

Java Agent en Action

Qui sommes nous ?

Yoann Buch & Yiquan Zhou

→ 2 développeurs

→ Lancent une entreprise autour d'outils de développement

<http://findtheflow.io>

→ Focalisent sur la visualisation et analyse des exécutions de programmes

→ Utilisent un java agent

Java Agents

YourKit

New Relic

VisualVM

DripStat

JProfiler

Dynatrace

XRebel

AppDynamics

Cobertura

Takipi

Jacoco

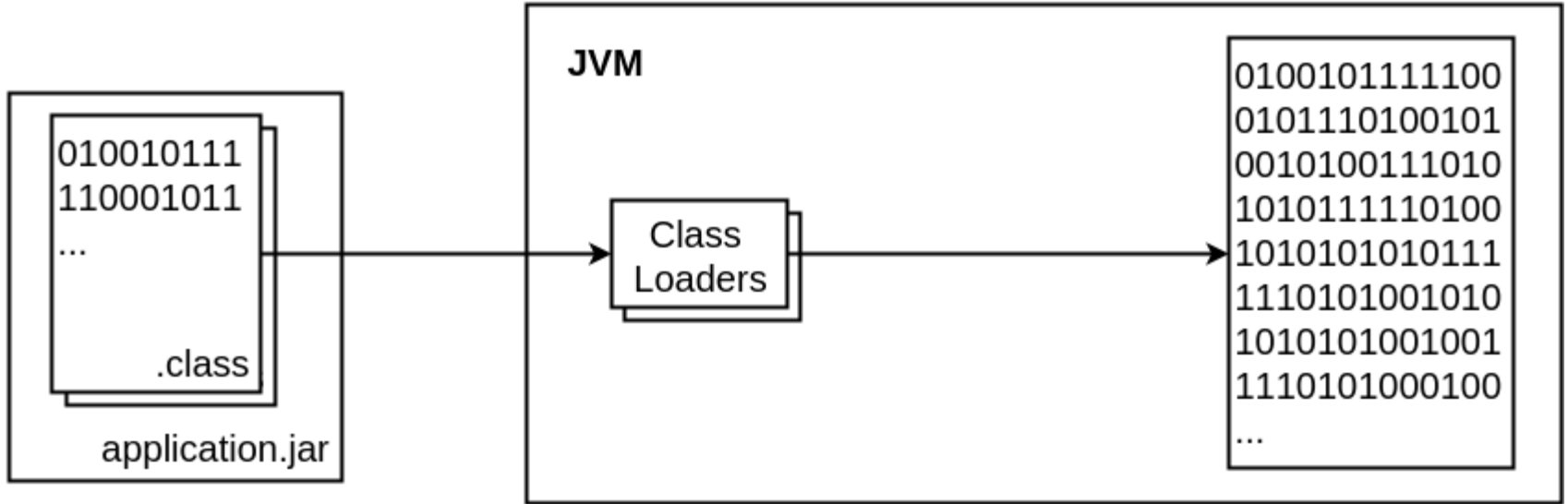
Plumbr

Chronon

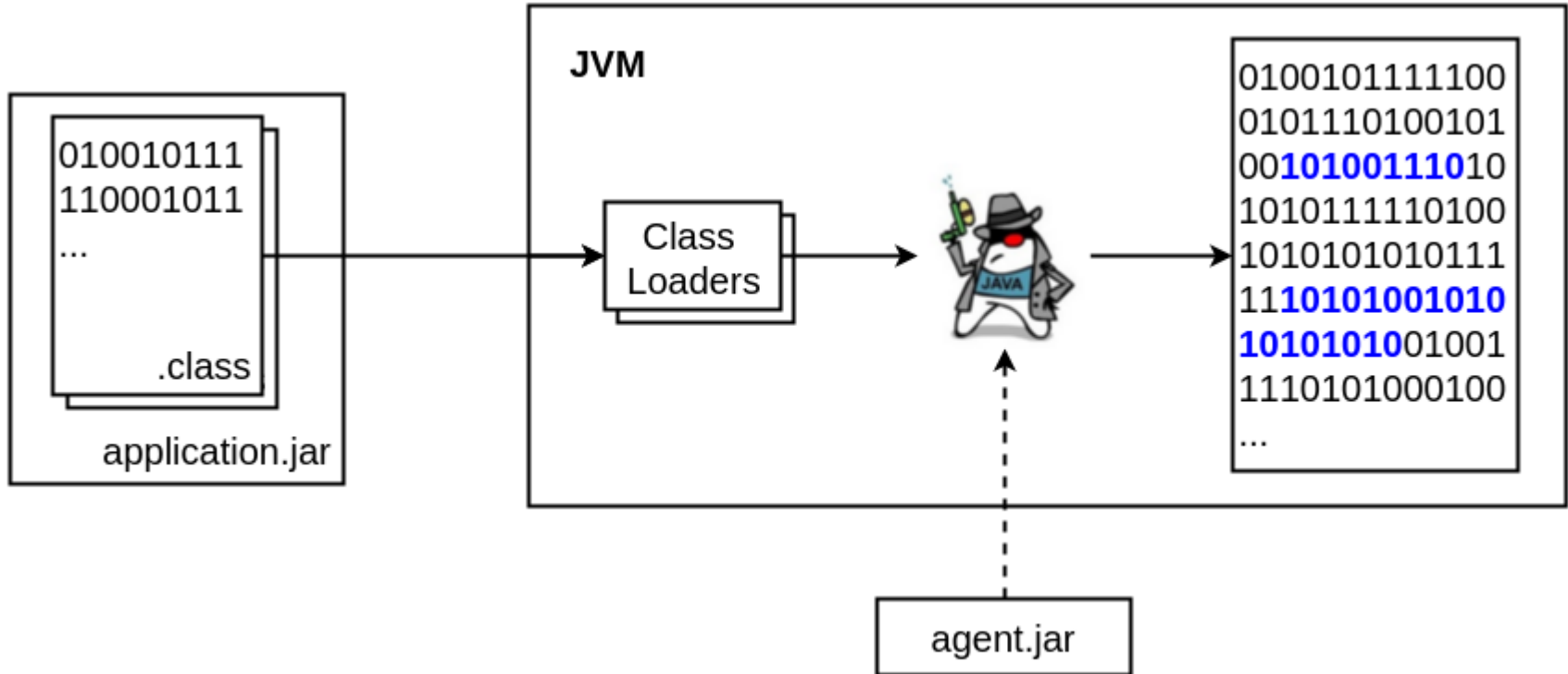
JRebel

Le votre ?

```
> java -jar application.jar
```



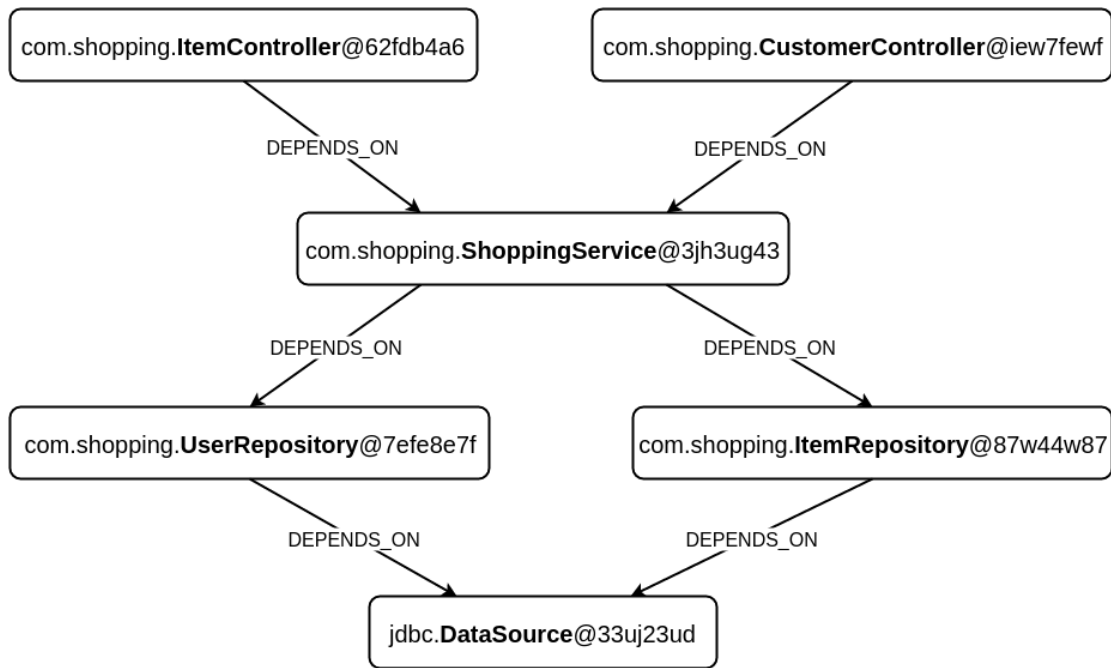
```
> java -jar -javaagent:agent.jar application.jar
```



