用 Slidev 写 PPT

PPT 的功能

什么是 Slidev, 有哪些特点?

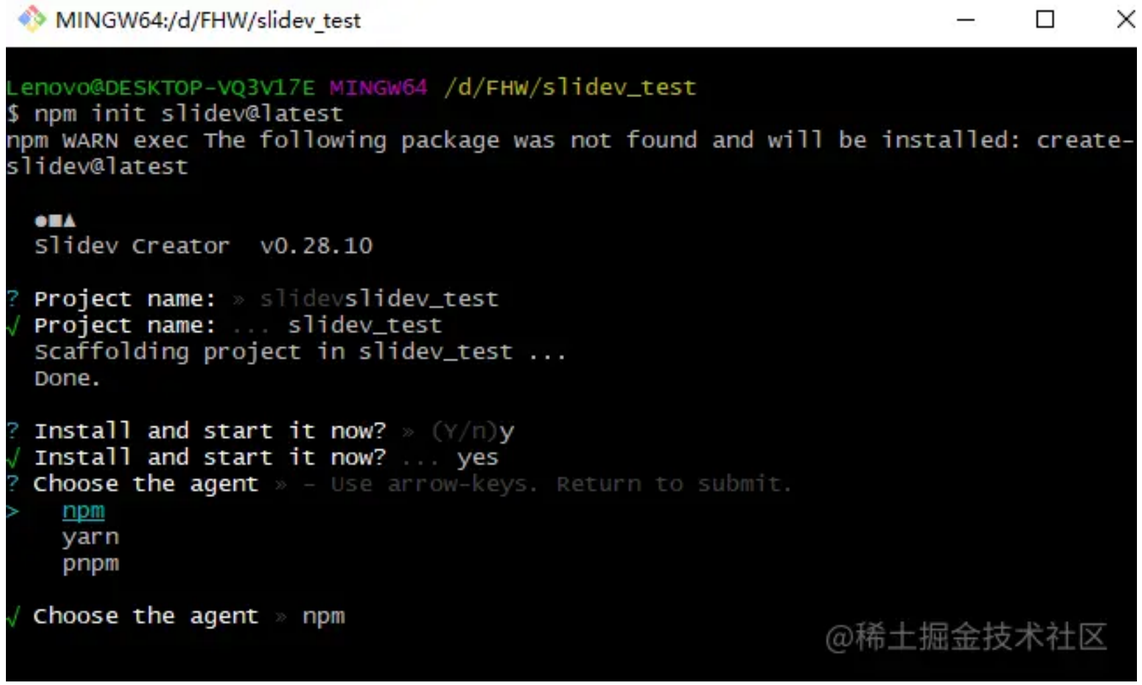
简而言之，Slidev 就是一个工具库，基于 Node.js、Vue.js ， 使用 Markdown 语法 辅以 tailwindcss 等模块来制作 PPT。

- 📝 **Markdown 基本语法支持**
- 🌈 **TailwindCSS 灵活样式**
- 🎨 **可选主题** - 当前只有官方主题可用 可访问几乎所有的开源图标集
- 🌟 **图标** —— 能够直接从任意图标库中获取图标
- 🎥 **可交互** - 嵌入 Vue 组件
- 🤹 🧑‍💻 **开发者友好** - 内置代码高亮(无自动补全功能)
- 🎙 **演讲者模式**
- 📤 **跨平台** - 导出 PDF、PNG 和单页面应用

