



Protocole d'évaluation de la compatibilité avec un LMS via SCORM

SCORM est un standard permettant l'échange de supports de formation entre dispositifs de production et de gestion de formation (LMS). Il spécifie comment la formation doit être structurée, comment les données produites pendant son déroulement doivent être échangées et archivées ainsi que comment les fichiers doivent être regroupés dans une archive zip.

SCORM existe actuellement dans deux versions, 1.2 et 1.3 (plus connue comme SCORM 2004), toutes deux gérées par Opale. Ce protocole indique la suite de tests à effectuer afin d'évaluer l'implémentation faite d'une ou des deux versions du standard.

Si les paragraphes précédents vous ont semblé abscons, vous devriez probablement vous renseigner plus sur ce standard avant de poursuivre votre lecture.

Table des matières

Démarche générale	3
Evaluation d'une implémentation du standard.....	3
Evaluation de l'intégration/déploiement d'un module Opale	3
Rapport de tests et partage des résultats.....	4
SCORM 2004.....	5
Evaluation d'une implémentation de SCORM 2004.....	5
Import du module factice dans le LMS.....	5
Prise en main du module factice.....	5
Scripts simulant le fonctionnement d'un module Opale	8
Evaluation de l'intégration/déploiement d'un module Opale	9
Opale Mono SCO.....	9
Opale Multi SCO	9
Conclusion.....	9
SCORM 1.2.....	10
Evaluation d'une implémentation de SCORM 1.2	10
Import du module factice dans le LMS.....	10
Prise en main du module factice.....	10
Scripts simulant le fonctionnement d'un module Opale	13
Evaluation de l'intégration/déploiement d'un module Opale	14
Opale Mono SCO.....	14
Opale Multi SCO	14



Conclusion.....	14
ANNEXES.....	15
Fichiers fournis avec le protocole.....	15

Démarche générale

Les objectifs des tests sont l'évaluation de la qualité d'une implémentation de SCORM dans un LMS et le partage des résultats obtenus avec la communauté d'utilisateurs d'Opale. Les tests portent tout d'abord sur les aspects techniques d'une implémentation, c'est-à-dire que nous allons éprouver les fonctionnalités qu'un LMS standardisé doit théoriquement respecter, et ensuite ils viseront à déterminer l'ergonomie (ou les défauts d'ergonomie) de l'intégration de supports produits avec Opale dans les LMS selon les différentes possibilités offertes.

Evaluation d'une implémentation du standard

Bien que SCORM soit un standard pour lequel une procédure de certification a été mise en place¹, ses implémentations exposent des variations qui peuvent être gênantes pour le déploiement de formations. En outre, les plates-formes open source, même les plus prisées, ne sont souvent pas certifiées et exposent des implémentations incomplètes. D'où la nécessité de tester leur implémentation avant de pouvoir les utiliser en production.

Les tests qui seront effectués auront pour cible uniquement les fonctionnalités requises pour faire fonctionner des supports de formation produits avec SCENARI dans des LMS SCORM. Par conséquent, nous n'allons pas tester l'ensemble des fonctionnalités SCORM et ce n'est pas parce qu'un LMS offrira une compatibilité SCORM satisfaisante pour faire fonctionner des contenus Opale qu'il sera nécessairement 100% SCORM.

Le principe pour l'évaluation d'un LMS à ce stade sera le suivant : un module factice sera importé dans le LMS (sous forme d'activité, parcours, module, etc. l'appellation variant d'un LMS à un autre). Lors de son lancement, il se comportera comme une formation empaquetée selon le standard sauf qu'il exposera une interface permettant la réalisation de tests interactifs : nous allons pouvoir choisir les fonctions que nous voudrions tester et observer en direct le comportement du LMS. Un script regroupant l'ensemble des tests à effectuer sera fourni pour cette étape.

L'avantage du recours à un module factice de formation, appelé aussi *déetective* ou *proddingSCO*, est qu'il nous permettra d'identifier de manière très précise les sources d'erreurs dans le cadre de la communication entre un contenu de formation et un LMS. Une fois les sources d'erreurs clairement identifiées, il nous sera plus facile de communiquer celles-ci aux éditeurs de LMS ou de mettre en place des solutions côté contenu pour contourner les erreurs (lorsque c'est applicable).

Evaluation de l'intégration/déploiement d'un module Opale

Une fois que nous aurons évalué la capacité technique d'un LMS à faire fonctionner un contenu produit par Opale, il nous restera à examiner :

- Comment le module s'insère dans l'interface du LMS : Est-ce que des problèmes d'ergonomie surviennent ? Est-ce que les liens et ressources externes s'affichent correctement ? Est-ce que les caractères accentués s'affichent correctement ?

- Est-ce que le LMS restitue correctement les données fournies par le contenu : Il s'agit de valider que les données transmises lors des interactions sont les mêmes que celles retournées par le LMS sous demande du contenu. C'est-à-dire que nous allons réaliser un ensemble d'interactions avec le contenu, quitter ce dernier et le relancer pour vérifier que les données produites seront bien présentes à nouveau dans le contenu.

