



[HORNET]

Migration d'un projet Hornet 3.4 vers Hornet 3.6B



Développement Hornet 3.6B

Cette création est mise à disposition selon le Contrat Paternité - Pas d'Utilisation Commerciale - Partage des Conditions Initiales à l'Identique disponible en ligne <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/> ou par courrier postal à Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA

SUIVI DES MODIFICATIONS

Version	Auteur	Description	Vérification	Date
1.0	O. Rousseil	Mise à jour de Hornet Version 3.4 vers Hornet Version 3.6B	S. Heurtematte	09/07/2014

DOCUMENTS DE REFERENCE

Titre
HORNET_GUI_Création d'un projet Hornet 3.6B
HORNET_GUI_Guide du développeur Hornet 3.6B
HORNET_GUI_Guide de paramétrage_1.4
HORNET_NOR_Normes d'encodage_1.0

SOMMAIRE

SUIVI DES MODIFICATIONS	2
DOCUMENTS DE REFERENCE	2
SOMMAIRE	3
TABLEAUX	4
FIGURES.....	4
1 OBJECTIFS DU DOCUMENT	5
1.1 PRINCIPALES EVOLUTIONS	5
1.2 VERSIONS CIBLE DES COMPOSANTS	5
1.3 MODE D'EMPLOIS.....	5
2 MISE A JOUR TECHNIQUE	6
2.1 CONFIGURATION DU PROJET SOUS ECLIPSE.....	6
2.1.1 <i>Prérequis</i>	6
2.2 CONFIGURATION DE L'ENVIRONNEMENT DE L'APPLICATION A MIGRER	6
2.2.1 <i>Migration des librairies clientes : Hornet client</i>	6
2.3 TACHE A REALISER : MISE A JOUR DES LIBRAIRIES	6
2.3.1 <i>Mise à jour des ressources statiques : hornetclient</i>	6
2.3.2 <i>Migration des librairies serveur : hornetserver</i>	7
2.3.3 <i>Prise en compte des nouvelles versions : hornetclient, et les thèmes CSS</i>	8
2.4 MIGRATION TECHNIQUE DE TILES 2.0.6 VERS 3.0.4 GA.....	9
2.4.1 <i>Changement des versions des TLD</i>	9
2.4.2 <i>Adaptation du code Java</i>	9
2.4.3 <i>Initialisation de la configuration dans le web.xml</i>	10
2.4.4 <i>Adaptation des JSP</i>	10
2.4.5 <i>Adaptation des fichiers de définition « tiles*.xml »</i>	11
2.4.6 <i>Lancement de l'application</i>	11
3 MISE A JOUR STRUCTURELLE.....	12
3.1 RESSOURCES STATIQUES.....	12
3.1.1 <i>Fichiers JavaScript</i>	12
3.1.2 <i>Fichiers CSS</i>	14
3.1 JSP	15
3.1.1 <i>JSP hornetConfig.jsp</i>	15
3.1.2 <i>JSP baseLayout.jsp</i>	16
3.2 TILES 3	17
3.2.1 <i>Impacts</i>	17
3.2.2 <i>Wildcards</i>	17
4 MISE A JOUR TECHNIQUE	18
4.1 AJOUT D'UN MODE COMBINE (OU COMBO).....	18
4.1.1 <i>Présentation</i>	18
4.1.2 <i>Combos disponibles</i>	18
4.1.3 <i>Mode par défaut</i>	19
4.1.4 <i>Choix du combo</i>	19
4.2 ACTIVER LE DEBUG DE YUI ET HORNETCLIENT	19
4.3 COMPOSANT TABLEAU	19
4.3.1 <i>Composant « Datatable » remplaçant « Pagintable »</i>	19
4.4 COMPOSANT PANEL	22
4.4.1 <i>Tâche à réaliser : modifier le code des Popin</i>	22

4.5	COMPOSANT CALENDRIER.....	22
4.6	COMPOSANT BOUTON.....	23

TABLEAUX

Tableau 1 : Migration vers Hornet 3.6B - Equivalences pour le composant « Tableau »	20
Tableau 2 : Migration vers Hornet 3.6B - Equivalences pour le composant « Popin ».....	22

FIGURES

Aucune entrée de table d'illustration n'a été trouvée.

1 Objectifs du document

Dans le cadre de la mise en place des évolutions au sein du Framework Hornet, ce document spécifie les actions à effectuer au sein d'un projet de type Hornet pour migrer de la version 3.4 de Hornet vers la nouvelle version du Framework Hornet 3.6B.

1.1 Principales évolutions

Cette nouvelle version 3.6B d'Hornet comporte principalement les évolutions et corrections suivantes par rapport à la version 3.4 :

- Migration des composants YUI 2 vers YUI3 (3.17.2)
 - Migration des styles CSS
 - Migration du code JavaScript
 - Suppression totale de 2in3 (YUI 2.9.0)
- Réalisation du mode combine
- Migration Tiles 3.0.3GA
- Migration Struts 2.3.16.3
- Migration de Tiles 2 vers Tiles 3 (3.0.4)
 - Utilisation de wildcard (*) dans les définitions de tiles pour mutualiser le code

1.2 Versions cible des composants

Ce document est basé sur les versions de composants suivantes :

- ✓ hornetserver 3.6.1
- ✓ hornetclient 3.6.1
- ✓ hornettemplate 3.6.1

1.3 Mode d'emplois

La migration doit se faire dans l'ordre des chapitres qui sont structuré de la même manière :

- Un rappel et/ou préambule conditionne la réalisation du chapitre.
- Une partie « **Tâche à réaliser** » décrit la migration unitaire. Un tableau montre les fichiers à modifier dans la version 3.4 et dans la nouvelle version. Les modifications sont surlignées **en jaune**. Si le nom du fichier est connu, il est noté dans le tableau.
- Une dernière partie permet de vérifier la bonne migration.

2 Mise à jour technique

2.1 Configuration du projet sous Eclipse

2.1.1 Prérequis

Avant de commencer cette migration, il faut avoir installé et configuré l'environnement de développement conformément au guide de paramétrage, en particulier pour la configuration de Tomcat 6 et JDK 6 dans Eclipse.

2.2 Configuration de l'environnement de l'application à migrer

2.2.1 Migration des bibliothèques clientes : Hornet client

Les thèmes et JavaScript doivent être installés sur un serveur de framework Hornet.

