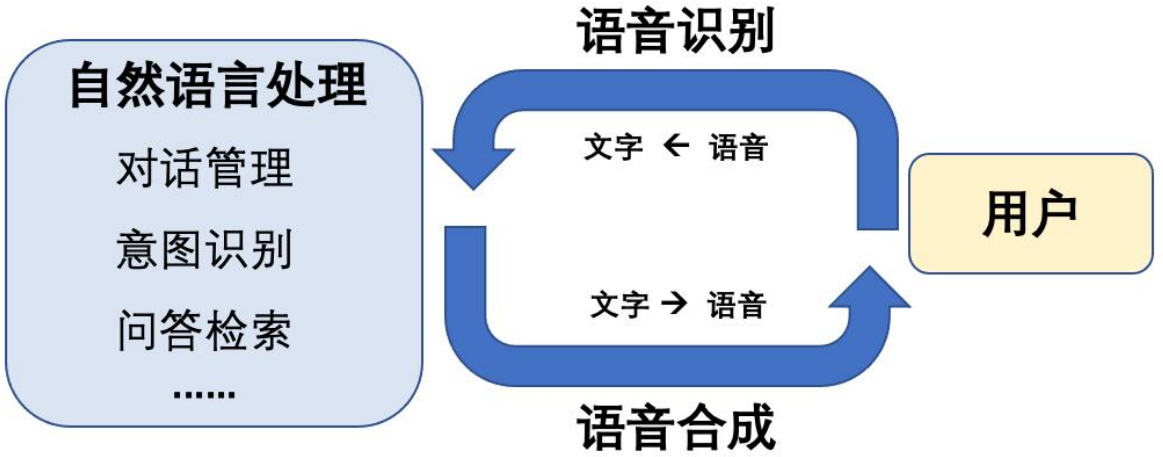
```
diff

身份：我是你的 <角色>，覆盖 <领域>，可使用 <工具/RAG/多模态>。
能力：我能 <能力1/能力2>；当证据不足时，我会先澄清问题并标注不确定性。
边界：对 <越权/敏感> 内容将拒答或转人工，遵循 <合规规则>。
输入指南：请提供 <关键字段/资料格式>，若缺失我会提出最多 3 条澄清。
输出契约：除主要回答外，我会附带 suggestions[]（最多 3 条），见下方 Schema。
操作要求：
- 每次回答保持身份一致。
- 每轮回答末尾输出 suggestions[]，按优先级排序；若无需建议返回空数组并给出 reason。
- 所有输出严格符合指定 JSON Schema。
```

语音识别与语音合成（ASR 与 TTS）

学习目标

学生通过本课程学习，应能够理解 ASR（自动语音识别）与 TTS（语音合成）的核心概念、技术逻辑及应用场景，掌握 WebSocket 与 HTTP API 等接口的调用流程及适用场景；能够结合实际需求选择合适的技术方案，实现实时语音识别与语音播报功能；理解 ASR 与 TTS 在降低交互门槛、提升实时性、满足个性化需求等方面对用户体验的优化作用；能够分析接口和技术局限性对体验的影响，并提出改进思路。



活动介绍

技术分享

课程时间：每周一次，每周三下午三点

- 本周暂时调整为下午三点半

具体形式：

1. 博远信息技术社技术部成员授课
2. 实践动手机会
3. 课程作业
4. 专门群聊答疑

活动介绍

技术分享

本学期的四次课：

AI 对话应用的基础构建

1. 理解与调用大模型接口（HTTP 基础）	11月19日
2. 构建面向对象的对话程序（上下文与输入管理）	11月26日
3. 提示词工程与人机交互设计	12月3日
4. 语音识别与语音合成（ASR 与 TTS）	12月10日

博远信息技术社
Boyuan IT Club

活动介绍

加入群聊

博远信息技术社 AI应...
群号: 1034572351

博远工作室 外部·答疑...
群号: 765667302

扫一扫二维码，加入群聊

QQ

博远工作室 外部·答疑
群号: 908284692

扫一扫二维码，加入群聊

QQ

群聊：博远妙妙屋2025

该二维码7天内(11月26日前)有效，重新进入将更新

博远信息技术社
Boyuan IT Club

技术分享

理解与调用大模型接口（HTTP 基础）

讲师：陈睿　助教：胡淙煜

2025年11月19日

第1节：理解与调用大模型接口（HTTP 基础）

博远信息技术社
Boyuan IT Club

CONTENTS

博远信息技术社
Boyuan IT Club

PART - 1

Introduction

介绍HTTP

01 Introduction

学习目标

本板块是Web开发和AI应用的通信基础——解决“我们与 AI 大模型如何交互”的核心问题。通过学习HTTP请求与响应机制，掌握"如何发送数据给服务器"和"如何接收服务器返回的内容"，特别是现代AI应用中常用的流式传输技术（AI中的“打字机”效果），为开发实时交互的Web应用打下坚实基础。

