



ClassIndex

A quicker alternative to classpath scanning

Sławek Piotrowski

What is an annotation?

```
@Color("Blue")  
class Sky {  
}
```

```
Color color =  
    Sky.class.getAnnotation(Color.class);  
  
assertEquals("Blue", color.value());
```

List of classes

```
@Important  
class Sky {  
}
```

How to get a list of @Important classes?

List of classes

```
@Important  
class Sky {  
}
```

How to get a list of @Important classes?

There is no API in Java for that!

Why?

```
class ClassLoader {  
...  
protected Class<?> findClass(String name)  
    throws ClassNotFoundException  
...  
}
```

Ugly hack

```
((UrlClassLoader) classloader).getURLs();
```

**Returns the list of URLs to scan
==> Classpath Scanning**

Ugly hack

```
((UrlClassLoader) classloader).getURLs();
```

**Returns the list of URLs to scan
==> Classpath Scanning**

**Scannotation, Reflections, Corn CPS,
Spring, JavaEE**

Reflections

```
new Reflections()  
  .getTypesAnnotatedWith(Important.class)
```


Classpath Scanning

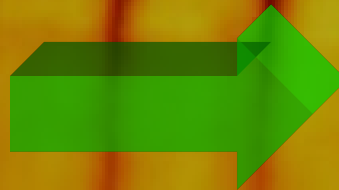
How long it takes?

ClassIndex

Annotation Processor



Index files



Java API

Annotation Processor

Compiler plugin which can:

- analyse syntax tree of the source files**
- analyse resource files**
- generate new source code files**
- generate new resource files**
- generate error messages**
- generate completions**

Annotation Processor

Automatic discovery (ServiceLoader)

```
<dependency>  
  <groupId>org.atteo.classindex</groupId>  
  <artifactId>classindex</artifactId>  
  <version>3.1</version>  
</dependency>
```

Except in Eclipse... [link](#)

ClassIndex usage

```
@IndexAnnotated  
public @interface Important {  
}
```

ClassIndex usage

```
@IndexAnnotated  
public @interface Important {  
}
```

```
Iterable<Class<?>> annotated =  
ClassIndex.getAnnotated(Important.class)
```

ClassIndex usage

```
@IndexSubclasses  
public interface Service {  
}
```

```
Iterable<Class<?>> subclasses =  
    ClassIndex.getSubclasses(Service.class)
```


ClassIndex usage

In package-info.java:

```
@IndexSubclasses  
package org.atteo.test;
```

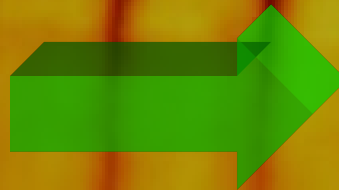
```
Iterable<Class<?>> packageClasses =  
    ClassIndex.getPackageClasses("org.atteo.test")
```


ClassIndex

Annotation Processor



Index files



Java API

ClassFilter

```
ClassFilter.only()  
  .topLevel()  
  .withModifiers(Modifier.PUBLIC)  
  .from(ClassIndex.getAnnotated(Important.class));
```

More filters: [link](#)

Javadoc Storage

```
@IndexAnnotated(storeJavadoc = true)
public @interface Important {
}

/**
 * This is car.
 */
@Important
public class Car {

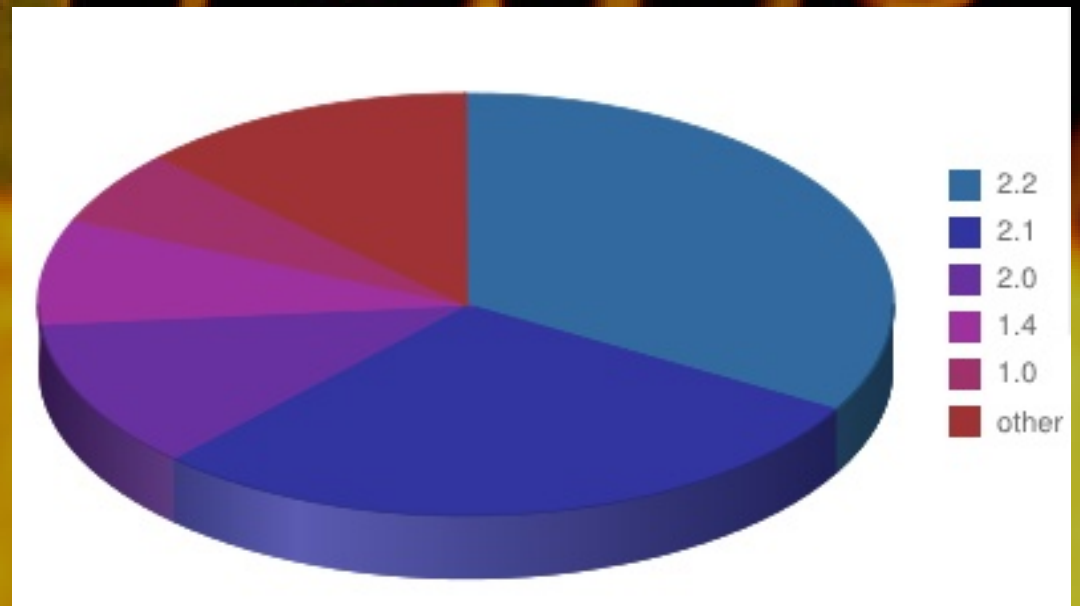
    assertEquals("This is car",
        ClassIndex.getClassSummary(Car.class));
}
```

Alternatives?

- **ServiceLoader**
- **Reflections with Maven plugin**
- **Custom annotation processor**

Usage

name	count ▼	percent
2.2	29	33%
2.1	25	29%
2.0	10	11%
1.4	7	8%
1.0	5	6%
0.6	4	5%
1.1	3	3%
0.5	2	2%
1.2	1	1%
1.3	1	1%



What's more

- **Transformer for shader jars**
- **Moonshine**
- **Google Guice based container**
- **uses ClassIndex for quick startup**

Links

- <https://github.com/atteo/classindex>
- <https://github.com/atteo/moonshine>
- <https://github.com/atteo/classindex-benchmark>
- <https://code.google.com/p/reflections/>