2.2.1.1 Rappel

Le framework Hornet est constitué de plusieurs parties :

- hornetserver : Fichiers de configuration ou fichiers Java, ils sont utilisés sur le serveur web.
- hornetclient : Fichiers JavaScript ou CSS (thème default, diplonet et francediplo), ils sont utilisés côté client (sur le navigateur)

2.2.1.2 Prérequis

Vous devez posséder les fichiers des bibliothèques à mettre à jour:

- hornetclient 3.6.1
- YUI 3.17.2,
- YUI Gallery 2014.02.13-03-13.

Si vous ne les possédez pas, le chapitre suivant décrit comment les obtenir.

2.2.1.2.1 Récupération des sources du framework Hornet à partir de hornettemplate

La migration vers Hornet 3.6B nécessite de récupérer de nouveaux fichiers et des fichiers mis à jour dans cette nouvelle version du framework. Ainsi, la source de ces fichiers doit être un projet « **modèle** » créé à partir de hornettemplate (cf. Guide de création d'un projet Hornet).

Dans la suite de ce document, sauf mention d'une autre source, les fichiers mentionnés sont donc ceux du projet créé à partir de hornettemplate **3.6.1**.

2.3 Tâche à réaliser : mise à jour des bibliothèques

2.3.1 Mise à jour des ressources statiques : hornetclient

- Ajouter les nouvelles versions sous apache :

Fonctionnalité	Code Hornet 3.4	Code Hornet 3.6B
Répertoire sous apache		
Ajouter une nouvelle version de Hornet javascript	htdocs\hornetclient\3.4.0\fwk**	Ajouter la nouvelle version htdocs\hornetclient\3.6.1\fwk** htdocs\hornetclient\3.4.0\fwk**

Ajouter une nouvelle version de Hornet styles CSS	<pre>htdocs\hornetclient\3.4.0\themes\hornet-skin-default** htdocs\hornetclient\3.4.0\themes\hornet-skin-diplonet-3.4.0** htdocs\hornetclient\3.4.0\themes\hornet-skin-francediplo-3.4.0**</pre>	Ajouter la nouvelle version <pre>htdocs\hornetclient\3.4.0\themes\hornet-skin-default** htdocs\hornetclient\3.4.0\themes\hornet-skin-diplonet-3.4.0** htdocs\hornetclient\3.4.0\themes\hornet-skin-francediplo-3.4.0** htdocs\hornetclient\3.6.1\themes\hornet-skin-default** htdocs\hornetclient\3.6.1\themes\hornet-skin-default-xcombine** htdocs\hornetclient\3.6.1\themes\hornet-skin-diplonet-3.6.0** htdocs\hornetclient\3.6.1\themes\hornet-skin-diplonet-xcombine-3.6.0** htdocs\hornetclient\3.6.1\themes\hornet-skin-francediplo-3.6.0** htdocs\hornetclient\3.6.1\themes\hornet-skin-francediplo-xcombine-3.6.0**</pre>
Ajouter une nouvelle version de YUI	<pre>htdocs\yui\yui\3.8.1**</pre>	Ajouter la nouvelle version <pre>htdocs\yui\yui\3.8.1** htdocs\yui\yui\3.17.2**</pre>
Ajouter une nouvelle version de YUI Gallery	<pre>htdocs\yui\gallery\2012.12.12-21-11**</pre>	Ajouter la nouvelle version <pre>htdocs\yui\gallery\2012.12.12-21-11** htdocs\yui\gallery\2014.02.13-03-13**</pre>

2.3.1.1 Vérification

- Lancer Apache httpd.
- A l'aide d'un navigateur internet, accéder à un des fichiers des librairies précédemment déposées sous apache.

Exemple : [<http://localhost/hornetclient/3.6.1/fwk/hornetconfig/hornetconfig-min.js>]

2.3.2 Migration des librairies serveur : hornetserver

A partir d'une tâche ANT, mise à jour des fichiers jar Hornet présents dans le dossier lib de l'application.

2.3.2.1 Prérequis

Vous devez posséder les fichiers des librairies à mettre à jour :

- hornetserver 3.6.1 (hornetserver-core et hornetserver-web)
- Autres librairies hornetserver (hornetserver-httpparam, hornetserver-typemime, ...)

Si vous ne les possédez pas, se reporter au chapitre: [Récupération des sources du framework Hornet à partir de hornettemplate](#)

2.3.2.2 Tâche à réaliser : gestion des dépendances

- Faire pointer le projet « **hornetserver-web** » et les autres librairies Hornet vers la version **3.6.1** :

Fonctionnalité	Code Hornet 3.4	Code Hornet 3.6B
Fichier ivy.xml		

Nouvelle version de hornetserver-web	<pre>• <dependency org="fr.gouv.diplomatie.hornet" name="hornetserver-web" rev="3.4.0" conf="compile- >master,libraries;provided;runtime- >master,libraries;test;sonar;cobertura" transitive="true"> [...] </dependency></pre>	<pre>• <dependency org="fr.gouv.diplomatie.hornet" name="hornetserver-web" rev="3.6.1" conf="compile->master,libraries;provided;runtime- >master,libraries;test;sonar;cobertura" transitive="true"> [...] </dependency></pre>
--------------------------------------	--	---

- Choix 1 : Lancer la tâche Ant « **init-dev** » afin de récupérer les nouvelles librairies dans le répertoire : « **WEB-INF/lib** ».
- Choix 2 : Remplacer les librairies : « **WEB-INF/lib** ». Les anciennes doivent, bien sûr, être supprimées.

➔ **Si des librairies spécifiques sont présentes pour votre projet dans WEB-INF/lib, vous devrez les recopier**

2.3.2.3 Vérification

Vérifier la présence et la mise à jour des nouvelles dépendances à la suite du build (choix1):

- hornetserver-core-3.6.1.jar
- hornetserver-web-3.6.1.jar

Les anciens jar de Hornet en version 3.4 ne doivent plus être présents.

La version 3.6B d'Hornet fait aussi une mise à jour des librairies liées au framework « **Tiles** » et « **struts** »; vérifier que vous disposez bien les nouvelles dépendances suivantes :

- slf4j-api-1.5.8.jar
- slf4j-log4j12-1.5.8.jar
- struts2-tiles3-plugin-2.3.16.jar
- tiles-api-3.0.3.jar
- tiles-autotag-core-runtime-1.1.0.jar
- tiles-core-3.0.3.jar
- tiles-jsp-3.0.3.jar
- tiles-request-api-1.0.3.jar
- tiles-request-jsp-1.0.3.jar
- tiles-request-servlet-1.0.3.jar
- tiles-request-servlet-wildcard-1.0.3.jar
- tiles-servlet-3.0.3.jar
- tiles-template-3.0.3.jar
- struts2-core-2.3.16.3.jar
- struts2-spring-plugin-2.3.16.3.jar
- struts2-tiles3-plugin-2.3.16.3.jar

2.3.3 Prise en compte des nouvelles versions : hornetclient, et les thèmes CSS

2.3.3.1 Rappel

La version de la librairie YUI Gallery précédemment mise à jour sous le serveur apache, est définie dans les sources d'Hornet client (fichier config.js, variable GALLERY_VERSION)

2.3.3.2 Tâche à réaliser : prise en compte nouvelle version

- Modifier le fichier « **envconfig/hornet.properties** », afin de pointer les variables d'environnement pour pointer vers les nouvelles versions des librairies Hornet.

