

Personalització de la Instrumentació de Canigó 3

A qui va dirigit

Aquest Howto va dirigit a tots aquells perfils tècnics encarregats d'auditar una aplicació Canigó3 per tal de determinar els diferents punts d'aquesta on pugui presentar. Es pressuposen uns coneixements mínims d'Spring3.

Versió de Canigó

Les versions de Canigó sobre les quals es pot aplicar aquest procediment son totes les 3.x.

Introducció

Amb el mòdul d'instrumentació de Canigó 3 ve per defecte una anotació d'instrumentació anomenada @Trace. Aquesta anotació, posada davant de la signatura de qualsevol mètode de l'aplicació escriu una traça en el log amb informació sobre l'estat de finalització del mètode i els temps que ha trigat en fer-ho.

L'objectiu d'aquest howto es mostrar com modificar l'anotació @Trace per personalitzar la informació que escriu en el log, donant capacitat a les aplicacions d'instrumentar moltes mes dades.

Instal·lació del mòdul d'Instrumentació

Pas previ per poder utilitzar les anotacions del mòdul d'instrumentació es la seva instal·lació. Per poder fer-ho es pot modificar l'aplicació manualment o bé utilitzar la tool del Canigó3.

Instal·lació amb la tool

- 1.- Obrir l'eclipse de l'entorn de treball de Canigó3.
- 2.- Clicar butò dret sobre el projecte Canigó3 on es vulgui afegir el mòdul d'instrumentació.
- 3.- Seleccionar l'opció "Canigó -> Afegir Nous Mòduls".
- 4.- Seleccionar el sub-mòdul d'Instrumentació del mòdul d'Operacions

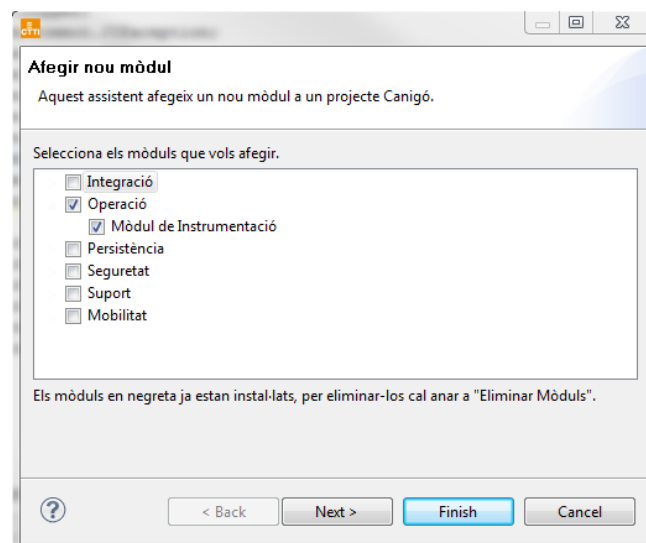


Figura 1 – Mòdul d'Instrumentació de la tool

Personalització de la Instrumentació de Canigó 3

5.- Prémer "Finish".

Per defecte s'obrirà l'arxiu instrumentation.properties del qual, i de cara a modificar el Trace, no fa falta canviar res.

Instal·lació Manual

- 1.- Obrir l'eclipse de l'entorn de treball de Canigó 3.
- 2.- Afegir al pom.xml del projecte de Canigó 3 el següent fragment:

```
<dependency>  
  <groupId>cat.gencat.ctti</groupId>  
  <artifactId>canigo.operation.instrumentation</artifactId>  
  <version>[1.0.0,1.1.0)</version>  
</dependency>
```

Ús del Trace

Una vegada instal·lat el mòdul d'instrumentació ja es pot utilitzar l' anotació @Trace en els diferents mètodes de l'aplicació.

