

MyPads point de la semaine 17

Comme annoncé la [semaine dernière](#), c'est désormais un point hebdomadaire qui émaillera le travail autour de MyPads. Cette semaine n'aura pas été de tout repos et les avancées visibles sont malheureusement peu nombreuses. Explications.



Les travaux

La mise en place des tests fonctionnels client, simulant une navigation réelle, a occupé les premiers jours de développement. Ensuite MyPads a subi quelques modifications pour fonctionner avec la version 4 d'[Express](#), le cadre de développement sur lequel repose Etherpad et donc MyPads. Cette migration a été initiée par le tout [premier contributeur externe](#) au plugin, et a été rendue nécessaire par la migration d'Etherpad une semaine plus tôt.

Cette migration a été l'occasion de tester à nouveau la compatibilité de MyPads avec Etherpad. Cela peut sembler étonnant, mais MyPads est développé de manière autonome vis à

vis d'Etherpad et est régulièrement testé en tant que plugin Etherpad, pour les raisons suivantes :

- accélérer le développement et éviter de devoir relancer Etherpad voire réinstaller le plugin à chaque modification;
- permettre les tests unitaires et fonctionnels serveur, très difficile sinon à partir d'un plugin Etherpad, isoler une base de tests du reste de l'instance;
- conserver une forme d'indépendance vis à vis du cœur d'Etherpad : afin de ne pas nécessiter des modifications d'Etherpad lui-même et de limiter les régressions en cas de changements internes de ce dernier.

Malheureusement le fonctionnement de MyPads s'est révélé erratique : parfois correct, parfois non. Pour les techniciens, seules les méthodes GET et HEAD sont autorisées et toute autre méthode HTTP est refusée. Le problème, nouveau, est intimement lié au logiciel intermédiaire (middleware) [yajsm1](#), lequel est employé par Etherpad afin d'optimiser les requêtes des fichiers dits statiques (scripts, images, styles etc). En théorie, les requêtes prises en charge par MyPads ne devraient pas être impactées par ce logiciel intermédiaire, mais pour une raison mal comprise, elles le sont parfois.

Le problème, c'est que MyPads fonctionne autour d'une interface de programmation standard, une API HTTP [REST](#), sur laquelle se connecte le client Web, et qui permettra à d'autres clients ou à des outils tiers de voir le jour. La résolution du soucis n'est pas aisée : l'anomalie intervient de manière aléatoire. Peu de plugins sont touchés car la plupart ne définissent pas leurs propres routes HTTP. Même si yajsm1 est modifié pour résoudre le soucis, il faudra que la fondation Etherpad accepte le patch et l'intègre avant de pouvoir retrouver un fonctionnement correct de MyPads. Or, il semble que yajsm1 sera [bientôt remplacé](#) par une technologie plus standard.

D'autres résolutions ont été envisagées, du fait de la situation de yajsm1, et entre autres :

1. Substituer, comme certains plugins le font, l'API HTTP REST par une API basée sur socket.io, la technologie employée par Etherpad pour la collaboration en temps réel et qui repose en premier lieu sur le standard WebSocket. Cette voie a été expérimentée cette semaine mais représente une charge considérable de travail et la réécriture de nombreux modules. De plus, il ne s'agit pas d'un remplacement propre : MyPads y perd une méthode de communication plus standard ainsi que son système d'authentification, lequel avait été choisi pour permettre à terme une connexion depuis des comptes externes ou encore un annuaire LDAP, OpenID etc
2. Faire de MyPads une application indépendante, de fait non plus un plugin, qui gèrerait les accès des utilisateurs aux pads en fonction des groupes définis. Le problème de cette solution est de complexifier l'installation de MyPads et de risquer des incompatibilités avec certains autres plugins. Aussi, nous sortirions de fait du cadre du cahier des charges initial.

Il a été décidé que la dernière piste ne serait à employer qu'en cas de dernier recours et c'est la migration vers socket.io qui a été d'abord privilégiée. Néanmoins, à la vue du travail nécessaire et surtout des pertes fonctionnelles que cela risque d'amener, cette solution ne sera pas poursuivie.

La semaine prochaine

Le travail va reprendre sur la version HTTP REST standard qui a été développée jusqu'ici. Il est prévu :

- qu'étant donné que la suppression du yajsm1 n'arrivera qu'à un terme inconnu, il faudra dépister l'anomalie et la résoudre, ou au moins proposer un contournement

