

Web 2.0 : EJB + Dojo + Webservices

COUTABLE Rémy
JAGUELIN François
MERCIER Julien

M2 MIAGE NTDP Nice - Promotion 2010

I. Sources

Vous pouvez trouver les sources du projet ici : http://github.com/Rymai/icms_final

Vous pouvez le télécharger directement en cliquant sur le bouton “Download” sur cette même page.

II. Modification de l'architecture

Nous avons modifier l'architecture concernant l'arborescence des pages. Auparavant, nous avions 3 types de pages (Category, Section, Article) héritant toutes de la classe Page. Cela posait un problème d'arborescence puisque celle-ci était limitée à 3 niveaux seulement.

Nous avons donc modifié ceci en ne gardant que la classe Page et en ajoutant un attribut myParent qui sert de pointeur vers la page parente.

En pratique :

1. une page sans page parente est de type “root” (niveau 0)
2. une page avec une page parente et sans pages filles est de type “article” (niveau n)
3. une page avec une page parente et des pages filles est de type “section” ($0 < \text{niveau} < n$)

III. Intégration de webservices

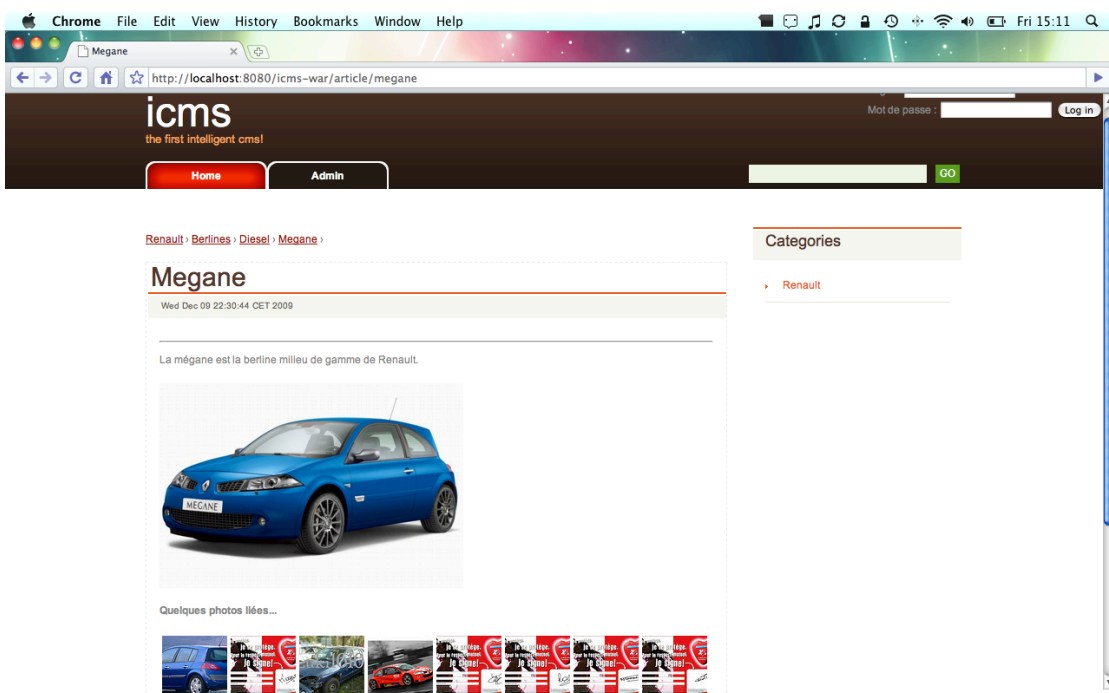
Classes : `icms-war/src/java/icms_helper/[Flickr|Translate]Helper.java`