你将学会

1. 理解：**HTTP请求的完整结构**（Method、Path、Header、Body）、响应的组成（Status、Header、Body）和流式处理和非流式处理的区别
2. 掌握：
   - 发送不同类型的HTTP请求（GET、POST）
   - 解析HTTP响应内容（JSON、文本）
   - 实现SSE流式传输（AI打字机效果）
3. 应用：独立完成"带流式响应的AI聊天接口"demo，实现类ChatGPT的实时回答效果

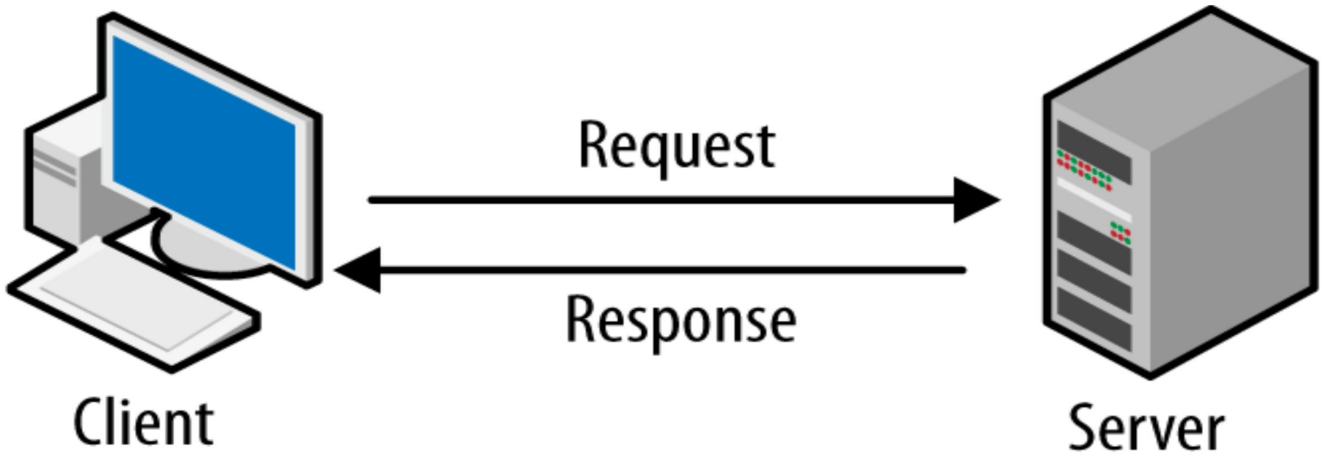
HTTP

· HTTP的全称是HyperText Transfer Protocol，翻译过来就是"超文本传输协议"。

· 协议的定义：对等实体在通信过程中遵守规则的集合

· 在 HTTP 这个协议下，对等的实体就是 Client 和 Server（C/S模式）
在 AI 应用中，我们常常认为应用端扮演了客户端的角色，而大模型扮演了服务器的角色

