



인생은 영화처럼

영화 조회 프로그램



목 차

1

프로그램 소개

2

코드 소개

3

구현 화면

4

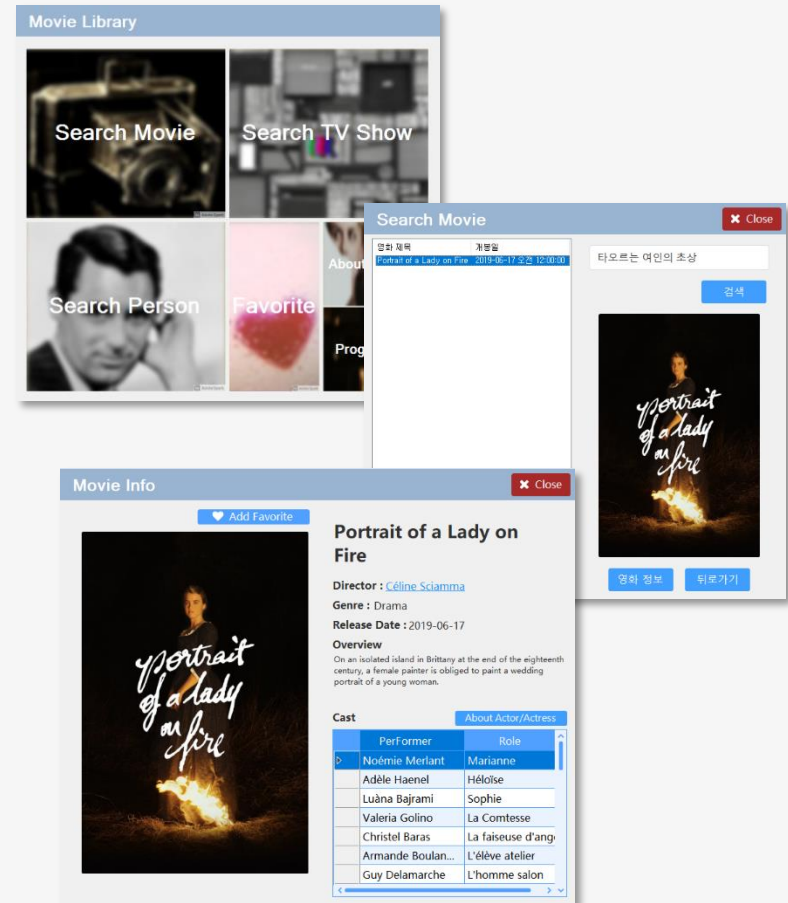
테스트 케이스

1

프로그램 소개

개요

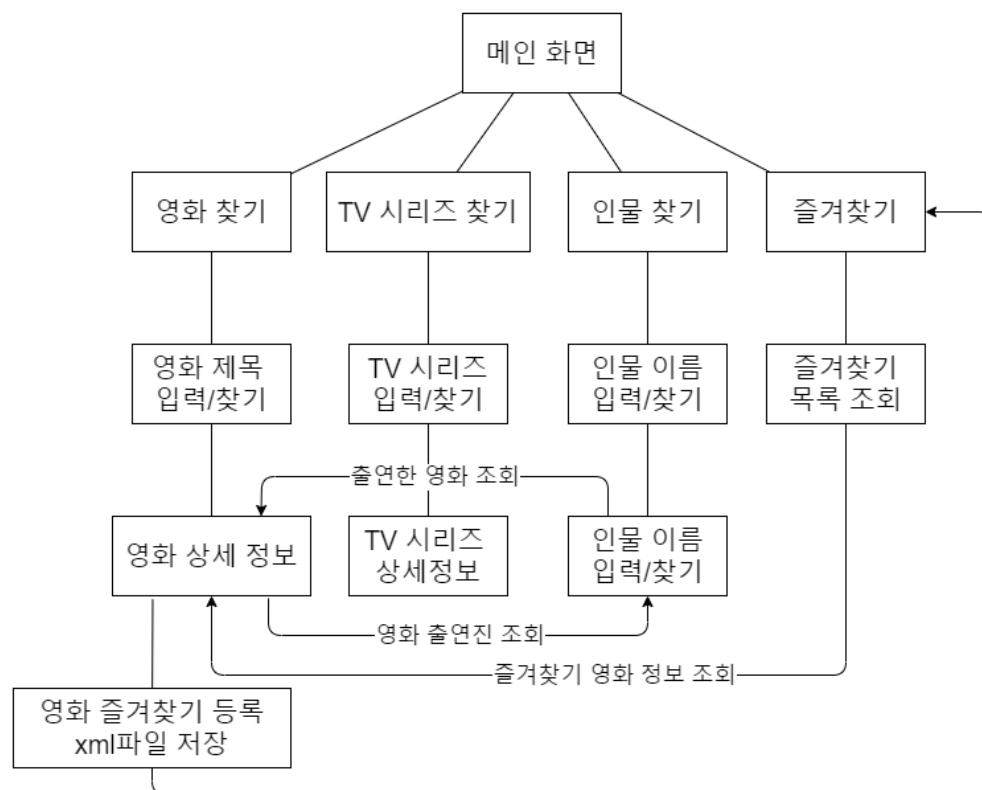
API를 활용하여 영화나 TV시리즈와 관련된 정보들을 더욱 간편하게 얻을 수 있게 하고, 즐겨 찾기로 등록하여 본인이 좋아하는 작품들을 편히 찾아볼 수 있게 디자인 했습니다.