La vérification de ces deux points est essentielle pour garantir que les contenus Opale auront un fonctionnement compréhensible par l'utilisateur. En effet, même si les tests de la

¹ La certification est gérée par ADL, organisme américain à l'origine du standard.

section précédente sont concluants, l'expérience d'apprentissage peut être complètement gâchée si l'ergonomie est aberrante ou si le contenu oublie systématiquement les réponses fournies par l'apprenant d'une session à une autre.

Rapport de tests et partage des résultats

Un rapport devra être rédigé au fur et à mesure de l'évaluation de chaque LMS. Cela permettra aux utilisateurs d'Opale de disposer d'informations comparables pour décrire chaque LMS testé. Cela fournira également les instructions nécessaires pour reproduire les erreurs, information utile pour les éditeurs de LMS.

Dans un souci d'homogénéité et d'exhaustivité, un modèle de rapport est fourni avec ce protocole (cf. Annexes). Il devra être utilisé pour tous les tests. Veuillez le renommer suivant la convention suivante :

AAAAMMJJ rapport de test nomLMS vxxx.pdf

Où :

- AAAA : année en 4 chiffres,
- MM : mois en 2 chiffres,
- JJ : jour en 2 chiffres,
- nomLMS : nom de la plate-forme testée (par exemple : Dokeos, Ganesha, Moodle, etc...)
- vxxx : version de la plate-forme testée (par exemple : v4.5.2)

La suite de ce document se présente comme deux parties indépendantes permettant de réaliser les tests de compatibilité soit avec SCORM 2004, soit avec SCORM 1.2. Des informations ont été rendues redondantes afin que chaque partie soit lisible de manière indépendante, ainsi vous pouvez vous passer directement à celle qui vous intéresse.

SCORM 2004

SCORM 2004 est la version la plus récente du standard (dont la 4^{ème} édition des spécifications est prévue vers la fin de l'année 2008). Celle-ci offre des possibilités de communication plus étendues que la précédente version, la 1.2, ce qui permet un échange plus riche entre LMS et contenu de formation. Néanmoins c'est aussi la version la moins bien maîtrisée du standard et il est fréquent que les implémentations dans les LMS open source soient incomplètes, seules les fonctionnalités les plus utilisées étant présentes.

Evaluation d'une implémentation de SCORM 2004

Comme indiqué dans la présentation de la démarche générale, l'évaluation s'effectue à l'aide d'un module factice. Il s'agit ici d'un module proposé sous licence *Creative Commons* produit par *Ostyn Consulting* et intitulé *Diagnostic SCO for SCORM runtime*². Il correspond à l'archive nommée *proddingsco2004.zip* jointe à ce protocole.

Import du module factice dans le LMS

Pour commencer le test, vous devez avant toute chose déployer le module sur votre LMS. Comme chaque LMS propose sa propre structuration de cours et son propre système d'intégration de modules SCORM, vous devrez vous référer à la documentation de votre plate-forme. En cas d'échec lors de l'import de l'archive zip dans le LMS, plusieurs points sont à vérifier :

- Avez-vous un compte utilisateur avec les permissions nécessaires pour la création d'une activité SCORM ?
- Avez-vous un serveur configuré pour permettre l'upload de fichiers et si oui de taille suffisante ?

Si aucun de ces points ne semble être à l'origine de votre problème, votre LMS vraisemblablement n'est pas capable d'importer des modules SCORM 2004.

Prise en main du module factice

Lorsque le module factice est correctement déployé sur votre LMS, vous pourrez le lancer et vous obtiendrez un écran correspondant à la figure 1 (à l'interface du LMS près).

² SCO de diagnostic pour environnement d'exécution SCORM.

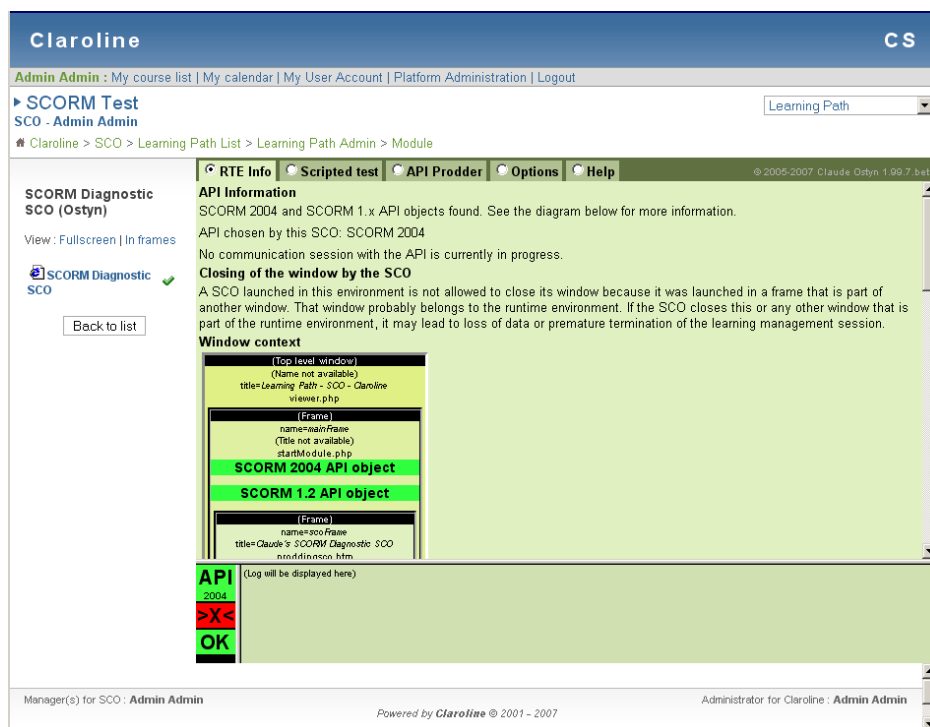


Figure 1 : Le module factice en exécution sur un LMS, ici Claroline 1.8.9

Le module expose un ensemble d'onglets :

