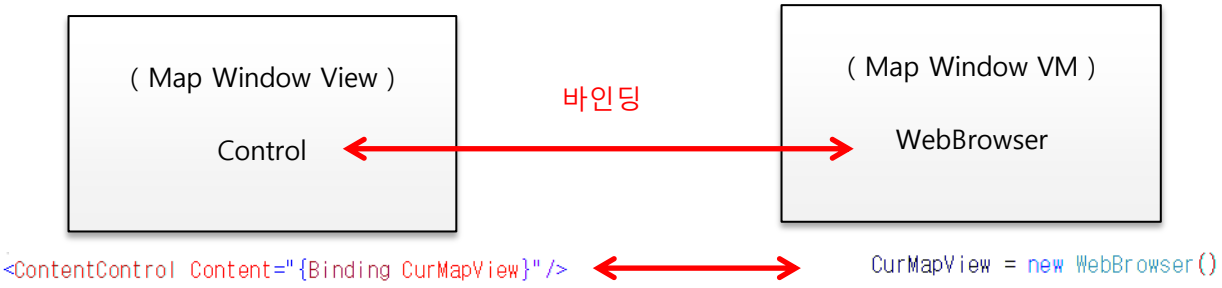
최종 프로젝트 보고서

이성규, 장동하, 양윤성

1. Map Window

- WebBrowser 컨트롤 바인딩.

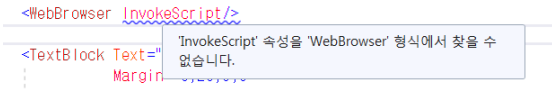


View Model 이 웹 브라우저 컨트롤 그 자체를 갖고 있도록 구현한 후, View 에서 그 컨트롤을 바인딩 하여 표시하도록 구현하였습니다.

WebBrowser 컨트롤은 InvokeScript 라는 이벤트를 통해 Web 과 WPF 간에 정보를 교환합니다. 근데 InvokeScript 이벤트는 컨트롤이 생성이 되야 지만 접근하여 사용할 수 있기 때문에 XAML 에서는 InvokeScript 에 접근하는 것 자체가 되지 않아, 커맨드를 작성할 수 없었습니다.

Remarks

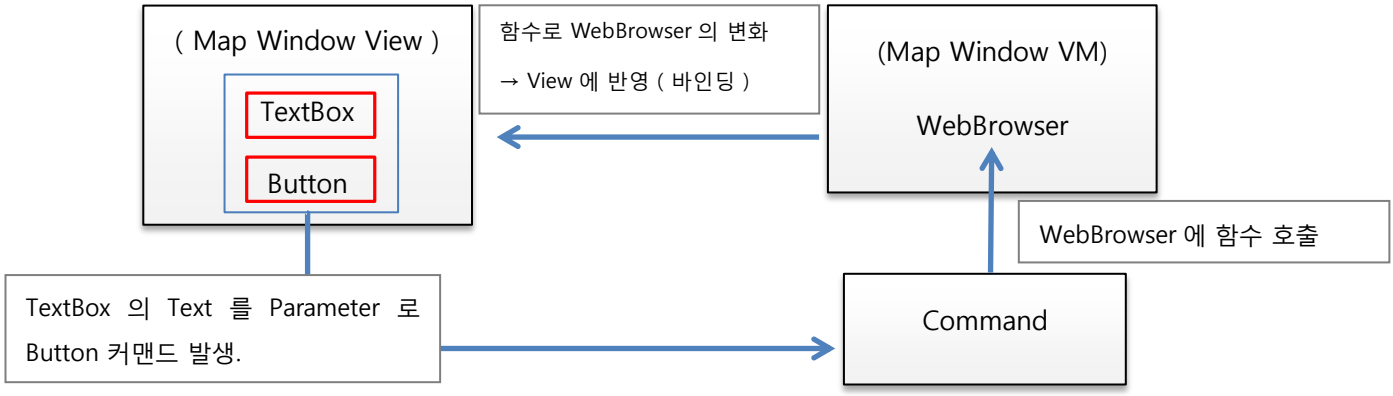
InvokeScript(String) should not be called before the document that implements it has finished loading. You can detect when a document has finished loading by handling the LoadCompleted event.