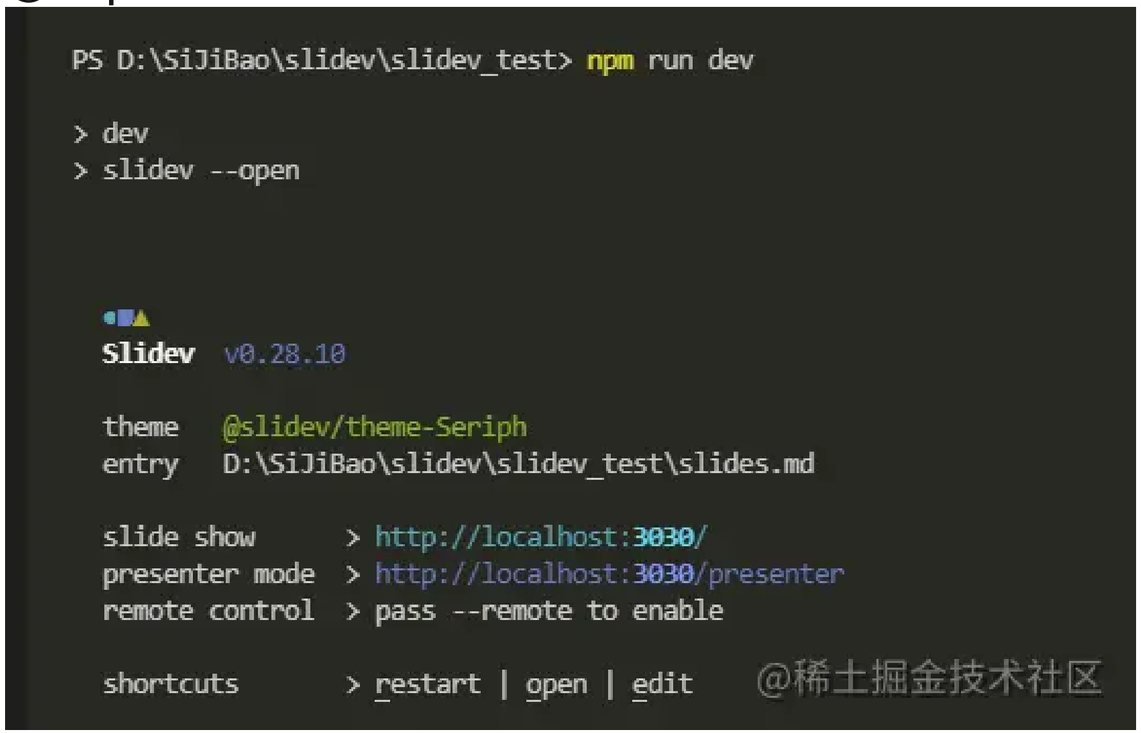
1. 安装和启动

Slidev 需要 Node.js 的版本 >=14.0.0

①. npm init slidev@latest



②. npm run dev



2. 项目结构

```
your-slidev/
  ├── components/       # 自定义组件
  ├── global-bottom.vue # 页脚
  ├── global-top.vue # 页眉
  └── slides.md         # 幻灯片主入口
```

Ⅰ. components

此目录中的组件可以在markdown中直接使用，其组件名与文件名相同。

```
your-slidev/
  ├── ...
  └── components/
      └── counter.ts
```

- 10 +

我是页眉

Ⅱ. 页眉页脚

约定：根目录新增文件名为 global-top.vue | global-bottom.vue即为页眉页脚

```
your-slidev/
  ├── ...
  ├── components/
  ├── global-bottom.vue
  └── global-top.vue
```

Ⅲ. slides.md，入口文件

Markdown 语法 + TailwindCss

我是页脚

1. 常用 MarkDown 语法

1.1 分页

```
第一页
---
第二页
---
第三页
---
```

1.2 代码块、内联样式

```
interface TableSearch {
  settlementBillNo: string | number; //采购结算单号
  warehouseBillNo: string | number; //所属入库单
  merchandiseName: string; //商品名称
  supplierName: string; //供应商名称
  paymentMode: string; //是否可线上支付
}
```

内联样式

1.3. 自定义布局

layout:two-cols(自定义布局名称)

333

999ee

555

3
2
1

1.4. 静态资源

1.5. 图表

`right` / `space`	1	下一个动画或幻灯片
`left` / `shift` `space`	2	上一个动画或幻灯片
`up`	3	上一张幻灯片
`down`	4	下一张幻灯片

2. 导航

将鼠标悬停在左下角以查看导航的控制面板

快捷键	按钮	说明
`f`		切换全屏
`right` / `space`		下一动画或幻灯片
`left`		上一动画或幻灯片
`up`	-	上一张幻灯片
`down`	-	下一张幻灯片
`o`		切换 幻灯片总览
`d`		切换暗黑模式
-		切换 摄像头视图
-		演讲录制
-		进入 演讲者模式
-		切换 集成编辑器
-		下载幻灯片 (仅在 单页（SPA）构建 中支持)
-		显示该演示文稿的信息
-		显示设置菜单

3. 动画.

