

프로젝트 보고서

컴퓨터공학부 202011681 안희주

컴퓨터공학부 202111668 최현솔

컴퓨터공학부 202117645 최아현

Model -DetectLangsInformationModel.cs -SearchingInformationsModel.cs -ShowInformationModel.cs -TranslateInformationModel.cs -VideoInformationModel.cs	View -Intro.xaml -MainView.xaml	ViewModel -MainVM.cs -TranslatorAPI.cs -YoutubeAPI.cs
Command -GoToMainView.cs -GoToWebsiteCMD.cs -SearchingCMD.cs	Converter -Likecntcomma.cs -Viewcntcomma.cs	

Model

DetectLangsInformationModel.cs

검색하고자 하는 제목을 다른 언어로 번역할 때 선택할 수 있는 언어(string 형으로 구현된 property인 langCode)들의 코드를 저장하기 위해 만든 Model이다. 선택하고자 하는 언어가 항상 바뀌기에 INotifyPropertyChanged를 이용하여 바뀌었다는 정보를 전달해주고 있다. .

SearchingInformationsModel.cs

각 유튜브 영상의 id와 해당하는 제목을 저장하기 위해 만든 Model이다. Items를 ObservableCollection으로 선언해주었다.

ShowInformationModel.cs

ShowInformationModel의 property들을 초기화해줌과 동시에 영상의 제목, 설명, 좋아요 수, 조회수, 업로드 날짜, imageurl의 정보를 담고있는 Model이다. 영상의 정보들을 관리하기 쉽도록 해당 정보를 저장하는 임의의 Model 이다. 해당 Model 또한INotifyPropertyChanged를 이용하여 바인딩되었다.

TranslateInformationModel.cs

번역하고자 하는 언어의 타입, 번역될 언어의 타입을 TranslationInformation 객체에 저장하기 위해 만든 Model이다. 파파고 API를 통해 번역된 텍스트를 저장하는 것이 핵심이며, 코드 내에서 검색어, 영상의 제목, 영상 설명글을 번역할 때 사용된다. 이 또한 INotifyPropertyChanged를 통해 바인딩되었다.

VideoInformationModel.cs

Listview 의 아이템(출력된 영상목록)을 선택했을 때 각 영상의 id를 유튜브 비디오 API를 사용하여 title, description, thumbnails, viewcount, likecount, uploadedDate들의 영상정보를 가져와 저장하는 Model이다. 각기 다른 영상이 선택될 때마다 영상의 정보들이 실시간으로 바뀌어야 하기 때문에 INotifyPropertyChanged를 이용하였다

View-ViewModel(MainVM)

Intro.xaml

① Button_Command ⇔ Binding ⇔ Commands_GoToMainViewCMD

코드비하인드에 코드를 쓰지 않게 하기 위해 Button(startpage)과 Command의 GoToMainViewCMD를 바인딩해주었다. Intro의 start버튼을 누르면 Mainview의 창으로 넘어가게된다.

MainView.xaml

① ComboBox_ItemsSource ⇔ Binding ⇔ MainVM_TarLangs

콤보박스에서 어떤 항목을 선택했는지에 따라 번역될 언어의 타입이 달라진다. 배열 형태로 저장된 정보를 View에 바인딩해주고, 보다 유연성 있는 데이터 형식을 위해 ObservableCollection<String>을 사용해 TarLangs를 별도로 저장해주었다.

② ComboBox_SelectedItem ⇔ Binding ⇔ MainVM_SelectedTarLangs

콤보박스의 item과 MainVM안에 있는 string형 property SelectedTarLangs를 바인딩하여 각 언어들의 코드(selectedTarLang)와 지역코드(RegionCode)를 선택한 item에 맞게 저장해주었다.

③ Button_CommandParameter ⇔ Binding ⇔ Textbox_Text(inputtitle)

inputtitle 텍스트박스의 텍스트와 검색하고자 하는 searchingtxt 텍스트를 바인딩시켰다.

④ ListView_ItemSource ⇔ Binding ⇔ MainVM_SearchedVideos

배열 형태로 저장된 정보를 View에 바인딩해주기 위해 ObservableCollection<String>을

사용해 SearchedVideos를 별도로 저장해주었다. SearchingCMD를 통하여 정보가 저장된 후 View에 보여진다.