· 我们可以简单但不精确地理解：Web应用中，HTTP就是应用端和大模型通信遵守规则的集合

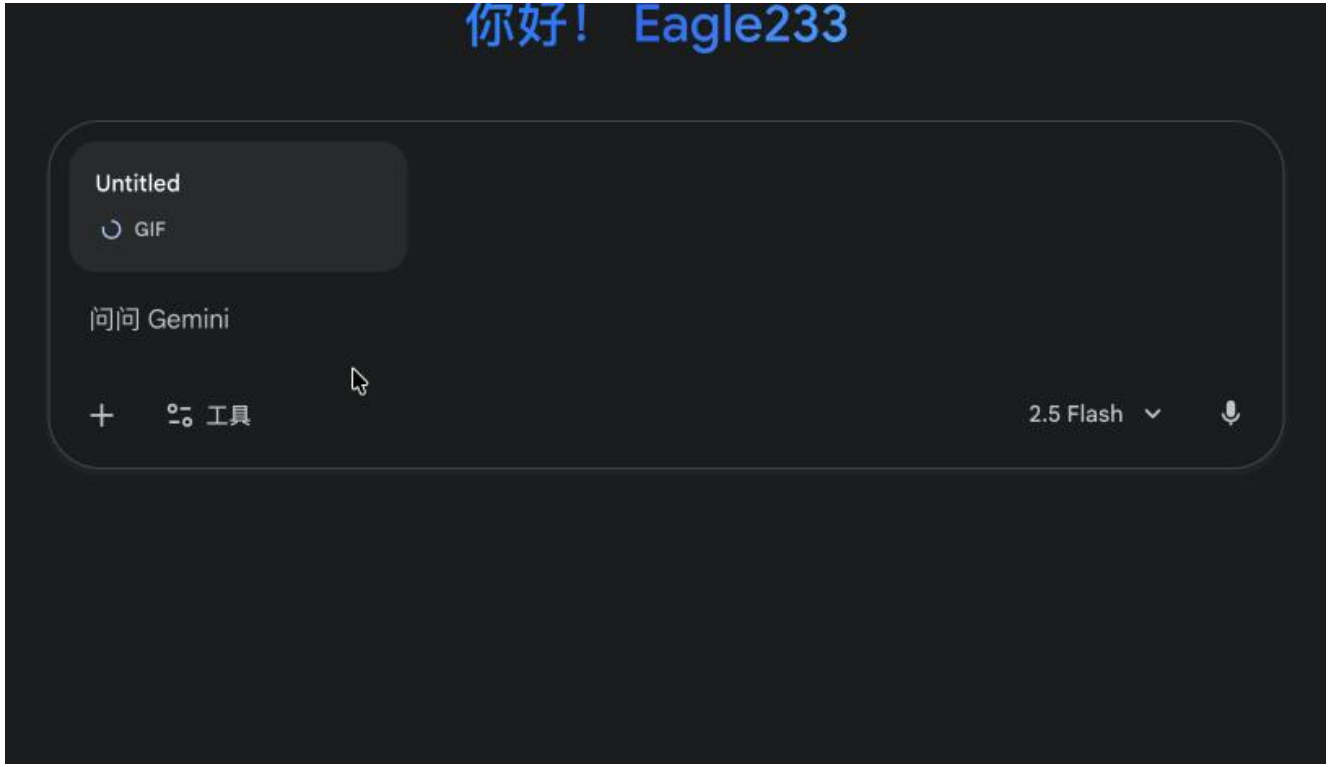


数据流转链

用户操作 → 发送HTTP请求 → 服务器处理 → 返回HTTP响应 → 展示结果

可能的应用


不安全 https://official.boyuan.club/main/publish
应用
陈睿
首页
简历投递
个人主页
简历查看
博远信息技术社招新申请表
欢迎加入博远信息技术社，请填写以下信息完成申请
填写提示
基本信息
* 姓名
请输入您的姓名
* 学号
请输入您的学号
个人照片
* 性别
男 女
* 年级
请选择年级
上传照片
* 专业
请输入您的专业
* 邮箱
请输入您的邮箱
建议上传正面免冠照片，大小不超过5MB
* 手机号
请输入您的手机号
GitHub主页
请输入您的GitHub主页...
自我介绍
* 自我介绍
请介绍一下您的个人特点、兴趣爱好、技能特长等...
* 加入理由


你好！ Eagle233
Untitled
GIF
问问 Gemini
工具
2.5 Flash

可能的应用


ChatGPT
免费升级
共享
你是谁
我是 ChatGPT，基于 GPT-5 mini 的 AI 聊天助手。我可以帮你回答问题、解释知识、写作、编程，或者陪你聊天、开脑洞。
你想让我侧重哪方面帮助你呢——学习、技术、生活，还是随便聊聊？
什么事

博远信息技术社
Boyuan IT Club

PART - 2

HTTP请求

02 HTTP请求

HTTP Request

HTTP请求 就像发快递

包裹：	报文 Message
包裹类型：数码产品、证件、邮件	Method（GET、POST、PUT、DELETE）
收件地址：上海市普陀区中山北路3663号	Path（/api/users/123）
备注信息：易碎品、需要本人签收	Header（Authorization、Content-Type）
包裹内容：你寄的实际物品	Body（JSON数据、文件等）

HTTP

前后端是用报文进行交互的，无论是HTTP请求还是HTTP响应

包含起始行 Request Line、头部 Header、空行 CRLF 和请求体 Body

```
Method SP Request-URI SP HTTP-Version CRLF
Header-Name: Header-Value CRLF
Header-Name: Header-Value CRLF
...
CRLF
Body
```

```
POST /api/users HTTP/1.1
Host: example.com
Content-Type: application/json
Authorization: Bearer abc123

{
  "username": "chenrui",
  "password": "123456"
}
```