- RTE Info : Donne des indications quant à l'environnement d'exécution dans lequel se situe le module.
- Scripted test : Permet l'exécution de tests basés sur des scripts.
- API Prodder : Permet d'appeler de manière interactive les différentes fonctions proposées par l'API SCORM.
- Options : Permet de régler quelques options offertes par le module.
- Help : Donne accès à l'aide sur le module ainsi que des détails quant à son utilisation.

Nous allons nous intéresser pour les tests uniquement aux deux premiers onglets.

Le premier onglet vous indique en effet quelles interfaces de communication (API) ont été découvertes dans l'environnement d'exécution. Dans le cas de la figure 2, vous constaterez que deux versions de l'API sont présentes et que le module a choisi par défaut d'utiliser la version 2004.

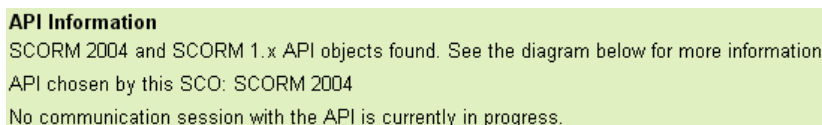


Figure 2 : API découvertes par le module de test

Si le module n'a détecté aucune API, vous obtiendrez un affichage correspondant à la figure 3. Dans ce cas, inutile de poursuivre les tests, votre LMS n'expose pas correctement son interface standardisée.



Figure 3 : Aucune API détectée

L'onglet *Scripted test* se décompose (cf. figure 4) :

- Une grande zone texte : C'est dans celle-ci que vous allez copier-coller les scripts de tests fournis avec ce protocole, à la place du texte "Enter or paste script here. See help for instructions."
- Un ensemble de boutons de contrôle : Juste en dessous de la zone de texte, ils permettent dans l'ordre une exécution pas à pas du script (*Step*), une exécution automatique du script (*Play*), de mettre en pause une exécution automatique (*Pause*), d'arrêter l'exécution d'un script (*Stop*) et enfin de définir l'intervalle de temps à attendre entre l'exécution de chaque instruction fournie dans le script en millisecondes (*Interval between calls*).
- Une zone de notification graphique : En vert et rouge fluo. Elle indique la version de l'API utilisée, l'état de la connexion entre le module et le LMS et enfin le statut d'erreur lié à la dernière instruction.
- Une zone d'affichage du journal des opérations (*log*) : C'est dans celle-ci que vous obtiendrez des traces d'exécution des scripts.

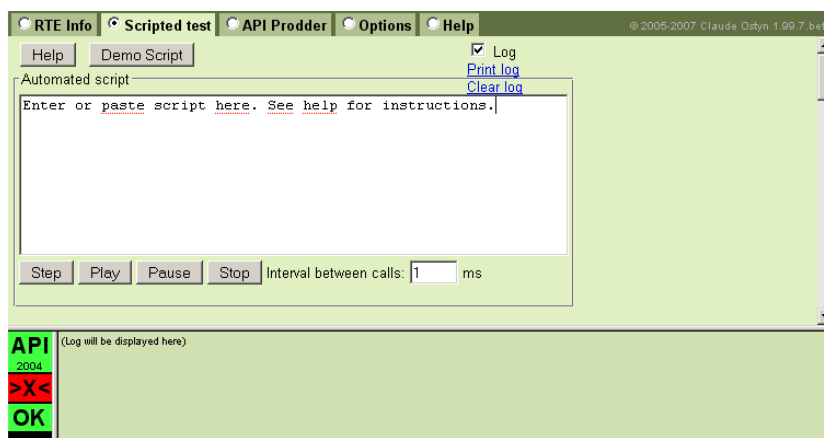


Figure 4 : L'onglet *Scripted test*

Afin de vous familiariser avec ce module de test, vous pouvez dans un premier temps cliquer sur le bouton *Demo Script* et commencer par un déroulement pas à pas en cliquant à plusieurs reprises sur le bouton *Step* et ensuite basculer en exécution automatique en appuyant sur le bouton *Play*. Faites bien attention au fait que toute donnée enregistrée sur le LMS durant le déroulement du script sera associée à ce module. **Il conviendra par conséquent d'utiliser une installation fraîche du module factice pour chaque script de test utilisé dans la suite du protocole.**