Flickr

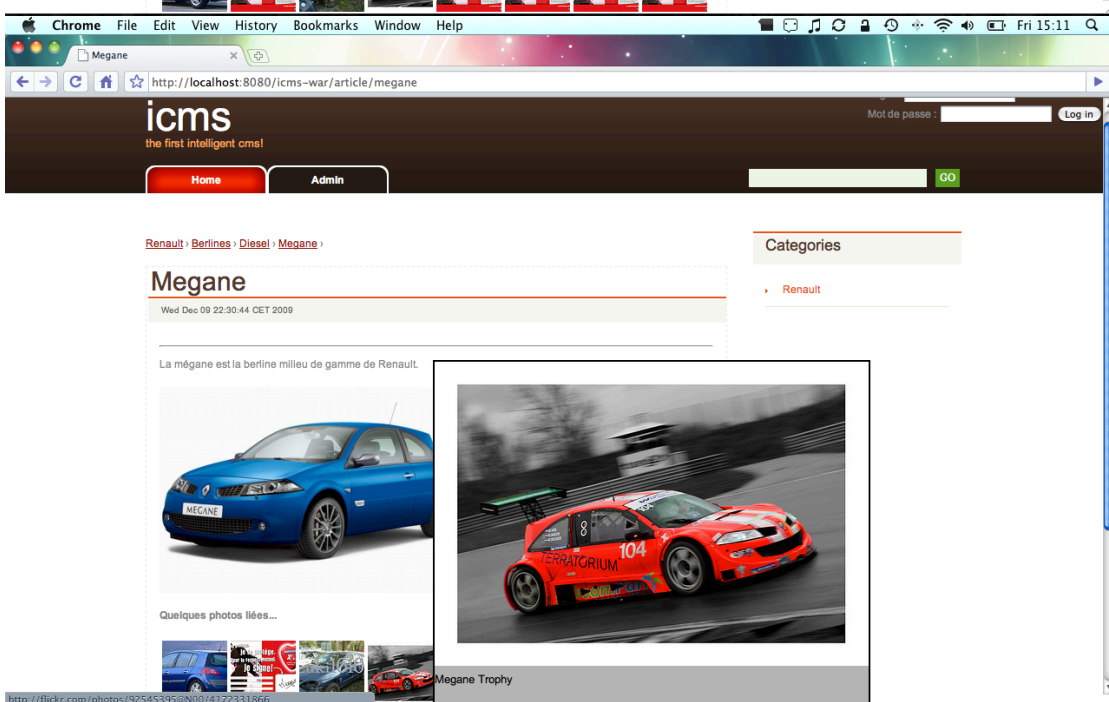
Nous avons intégré Flickr dans nos pages en affichant des images en rapport avec les informations d'une page.

La méthode pour définir les mots-clés à rechercher est agile et permet de s'assurer au maximum d'avoir des images à afficher :

- La première recherche de mots clés s'effectue sur le titre de la page et celui de sa page parente.
- Si aucun résultat n'est trouvé, la recherche continue sur les niveaux supérieurs (le titre de la page parente et le titre de la page parente de celle-ci)
- Et ainsi de suite jusqu'à la page parente de niveau 0.



Les images trouvées sur Flickr s'affichent à la fin de l'article.

