Event Driven Design Pattern
MVVM

wizeline.com | nicole.terc@wizeline.com

WIZELINE®

mvvm-sample

WIZELINE.COM

What is MVVM?

- MVVM stands for Model-View-ViewModel
- Design pattern focused on event driven programming.
- Abstracts the view behavior, removes all the logic from it.

Components

- **Model:** Data source, handles all the business logic.
- **View:** Receives user input and binds to the ViewModel exposed hooks and data streams.
- **ViewModel:** Core of the app logic. Provides data streams and hooks for the view. Link between View and Model.

Model

Pojo.java

```
public class Pojo {
  private int id;
  private String message;

  public int getId() {
     return id;
  }

  public void setId(int id) {
     this.id = id;
  }

  public String getMessage() {
     return message;
  }

  public void setMessage(String message) {
     this.message = message;
  }
}
```

Model

DataBase.java

```
public class DataBase {

   public List<Pojo> getSavedPojos(){
      List<Pojo> list = new ArrayList<>();
      // retrieve pojo list from local storage
      return list;
   }

   public void savePojo(Pojo pojo){
      // Save Pojo in database
   }

   public void updatePojo(Pojo pojo){
      // Update Pojo in database
   }
}
```

ViewModel

ViewModel.java

```
public abstract class ViewModel extends BaseObservable {

  public void onStart() {
  }

  public void onResume() {
  }

  public void onPause() {
  }

  public void onStop() {
  }

  public void onDestroy() {
  }

}
```

ViewModel

PojoViewModel.java

```
public class PojoViewModel extends ViewModel {
  private Pojo item;
  private DataBase dataBase;

  ...

  public String getCapitalizedMessage(){
     String message = capitalizeFirstLetters(item.getMessage());
     return message;
  }

  public void updateMessage(String message){
     item.setMessage(message);
     dataBase.updatePojo(item);
  }

  ...

}
```

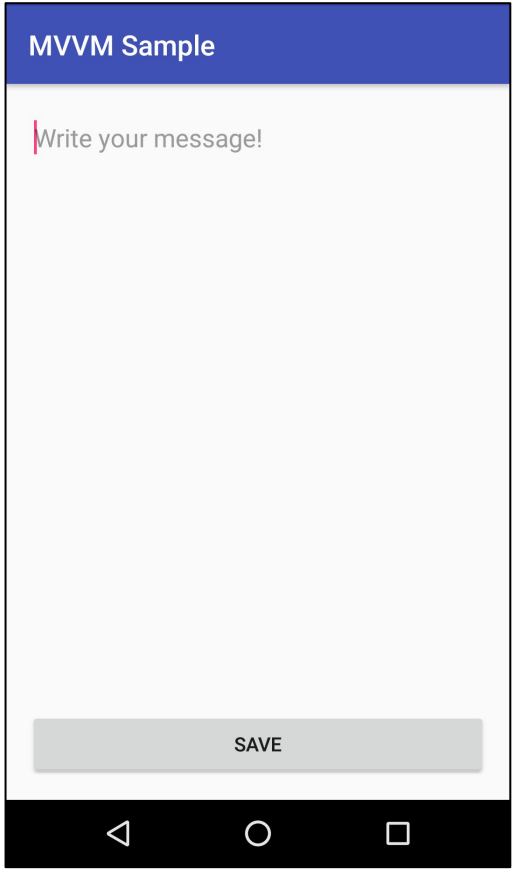
View

View.xml

```
<RelativeLayout
  android:layout_width="match_parent"
  android:layout_height="match_parent">

  <EditText
     android:id="@+id/message"
     android:layout_width="match_parent"
     android:layout_height="wrap_content"
     android:layout_above="@+id/btn_save"
     android:layout_alignParentTop="true"
     android:hint="Write your message!"
     android:inputType="textMultiLine"/>

  <Button
     android:id="@+id/btn_save"
     android:layout_width="match_parent"
     android:layout_height="wrap_content"
     android:layout_alignParentBottom="true"
     android:text="Save"/>
</RelativeLayout>
```



