

RAG를 활용한 프로젝트 분석 및 설계

이 발표에서는 RAG를 이용하여 기존의 프로젝트를 분석하거나 프로젝트를 수행하는 과정에 대해서 설명할 예정입니다. CodeSage라는 오픈 소스를 사용하여 발표를 진행할건데요. 마무리 과정에서 오픈소스 인공지능을 활용하여 개선된 부분에 대해서도 살펴볼 예정입니다.

CodeSage에 대한 소개

CodeSage는 사용자가 참고자료가 있는 폴더를 지정하고 질문을 하면, 인공지능이 폴더의 모든 자료 중에서 필요한 것만을 참고해서 답변을 해주는 프로그램입니다. 주로 사용되는 형태는 다음과 같습니다.

- 프로젝트 소스 코드가 있는 폴더 지정
- 노선의 자료를 다운받아서 사용
- 기존 답변을 참고자료로 활용
- PDF 문서 참고 기능

설치 방법

CodeSage의 소스코드는 아래의 링크에서 다운받을 수 있습니다. 자세한 설치 및 사용법은 아래 링크의 문서를 참고하시기 바랍니다.

- <https://github.com/ryujt/CodeSage>

pip 사용

파이썬이 설치되어 있다면 아래의 스크립트로 간단하게 설치하여 이용하실 수 있습니다.

```
pip install -r requirements.txt
```

docker 사용

Docker를 사용하면 배포하고 실행할 수도 있습니다.

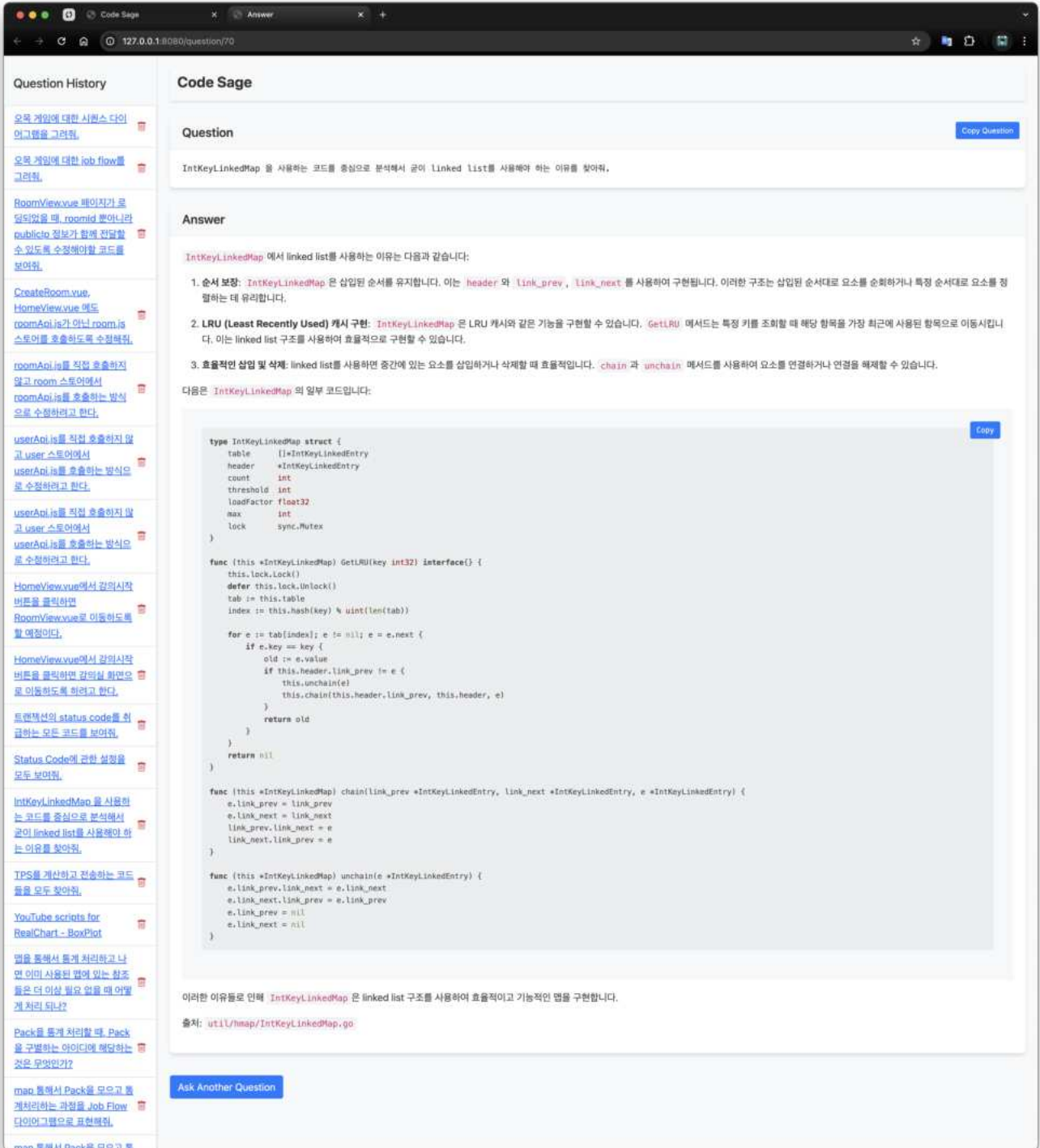
```
docker build -t codesage:latest .  
docker run -d -p 8080:8080 --name codesage_container codesage:latest
```

동작확인

CodeSage가 프로젝트 수행에 어떠한 도움을 주는 지에 대해서 다음과 같이 알아보겠습니다. 이 사전자료 문서에서는 스크린 샷만을 보여드리고 발표장에서 자세한 설명을 드릴 예정입니다. 스크린샷의 길이가 긴 것은 하단 부분을 잘라서 사용했습니다.