Mission : créer un outil de développement

Visualiser des conteneurs
d'injection de dépendances

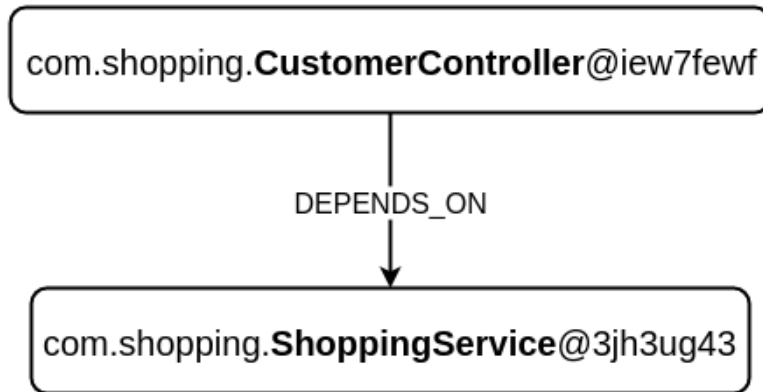
- Spring
- CDI
- Dagger
- Guice



Par où commencer ?

```
class CustomerController {  
    @Inject  
    public CustomerController(  
        ShoppingService shoppingService)  
    {  
        ...  
    }  
}
```

```
class CustomerController {  
    @Inject  
    public CustomerController(  
        ShoppingService shoppingService) {  
        // DependencyGraph.addDependency(this,  
        shoppingService);  
        ...  
    }  
}
```



Démo

Récap

- Très simple de créer un java agent
- Instrumentation du byte code simplifiée grâce à ASM, Byte Buddy, etc.
- Pas limité qu'au profiling
- Pas limité qu'à l'instrumentation du byte code
- Non-intrusif

http://findtheflow.io

The screenshot displays the FindTheFlow IDE interface. At the top, there is a header with 'call 113 / 279', a progress bar, and buttons for 'Record', 'Stop', 'Settings', and a search bar. Below the header, a class hierarchy diagram is shown, starting with 'AbstractClinicServiceTests' at the top. The hierarchy includes several classes such as 'ClinicServiceImpl', 'JdbcPetRepositoryImpl', 'JdbcOwnerRepositoryImpl', 'JdbcPetRepositoryImpl', 'JdbcVisitRepositoryImpl', 'JdbcVisitRepositoryImpl1', 'BaseEntity', 'Owner', 'Pet', 'NamedEntity', 'JdbcPet', 'EntityUtils', and 'BaseEntity'. A 'Package' dropdown menu is visible on the left, showing 'Type' and 'Method' options. Below the class hierarchy, a call stack is displayed, showing the execution flow through various methods like 'findPetTypes', 'findByPetId', 'getPetTypes', 'mapRow', and 'loadPetsAndVisits'. The call stack is color-coded, with green for the current call and orange for other calls. A green waveform is visible above the call stack, representing the execution timeline.

Java Agent - Hello World

```
public class DIAgent {  
    public static void premain(  
String args,  
Instrumentation instrumentation) {  
        System.out.println("Hello World from Agent");  
    }  
}
```


Java Agent - Instrumentation du byte code

```
instrumentation.addTransformer(new ClassFileTransformer() {  
  
    @Override  
    public byte[] transform(  
        ClassLoader loader,  
        String className,  
        Class<?> classBeingRedefined,  
        ProtectionDomain protectionDomain,  
        byte[] classfileBuffer) throws IllegalsClassFormatException {  
  
        // Return modified or unchanged classfileBuffer  
        return classfileBuffer;  
    }  
});
```

