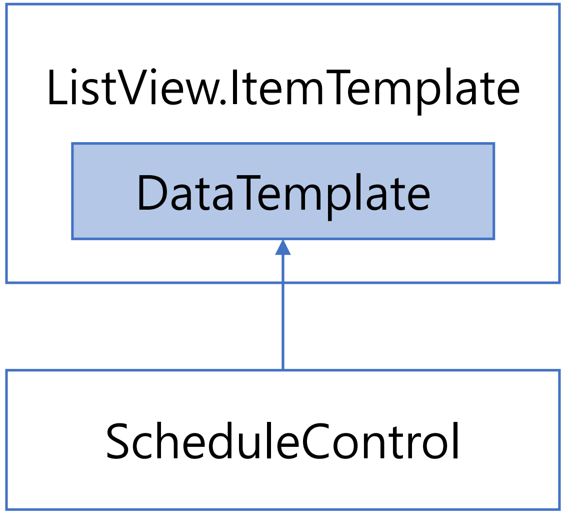
프로그램 설명문서
MergeCalendar

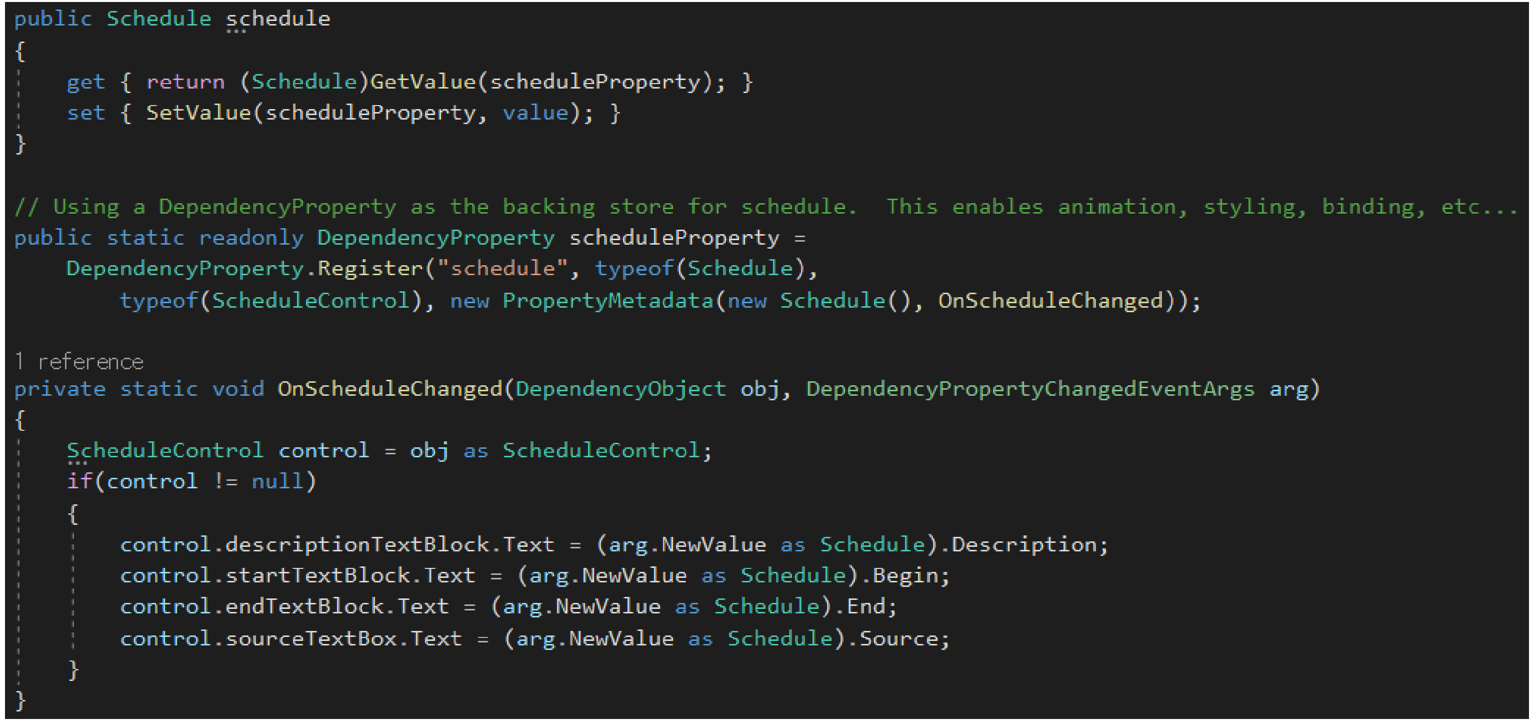
윈도우즈프로그래밍 3분반 1조

201610674 이영훈

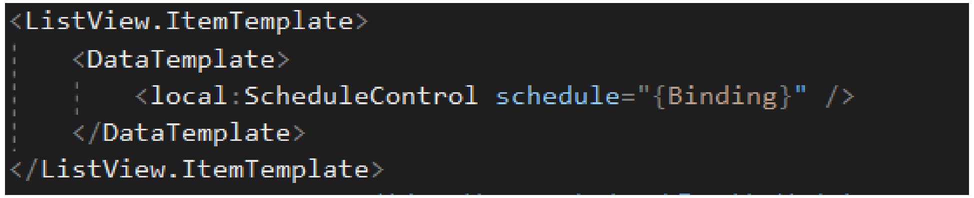
201614906 정종범

DataBinding & DependencyProperty





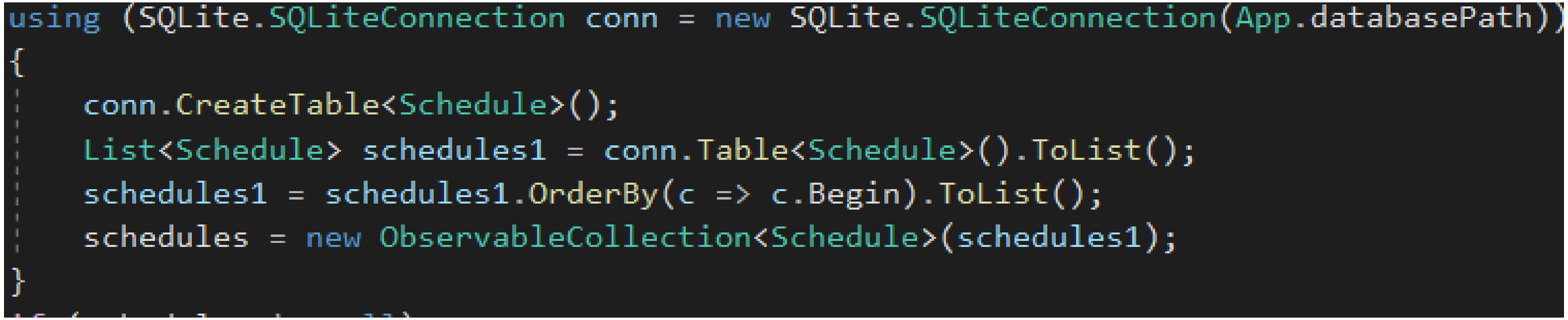
```
public Schedule schedule
{
    get { return (Schedule)GetValue(scheduleProperty); }
    set { SetValue(scheduleProperty, value); }
}

// Using a DependencyProperty as the backing store for schedule.  This enables animation, styling, binding, etc...
public static readonly DependencyProperty scheduleProperty =
    DependencyProperty.Register("schedule", typeof(Schedule),
        typeof(ScheduleControl), new PropertyMetadata(new Schedule(), OnScheduleChanged));

1 reference
private static void OnScheduleChanged(DependencyObject obj, DependencyPropertyChangedEventArgs arg)
{
    ScheduleControl control = obj as ScheduleControl;
    if(control != null)
    {
        control.descriptionTextBlock.Text = (arg.NewValue as Schedule).Description;
        control.startTextBlock.Text = (arg.NewValue as Schedule).Begin;
        control.endTextBlock.Text = (arg.NewValue as Schedule).End;
        control.sourceTextBox.Text = (arg.NewValue as Schedule).Source;
    }
}
```