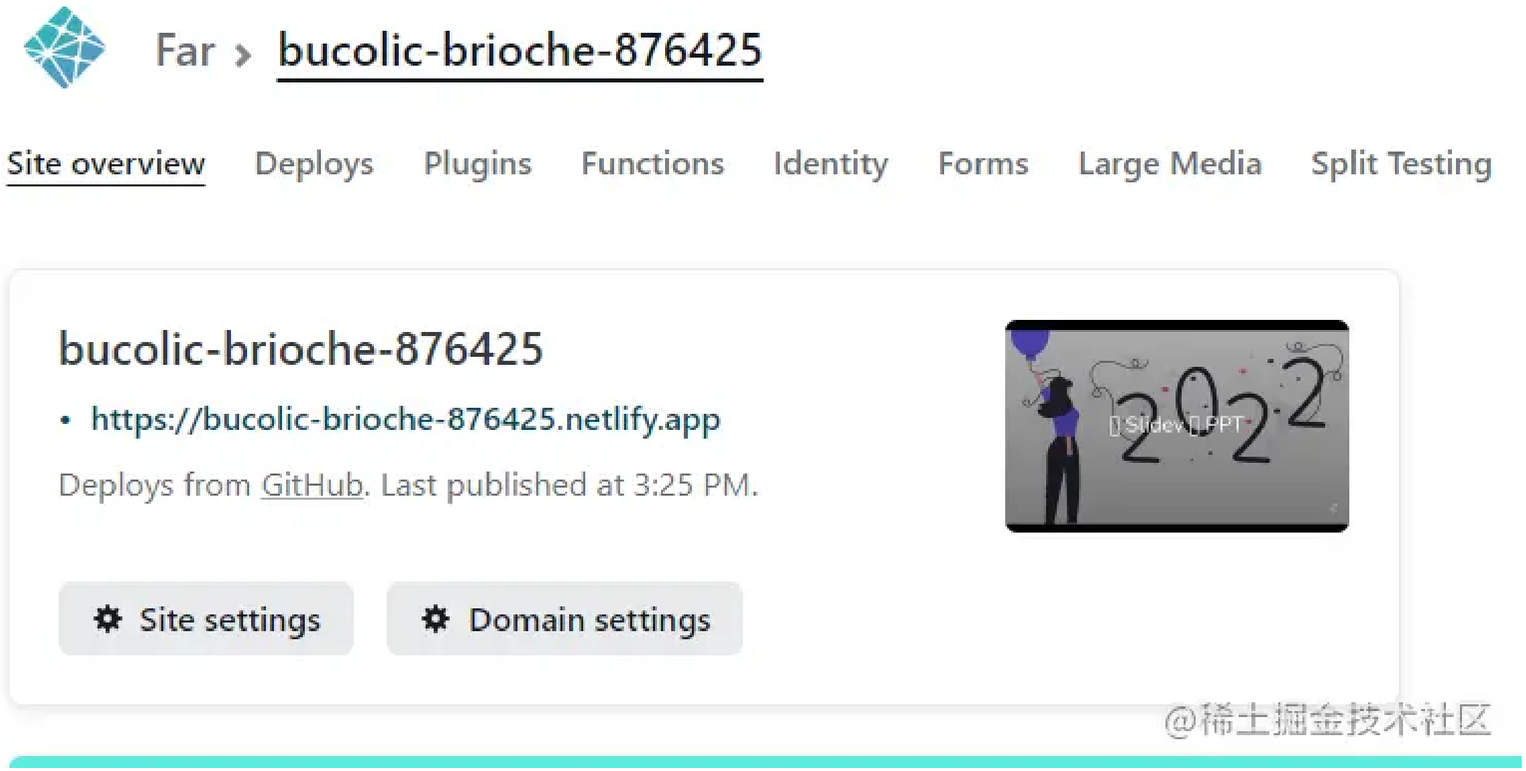
```
<div v-motion :initial="{ x: -80 }" :enter="{ x: 0 }">Slidev</div>
```