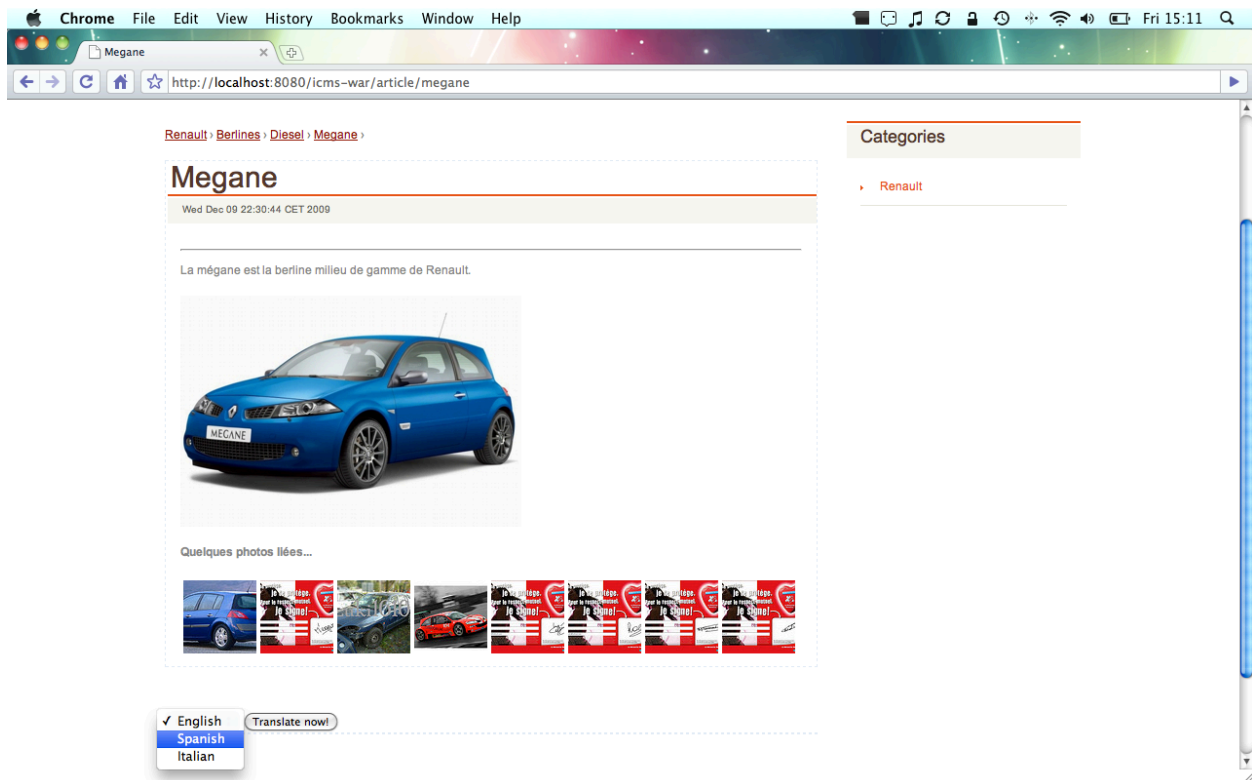


Une version plus grande de chaque image s'affiche lors du survol d'une miniature (avec Dojo Tooltip).

