# Articolo Boot appunti di progetto

Ha come scopo la visualizzazione di item, nelle intenzioni dovrebbe adattarsi a visualizzare sia oggetti di tipo testo nel caso lo si voglia utilizzare come blog, sia oggetti di tipo slideshow quindi immagini che scorrono, sia oggetti che descrivono oggetti da acquistare nel caso si voglia utilizzare come sito di ecommerce.

## Architettura

Il progetto è fatto con Eclipse STS ed usa spring boot e quindi tutto il framework spring per l’organizzazione. Il front-end è fatto con angular js che si occupa di tutto il sistema delle view. Di fatto Spring tiene la gestione del backend e quindi accesso ai dati e security, la navigazione è demandata tutta ad angular. La validazione, nelle intenzioni, è fatta sia lato client che lato server.

### TODO

* Inserire procudera di creazione di un nuovo item
* Inserire creazione della pagina della lista per categorie

## Security

La security è fatta con spring security in particolare il filtro crsf è fatto utilizzando oltre ai filtri standard un filtro custom che si chiama **CsrfHeaderFilter** che aggiunge un cookie di sessione che tiene il token XRSF. Il cookie viene recuperato dai vari controller angular e rispedito ogni volta che si richiede un webservice rest.

## Creazione di un nuovo tipo di item (ItemShop)

Occorre aggiungere la voce di new item nel menu **Actions** per farlo occorre andare in **IndexCtrl.js** e aggiungere il nuovo label che va inserito anche nel file **messaggi.properties**. In **appConfig.js** va definito il nuovo state per **urlRouteProvider**. Un nuovo state comprende: il nome dello state, l’url, il template url (la pagina html), il controller che gestisce quella pagina. Il file js del controller va importato in **index.html**.

Per creare un nuovo Item in generale occorrerà creare:

* pagina di view
* pagina di create:
  + nel folder html il file con la pagina **newItemShop.html**;
  + nel folder js il nuovo controller **NewItemShopCtrl.js**. Attenzione perché il nome del controller deve combaciare con quello definito in **appConfig.js** dove sono definiti gli state di url provider.

La definizione della pagina prevede la creazione delle label nel file **messages.properties** (la versione standard andrebbe creato uno per lingua per la internazionalizzazione). Le varie chiavi del messaggio sono la prima cosa che il controller deve elaborare. Infine queste chiavi con i messaggi vanno inserite nella pagina html

* pagina di edit

Ognuno di queste pagine sarà affiancanto dal relativo controller.