



G Team(ISP) Weekly Progress 10.13

MyCampus

박지민, 옥시원, 최재영, 하슬비

Index

1. 지난주 피드백
2. 화면 표시 방법
3. 백엔드 진행 상황



1. 지난주 피드백



Feedback

- 모든 데이터를 불러오지 않고 최신 데이터만 불러오는 방법 고려

1. 실시간 업데이트는 불가능 : 해당 API를 간접적으로 사용하기 때문에 강의를 출석 완료하였을 때, 해당 이벤트를 자동으로 알 수 없음. 직접 업데이트 해주어야 함.

2. 1주일에 한번 자동으로 업데이트 시, 아이캠퍼스에서 사용하는 API를 사용하여 데이터를 불러오면 전체 데이터를 불러오게 됨

→ 이때 이미 storage에 저장된 데이터와 새로 불러온 데이터를 비교하여 새로운 데이터만 업데이트 하는 시간과 전체 데이터를 삭제 하고 전체 데이터를 저장하는 방식의 시간 차이는 거의 없을 거라 생각

1 Feedback 남은 용량 표시

- 사용자가 용량을 확인해서 관리 할 수 있도록 남은 용량 표시
 1. chrome storage의 최대 용량 = 5MB
 2. Chrome.storage.sync.getBytesInUse 를 사용해서 현재까지 사용된 용량 측정→ 전체 용량(5MB)에서 제하고 알려줌

```
function logBytes(bytes) {  
    console.log(bytes);  
}  
  
// gets the number of bytes used in sync storage area  
chrome.storage.sync.getBytesInUse(['settings'], logBytes);  
  
// gets the number of bytes used in the local storage area  
chrome.storage.local.getBytesInUse(['settings'], logBytes);
```



Feedback

모바일 고려

- chrome extension 모바일 고려
 - 고려하지 않기로 결정. 모바일에서는 chrome이 아니라 learning X 어플로 아이캠퍼스에 접속하기 때문에 확장 프로그램의 의미가 없다고 생각



Feedback 테스트 방법

- 배포하지 않고 테스트 할 수 있는 방법
 1. 팀 내에서 test
 2. 베타 버전으로 올려서 승인 요청 (문제가 없다면 24시간 이내에 승인)
 3. 베타 버전으로 배포된 extension을 이용해서 수강생분들을 대상으로 test (extension 사용 메뉴얼을 함께 제공할 예정)
 4. 피드백을 바탕으로 수정 후 완성본 배포