Toujours dans un objectif de familiarisation avec l'outil, veuillez copier-coller le contenu du fichier *ExampleOfBadScript.txt*³ dans la zone texte de l'onglet *Scripted test* (dans une nouvelle session d'exécution du module, i.e. quittez et relancez le module si vous avez déjà joué avec). Ce script essaie de récupérer une valeur inexistante dans le LMS dans sa 2^{ème} instruction, c'est pourquoi il devrait toujours générer une erreur lors du traitement de celle-ci. La figure 5 illustre le comportement du module lorsqu'une erreur est rencontrée : la partie inférieure de la zone de notification graphique indique en rouge le code de l'erreur (401) et la zone d'affichage du journal des opérations indique bien que l'exécution de la 2^{ème} instruction a retourné une erreur (*Returns "" Error = 401*). Vous pouvez configurer le module pour un déroulement automatique avec un intervalle de temps d'au moins 100ms (une valeur plus grande comme 500ms sera préférable lors des premiers tests) et dérouler ce script à plusieurs reprises afin de vous familiariser avec l'observation d'erreurs.

³ Fourni avec ce protocole dans le dossier correspondant à la version de SCORM testée.

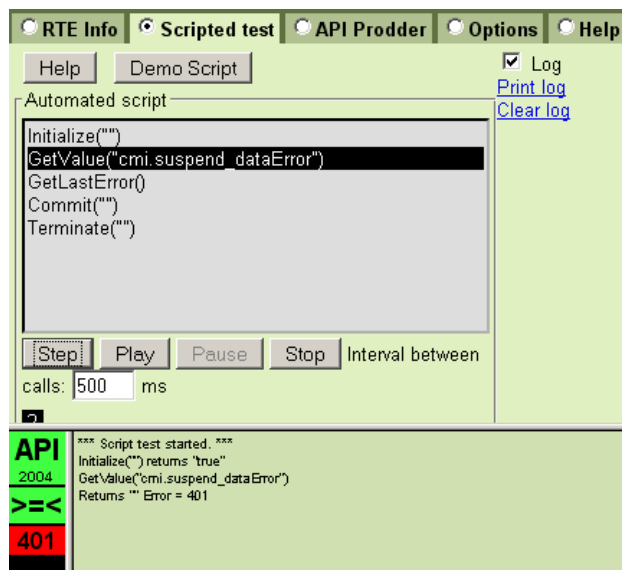


Figure 5 : Exemple d'erreur lors de l'exécution d'un script

Scripts simulant le fonctionnement d'un module Opale

A ce stade vous devez être à l'aise avec le module factice pour le déroulement des tests basés sur des scripts. Nous vous rappelons que chaque script de test doit être exécuté sur une installation différente du module factice. En effet, comme les scripts simulent le fonctionnement de modules Opale enregistrant des données sur le LMS, l'exécution successive des scripts sur la même installation du module factice risque de fausser les résultats de l'exécution dès le deuxième script.

Le test correspond à faire dérouler deux scripts grâce au module factice :

- *OpaleV3_1multiSCO2004_script.txt* : Simule le fonctionnement d'une partie très interactive d'un module Opale publié en mode multi-SCO.
- *OpaleV3_1monoSCO2004_script.txt* : Simule le fonctionnement d'un module Opale publié en mode mono-SCO. Celui-ci comprend par conséquent beaucoup plus d'instructions, puisqu'il correspond au déroulé de l'intégralité d'un module Opale et non pas uniquement une partie.

Nous vous conseillons de lancer une exécution automatique de chaque script avec un intervalle de 500ms entre chaque instruction. Placez le curseur de la souris sur le bouton *Pause* et surveillez la zone de notification graphique. Si une erreur survient et que son code s'affiche en rouge, cliquez immédiatement sur *Pause*. Notez alors le code d'erreur rencontré dans le rapport ainsi que l'instruction l'ayant générée dans le rapport de test. Vous pouvez ensuite reprendre l'exécution soit pas à pas en appuyant sur *Step*, soit en automatique en appuyant sur *Play*.

Lorsque vous arrivez à la fin du script (**avec ou sans erreurs**), placez le curseur dans la zone d'affichage du journal des opérations et faites un *Ctrl+A* pour sélectionner l'ensemble de messages générés par l'exécution du script. Copiez-le dans le presse-papier (*Ctrl+C*). Ouvrez ensuite un éditeur de texte et collez le texte du presse-papier (*Ctrl+V*). Sauvegardez le fichier texte pour le joindre au rapport de test. Nommez le fichier de la même manière que le script auquel il correspond en ajoutant le nom du LMS et « *_log* » à la fin (par exemple *OpaleV3_1multiSCO2004_script_Moodle_log.txt*).

Attention : Si vous rencontrez des erreurs reportées comme « **SCRIPT ERROR** » dans le journal d'exécution, cela signifie que le module a rencontré un problème dans l'interprétation du script. Cela ne devrait pas se produire mais nous l'avons constaté quand par exemple le script est initialement ouvert dans un éditeur de texte qui ajoute des sauts de ligne

automatiquement. Essayez de changer d'éditeur de texte pour réaliser le copier-coller du script et si l'erreur persiste contactez l'équipe SCENARI.

Evaluation de l'intégration/déploiement d'un module Opale

Cette phase du test vise à identifier des éventuels problèmes graphiques ou ergonomiques liés à l'intégration de l'interface propre d'un module Opale dans un LMS SCORM. Elle permet également de vérifier si les données envoyées au LMS par le module sont bien enregistrées et bien restituées lorsque le module désire les retrouver.

