



بسم الله الرحمن الرحيم

## پروژه تست تیپسل برای کار با چارچوب Spring

۱۶ بهمن ۱۳۹۷

### ۱ شرح پروژه

هدف از این پروژه سنجش میزان آشنایی و توانایی داوطلب برای کار با چارچوب اسپرینگ<sup>۱</sup> برای کد نویسی با زبان کاتلین<sup>۲</sup> و جاوا<sup>۳</sup> است. سیستمی تبلیغاتی را در نظر بگیرید که در آن اطلاعات آماری اپلیکیشن‌ها در جدولی مطابق زیر نگهداری می‌شود و هر اپلیکیشن را بر اساس شناسه<sup>۴</sup> آن می‌شناسیم.

Listing 1: App Statistic Fields

```
class AppStatistics(  
    val id: String,  
    val reportTime: Date,  
    val type: Int,  
    val videoRequests: Int,  
    val webViewRequests: Int,
```

<sup>1</sup>Spring Framework

<sup>2</sup>Kotlin

<sup>3</sup>Java

<sup>4</sup>Id

```
val videoClicks: Int,  
val webViewClicks: Int,  
val videoInstalls: Int,  
val webViewInstalls: Int)
```

می‌خواهیم سرویسی داشته باشیم که اطلاعات کلی اپلیکیشن‌ها را بر اساس type‌های مختلف به صورت هفتگی داشته باشیم. این اطلاعات برای یک بازه مشخص درخواست می‌شود و هفته‌ها باید براساس تقویم ایرانی باشند. فراخوانی سرویس به صورت زیر خواهد بود:

#### Listing 2: Service Template

```
fun getStats(startDate: Date, endDate: Date, type: Int): AppStatisticsListResponse
```

و خروجی موردنظر به صورت زیر خواهد بود:

#### Listing 3: Response Fields

```
class AppStatisticsListResponse(  
    val stats: List<AppStatisticsModel>)
```

#### Listing 4: App Statistics Model Fields

```
class AppStatisticsModel(  
    val weekNum: Int,  
    val year: Int,  
    val requests: Int,  
    val clicks: Int,  
    val installs: Int)
```

توجه داشته باشید که مقادیر requests، clicks و installs در خروجی مجموع آن‌ها برای دو نوع video و webView خواهد بود. سال شمسی است و شماره هفته برای هر سال است که هر سال ۵۳ هفته دارد. داده خروجی باید براساس سال و شماره هفته مرتب شده باشد. فرض بر این است که داده‌ها از اول فروردین سال ۹۵ در پایگاه داده وجود دارد.

**نکته:** موارد زیر باید در پیاده‌سازی سرویس در نظر گرفته شود:

- استفاده از زبان کاتلین و در بستر spring
- استفاده از Maven
- استفاده از spring data
- استفاده از پایگاه داده Mongo

- استفاده از caching در spring
- استفاده از redis برای کش کردن.
- داده‌ها باید در سمت سرور به مدت ۲۰ دقیقه کش شوند. کلید کش ترکیبی از startDate، endDate و type است.
- تمیزی و خوانایی کد دارای اهمیت زیادی است. لطفا توضیحات مناسب نیز اضافه گردد.