(출처 : http://msdn.microsoft.com/en-us/library/cc491132.aspx)

그렇기 때문에 WebBrowser 가 생성 된 이후 View Model 에서 직접적으로 접근할 수 있는 수단이 필요하였고, 바인딩을 통해 이러한 접근을 할 수 있는 수단을 만들어 주었습니다.

➢ 검색 버튼 커맨드로서 커맨드 파라미터 값을 Text 박스 내용으로 전송.

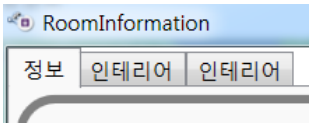


TextBox 의 Text 를 CommandParameter 로 받도록 버튼 Command 를 생성하여, VM 속의 WebBrowser 가 함수를 호출하도록 구현하였고, Binding 관계에 의해 ViewModel 변화가 View 에 반영 되게 됩니다. (WebBrowser 의 변화가 반영)

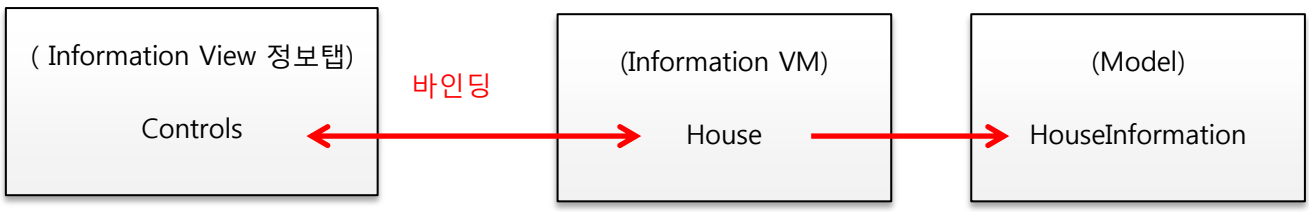
WebBrowser 에 함수를 호출할 인자로 TextBox 의 값을 하고 있는데, 예외처리가 WebBrowser 내부에 딱히 되어있지 않습니다. 이에 예외처리가 필요 했고, 버튼 커맨드 속 CanExecute 함수를 통해, 올바르지 않은 TextBox 값이 WebBrowser 함수 인자에 사용되지 않도록 막아주는 역할을 해 주었습니다.

2. Information Window

Information Window 는 3 개의 탭으로 이루어져있습니다. 이에, 각 탭마다 다른 VM 을 갖고 있어, 총 3 개의 VM 이 한 개의 윈도우에 존재하게 되었습니다.



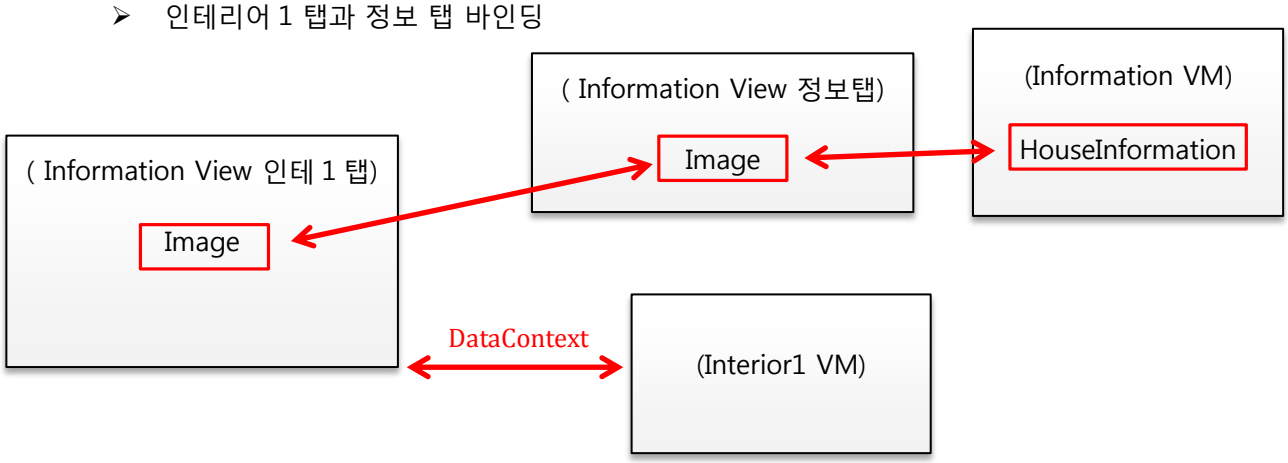
➢ 정보 탭 VM 바인딩.



