



G Team(ISP) Project Proposal

---

**MyCampus**

박지민, 옥시원, 최재영, 하슬비

---

# Index

---

1. Logo
2. 데이터 저장 방식
3. 개발 환경 세팅
4. 데이터 포맷 및 API
5. 향후 계획



---

## 1. Logo

---



# MyCampus 의 새로운 로고

# 1 로고



---

## 2. 저장공간

---



# Storage API

1

chrome.storage.local

## chrome.storage.local에 데이터 저장하기

학교 API 를 통해 받아 오는 정보  
(강의컨텐츠, 수업자료실 등)



chrome.storage.local

사용자의 입력 정보  
(메모, 강의 체크리스트 등)



chrome.storage.sync

## 2

chrome.storage.local

## chrome.storage.local에 데이터 저장하기

```
chrome.storage.local.set({key: value}, function() {  
  console.log('Value is set to ' + value);  
});
```

1. axios 에서 json 데이터를 받음

- axios
  - 아이캠퍼스로부터 데이터를 받아 오기 위해 사용
  - http 요청과 응답을 json 형태로 자동 변경

2. 해당 데이터를 chrome.storage.local 에 set

3

chrome.storage.local

## chrome.storage.local에 데이터 불러오기

```
chrome.storage.local.get(['key'], function(result) {  
    console.log('Value currently is ' + result.key);  
});
```

빠른 로딩을 위해 정보 업데이트 기간을 1주일로 설정

정보의 업데이트 기간(1주일) 이 만족되지 않았을 때 사용



## 4

## chrome.storage.local chrome.storage.local 예시코드

```
import React, {useState, useEffect} from 'react';  
import axios from 'axios';  
  
function Example2() {  
  const set_update = false;  
  const [course, setCourse]=useState();  
  
  useEffect(()=>{  
    axios.get("https://canvas.skku.edu/api/v1/users/self/favorites/courses")  
      .then((result)=>{  
        chrome.storage.local.set({"course": result});  
        setCourse(result);  
      })  
      .catch((err)=>{console.log(err)})  
  
    set_update = false;  
  },[set_update])  
  
  return {
```

Axios를 이용하여  
iCampus 로부터 데이터를 받아  
chrome.storage.local 에 저장하고  
현재 course state 역시 업데이트

## 5

chrome.storage.sync

## chrome.storage.sync에 데이터 저장 및 불러오기

```
chrome.storage.sync.set({key: value}, function() {  
  console.log('Value is set to ' + value);  
});  
  
chrome.storage.sync.get(['key'], function(result) {  
  console.log('Value currently is ' + result.key);  
});
```

사용자가 다른 컴퓨터를 사용하더라도 동일한 정보가 유지될 수 있음.  
데이터가 cloud 에 저장되는 것이기 때문에 데이터 손실의 위험이 없음.

## 6

chrome.storage.sync

## chrome.storage.sync 예시코드

```
import React, {useState, useEffect} from 'react';

function Example() {

  const memo={
    title: '조교님 메일',
    content: '123@naver.com'
  }

  const [memoList, setMemoList]=useState([]);

  useEffect(()=>{
    chrome.Storage.local.get("memoList", ({memoList})=>{
      setMemoList([...memoList, memo]);
    })
  },[])

  return {
```

chrome.storage.sync 에 저장된  
memoList 데이터를 가져와  
현재 memoList 를 업데이트

---

### 3. 개발환경세팅

---



## 환경 세팅 및 디렉토리

## 1

개발환경 세팅

## Base File 세팅

The screenshot shows the GitHub repository page for 'chrome-extension-react-typescript-boilerplate' by user 'martellaj'. The repository is public and has 458 stars, 88 forks, and 4 watchers. The 'Code' tab is selected, showing the file structure and commit history. The file list includes 'dist', 'src', '.gitignore', 'LICENSE', 'README.md', 'package-lock.json', 'package.json', and 'tsconfig.json'. The commit history shows that 'dist' was added 16 months ago, 'src' was added 3 years ago, and '.gitignore' was added 5 years ago. The repository description states: 'A boilerplate project to quickly build a Chrome extension using TypeScript and React (built using webpack)'. The repository tags include 'react', 'chrome-extension', 'boilerplate', 'typescript', 'webpack', 'seed', 'boilerplate-front-end', and 'seed-project'.

martellaj / chrome-extension-react-typescript-boilerplate Public

Watch 4 Fork 88 Star 458

Code Issues 1 Pull requests 11 Actions Projects Security Insights

master 12 branches 0 tags Go to file Add file Code

About

A boilerplate project to quickly build a Chrome extension using TypeScript and React (built using webpack).