- Ajouter le nouveau paramètre « **combineType** ». Les valeurs autorisées pour ce paramètre sont : « **none** », « **basic** », « **normal** » et « **full** ». Si le paramètre est vide ou non présent, la valeur par défaut utilisée est « **normale** ». Voir le § 4.1 pour plus de détail sur le mode combine.
- Ajouter le nouveau paramètre « **debugHornetClient** ». Les valeurs autorisées pour ce paramètre sont : « **true** », « **false** ». Si le paramètre est vide ou non présent, la valeur par défaut utilisée est « **false** ».

Fonctionnalité	Code Hornet 3.4	Code Hornet 3.6B
Fichier envconfig/hornet.properties		
Nouvelle version de hornetserver-web	<ul style="list-style-type: none">• # Éléments de configuration du framework fwkRoot= http://<URL SERVEUR>/hornetclient/3.4.0/fwkyui3Root= http://<URL SERVEUR>/yui/yui/3.8.1themeName=diplonetthemeVersion=3.4.0	<ul style="list-style-type: none">• # Éléments de configuration du framework fwkRoot= http://<URL SERVEUR>/hornetclient/3.6.1/fwk yui3Root= http://<URL SERVEUR>/yui/yui/3.17.2 themeName=diplonet themeVersion=3.6.0 combineType=normal debugHornetClient=false

2.3.3.3 Vérification

- Lancer votre application Web.
- Ouvrir la page d'accueil à l'aide des outils de développement sous firefox ou chrome avec le panneau réseau. Les fichiers « **css** » ou « **js** » chargés doivent être dans la bonne version
 - > 3.17.2 pour YUI
 - > 3.6.1 pour des fichiers Hornet

2.4 Migration technique de Tiles 2.0.6 vers 3.0.4 GA

Afin de faire fonctionner votre application avec la nouvelle version de Tiles, il est nécessaire d'opérer les modifications décrites dans ce chapitre.

2.4.1 Changement des versions des TLD

Modifier tous les fichiers `tiles*.xml` pour changer le DOCTYPE ; mettre la TLD en version 3 :

```
<!DOCTYPE tiles-definitions PUBLIC "-//Apache Software Foundation//DTD Tiles Configuration 3.0//EN" "http://tiles.apache.org/dtds/tiles-config_3_0.dtd">
```

2.4.2 Adaptation du code Java

- La classe `web\action\frameset\MenuViewPreparer.java` ne compile plus, car la classe `TilesRequestContext` n'existe plus en Tiles 3. On doit utiliser à la place `org.apache.tiles.request.Request` qui est définie dans « tiles-request » (tiles-request-api-x.y.z.jar).

```
/** {@inheritDoc} */  
public void execute(  
    Request tilesContext, AttributeContext attributeContext)  
    throws PreparerException {  
    this.menuService = MenuServiceImpl.getInstance();  
    final MenuVO menu = this.menuService.getMenu();  
    attributeContext.putAttribute(  
        "menuVO", new Attribute(  
            menu));  
}
```

}

- La classe `hornet\framework\test\BaseStrutsTestCase.java` ne compile plus non plus. Il faut changer la classe `BasicTilesContainer` par `HornetTilesContainerFactory` et la classe `StrutsTilesListener` par `HornetTilesListener` comme ci-dessous :

```
final String tilesDefinitions = BaseStrutsTestCase.getInitParams().get(
    HornetTilesContainerFactory.DEFINITIONS_CONFIG);
if (tilesDefinitions != null) {
    BaseStrutsTestCase.servletContext.addInitParameter(
        HornetTilesContainerFactory.DEFINITIONS_CONFIG, tilesDefinitions);
    // Creating the tiles listener statically (not via class
    // loader).
    final HornetTilesListener tilesListener = new HornetTilesListener();
    final ServletContextEvent event = new ServletContextEvent(
        BaseStrutsTestCase.servletContext);
    tilesListener.contextInitialized(event);
}
```

Dans vos tests unitaires, il faut également modifier la méthode `initTests()` par :

```
BaseStrutsTestCase.getInitParams().put("config", "struts.xml");
BaseStrutsTestCase.getInitParams().put(HornetTilesContainerFactory.DEFINITIONS_CONFIG, "../src/config/tiles.xml");
BaseStrutsTestCase.setContextConfiguration("classpath:/appContext-mock.xml");
```

2.4.3 Initialisation de la configuration dans le web.xml

Par défaut, Tiles 3 va chercher le fichier de configuration `tiles.xml` dans `WEB-INF/tiles.xml` alors que Hornet s'attend à le trouver dans `WEB-INF/classes/tiles.xml`. On doit donc utiliser un listener pour Tiles spécifique à « Hornet ».

1/ Dans le web.xml changer le listener mettre `hornet.framework.web.listener.HornetTilesListener` à la place de `org.apache.struts2.tiles.StrutsTilesListener` :

```
<listener>
  <listener-class>hornet.framework.web.listener.HornetTilesListener</listener-class>
</listener>
```

Ce listener permet de charger tous les fichiers de paramétrage Tiles nommés "**tiles*.xml**"

- sous le répertoire WEB-INF et ses sous répertoires (i.e. dans `"/WEB-INF/**/tiles*.xml"`)
- et sous le répertoire META-INF et ses sous répertoires de chaque JAR du classpath (i.e. dans `"classpath*:META-INF/**/tiles*.xml"`)

2/ Supprimer également du web.xml tout le bloc suivant devenu obsolète :

```
<!-- Tiles -->
<context-param>
  <param-name>org.apache.tiles.impl.BasicTilesContainer.DEFINITIONS_CONFIG</param-name>
  <param-value>/WEB-INF/classes/tiles.xml</param-value>
</context-param>
```

2.4.4 Adaptation des JSP

2.4.4.1 Tag `<tiles:useAttribute>`

Dans le fichier `baseLayout.jsp` et dans toutes vos JSP utilisant le tag `<tiles:useAttribute>` faire les modifications suivantes :

Ajouter la déclaration à la tld (en haut de la jsp) :

```
<%@ taglib uri="http://tiles.apache.org/tags-tiles-extras" prefix="tilesx" %>
```

Changer les préfix des 2 lignes suivantes (l. 52 et 53) en remplaçant `tiles` par `tilesx` :

```
<tilesx:useAttribute name="yuiCssItems" scope="request" classname="java.util.List" />
<tilesx:useAttribute name="appCssItems" scope="request" classname="java.util.List" />
```

Modifier aussi de la même manière tous les autres `tiles:useAttribute` présents dans vos JSP.