정보 탭에 있는 거의 모든 컨트롤들은 VM 에 있는 House 프로퍼티와 바인딩이 되어있습니다. House 는 HouseInformation 클래스 프로퍼티인데, 이는 API 로부터 받은 Json 정보를 클래스 형식으로 변경한 Model 클래스 입니다.

VM 의 House 는 받아들이는 API 정보에 따라 데이터가 변하게 되므로, 이를 View 에 바인딩 해주어서 데이터에 따른 알맞은 View 가 제공될 수 있도록 바인딩을 구현해 주었습니다.
이는 수업시간에 하였던 WheatherApp 과 유사한 구조입니다.

➢ 인테리어 1 탭과 정보 탭 바인딩

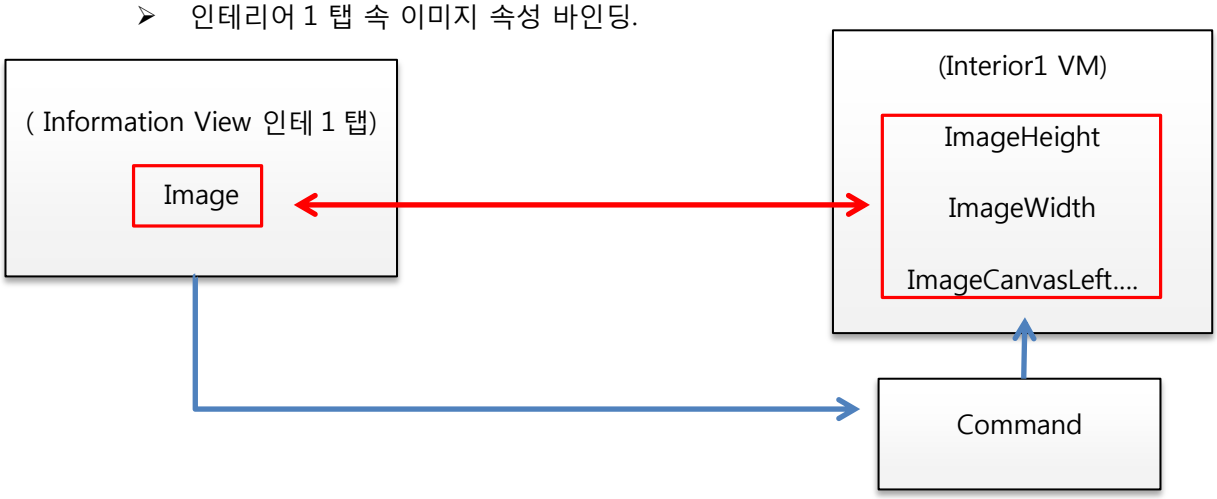


Information Window 에서는 3 개의 탭 정보를 모두 한 개의 VM 이 갖게 구현하면, 너무나도 많은 정보가 한 개의 모델에 들어가게 되므로 각 탭마다 갖는 VM 을 다르게 구현하였습니다.
그렇기 때문에 인테리어 1 탭의 VM 에는 HouseInformation 의 정보를 갖고 있지 않습니다.