Opale Mono SCO

Pour vérifier l'intégration d'un contenu Opale publié sous forme d'un seul et unique SCO, veuillez installer sur votre LMS le module contenu dans l'archive *testOpaleSCORMmonoSCO2004.zip*. Ce module va présenter une interface où un menu de navigation est affiché directement dans le contenu. Si votre LMS le permet, veuillez désactiver l'affichage du menu de navigation du LMS. Pour effectuer les tests, lancez le module et suivez les instructions affichées durant son déroulement.

Veillez bien vous assurer que vous quittez et relancez le module avant de valider que le LMS restitue bien les données produites par le module durant une interaction.

Opale Multi SCO

Pour vérifier l'intégration d'un contenu Opale publié sous forme d'un ensemble de SCOs pour lesquels la navigation est assurée par le LMS, veuillez installer sur votre LMS le module contenu dans l'archive *testOpaleSCORMmultiSCO2004.zip*. Pour effectuer les tests, lancez le module et suivez les instructions affichées durant son déroulement.

Conclusion

Vous voilà rendus à la fin de la procédure de test d'un LMS compatible SCORM2004. Veuillez renseigner vos résultats dans le rapport de test. Nous vous invitons à publier celui-ci sur le wiki du projet SCENARI-platform à l'adresse suivante : *adresse à fournir*

SCORM 1.2

SCORM 1.2 est la version la plus commune du standard et également la mieux maîtrisée par les LMS. Elle propose des fonctionnalités moins étendues que SCORM 2004 mais en contrepartie elle est plus simple, ce qui rend sa compréhension plus aisée.

Evaluation d'une implémentation de SCORM 1.2

Comme indiqué dans la présentation de la démarche générale, l'évaluation s'effectue à l'aide d'un module factice. Il s'agit ici d'un module proposé sous licence *Creative Commons* produit par *Ostyn Consulting* et intitulé *Diagnostic SCO for SCORM runtime*⁴. Il correspond à l'archive nommée *proddingsco1.2.zip* jointe à ce protocole. Cette archive contient une version légèrement modifiée de l'outil qui ne détecte que les API SCORM 1.2. En effet, des LMS compatibles avec les deux standards exposent souvent les deux API en même temps et le module de test original privilégie l'API SCORM 2004 dans ce cas de figure, ce qui empêcherait tout test de l'API SCORM 1.2.

Import du module factice dans le LMS

Pour commencer le test, vous devez avant toute chose déployer le module sur votre LMS. Comme chaque LMS propose sa propre structuration de cours et son propre système d'intégration de modules SCORM, vous devrez vous référer à la documentation de votre plate-forme. En cas d'échec lors de l'import de l'archive zip dans le LMS, plusieurs points sont à vérifier :

- Avez-vous un compte utilisateur avec les permissions nécessaires pour la création d'une activité SCORM ?
- Avez-vous un serveur configuré pour permettre l'upload de fichiers et si oui de taille suffisante ?

Si aucun de ces points ne semble être à l'origine de votre problème, votre LMS vraisemblablement n'est pas capable d'importer des modules SCORM 1.2.

Prise en main du module factice

Lorsque le module factice est correctement déployé sur votre LMS, vous pourrez le lancer et vous obtiendrez un écran correspondant à la figure 1 (à l'interface du LMS près).

⁴ SCO de diagnostic pour environnement d'exécution SCORM.

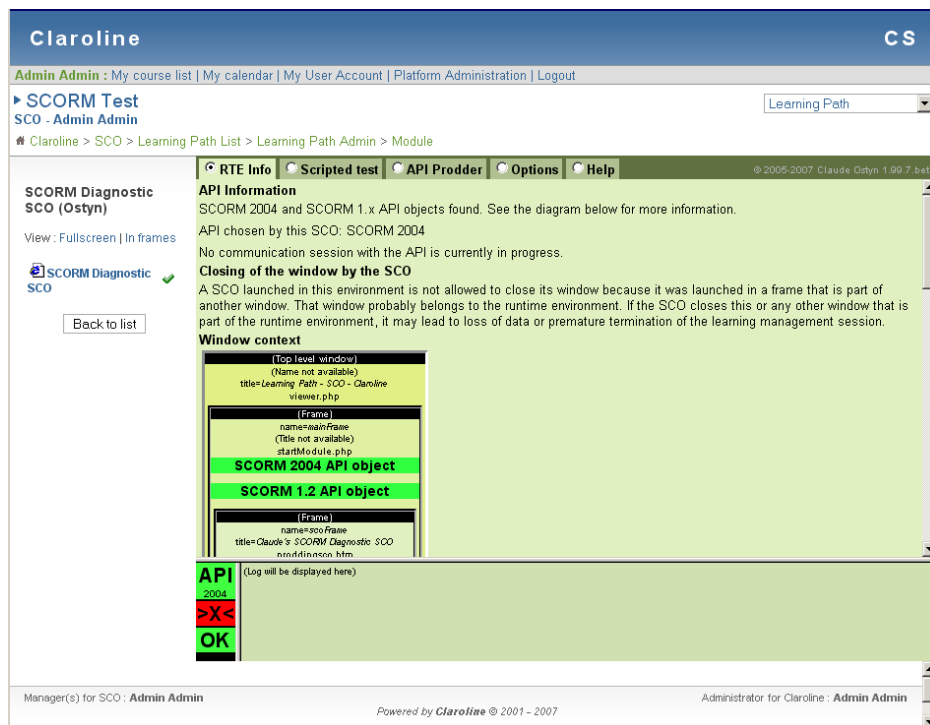


Figure 6 : Le module factice en exécution sur un LMS, ici Claroline 1.8.9

Le module expose un ensemble d'onglets :