URL

统一资源定位符 Uniform Resource Locator

俗称网页地址，简称网址，是 Internet 上标准的资源的地址（Address），如同在网络上的门牌。

02 HTTP请求

URL

bilibili.com/video/BV17E1pBBEbB/?spm_id_from=333.1007.tianma.1-2-2.click&vd_source=a24ab840c2e1c1bb1ba3435ab64fc41c

Protocol: https

http, ftp, smtp 都属于应用层协议

Domain: bilibili.com

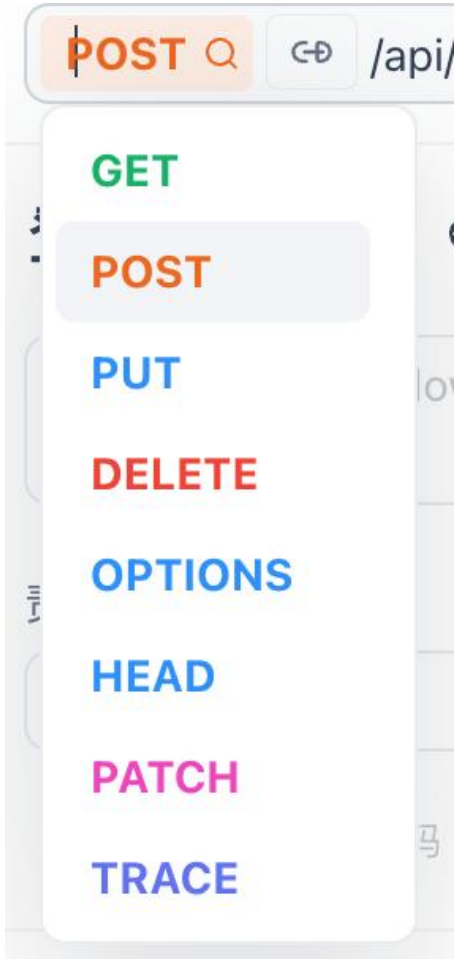
Path: /video/BV17E1pBBEbB/

Query: 实则就是一个 Key-Value Pair

spm_id_from=333.1007.tianma.1-2-2.click
vd_source=a24ab840c2e1c1bb1ba3435ab64fc41c

一个完整的URL：协议://域名/路径?查询参数

Method



Method	用途	社团系统场景示例
GET	查询数据（Retrieve Data，不修改服务器状态）	获取成员列表：GET /api/members
POST	提交新数据（Create Data）	新增成员：POST /api/members + Body携带成员信息

Method

GET 只能用来获得数据吗？只有 POST 才能修改数据吗？

```
app.get('/user/123', (req, res) =>
{ deleteUser(123);
  res.send('user deleted');
});
```

GET/POST只是语义标签！
但我们应该遵守“设计规范”

Header

```
headers = {
    # Authorization: 身份验证（必须登录才能访问）
    "Authorization": "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",

    # Content-Type: 告诉服务器"我发的数据是什么格式"
    "Content-Type": "application/json",  # JSON格式（最常用）
    # "Content-Type": "application/x-www-form-urlencoded",  # 表单格式
    # "Content-Type": "multipart/form-data",  # 文件上传

    # Accept: 告诉服务器"我想要什么格式的响应"
    "Accept": "application/json",  # 希望返回JSON

    # User-Agent: 标识客户端类型
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)"
}
```

Authorization

用途：验证身份的凭证，让服务器识别请求者是否合法。

格式：Authorization: Bearer <你的Token>

机制：服务器检查 token 是否有效（正确且未过期），决定是否允许访问接口。

为什么不使用用户名和密码作为凭证呢？

安全：Token 可以随时撤销或过期，不必暴露账号密码。

灵活：可以为不同的应用、不同权限生成不同的 Token。

简单：HTTP 请求只需带 Token，服务器就能识别你。

Content-Type

告诉服务器或客户端发送的数据类型

格式：Content-Type: <media-type>

Content-Type	描述
application/json	JSON 数据
application/x-www-form-urlencoded	表单数据（类似 key=value&key2=value2）
text/event-stream	SSE（Server-Sent Events）数据流，用于服务器向浏览器推送事件

Body —— JSON

```
Content-Type: application/json

Body示例:
{
  "name": "张三",
  "age": 20
}
```

Body —— 表单格式

```
Content-Type: application/x-www-form-urlencoded

Body示例:
name=%E5%BC%A0%E4%B8%89&age=20
```