La forma habitual d'utilització es posar @Trace davant de la signatura del mètode a instrumentar. Per exemple;

```
@Repository("myService")  
public class ServiceImpl implements Service {  
  
    @Trace  
    public void doSomething() {  
        .....  
        .....  
    }  
}
```

En finalitzar el mètode, en el log apareixerà una traça semblant a aquesta:

```
canigo Message: XX XXX 2010 16:45:40,774 INFO [main]  
cat.gencat.ctti.canigo.arch.operation.instrumentation.trace.AspectTrace - ea84cbe0-60ff-4ba5-a4e8-  
d90465101ac3;loc>null;234;Service.doSomething();OK;
```

Aquesta informació la mostra per defecte @Trace. El detall dels camps es pot consultar en el portal del [CS Canigó](#).

NOTA: Es molt important tindre en compte que el @Trace només funciona sobre beans d' Spring i per tant només es pot utilitzar @Trace sobre mètodes d'una classe que implementi una interfície i a més estigui informada com un bean d' Spring amb una anotació @Repository, @Component, @Controller i @Service.

Personalització de la Instrumentació de Canigó 3

Modificació del Trace

Per poder editar el contingut de la traça que mostra el Trace s'ha de crear un Aspecte nou. Aquest Aspecte ve determinat per una classe que defineix el JoinPoint i pointCut ()

```
@Aspect
public class AspectTraceDummy implements InitializingBean {

    private static final Log log =
LogFactory.getLog(AspectTraceDummy.class);
    private static final String FIELD_SEPARATOR = ";";

    @Around("@annotation(cat.gencat.ctti.canigo.arch.operation.instrumentati
on.trace.annotation.Trace)")
    public Object profile(ProceedingJoinPoint jp) throws Throwable{
        Stopwatch watch = new Stopwatch();
        String error = "OK";
        watch.start();
        try{
            Object resultat = jp.proceed();
            if(resultat instanceof Instrumentable) {
                Instrumentable objecte = (Instrumentable) resultat;
                if(objecte.getStatus() < 0) {
                    error =
"KO"+FIELD_SEPARATOR+objecte.getMissatgeError();
                }
            }
            return resultat;
        }catch(Throwable e){
            error = "KO";
            throw e;
        }finally{
            watch.stop();
            Object[] arguments = jp.getArgs();
            String uuid = UUID.randomUUID().toString();
            String metodeExecutat = jp.getSignature().toShortString();

            if(metodeExecutat != null &&
(metodeExecutat.indexOf("[nom_interficie]. //
[nom_metode]("") != -1 )) {
                //[Extreure dades del paràmetre "arguments[]"]
            }

            StringBuffer buffer = new StringBuffer();
            buffer.append(uuid).append(FIELD_SEPARATOR);

            buffer.append(watch.getLastTaskTimeMillis()).append(FIELD_SEPARATOR);
//temps de resposta
            //Afegir les dades recuperades del vector de paràmetres del
JoinPoint al buffer

            buffer.append(error).append(FIELD_SEPARATOR);
            log.info(buffer.toString());
        }
    }

    /**
     * Log module loaded
     */
    public void afterPropertiesSet() throws Exception {
        log.info("Instrumentation module loaded");
    }
}
```

Personalització de la Instrumentació de Canigó 3

```
}
```

Com es pot observar les dades del missatge que es mostren en les traces es recuperen del valor dels paràmetres que rep el mètode instrumentat amb @Trace.

En el cas de l'exemple, per cada mètode executat de l'interface [nom_interface] amb nom [nom_metode] @Trace escriurà en el log un uuid, el temps d'execució del mètode i l'estat (OK/KO).

Per poder afegir altres mètodes o dades a recuperar només s'ha de modificar el finally d'aquesta classe.

Finalment per qualsevol dubte o problema que pugui sorgir es pot fer una consulta al CS Canigó a la bústia oficina-tecnica.canigo.ctti@gencat.net o bé mitjançant una petició de suport a l'eina JIRA del CTTI en <http://cstd.ctti.gencat.cat/jiracstd>.