2.4.4.2 Tag `<tiles:importAttribute>`

A l'exécution de vos JSP, il se peut que vous rencontriez une erreur du type :

`org.apache.tiles.template.NoSuchAttributeException: Error importing attributes.
Attribute 'XXXX' has a null value`

Pour la résoudre, ajouter simplement l'attribut `ignore="true"` au tag `<tiles:importAttribute>` :

```
<tiles:importAttribute name="actionName" ignore="true"/>
```

2.4.5 Adaptation des fichiers de définition « tiles*.xml »

La DTD a été modifiée, notamment sur le(s) point(s) suivant(s) :

```
<put-attribute name="actionName">  
  <definition>edition</definition>edition  
</put-attribute>
```

Qui s'écrit sous la forme plus simple :

```
<put-attribute name="actionName" value="edition" />
```

L'ancienne syntaxe n'est plus supportée, il convient donc de remplacer tous les tags `<put-attribute>` dans les fichiers `tiles*.xml` pour respecter cette nouvelle forme d'écriture.

2.4.6 Lancement de l'application

En cas d'erreur `org.apache.tiles.jsp.taglib.InsertAttributeTag cannot be cast to javax.servlet.jsp.tagext.Tag` sur les JSP, il faut vider le work de tomcat.

Pour cela, sous Eclipse, dans l'onglet « **Servers** » faire bouton droit sur le serveur Tomcat puis « **Clean Tomcat Work Directory...** »

3 Mise à jour structurelle

Il convient de modifier les fichiers communs aux projets qui sont à l'origine générés par hornettemplate.

Ce chapitre décrit les modifications à apporter sur ces fichiers.

➤ **Autre démarche : Les fichiers décrits dans ce chapitre peuvent être générés par hornettemplate 3.6.1 puis être copiés dans votre projet. Si des développements spécifiques ont été réalisés sur ces fichiers, ils seront écrasés. Les spécificités seront éventuellement à reporter dans les nouvelles pages.**

A priori présent dès la création de votre application, ces fichiers communs ont pu être modifiés, supprimés ou découpés. Dans ce cas, vous devrez adapter les préconisations de ce chapitre.

3.1 Ressources statiques

Le framework « 2in3 » permettant l'utilisation de composant YU2 a été complètement supprimé, le framework hornetclient n'utilise plus aucun composant en version YUI2. Le framework « 2in3 » n'est donc plus disponible.

3.1.1 Fichiers JavaScript

3.1.1.1 *Tâche à réaliser : migration des fichiers javascript généré par hornettemplate*

Mettre à jour les ressources statiques dans le répertoire : **/WebContent/static/js/**

Fichier concerné :

- **base.js**
- **form.js**
- **tableau.js**

Impacts :

- Suppression des références à Y.YUI2
- Modification de l'instanciation du tableau hornet

Fonctionnalité	Code Hornet 3.4	Code Hornet 3.6B
Fichier WebContent\static\js\tableau.js		
Modification de l'instanciation d'un tableau. Suppression référence YUI2 Mise à jour javadoc	<pre> • * - ds {YAHOO.widget.Datasource} la datasource creee * - dt {YAHOO.widget.DataTable} la datatable creee */ function initDatatable(Y, tableId, containerId, columnDefs, config) { var YAHOO = Y.YUI2, oDs = null, oDt = null; //tableau html initial present seulement si le tableau yui n'est pas déjà cree var tableNode = Y.one("table#" + tableId); if (tableNode) { oDs = new YAHOO.util.DataSource(YAHOO.util.Dom.get(tableId)); oDs.responseType = YAHOO.util.DataSource.TYPE_HTMLTABLE; oDs.responseSchema = { fields: columnDefs }; oDt = Y.hornet.PaginTable.createTable(containerId, columnDefs, oDs, config); } return { ds : oDs, dt : oDt }; } </pre>	<pre> • * - ds {Y.hornet.PEDataSource} la datasource creee * - dt {Y.DataTable} la datatable creee */ function initDatatable(Y, tableId, containerId, columnDefs, config) { oDs = null, oDt = null; //tableau html initial present seulement si le tableau yui n'est pas déjà cree var tableNode = Y.one("table#" + tableId); if (tableNode) { oDs = Y.hornet.PEDataTable.createDatasource(null, columnDefs, "item", {table: tableId}); oDt = Y.hornet.PEDataTable.createTable(containerId, columnDefs, oDs, config); } return { ds : oDs, dt : oDt }; } </pre>
Fichier WebContent\static\js\form.js		
Suppression référence YUI2	<pre> • function createForm(Y, formId, config) { var YAHOO = Y.YUI2; var formClass = Y.Base.create ('formClass', Y.Base, [], </pre>	<pre> • function createForm(Y, formId, config) { var formClass = Y.Base.create ('formClass', Y.Base, [], </pre>
Fichier WebContent\static\js\base.js		

Modification de l'instanciation d'un tableau. Suppression référence YUI2 Mise à jour javadoc	<pre> function createButton(Y, node) { var YAHOO = Y.YUI2, id = node.get('id'); if (!id) { id = Y.guid(); node.set('id', id); } var button = new YAHOO.widget.Button(id), buttonNode = Y.one('#' + id); // Recopie des classes css et du titre s'ils existent var buttonElement = Y.one('#' + id + '-' + 'button'); if (buttonElement) { var title = node.get('title'), className = node.get('className'); if (!!title) { buttonElement.set('title', title); } if (!!className) { buttonElement.addClass(className); } } return buttonNode; } </pre>	<pre> /** * Fonction pour creer un bouton widget appliquant * les styles et les evenements * associes. */ function createButton(Y, node) { var id = node.get('id'), // title = node.get('title'), // className = node.get('className'), // _button = node.one('>.first-child>*'), // newButton = false; if (!id) { id = Y.guid(); node.set('id', id); } if (!_button) { newButton = true; } new Y.Button({ srcNode : _button, }).render(); node = Y.one('#' + id); _button = node.one('>.first-child>*'); if (newButton && _button) { // Recopie des classes css et du titre sur le sous-element s'ils existent if (!!title) { _button.set('title', title); } if (!!className) { _button.addClass(className); } } return node; } </pre>
Mise à jour du corps de la méthode et de la javadoc	<pre> /** * fonction pour creer le style des boutons sur * les elements ayant la classe "bouton" */ function initBoutons(Y, node) { Y.all(".bouton").each(function(node){ createButton(Y, node, node.get('title')); }); } </pre>	<pre> /** * Fonction pour creer des boutons a partir des * elements identifiés par la * selection. Si des evenements sont rattaches aux * elements, ils ne seront pas * conserves. */ function initBoutons(Y, selection) { Y.all(selection).each(function(node) { createButton(Y, node); }); } </pre>

