

yaarxiv介绍

2020年9月29日

目录

- 项目简介
- 项目需求和演示
- 架构、技术栈
 - 总体
 - 前端、后端、部署、DevOps
- 展望

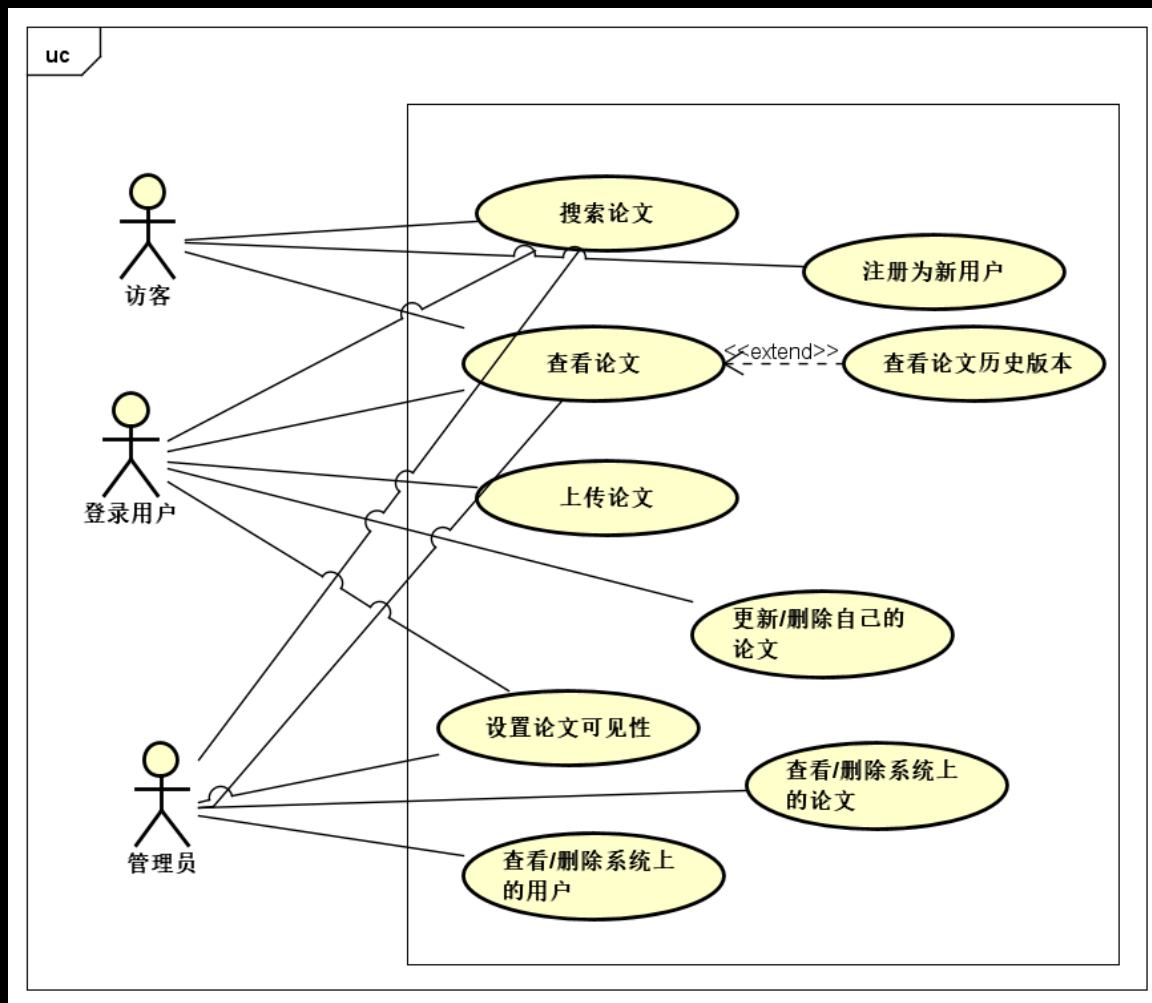
项目介绍

- 名字: yaarxiv (yet another arxiv) (要改)
- 需求: 国内缺少一个类似arxiv的论文平台
 - 中科院的chinaxiv: 缺乏维护, 体验不佳
- 定位: 类似arxiv的论文预印本平台

The logo for 'yaarxiv' is displayed in a bold, blue, sans-serif font. The letters are lowercase, and the 'y' has a distinctive shape with a small gap at the top. The 'i' has a dot. The 'x' is formed by two intersecting lines. The 'v' is a simple, slightly rounded shape. The overall appearance is clean and modern.

