

# Le design applicatif avec Alias



ANNÉE 2014, N° 17

DATE DE PARUTION : MAI 2014

## *Comment construit-on une application avec Alias ?*

Ce mois-ci dans la newsletter, un numéro spécial que nous allons consacrer à la construction applicative sous Alias et à la façon d'aborder le design de votre future application de gestion.

Les retours d'expérience de celles et ceux qui ont bien voulu échanger avec notre équipe de designers, nous permettent de vérifier de manière continue l'adéquation de la solution Alias avec vos attentes.



### Dans ce numéro :

Alias dans le monde	1
Focus Spécial - Comment aborder la réalisation d'une application	2
Alias et son évolution	4

Et vous êtes nombreux à nous demander pourquoi dans Alias, on donne autant la priorité au design du modèle de données par rapport notamment au design des écrans. C'est ce que nous allons voir dans ce numéro de mai.

Et comme chez Sun i-tech, on a à cœur de satisfaire celles et ceux qui nous font confiance, nous lèverons également le voile sur deux évolutions à venir qui devraient en ravir plus d'un....

## Alias dans le monde - mai 2014



### Utilisateurs Alias dans le monde :

Algérie, Angleterre, Bénin,  
Burkina Faso, Cameroun,  
Congo (RDC), Côte d'Ivoire, Espagne,  
France, Guadeloupe, Guinée,  
Guyane Française, Luxembourg,  
Maroc, Martinique,  
Nouvelle Calédonie, Polynésie, Sénégal,  
Togo, Tunisie.



## Focus spécial

### Comment aborder la réalisation d'une application avec Alias ?



#### Par où commence t-on ?

Une application commence par un objectif. **Que veut-on gérer ?** Ensuite, d'autres questions viennent se poser.

**Dispose-t-on déjà du système d'information** (la base de données et les tables) ? Dans ce cas, on parle souvent de retro engineering. Ou bien doit-on le créer de toute pièce ? On entre alors dans le design du système d'information.

**Comment l'application doit-elle fonctionner ?** Est-ce une application transactionnelle faisant appel à des formulaires de saisie ? Est-ce une application de services qui dialoguera via des Web services ou des flux de données ? Ou encore un mix de tous ces fonctionnements réunis ? Selon le cas de figure, l'IHM n'a pas la même ampleur (elle peut se résumer parfois à de simples formulaires de contrôle).

**Comment va-t-on accéder à l'application ?** L'application va-t-elle être utilisée dans le cadre d'un intranet local, via VPN ou alors dans un cadre internet ? Selon le cas, il faudra adapter le design pour une plus grande robustesse face aux attaques potentielles.



On voit ainsi qu'il existe de nombreuses situations qui entraînent une approche du design différente.



#### Note Alias Team :

Alias est conçu pour les développements en nouvelles technologies qui remplaceront les applications métiers développées avec les technologies d'autrefois... c'est-à-dire tous les développements antérieurs à 2000 sans exception. Dans ces applications métier anciennes, la partie système de données est complexe et les règles de gestion nombreuses, c'est pourquoi nous avons mis l'accent sur ce point.



#### Dev, Test, Prod : Comment organiser l'environnement d'utilisation ?

Tout dépend de la complexité et de l'étendue fonctionnelle de l'application.

Sauf à ce qu'une application comporte peu de tables (voire même qu'une seule), une application plus complexe nécessitera de définir des tables de paramètres pour donner de la souplesse dans l'usage de l'application. Les propriétés fonctionnelles des tables de paramètres sont différentes de celles des tables de production. Par exemple, les taux de taxes sont stockés dans des tables de paramètres car ils évoluent.

Si l'on veut traiter de manière professionnelle, il faut prendre en compte un environnement de développement, un environnement d'intégration et de recette ainsi qu'un environnement de production. On pourra ainsi garantir des évolutions applicatives sans risque. Il faut donc des outils capables de gérer ces différents environnements et de migrer d'un environnement à l'autre les composants pour les intégrer, les tester et enfin les mettre en production.



**On note au passage la distinction entre table de paramètres et table de production :** les données paramètres doivent être copiées de l'environnement de recette vers l'environnement de production alors que les tests eux-mêmes ne doivent évidemment pas l'être.

La phase de recette permet d'ajuster les paramètres pour être conformes aux spécifications de votre application.



### Note Alias Team :

Si Alias Entry est un système de développement monoposte, il peut évoluer par l'adjonction d'un serveur Alias Runtime. Dans ce cas, le serveur interne à l'application Alias devient le serveur de recette. Le serveur Runtime est quant à lui le serveur de production. Alias comprend les utilitaires associés, appelés « Packaging de composants » pour passer du développement à l'intégration et la recette, et « Packaging Application » pour passer de l'environnement de recette à l'environnement de production. Alias intègre parfaitement les problématiques d'upgrade d'exploitation qui, sans outil, sont coûteuses et engendrent des frayeurs inutiles !



### Où déclarer les règles de gestion?

L'objet des règles de gestion est de s'assurer que les données saisies sont conformes aux spécifications. Ces règles peuvent être stipulées à 3 endroits : au niveau de la base de données, au niveau du moteur de données Alias ou au niveau du formulaire de saisie.

Quel est le meilleur choix ? Tout dépend du contexte de gestion et de la conception souhaitée.



Prenons l'exemple d'une souscription d'un contrat d'assurance vie. Indiquons la règle suivante : l'assuré ne doit pas dépasser 65 ans à la fin de son contrat. On a donc une règle où l'âge à la souscription plus la durée du contrat doit être inférieur ou égal à 65.