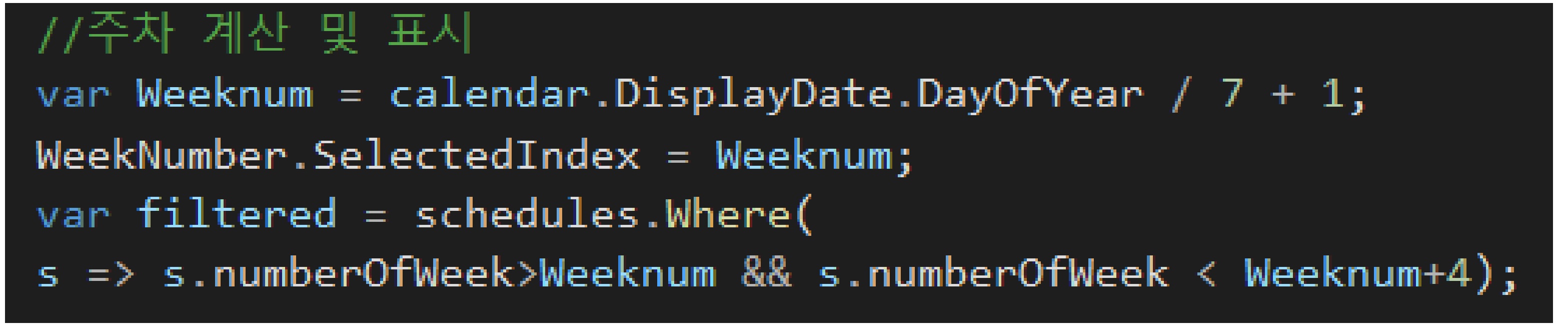
```
<ListView.ItemTemplate>
    <DataTemplate>
        <local:ScheduleControl schedule="{Binding}" />
    </DataTemplate>
</ListView.ItemTemplate>
```

- ListView에 사용하기 위해 ScheduleControl 사용자 정의 컨트롤 생성
- ScheduleControl을 Schedule의 DependcyProperty를 이용해 구현
- ListView의 DataTemplate를 지정해주기 위해서 바인딩 이용