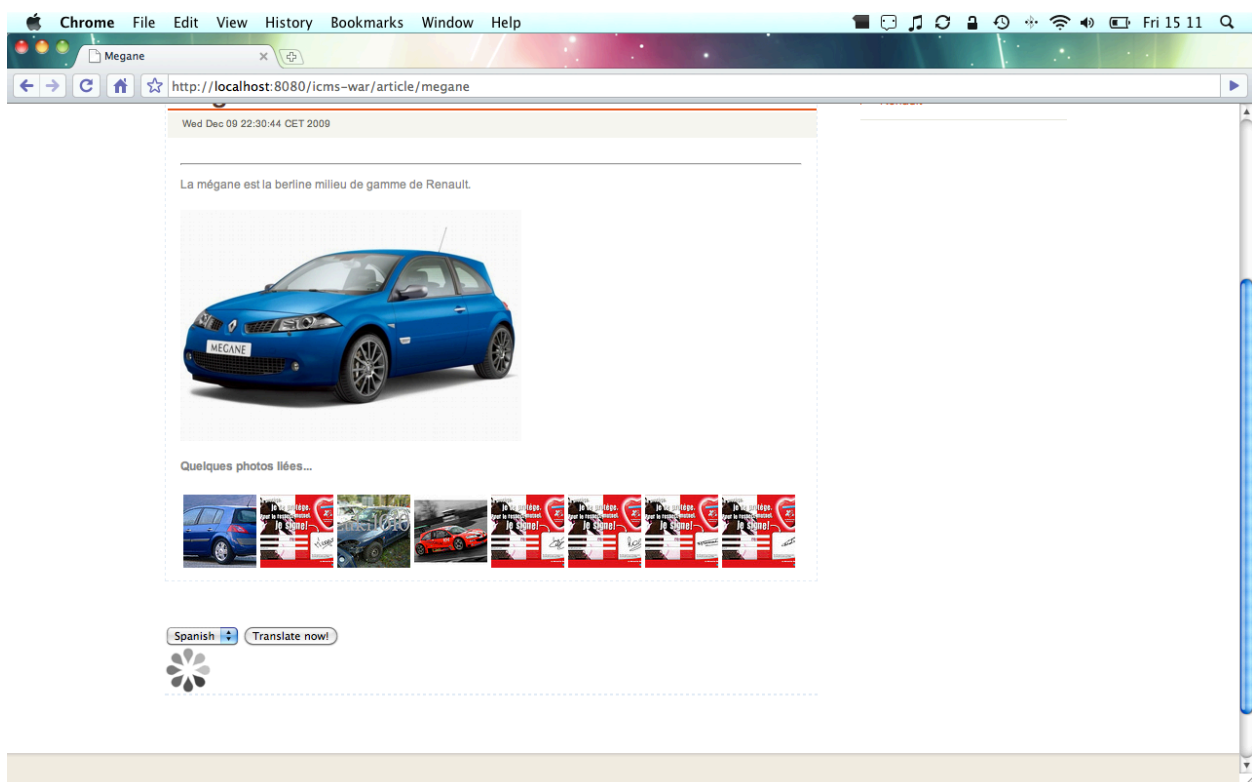
Translation

Nous avons intégré un webservice SOAP de traduction vu en cours de “Services Web” pour permettre la traduction en direct d’un article (anglais, espagnol et italien pour l’expérimentation) :