react chrome-extension boilerplate

typescript webpack seed

boilerplate-front-end seed-project

Readme MIT license 458 stars 4 watching 88 forks

File	Commit Message	Time Ago
dist	Added charset to popup.html	16 months ago
src	Class => function	3 years ago
.gitignore	Initial commit	5 years ago
LICENSE	Adds license	2 years ago
README.md	Adds dev/prod webpack configs	4 years ago
package-lock.json	Updates package-lock.json	3 years ago
package.json	Bumps version	3 years ago
tsconfig.json	Class => function	3 years ago

- 웹페이지를 개발할 때와 다르게 더 많은 파일들이 필요
- 크롬 익스텐션을 실행하기 위해 빌드 방법을 수정하는 등 복잡한 작업 필요



chrome-extension-react-  
typescript-boilerplate  
레포지토리를 클론해서 base file 생성

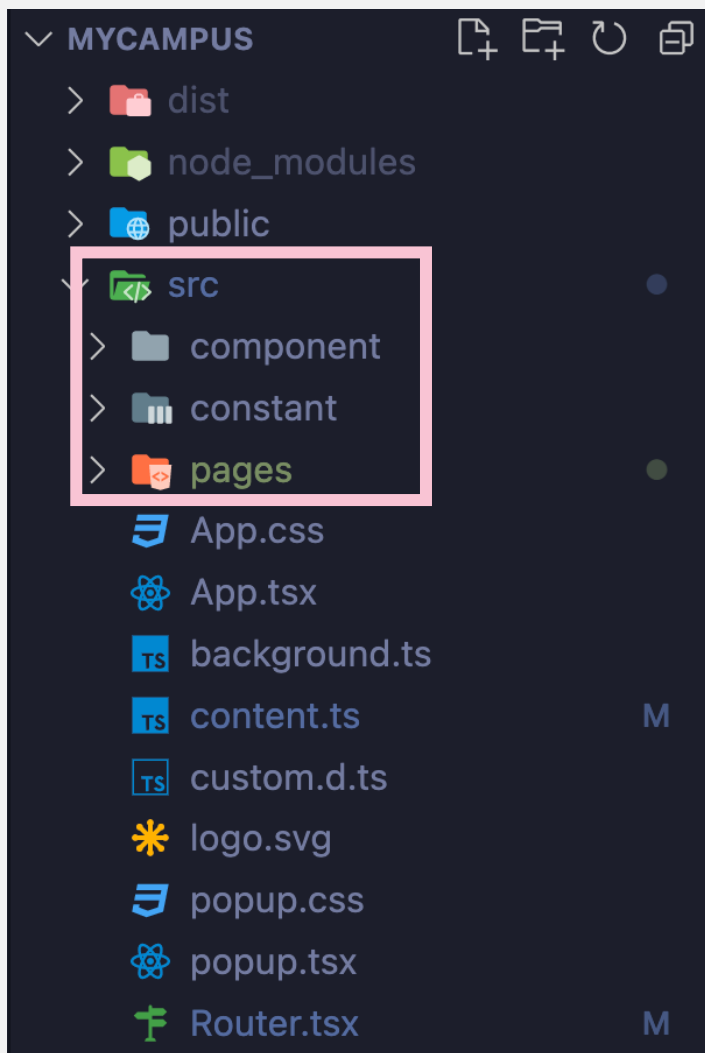
- chrome extension 설정을 할 수 있는 파일을 우리 extension 에 맞게 수정
  - 이름, 설명, 아이콘 설정
  - 권한 설정 등
- 버전 업데이트
  - 2021년 1월부터 Google 에서 Manifest Version3 (MV3) 을 도입, v2 는 단계적으로 폐지
  - 하지만 base file 세팅을 위해 사용한 레포지토리는 v2 을 이용
  - 버전을 업데이트 하면서 그에 의존성을 가지고 있던 여러 라이브러리 업데이트 (ts-loader, @types/chrome, ts-node 등)

## React-router 도입

```
<Routes>
  <Route path="" element={<App/>}/>
  <Route path="/dashboard" element={<Screen.dashboardScreen/>} />
  <Route path="/subject/:id" element={<Screen.subjectScreen/>} />
  <Route path="/subject/:id/material" element={<Screen.materialScreen/>} />
  <Route path="/subject/:id/assignment" element={<Screen.assignmentScreen/>} />
  <Route path="/subject/:id/notice" element={<Screen.noticeScreen/>} />
  <Route path="/subject/:id/lecture" element={<Screen.lectureScreen/>} />
</Routes>
```

