

RoboGuice

Basics

```
@Inject  
private MyDataStore store;
```

```
private MyDataStore store = new MyDataStore();
```

Modules and bindings

@Inject

```
public MyDataStore(IBackend backend);
```

```
public class MyModule extends AbstractModule {  
    protected void configure() {  
        bind(IBackend.class).to(SqlBackend.class);  
    }  
}
```

/res/values/roboguice.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<resources>  
    <string-array name="roboguice_modules">  
        <item>org.example.MyModule</item>  
    </string-array>  
</resources>
```

Providers

```
public class MyProvider implements Provider<ComplexObject> {  
  
    @Inject  
    private Context context;  
  
    public ComplexObject get() {  
        // do some complex logic to create our object  
        return complexObject;  
    }  
}
```

```
bind(IObject.class).toProvider(MyProvider.class);
```

RoboGuice defaults

```
@InjectView(R.id.layout)
private LinearLayout myLayout;

// NO MORE

myLayout = (LinearLayout)
findViewById(R.id.layout);
```

Instrumentation tests example

```
public void setUp() {  
    Application app = (Application)  
    getInstrumentation().getTargetContext().getApplicationContext();  
  
    Module comboModule = Module.override(origModule, testModule);  
  
    RoboGuice.setBaseApplicationInjector(app, RoboGuice.DEFAULT_STAGE,  
    RoboGuice.newDefaultRoboModule(app), comboModule);  
}
```

Robolectric example

```
Public void setUp() {  
    RoboGuice.setBaseApplicationInjector(Robolectric.application,  
    RoboGuice.DEFAULT_STAGE,  
    RoboGuice.newDefaultRoboModule(Robolectric.application),  
    Modules.override(new MyModule()).with(new TestModule()));  
}
```

Questions?

Links

- RoboGuice: <https://github.com/roboguice/roboguice>
- Guice: <https://code.google.com/p/google-guice/>
- Great talk by creator of RoboGuice:
<https://www.youtube.com/watch?v=6wU2vwwiwPE>