解码后:
name=张三
age=20

不支持嵌套结构
只能使用 URL 编码

Body —— JSON

```
Content-Type: application/json

Body示例:
{
  "name": "张三",
  "age": 20
}
```

Body —— JSON

```
{
  "user": {
    "name":"张三",
    "age":20
  }
}
```

PART - 3

HTTP响应

HTTP Response

前后端是用报文进行交互的，无论是HTTP请求还是HTTP响应

包含起始行 Request Line、头部 Header、空行 CRLF 和响应体 Body

```
HTTP/1.1 200 OK                          ← 状态行（协议版本 状态码 状态描述）
Content-Type: application/json           ← 响应头
Cache-Control: no-cache
Transfer-Encoding: chunked

{"message": "操作成功", "data": [...]}     ← 响应体（实际数据）
```

状态码 Status Code

状态码	含义	社团系统场景
200	成功	获取成员列表成功
201	创建成功	新增成员成功
400	请求参数错误	提交表单时缺少必填字段
401	未授权	未登录访问需要权限的接口
404	资源不存在	访问不存在的成员ID
500	服务器错误	数据库连接失败

Header

与HTTP请求类似，HTTP请求也有很多的 Header 参数

我们将在实例中详细介绍他们...

Code with Python...

大家已经熟悉了 Header、Body 的概念

我们将会呈现出一个完成的请求-响应的过程，在这个过程中深化对这两个知识的理解

PART - 4

流式处理

非流式响应

服务器处理完所有数据后，一次性返回给客户端

缺点：

数据量大时，用户等待时间长（如生成报表需要10秒）

无法实时显示进度（AI生成文章要等全部完成才能看到）

想象一下，如果我们的AI应用需要全部生成后才给我们返回，那我们的体验感就会差很多了

流式响应

边生成边返回（现代AI应用标配）

技术	适用场景	连接方式	数据方向
Chunked	大文件传输、日志流	HTTP分块传输	单向
WebSocket	在线游戏、协同编辑	独立协议	双向
SSE	AI聊天、实时通知	HTTP长连接	单向（服务器→客户端）

SSE（Server-Sent Events）

通俗理解： SSE就像"订阅新闻推送"

- 客户端发起连接后保持打开状态
- 服务器有新数据就立即推送（data: 消息内容\n\n格式）
- 适合单向推送场景（服务器主动发，客户端被动收）

典型应用：

- AI聊天（逐字返回答案）
- 股票行情（实时价格推送）
- 系统通知（新消息提醒）

SSE（Server-Sent Events）

```
data: 这是第一条消息
id: 1
event: message

data: 这是第二条消息
data: 可以分多行
id: 2

data: {"type":"ai_response","content":"你好"}
```

04 流式处理

OpenAI API

```
data: {
  "id": "019a8fdafd091c809cf14705cced6e75",
  "object": "chat.completion.chunk",
  "created": 1763350084,
  "model": "Qwen/Qwen2.5-7B-Instruct",
  "choices": [
    {
      "index": 0,
      "delta": {
        "content": "",
        "reasoning_content": null,
        "role": "assistant"
      },
      "finish_reason": null
    }
  ],
  "system_fingerprint": "",
  "usage": {
    "prompt_tokens": 30,
    "completion_tokens": 0,
    "total_tokens": 30
  }
}
```

在标准的 SSE 协议之上，提供一个功能更强大、信息更完整的流式数据载荷Payload

字段	设计目的
data: {JSON}	利用 SSE 机制进行实时推送。
id	关联所有数据块，用于追踪**同一请求**。
object	告诉这是一个聊天完成流中的片段。
model	方便记录和显示使用了哪个模型。
choices	预留了支持 n > 1 并行生成的能力。
delta.content	专门用来承载**增量的文本输出**，实现打字机效果。
usage	提供本次请求的总 Token 消耗，用于计费和统计。
finish_reason	内容生成停止的原因（"stop", "length", "tool_calls" 等）。

Code with Python...

Homework

使用 GitHub Classroom: https://classroom.github.com/a/A-tj83T9

要求

- 获得一个 API Key，并且在.env文件中将 API Key 添加进去。（推荐使用硅基流动）
- 补充未完成的app.py程序，并且试着运行，看看能否实现流式响应。
- 一切完成之后，使用 Git 提交你的app.py。请不要提交你的.env文件！！！

Reference

- 大模型们（ChatGPT、Gemini...）
- 硅基流动API手册：https://docs.siliconflow.cn/cn/api-reference/chat-completions/chat-completions

博远信息技术社
Boyuan IT Club

THANKS