프로그램 소개

순서도

영화나 TV시리즈, 인물의 각 정보를 검색하고, 검색결과에서 서로서로 유동적으로 정보 조회가 가능하게 설계했습니다.



2

코드 소개

// 이름으로 영화 검색

참조 1개

```
public static List<MovieInfo> searchMovies(string movieName)
{
    List<MovieInfo> movies = new List<MovieInfo>();
    SearchContainer<SearchMovie> results = client.SearchMovieAsync(movieName).Result;
    Console.WriteLine($"Got {results.Results.Count:N0} of {results.TotalResults:N0} results");
    System.Windows.Forms.MessageBox.Show($"Got {results.Results.Count:N0} of {results.TotalResults:N0} results", "검색 완료");
    foreach (SearchMovie movie in results.Results)
    {
        movies.Add(new MovieInfo(movie.Id, movie.Title, movie.ReleaseDate.ToString()));
    }
    return movies;
}
```

영화 찾기 API

영화 이름을 영어나 한글로 입력 받아, 동일하거나 입력 받은 문자를 포함한 영화들을 모두 List에 담아 반환해준다.

2

코드 소개

```
// 영화 포스터 불러오기
참조 3개
public static Bitmap getMovieImage(int id)
{
    Movie movie1 = client.GetMovieAsync(id).Result;
    WebClient Downloader = new WebClient();
    Stream ImageStream = Downloader.OpenRead("https://www.themoviedb.org/t/p/w600_and_h900_bestv2"
        + movie1.PosterPath);
    Bitmap DownloadImage = Bitmap.FromStream(ImageStream) as Bitmap;
    return DownloadImage;
}
```

영화 포스터 API

선택한 영화의 id값을 받아 영화 객체를 생성하고
영화 객체의 PosterPath값을 활용하여 이미지를 불러온다.

2

코드 소개

```
// 특정 영화 정보 불러오기  
참조 1개  
public static async Task<Movie> getMovieAsync(int id)  
{  
    ...  
    return await client.GetMovieAsync(id, MovieMethods.Credits);  
}
```

영화 정보 API

선택한 영화의 id값을 받아 영화 객체를 생성하여 객체를 반환해준다. (Task 비동기 프로그래밍 사용)