기존 프로젝트의 내용을 분석

질문의 내용이 전체 코드에 의해서 영향을 받을 때



문서를 모두 검색해서 원하는 답변을 얻고 싶을 때

Question History

- 오류 계입에 대한 시퀀스 다이어그램을 그려줘.
- 오류 계입에 대한 job flow를 그려줘.
- RoomView.vue 페이지가 로딩되었을 때, roomId 뿐만 아니라 publicId 정보가 함께 전달할 수 있도록 수정해야 할 코드를 보여줘.
- CreateRoom.vue, HomeView.vue 예도 roomApi.is가 아닌 room.is 스토어를 호출하도록 수정해줘.
- roomApi.is를 직접 호출하지 않고 room 스토어에서 roomApi.is를 호출하는 방식으로 수정하려고 한다.
- userApi.is를 직접 호출하지 않고 user 스토어에서 userApi.is를 호출하는 방식으로 수정하려고 한다.
- userApi.is를 직접 호출하지 않고 user 스토어에서 userApi.is를 호출하는 방식으로 수정하려고 한다.
- HomeView.vue에서 강의시작 버튼을 클릭하면 RoomView.vue로 이동하도록 할 예정이다.
- HomeView.vue에서 강의시작 버튼을 클릭하면 강의실 화면으로 이동하도록 하려고 한다.
- 트랜잭션의 status code를 취급하는 모든 코드를 보여줘.
- Status Code에 관한 설정을 모두 보여줘.
- IntKeyLinkedMap을 사용하는 코드를 중심으로 분석해서 코인 linked list를 사용해야 하는 이유를 찾아줘.
- TPS를 계산하고 전송하는 코드들을 모두 찾아줘.
- YouTube scripts for RealChart - BoxPlot
- 백을 통해서 통계 처리하고 나면 이미 사용된 면에 있는 참조들은 더 이상 필요 없을 때 어떻게 처리 되나?
- Pack을 통해 처리할 때, Pack을 구별하는 아이디어에 해당하는 것은 무엇인가?
- map 통해서 Pack을 모으고 통계 처리하는 과정을 Job Flow 다이어그램으로 표현해줘.
- map 통해서 Pack을 모으고 통계

Code Sage

Question Copy Question

Status Code에 관한 설정을 모두 보여줘.

Answer

현재 제공된 문서에서 "Status Code"에 관한 설정을 직접적으로 다루는 내용은 찾을 수 없습니다. 그러나 HTTP 상태 코드와 관련된 몇 가지 설정을 포함한 문서들이 있습니다. 다음은 관련된 설정들입니다:

- 1. HTTP 상태 코드 무시 설정:**
 - 특정 HTTP 상태 코드를 무시할 수 있습니다.

```
status_ignore=400,500
```

Copy
- 2. HTTP 상태 코드 예외 무시 설정:**
 - 특정 URL에 대해 HTTP 상태 코드 예외를 무시할 수 있습니다.

```
status_ignore_set=/api/test/timeout/{time}:400,/error:500
```

Copy
- 3. HTTP 상태 코드 예외 수집 설정:**
 - HTTP 상태 코드가 예외인 경우 수집 여부를 설정할 수 있습니다.

```
http_status_error_enable=true
```

Copy
- 4. HTTP 상태 코드 예외 무시 설정:**
 - 특정 HTTP 상태 코드 예외를 무시할 수 있습니다.

```
http_status_ignore=404,500
```

Copy
- 5. HTTP 상태 코드 URL 무시 설정:**
 - 특정 URL에 대해 HTTP 상태 코드 예외를 무시할 수 있습니다.

```
http_status_url_ignore_set=a/b/c:404,d/e/f:500
```

Copy

이 외에도 HTTP 상태 코드와 관련된 다양한 설정이 있을 수 있으므로, 더 구체적인 설정이 필요하다면 추가적인 문서나 설정 파일을 참조하시기 바랍니다.

참고 문서:

- `_java-exception-options.mdx`
- `_java-http-transaction.mdx`

위의 설정 예시를 통해 필요한 설정을 적용해 보시기 바랍니다. 추가적인 정보가 필요하다면 더 구체적인 질문을 해주세요.

Ask Another Question

프로젝트 분석을 위하여 다이어그램 요청

Question History

- 오목 게임에 대한 시퀀스 다이어그램을 그려줘.
- 오목 게임에 대한 job flow를 그려줘.
- RoomView.vue 페이지가 로딩되었을 때, roomId 뿐만 아니라 publicId 정보가 함께 전달할 수 있도록 수정해야 할 코드를 보여줘.
- CreateRoom.vue, HomeView.vue 옆에 roomApi.is가 아닌 room.is 스토어를 호출하도록 수정해줘.
- roomApi.is를 직접 호출하지 않고 room 스토어에서 roomApi.is를 호출하는 방식으로 수정하려고 한다.
- userApi.is를 직접 호출하지 않고 user 스토어에서 userApi.is를 호출하는 방식으로 수정하려고 한다.
- userApi.is를 직접 호출하지 않고 user 스토어에서 userApi.is를 호출하는 방식으로 수정하려고 한다.
- HomeView.vue에서 강의시작 버튼을 클릭하면 RoomView.vue로 이동하도록 할 예정이다.
- HomeView.vue에서 강의시작 버튼을 클릭하면 강의실 화면으로 이동하도록 하려고 한다.
- 트랜잭션의 status code를 취급하는 모든 코드를 보여줘.
- Status Code에 관한 설정을 모두 보여줘.
- IntKeyLinkedMap을 사용하는 코드를 중심으로 분석해서 굳이 linked list를 사용해야 하는 이유를 찾아줘.
- TPS를 계산하고 전송하는 코드들을 모두 찾아줘.
- YouTube scripts for RealChart - BoxPlot
- 뱀을 통해서 동계 처리하고 나면 이미 사용된 면에 있는 칩조들은 더 이상 필요 없을 때 어떻게 처리되나?
- Pack을 통해 처리할 때, Pack을 구별하는 아이디어에 해당하는 것은 무엇인가?
- map 통해서 Pack을 모으고 동계 처리하는 과정을 Job Flow 다이어그램으로 표현해줘.
- map 통해서 Pack을 모으고 동

Code Sage

Question

오목 게임에 대한 시퀀스 다이어그램을 그려줘.

[Copy Question](#)

Answer

오목 게임에 대한 시퀀스 다이어그램을 그리기 위해서는 게임의 주요 이벤트와 그에 따른 객체 간의 상호작용을 정의해야 합니다. 제공된 문서들에서 오목 게임의 주요 이벤트와 객체 간의 상호작용을 추출할 수 있습니다.