3.1.2 Fichiers CSS

3.1.2.1 Tâche à réaliser : changement de nom de fichiers CSS dans la configuration de Tiles

Les fichiers « **reset-min.css** », « **base-min.css** », « **fonts-min.css** » et « **grids-min.css** » de YUI ont été renommés. Leurs nouveaux noms sont désormais préfixé par « **css** » (exemple « **cssreset-min.css** »). Il convient donc de mettre à jour le fichier « **tiles.xml** » avec les nouvelles valeurs suivantes :

Fonctionnalité	Code Hornet 3.4	Code Hornet 3.6B
Fichier src\config\tiles.xml		
Changement de nom	<ul style="list-style-type: none"> <code><put-list-attribute name="yuiCssItems"></code> <code><add-attribute value="/cssreset/reset-min.css" /></code> <code><add-attribute value="/cssbase/base-min.css" /></code> <code><add-attribute value="/cssfonts/fonts-min.css" /></code> <code><add-attribute value="/cssgrids/grids-min.css" /></code> <code></put-list-attribute></code> 	<ul style="list-style-type: none"> <code><put-list-attribute name="yuiCssItems"></code> <code><add-attribute value="/cssreset/cssreset-min.css" /></code> <code><add-attribute value="/cssbase/cssbase-min.css" /></code> <code><add-attribute value="/cssfonts/cssfonts-min.css" /></code> <code><add-attribute value="/cssgrids/cssgrids-min.css" /></code> <code></put-list-attribute></code>

3.1 JSP

3.1.1 JSP hornetConfig.jsp

3.1.1.1 Tâche à réaliser : prise en compte des nouveaux noms des modules

Modifier le fichier « `WebContent\WEB-INF\tiles-jsp\layout\hornetConfig.jsp` », afin de modifier les modules utilisés.

- Remplacer la référence « `yui2-button` » par « `button` »
- Remplacer la référence « `hornet-pagintable` » par « `hornet-datatable` »
- Supprimer la référence à « `yui2-container` »

Fonctionnalité	Code Hornet 3.4	Code Hornet 3.6B
Fichier WebContent\WEB-INF\tiles-jsp\layout\hornetConfig.jsp		
Mise à jour du nom des modules	<ul style="list-style-type: none"> <code>appModules = {</code> <code> "hornet_base" : {</code> <code> type: "js", fullpath :</code> <code> "\${pageContext.request.contextPath}/static/js/base.js",</code> <code> requires : ['event', 'node', 'yui2-button', 'node-menunav'] },</code> <code> "hornet_form" : {</code> <code> type: "js", fullpath :</code> <code> "\${pageContext.request.contextPath}/static/js/form.js",</code> <code> requires : ['io-form', 'yui2-container', 'hornet-ajax'] },</code> <code> "hornet_tableau" : {</code> <code> type: "js", fullpath :</code> <code> "\${pageContext.request.contextPath}/static/js/tableau.js",</code> <code> requires : ['event', 'node', 'hornet-pagintable'] },</code> <code> "appModules" : {</code> <code> use: ['hornet_base', 'hornet_form', 'hornet_tableau'] }</code> <code>};</code> 	<ul style="list-style-type: none"> <code>appModules = {</code> <code> "hornet_base" : {</code> <code> type: "js", fullpath :</code> <code> "\${pageContext.request.contextPath}/static/js/base.js",</code> <code> requires : ['event', 'node', 'button', 'node-menunav'] },</code> <code> "hornet_form" : {</code> <code> type: "js", fullpath :</code> <code> "\${pageContext.request.contextPath}/static/js/form.js",</code> <code> requires : ['io-form', 'hornet-ajax'] },</code> <code> "hornet_tableau" : {</code> <code> type: "js", fullpath :</code> <code> "\${pageContext.request.contextPath}/static/js/tableau.js",</code> <code> requires : ['event', 'node', 'hornet-datatable'] },</code> <code> "appModules" : {</code> <code> use: ['hornet_base', 'hornet_form', 'hornet_tableau'] }</code> <code>};</code>

3.1.1.2 Tâche à réaliser : prise en compte du paramètre

Modifier le fichier « `WebContent\WEB-INF\tiles-jsp\layout\hornetConfig.jsp` », afin d'y ajouter les lignes suivantes (à placer avant l'instruction `hornet(yuiConfig)`) :

```

yuiConfig.groups.appModules = {
  base: '${pageContext.request.contextPath}/static/js/',
  combine: false, filter: 'RAW', modules: appModules };

yuiConfig.hornetCombineType='${application.combineType}';

<c:if test="${application.debugHornetClient=='true'}">
  yuiConfig.debug=true;
  yuiConfig.filter='debug';
</c:if>
    
```

```
//Initialise une instance yui avec la configuration personnalisée  
hornet(yuiConfig);
```

3.1.2 JSP baseLayout.jsp

3.1.2.1 Tâche à réaliser : migration du « baseLayout.jsp » pour le mode combine

Si vous n'avez pas modifié le fichier « **baseLayout.jsp** » remplacer le simplement par sa nouvelle version fournie dans hornettemplate ; sinon appliquer les points ci-dessous pour l'adapter.

