

RoboGuice ve Android Annotations ile Verimli Android Uygulama Geliştirme

Emrah Ayanoglu, Onur Özcan, Alperen Eraslan

STM A.Ş., Ankara

eayanoglu@stm.com.tr, oozcan@stm.com.tr, aeraslan@stm.com.tr

Özet: Günümüzde mobil teknolojiler artık bilişim teknolojilerinin her alanında kullanılmaya başlanmıştır. Android, mobil işletim sistemleri arasında, şu anda en çok tercih edilen ve ilgili araştırma raporlarında en çok paya sahip olması planlanan açık kaynak kodlu mobil işletim sistemidir. Android üzerinde yazılım geliştirmenin, tabiki diğer yazılım geliştirme süreçlerinden herhangi bir farkı olmamaktadır. Aynı yazılım geliştirme sorunlarıyla, Android uygulama geliştirirken de karşılaşmak mümkündür. RoboGuice ve Android Annotations projeleri, Android ile yazılım geliştirirken ortaya çıkan sorunların bir kısmını çözmek için amaçlanmıştır. RoboGuice aslında yazılım geliştirme süreçlerinde oldukça kullandığımız “dependency injection” ilkesine dayanmaktadır. Android Annotations projesi ise, Java'nın Annotations özelliğini kullanarak daha az sürede kod yazmamızı ve kod bütünlüğünü korumamızı sağlamaktadır. Bildirimiz içerisinde ilgili projeler anlatılarak, beraber kullanılması amaçlanarak maliyet etkin ve kod etkin Android yazılım geliştirme sağlanacaktır.

Anahtar Sözcükler: *Android, RoboGuice, AndroidAnnotations, Mobil Yazılım*

Abstract: Today, mobile technologies are now being used in all aspects of ICT. Android, an open-source mobile operating system, is currently the most preferred and is expected to have the most shares in relevant research reports between the mobile operating systems. Software development on Android, of course, does not have any difference in other software development processes. At the same software development problems, is possible to encounter in developing Android applications. RoboGuice and Android Annotations projects are intended to solve some of the problems, emerging during software development on Android. RoboGuice is based on the principle, "dependency injection" that is actually quite used during software development process. In addition, Android Annotations project ensures that we write more code in less time and maintain the integrity of the code using Java annotations. In this paper, we explain the related projects with the goal of using the both of them to have cost-effective and efficient code Android software development.

1. RoboGuice

Bir Android uygulaması geliştirirken aslında iş sınıfının dışında yazdığımız birçok satır kod bulunmaktadır. Bu bütün Android uygulamalarında ortaktır. Her Android projesine başlarken bu kodları yazmak vakit kaybına yol açmaktadır. Ayrıca bu durum, sadece android geliştirirken geçerli olmayan başka bir dezavantaja da yol açmaktadır; bir uygulama geliştirirken ne kadar fazla kod yazarsak, hatayla karşılaşma oranı o kadar artmaktadır. Bu noktada Google'ın “Guice” framework'ünün android geliştiricileri için geliştirilmiş “RoboGuice” framework'ü Android uygulaması geliştirirken büyük kolaylıklar sağlamaktadır [1].

Roboguice Nedir? Nasıl Çalışır?

RoboGuice'in temel çalışma mantığı “dependency injection” üzerinedir. Aslında uygulama geliştirirken kullanmak istediğimiz nesnelere daha kısa ve kontrollü kod parçalarıyla RoboGuice'in kendisi yaratmaktadır. Roboguice'in avantajları nelerdir?

- Projedeki nesnelere yaratılmasında kullanılan karışık kod parçalarıyla uğraşmak zorunda bırakmaması, nesnelere yaratılırken kullanılan kodun yarı yarıya azalması
- Değişkenleri doğru zamanlarda yaratması, geliştiricinin kod yazma dışındaki sorumluluğunu azaltması
- Geliştiriciye “Null check” gibi konularda yardımcı olması, basit hataları azaltması
- Bazı durumlarda uygulamayı çalıştırmadan ortaya çıkması imkansız olan hataları,

başlangıçta vermesi, yani “TypeSafe” olması.

- Performans bakımından normal Android geliştirmeyle bir farkı olmaması
- Kod okunabilirliğini ve takip etmeyi kolaylaştırması
- Tekrar kullanılabilirliği artırması

Bu avantajlar uygulama geliştirirken yazılımcının kendisiyle ilgili yerlere odaklanmasını kolaylaştırmaktadır ve zamandan tasarruf sağlamaya yardımcı olmaktadır. “RoboGuice”in sağladığı nesne kullanımı yapısı basitçe aşağıdaki gibidir.

Bu örnek kod “textView” isimli “TextView” nesnesini “RoboGuice” in enjekte etmesini sağlamaktadır.

```
@InjectView(R.id.text1) TextView textView;
```

TextView nesnesini “Roboguice” olmadan oluşturmaya çalışsaydık yazacağımız kod şu şekilde olacaktı:

```
TextView textView;  
textView = (TextView)  
findViewById(R.id.textView);
```

Görüldüğü gibi “RoboGuice”, “casting”de yapacağımız olası bir hatayı önlemekte ve yazdığımız kod miktarını azaltarak kodu daha anlaşılabilir hale getirmektedir.