Slidev 会预加载下一张幻灯片以提高性能，导致动画无法被看见，可以禁用幻灯片预加载或使用v-if来控制

```
--- preload: false ---
<div
  v-if="$slidev.nav.currentPage === 12"
  v-motion
  :initial="{ x: -80 }"
  :enter="{ x: 0 }"
>
  Slidev
</div>
```

4.部署(Netlify、Vercel、GitHub Pages)

4.1 Netlify 部署



4.2 Vercel 部署

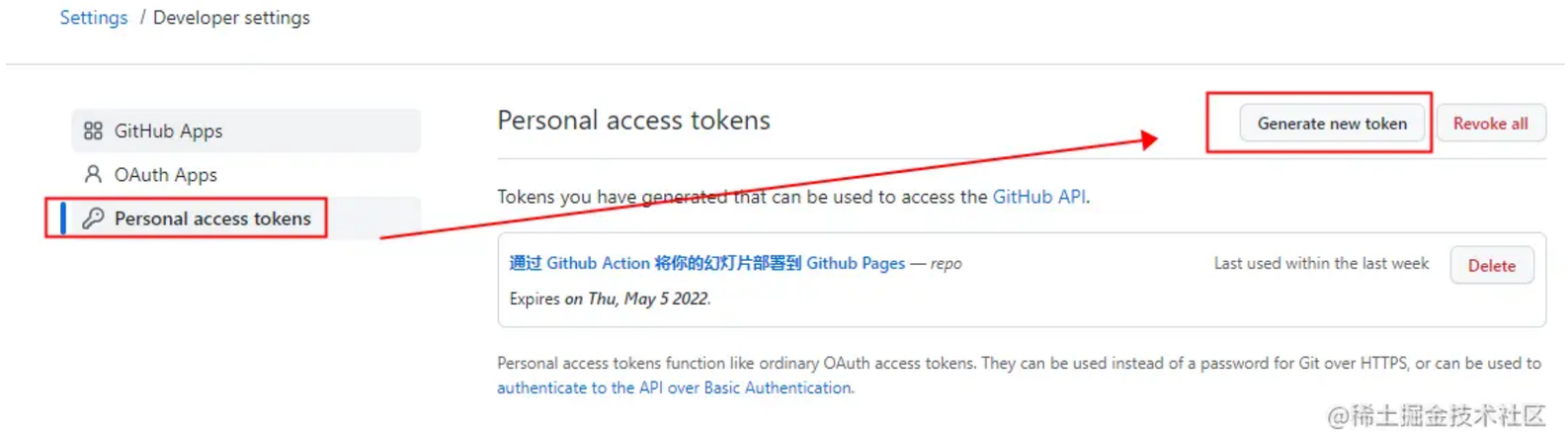
4.3 通过 Github Action 部署到 GitHub Pages

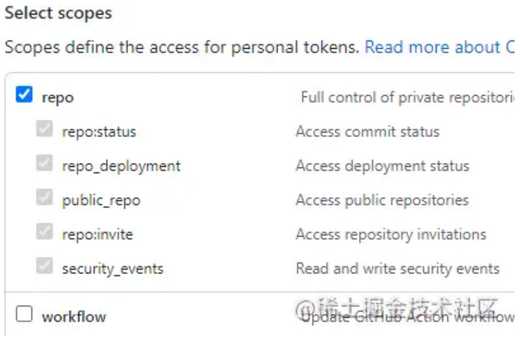
a. 创建 .github/workflows/deploy.yml 文件

Github 只要发现该目录中由 yml 文件就会自动运行该文件。

```yaml
name: Deploy pages
on: push
jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - uses: actions/setup-node@v2
        with:
          node-version: '14'
      - name: Install dependencies
        run: npm install
      - name: Build
        run: npm run build
      - name: Deploy pages
        uses: crazy-max/ghaction-github-pages@v2
        with:
          build_dir: dist
        env:
          GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }} //下一页中配置
```