오목 게임의 주요 이벤트는 다음과 같습니다:

- 게임 시작 (onStart)
- 마우스 클릭 (onMouseClick)
- AI의 턴 (myTurn)
- 게임 종료 (end)

각 이벤트에 대한 객체 간의 상호작용을 정의하면 다음과 같습니다:

- 게임 시작:
 - User -> omokGame : onStart()
 - omokGame -> omokStage : reset()
 - omokGame -> omokStage : drawStage()
- 마우스 클릭:
 - User -> omokGame : onMouseClick(event)
 - omokGame -> omokStage : onMouseClick(event)
 - omokStage -> omokJudge : isFinished()
 - omokJudge -> omokGame : end()
- AI의 턴:
 - omokGame -> omokAI : myTurn()
 - omokAI -> omokStage : addStone(x, y, color)
 - omokAI -> omokStage : drawStone(x, y)
 - omokStage -> omokJudge : isFinished()
 - omokJudge -> omokGame : end()
- 게임 종료:
 - omokGame -> omokStage : end()

이를 바탕으로 시퀀스 다이어그램을 작성하면 다음과 같습니다:

```
sequenceDiagram
    participant User
    participant omokGame
    participant omokStage
    participant omokJudge
    participant omokAI

    User->>omokGame: onStart()
    activate omokGame
    omokGame->>omokStage: reset()
    deactivate omokGame
    activate omokStage
    omokStage->>omokStage: drawStage()
    deactivate omokStage
    activate omokGame
    User->>omokGame: onMouseClick(event)
    activate omokGame
    omokGame->>omokStage: onMouseClick(event)
    deactivate omokGame
    activate omokStage
    omokStage->>omokJudge: isFinished()
    deactivate omokStage
    activate omokJudge
    omokJudge->>omokGame: end()
    deactivate omokJudge
    activate omokGame
    omokGame->>omokStage: end()
    deactivate omokGame
    activate omokStage
    omokStage->>omokAI: myTurn()
    deactivate omokStage
    activate omokAI
    omokAI->>omokStage: addStone(x, y, color)
    deactivate omokAI
    activate omokStage
    omokStage->>omokStage: drawStone(x, y)
    deactivate omokStage
    deactivate omokAI
```

기존에 없는 자신만의 다이어그램 형식 요청

Question History

- 오목 게임에 대한 시퀀스 다이어그램을 그려줘.
- 오목 게임에 대한 job flow를 그려줘.
- RoomView.vue 페이지가 로딩되었을 때, roomId 뿐만 아니라 publicId 정보도 함께 전달할 수 있도록 수정해야 할 코드를 보여줘.
- CreateRoom.vue, HomeView.vue 옆에 roomApi.is가 아닌 room.is 스토어를 호출하도록 수정해줘.
- roomApi.is를 직접 호출하지 않고 room 스토어에서 roomApi.is를 호출하는 방식으로 수정하려고 한다.
- userApi.is를 직접 호출하지 않고 user 스토어에서 userApi.is를 호출하는 방식으로 수정하려고 한다.
- userApi.is를 직접 호출하지 않고 user 스토어에서 userApi.is를 호출하는 방식으로 수정하려고 한다.
- HomeView.vue에서 강의시작 버튼을 클릭하면 RoomView.vue로 이동하도록 할 예정이다.
- HomeView.vue에서 강의시작 버튼을 클릭하면 강의실 화면으로 이동하도록 하려고 한다.
- 트랜잭션의 status code를 취급하는 모든 코드를 보여줘.
- Status Code에 관한 설정을 모두 보여줘.
- IntKeyLinkedMap을 사용하는 코드를 중심으로 분석해서 코인 linked list를 사용해야 하는 이유를 찾아줘.
- TPS를 계산하고 진술하는 코드들을 모두 찾아줘.
- YouTube scripts for RealChart - BoxPlot
- 뱀을 통해서 동계 처리하고 나면 이미 사용된 면에 있는 칩들은 더 이상 필요 없을 때 어떻게 처리되나?
- Pack을 통해 처리할 때, Pack을 구별하는 아이디어에 해당하는 것은 무엇인가?
- map 통해서 Pack을 모으고 동계 처리하는 과정을 Job Flow 다이어그램으로 표현해줘.
- map 통해서 Pack을 모으고 동

Code Sage

Question Copy Question

오목 게임에 대한 Job flow를 그려줘.

Answer

오목 게임에 대한 Job Flow 다이어그램을 작성하기 위해서는 오목 게임의 주요 이벤트와 메서드 호출을 정의해야 합니다. 아래는 오목 게임의 Job Flow 다이어그램 예시입니다.

OmokGame	OmokStage	OmokJudge	OmokAI
Start	reset		
Start	drawStage		
MouseClicked	MouseClicked		
	MouseClicked	isFinished	
		false	myTurn
	addStone		myTurn
	drawStone		myTurn
		isFinished	myTurn
		true	
end			

이 다이어그램은 오목 게임의 주요 이벤트 호출을 나타냅니다. 예를 들어, OmokGame 객체에서 onStart 이벤트가 발생하면 OmokStage 객체의 reset 및 drawStage 메서드가 호출됩니다. 마우스 클릭 이벤트 (onMouseClicked)가 발생하면 OmokStage 객체의 onMouseClick 메서드가 호출되고, 그 후 OmokJudge 객체의 isFinished 메서드가 호출되어 게임이 끝났는지 확인합니다. 게임이 끝나지 않았다면 OmokAI 객체의 myTurn 메서드가 호출되어 AI가 돌을 놓고, 다시 OmokJudge 객체의 isFinished 메서드가 호출되어 게임 종료 여부를 확인합니다.

이 다이어그램을 통해 오목 게임의 이벤트 흐름을 시각적으로 이해할 수 있습니다.

참고 문서: [JobFlow.md](#)

Ask Another Question

프로젝트 전체 코드를 고려하여 기능 추가하기

인공지능과 페어 프로그래밍 하듯이 프로젝트를 진행하는 예제입니다.