2. Android Annotations

Android Annotations (AA) Android’de kod geliştirmeyi hızlandıran açık kaynaklı bir framework’tür [2]. Bu framework’ün amacı, Android uygulamalarını geliştirirken kod yazmayı ve kodun sonradan değiştirilebilmesini kolaylaştırmaktır. Java Annotation’ların kullanımıyla projede sık sık kullanılmış kod parçalarının sayısını azaltmaya yardımcı olmaktadır. Annotation, Java programa dilinde kullanılan bir kavramdır, Java kaynak koduna eklenebilen sözdizimsel metadata’nın özel bir şeklidir [3].

Özellikler:

- Bağımlılık Enjeksiyonu: Görünümler, kaynaklar, sistem servisleri, vb. enjeksiyonu için benzer annotation’lar sağlar.
- Basitleştirilmiş iş parçacığı modeli: Kullanıcı arayüzü ya da arka plan iş parçacığında çalıştırabilmek üzere kendi metodlarınızı annotate etmenize olanak sağlar.
- Olay Bağlama: Anonim dinleyici sınıflar olmadan, görünümler üzerinde olan olayları idare etmek için metodlarınızı annotate etmenize olanak sağlar.
- REST istemci: Bir istemci arabirimi oluşturabilirsiniz, AA uygulamayı oluşturur.
- Bütün bu ve diğer ekstra özelliklerini 50 kb’dan daha az gereksinim duyar. AA ortak kodu derleme zamanında üretir ve yazılımın çalışmasında bir etkisi yoktur. Her aktivitenin birer alt sınıfını üreterek ve annotation’ları standart ortak kod ile değiştirerek yapar bunu.

3. Roboguice ile Android Annotations’i entegre olarak kullanmak

Yukarıda anlatmış olduğumuz RoboGuice ve Android Annotations’ı beraber kullanmak ile hem kod kalitesini en mükemmelere doğru çevirmeye hem de en az satır kod yazarak istediklerimizi maliyet ve zaman etkin olarak yapabilmeyi elde edebiliriz. Android Annotations projesi RoboGuice ile tamamen entegrasyonu sağlanabilecek durumdadır [4].

Her ikisinin de ortak kullandığımızda bize getireceği avantajları aşağıdakilerle sıralayabiliriz:

- Android bileşenlerin derleme zamanı enjeksiyonu ile zaman etkin çözümü
- Android Annotations sayesinde RoboGuice için kullanmamız gereken RoboActivity’e artık ihtiyacımız kalmayacak
- Bütün Android Annotations maddelerine destek vermektedir. (@Click, @Background, @UiThread, @UiThreadDelayed, @ItemSelected, v.b)

Peki kodlar üzerinde görmek istediğimiz zaman bize ne kadar fayda sağladığını çok açık olarak görebiliriz:

```

@EActivity(R.layout.main)
@RoboGuice({AstroListener.class,
AnotherListener.class})
public class AstroGirl extends Activity {
    @ViewById
    EditText edit;
    @Inject
    GreetingService greetingService;
    @Click
    void button() {
        String name =
edit.getText().toString();

        greetingService.greet(name);
    }
}

public class AstroListener {
    @Inject
    Context context;
    public void
doSomethingOnResume(@Observes
OnResumeEvent onResume) {
        Toast.makeText(context, "Activity
has been resumed",
Toast.LENGTH_LONG).show();
    }
}

```

Yukarıda görüldüğü üzere, toplam kod satırımız yaklaşık olarak %25 azalmıştır. Böylece her iki bileşeni de kullandığımızda, bize hem zaman maliyetinden hem de kalite maliyetinden kazanç sağlamış olacaktır.

4. Sonuç

Android yazılım geliştirme süreçlerinde de, diğer yazılım süreçlerinde kullandığımız mimarilerin ve yapıların kullanılması hem zaman etkin hem de kod etkin olarak geliştirme olanaklarına sahip olmaktadır. Burada anlatmış olduğumuz, RoboGuice ve Android Annotations projeleri bize Android geliştirme süreçlerinde verimli olmamızı sağlayacaktır.

Kaynaklar

- [1] Michael Burton. (2011). *RoboGuice*. <http://code.google.com/p/roboGuice/>. Son Erişim tarihi: 15 Eylül 2012.
- [2] Excilys. (2011). *Android Annotations*. <http://androidannotations.org/>. Son Erişim tarihi: 15 Eylül 2012.
- [3] Java Annotation. http://en.wikipedia.org/wiki/Java_annotation. Son Erişim tarihi: 15 Eylül 2012.
- [4] Excilys. (2012). *RoboGuice Integration*. <https://github.com/excilys/androidannotations/wiki/RoboGuiceIntegration>. Son Erişim tarihi: 15 Eylül 2012.