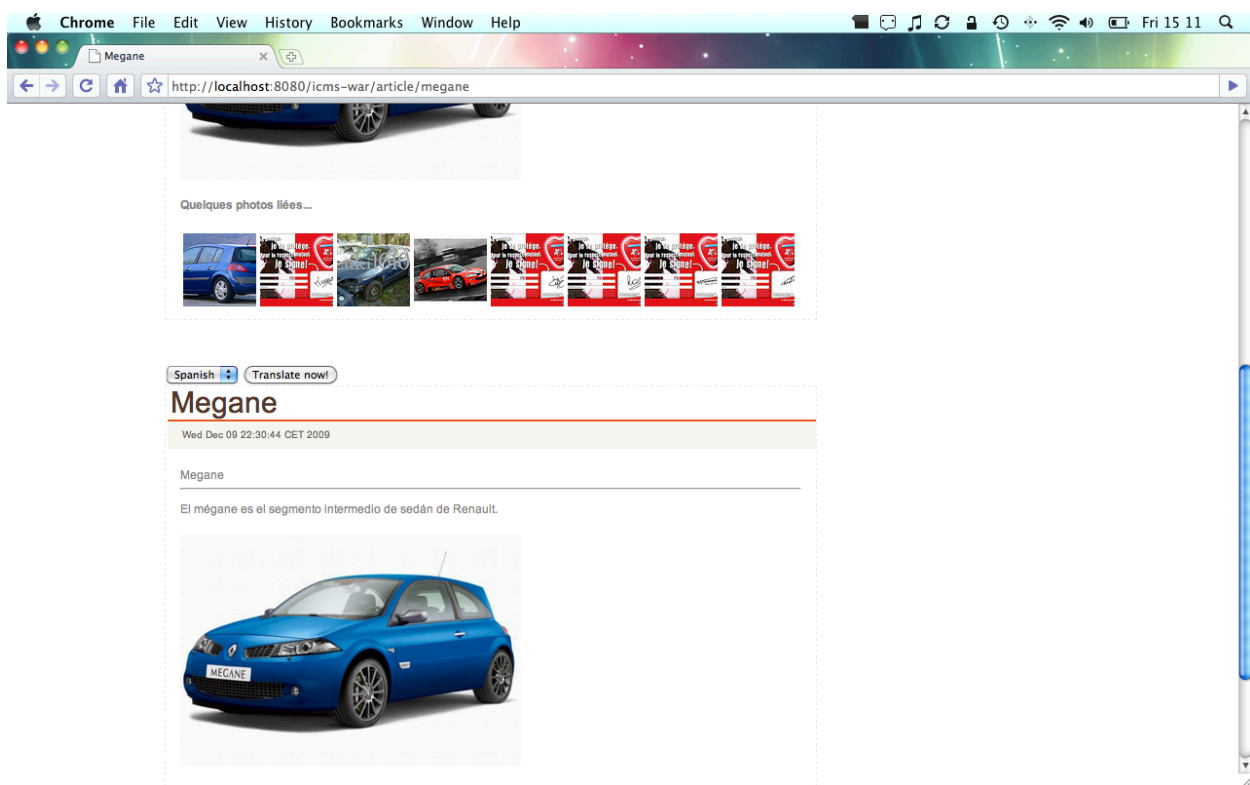
1. Choix de la langue de destination



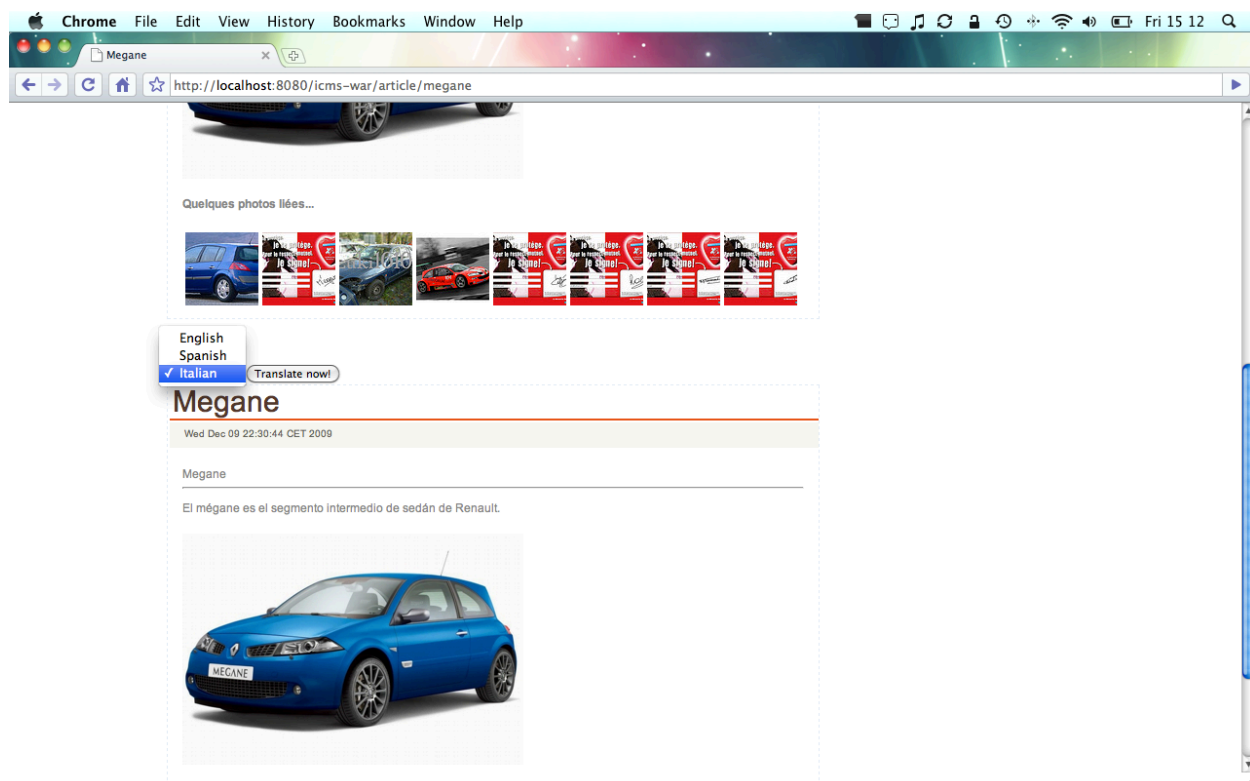
2. Lancement de la traduction, apparition d’un spinner pour indiquer que la traduction est en cours.