DataBase 튜플 정렬 및 필터링



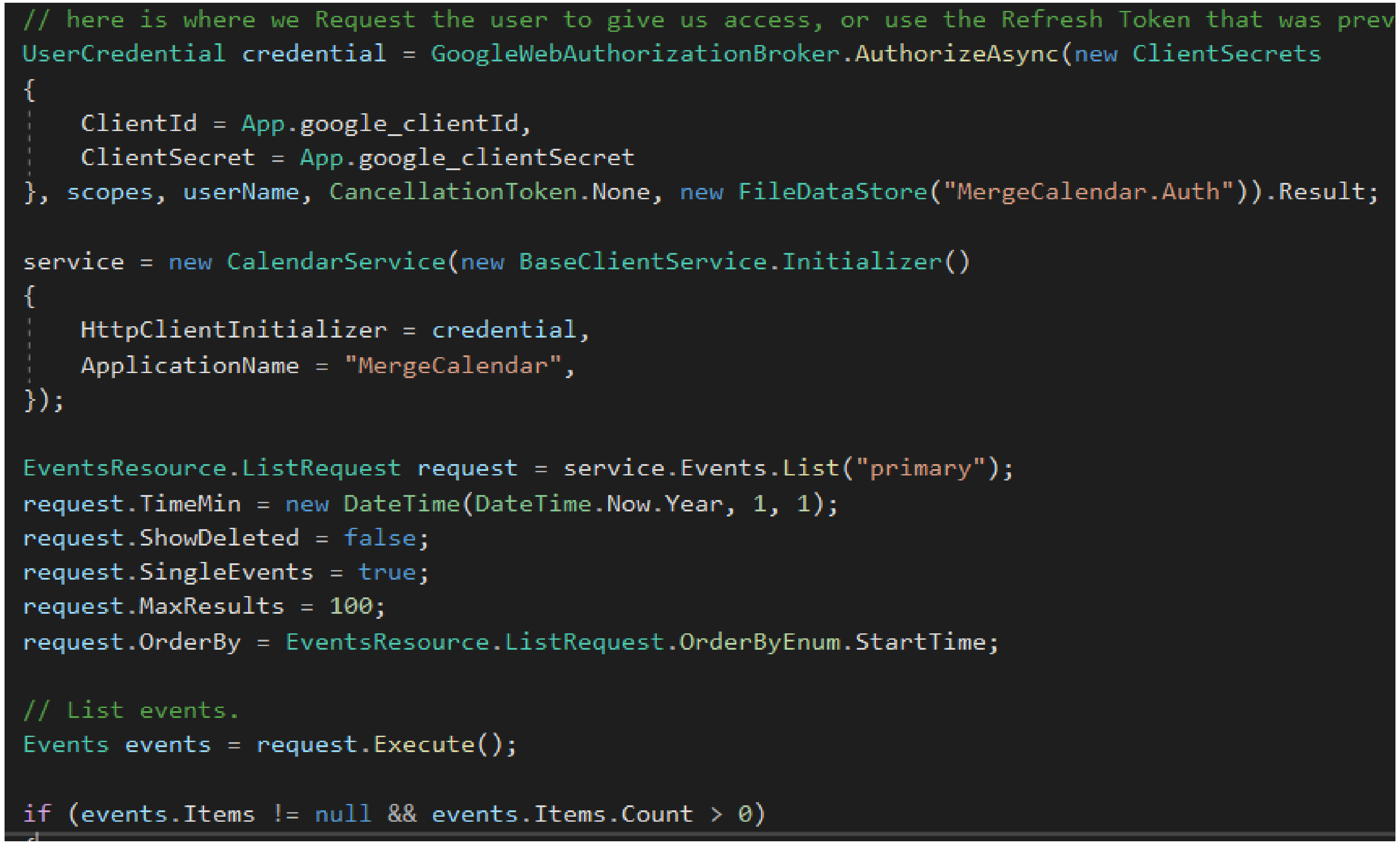
```
using (SQLite.SQLiteConnection conn = new SQLite.SQLiteConnection(App.databasePath))
{
    conn.CreateTable<Schedule>();
    List<Schedule> schedules1 = conn.Table<Schedule>().ToList();
    schedules1 = schedules1.OrderBy(c => c.Begin).ToList();
    schedules = new ObservableCollection<Schedule>(schedules1);
}
```



```
//주차 계산 및 표시
var Weeknum = calendar.DisplayDate.DayOfYear / 7 + 1;
WeekNumber.SelectedIndex = Weeknum;
var filtered = schedules.Where(
s => s.numberOfWeek>Weeknum && s.numberOfWeek < Weeknum+4);
```

- 람다식 이용
- ReadDatabase() 메소드에서 Database의 데이터를 일정 시작 날짜를 기준으로 정렬
- 주차별 계산 후 주차에 맞게 월별 필터링 작업 진행

