

SAÉ 3.01 Version :	4.0
Document :	Dossier de tests
Date :	16/02/2024
Responsables de la rédaction :	Pierre JAUFFRES, Cyril TILAN

Dossier de tests des fonctions de l’application

1. Introduction

On a réalisé un dossier de tests pour chacune des fonctions des fichiers `Crypto.php` et `functions.php`. Ce dossier de test comporte 6 tableaux, un pour chacune des fonctions testées.

2. Description de la procédure de test

Pour la campagne de tests, nous avons utilisé des tests unitaires en boîte blanche. Les tests unitaires en boîte blanche impliquent la création de tests basés sur la structure interne du code source. Les développeurs écrivent des tests qui visent à couvrir toutes les branches de décision, les boucles et les chemins logiques du code. Ensuite, ces tests sont exécutés sur le code source pour s'assurer qu'il fonctionne correctement selon sa structure interne.

3. Description des informations à enregistrer pour les tests

Campagne de test

- Test de `Crypto.php`

Produit testé :	<code>Crypto.php</code>
Configuration logicielle :	Windows 10 ou 11 ; PhpStorm avec PHP 5.6
Configuration matérielle :	PC sous Windows
Date de début :	16/02/2024
Date de finalisation :	[À compléter]
Tests à appliquer :	Test sur les fonctions cryptographiques du fichier <code>Crypto.php</code>
Responsable de la campagne de test :	Pierre JAUFFRES

Identification du test :	<code>chiffrement_RC4_key</code> Version : 1
Description du test :	<code>chiffrement_RC4_key</code> est une fonction de hachage cryptographique qui prend en paramètre un message, une clé et renvoie une version chiffrée du message.
Ressources requises :	PhpStorm et le module PHPUnit

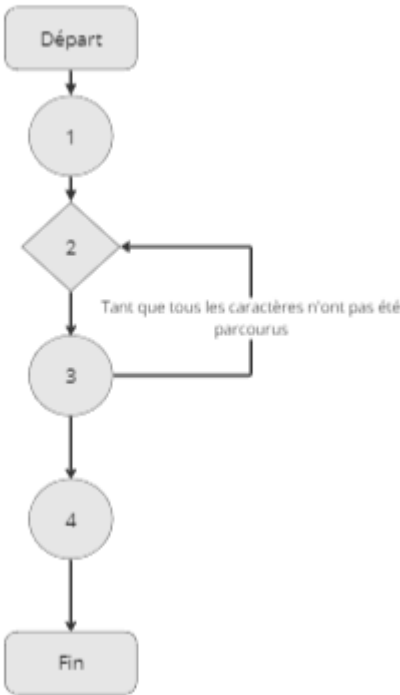
Identification
du test : **chiffrement_RC4_key Version : 1**

Responsable
: Pierre JAUFFRES

Fonction testée :

```
function chiffrement_RC4_key($password, $key) {  
    #1  
    $key_bytes = $key;  
    echo "Clé utilisée : " . $key_bytes . "\n";  
    $password_bytes = $password;  
    $S = KSA($key_bytes);  
    $keystream = PRGA($S, strlen($password_bytes));  
    $encrypted_password = '';  
  
    foreach (str_split($password_bytes) as $index => $char) { #2  
        #3  
        $encrypted_password .= sprintf('%02x', ord($char) ^  
$keystream[$index]);  
    }  
  
    return $encrypted_password; #4  
}
```

chiffrement_RC4_key



Lors de l'étape 1, on utilise les fonctions **KSA** et **PRGA** qui seront traitées dans la suite du dossier de tests.

On remarque que cette fonction n'a qu'un seul chemin possible qui est C1 : {1;2;3;4}. Dans ce cas pour vérifier le fonctionnement de la fonction **chiffrement_RC4_key**, on a juste besoin de réaliser un seul test.

Référence du test appliqué : chiffrement_RC4_key				
Responsable :		[Nom du responsable]		
Date de réalisation du test :		[Date de réalisation]		
Résultat du test :		[OK, KO, non fait, dérogé]		
Chemin	password	key	Résultat attendu	Résultat du test
C1	pedia	wiki	1021BF0420	

Produit testé	functions.php
Configuration logicielle	Windows 10 ou 11 ; PhpStorm avec PHP 5.6
Configuration matérielle	PC sous Windows
Date de début	16/02/2024
Date de finalisation	[à compléter]
Tests à appliquer	test sur les fonctions du fichier functions.php

Responsable de la campagne de test Cyril TILAN

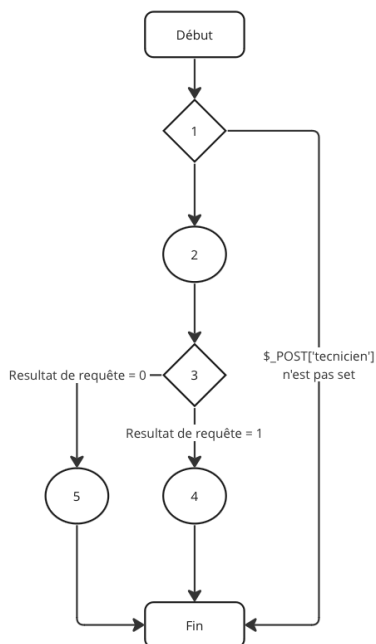
Identification du test	assignTechnicianToTicket
Version	1
Description du test	assignTechnicianToTicket est une fonction qui permet à l'administrateur web d'assigner un ticket à un technicien et changer son état en "Assigné"
Ressources requises	PhpStorm et le module PHPUnit
Responsable	Cyril TILAN

fonction testée :

```

function assignTechnicianToTicket($conn, $ticketId) {
    #1
    if (isset($_POST['technicien'])) {
        #2
        $technicienLogin = $_POST['technicien'];
        $requete_update = "UPDATE tickets SET technicien_login = ?, etat =
'Assigné' WHERE id = ?";
        $reqpre_update = mysqli_prepare($conn, $requete_update);
        mysqli_stmt_bind_param($reqpre_update, "si", $technicienLogin, $ticketId);
        mysqli_stmt_execute($reqpre_update);
        #3
        if (mysqli_stmt_affected_rows($reqpre_update) > 0) {
            #4
            echo "<p>$technicienLogin a été assigné avec succès au ticket.</p>";
            updateTicketStatus($conn, $ticketId);
        } else {
            #5
            echo "<p>Erreur lors de l'assignation du technicien au ticket.</p>";
        }
    }
}

```



On remarque qu'il y a plusieurs chemins possibles :

C1 : {1}

C2 : {1;2;3;4}

C3 : {1;2;3;5}

On aura donc besoin de faire un tableau qui teste ces 3 cas :

Chemin	\$ticket_id	\$_POST['technicien']	Résultat attendu	Résultat du test
C1	31	Ø	Ø (Erreur d'assignation)	
C2	31	cayoux (technicien)	"\$technicienLogin a été assigné avec succès au ticket."	
C3	Ø	cayoux (technicien)	"Erreur lors de l'assignation du technicien au ticket."	

Lors de la phase de test, nous devons créer des tickets temporaires que nous désignerons a un faux technicien pour tester les différentes méthodes.