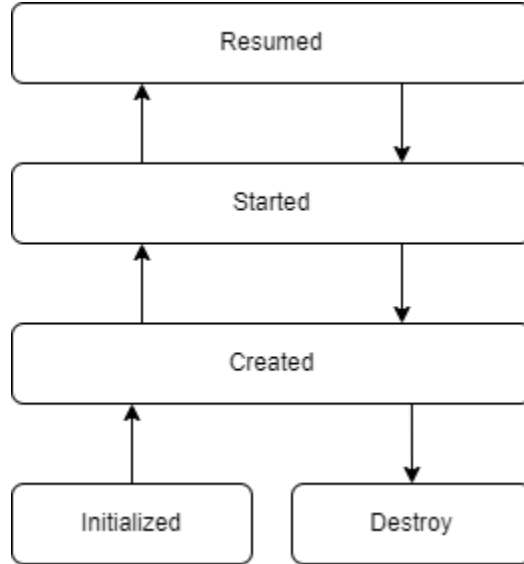
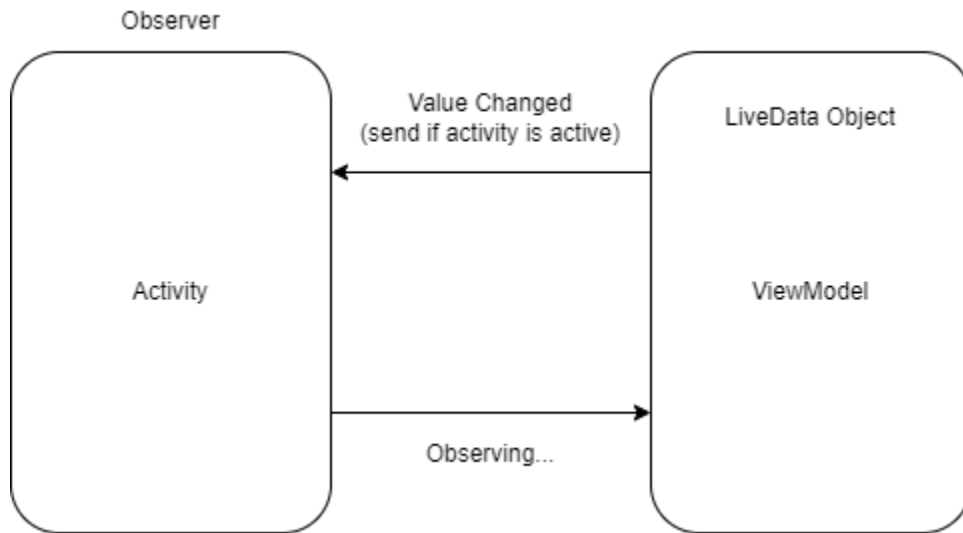


LiveData Android আর্কিটেকচারের একটি কম্পোনেন্ট। ViewModel যেমন Android আর্কিটেকচারের একটি কম্পোনেন্ট, ঠিক তেমনি LiveData Android আর্কিটেকচারের একটি কম্পোনেন্ট। এখানে Model View View Model Android আর্কিটেকচারের সাথে LiveData এর ব্যবহার দেখবো।

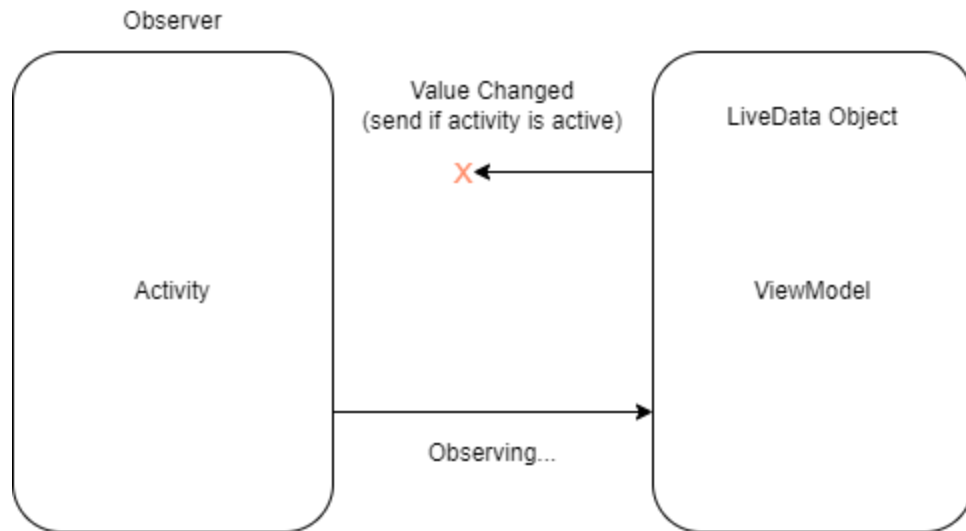
তাহলে LiveData কি, LiveData একটি observable ডেটা হোল্ডার ক্লাস। যা অন্যান্য রেগুলার observable এর মতো নয়। LiveData লাইফসাইকেল সম্পর্কে সচেতন বা লাইফসাইকেলের সাথে সামঞ্জস্য রেখে কাজ করে। অর্থাৎ app এর অন্যান্য কম্পোনেন্ট যেমন activity, fragment বা service র লাইফসাইকেলের সাথে সামঞ্জস্য রেখে কাজ করে।



LiveData শুধুমাত্র app কম্পোনেন্ট একটিভ স্টেটে থাকলেই ডাটা আপডেট করে।



যদি activity paused বা destroyed স্টেটে চলে যায় তাহলে LiveData অবজেক্ট activity কে আপডেট দেয়া বন্ধ করে দেয়। যদি activity, paused বা destroyed স্টেটে চলে যায় তাহলে LiveData অবজেক্ট activity কে আপডেট দেয়া বন্ধ করে দেয়। এবং LiveData, activity আবার এক্টিভ স্টেটে না আসা পর্যন্ত অপেক্ষা করে।



LiveData

Data that is aware of the LifeCycle of it's observer.

