

FRAMEWORK SQUASH-TA 1.10

Correctifs Framework TA-1.10.1 (25/05/2018)

[Saop UI] Changement de version de la librairie 'commons-codec' embarquée, suite à l'observation de son incompatibilité.

Le build Maven avec paramètre 'list' aboutissait à un « NullPointerException » dans la version 1.10.0-RELEASE. Il fonctionne désormais correctement.

[Integration] Le build du framework Squash TA était en échec dans la nouvelle forge dû aux problèmes de compatibilité entre le compilateur Groovy et la jdk8. Problème résolu.

Améliorations Framework TA-1.10.0 (20/04/2018)

Les améliorations apportées au framework Squash-TA 1.10.0 sont nombreuses. Elles touchent tout à la fois les composants d'interface du framework (*driver layer*) et les plugins Squash-TA.

Driver Layer

- Squash TA maven plugin

Un nouveau paramètre permet d'affiner le seuil d'échec d'un build maven. Selon le paramétrage utilisé, l'échec d'un test TA n'aboutit plus fatalement sur un « build fail ». On a désormais la possibilité de distinguer les erreurs d'ordre fonctionnel (ex : une assertion) et les erreurs techniques. Le build résultera donc du comportement désiré.

- DSL et macro-commandes (shortcuts)

- L'écriture d'une macro appelant d'autres macros est rendue possible.

- {void} devient la syntaxe dédiée et interprétée par le DSL pour « VoidResource ». Utile dans le cas d'un composant ne prenant pas de ressource en entrée, ou ne retournant pas de ressource en sortie.

- Il est désormais possible de charger au sein d'un script de test Squash-TA n'importe quelle ressource packagée dans un plugin (.jar).

- Amélioration des messages d'exécution sur la détection des macros (mention de la provenance des définitions et listing des définitions détectées).

- Amélioration des messages concernant la recherche d'une ressource lorsque celle-ci ne figure pas dans le répertoire cible.

- Reporting

- Ajout d'un rapport HTML simplifié, appelé « HTML summary », plus adaptés aux présentations dédiées à un public non-technique.

- Ajout d'une fonctionnalité optionnelle (opt-in) pour permettre au lien TM/TA de remonter des messages structurés en HTML lorsque certains composants en dépendent. A noter que les messages d'exécution du lien TM/TA ont été améliorés de manière à faciliter le débogage.

Squash TA plugins

- Database Plugin

- Ajout d'un convertisseur de résultat SQL en ressource XML permettant l'extraction en base de données pendant le test.

- Affichage d'un warning en log lorsqu'une 'USING clause' contient une ou plusieurs ressources inutiles.

- Selenium Plugin

- Amélioration du rapport d'erreur permettant un diagnostic plus précis en cas d'exécution de Squash TA sans compilateur java détecté (ex : exécution de Squash TA avec une JRE en lieu et place d'une JDK).

- Possibilité de générer un rapport d'échec à partir du code java d'un test Selenium.

- SoapUI Plugin

Les propriétés d'un cas de test SoapUI peuvent désormais être valorisées depuis un fichier de configuration appelé en 'USING clause' d'un script TA.

- SSH Plugin

Le plugin SSH permet désormais l'authentification par clé.

- Common Components Plugin

- **FileResource** - Ajout de nouvelles fonctions à la formule 'date' utilisée par le convertisseur 'File to File(param.date)' :

- `date({property.key}, <input date format>)` permet de renseigner une date à partir de la valeur d'une propriété.
- La fonction `ensureWorkingDay()` ajuste la date renseignée au jour travaillé le plus proche (avant ou après) .
- `addWorkingDay()` permet d'ajouter ou de soustraire un nombre de jours travaillés à la date d'entrée.

- **USER_MESSAGE** - Ajout d'une macro pour personnaliser les logs d'exécution de test avec le contenu d'un fichier ressource.

- **CVSResource** - Création d'un convertisseur permettant de produire une configuration CVS à partir d'un FileResource. Le CVSConfigurationResource résultant peut ainsi être employé en 'USING clause' d'un convertisseur 'File to CSV(structured)' pour l'interprétation du fichier CSV avec les séparateurs souhaités.

- Ajout de composants permettant de définir un dictionnaire de traduction comme ressources de test utilisable en 'USING clause'.

- **XmlResource** - Ajout d'une assertion unaire 'xml.well.formed' pour validation XML. d'un FileResource utilisé comme ressource de test. Cela est particulièrement utile lorsque l'origine du XML est le SUT lui-même.

- **PropertiesResource** - Ajout d'une fonction contournant l'encodage des valeurs lors de la conversion d'un fichier de propriété.

- **PropertiesResource** - La définition d'une propriété peut désormais être vide. Seul un 'WARNING' apparaîtra dans les logs.

- **PropertiesResource** - Amélioration du message d'erreur lorsque l'entrée d'une propriété est invalide (numéro de la ligne en cause et son contenu).