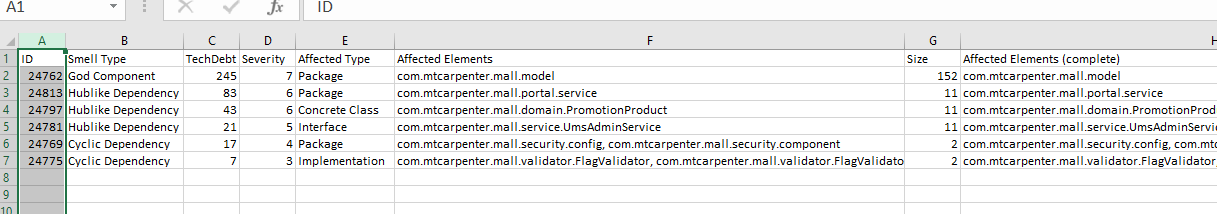
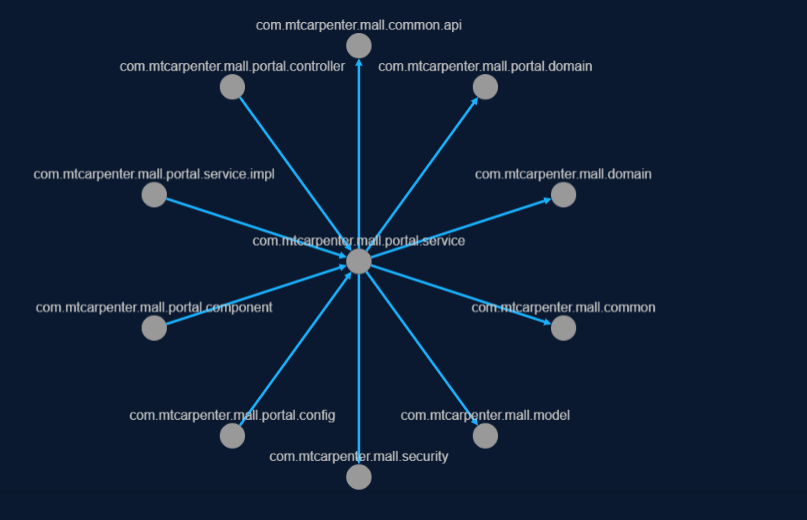
mall-cloud-alibaba

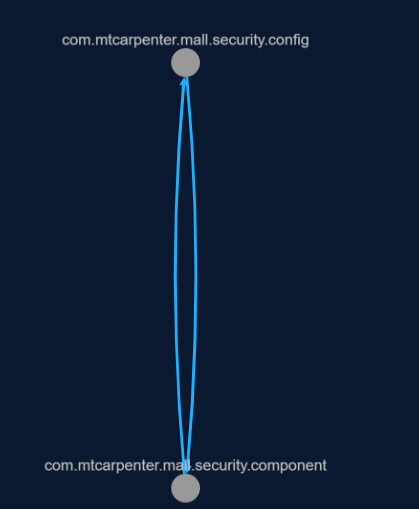
6 smells



com.mtcarpenter.mall.portal.service -Hublike Dependency



com.mtcarpenter.mall.security.config, com.mtcarp – Cyclic dependency



SELF CYCLE NON RIMOVIBILE

È un ciclo “statico” (compile-time) e non genera ricorsioni a runtime: serve a permettere all’annotation di specificare quale classe la esegue e al validator di leggere i parametri dell’annotation stessa. Non c’è modo di “rompere” questo ciclo se vuoi usare l’API standard di Bean Validation.

Anche se “spezzi” la classe in più file, o ne trasformi una parte in helper, i due tipi continueranno a dipendere l’uno dall’altro perché:

* l’annotation deve dichiarare in validatedBy la classe che la esegue;
* il ConstraintValidator deve conoscere il tipo di annotation per leggerne i valori.

Quel ciclo è solo statico (compile-time), non genera ricorsioni a runtime, ed è pienamente supportato dal framework di validazione. Se il tuo strumento di analisi delle dipendenze lo segnala come “ciclo pericoloso”, l’unica vera “soluzione” è escludere da quel controllo le coppie Annotation↔Validator, perché nella Bean-Validation è un pattern necessario.

**FORSE NUOVO MICRO SERVIZIO?????**

Durante il ref 3 ho notato che

mall-portal-order\order-server\src\main\java\com\mtcarpenter\mall\portal\service\impl\OmsPromotionServiceImpl.java

Dipendeva direttamente da un altro modulo:  
mall-portal-product\product-common\src\main\java\com\mtcarpenter\mall\domain\PromotionProduct.java

Ho fatto ref con un DTO ma forse posso creare un micro servizio solo per questo DTO.

Per evitare di dover sincronizzare a mano tutte le copie di PromotionProduct (e di qualunque DTO/domain) nei vari micro-servizi, ti conviene estrarre tutte le classi condivise in un modulo Maven “common” che venga importato dai cinque progetti.

1. Aggiungi un nuovo modulo al tuo parent POM, ad es. **mall-portal-common**:– nel pom.xml del parent, dentro <modules>:
2. mall-portal-common
3. mall-portal-content
4. mall-portal-coupon
5. …

– crea la directory mall-portal-common/ con questo pom.xml minimale:

com.mtcarpenter

mall-portal

0.0.1-SNAPSHOT

4.0.0

mall-portal-common

jar

org.projectlombok

lombok

1.18.20

provided

com.fasterxml.jackson.core

jackson‐annotations

1. Sposta in mall-portal-common/src/main/java/com/mtcarpenter/mall/domain e …/dto tutte le classi condivise:
   * PromotionDetail
   * PromotionProduct
   * CartPromotionItem, OmsCartItem, ecc.
   * Eventuali altri DTO che viaggiano tra portal-product e portal-order/etc.
2. Rimuovi le versioni duplicate da mall-portal-product e dagli altri moduli.
3. In ciascun modulo che ti serve importare questi DTO, aggiungi una dipendenza su **mall-portal-common**.Esempio in mall-portal-product/pom.xml:
4. com.mtcarpenter
5. mall-portal-common
6. ${project.version}

Fai lo stesso in mall-portal-order, mall-portal-member, ecc.

1. Fai un mvn clean install sul parent, e adesso tutti i micro-servizi vedranno **lo stesso** JAR con le classi domain/dto.– Modificando PromotionDetail o PromotionProduct in un unico posto, si aggiorna l’API di tutti i client.– Mantieni versioning semantico per il common: se cambi il formato JSON, fai bump di major, aggiorna le dipendenze client.

In questo modo: • Non duplichi più codice di comunicazione tra servizi. • Riduci drasticamente le rotture “a catena” quando modifichi un DTO. • Mantieni chiara la separazione di responsabilità: ogni micro-servizio ha i suoi controller/logica, ma condivide i contratti via un modulo Maven.