변경된 설계를 프로젝트 전체에 적용하기

Question History

- [오류 메시지에 대한 시퀀스 다이어그램을 그려줘.](#)
- [오류 메시지에 대한 job.flow를 그려줘.](#)
- [RoomView.vue 페이지가 로딩되었을 때, roomId 뿐 아니라 publicId 정보가 함께 전달할 수 있도록 수정해야 할 코드를 보여줘.](#)
- [CreateRoom.vue, HomeView.vue 에도 roomApi.js 가 아닌 room.js 스토어를 호출하도록 수정해줘.](#)
- [roomApi.js를 직접 호출하지 않고 room 스토어에서 roomApi.js를 호출하는 방식으로 수정하려고 한다.](#)
- [userApi.js를 직접 호출하지 않고 user 스토어에서 userApi.js를 호출하는 방식으로 수정하려고 한다.](#)
- [userApi.js를 직접 호출하지 않고 user 스토어에서 userApi.js를 호출하는 방식으로 수정하려고 한다.](#)
- [userApi.js를 직접 호출하지 않고 user 스토어에서 userApi.js를 호출하는 방식으로 수정하려고 한다.](#)
- [HomeView.vue에서 강의시작 버튼을 클릭하면 RoomView.vue로 이동하도록 할 예정이다.](#)
- [HomeView.vue에서 강의시작 버튼을 클릭하면 강의실 화면으로 이동하도록 하려고 한다.](#)
- [트랜잭션의 status code를 취급하는 모든 코드를 보여줘.](#)
- [Status Code에 관한 설정을 모두 보여줘.](#)
- [IntKeyLinkedMap을 사용하는 코드를 중심으로 분석해서 코이 linked list를 사용해야 하는 이유를 찾아줘.](#)
- [TPS를 계산하고 전송하는 코드들을 모두 찾아줘.](#)
- [YouTube scripts for RealChart - BoxPlot](#)
- [팩을 통해서 물게 처리하고 난 뒤 이미 사용된 팩에 있는 합조들은 더 이상 필요 없을 때 어떻게 처리 되나?](#)
- [Pack을 물게 처리할 때, Pack을 구별하는 이디에 해당하는 것은 무엇인가?](#)
- [map 통해서 Pack을 모으고 물게 처리하는 과정을 Job Flow 다이어그램으로 표현해줘.](#)
- [map 통해서 Pack을 모으고 물](#)

Code Sage

Question

userApi.js를 직접 호출하지 않고 user 스토어에서 userApi.js를 호출하는 방식으로 수정하려고 한다. vue에서는 user 스토어를 통해서 signin, signup, signout 액션을 취하고 정보를 전달 받도록 코드를 수정해줘.

Answer

userApi.js를 직접 호출하지 않고 user 스토어를 통해서 signin, signup, signout 액션을 취하도록 코드를 수정하려면 다음과 같이 할 수 있습니다.

- userApi.js 에서 정의된 메서드를 user.js 스토어로 옮깁니다.
- Vue 컴포넌트에서 userApi.js 대신 user.js 스토어를 사용하도록 수정합니다.

1. user.js 스토어 수정

```
// src/stores/user.js
import { defineStore } from 'pinia';
import rest from '@api/rest';

export const useUserStore = defineStore({
  id: 'user',
  persist: true,
  state: () => ({
    token: null,
    username: '',
    userLevel: 0,
    muted: false,
  }),
  getters: {
    loggedIn: (state) => state.token !== null,
  },
  actions: {
    async signin(username, user_password) {
      const data = { user_id: username, user_password };
      try {
        const res = await rest.post('https://zjg2zfoh1a2hautiki7w2deba@hcxle.lambda-url.ap-northeast-2.on.aws', data);
        this.setUsername(username);
        this.setToken(res.data.token);
        return Promise.resolve();
      } catch (error) {
        this.logout();
        return Promise.reject(new Error(error.response.data.message));
      }
    },
    async signup(username, password, password_confirm) {
      if (!username) return Promise.reject(new Error("사용자 이름은 필수 항목입니다."));
      if (!password) return Promise.reject(new Error("비밀번호는 필수 항목입니다."));
      if (password !== password_confirm) {
        return Promise.reject(new Error("비밀번호와 비밀번호 확인이 일치하지 않습니다."));
      }
      const data = { user_id: username, password, password_confirm };
      try {
        const res = await rest.post('https://mccbnviphkwtoknbn24dn9jbfzd.lambda-url.ap-northeast-2.on.aws', data);
        this.setUsername(username);
        this.setToken(res.data.token);
        return Promise.resolve(res.data.message);
      } catch (error) {
        return Promise.reject(new Error(error.response.data.message));
      }
    },
    async signout() {
      const username = this.username;
      this.logout();
      const data = { user_id: username };
      try {
        const res = await rest.post('https://ev0wqymvbgomr54poawlat4u8vszrn.lambda-url.ap-northeast-2.on.aws', data);
        return Promise.resolve(res.data);
      } catch (error) {
        return Promise.reject(new Error(error.response.data.message));
      }
    },
    logout() {
      this.token = null;
      this.username = '';
    },
    setToken(token) {
      this.token = token;
    },
  },
});
```


프로젝트에 추가 기능 개발하기

The screenshot shows a web browser with a question and answer page. The question asks for help with adding a new screen to a Go game. The answer provides a list of files to edit and a code snippet for the `index.html` file.

Question History

- 오목 게임에 대한 시퀀스 다이어그램을 그려줘.
- 오목 게임에 대한 job flow를 그려줘.
- RoomView.vue 메시지가 로딩되었을 때, roomId 뿐만 아니라 publico 정보도 함께 전달할 수 있도록 수정해야 할 코드를 보여줘.
- CreateRoom.vue, HomeView.vue 옆도 roomApi.js가 아닌 room.js 스토어를 호출하도록 수정해줘.
- roomApi.js를 직접 호출하지 않고 room 스토어에서 roomApi.js를 호출하는 방식으로 수정하려고 한다.
- userApi.js를 직접 호출하지 않고 user 스토어에서 userApi.js를 호출하는 방식으로 수정하려고 한다.
- userApi.js를 직접 호출하지 않고 user 스토어에서 userApi.js를 호출하는 방식으로 수정하려고 한다.
- HomeView.vue에서 강의시작 버튼을 클릭하면 RoomView.vue로 이동하도록 할 예정이다.
- HomeView.vue에서 강의시작 버튼을 클릭하면 강의실 화면으로 이동하도록 하려고 한다.
- 트랜잭션의 status code를 뒤집히는 모든 코드를 보여줘.
- Status Code에 관한 설정을 모두 보여줘.
- IntKeyLinkedMap을 사용하는 코드를 중심으로 분석해서 굳이 linked list를 사용해야 하는 이유를 찾아줘.
- TPS를 계산하고 전송하는 코드들을 모두 찾아줘.
- YouTube scripts for RealChart - BoxPlot
- 백을 통해서 동계 처리하고 나면 이미 사용된 면에 있는 칩조들은 더 이상 필요 없을 때 어떻게 처리 되나?
- Pack을 통해 처리할 때, Pack을 구별하는 아이디어에 해당하는 것은 무엇인가?
- map 통해서 Pack을 모으고 동계 처리하는 과정을 Job Flow 다이어그램으로 표현해줘.
- map 통해서 Pack을 모으고 동

Code Sage

Question

오목게임에 결과 화면을 추가해줘. 수정해야 할 파일들을 함께 표시해줘.