Si cette règle n'est pas respectée lors de la saisie, on a deux choix : le rejet ou l'enregistrement en spécifiant un contenu erroné. Si l'on rejette la saisie, il n'y aura que la possibilité manuelle de contacter l'agent d'assurance ou le souscripteur pour lui indiquer l'impossibilité de souscription. Et s'il ne s'agit que d'une erreur de transcription, il faudra tout ressaisir...

Dans le cas de l'enregistrement spécifiant un contenu erroné, on peut générer un courrier automatique indiquant le problème et mettre le dossier en attente de la réponse. Lorsque celle-ci arrive, il y aura juste correction des éléments erronés sans ressaisie intégrale.

Sans le dire, **on fait ici appel à la technique de paramétrage de process et du workflow associé**. On découvre donc l'intérêt d'utiliser des « états » associés à certaines tables (Alias nomme ces tables « tables de stock »).

Dans notre exemple, lors du processus de souscription, les règles associées au formulaire de saisie sont plus « souples » que les règles associées au moteur de données. Par contre, les règles du moteur de données veilleront à leur stricte application en fin de processus lors de la validation de la souscription.



Avec Alias, on peut ainsi combiner des règles au niveau du formulaire de saisie et au niveau du moteur de données.



### Note Alias Team :

Compte tenu du fait qu'Alias s'inscrit dans le contexte réseau d'internet, les données peuvent provenir aussi bien de bases de données (de type divers) que de Web-services. On a choisi de ne pas utiliser les contraintes d'intégrité au niveau des bases elles-mêmes mais de les faire remonter au niveau du moteur de données. Ainsi peu importe la manière de stocker les données (base de données ou autre). Par ailleurs, en commençant par spécifier les données, puis les dossiers fonctionnels et enfin les formulaires, ces derniers hériteront automatiquement des règles. Il sera ensuite plus facile de les « alléger » si besoin pour tenir comptes de la discussion ci-dessus.



### Les tables sont-elles formées de champs sans lien entre eux ?

La réponse est évidemment non. Si l'on prend simplement une adresse, toutes les lignes de l'adresse appartiennent au même groupe de données et sont manipulées ensemble. On pourrait citer d'autres exemples d'associations de champs. Alias a utilisé le terme de « Catégorie » pour désigner ces associations (ou regroupements) de champs.

Dans la description des tables ou des dossiers fonctionnels, les champs sont regroupés (si on le souhaite) sous leur catégorie. Un mécanisme d'héritage propage la catégorie depuis la table au dossier fonctionnel et au formulaire.



Cette notion de catégorie est également utilisée par le générateur de formulaire pour automatiser la création des écrans.



#### Note Alias Team :

Lorsque l'on va dessiner un formulaire, les champs appartenant à une même catégorie seront rassemblés sur le formulaire. Dans Alias, on a considéré que si au sein d'une même catégorie on rangeait les champs dans l'ordre de la visualisation ou de la saisie, on pourrait générer automatiquement un bloc d'affichage. La construction du formulaire se résume alors à l'assemblage de ces blocs.

Dans une optique de re-engineering d'une application, puisque l'on part d'un système de données existant, la bonne organisation des champs au niveau du dossier fonctionnel est de les regrouper en catégories. Cela permet une grande productivité dans la fabrication des nouveaux formulaires. Mais rien n'empêche de concevoir les formulaires « à la main ». Il faut laisser place à l'imagination !

## Alias et son évolution

Jusqu'à présent, nous avons privilégié une approche de la conception applicative suivante : conception du système d'information, conception des fonctionnalités et traitements associés et enfin, construction de formulaires (écrans).

Certains utilisateurs nous ont fait part du souhait de pouvoir « dessiner » un formulaire sur une page « blanche », c'est à dire sans avoir besoin de passer préalablement par les étapes de conception du système d'information.



Notre équipe d'ingénieurs finalise actuellement une interface utilisateur qui permettra à la fois un design manuel en mode graphique des formulaires tout en gardant la possibilité de génération automatique.

Cette évolution très attendue est prévue pour juin.

Autre nouveauté à venir, un Alias Wizard (assistant Alias) qui permettra une prise en main simplifiée (et donc plus rapide) d'Alias Application Designer.

Alors, encore un peu de patience...



Service Communication  
59 rue Glesener  
L-1631 Luxembourg  
Grand-Duché de Luxembourg

**Avec Alias Spécifier c'est développer !**



**Nouveau  
packaging  
d'installation  
disponible**

Deux mois d'essai gratuit

Aucun engagement

Renouvellement facultatif par tranche de 30 jours ou plus

## Découvrez Alias à la demande

**2 mois  
d'essai gratuit !**

### Profitez d'Alias Entry sans compromis !

Avec Alias à la demande et son offre exclusive de deux mois gratuits\*, vous pouvez désormais essayer Alias Entry sans aucun engagement.

Créez des applications de gestion 100% Web, sans aucune ligne de code.

Passez du mode intranet au mode Web d'un simple clic grâce aux fonctionnalités Entry Internaute et beaucoup d'autres possibilités encore...

### Libérez votre créativité sans aucune contrainte budgétaire !

Le package Alias comprend Alias V1.1, Apache Tomcat 7 et Java JDK 1.6 incluant la base de données Apache DERBY.

\*Offre sans obligation d'achat

Actuellement disponible sur notre site

**[www.alias-ad.com](http://www.alias-ad.com)**

Copyright Sun i-Tech 2014.

La reproduction des textes de cette newsletter est autorisée sous réserve du respect des trois conditions cumulatives suivantes : - que la diffusion de ces reproductions soit gratuite ; - que les documents reproduits respectent l'intégrité du contenu des documents d'origine, sans modification ni altération d'aucune sorte ; - que la source soit citée clairement et lisiblement. Les éventuelles marques citées dans la Newsletter sont déposées par les sociétés qui en sont propriétaires. Images d'illustration Fotolia.com