ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA
SEDE DI CESENA
SECONDA FACOLTÀ DI INGEGNERIA CON SEDE A CESENA
CORSO DI LAUREA IN INGEGNERIA INFORMATICA

XWiki Mobile: architetture collaborative per device mobili

Elaborato in:
SISTEMI DISTRIBUITI

Relatore
Prof Andrea Omicini

Correlatore
Nazzareno Pompei

Presentata da:
Marco Renio

Sessione: Prima
Anno Accademico: 2011-2012

Parole Chiave

- XWiki
- Architetture collaborative
- Mobile Web
- APICe

Sommario

Indice delle figure e dei listati

Capitolo 1

Capitolo 2

Capitolo 3

Capitolo 4

Capitolo 1: Architettura XWiki

1.1 Introduzione

Internet ha subito, negli anni in cui è stato creato e sviluppato, numerosi avanzamenti tecnologici importanti che lo hanno reso quello che è oggi, basti pensare alla creazione dei primi newsgroup, il protocollo TCP/IP, il primo browser, solo per citarne alcuni e sicuramente una tra le grandi conquiste e creazioni dei nostri giorni lo possiamo identificare con la piattaforma Wiki;

Cos'è di preciso il Wiki?

Lo possiamo definire come una pagina a cui i suoi utilizzatori possono apportare modifiche ai contenuti, arricchendoli e mettendo a frutto la loro esperienza su un dato argomento, uno strumento collaborativo che gli utenti possono usare per condividere la conoscenza, aggiungere nuove voci o modificare le voci attuali, in quanto i contenuti sono aperti.

I Wiki sono considerati come una strada per lo sviluppo della conoscenza, a tutti è capitato di aver utilizzato la piattaforma Wiki in quanto è stata resa celebre dall'enciclopedia "Wikipedia" con il suo lancio nel 2001.

Il Wiki permette la scrittura dei documenti tramite linguaggio di markup tramite browser web, il linguaggio utilizzato per la scrittura viene chiamato "wikitext" ed è molto differente dall'html in quanto non è possibile utilizzare le funzionalità di quest'ultimo ne

quello che riguarda lo javascript o i CSS, penserà il motore wiki a tradurre quelle informazioni in una pagina html che il browser a sua volta interpreterà. Questa è sicuramente una delle debolezze del motore wiki ed è qui che XWiki entra in gioco.

XWiki è una piattaforma web open source evoluta sviluppata tramite l'ausilio del linguaggio java, in un motore wiki similare a quello su cui è basata wikipedia, il cui principio fondamentale è quello di permettere la realizzazione di applicazioni web orientate alla condivisione di informazioni, la collaborazione tra gli utenti, ovvero ognuno può contribuire alla creazione e redazione dei contenuti.

La definizione di semplice motore wiki gli sta un po stretta, in quanto si contraddistingue per molte caratteristiche interessanti, ma prima di arrivare a descriverle forse è meglio capire anzitutto gli scopi e le applicazioni di XWiki. Il concetto alla base è la condivisione della conoscenza proprio dell'architettura Wiki, ed è strutturato in modo che possa piacere anche alle aziende o ai team di sviluppo che creano i loro wiki space privati per la collaborazione, oppure la pubblicazione di documentazione di applicazioni, sia open source che non, in tutti questi campi una architettura di questo tipo sembra essere una soluzione ideale, sia per la semplicità di installazione e configurazione, sia per la quantità di pagine di esempi e una interfaccia di amministrazione veramente molto ampia, estende le funzionalità di un wiki classico consentendo di inserire dati strutturati all'interno delle pagine, indicizzabili e ricercabili.

