**웹 시스템 설계 – 11주차 학습 보고서 4 (2020년 11월 18일까지 제출)**

|  |  |  |
| --- | --- | --- |
| 성명 : 유 상 정 | 학과 : 사이버보안학과 | 학번 : 201620641 |

※ 본 학습보고서를 작성함에 있어 다른 학생의 문서로부터 일부 또는 전체를 복사하였습니까?

예( ) 아니오(✓) (복사 하였다면 예에 체크하고 아니라면 아니오에 체크하시오)

**I. Objectives**

* XMLHttpRequest API를 활용하여 백엔드 프론트엔드 간 비동기 통신(AJAX)을 수행할 수 있다.

**II. Exercises**

* “11주차 학습보고서 3”에서 AJAX로 변경된 통신을 수행하는 프론트엔드 측의 코드와 백엔드 측의 코드를 복사 붙여 넣기 한 후, 간단히 설명하라.

|  |  |
| --- | --- |
| **AJAX를 수행하기 위한 프론트엔드 측 코드** | /\*ajax Trending Update \*/  function ajaxTrendingUpdate(id,trending){    //XMLHttpRequest 객체 생성    var xhr = new XMLHttpRequest();      //전송할 데이터    var sendData = {      id: id,      trending: !trending    }    //요청의 상태가 DONE 일 경우 프론트엔드 측에서 요청을 처리하는 방식을 정의    xhr.onload = function(){      if(xhr.status === 200 || xhr.status === 201){        //string 으로 받은 Json 타입의 데이터를 JSON 객체화        let response = JSON.parse(xhr.response);        // response JSON 타입 객체 내 people 데이터에 접근        //이후, 응답 값을 통해 HTML 문서 내 정의된 정보를 변경가능;          let star = document.querySelector(`#starPosition\_${id}`);        if (!trending)        {          star.childNodes[0].src = "http://localhost:3000/images/YesTrending\_star.png" ;          if(document.querySelector('#yellowStar'))            document.querySelector('#yellowStar').onclick = () => { ajaxTrendingUpdate(id,!trending)};          else            document.querySelector('#blackStar').onclick = () => { ajaxTrendingUpdate(id,!trending)};        }        else        {          star.childNodes[0].src = "http://localhost:3000/images/NoTrending\_star.png"  ;          if(document.querySelector('#yellowStar'))            document.querySelector('#yellowStar').onclick = () => {ajaxTrendingUpdate(id,!trending)};          else            document.querySelector('#blackStar').onclick = () => { ajaxTrendingUpdate(id,!trending)};        }      }      else{        alert("movie Ajax Trending Update fail!");      }    }      //생성된 XmlHttpRequest 객체의 초기화    xhr.open("POST", `/routes/movie/ajaxTrendingUpdate/`);    //요청의 헤더 설정(json 타입을 통신에 활용할 것이므로, 컨텐츠 타입을 json 으로 선언)    xhr.setRequestHeader("Content-Type", 'application/json');    //서버측으로 전달할 json 타입의 데이터를 string 화 한 데이터와 함께 요청을 전달.    xhr.send(JSON.stringify(sendData));  }  과제 수행 채점은 Chrome이므로, 따로 Request Object 체크를 수행하지 않았다. 그래서 바로 XMLHttpRequest 객체를 생성한 후, server에 전송할 데이터인, 현재 movie와 반대되는 Trending값과 id를 data 객체에 담는다. 그 후, 생성된 xmlHttpRequest 객체를 http method와 action을 지정해주고, header에서 content-type을 json으로 설정해준다. 그리고 json 타입의 데이터를 string화 한 데이터와 함께 요청을 send한다. 마지막으로, 보낸 요청의 상태가 DONE이 되면, server로부터 받은 status를 확인하고, 오류문을 띄우거나 정상적으로 수정된 trending에 해당하는 star이미지로 변경해준다. 또한, html을 변경하므로, 맨 처음에 yellowStar였으면 yellowStar의 onclick을 변경하고, blackStar였으면 blackStar의 onclick을 변경한다. |

|  |  |
| --- | --- |
| **AJAX를 수행하기 위한 백엔드 측 코드** | /\* ajax Trending Update \*/  router.post('/routes/movie/ajaxTrendingUpdate/', function(req, res, next){    let id = req.body.id;    let trending = req.body.trending;    Movie.findByIdAndUpdate(id,{      $set: {        trending: trending      }    })    .then(() => {      let data = {        success : true,      }      res.status(200).json(data); //json type으로 데이터 전송    })    .catch((err) => {      console.log(err)      let data = {        success : false      }      res.status(500).json(data);    });  });  Backend code는 front 로부터 전송받은 ajax 요청을 처리한다.  받은 request의 body를 확인해서 data를 저장하고, Movie model에서 id에 해당하는 movie를 찾은 후, trending값을 변경해준다. 그리고 성공여부를 response에 json 형식으로 담아서 전송한다. 이 때, http status code를 설정해서 보낸다. |