Java Agent - Byte Buddy



```
new AgentBuilder.Default().type(getTypeMatchers()).transform(new Transformer() {
```

```
    @Override
```

```
    public Builder<?> transform(Builder<?> builder, TypeDescription typeDescription) {
```

```
        return builder
```

```
            .invokable(isAnnotatedWith(named(INJECT_ANNOTATION))
                .or(isAnnotatedWith(named(AUTOWI
```

```
RED_ANNOTATION))))))
```

```
            .intercept(SuperMethodCall.INSTANCE
```

```
                .andThen(MethodDelegation.to(Inject
```

```
ionInterceptor.class)))));
```

```
    }
```

```
}).installOn(instrumentation);
```

Java Agent - Byte Buddy



```
public class InjectionInterceptor {  
  
    public static void intercept(  
        @This Object currentInstance,  
        @AllArguments Object[] injectedInstances) {  
  
        DependencyGraphService.addDependency(currentInstance,  
injectedInstances);  
    }  
}
```

Java Agent - Byte Buddy



```
Console Properties Problems
shopping - ShoppingApplication [Spring Boot App] /usr/lib/jvm/java-8-oracle/bin/java (16 Dec 2015 3:53:45 pm)
Hello World from Agent

  ____
 /  __ \
| |  | |
| |__| |
|  __ /
| |  | |
|_|  |_|
:: Spring Boot ::      (v1.3.0.RELEASE)

2015-12-16 15:53:49.314 INFO 1713 --- [main] com.shopping.ShoppingApplication : St
2015-12-16 15:53:49.325 INFO 1713 --- [main] com.shopping.ShoppingApplication : Nc
2015-12-16 15:53:49.587 INFO 1713 --- [main] ationConfigEmbeddedWebApplicationContext : Re
2015-12-16 15:53:51.379 INFO 1713 --- [main] o.s.b.f.s.DefaultListableBeanFactory : Ov
2015-12-16 15:53:51.992 INFO 1713 --- [main] f.a.AutowiredAnnotationBeanPostProcessor : JS
2015-12-16 15:53:56.806 INFO 1713 --- [main] s.b.c.e.t.TomcatEmbeddedServletContainer : Tc
2015-12-16 15:53:56.844 INFO 1713 --- [main] o.apache.catalina.core.StandardService : St
2015-12-16 15:53:56.847 INFO 1713 --- [main] org.apache.catalina.core.StandardEngine : St
2015-12-16 15:53:57.111 INFO 1713 --- [ost-startStop-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Ir
2015-12-16 15:53:57.111 INFO 1713 --- [ost-startStop-1] o.s.web.context.ContextLoader : Rc
2015-12-16 15:53:57.770 INFO 1713 --- [ost-startStop-1] o.s.b.c.e.ServletRegistrationBean : Me
2015-12-16 15:53:57.783 INFO 1713 --- [ost-startStop-1] o.s.b.c.embedded.FilterRegistrationBean : Me
2015-12-16 15:53:57.784 INFO 1713 --- [ost-startStop-1] o.s.b.c.embedded.FilterRegistrationBean : Me
2015-12-16 15:53:57.784 INFO 1713 --- [ost-startStop-1] o.s.b.c.embedded.FilterRegistrationBean : Me
2015-12-16 15:53:57.784 INFO 1713 --- [ost-startStop-1] o.s.b.c.embedded.FilterRegistrationBean : Me

===== Dependency Injection Agent =====
com.shopping.services.ShoppingServiceImpl@79169472 depends on [com.shopping.repositories.ItemRepositor

===== Dependency Injection Agent =====
com.shopping.controllers.ItemController@63d43a5 depends on [com.shopping.services.ShoppingServiceImpl@

===== Dependency Injection Agent =====
com.shopping.controllers.CustomerController@2a49753 depends on [com.shopping.services.ShoppingServiceI

2015-12-16 15:53:58.237 INFO 1713 --- [main] s.w.s.m.m.a.RequestMappingHandlerAdapter : LC
2015-12-16 15:53:58.590 INFO 1713 --- [main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mz
```


Java Agent - Serveur Web embarque

```
public static void premain(String args, Instrumentation instrumentation) {  
  
    System.out.println("Hello World from Agent");  
    startWebServer();  
    transformClasses(instrumentation);  
}
```

