



Documentation
Projet gus05

Titre : Introduction aux chantiers

Auteur : Augustin Delale

Version : 29/09/2009

INTRODUCTION AUX CHANTIERS

Version du 29/09/2009

Les chantiers sont des sous-projets à l'intérieur du projet gus05. Ils regroupent les projets applicatifs (centrés autour du développement d'une application en particulier) ainsi que les études techniques, fonctionnelles, méthodologiques ou didactiques. Ce sont les chantiers qui vont permettre aux développeurs Java le désirant de s'impliquer dans le projet gus05, qui vont permettre au projet de s'organiser et faire vivre sa communauté de développeurs. Bien sûr, il peut y avoir autant de chantiers que de personnes motivées amenant de nouvelles idées pour améliorer le projet gus05. Néanmoins, l'objectif de ce document est de vous présenter quelques chantiers de base afin de nous amener à un travail collaboratif effectif, une situation dans laquelle nous utiliserions des outils pour développer, documenter et échanger nos travaux, entre autres choses.

Mais dans un premier temps, il va être nécessaire de faire le tour des problématiques liées au projet gus05 lui-même, qui font que la simple connaissance des mécanismes de programmation acquis avec les ateliers de formation ne suffit pas à assurer ce travail collaboratif.

Mise à jour des problématiques du projet gus05

Les ateliers de formation, et après ?

Après avoir lu l'ensemble des ateliers de formation sur le framework gus05 et le gestionnaire Kassia, vous en êtes très certainement venu à la même conclusion que moi. Tout cela est très bien mais ce n'est pas suffisant. Et pour cause, il ne s'agit que des rudiments de programmation pour vous permettre de développer des entités. Mais les entités ne font pas tout. Il faut les paramétrer et les faire interagir dans des structures fonctionnelles plus ou moins complexes dont la gestion ne se fait pas sans mal. Nous avons appris les mots et les règles grammaticales de base de notre langage mais la route est encore longue pour tenir des discours cohérents. Tout reste à faire !

Paramétrage manuel et application test

Vers quoi nous dirigeons-nous si nous persistons à développer des entités de la même manière que dans les ateliers de formation ? Vers une application de test dont les entités ont été paramétrées manuellement au fur et à mesure de leurs apparitions, une application qui forme désormais un tout hétéroclite et dont il n'est pas aisé de séparer les paramétrages et ressources propres à chaque entité.

Bien sûr, une entité est par essence réutilisable dans d'autres contextes et d'autres applications (sur un plan purement structurel) mais encore faut-il pouvoir adapter son paramétrage. Dans notre méthodologie de développement d'entité, il va être nécessaire de prendre en compte cette problématique très tôt pour éviter de se retrouver avec des milliers d'entités dont les paramétrages se mélangent en une seule application devenue tellement grosse qu'il lui faut à chaque fois plus d'une minute pour se lancer (c'est l'expérience qui parle !)

Les trois niveaux de structure : entité - plugin - application

Tout l'enjeu va être de rendre aisée l'adaptation du paramétrage des entités à divers contextes d'utilisation. Rendre une entité réutilisable sur un plan applicatif suppose d'avoir délimité son paramétrage mais aussi ces dépendances et la manière dont elles doivent se résoudre. Une nouvelle structure devient sous-jacente, une structure intermédiaire entre l'entité et l'application. Il s'agit du **plugin**. Contrairement à d'autres frameworks, les plugins définis dans le cadre du projet gus05 ne sont pas des composants de programmation, mais l'association d'un ou plusieurs de ces composants avec du paramétrage et des ressources. En pratique, un plugin se formalisera sous la forme d'un fichier texte de règles indiquant à l'application quels entités utiliser et de quelle manière. Il nous faudra alors définir un système de gestion de plugins permettant à une application d'installer et de désinstaller facilement plusieurs plugins mais surtout de combiner leurs fonctionnalités pour obtenir un résultat applicatif cohérent et réversible.

Les deux types d'application

De manière générale, on distinguera deux types d'application :

- **L'application finalisée** dont la structure est a priori figée et qui se compose d'un unique fichier JAR autonome

- **l'application modulaire** dont le fichier JAR d'origine n'est qu'un socle prévu pour être enrichi par les fonctionnalités de plugins (on parlera dans ce cas de plateforme applicative)

Les applications finalisées présentent un paramétrage semblable à celui de notre projet Eclipse (à l'exception que celui-ci aura été généré automatiquement grâce à des outils)

Les applications modulables ou plateformes applicatives sont quant à elle des applications à la base finalisées mais prévues pour accepter un paramétrage extérieur bien particulier permettant de gérer des plugins.

Les différents chantiers de base

Les différents chantiers :

- outils d'aide au développement
- planification et développement des entités
- déploiement et documentation des entités
- déploiement des applications
- système de gestion de plugins