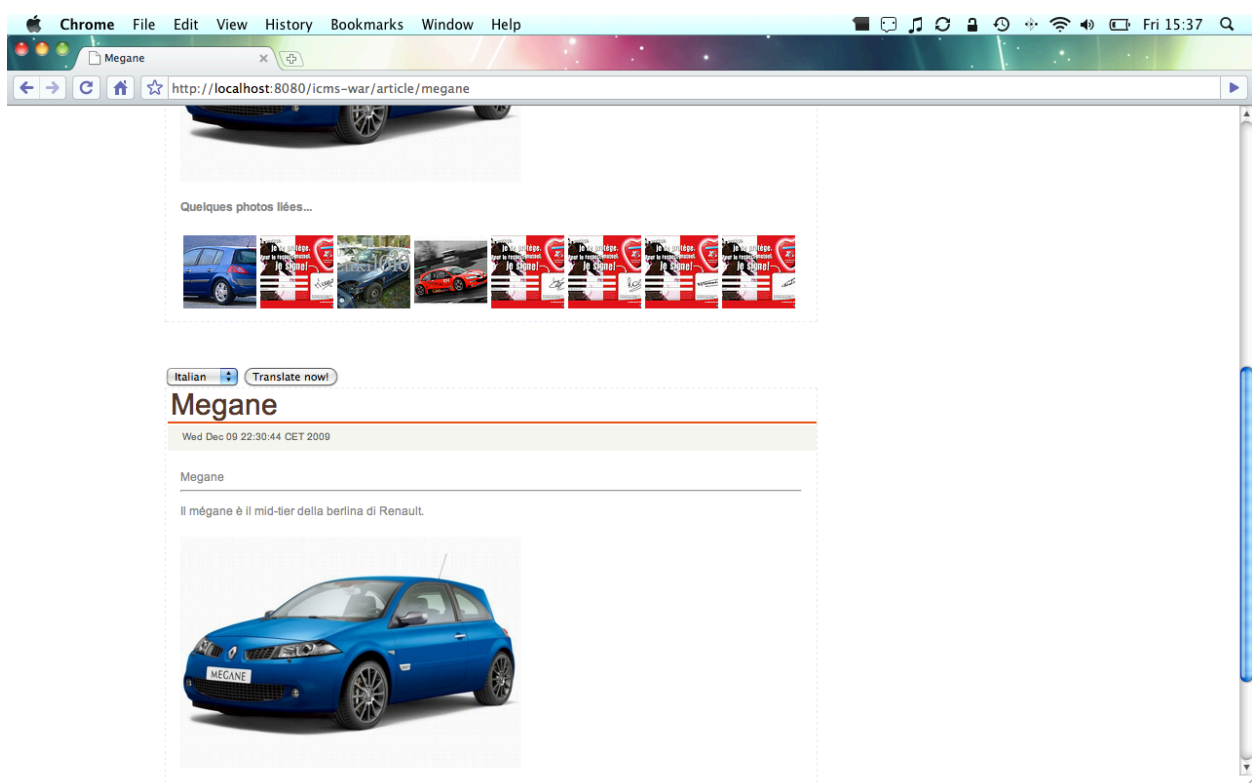
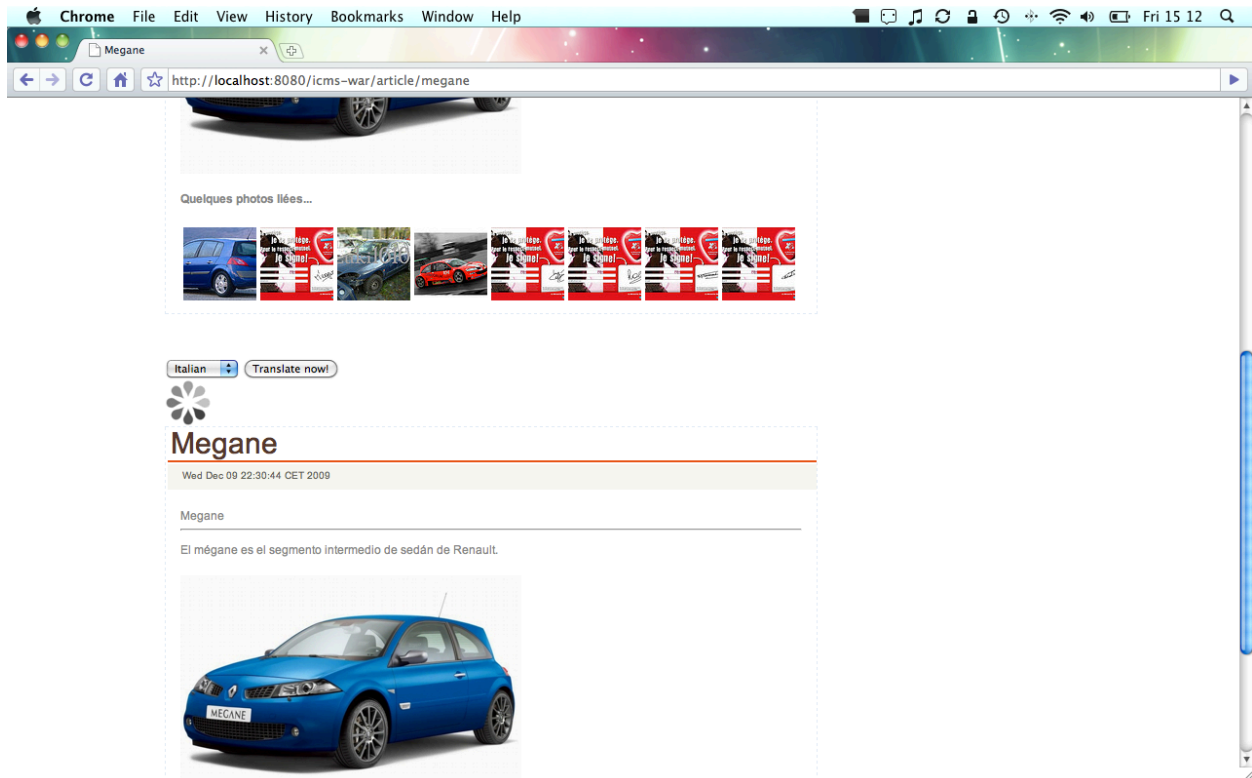
3. et apparition de l'article traduit via AJAX



