

Tests : Objets factices

Etape 1:

Toutes les questions 1 à 4 ont été bien réalisées suivant les consignes du tp .

Cas de tests pour les méthodes **isAMatch** et **dataStrength** pour les classes:

❖ **isAMatch()**

Pour **isAMatch()** les cas de tests sont les mêmes pour les deux classes.

Ici on teste si les données d'authentification fournies (**authenticationData**) correspondent aux données d'authentification associées à l'identifiant spécifié (**identifiant**) dans l'objet de type **Directory**.

Un cas de test serait un identifiant : **regix** une donnée d'authentification: **azerty**

Comme cela a été fait dans les classes de tests de ces deux classes.

Il faut au préalable que l'identifiant et sa donnée de test soit dans une base de donnée.

Ici on a utilisé des classes factices **MockDirectory** et

MockStrengthEstimationService

❖ **dataStrength()**

Cette méthode évalue la force des données d'authentification fournies (**authenticationData**) en fonction de la longueur de ces données et d'une estimation de leur force calculée à l'aide de la méthode **equivalentBitLength** de **strengthEstimationService**.

• **AuthenticationByPasswordService**

La méthode **equivalentBitLength()** à ce niveau a été redéfini dans mon code en fonction de la taille de **authenticationData** de sorte que si la taille est supérieure ou égale à 20 alors la force est **STRONG**. De même que pour **REGULAR**, **VERY_WEAK** et **WEAK** avec leurs tailles minimales correspondantes.

Un cas de test: tester que “**az**” à bien une force “**VERY_WEAK**”

Vous pouvez vous référer à la classe de test pour les différents tests.

• **AuthenticationByBiometryService**

Cette méthode retourne toujours une force égale à **STRONG** puisqu'on considère qu'une authentification biométrique est toujours forte.







Vous pouvez vous référer à la classe de test pour les différents tests.

Les classes **MockDirectory** et **MockStrengthEstimationService** ont bien été implémentées.

De même que pour les classes de tests.

Voici la couverture obtenu:

interfacemanualmocking

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Ctxty	Missed	Lines	Missed	Methods	Missed	Classes
MockStrengthEstimationService		93 %		80 %	2	7	1	13	0	2	0	1
AuthenticationByPasswordService		100 %		71 %	4	12	0	23	0	5	0	1
MockDirectory		100 %	n/a	n/a	0	2	0	7	0	2	0	1
AuthenticationByBiometryService		100 %	n/a	n/a	0	4	0	9	0	4	0	1
Total	2 of 162	98 %	6 of 24	75 %	6	25	1	52	0	13	0	4

Sachant que le rouge est normal parce qu'il s'agit d'instructions ou de branches qui ne peuvent pas être parcourues du fait par exemple qu'une taille d'un string ne peut pas être inférieure à zéro.

Mes tests couvrent donc de manière très satisfaisante les méthodes des classes **AuthenticationByPasswordService** et **AuthenticationByBiometryService**.