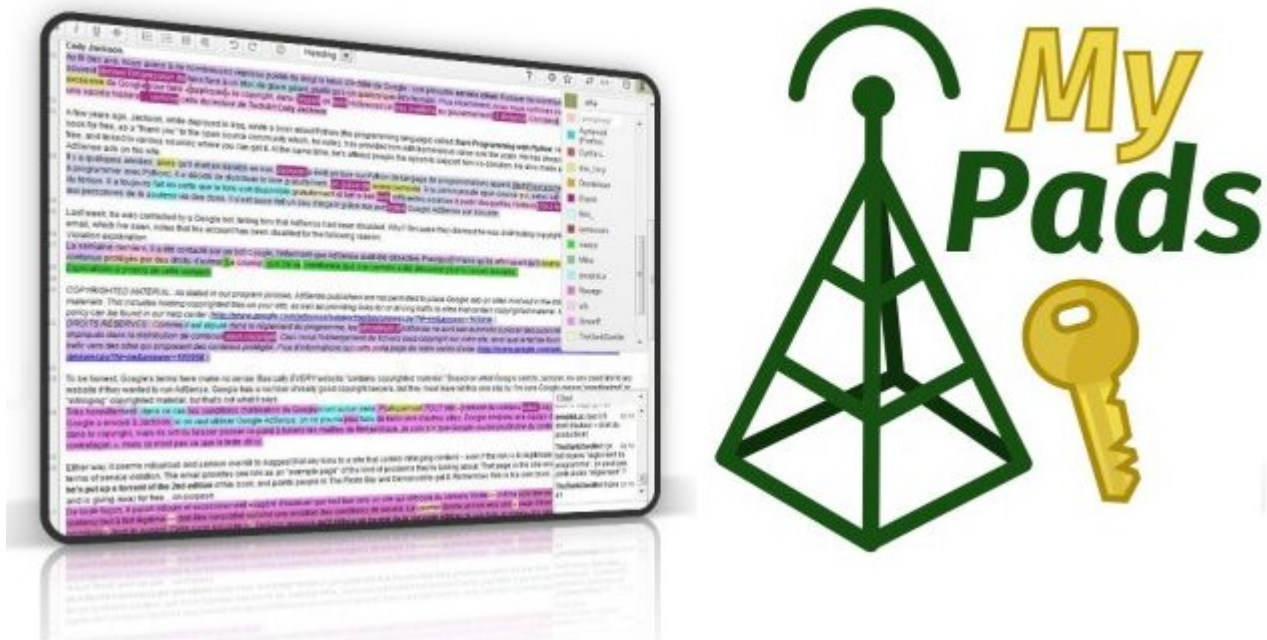
simple;

- de poursuivre le développement, moins actif que prévu cette semaine, avec notamment
 - le passage d'une authentification en propre classique vers [JSON Web Token](#), dont le travail a commencé cette semaine avec le test de socket.io, de manière à renforcer la sécurité des échanges de données chiffrées entre serveur et client;
 - les groupes et pads, évidemment.

Rendez-vous jeudi prochain pour le point de la semaine 18.

MyPads week 17

As announced [last week](#), we now give some news about MyPads development weekly. Last couple of days haven't been picnic and few enhancements are visible. Explanations below.



Work

Frontend functional testing setup, aiming to simulate real navigation, has filled the first days. Then MyPads has been

updated to work with [Express](#) version 4. Express is the framework which powers Etherpad and so MyPads. This migration has been introduced by the very [first MyPads external contributor](#), and was necessary because of the Etherpad migration a week earlier.

These modifications were a good moment to test MyPads' compatibility towards Etherpad. That can be surprising but MyPads has been programmed independently from Etherpad and is regularly tested as an Etherpad plugin, here's why :

- speeding up the development and avoiding Etherpad reboot or plugin re-installation at each update;
- allowing unit and functional server testing, quite hard from an Etherpad plugin, and isolate a test database from the whole node;
- retaining a distance regarding Etherpad core in order to avoid need of Etherpad updates and limit regressions in case of internal modifications of it.

Sadly MyPads behavior becomes erratic : sometimes correct, sometimes buggy. For technicians : only GET and HEAD HTTP verbs were allowed and all other method has been forbidden. This problem seems to be linked to the [yajsmml](#) middleware, used by Etherpad in order to optimize static files requests (scripts, images, styles etc). In theory, MyPads handled routes should not be impacted by this middleware, but for an misunderstood reason, they sometimes are.

Problem is that MyPads is based on a standard home-defined HTTP [REST](#) API, which the Web client connects to. This interface may allow other clients and third party tools to be created more easily. Debugging the problem is not an easy task, due to the randomness of the behavior. Few plugins should be concerned because most of them don't define their own routes. Even if yajsmml is updated to fix the issue, the Etherpad developers will have to accept the patch and merge it before we have MyPads working correctly. Now it seems that

yajsm1 will be [soon replaced](#) by a more standard technology.

Others resolutions have been considered, regarding to yajsm1 situation :

1. Replace, as others plugins do, the HTTP REST API by a [socket.io](#) one. socket.io is the technology used by Etherpad for realtime collaboration, that use as a first class citizen the WebSocket protocol. This approach has been tried this week but requires a considerable amount of work and many modules rewriting. Moreover, it's not a proper replacement : MyPads loses its more standard communication method but in addition its authentication system, which have been chosen to allow, later, connection through external accounts, LDAP directory, OpenID etc
2. Move MyPads from a plugin to a standalone application, which handle user access according to created groups and pads. Problem with this solution : harden the MyPads installation, risks of incompatibilities with some other plugins. Also, making a standalone app won't conform to the initial specifications.

We have decided to follow the last proposition only as a last resort. The migration to socket.io has been preferred but, with the light of required work and moreover functional loses, it won't continue.

Next week

Work will be resumed on the HTTP REST version, the one developed until now. We expect :

- because yajsm1 removal won't happen before an unknown time, it will be important to find and fix the bug, or at least to provide a simple workaround;
- move forward, with
 - migration from a classical authentication to [JSON](#)

[Web Token](#), which has been partially done this week as part of socket.io test, in order to harden encrypted data exchanges between client and server;

- groups and pads, obviously.

See you next Thursday for week 18 point.