1/ Modifier le fichier « **WebContent\WEB-INF\tiles-jsp\layout\baseLayout.jsp** », afin d'y ajouter/modifier les blocs suivants :

```
<!-- mode debug -->  
<c:choose>  
  <c:when test="\${application.debugHornetClient=='true'}">  
    <c:set var="jsFilter" value="-debug"/>  
    <c:set var="cssFilter" value="" />  
  </c:when>  
  <c:otherwise>  
    <c:set var="jsFilter" value="-min"/>  
    <c:set var="cssFilter" value="-min"/>  
  </c:otherwise>  
</c:choose>  
  
<!-- theme -->  
<c:if test="\${empty themeName}">  
  <c:set var="themeName" value="\${application.themeName}"/>  
  <c:set var="themeVersion" value="\${application.themeVersion}"/>  
</c:if>  
<c:choose>  
  <c:when test="\${!empty themeName and empty themeVersion}">  
    <c:set var="themeClass" value="hornet-skin-\${themeName}"/>  
    <c:set var="themeComponent" value="hornet-skin-\${themeName}"/>  
  </c:when>  
  <c:when test="\${!empty themeName and !empty themeVersion}">  
    <c:set var="themeClass" value="hornet-skin-\${themeName}"/>  
    <c:set var="themeComponent" value="hornet-skin-\${themeName}-\${themeVersion}"/>  
  </c:when>  
  <c:otherwise>  
    <c:set var="themeClass" value="hornet-skin-default"/>  
    <c:set var="themeComponent" value="hornet-skin-default"/>  
  </c:otherwise>  
</c:choose>  
<c:set var="themeNameCombine" scope="page" value="\${themeName}-xcombine"/>  
<c:choose>  
  <c:when test="\${!empty themeNameCombine and empty themeVersion}">  
    <c:set var="themeClassCombine" value="hornet-skin-\${themeNameCombine}"/>  
    <c:set var="themeComponentCombine" value="hornet-skin-\${themeNameCombine}"/>  
  </c:when>  
  <c:when test="\${!empty themeNameCombine and !empty themeVersion}">  
    <c:set var="themeClassCombine" value="hornet-skin-\${themeNameCombine}"/>  
    <c:set var="themeComponentCombine" value="hornet-skin-\${themeNameCombine}-\${themeVersion}"/>  
  </c:when>  
  <c:otherwise>  
    <c:set var="themeClassCombine" value="hornet-skin-default-xcombine"/>  
    <c:set var="themeComponentCombine" value="hornet-skin-default-xcombine"/>  
  </c:otherwise>  
</c:choose>
```

2/ Au niveau de la génération du header de la page html, rajouter les balises conditionnelles <c:if> suivantes :

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />  
<title><c:if test="\${pageErreur}">\${errorTitleLabel} - </c:if><c:if  
test="\${!fn:contains(pageTitleLabel, '???')}">\${pageTitleLabel} - </c:if>\${titleLabel}</title>  
  
<c:if test="\${application.combineType=='none'}">  
  <c:forEach items="\${yuiCssItems}" var="item">  
    <link rel="stylesheet" type="text/css" href="\${application.yui3Root}\${item}' /> />  
  </c:forEach>  
</c:if>  
<c:forEach items="\${appCssItems}" var="item">
```

```

<link rel="stylesheet" type="text/css" href="<c:url value='${item}' />" />
</c:forEach>

<c:if test="${application.combineType=='none'}">
  <c:if test="${!empty application.themeDefaultPath}">
    <link rel="stylesheet" type="text/css" href="<c:url
value='${application.themeDefaultPath}/base${cssFilter}.css' />" />
  </c:if>
  <c:if test="${!empty themeComponent and !empty application.themeRoot}">
    <link rel="stylesheet" type="text/css" href="<c:url
value='${application.themeRoot}/${themeComponent}/theme${cssFilter}.css' />" />
  </c:if>
</c:if>
<c:if test="${application.combineType!='none'}">
  <link rel="stylesheet" type="text/css" href="<c:url
value='${application.themeRoot}/${themeComponentCombine}/${themeComponentCombine}${cssFilter}.css' />" />
</c:if>
<script type="text/javascript" src="<c:url value='${application.yui3Root}/yui/yui${jsFilter}.js' />"
></script>
<script type="text/javascript" src="<c:url
value='${application.fwkRoot}/hornetconfig/hornetconfig${jsFilter}.js' />" ></script>
    