- RTE Info : Donne des indications quant à l'environnement d'exécution dans lequel se situe le module.
- Scripted test : Permet l'exécution de tests basés sur des scripts.
- API Prodder : Permet d'appeler de manière interactive les différentes fonctions proposées par l'API SCORM.
- Options : Permet de régler quelques options offertes par le module.
- Help : Donne accès à l'aide sur le module ainsi que des détails quant à son utilisation.

Nous allons nous intéresser pour les tests uniquement aux deux premiers onglets.

Le premier onglet vous indique en effet quelles interfaces de communication (API) ont été découvertes dans l'environnement d'exécution. Dans le cas de la figure 2, vous constaterez que deux versions de l'API sont présentes et que le module a choisi par défaut d'utiliser la version 2004. Dans le cas de votre installation, vous ne devriez avoir que l'API SCORM 1.2 de détectée puisque le module a été modifié pour ne plus détecter l'API SCORM 2004.

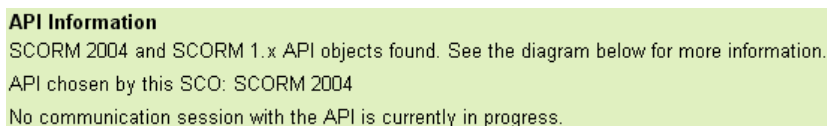


Figure 7 : API découvertes par le module de test

Si le module n'a détecté aucune API, vous obtiendrez un affichage correspondant à la figure 3. Dans ce cas, inutile de poursuivre les tests, votre LMS n'expose pas correctement son interface standardisée.



Figure 8 : Aucune API détectée

L'onglet *Scripted test* se décompose (cf. figure 4) :

- Une grande zone texte : C'est dans celle-ci que vous allez copier-coller les scripts de tests fournis avec ce protocole, à la place du texte "Enter or paste script here. See help for instructions."
- Un ensemble de boutons de contrôle : Juste en dessous de la zone de texte, ils permettent dans l'ordre une exécution pas à pas du script (*Step*), une exécution automatique du script (*Play*), de mettre en pause une exécution automatique (*Pause*), d'arrêter l'exécution d'un script (*Stop*) et enfin de définir l'intervalle de temps à attendre entre l'exécution de chaque instruction fournie dans le script en millisecondes (*Interval between calls*).
- Une zone de notification graphique : En vert et rouge fluo. Elle indique la version de l'API utilisée, l'état de la connexion entre le module et le LMS et enfin le statut d'erreur lié à la dernière instruction.
- Une zone d'affichage du journal des opérations (*log*) : C'est dans celle-ci que vous obtiendrez des traces d'exécution des scripts.

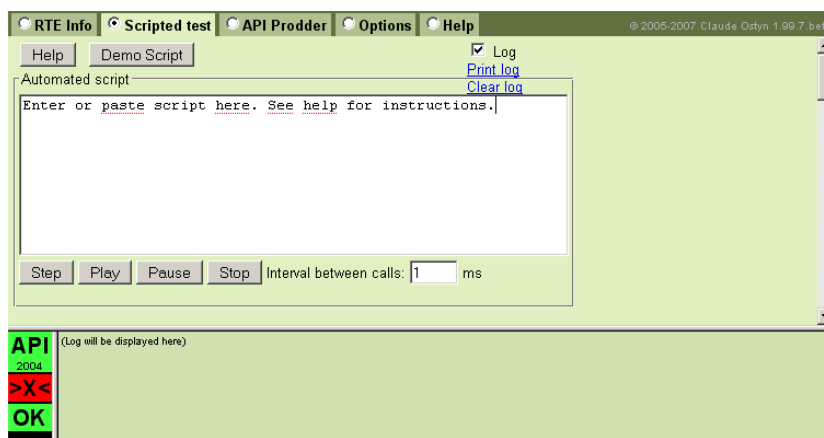


Figure 9 : L'onglet *Scripted test*

Afin de vous familiariser avec ce module de test, vous pouvez dans un premier temps cliquer sur le bouton *Demo Script* et commencer par un déroulement pas à pas en cliquant à plusieurs reprises sur le bouton *Step* et ensuite basculer en exécution automatique en appuyant sur le bouton *Play*. Faites bien attention au fait que toute donnée enregistrée sur le LMS durant le déroulement du script sera associée à ce module. **Il conviendra par conséquent d'utiliser une installation fraîche du module factice pour chaque script de test utilisé dans la suite du protocole.**