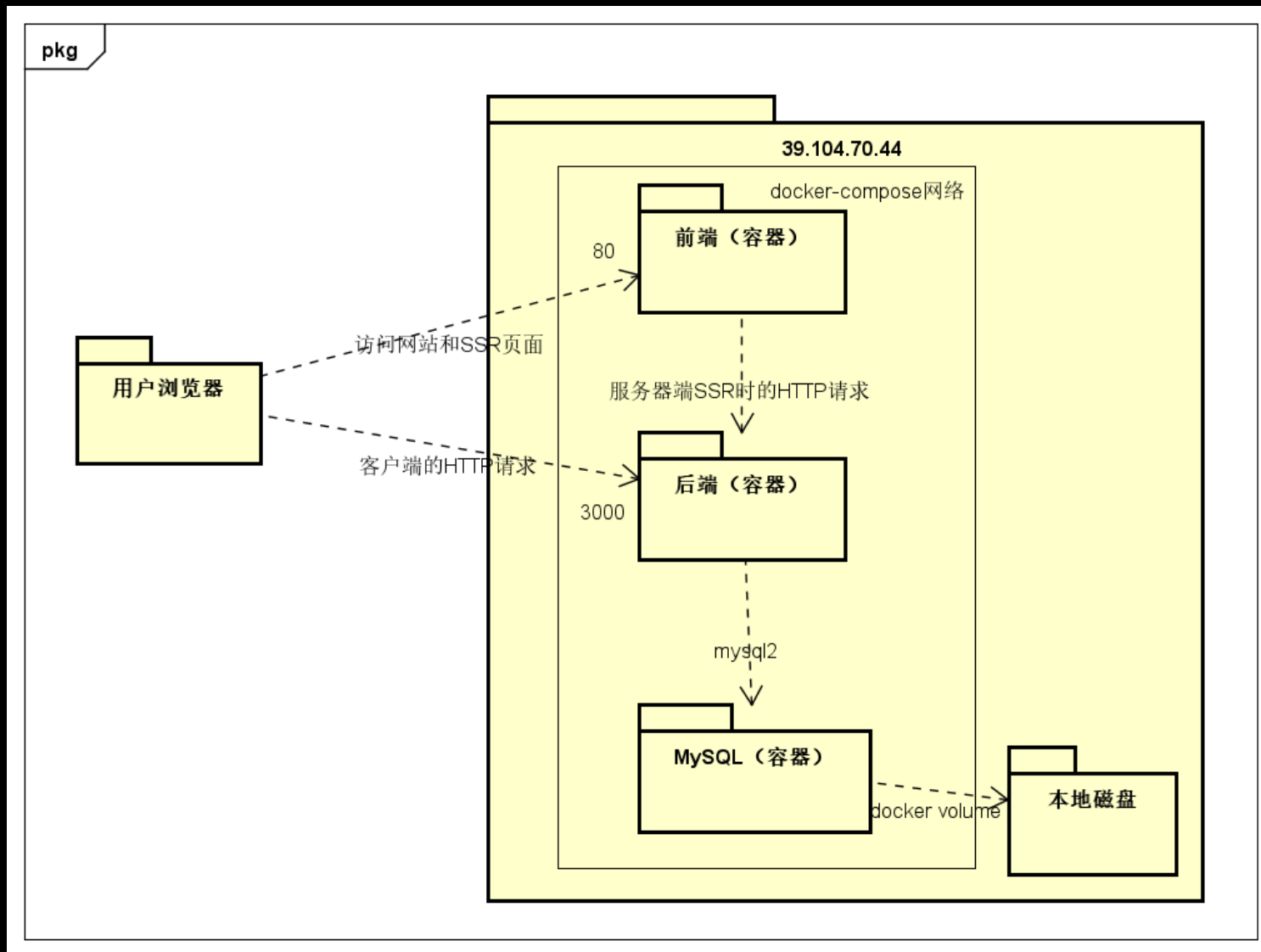
Svg格式的logo, 使用主题色#0099e5

项目需求和演示



当前的用例图

架构



架构图以及用户访问时各个部分的交互图

技术栈

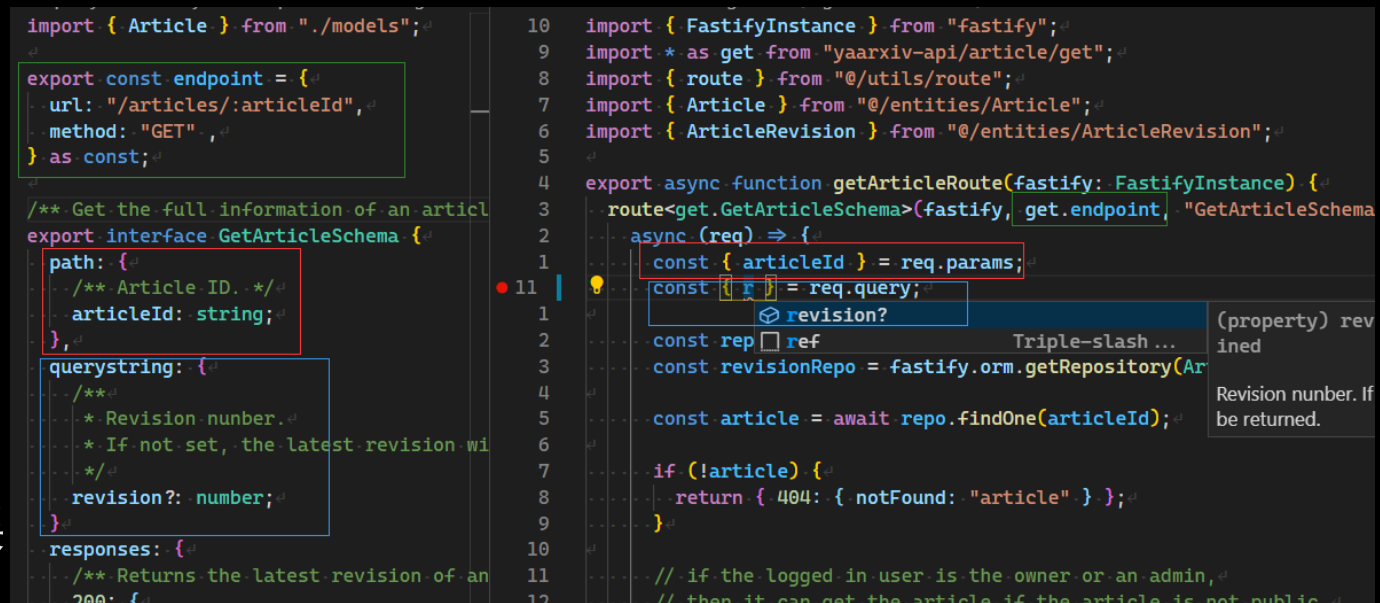
- 前端：
 - React + Next.js + TypeScript
 - 考虑移动端体验，响应式布局，全flexbox
 - 服务器端渲染（SSR）
 - 搜索引擎优化（SSO）友好
 - 加载快（服务器直接发渲染好的HTML/CSS）
 - 可记忆的国际化、暗色模式……
 - 使用自己的轮子或者轻的框架
 - 状态管理：simstate
 - 国际化：simstate-i18n
 - UI库：grommet
 - CSS-in-JS
 - 可切换主题
 - IE11