```

Ceci permet de supprimer les chargements unitaire des fichiers de styles CSS dès lors que le paramètre « **combineType** » est différent de « **none** ».

3.2 Tiles 3

3.2.1 Impacts

Fonctionnalité	Code Hornet 3.4	Code Hornet 3.6B
Fichier src\config\tiles.xml		
Définition put-attribute	<put-attribute name="actionName">ajouter</put-attribute>	<put-attribute name="actionName" value=" ajouter" />
Fichier WebContent\WEB-INF\web.xml		
context-param	<pre> <context-param> <param- name>org.apache.tiles.impl.BasicTilesContainer.D EFINITIONS_CONFIG</param-name> <param-value>/WEB- INF/classes/tiles.xml</param-value> <param-value>/WEB-INF/classes/tiles- fragment1.xml</param-value> <param-value>/WEB-INF/classes/tiles- fragment2.xml</param-value> </context-param> </pre>	Supprimé La configuration est prise en charge par le nouveau listener HornetTilesListener
Listener	<pre> <listener> <listener- class>org.apache.struts2.tiles.StrutsTilesListen er</listener-class> </listener> </pre>	<pre> <listener> <listener- class>hornet.framework.web.listener.HornetTilesList ener</listener-class> </listener> </pre>

3.2.2 Wildcards

La migration de Tiles 2 vers Tiles 3 permet l'utilisation de wildcard (« * ») :

- Pour la détection des fichiers de définition de tiles (fichiers « **tiles-*.xml** »)
- Pour la définition des tiles en elles-mêmes

Toutefois, vous n'êtes pas obligé d'utiliser le mode de définitions de tiles via wildcard.

Pour plus d'information sur l'utilisation des wildcard Tiles, ce référer au document « **HORNET_GUI_Guide du développeur Hornet 3.6B** ».

4 Mise à jour technique

4.1 Ajout d'un mode Combine (ou Combo)

4.1.1 Présentation

Ce mode permet de combiner et de minimiser les ressources JavaScript et CSS en paquetage moins finement découpés, augmentant ainsi la vitesse de chargement des pages.

L'objectif principal est de réduire le nombre de requêtes http produites par le navigateur pour le chargement d'une page donnée.

Ce type d'optimisation des performances fait partie des « best practices » incluses dans la catégorie « Minimizing round-trip times — reducing the number of serial request-response cycles » établie par Google pour l'obtention d'un web plus rapide.

4.1.2 Combos disponibles

4.1.2.1 Combos CSS

Les combos CSS sont disponibles sous forme de thèmes (1 combo par thème). Ils sont présents dans le répertoire `/hornetclient/<X.Y.Z>/themes/hornet-skin-<AAAA-x.y.z>` sur le serveur de framework. Leurs noms techniques sont simplement postfixés par « -xcombine ».

Thème	Nom skin « classique »	Nom skin « combo »
Addulact (default)	hornet-skin-default	hornet-skin-default-xcombine
MAEDI Intranet (diplonet)	hornet-skin-diplonet[-x.y.z]	hornet-skin-diplonet-xcombine[-x.y.z]
MAEDI Internet (francediplo)	hornet-skin-francediplo[-x.y.z]	hornet-skin-francediplo-xcombine[-x.y.z]

Le choix du skin est opérée automatiquement à partir des propriétés du fichier « **hornet.properties** » :

- themeName
- themeVersion
- comboType

4.1.2.2 Combos JS

Les combos JS sont au nombre de 3 : « **basic** », « **normal** », « **full** ». Techniquement, on trouve un module hornet par combo.

Combo JS	Nom module hornet	Description
Basic	hornet-xcombine-js-basic	Contient <ul style="list-style-type: none">• Tous les fichiers JS Hornet• Tous les fichiers JS Gallery utilisés par hornet• Les modules YUI de base
Normal	hornet-xcombine-js-normal	Contient <ul style="list-style-type: none">• Tous les fichiers JS Hornet• Tous les fichiers JS Gallery utilisés par hornet• Les modules YUI permettant de faire fonctionner l'applitutoriel

Full	hornet-xcombine-js-full	Contient <ul style="list-style-type: none">Tous les fichiers JS HornetTous les fichiers JS Gallery utilisés par hornetTous les modules YUI
------	-------------------------	---

Note : Si l'application utilise un module YUI qui n'est pas présent initialement dans le combo, le module est chargé dynamiquement par le YUI Loader de la même manière que lorsque le mode Combine n'est pas activé.

4.1.3 Mode par défaut

L'utilisation du mode Combine est activée par défaut à partir de la version d'Hornet 3.6B. Ceci permet de faire bénéficier par défaut aux applications des gains de performance induits par l'utilisation de ce mode d'optimisation.

Le combo par défaut utilisé est le combo « **normal** ». Toutefois, il reste possible de le désactiver explicitement (via `comboType=none`).

4.1.4 Choix du combo

Le choix du combo est réalisé par le paramètre « **comboType** » du fichier « `hornet.properties` ».

Les valeurs possibles pour ce paramètre sont :

- **basic** : choix du combo JS "basic" et activation du combo CSS du thème « `themeName` »
- **normal** : choix du combo JS "normal" et activation du combo CSS du thème « `themeName` »
- **full** : choix du combo JS "full" et activation du combo CSS du thème « `themeName` »
- **none** : désactivation du mode combo (CSS et JS)

La valeur par défaut appliquée (quand le paramètre « `comboType` » est vide ou non présent) est « `normal` ».

4.2 Activer le debug de YUI et HornetClient

Ceci est réalisé par le nouveau paramètre « **debugHornetClient** » du fichier « `hornet.properties` ». Les valeurs possibles pour ce paramètre sont « **true** » et « **false** ». La valeur par défaut appliquée est « **false** ».

4.3 Composant Tableau

4.3.1 Composant « **Datatable** » remplaçant « **Pagintable** »

Type : Nouveau composant.

Le composant « **Pagintable** » basé sur YUI2 est supprimé. A la place, il convient d'utiliser le composant « **Datatable** » qui se base sur une implémentation en YUI3. Le module à inclure pour pouvoir utiliser ce nouveau composant est 'hornet-datatable'. Il déclare la classe « **hornet.PEDdataTable** » (PE pour « **Progressive Enhancement** ») qui remplace la classe « **hornet.PaginTable** ».

La classe de DataSource qui lui est associée « **hornet.PEDatasource** » est la remplaçante de « **YAHOO.hornet.HijaxDataSource** » qui est également supprimée.

L'utilisation de ce composant est relativement similaire à l'ancien. La signature des méthodes « **createDataSource** » et « **createTable** » a été conservé pour favoriser la retrocompatibilité. Cependant celles-ci n'acceptent pas forcément les mêmes paramètres de configuration.

Se référer à la documentation de l'API hornet (YUIDoc) et au document « **HORNET_GUI_Guide du développeur Hornet 3.6B** » pour en savoir plus.

4.3.1.1 Tâche à réaliser : modifier le code manipulant les tableaux Hornet

Tableau 1 : Migration vers Hornet 3.6B - Equivalences pour le composant « Tableau »

Fonctionnalité	Code Hornet 3.4	Code Hornet 3.6B
Module (dépendances)	hornet().use("hornet-pagintable", ...	hornet().use("hornet-datatable", ...
Instanciation de la datasource	<pre>// DataSource instance myDataSource = Y.hornet.PaginTable.createDatasource(urlRecherche, myColumnDefs, "item", {table: tableId}, configNotification);</pre>	<pre>// DataSource instance myDataSource = Y.hornet.PEDataTable.createDatasource(urlRecherche, myColumnDefs, "item", { table: tableId }, configNotification); >>Utiliser la classe Y.hornet.PEDataTable à la place de Y.hornet.PaginTable</pre>