Answer

오목 게임에 결과 화면을 추가하려면 다음 파일들을 수정해야 합니다:

- index.html
- omokGame.js
- omokJudge.js

1. index.html 수정

index.html 파일에 결과를 표시할 HTML 요소를 추가합니다. 예를 들어, 결과를 표시할 div 태그를 추가할 수 있습니다.

```
<!DOCTYPE html>
<html>
<header>
<title>오목 게임</title>
<meta charset="utf-8">
<script src="omokGame.js"></script>
<script src="omokStage.js"></script>
<script src="omokJudge.js"></script>
<script src="omokAI.js"></script>
</header>

<body onload="loadGame()" onresize="resizeGame()">
<h1>오목 게임 </h1>
<button onclick="onStart()"> Start </button>
<button onclick="onUndo()"> Undo </button>
<div id="myStatus"> 게임 시작, 진행 차제. </div>
<div id="result" style="display:none"> 결과: </div>
<canvas id="myCanvas" width="500" height="500" style="border:1px solid #d3d3d3;" onmouseover="onMouseMove(event)"
onmousedown="onMouseClicked(event)" onmouseout="onMouseOut(this)">
Your browser does not support the HTML5 canvas tag.</canvas>

<script>
var game = null;
function loadGame() {
game = new omokGame("small");
game.start(); // Automatically start the game
}

function resizeGame() {
//game.resizeStage();
}

function onStart() {
game.start();
}

function onUndo() {
game.undo();
}

function onMouseMove(event) {
game.onMouseMove(event);
}

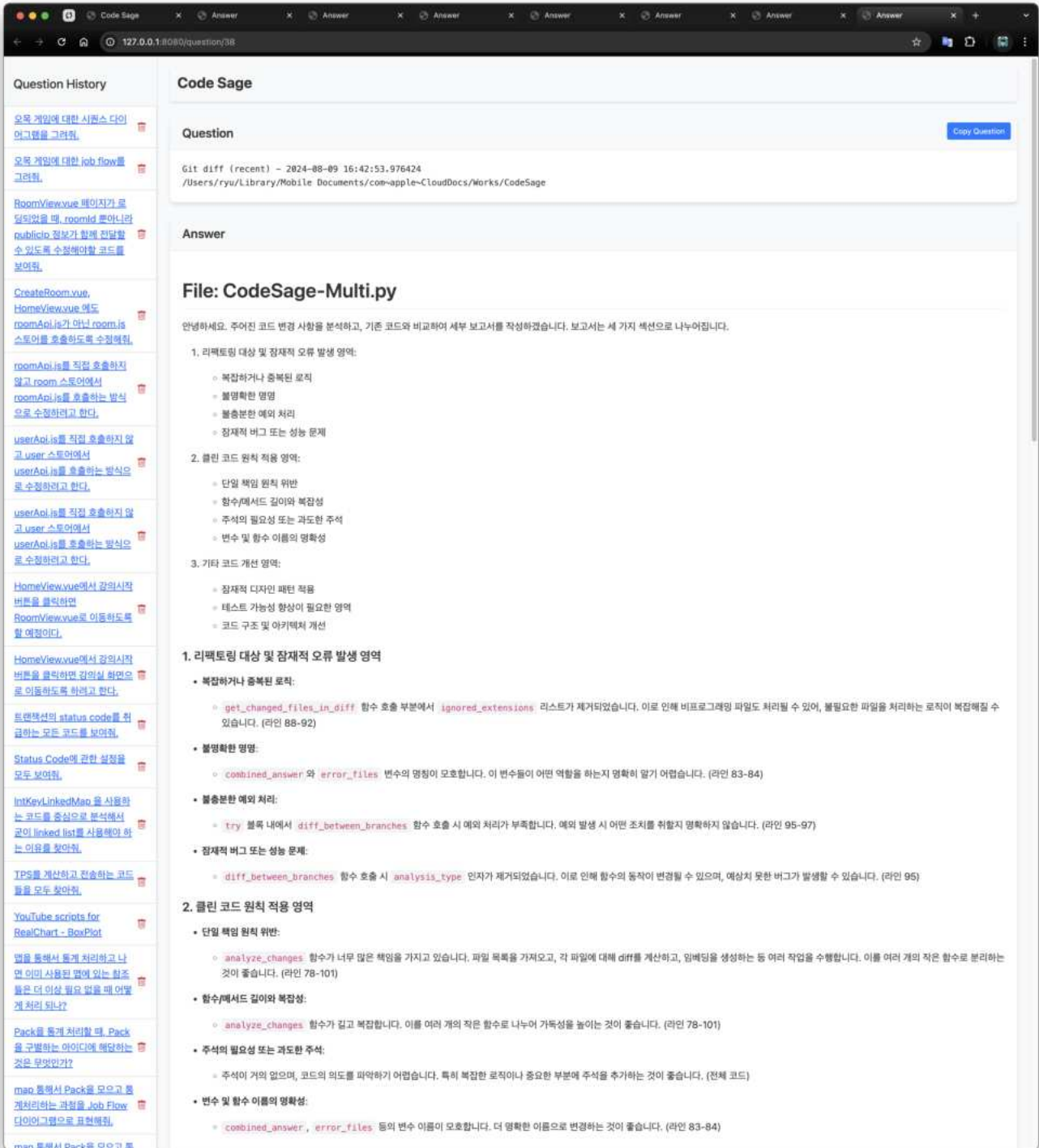
function onMouseClick(event) {
game.onMouseClick(event);
}

function onMouseOut(obj) {
game.onMouseOut(obj);
}
</script>
</body>
</html>
```

PR 전에 코드 분석 자동화

버튼 하나로 저장소에서 변경부부만 골라내서 코드를 분석합니다.
리팩토링 대상이나 개선 사항들을 찾아서 보고서 형식으로 만들어 줍니다.