Data Binding

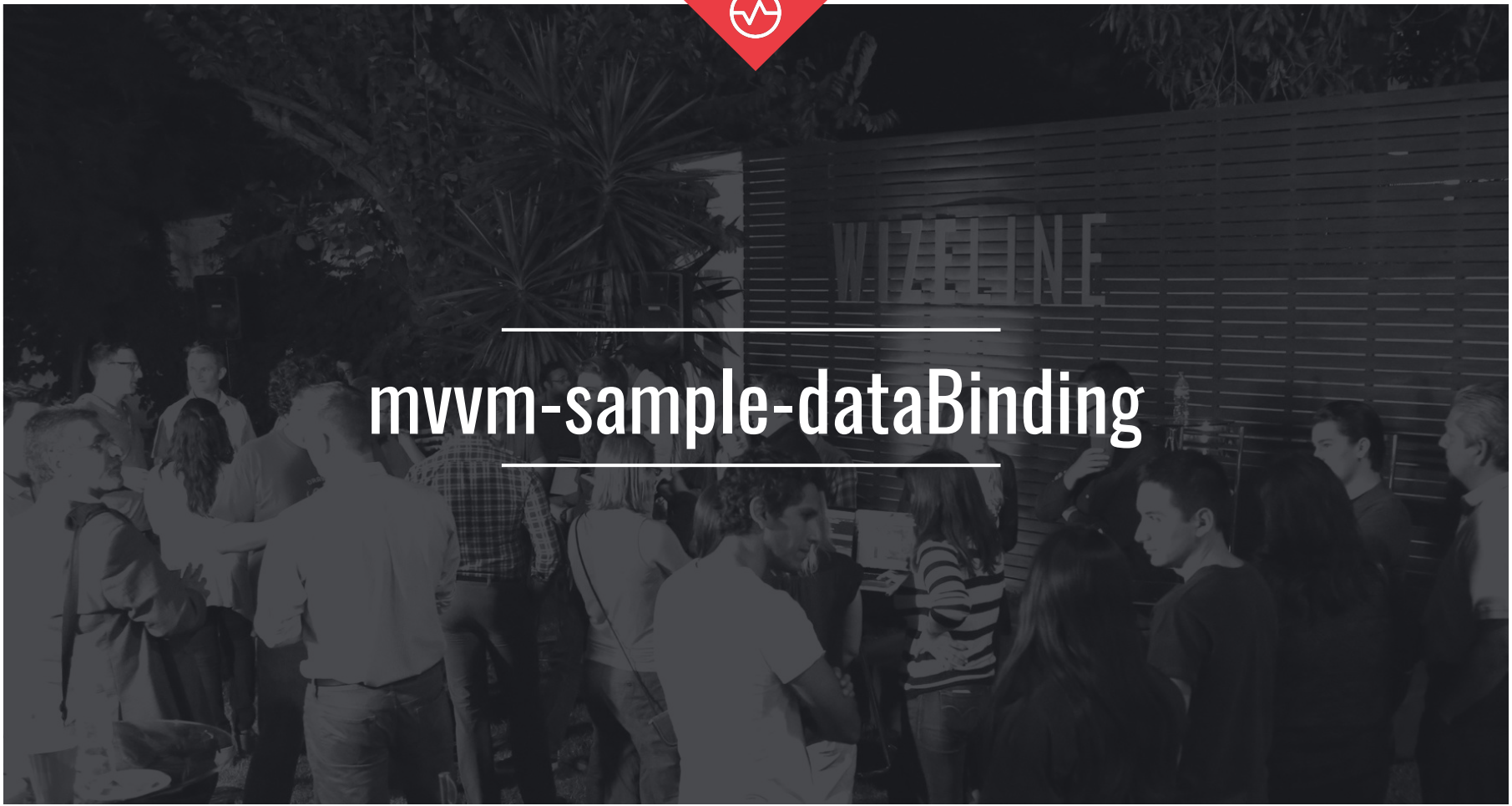
- Android library that simplifies the View-ViewModel binding.

build.gradle

```
android {

  ...

  dataBinding {
     enabled true
  }
}
```



mvvm-sample-dataBinding


CODE TIME!

Project


View
- List View
- List Item View
ViewModel
- List VM
- List Item VM
Model
- Conversion

Pros

- Simple
- Modular
- Easy testing
- Code recycling
- Easy UI update

Cons

- Ties the app to android libraries (data binding)
- Ties views to ViewModels


THANK
YOU
WIZELINE®