前端获得的是渲染好的HTML/CSS

技术栈

- 后端：
 - 常规的单体HTTP API
 - Fastify: 速度快的Node.js HTTP框架
 - TypeScript: 类型系统和类型推断
 - Webpack打包
- 独立的API项目
 - 自动生成前后端API的接口定义
 - 给后端提供JSON Schema
 - 加速JSON序列化/反序列化
 - 参数验证 (validation) : 安全性
 - 自动生成swagger文档



```
import { Article } from "../models";

export const endpoint = {
  url: "/articles/:articleId",
  method: "GET",
} as const;

/** Get the full information of an article */
export interface GetArticleSchema {
  path: {
    /** Article ID */
    articleId: string;
  },
  querystring: {
    /**
     * Revision number.
     * If not set, the latest revision will be returned.
     */
    revision?: number;
  },
  responses: {
    /** Returns the latest revision of an article */
    200: {
      type: "object",
      properties: {
        article: Article;
      };
    };
  };
}

import { FastifyInstance } from "fastify";
import * as get from "yaarxiv-api/article/get";
import { route } from "@utils/route";
import { Article } from "@entities/Article";
import { ArticleRevision } from "@entities/ArticleRevision";

export async function getArticleRoute(fastify: FastifyInstance) {
  route<get.GetArticleSchema>(fastify, get.endpoint, "GetArticleSchema")
    .async(req) => {
      const { articleId } = req.params;
      const { revision } = req.query;

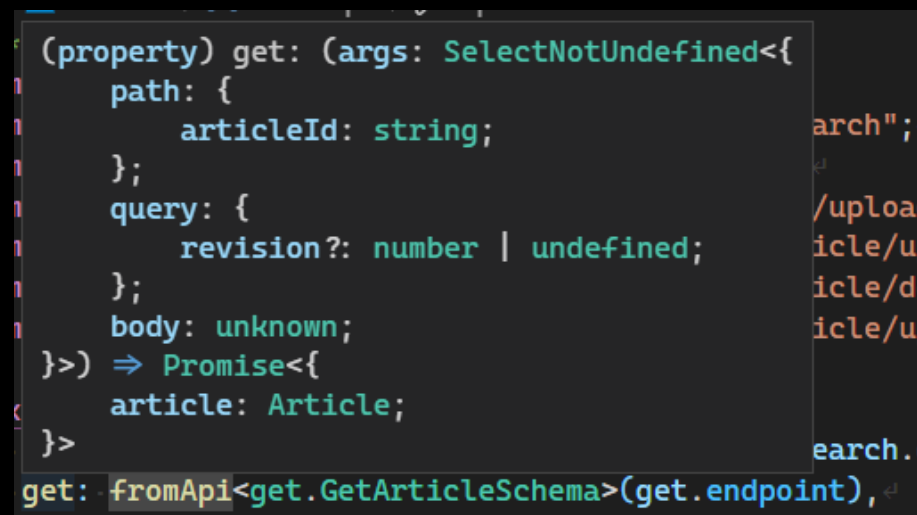
      const repo = fastify.orm.getRepository(ArticleRevision);
      const revisionRepo = fastify.orm.getRepository(ArticleRevision);

      const article = await repo.findOne(articleId);

      if (!article) {
        return { 404: { notFound: "article" } };
      }

      // if the logged in user is the owner or an admin,
      // then it can get the article if the article is not public
    }
}
```

后端使用API定义的接口



```
(property) get: (args: SelectNotUndefined<{
  path: {
    articleId: string;
  };
  query: {
    revision?: number | undefined;
  };
  body: unknown;
}>) => Promise<{
  article: Article;
}>

get: fromApi<get.GetArticleSchema>(get.endpoint),
```

前端自动通过Schema推断参数和返回值类型

后端性能对比

- <https://github.com/ddadaal/yaarxiv-backend-benchmark>
- GET /articles/{articleId}
- 需要执行三个SQL查询语句

```
select * from article
... where id = {articleId};
select time, revisionNumber from article_revision
... where articleId = {articleId};
select * from article_revision r
... join pdf_upload p on r.pdfId = p.id
... where r.articleId = {articleId}
... and r.revisionNumber = {revisionId};
```