⑤ ListView_SelectedItem ⇔ Binding ⇔ MainVM_SelectedVideo

listview안에 있는 item을 선택하면 SelectedVideo의 GetVideoInfo 함수에 접근하여 SelectedVideo변수에 선택된 동영상의 id를 유튜브 비디오 api를 사용하여 like, view, UploadedDate, Title, Description, imageurl의 정보를 ShowInformationModel 타입 객체에 저장한다. 이렇게 저장된 객체 정보를 바인딩 된 각각의 텍스트박스에 해당하는 정보를 보여준다.

⑥ TextBlock_Text (titletokr) ⇔ Binding ⇔ MainVM_ShowInfo_Title (영상제목 출력)

String형 property인 title과 textblock의 text를 ShowInformationModel에서 INotifyPropertyChanged를 통해 바인딩시켜주었다.

번역되어 출력될 텍스트를 TranslatorAPI를 사용하여 ShowInfo의 title에 저장되게 하였다.

⑦ TextBlock_Text (descriptiontokr) ⇔ Binding ⇔ MainVM_ShowInfo_Description

(영상설명 출력) String형 property인 description과 textblock의 text를

ShowInformationModel에서 INotifyPropertyChanged를 통해 바인딩시켜주었다.

번역되어 출력될 텍스트를 TranslatorAPI를 사용하여 ShowInfo의 description에 저장되게 하였고, 만약 보여질 영상의 부연설명이 없다면 빈문자열을 출력하도록 구현하였다.

⑧ TextBlock_Text (viewcnt) ⇔ Binding ⇔ MainVM_ShowInfo_View (영상 조회수 출력)

Long형 property인 view와 textblock의 text를 ShowInformationModel에서

INotifyPropertyChanged를 통해 바인딩시켜주었다.

⑨ TextBlock_Text (likecnt) ⇔ Binding ⇔ MainVM_ShowInfo_Like (영상 좋아요 수 출력)

Long형 property인 like와 textblock의 text를 ShowInformationModel에서

INotifyPropertyChanged를 통해 바인딩시켜주었다.

⑩ TextBlock_Text (date) ⇔ Binding ⇔ MainVM_ShowInfo_UploadedDate

(영상 업로드일 출력) string형 property인 uploadedDate와 textblock의 text를

ShowInformationModel에서 INotifyPropertyChanged를 통해 바인딩시켜주었다.

GoToMainView.cs

Button_Command ⇔ Binding ⇔ Commands_GoToMainViewCMD

Command가 실행되면 MainView의 객체를 생성하고 mainwindow가 보여지게 된다.

Intro의 startpage버튼을 누르면 Mainview의 창으로 넘어간다.

GoToWebsiteCMD.cs

Button_Command ⇔ **Binding** ⇔ Commands_GoToWebsiteCMD(해당 영상의 url로 이동)
Command가 실행되었을 때 Process.Start메서드를 사용하여 웹페이지를 띄우게 구현했다.
Listview에서 선택된 영상을 각 유튜브 링크에 해당되는 영상의 SelectedVideo_Id 를 사용하여 웹페이지를 띄우게 된다.

SearchingCMD.cs

Button_Command ⇔ **Binding** ⇔ Commands_SearchingCMD
search버튼을 클릭하면 Searchingtxt의 GetSearchingInfo함수에 접근하여 만약 콤보박스의 item이 어느것도 선택되지 않았거나, searchingtxt의 값이 null이거나, searchingtxt의 언어 타입이 한국어나 영어가 아닐 경우 MessageBox 를 사용하여 오류메세지를 띄우도록 구현했다. item이 알맞게 선택되었다면 텍스트박스의 텍스트의 언어타입을 가져와 저장한 후 콤보박스를 통해 선택된 언어로 번역한다. 번역되어진 텍스트를 유튜브 검색 api를 사용하여 반환한 후 이렇게 반환 된 영상들이 searched 변수에 저장되어지고, 반복문을 돌며 50개의 영상들이 Listview에 보여지게 된다.

Likecntcomma.cs

가독성을 높이고자 좋아요수를 천의자리수마다 ,(coma)가 보여지게 하기 위해 long형으로 캐스팅 한 후 string.Format을 활용해 세자리수마다 , 가 보여지게 구현하였다.

Viewcntcomma.cs

가독성을 높이고자 조회수를 천의자리수마다 ,(coma)가 보여지게 하기 위해 long형으로 캐스팅 한 후 string.Format을 활용해 세자리수마다 , 가 보여지게 구현하였다.