현재와 이전 커밋 코드의 변경 내용을 분석하기



메인 브랜치부터 현재까지 변경 내용을 분석하기

Code Sage

Question

Git diff (main) - 2024-08-09 16:41:24.885605
/Users/ryu/Library/Mobile Documents/com-apple-CloudDocs/Works/CodeSage

Answer

File: CodeSage-Multi.py

1. 리팩토링 대상 및 잠재적 오류 발생 영역:

- 복잡하거나 중복된 로직:
 - `analyze_changes` 함수 내에서 폴더 선택 및 검증 로직이 추가되었습니다. 이 로직은 함수의 다른 부분과 중복될 가능성이 있습니다. (라인 4-9)
 - `question` 문자열이 길고 복잡합니다. 이를 별도의 함수나 상수로 분리하는 것이 좋습니다. (라인 10-30)
- 불명확한 네이밍:
 - `folders` 변수는 `selected_folders` 로 변경하여 더 명확하게 할 수 있습니다. (라인 4)
 - `folder` 변수는 `selected_folder` 로 변경하여 더 명확하게 할 수 있습니다. (라인 9)
- 예외 처리 부족:
 - 폴더 선택이 잘못된 경우에 대한 예외 처리가 있지만, 다른 부분에서는 예외 처리가 부족합니다. 예를 들어, `get_changed_files_in_diff` 함수 호출 시 예외 처리가 필요합니다. (라인 32)
- 잠재적 버그 또는 성능 문제:
 - `get_relevant_documents` 함수 호출 시 폴더를 리스트로 전달하는 부분에서 성능 문제가 발생할 수 있습니다. (라인 41)

2. 클린 코드 원칙 적용 영역:

- 단일 책임 원칙 위반:
 - `analyze_changes` 함수는 여러 가지 책임을 가지고 있습니다. 폴더 선택, 질문 생성, 파일 변경 사항 처리 등 여러 기능을 분리하는 것이 좋습니다. (라인 4-48)
- 함수/메서드 길이 및 복잡성:
 - `analyze_changes` 함수가 너무 길고 복잡합니다. 이를 여러 개의 작은 함수로 분리하는 것이 좋습니다. (라인 4-48)
- 주석의 필요성 또는 과도한 주석:
 - 주석이 거의 없으며, 코드의 의도를 명확히 하기 위해 주석을 추가하는 것이 좋습니다. 특히 복잡한 로직이 있는 부분에 주석을 추가하는 것이 좋습니다. (라인 4-48)
- 변수 및 함수 이름의 명확성:
 - `folders`, `folder`, `file_names` 등의 변수 이름이 더 명확하게 변경될 필요가 있습니다. (라인 4, 9, 32)

3. 기타 코드 개선 영역:

- 디자인 패턴 적용 가능성:
 - 현재 코드에서는 특정 디자인 패턴이 적용되지 않았습니다. 예를 들어, 전략 패턴을 사용하여 분석 유형에 따라 다른 분석 방법을 적용할 수 있습니다. (라인 4-48)
- 테스트 가능성 향상 필요:
 - 함수가 너무 복잡하여 단위 테스트가 어렵습니다. 작은 함수로 분리하여 테스트 가능성을 높이는 것이 좋습니다. (라인 4-48)
- 코드 구조 및 아키텍처 개선:
 - 현재 코드 구조는 단일 파일에 모든 로직이 포함되어 있어 유지보수가 어렵습니다. 이를 여러 파일로 분리하여 모듈화하는 것이 좋습니다. (라인 4-48)

구체적인 개선 제안:

- 라인 4-9: 폴더 선택 및 검증 로직을 별도의 함수로 분리합니다.
- 라인 10-30: `question` 문자열을 별도의 상수나 함수로 분리합니다.
- 라인 32: `get_changed_files_in_diff` 함수 호출 시 예외 처리를 추가합니다.
- 라인 41: `get_relevant_documents` 함수 호출 시 폴더를 리스트로 전달하는 부분을 최적화합니다.
- 라인 4-48: `analyze_changes` 함수를 여러 개의 작은 함수로 분리하여 단일 책임 원칙을 준수합니다.

이와 같은 개선을 통해 코드의 가독성, 유지보수성, 성능을 향상시킬 수 있습니다.

기술문서 작성

회사 제품에 대한 동영상 대본이나 문서를 만들 때 활용합니다.

회사 내에 있는 기존 소스와 문서를 스스로 참고해서 결과물을 작성합니다.

The screenshot shows a web browser window with a question history sidebar on the left and a main content area. The question history sidebar lists various questions related to Vue.js, RealChart, and data visualization. The main content area displays a question and its answer, followed by two code snippets for RealChart implementation.

Question History

- 오류 메시지에 대한 시퀀스 다이어그램을 그려줘.
- 오류 메시지에 대한 job_flow를 그려줘.
- RoomView.vue 페이지가 로딩되었을 때 roomId 불이나라 public로 정보가 함께 전달할 수 있도록 수정해야할 코드를 보여줘.
- CreateRoom.vue, HomeView.vue 예도 roomApi.is가 아닌 room.is 스토어를 호출하도록 수정해줘.
- roomApi.is를 직접 호출하지 않고 room 스토어에서 roomApi.is를 호출하는 방식으로 수정하려고 한다.
- userApi.is를 직접 호출하지 않고 user 스토어에서 userApi.is를 호출하는 방식으로 수정하려고 한다.
- userApi.is를 직접 호출하지 않고 user 스토어에서 userApi.is를 호출하는 방식으로 수정하려고 한다.
- HomeView.vue에서 관리시작 버튼을 클릭하면 RoomView.vue로 이동하도록 할 예정이다.
- HomeView.vue에서 관리시작 버튼을 클릭하면 관리실 화면으로 이동하도록 하려고 한다.
- 프랜차이즈의 status code를 취급하는 모든 코드를 보여줘.
- Status Code에 관한 설정을 모두 보여줘.
- IntKeyLinkedMap을 사용하는 코드를 중심으로 분석해서 굳이 linked list를 사용해야 하는 이유를 찾아줘.
- TPS를 계산하고 전송하는 코드들을 모두 찾아줘.
- YouTube scripts for RealChart - BoxPlot
- 말을 통해서 듣게 처리하거나 안 이미 사용된 말에 있는 참조들은 더 이상 필요 없을 때 어떻게 처리 되나?
- Pack을 듣게 처리할 때 Pack을 구별하는 아이디어에 해당하는 것은 무엇인가?
- map 통해서 Pack을 모으고 통계처리하는 과정을 Job Flow 다이어그램으로 표현해줘.
- map 통해서 Pack을 모으고 통

Question

위의 두 가지 RealChart 소스를 보고, 기존 영상 스크립트(scripts.md)를 참고하여, 영상 스크립트를 작성해줘.