- 10个并发连接，连续运行20秒压力测试
- 用C# (ASP.NET Core)和Java (Spring Boot) 实现同样的API

后端性能对比结果

- I7 9700 + 32G 2666Mhz + NVMe SSD + Windows 10 Pro
- Commit: c188a35; 2020年9月28日

技术栈	平均响应量/秒 (QPS)	错误
Fastify + TypeScript	702.85	0
ASP.NET Core + EF Core (C#)	510.95	0
Spring Boot + JPA + tomcat (Java)	551.4	28
Spring Boot + JPA + netty (Java)	544.1	0

各种后端实现的压力测试结果

测试

- 只写了**后端集成测试**
 - 后端单元测试：业务逻辑简单
 - 前端测试：复杂、需求多变且不定
 - E2E：同上
- 使用与生成环境相同的MySQL进行测试
 - 未使用SQLite、H2等
 - CI流水线中的测试也使用了MySQL容器
- 79个测试
- 行覆盖率97%

File	% Stmts	% Branch	% Funcs	% Lines
All files	97.69	82.29	96.83	97.88
config	100	100	100	100
default.ts	100	100	100	100

Jest --coverage

Ops

- <https://github.com/ddadaal/yaarxiv>
- 单机docker-compose部署
- 单机MySQL
- CI: GitHub Actions
 1. 测试
 2. 构建Docker镜像（前后端各一个）
 3. 发布到Dockerhub
- CD:
 1. 上传镜像后向部署服务器发送HTTP请求
 2. 部署服务器从Dockerhub更新本机docker镜像
 3. 部署服务器重启服务

```
const server = http.createServer((req, res) => {
  // validate the deploy request
  if (
    req.method !== "POST" ||
    req.headers["content-type"] !== "text/plain" ||
    req.headers[authHeader] !== deployAgentKey
  ) {
    res.writeHead(403).end();
    return;
  }

  let data = "";
  req.on("data", (d) => {
    data += d;
    if (data.length > dataLengthLimit) {
      data = "";
      res.writeHead(413, {'Content-Type': 'text/plain'}).end();
      req.connection.destroy();
    }
  });

  req.on("end", () => {
    if (data === "frontend" || data === "backend") {
      console.log(`${data} deployment requested. Trying to update deployment.`);
      spawn(`docker-compose pull ${data} && docker-compose up -d ${data}`, { stdio: "inherit", shell: true });
      res.writeHead(200);
    } else {
      res.writeHead(400);
    }
  });
});
```

部署服务器监听部署请求的node.js服务

All workflows				
Filter workflows				
89 results				
Event Status Branch Actor				
✓ only select part of article revision info	Build and Publish backend #42: Commit a8ae90a pushed by ddadaal	master	🕒 13 hours ago 🕒 4m 36s	...
✓ use updated grommet-icons icons	Build and Publish frontend #47: Commit 1e2b200 pushed by ddadaal	master	🕒 14 hours ago 🕒 2m 15s	...
✓ use updated grommet-icons icons	Build and Publish backend #41: Commit 1e2b200 pushed by ddadaal	master	🕒 14 hours ago 🕒 4m 52s	...
✓ jump back to previous page after login	Build and Publish frontend #46: Commit 0564c4d pushed by ddadaal	master	🕒 3 days ago 🕒 2m 16s	...
✓ refresh article page when user state changes	Build and Publish frontend #45: Commit 01b1647 pushed by ddadaal	master	🕒 3 days ago 🕒 2m 24s	...
✓ remember language setting	Build and Publish frontend #44: Commit 0ecd6a7 pushed by ddadaal	master	🕒 4 days ago 🕒 2m 20s	...

每次commit时执行的workflow

展望

功能

- 高级搜索
- 学术机构邮箱认证
- 统计
- 监控
- [OAI学术标准接口](#)

测试

- 更完善的测试
- 多样测试环境

性能

- Cache
- CQRS

可扩展

- 多机部署
- 对象存储
- CI/CD流水线

其他

- 扩大测试范围
- 需求
- 安全性
- 法律问题
- 对外合作

谢谢！