সাধারণত LiveData শুধুমাত্র তখনই আপডেট দেয় যখন ডেটা পরিবর্তন হয় এবং observers active থাকে। observer inactive state এ থাকা অবস্থায় কোনো আপডেট হলে, observer inactive state থেকে active state এলে আপডেট চেক্স পায়।

app build.gradle

```
implementation "androidx.lifecycle:lifecycle-extensions:2.2.0"
//noinspection LifecycleAnnotationProcessorWithJava8
annotationProcessor "androidx.lifecycle:lifecycle-compiler:2.4.0"
// ViewModel
implementation "androidx.lifecycle:lifecycle-viewmodel-ktx:2.4.0"

buildFeatures {
    viewBinding true
}
```

MainActivityViewModel নামের একটি ক্লাস তৈরি করি, যে ক্লাস ViewModel ক্লাসকে এক্সটেন্ড করে।

MainActivityViewModel class

```
package com.rzrasel.rzrasetutorial
import androidx.lifecycle.ViewModel

class MainActivityViewModel : ViewModel() {
}
```

MainActivityViewModel ক্লাস ViewModel ক্লাসকে এক্সটেন্ড করা হয়েছে।

MainActivity class

এখন MainActivity ক্লাসে ViewModel প্রোভাইডারের মাধ্যমে MainActivityViewModel ক্লাসকে ইনটিগ্রেট করি।

```
package com.rzrasel.rzrasetutorial
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import androidx.lifecycle.ViewModelProvider
import com.rzrasel.rzrasetutorial.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {
    private lateinit var viewBinding: ActivityMainBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        viewBinding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(viewBinding.root)
        val viewModel =
ViewModelProvider(this) [MainActivityViewModel::class.java]
    }
}
```

viewModel এ ViewModel Provider এর মাধ্যমে MainActivityViewModel ক্লাসকে assigne করা হয়েছে।

MainActivityViewModel class

```
package com.rzrasel.rzrasetutorial
import android.os.CountDownTimer
import androidx.lifecycle.MutableLiveData
import androidx.lifecycle.ViewModel

class MainActivityViewModel : ViewModel() {
    private lateinit var timer: CountDownTimer
    private val _second = MutableLiveData<Int>()
}
```

এখানে timer হচ্ছে CountDownTimer এর একটি অবজেক্ট। এবং _second একটি MutableLiveData অবজেক্ট। LiveData এবং MutableLiveData এর ভিতর পার্থক্য হচ্ছে MutableLiveData একাধিকবার assigne করা যায়।

```
class MainActivityViewModel : ViewModel() {
    private lateinit var timer: CountDownTimer
    private val _second = MutableLiveData<Int>()

    fun startTimer() {
        timer = object : CountDownTimer(10000, 1000) {
            override fun onTick(p0: Long) {
```

```

    }

    override fun onFinish() {
    }
}
}
}

```

startTimer ফাংশনের ভিতরে timer CountdownTimer অবজেক্টকে assigne করা হয়েছে। CountdownTimer দুইটা প্যারামিটার নিয়ে থাকে, প্রথমটি কতক্ষণ ধরে টাইমারটি চলবে বা delay time, আর দ্বিতীয় প্যারামিটারটি কতক্ষণ পরপর চলবে বা interval, আর CountdownTimer এর দুইটি মেথড ওভাররাইড করতে হয় - একটি onTick ও onFinish মেথড। এখানে onTick মেথডটি প্রতিসেকেন্ডে কল হবে, আর onFinish মেথডটি CountdownTimer শেষ হলে কল হবে।

```

timer = object : CountdownTimer(10000, 1000) {
    override fun onTick(p0: Long) {
        val timeLeft = p0 / 1000 // Convert millisecond to second
        _second.value = timeLeft.toInt()
    }

    override fun onFinish() {
    }
}

```

CountdownTimer এর onTick মেথডে timeLeft ভেরিএবলে মিলিসেকেন্ডকে সেকেন্ডে কনভার্ট করে রাখা হয়েছে। এবং পরবর্তীতে তা MutableLiveData তে assigne করা হয়েছে। এখানে onTick মেথডটি প্রতিসেকেন্ডে কল হবে এবং MutableLiveData ও প্রতিসেকেন্ডে আপডেট হবে।

```

timer = object : CountdownTimer(10000, 1000) {
    override fun onTick(p0: Long) {
        val timeLeft = p0 / 1000 // Convert millisecond to second
        _second.value = timeLeft.toInt()
    }

    override fun onFinish() {
    }
}.start()

```

এবং সবশেষে CountdownTimer এর start ফাংশনটি কল করা হয়েছে।

```

fun stopTimer() {
    timer.cancel()
}

```

MainActivityViewModel ক্লাসে stopTimer নামের আরেকটি মেথড তৈরি করি। এখানে timer কে বন্ধ বা cancel করতে পারবে।

```

fun second(): LiveData<Int> {

```

```
    return _second  
}
```

Reference

[LiveData Explained - Android Architecture Component | Tutorial \(Stevdza-San\)](https://youtu.be/suC0OM5gGAA)

<https://youtu.be/suC0OM5gGAA>

Rz Rasel