Answer

기존 영상 스크립트를 참고하여 두 가지 RealChart 소스를 설명하는 영상 스크립트를 작성해드리겠습니다.

cover

이번 영상에서는 BoxPlot 차트에 대해 알아보도록 하겠습니다.
단일 데이터 시리즈와 다중 데이터 시리즈를 사용한 두 가지 형태의 예제를 살펴보겠습니다.

slides_001

```
-<html>
-<head>
  <title>BoxPlot - Data Visualization</title>
  <link href="/realchart-style.css" rel="stylesheet">
  <script type="text/javascript" src="/realchart-lic.js"></script>
  <script type="text/javascript" src="/realchart-0.9.43.min.js"></script>
</head>
-<body>
  <div id="realchart" style="width: 800px; height: 500px;">
    <script>
      ...
    </script>
  </div>
</body>
</html>
```

이제 기본적인 코드 구조를 살펴보겠습니다.
대부분은 이전 영상들과 유사합니다.
head 부분에서는 css, 라이선스 그리고 차트 라이브러리를 불러오고 있습니다.
body에는 차트가 표시되는 div 태그 선언이 있고요, 자바스크립트로 차트를 설정하고 꾸미는 영역이 있습니다.

slides_002

이제 첫 번째 예제의 차트 설정 부분을 살펴보겠습니다.
(화상표) 차트의 제목은 'BoxPlot'로 설정했습니다.
(박스표시) x축에는 '스리엠', '아디다스', '디즈니', 'Amazon', '이마트'의 카테고리를 지정했습니다.

```
const config = {
  options: {},
  title: 'BoxPlot',
  xAxis: {
    categories: ['스리엠', '아디다스', '디즈니', 'Amazon', '이마트']
  },
  yAxis: {},
  series: {
    type: 'boxplot',
    pointLabel: true,
    data: [
      [560, 651, 748, 895, 965],
      [533, 753, 939, 900, 1000],
      [514, 662, 817, 870, 918],
      [624, 882, 816, 871, 950],
      [634, 736, 884, 882, 918]
    ]
  }
};
```

오픈소스 인공지능 활용

오픈소스로 공개된 인공지능을 활용하는 몇 가지 방법에 대해서 알아봅니다.

Llama 3.1 405B와 같은 모델을 사용한다면 좋겠지만 일반적인 개발자가 운영할 수 있는 환경은 아니기 때문에 개발자 PC에서 충분히 사용할 수 있는 Gemma 2 9B를 활용하고 있습니다.

Gemma 2를 선택한 이유는 다른 모델에 비해서 한국어 지원 및 균형이 잘 맞아서입니다.

하지만 9B 모델을 메인으로 사용하기에는 답변 품질이 떨어지기 때문에 메인은 유료인 Open AI나 Claude를 이용하고, 부수적인 작업을 Gemma 2에서 진행하도록 구성하였습니다.

반복적으로 발생하는 부수적인 작업을 처리하는 과정에서 비용을 절감할 수 있었습니다.

임베딩 모델의 한글 지원 이슈

00P에 관한 강의를 작성해줘.

강의 제작을 위해서 소재나 메모를 모아둔 폴더에서 위와 같은 질문을 했을 때, 한글과 영문으로 진행된 결과에 대한 스크린 샷입니다.

한글된 자료를 사용했을 때는 전혀 상관없는 참고 자료들이 더 많이 사용되는 것을 알 수 있습니다.

자료를 영어로 번역하고 영어로 질문 했을 때

```
python3 CodeSage.py
* Debug mode: on
2024-08-13 11:48:54,745 - INFO - WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:8080
* Running on http://10.160.136.75:8080
2024-08-13 11:48:54,745 - INFO - Press CTRL+C to quit
2024-08-13 11:48:54,745 - INFO - * Restarting with stat
2024-08-13 11:48:56,470 - INFO - Settings loaded.
2024-08-13 11:48:56,470 - INFO - API URL: https://api.openai.com/v1/embeddings
2024-08-13 11:48:56,470 - INFO - EMBEDDINGS_MODEL: text-embedding-3-large
2024-08-13 11:48:56,470 - INFO - CHAT_MODEL: gpt-4o
2024-08-13 11:48:56,478 - WARNING - * Debugger is active!
2024-08-13 11:48:56,488 - INFO - * Debugger PIN: 133-712-530
2024-08-13 11:49:03,908 - DEBUG - 질문 임베딩 생성 시작
2024-08-13 11:49:03,908 - INFO - Non-English character detected: '에' (Unicode: U+CS08) at position 3 in text: '00P에 관한 강의를 작성해줘.'
2024-08-13 11:49:03,908 - DEBUG - Ollama API 호출 시작
2024-08-13 11:49:03,916 - DEBUG - Starting new HTTP connection (1): localhost:11434
2024-08-13 11:49:08,400 - DEBUG - http://localhost:11434 "POST /api/generate HTTP/1.1" 200 974
Lines 1-1 (Translated):
Create a lecture about OOP.
Original:
00P에 관한 강의를 작성해줘.
Line 1 (Original): Create a lecture about OOP.
2024-08-13 11:49:08,407 - DEBUG - Starting new HTTPS connection (1): api.openai.com:443
2024-08-13 11:49:09,044 - DEBUG - https://api.openai.com:443 "POST /v1/embeddings HTTP/1.1" 200 None
2024-08-13 11:49:09,174 - DEBUG - 질문 임베딩 생성 완료
2024-08-13 11:49:09,370 - INFO - Retrieved selected folders: ['/Users/ryu/Library/Mobile Documents/com-apple-CloudDocs/Works/CodeSage/Embeddings/articles/programming-eng']
2024-08-13 11:49:09,370 - DEBUG - 임베딩 파일 목록: /Users/ryu/Library/Mobile Documents/com-apple-CloudDocs/Works/CodeSage/Embeddings/articles/programming-eng/embeddings.jsonl
2024-08-13 11:49:09,431 - DEBUG - /Users/ryu/Library/Mobile Documents/com-apple-CloudDocs/Works/CodeSage/Embeddings/articles/programming-eng에서 유사도로 선택된 파일:
2024-08-13 11:49:09,431 - DEBUG - README.md: 10
2024-08-13 11:49:09,431 - DEBUG - package.json: 10
2024-08-13 11:49:09,431 - DEBUG - src/router/index.js: 10
2024-08-13 11:49:09,431 - DEBUG - project-study-for-beginners/03.자바스크립트 입문서 - 요약판 .md: 0.38245798143941157
2024-08-13 11:49:09,432 - DEBUG - 가장 쉽게 설명한 개방 패러다임 /README.md: 0.3703792150090823
2024-08-13 11:49:09,432 - DEBUG - SOLID 원칙 .md: 0.33056038831232826
2024-08-13 11:49:09,432 - DEBUG - 001/README.md: 0.3198143095448819
2024-08-13 11:49:09,432 - DEBUG - vue-quick-start/001/README.md: 0.3025047215732025
2024-08-13 11:49:09,432 - WARNING - File README.md not found in embeddings. Skipping.
2024-08-13 11:49:09,432 - WARNING - File package.json not found in embeddings. Skipping.
2024-08-13 11:49:09,432 - WARNING - File src/router/index.js not found in embeddings. Skipping.
2024-08-13 11:49:09,441 - DEBUG - 프롬프트 생성 정보:
2024-08-13 11:49:09,441 - DEBUG - 출도문 수: 18529
2024-08-13 11:49:09,442 - DEBUG - 사용된 파일: ['project-study-for-beginners/03.자바스크립트 입문서 - 요약판 .md', '가장 쉽게 설명한 개방 패러다임 /README.md', 'SOLID 원칙 .md', '001/README.md', 'vue-quick-start/001/README.md']
2024-08-13 11:49:09,444 - DEBUG - OpenAI API 호출 시작
2024-08-13 11:49:09,445 - DEBUG - Starting new HTTPS connection (1): api.openai.com:443
```