Toujours dans un objectif de familiarisation avec l'outil, veuillez copier-coller le contenu du fichier *ExampleOfBadScript.txt*⁵ dans la zone texte de l'onglet *Scripted test* (dans une nouvelle session d'exécution du module, i.e. quittez et relancez le module si vous avez déjà joué avec). Ce script essaie de récupérer une valeur inexistante dans le LMS dans sa 2^{ème} instruction, c'est pourquoi il devrait toujours générer une erreur lors du traitement de celle-ci. La figure 5 illustre le comportement du module lorsqu'une erreur est rencontrée : la partie inférieure de la zone de notification graphique indique en rouge le code de l'erreur (401) et la zone d'affichage du journal des opérations indique bien que l'exécution de la 2^{ème} instruction a retourné une erreur (*Returns "" Error = 401*). Vous pouvez configurer le module pour un déroulement automatique avec un intervalle de temps d'au moins 100ms (une valeur plus grande comme 500ms sera préférable lors des premiers tests) et dérouler ce script à plusieurs reprises afin de vous familiariser avec l'observation d'erreurs.

⁵ Fourni avec ce protocole dans le dossier correspondant à la version de SCORM testée.

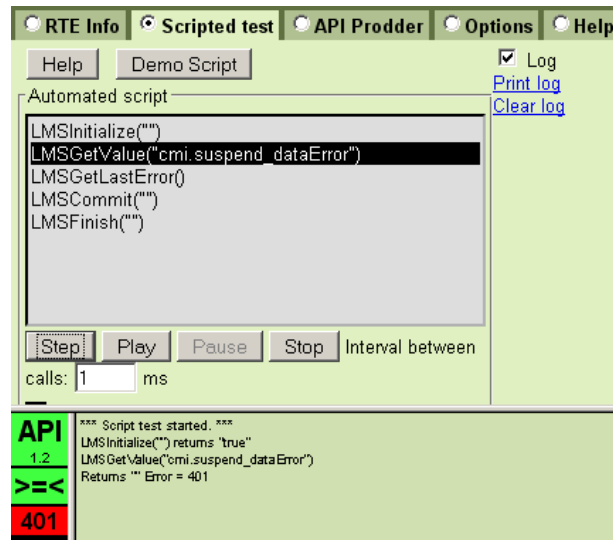


Figure 10 : Exemple d'erreur lors de l'exécution d'un script

Scripts simulant le fonctionnement d'un module Opale

A ce stade vous devez être à l'aise avec le module factice pour le déroulement des tests basés sur des scripts. Nous vous rappelons que chaque script de test doit être exécuté sur une installation différente du module factice. En effet, comme les scripts simulent le fonctionnement de modules Opale enregistrant des données sur le LMS, l'exécution successive des scripts sur la même installation du module factice risque de fausser les résultats de l'exécution dès le deuxième script.

Le test correspond à faire dérouler deux scripts grâce au module factice :

- *OpaleV3_1multiSCO12_script.txt* : Simule le fonctionnement d'une partie très interactive d'un module Opale publié en mode multi-SCO.
- *OpaleV3_1monoSCO12_script.txt* : Simule le fonctionnement d'un module Opale publié en mode mono-SCO. Celui-ci comprend par conséquent beaucoup plus d'instructions, puisqu'il correspond au déroulé de l'intégralité d'un module Opale et non pas uniquement une partie.

Nous vous conseillons de lancer une exécution automatique de chaque script avec un intervalle de 500ms entre chaque instruction. Placez le curseur de la souris sur le bouton *Pause* et surveillez la zone de notification graphique. Si une erreur survient et que son code s'affiche en rouge, cliquez immédiatement sur *Pause*. Notez alors le code d'erreur rencontré dans le rapport ainsi que l'instruction l'ayant générée dans le rapport de test. Vous pouvez ensuite reprendre l'exécution soit pas à pas en appuyant sur *Step*, soit en automatique en appuyant sur *Play*.

Lorsque vous arrivez à la fin du script (**avec ou sans erreurs**), placez le curseur dans la zone d'affichage du journal des opérations et faites un *Ctrl+A* pour sélectionner l'ensemble de messages générés par l'exécution du script. Copiez-le dans le presse-papier (*Ctrl+C*). Ouvrez ensuite un éditeur de texte et collez le texte du presse-papier (*Ctrl+V*). Sauvegardez le fichier texte pour le joindre au rapport de test. Nommez le fichier de la même manière que le script auquel il correspond en ajoutant le nom du LMS et « *_log* » à la fin (par exemple *OpaleV3_1multiSCO12_script_Moodle_log.txt*).

Attention : Si vous rencontrez des erreurs reportées comme « **SCRIPT ERROR** » dans le journal d'exécution, cela signifie que le module a rencontré un problème dans l'interprétation du script. Cela ne devrait pas se produire mais nous l'avons constaté quand par exemple le script est initialement ouvert dans un éditeur de texte qui ajoute des sauts de ligne automatiquement. Essayez de changer d'éditeur de texte pour réaliser le copier-coller du script et si l'erreur persiste contactez l'équipe SCENARI.

Evaluation de l'intégration/déploiement d'un module Opale

Cette phase du test vise à identifier des éventuels problèmes graphiques ou ergonomiques liés à l'intégration de l'interface propre d'un module Opale dans un LMS SCORM. Elle permet également de vérifier si les données envoyées au LMS par le module sont bien enregistrées et bien restituées lorsque le module désire les retrouver.