우리가 제작하고자 하는 extension 은 여러 페이지가 필요하기 때문에  
라우팅을 위한 라이브러리인 react-router 도입  
생성한 페이지들을 원하는 주소로 매핑하기 위한 라우터 파일 세팅

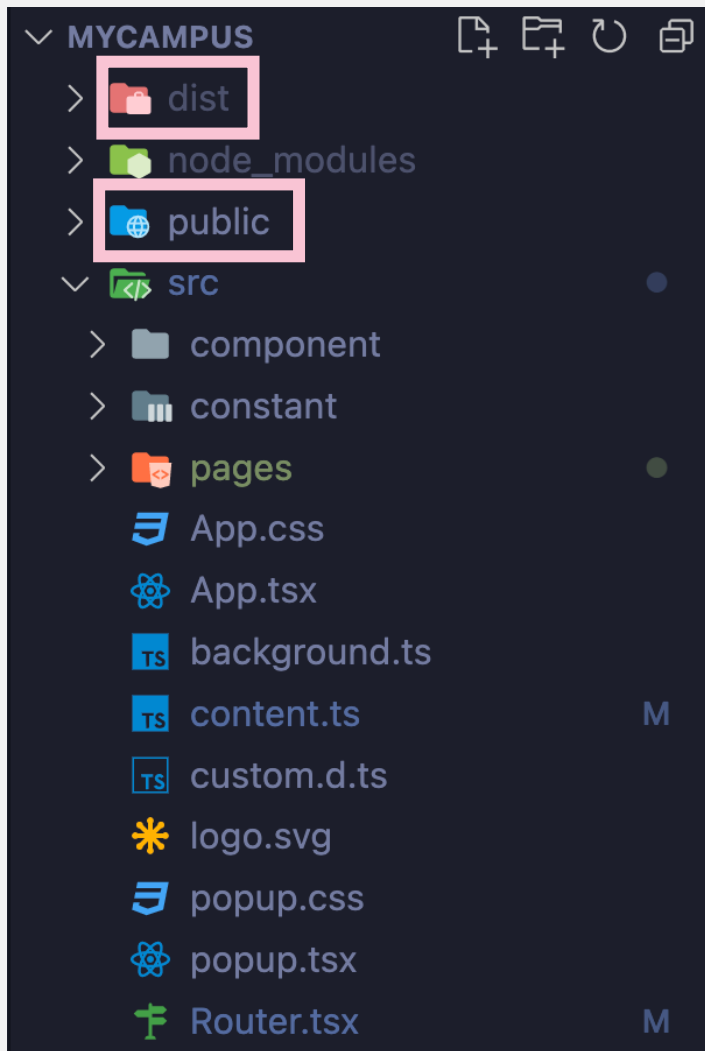
## 4

디렉토리 설정  
전체 파일 구조

- **src**
  - 개발하면서 작성하는 모든 소스코드 파일들을 모아 놓은 폴더
- **component**
  - 프로젝트에서 여러 번 사용되는 component 들을 모아 놓은 폴더
- **constant**
  - 프로젝트에서 사용되는 상수를 선언해 놓은 폴더 (ex. color 등)
- **pages**
  - 사용자가 직접 이용하는 페이지를 생성하기 위한 코드파일을 모아 놓은 폴더
    - 전체 대시보드, 과목 대시보드
    - 과목별 페이지 (공지사항, 자료실, 과제, 강의)



## 5

디렉토리 설정  
전체 파일 구조

- dist
  - 파일을 빌드하면 extension 을 실행할 수 있는 파일이 생성되는 폴더
  - 해당 폴더를 chrome extension 에 업로드하면 실제로 실행해 볼 수 있음
- public
  - 프로젝트 전반적으로 사용되는 파일들을 모아 놓은 폴더
  - 아이콘 이미지, manifest 파일
- 다른 파일들
  - chrome extension 기본 파일들

---

## 4. Data Format

---



# Data Format 및 API

## 1 Data Format 필요한 데이터

- 총 7가지의 데이터 및 필요한 용량

대시보드 (420B)

공지 (420B)

공지 댓글 (420B)

강의 콘텐츠 & 과제 및 평가 (2.7KB)

수업자료실 (16.7KB)

수업계획서 (400B)