Instanciation du datatable	<pre>// DataTable instance myDataTable = Y.hornet.PaginTable.createServerTable(tableauCon tainerId, myColumnDefs, myDataSource, tableConfig);</pre>	<pre>// DataTable instance myDataTable = Y.hornet.PEDataTable.createServerTable(tableauConta inerId, myColumnDefs, myDataSource, tableConfig); >>Utiliser la classe Y.hornet.PEDataTable à la place de Y.hornet.PaginTable</pre>
Callback après retour des données	<pre>myDataTable.subscribe("dataReturnEvent", function(e, oSelf){ ... });</pre>	<pre>myDataSource.xhrDatasource.after("response", function(e) { ... }); OU BIEN myDataTable.datasource.get("datasource").after("res ponse", function(e) { ... }); >>Utiliser myDataSource obtenue par la méthode Y.hornet.PEDataTable.createDatasource. On ne peut pas utiliser directement « myDataTable.datasource.after(...) » qui ne fonctionnera pas ; il faut faire à la place un « myDataTable.datasource.get("datasource").after(...) »</pre>
Chargement des données pour un datatable	myDataTable.load();	myDataTable.datasource.load();
Changement de source	<pre>myDataSource.liveData="successInfo.xml"; myDataTable.load();</pre>	<pre>myDataSource.setLiveData("successInfo.xml"); myDataTable.datasource.load();</pre>
Réinitialisation du tableau	myDataTable.initializeTable()	Pas d'équivalent, la fonction load() est auto-suffisante
Datasource HtmlTable	<pre>var myColumnDefs = [{ key: "nom", label: "Nom", sortable:true }]; var tableau = Y.one("table[id='myTable']"); myLocalDataSource = new YAHOO.util.DataSource(tableau.getDOMNode()); myLocalDataSource.responseType = YAHOO.util.DataSource.TYPE_HTMLTABLE; myLocalDataSource.responseSchema = { fields: myColumnDefs };</pre>	<pre>var tableau = Y.one("table[id='myTable']"); myLocalDataSource = new Y.DataSource.Local({source:tableau}); myLocalDataSource.plug(Y.Plugin.DataSourceXMLSchema , { schema: { resultListLocator: "tr", resultFields: [{key:"nom", locator:"td[1]"}] });</pre>

Ajout des critères de filtrage	<pre>myDataTable.getDataSource().connMgr.setForm(formTableNode.getDOMNode()); myDatatable.load();</pre>	<pre>var myRequest = Y.IO.stringify(formTableNode.getDOMNode()); myDataTable.datasource.load({request: myRequest});</pre> <p>>> passer par le module io-form qui permet de sérialiser les données d'un formulaire HTML via la méthode stringify().</p>
Configuration pour activer la pagination avec \${itemsParPage} lignes par page	<pre>// Configuration de la pagination var paginatorConfig = { rowsPerPage : \${itemsParPage}, containers:["pag-below"], alwaysVisible: false }; // Configuration du tableau var tableConfig = { tableName : "partenaires", paginator : Y.hornet.PaginTable .createPagination(paginatorConfig.rowsPerPage 10, paginatorConfig) };</pre>	<pre>// Configuration du tableau & de la pagination var tableConfig = { tableName : "partenaires", rowsPerPage : \${itemsParPage} };</pre> <p>>> La création d'un objet de pagination via Y.hornetPaginTable.createPagination n'est plus nécessaire, il suffit simplement d'initialiser l'attribut rowsPerPage dans la configuration du tableau</p>
Tableau éditable : parsing du détail d'un objet	<pre>data_out = Y.DataSchema.XML.apply(schema_update, response.responseXML);</pre>	<pre>data_out = Y.DataSchema.XML.apply(schema_update, Y.DataType.XML.parse(response.responseText));</pre> <p>>> Important !!!! pour une compatibilité avec IE11 ne pas utiliser directement la propriété response.responseXML.</p>
Tableau éditable : modification d'une ligne	<pre>gDataTable.updateRow(...);</pre>	<pre>gDataTable.modifyRow(...);</pre>
Tableau éditable : suppression d'une ligne	<pre>tableEditObject.dataTable.deleteRow(currentRowId);</pre>	<pre>tableEditObject.dataTable.removeRow(row);</pre> <p>>> Important !!! Transmettre directement l'objet « ligne » du tableau à supprimer et non son index comme précédemment qui dépend de la page courante de la pagination.</p>
Tableau éditable : accès aux données d'une ligne	<pre>var row = tableEditObject.dataTable.getRecord(line); var currentEditRowId = row.getId(); var nomSecteur = row.getData("nomSecteur")</pre>	<pre>var row = tableEditObject.dataTable.getRecord(e.target); var currentEditRowId = row.get('id'); var nomSecteur = row.get("nomSecteur")</pre>

4.4 Composant Panel

4.4.1 Tâche à réaliser : modifier le code des Popin

La classe Dialog est remplacée par la classe Panel.

Tableau 2 : Migration vers Hornet 3.6B - Equivalences pour le composant « Popin »

Fonctionnalité	Code Hornet 3.4	Code Hornet 3.6B
Module (dépendances)	hornet().use("yui2-container", ...	hornet().use("panel", ...
Dialog -> Panel	<p>JSP :</p> <pre><div id="panelForm" class="hornet-hidden"> <div class="hd"> <h2><tiles:getAsString ignore="true" name="form_title"/></h2> </div> <div class="bd"> <tiles:insertAttribute name="form"/> </div> </div></pre> <p>JavaScript :</p> <pre>hornet().use("hornet_tableau", "hornet_form", function (Y) { var YAHOO = Y.YUI2, // ... //Declaration du formulaire en modal tableEditObject.dialogEditForm = new YAHOO.widget.Dialog(panelFormNode.get('id'), { modal: true, close: true, visible: false, fixedcenter: true, constraintviewport: true, width: "600px", postmethod: "none", hideaftersubmit : false }); }</pre>	<p>JSP :</p> <pre><div id="panelForm" class="hornet-hidden"> <div class="bd yui3-widget-bd"> <tiles:insertAttribute name="form"/> </div> </div></pre> <p>JavaScript :</p> <pre>hornet().use("hornet_tableau", "hornet_form", "panel", function (Y) { // ... //Declaration du formulaire en modal tableEditObject.dialogEditForm = new Y.Panel({ contentBox : panelFormNode, modal: true, close: true, visible: false, centered : true, width: "600px", }); }</pre> <p>Pour plus d'info cf. https://yuilibrary.com/yui/docs/api/classes/Panel.html</p>
Changement du titre de la fenêtre	tableEditObject.dialogEditForm.setHeader("<h2>..</h2>");	tableEditObject.dialogEditForm.set("headerContent", "<h2>...</h2>"); >> On procède de la même manière pour changer le body
Annulation (fermeture de la fenêtre)	tableEditObject.dialogEditForm.cancel();	tableEditObject.dialogEditForm.hide(); La méthode cancel() n'existe plus, utiliser hide() à la place.

4.5 Composant Calendrier

4.5.1.1 Vérification

Les méthodes publiques du calendrier sont les mêmes entre les deux versions, il n'y a donc pas de migration.

Malgré cela, le code interne au calendrier a été très impacté par le changement de version.

- Vérifier tous les calendriers de votre application

4.6 Composant Bouton

4.6.1.1 *Rappel*

Dans les jsp, les tags struts génèrent du code HML à travers des fichiers « .ftl ». Ces fichiers « .ftl » sont définis dans le Framework Hornet Server.

La nouvelle version hornetserver 3.6.1 modifie le flux HTML généré pour les boutons.

4.6.1.2 *Vérification*

Avant le changement de version, le HTML généré contenait 2 balises . Ces balises SPAN ne sont plus écrites dans le flux.

- Vérifier tous les boutons de votre application car le chemin dans le DOM est différent (impact possible sur les CSS ou le javascript)