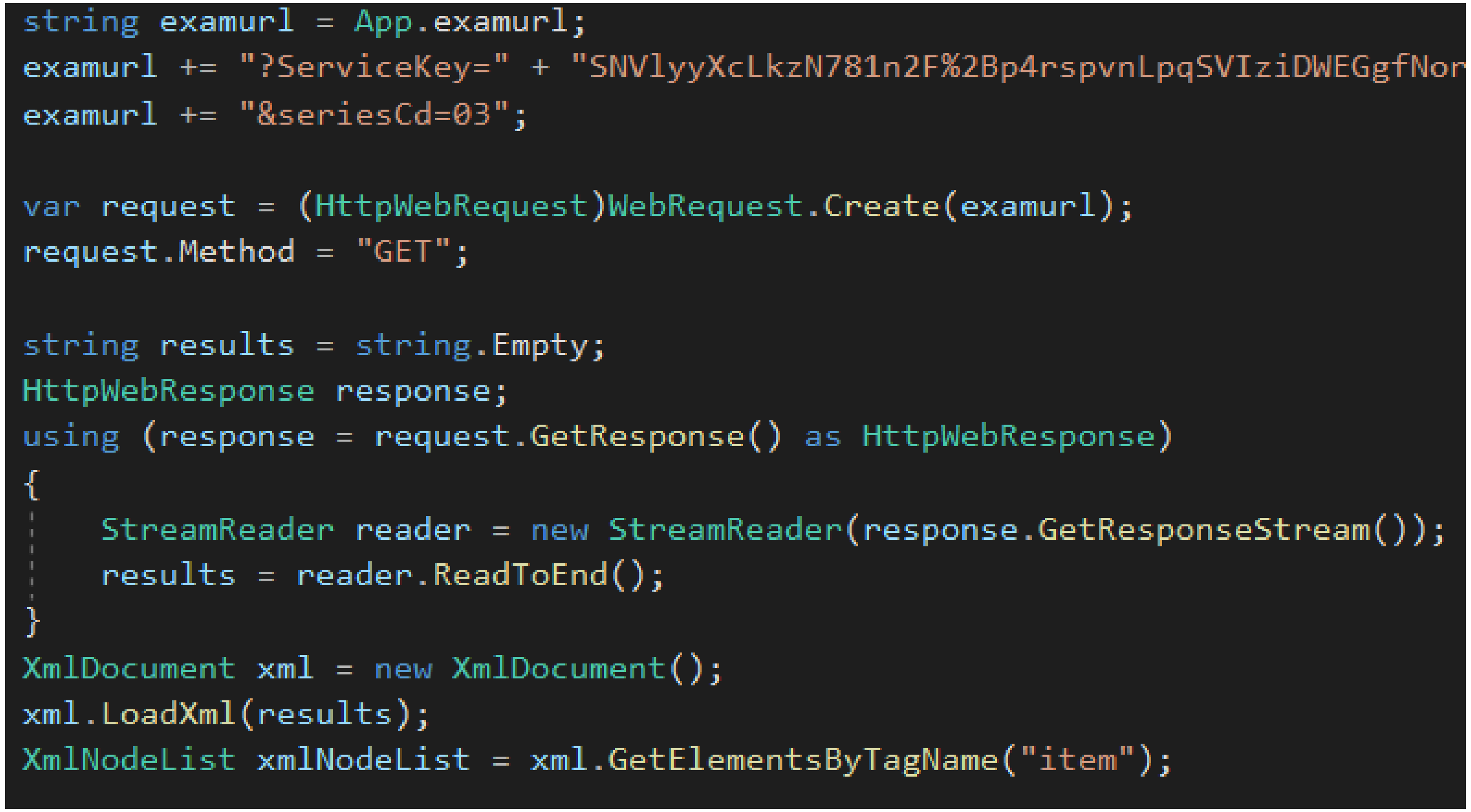
Google API 이용



```
// here is where we Request the user to give us access, or use the Refresh Token that was prev
UserCredential credential = GoogleWebAuthorizationBroker.AuthorizeAsync(new ClientSecrets
{
    ClientId = App.google_clientId,
    ClientSecret = App.google_clientSecret
}, scopes, userName, CancellationToken.None, new FileDataStore("MergeCalendar.Auth")).Result;

service = new CalendarService(new BaseClientService.Initializer()
{
    HttpClientInitializer = credential,
    ApplicationName = "MergeCalendar",
});

EventsResource.ListRequest request = service.Events.List("primary");
request.TimeMin = new DateTime(DateTime.Now.Year, 1, 1);
request.ShowDeleted = false;
request.SingleEvents = true;
request.MaxResults = 100;
request.OrderBy = EventsResource.ListRequest.OrderByEnum.StartTime;

// List events.
Events events = request.Execute();

if (events.Items != null && events.Items.Count > 0)
```

- Oauth 2.0 인증 기능 적용
- Google.Apis.Auth 패키지, Google.Apis.Calendar 패키지 이용

Q-Net API



```
string examurl = App.examurl;
examurl += "?ServiceKey=" + "SNVlyyXcLkzN781n2F%2Bp4rspvnLpqSVIziDWEGgfNor
examurl += "&seriesCd=03";

var request = (HttpWebRequest)WebRequest.Create(examurl);
request.Method = "GET";

string results = string.Empty;
HttpWebResponse response;
using (response = request.GetResponse() as HttpWebResponse)
{
    StreamReader reader = new StreamReader(response.GetResponseStream());
    results = reader.ReadToEnd();
}
XmlDocument xml = new XmlDocument();
xml.LoadXml(results);
XmlNodeList xmlNodeList = xml.GetElementsByTagName("item");
```

- XML 형식으로 받아오기
- XmlNodeList와 XmlNode를 이용해 데이터 Parsing 작업
- seriesCd를 수정하면 시험 과목 변경 가능