## 2

Data Format  
대시보드

```
{
  → assetString: "course_18372",
  → courseCode: "데이터베이스개론_SWE3003_43(김응모)",
  → enrollmentType: "StudentEnrollment",
  → href: "/courses/18372",
  → id: "18372",
  → image: null,
  → isFavorited: true,
  → links: [{
    → → css_class: "assignments"
    → → hidden: null
    → → icon: "icon-assignment"
    → → label: "과제 및 평가"
    → → path: "/courses/18372/assignments"
  → }],
  → longName: "데이터베이스개론_SWE3003_43(김응모) - 데이터베이스개론_SWE3003_43(김응모)",
  → observee: null,
  → originalName: "데이터베이스개론_SWE3003_43(김응모)",
  → position: 0,
  → shortName: "데이터베이스개론_SWE3003_43(김응모)",
  → subtitle: "로 등록: 학생",
  → term: "2021년 1학기
}
```

API

: /dashboard/dashboard\_cards

## 3

Data Format  
공지

```
{
  attachments: [],
  author:{
    avatar_image_url: null,
    display_name: "한환수(HAN HWANSOO)",
    html_url: https://canvas.skku.edu/courses/34286/users/37785,
    id: "37785",
    pronouns: null
  },
  created_at: "2022-09-23T03:21:26Z",
  html_url: https://canvas.skku.edu/courses/34286/discussion\_topics/316166,
  id: "316166",
  last_reply_at: "2022-09-23T03:21:26Z",
  lock_explanation: "이 주제는 댓글로 마감되었습니다.",
  message: "공지사항 내용",
  permissions:{
    attach: false,
    delete: false,
    reply: false,
    update: false
  },
  pinned: false,
  position: 8,
  posted_at: "2022-09-23T03:21:26Z",
  published: true,
  read_state: "read",
  title: "[Next Week] No offline lectures ",
  unread_count: 0,
  url: https://canvas.skku.edu/courses/34286/discussion\_topics/316166,
  user_count: 69,
  user_name: "한환수(HAN HWANSOO)"
}
```

API

: /courses/:course\_id/discussion\_topics

## 4

Data Format  
공지 댓글

```
{
  entry_ratings: {},
  forced_entries: [],
  new_entries: [],
  participants: [{
    avatar_image_url: https://canvas.skku.edu/images/messages/avatar-50.png,
    course_id: "32379",
    display_name: "구형준(KOO HYUNG JOON)",
    html_url: https://canvas.skku.edu/courses/32379/users/82636,
    id: "82636",
    is_student: false,
    pronouns: null
  }],
  unread_entries: [],
  view: [{
    created_at: "2022-09-20T00:49:29Z"
    id: "941121"
    message: 댓글 내용
    parent_id: null
    rating_count: null
    rating_sum: null
    updated_at: "2022-09-20T00:49:29Z"
    user_id: "82636"
  }]
}
```

API

: /courses/:course\_id/discussion\_topics/:topic\_id/view

## 5

Data Format  
강의컨텐츠

```
{
  due_at: "2021-03-07T14:59:59Z",
  has_component: true,
  is_master_course_child_content: false,
  late_at: null,
  lock_at: null,
  position: 1,
  published: null,
  section_id: 267880,
  subsections: [
    {
      is_master_course_child_content: false
      position: 1
      subsection_id: 1442174
      title: "1차시",
      units: [{
        position: 2,
        title: "1페이지",
        unit_id: 1442175,
        components: [
          {
            component_id: 1499747
            position: 3
            published: null
            title: "1주 공지사항"
          }
        ]
      }]
    }
  ],
  title: "1주차"
  unlock_at: "2021-02-21T15:01:00Z"
}
```

API

: /courses/:course\_id/sections\_db

## 6

Data Format  
수업자료실

```
{
  commons_content: {
    content_id: "631ffe3a1c8d9",
    content_type: "pdf",
    file_name: "T2-ANTLR-C++-tutorial.pdf",
    progress_support: false,
    thumbnail_url: https://lcms.skku.edu/CommonsCore2/v2/contents/b9d20263-d1a9-4b61-93cf-96e3fc17067e.jpg,
    view_url: https://lcms.skku.edu/em/631ffe3a1c8d9
  },
  description: "2022F version",
  position: 2,
  published: false,
  resource_id: 1187203,
  title: "T2-ANTLR-C++-tutorial"
}
```

API

: /courses/:course\_id/resources\_db



---

## 5. Plan

---



**앞으로의 계획**



Plan

## 앞으로의 계획

- ~10/21(2주) 까지 페이지 퍼블리싱
- ~11/04(2주) 까지 데이터 받아온 후 연결
- ~ 11/11(1주) 배포
  - 검토 1주일 걸린다고 함
- ~ 배포 후 실사용 및 테스트 / 피드백 후 수정



---

감사합니다

---