한글 자료 원본과 한글로 질문 했 때

```
python3 CodeSage.py
* Running on http://10.160.136.75:8080
2024-08-13 11:46:11,795 - INFO - Press CTRL+C to quit
2024-08-13 11:46:11,796 - INFO - * Restarting with stat
2024-08-13 11:46:13,668 - INFO - Settings loaded.
2024-08-13 11:46:13,668 - INFO - API URL: https://api.openai.com/v1/embeddings
2024-08-13 11:46:13,668 - INFO - EMBEDDINGS_MODEL: text-embedding-3-large
2024-08-13 11:46:13,668 - INFO - CHAT_MODEL: gpt-4o
2024-08-13 11:46:13,676 - WARNING - * Debugger is active!
2024-08-13 11:46:13,688 - INFO - * Debugger PIN: 133-712-530
2024-08-13 11:46:16,461 - DEBUG - 질문 임베딩 생성 시작
2024-08-13 11:46:16,471 - DEBUG - Starting new HTTPS connection (1): api.openai.com:443
2024-08-13 11:46:16,852 - DEBUG - https://api.openai.com:443 "POST /v1/embeddings HTTP/1.1" 200 None
2024-08-13 11:46:17,171 - DEBUG - 질문 임베딩 생성 완료
2024-08-13 11:46:17,367 - INFO - Retrieved selected folders: ['/Users/ryu/Library/Mobile Documents/com-apple-CloudDocs/Works/CodeSage/Embeddings/articles/programming-kor/']
2024-08-13 11:46:17,367 - DEBUG - 임베딩 파일 로드: /Users/ryu/Library/Mobile Documents/com-apple-CloudDocs/Works/CodeSage/Embeddings/articles/programming-kor/embeddings.jsonl
2024-08-13 11:46:17,411 - DEBUG - /Users/ryu/Library/Mobile Documents/com-apple-CloudDocs/Works/CodeSage/Embeddings/articles/programming-kor/에서 유사도로 선택된 파일:
2024-08-13 11:46:17,411 - DEBUG - README.md: 10
2024-08-13 11:46:17,411 - DEBUG - package.json: 10
2024-08-13 11:46:17,411 - DEBUG - src/router/index.js: 10
2024-08-13 11:46:17,411 - DEBUG - project-study-for-beginners/03.자바스크립트 입문서 - 요약편.md: 0.41886501053652514
2024-08-13 11:46:17,411 - DEBUG - Method R/README.md: 0.4015327484624077
2024-08-13 11:46:17,411 - DEBUG - SOLID 원칙.md: 0.38812780657965695
2024-08-13 11:46:17,411 - DEBUG - 가장 쉽게 설명한 개방패러의 원칙/README.md: 0.3825082411758781
2024-08-13 11:46:17,411 - DEBUG - 001/README.md: 0.36188061397385504
2024-08-13 11:46:17,411 - DEBUG - vue-quick-start/001/README.md: 0.3498053431858791
2024-08-13 11:46:17,411 - DEBUG - 003/README.md: 0.3373309725386513
2024-08-13 11:46:17,411 - DEBUG - 가장 특화된 소프트웨어 엔지니어에게 배운 10가지 교훈.md: 0.3346902667054954
2024-08-13 11:46:17,411 - DEBUG - project-docs/README.md: 0.32888639504107453
2024-08-13 11:46:17,411 - DEBUG - project-rules/README.md: 0.3273162259540164
2024-08-13 11:46:17,411 - DEBUG - .NET profiling API로 함수 실행시간 구하기/README.md: 0.3232690922003141
2024-08-13 11:46:17,411 - DEBUG - 기술 면접 시나리오.md: 0.32303096650738885
2024-08-13 11:46:17,411 - DEBUG - project-rules/vue/001/README.md: 0.31732495210506445
2024-08-13 11:46:17,411 - DEBUG - chromium/01/README.md: 0.31552211695209065
2024-08-13 11:46:17,411 - DEBUG - 서버, 네트워크, 데이터베이스에 대한 설명.md: 0.3102729418736483
2024-08-13 11:46:17,411 - DEBUG - project-docs/002/README.md: 0.30751276792283017
2024-08-13 11:46:17,411 - WARNING - File README.002 not found in embeddings. Skipping.
2024-08-13 11:46:17,411 - WARNING - File package.json not found in embeddings. Skipping.
2024-08-13 11:46:17,411 - WARNING - File src/router/index.js not found in embeddings. Skipping.
2024-08-13 11:46:17,430 - DEBUG - 프록시드 생성 완료:
2024-08-13 11:46:17,430 - DEBUG - 총 프록시 수: 17692
2024-08-13 11:46:17,431 - DEBUG - 사용된 파일: ['project-study-for-beginners/03.자바스크립트 입문서 - 요약편.md', 'Method R/README.md', 'SOLID 원칙.md', '가장 쉽게 설명한 개방 패러의 원칙/README.md', '001/README.md', 'vue-quick-start/001/README.md', '003/README.md', '가장 특화된 소프트웨어 엔지니어에게 배운 10가지 교훈.md', 'project-docs/README.md', 'project-rules/README.md', '.NET profiling API로 함수 실행시간 구하기/README.md', '기술 면접 시나리오.md', 'project-rules/vue/001/README.md', 'chromium/01/README.md', '서버, 네 트워크, 데이터베이스에 대한 설명.md', 'project-docs/002/README.md']
2024-08-13 11:46:17,433 - DEBUG - OpenAI API 호출 시작
2024-08-13 11:46:17,434 - DEBUG - Starting new HTTPS connection (1): api.openai.com:443
```

입력 토큰 부족 이슈

오픈소스 인공지능을 이용하여 참고자료를 1차 요약하여 전달하는 방식으로 참고자료가 방대한 경우에 대한 지원이 가능해집니다.