2

코드 소개

Xml파일 불러오기

임의의 List에 xml
파일의 정보들을
저장하여 사용한다.

```
// xml파일에 있는 정보들을 List에 추가
참조 2개
public static void Load()
{
    movies.Clear();
    try
    {
        string carsOutput = File.ReadAllText(@"./Movies.xml");
        XElement carXElement = XElement.Parse(carsOutput);
        foreach (var item in carXElement.Descendants("movie"))
        {
            string tempMovieTitle = item.Element("movieTitle").Value;
            int tempMovieId = int.Parse(item.Element("movieId").Value);
            string tempMovieGenre = item.Element("movieGenre").Value;
            string tempMovieRDate = item.Element("movieRDate").Value;

            MovieInfo tempMovie = new MovieInfo(tempMovieTitle, tempMovieId, tempMovieGenre, tempMovieRDate);
            movies.Add(tempMovie);
        }
    }
    catch (Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
        printLog(ex.Message);
        printLog(ex.StackTrace);

        CreateFile();
        Save();
        Load();
    }
}
```


2

코드 소개

Xml파일 저장하기

List에 들어있는 정보들을 Xml파일 텍스트로 저장한다.

```
참조 3개
public static void Save()
{
    string booksOutput = string.Empty;
    booksOutput += "<movies>\n";
    // 파일이 있을 경우 List에 새로 추가한 데이터를 xml에 저장
    if (movies.Count > 0)
    {
        foreach (var item in movies)
        {
            booksOutput += "<movie>\n";
            booksOutput += $"    <movieTitle>{item.MovieTitle}</movieTitle>\n";
            booksOutput += $"    <movieId>{item.MovieId}</movieId>\n";
            booksOutput += $"    <movieGenre>{item.MovieGenre}</movieGenre>\n";
            booksOutput += $"    <movieRDate>{item.MovieReleaseDate}</movieRDate>\n";
            booksOutput += "</movie>\n";
        }
    }

    // List에 아무것도 없는 경우 임시 데이터 하나를 추가
    else
    {
        booksOutput += "<movie>\n";
        booksOutput += "    <movieTitle>타오르는 여인의 초상</movieTitle>\n";
        booksOutput += "    <movieId>531428</movieId>\n";
        booksOutput += "    <movieGenre>Drama</movieGenre>\n";
        booksOutput += "    <movieRDate>2019-06-17</movieRDate>\n";
        booksOutput += "</movie>\n";
    }
    booksOutput += "</movies>";
    File.WriteAllText(@"./Movies.xml", booksOutput);
}
```

2

코드 소개

```
// 로그파일 기록
참조 2개
public static void printLog(string contents, string name = "favoriteMovies")
{
    string directory = "favoriteMovies";
    DirectoryInfo di = new DirectoryInfo(directory);
    if (!di.Exists)
    {
        di.Create();
    }

    using (StreamWriter writer = new StreamWriter(directory + @"\" + name + ".txt", true))
    {
        writer.WriteLine(contents);
    }
}
```

Xml파일 저장하기

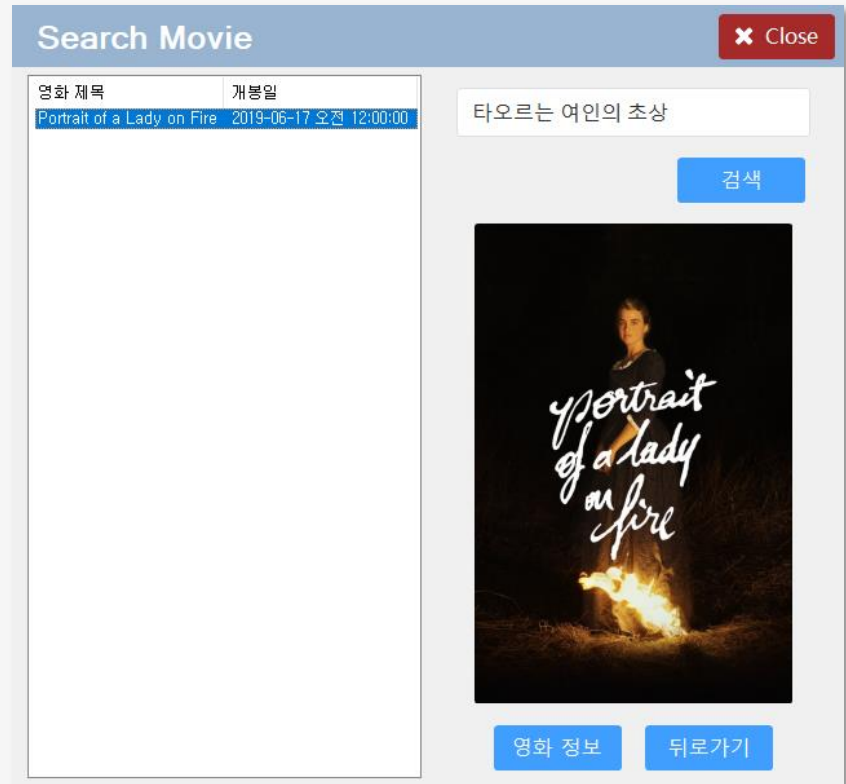
유용하게 쓸 수 있는 로그파일을 생성하고
로그를 저장한다.

