

[Design of Data Structure]

창의ICT공과대학 융합공학부
디지털이미징 2학년 정진혁

Implementation

Development

Language : Java Script

IDE : Visual Studio Code

GUI framework : Electron (Front-end : Chromium / Back-end : Node.js)

JS framework : Vue.js (MVVM Pattern)



Function

Component

방송프로그램 정보입력 : **RegisterProgram**

방송프로그램 정보삭제 : **DeleteProgram**

방송프로그램 순위보기 : **RatingProgram**

방송프로그램 편성하기 : **ScheduleProgram**

Function #1

RegisterProgram

: 방송프로그램의 정보를 입력 받아 저장한다.

```
tv_program: {  
  title: "  
  genre: "  
  rating: "  
  runtime: "  
}
```

```
saveFSData() {  
  /*  
   * json_container: 파일시스템에서 data.json 읽은 정보  
   * base_format: data.json에 공백일 경우 생성되는 기초 포맷 정보값  
   * program_array: JSON 파일 형식을 Parse한 값  
   * writeFileSync: 파일을 저장 가능하도록 도와주는 파일 시스템 메소드  
   * readFileSync: 파일을 읽도록 도와주는 파일시스템 메소드  
   */  
  const json_container = fs.readFileSync(path.join(__static, '/data.json'), 'utf8');  
  if(json_container === '') {  
    let base_format = {program: []}  
    fs.writeFileSync(path.join(__static, '/data.json'), JSON.stringify(base_format), 'utf8');  
    this.initData();  
    alert('data.json의 기초 포맷을 생성합니다. 다시 입력하세요.')  
  }  
  let program_array = JSON.parse(json_container);  
  program_array.program.push(this.tv_program);  
  fs.writeFileSync(path.join(__static, '/data.json'), JSON.stringify(program_array), 'utf8');  
  alert(this.tv_program.title + ' 해당 방송 프로그램 정보를 추가합니다.')  
  this.initData();  
  this.reloadOther();  
},
```

Function #2

DeleteProgram

: 저장된 데이터에서 선택된 프로그램 정보 삭제한다.

```
methods: {
  loadProgram() {
    const json_container = fs.readFileSync(path.join(__static, '/data.json'), 'utf8');
    this.program_array = JSON.parse(json_container);
  },
  deleteProgram() {
    this.program_array.program.forEach(element => {
      this.program_array.program.splice(element.title.indexOf(this.delete_value),1);
    });
    fs.writeFileSync(path.join(__static, '/data.json'), JSON.stringify(this.program_array), 'utf8');
    this.loadProgram();
  },
  allDeleteProgram() {
  }
}
```

Function #3

RatingProgram

: 저장된 방송프로그램에서 장르별 시청률 순위를 제공한다.

.

```
methods: {  
  ratingProgram() {  
    const json_container = fs.readFileSync(path.join(__static, '/data.json'), 'utf8');  
    let array = JSON.parse(json_container);  
    this.program_array = array.program.sort((prev,next)=> {  
      return next['rating'] - prev['rating'];  
    })  
  }  
}
```

Function #4


ScheduleProgram

: 편성 일자를 입력 받아 편성메커니즘에 맞게 편성표를 제공한다.


```
getDayOfTheWeek() {  
  let day_of_the_week = new Date(this.date).getDay();  
  if(day_of_the_week == '0' || day_of_the_week == '6') {  
    return 'weekend';  
  } else {  
    return 'weekday';  
  }  
}
```



```
, ratingProgram() {  
  const json_container = fs.readFileSync(path.join(__static, '/data.json'), 'utf8');  
  let array = JSON.parse(json_container);  
  let drama_array = array.program.filter(data => data.genre == '드라마');  
  let entertainment_array = array.program.filter(data => data.genre == '예능');  
  this.sort_program.drama = this.sortingArray(drama_array);  
  this.sort_program.entertainment = this.sortingArray(entertainment_array);  
}
```



```
, calcTime(prev_time, runtime) {  
  let next_time = prev_time;  
  if(Number(next_time.minute) + Number(runtime) >= 60) {  
    next_time.hour = Number(next_time.hour) + 1;  
    next_time.minute = Number(next_time.minute) + Number(runtime) - 60;  
    if(Number(next_time.minute) >= 60) {  
      next_time.hour = Number(next_time.hour) + 1;  
      next_time.minute = Number(next_time.minute) - 60;  
    }  
  } else {  
    if(Number(next_time.minute) >= 60) {  
      next_time.hour = Number(next_time.hour) + 1;  
      next_time.minute = Number(next_time.minute) - 60;  
    } else {  
      next_time.minute = Number(next_time.minute) + Number(runtime);  
    }  
  }  
}
```



```
switch (this.getDayOfTheWeek()) {  
  case 'weekday':  
    this.schedule = [  
      drama[3],  
      drama[2],  
      entertainment[1],  
      entertainment[3],  
      drama[1],  
      drama[0],  
      entertainment[0],  
      entertainment[2]  
    ];  
    this.day_of_week = '평일';  
    break;  
  case 'weekend':  
    this.schedule = [  
      drama[3],  
      drama[2],  
      entertainment[1],  
      entertainment[0],  
      entertainment[2],  
      entertainment[3],  
      drama[0],  
      drama[1]  
    ];  
    this.day_of_week = '주말';  
    break;  
}
```


Q & A