Opale Mono SCO

Pour vérifier l'intégration d'un contenu Opale publié sous forme d'un seul et unique SCO, veuillez installer sur votre LMS le module contenu dans l'archive *testOpaleSCORMmonoSCO12.zip*. Ce module va présenter une interface où un menu de navigation est affiché directement dans le contenu. Si votre LMS le permet, veuillez désactiver l'affichage du menu de navigation du LMS. Pour effectuer les tests, lancez le module et suivez les instructions affichées durant son déroulement.

Veillez bien vous assurer que vous quittez et relancez le module avant de valider que le LMS restitue bien les données produites par le module durant une interaction.

Opale Multi SCO

Pour vérifier l'intégration d'un contenu Opale publié sous forme d'un ensemble de SCOs pour lesquels la navigation est assurée par le LMS, veuillez installer sur votre LMS le module contenu dans l'archive *testOpaleSCORMmultiSCO12.zip*. Pour effectuer les tests, lancez le module et suivez les instructions affichées durant son déroulement.

Conclusion

Vous voilà rendus à la fin de la procédure de test d'un LMS compatible SCORM12. Veuillez renseigner vos résultats dans le rapport de test. Nous vous invitons à publier celui-ci sur le wiki du projet SCENARI-platform à l'adresse suivante : *adresse à fournir*

ANNEXES

Fichiers fournis avec le protocole

PackagingSpecifiques

testOpaleSCORMmonoClaroline.zip

➔ Le module de test publié en mono-SCO dans un paquet SCORM adapté au LMS Claroline.

testOpaleSCORMmonoGanesha.zip

➔ Le module de test publié en mono-SCO dans un paquet SCORM adapté au LMS Ganesha.

testOpaleSCORMmonoMoodle.zip

➔ Le module de test publié en mono-SCO dans un paquet SCORM adapté au LMS Moodle.

testOpaleSCORMmonoSyfadis.zip

➔ Le module de test publié en mono-SCO dans un paquet SCORM adapté au LMS Syfadis.

testOpaleSCORMmultiClaroline.zip

➔ Le module de test publié en multi-SCO dans un paquet SCORM adapté au LMS Claroline.

testOpaleSCORMmultiGanesha.zip

➔ Le module de test publié en multi-SCO dans un paquet SCORM adapté au LMS Ganesha.

testOpaleSCORMmultiMoodle.zip

➔ Le module de test publié en multi-SCO dans un paquet SCORM adapté au LMS Moodle.

testOpaleSCORMmultiSyfadis.zip

➔ Le module de test publié en multi-SCO dans un paquet SCORM adapté au LMS Syfadis.

SCORM1.2

ExampleOfBadScript.txt

➔ Un script de test simulant un module SCORM 1.2 produisant une erreur à titre d'exemple.

OpaleV3_1monoSCO12_script.txt

➔ Script simulant le fonctionnement d'un module Opale publié en mono-SCO et communicant avec l'API SCORM 1.2.

OpaleV3_1multiSCO12_script.txt

➔ Script simulant un le fonctionnement d'un SCO d'un module Opale publié en multi-SCO et communicant avec l'API SCORM 1.2.

proddingsco1.2.zip

➔ Module factice développé par Ostyn Consulting et permettant de réaliser des tests scriptés du comportement de l'API d'un LMS. Paquet au format SCORM 1.2 et module modifié pour ignorer les API SCORM 2004.

📄 **testOpaleSCORMmonoSCO12.zip**

➔ Le module de test Opale publié en mono-SCO dans un paquet SCORM 1.2.

📄 **testOpaleSCORMmultiSCO12.zip**

➔ Le module de test Opale publié en multi-SCO dans un paquet SCORM 1.2.

📁 SCORM2004

📄 **ExampleOfBadScript.txt**

➔ Un script de test simulant un module SCORM 2004 produisant une erreur à titre d'exemple.

📄 **OpaleV3_1monoSCO2004_script.txt**

➔ Script simulant le fonctionnement d'un module Opale publié en mono-SCO et communicant avec l'API SCORM 2004.

📄 **OpaleV3_1multiSCO2004_script.txt**

➔ Script simulant un le fonctionnement d'un SCO d'un module Opale publié en multi-SCO et communicant avec l'API SCORM 2004.

📄 **proddingsco2004.zip**

➔ Module factice développé par Ostyn Consulting et permettant de réaliser des tests scriptés du comportement de l'API d'un LMS. Paquet au format SCORM 2004.

📄 **testOpaleSCORMmonoSCO2004.zip**

➔ Le module de test Opale publié en mono-SCO dans un paquet SCORM 2004.

📄 **testOpaleSCORMmultiSCO2004.zip**

➔ Le module de test Opale publié en multi-SCO dans un paquet SCORM 2004.

📄 **AAAAMMJJ rapport de test nomLMS vxxx.docx**

➔ Modèle de rapport à renommer et à distribuer sous format PDF.

📄 **protocole de test LMS.pdf**

➔ Fichier décrivant la procédure de test que vous êtes en train de lire.