3

구현 화면

검색 화면

찾을 제목을 입력하고 검색하여 정보를 찾고, ListView에서 선택한 정보의 상세정보 보기를 선택하여 영화 정보 창으로 들어간다.




3

구현 화면

영화 정보 화면

영화 상세 정보들이 표시되며, 인물 정보로 바로 접속 가능하고, 즐겨 찾기 추가가 가능하다.

Movie Info
Close



Add Favorite

Portrait of a Lady on Fire

Director : [Céline Sciamma](#)

Genre : Drama

Release Date : 2019-06-17


Overview

On an isolated island in Brittany at the end of the eighteenth century, a female painter is obliged to paint a wedding portrait of a young woman.

Cast

About Actor/Actress	
Performer	Role
Noémie Merlant	Marianne
Adèle Haenel	Héloïse
Luàna Bajrami	Sophie
Arnaud Binard	La Comtesse
Thomas Giorgetti	La faiseuse d'ange
Julien Boisselier	L'élève atelier
Armande Maréchal	L'homme salon

Person Info
Close



Noémie Merlant

Birth : 1988-11-27

Place of Birth : Paris, France

Biography

Noémie Merlant (born November 27, 1988) is a French actress.

Known For

About Movie	
Movie	Role
A Good Man	Benjamin Adler
Jumbo	Jeanne
Shakira	The Saleswoman
République	Nora
Portrait of a Lady ...	Marianne
Curiosa	Marie de Regnier

3

구현 화면


즐거 찾기 화면

즐거 찾기에 추가한 영화
들이 표시되며, 해당 영화
정보 조회가 가능하다.

My Favorite

Close

영화 제목	장르	개봉일
Portrait of a L...	Drama	2019-06-17
One Cut of th...	Comedy	2017-11-04 오...



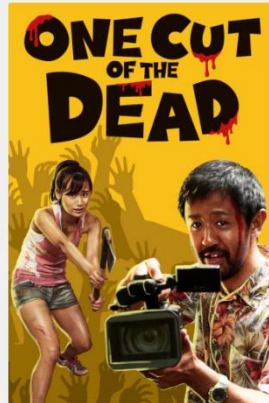
영화 정보

뒤로가기

Movie Info

Close

Add Favorite



One Cut of the Dead

Director : Shinichiro Ueda

Genre : Comedy

Release Date : 2017-11-04

Overview

Things go badly for a hack director and film crew shooting a low budget zombie movie in an abandoned Second World War Japanese facility, when they are attacked by real zombies.

Cast

PerFormer	Role
Takayuki Hamatsu	Director Higurashi
Yuzuki Akiyama	Chinatsu
Kazuaki Nagaya	Kazuaki Kamiya
Harumi Shuhama	Nao
Mao	Mao Higurashi
Hiroshi Ichihara	Kasahara
Manabu Hosoi	Manabu Hosoda

4

테스트 케이스

분류	조건	결과
API 접속	API에 접속하여 정보를 불러올 수 있습니까?	○
영화검색	영화를 검색하고, 포스터를 불러옵니까?	○
정보 연동	영화 상세정보에서 인물 상세정보를 불러옵니까?	○
즐거 찾기	즐거 찾기 버튼을 눌러 xml파일에 저장 가능합니까	○
즐거 찾기 정보	즐거 찾기 한 영화들을 제대로 불러옵니까?	○
	즐거 찾기 한 영화들의 상세 정보를 불러옵니까?	○



감사합니다