이에, 인테리어 1 탭에서 정보 탭이 사용한 HouseInformation 정보를 사용하고, 정보 탭에서의 특정 컨트롤 동작 방식을 모방하고자 바인딩을 사용하게 되었습니다.

즉 이 바인딩을 통해 인테리어 1 VM 에 없는 정보를 사용할 수 있었으며, 이미 구현된 동작방식을 쉽게 구현해 낼 수 있었습니다.

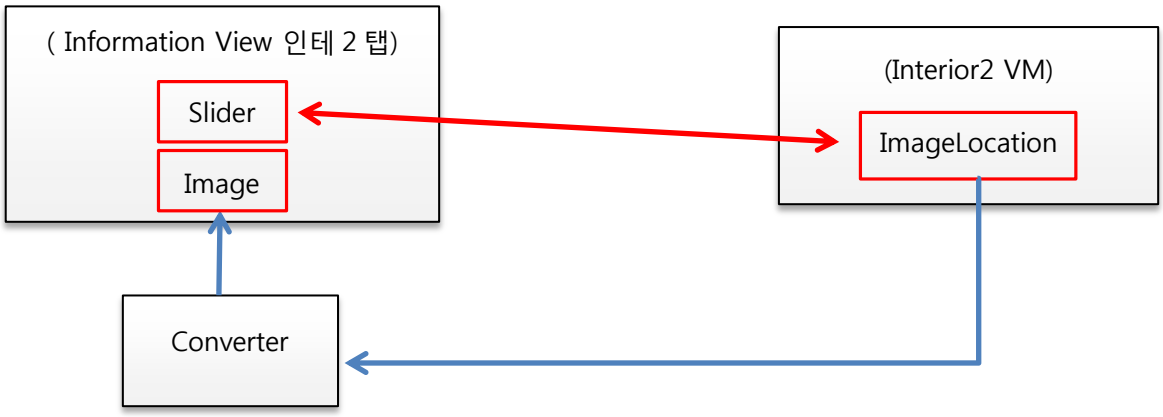
➢ 인테리어 1 탭 속 이미지 속성 바인딩.



인테리어 1 탭에는 인테리어를 해볼 가구 이미지가 Canvas 컨트롤 안에 존재합니다. 이에 관련되어 이미지 컨트롤의 속성인 Source, Height, Width, Canvas.Left, Canvas.Top 속성을 VM 에 특정 프로퍼티와 바인딩 시켜주었고, 이들은 버튼 등의 커맨드에 의해 제어하였습니다.

이렇게 View 의 컨트롤 속성을 바인딩을 통해 커맨드로 제어 해주게 되면 의도하지 않았던 값이 속성에 적용되는 것을 방지하기 편리합니다. 예를 들어, Height, Width 속성의 경우 음수가 들어가면 적용되지 않도록 구현하였고, Canvas.Left, Canvas.Top 은 이미지가 Canveas 안에서만 존재 할 수 있도록 일정 수치 안에서만 조정될 수 있게 구현하였습니다.

➢ 인테리어 2 탭의 Slider Value 바인딩



인테리어 2 탭에는 Slider 와 Image 가 있습니다. Slider 의 Value 값은 VM 에 있는 ImageLocation 프로퍼티와 바인딩이 되어있고, 이 프로퍼티는 Image 의 속성들과 Converter 를 통해 변환되어 바인딩 되어 있습니다.

Image 의 속성을 사용자가 변경시킬 때 시각적으로 변경의 정도를 알 수 있는 방법이 없을까 생각하였습니다. 그에 생각한 것이 Slider 였습니다. Slider 는 값을 사용자가 변경시킴에 따라 시각적으로 어느 정도의 값인지 알 수 있습니다. 이러한 Slider 의 Value 값을 Image 속성에 Converter 로 변환하여 바인딩 해줌으로써, 사용자는 Image 의 속성이 어느 정도 값인지 시각적으로 알 수 있게 되었습니다.