4. nouvelle traduction possible...



5. et remplacement de la première traduction par la deuxième



Pour chaque webservice, nous avons créé une classe Java, que nous appelons helper (FlickrHelper et TranslateHelper), afin d'avoir un code propre, réutilisable et facile à modifier / refactorer.

IV. Intégration de Dojo

Tout d'abord, tous les appels à Dojo peuvent être retrouvés dans le fichier `/icms-war/web/js/behaviours.js`.

Nous avons utilisé Dojo afin de répondre à 2 besoins ici :

1. Afficher les images de résolution supérieures lors du passage sur les vignettes Flickr.

À noter que nous n'avons utilisé que du JavaScript non-obstrusif (le JavaScript n'est pas dans le code HTML, nous bindons des événements à des méthodes grâce à "dojo.connect" notamment).

Nous avons utilisé la méthode `dijit.Tooltip` pour afficher les images de plus grande résolution au passage de la souris sur les vignettes Flickr.

2. Charger la traduction d'un article via AJAX

Nous avons utilisé la méthode `dojo.xhrGet` pour appeler l'url de l'article `/icms-war/article/<permalink_article>` (nous récupérons cette url à partir du champ "action" du formulaire de traduction) à laquelle nous ajoutons les éléments du formulaire (grâce à la méthode `dojo.formToQuery`). C'est l'élément `translate_to` de ce formulaire qui sera utilisé lors de l'appel au webservice de traduction.