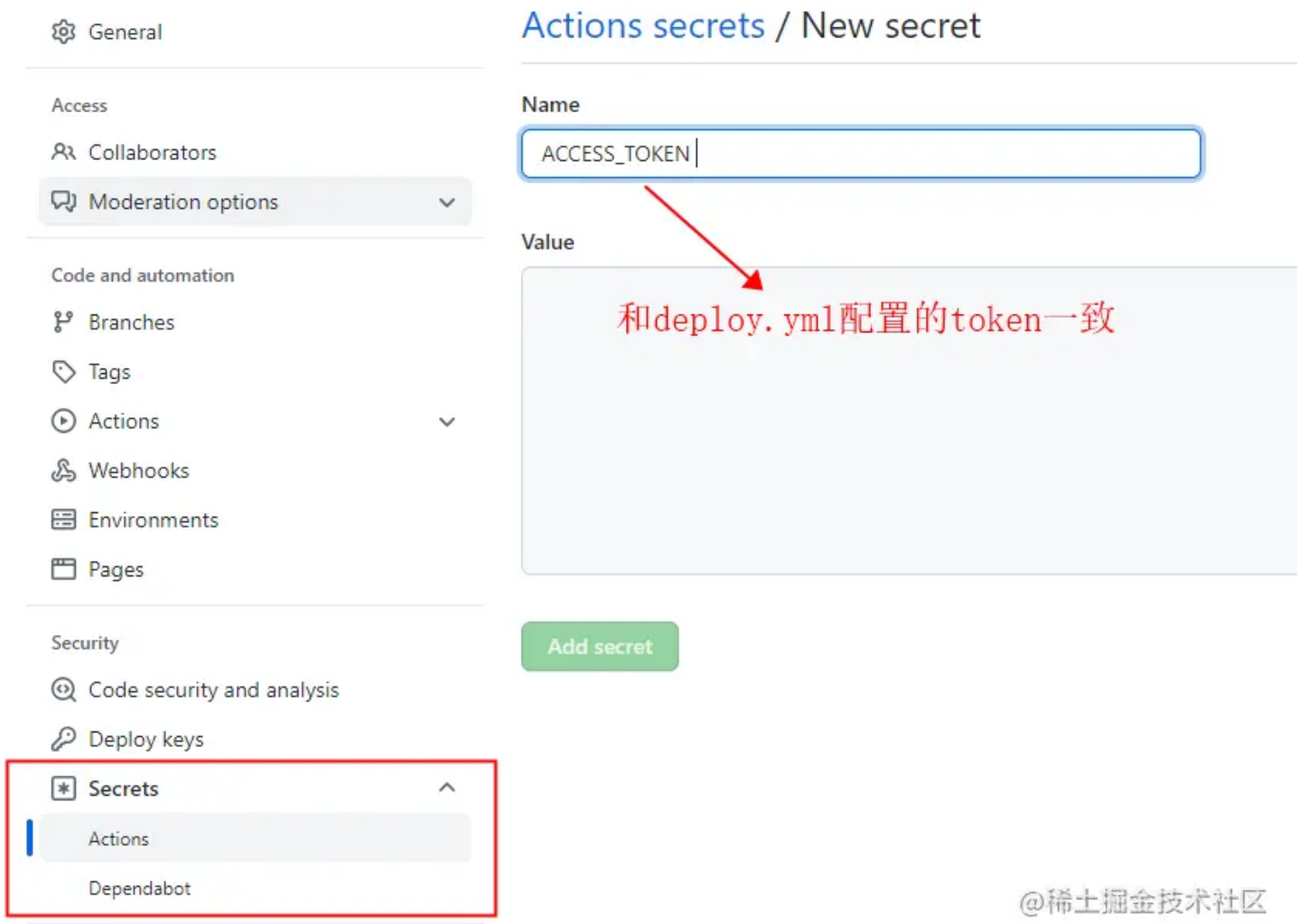
b. 获取密钥并存储到 Github 仓库中





获取到token。

c. 给项目设置 token



d.代码提交到 GitHub 后，进入.github/workflows/deploy.yml 点击 View runs 查看部署情况