2. 화면 표시 방법



화면 표시

1

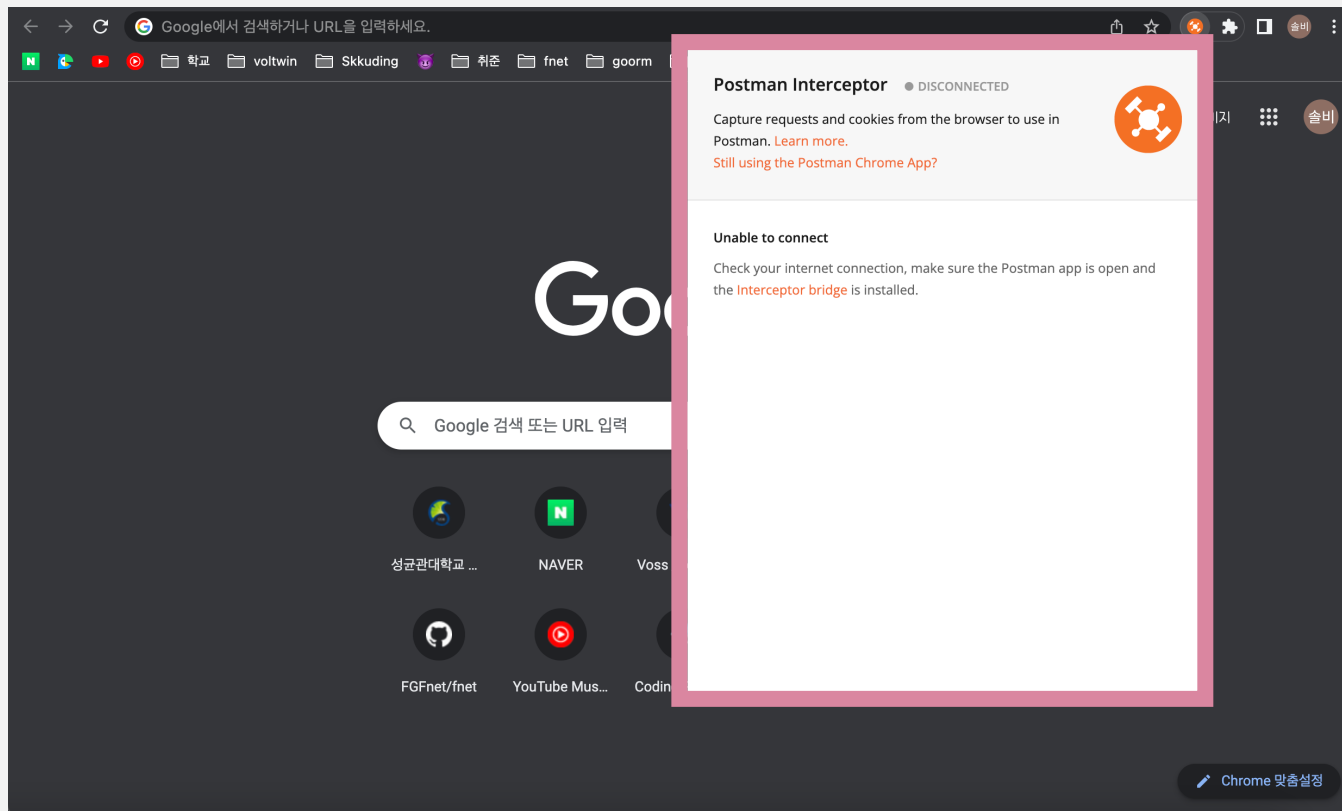
화면표시

Trial 1: popup.tsx 이용

- popup.tsx의 라우팅 요소로 렌더링 될 페이지 결정
- popup.tsx는 확장 프로그램을 눌렀을 때 나타나는 팝업으로, 크기 조정은 가능하나 위치를 조정할 수 없기 때문에 아이캠퍼스 페이지를 덮으려는 우리 프로젝트에는 적절하지 않음



- content.ts를 이용해서 아이캠퍼스에 우리가 만든 페이지를 올려야함!



2

화면표시

Trial2: iframe 이용

- HTML 태그 중 현재 HTML 문서에 다른 문서를 포함시킬 때 사용할 수 있는 태그인 <iframe>을 이용

```
fetch(chrome.runtime.getURL('/dashboard.tsx'))
  .then(response => {
    const iframe = document.createElement('iframe')
    iframe.src = response.url
    main.append(iframe)
  })
  .catch(err => {
    // handle error
  });
```

2

화면표시

Trial2: iframe 이용

- DOM이 react 코드를 해석하지 못해서 코드가 그대로 화면에 표시 된다
- react는 브라우저가 이해하지 못하는 jsx / tsx 포맷을 이용

```
import React from "react";
class dashboardScreen extends React.Component {
  render() {
    return (
      <div>
        <h1>Hello world - My first
Extension</h1>
      </div>
    )
  }
}
```

- dashboard.tsx 코드

Trial3: React DOM 이용

- React DOM

React에서 사용하는 가상의 DOM

- ReactDOM.render()

react element를 DOM 상에서 원하는 위치에 넣을 수 있도록 해주는 함수

```
import React from 'react';  
import ReactDOM from 'react-dom';  
import {dashboardScreen} from './pages';
```

```
const dash = document.createElement('div')  
dash.id = 'dashboard'  
main.appendChild(dash)  
ReactDOM.render(React.createElement(dashboardScreen), dash);
```

The screenshot displays a web application interface for 'MyCampus'. On the left is a dark blue sidebar with icons and labels for navigation: a logo, '계정' (Account), '대시보드' (Dashboard), '과목' (Courses), '그룹' (Groups), '캘린더' (Calendar), '메시지함' (Mailbox), '이용안내' (Usage Guide), '시간표' (Timetable), and a back arrow. The main content area has a light blue header with the text 'Hello world - This is MyCampus!' and a 'Press!' button. Below the header are four colored cards (purple, red, orange, green) each representing a course or announcement. The purple card is titled 'Introduction to Computer Architec...' and the red card is titled '2022 학생을 위한 폭력예방교육(법...'. The orange card is titled '안전교육_SFE7222_I1(육기봉)' and the green card is titled '캡스톤설계프로젝트_SWE3028_41(...)'. To the right of the cards is a '할 일' (To Do) list with six items, each preceded by a bell icon and followed by a close button (X). The items include '9/29 (목) Virtual Meeting', 'Presentation on Technical ...', '[PA#1] Due extended to O...', '2022_Web Programming_...', 'Chapter 05-Advanced CSS...', and 'PLC-Slide05-1'.

1998

계정

대시보드

과목

그룹

캘린더

메시지함

이용안내

시간표

←

Hello world - This is MyCampus!

Press!

Introduction to Computer Architec...

Introduction to Computer Arc...

2022년 1학기

2022 학생을 위한 폭력예방교육(법...

2022 학생을 위한 폭력예방교육(...

2022 비정규

안전교육_SFE7222_I1(육기봉)

안전교육_SFE7222_I1(육기봉)

2022년 2학기

캡스톤설계프로젝트_SWE3028_41(...

캡스톤설계프로젝트_SWE3028_...

2022년 2학기

할 일

9/29 (목) Virtual Meeting

캡스톤설계프로젝트_SWE3028_41(구형준)

9월 28일 오전 9:31 |

Presentation on Technical ...

캡스톤설계프로젝트_SWE3028_41(구형준)

9월 29일 오후 3:00 |

[PA#1] Due extended to O...

Programming Languages_SWE3006_41(한환수)

9월 29일 오후 4:52 |

2022_Web Programming_...

Web Programming Lab_SWE3048_44(황성재)

1 points |

10월 2일 오후 11:59 |

Chapter 05-Advanced CSS...

Web Programming Lab_SWE3048_44(황성재)

10월 2일 오후 11:59 |

PLC-Slide05-1

Programming Languages_SWE3006_41(한환수)

1 points |

10월 2일 오후 11:59 |

- 그렇다면 라우팅은?

window.location.href로 현재 위치한 페이지의 url을 파악하여 렌더링 되는 페이지가 다르게 설정 할 예정

3. 백엔드 진행 상황



백엔드 진행 상황

1

백엔드 진행상황

Base file 수정 시도

- 기존의 base file은 React 라이브러리와 webpack 번들러 사용
1. React → Next.js 교체 시도: React 프레임워크인 Next.js를 사용하면 편리하지 않을까 생각
 - Next.js에서 제공하는 기능들은 크롬 확장프로그램에 적합하지 않음
 - Search Engine Optimization: 확장프로그램은 검색될 필요가 없음
 - Server-Side Rendering: 백엔드 서버를 운영하지 않음, 반응성이 나쁨
 - 디렉토리 기반 라우팅: 실제 URL을 이동할 필요가 없음
 2. Webpack → SWC 교체 시도: SWC가 성능이 매우 뛰어나다는 소식을 듣고 개발 시간을 줄이기 위해 교체를 시도했지만, 프로젝트의 크기가 작아 webpack으로도 충분히 사용 가능하고, 번들러에 대한 이해 부족으로 번들러 교체의 오버헤드가 너무 크다고 판단되어 포기

- 기존에는 Axios를 사용하려고 계획했으나, JavaScript의 fetch를 사용하기로 변경
 - Axios는 Node.js에서 사용할 수 있는데, Node가 아닌 일반 브라우저(클라이언트) 상에서는 adapter가 없어 Axios를 사용할 수 없다.
 - Adapter를 따로 등록해 줄 수 있는 패키지가 있지만, 적용을 해봐도 작동을 하지 않아 포기
 - Fetch를 이용하여 데이터를 받아오는 것까지 성공

1. Readable Stream

- 학교에서 제공하는 API들은 응답을 완전한 JSON 형태로 제공하지 않고 Readable Stream이라는 형식을 사용함. Readable Stream을 읽는 방법을 찾아야 함.

2. 스토리지에 읽고 쓰기

- API에서 데이터를 받아오는 것까지 성공하면, 이 데이터를 원하는 형식으로 스토리지에 저장하고 읽어오는 기능을 만들어야 함.



감사합니다