Sulla realizzazione delle pagine XWiki ha il suo forte, perchè permette la realizzazione di pagine dinamiche, ovvero è possibile

scrivere script che generano codice html utilizzando i linguaggi velocity e groovy, che sono linguaggi di scripting, per estendere le funzioni base, oltre ad avere delle API versatili, in pratica in una pagina possiamo veramente realizzare di tutto, pagine che richiamano contenuti da altre o ne generano di nuove oppure ne modificano gli attributi, che spediscono email, in pratica delle vere e proprie applicazioni destinate ad usi di qualunque genere che permettono di avvicinarsi alle necessità più disparate, tutto questo lavorando ad un livello alto, senza bisogno di modificare l'engine o lavorare con DB, dovremo solo scrivere codice tramite browser. In più XWiki mantiene anche tutte le revisioni fatte per ogni pagina in ordine cronologico.
Uno dei punti forti di XWiki è proprio l'essere sviluppato in java, in quanto è compatibile con tutti i web servlet (o Java servlet) con i quali vengono estesi le applicazioni hostate dai web server.
XWiki si dimostra uno strumento eccezionale, ora andremo a vederne l'architettura in dettaglio.

1.2 Architettura

L'ecosistema XWiki è composto da vari prodotti, noi tratteremo quelli facenti parte dell'XWiki enterprise, che possiede feature avanzate quali blog, rights managment, pdf export, skinning, supporto per advanced forms e scripting avanzato.

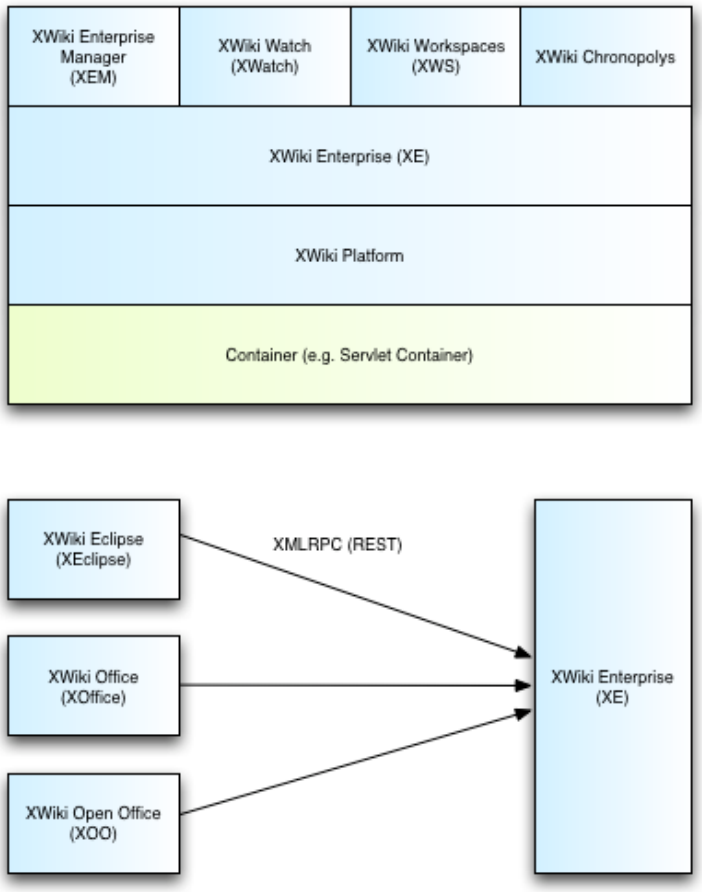


Figura 1.1: L'ecosistema XWiki

La piattaforma XWiki offre API e UI comuni, nel service API sono compresi il Core, i plugins e i moduli, come UI sono compresi skins, templates e applications.

L'architettura di XWiki si basa su prodotti prettamente open sources, tra cui:

- DBCP per la gestione del pooling delle connessioni al database.
- JRCS per il versioning degli attachment.
- Hibernate per la gestione del database (Connessione al DB, memorizzazione)
- Struts per la gestione degli URL scheme
- Velocity per la gestione delle viste
- Radeox come motore per il rendering
- OSCache per la gestione della cache e dei documenti
- OSUser e Security Filter per la gestione dell'autenticazione e per le autorizzazioni

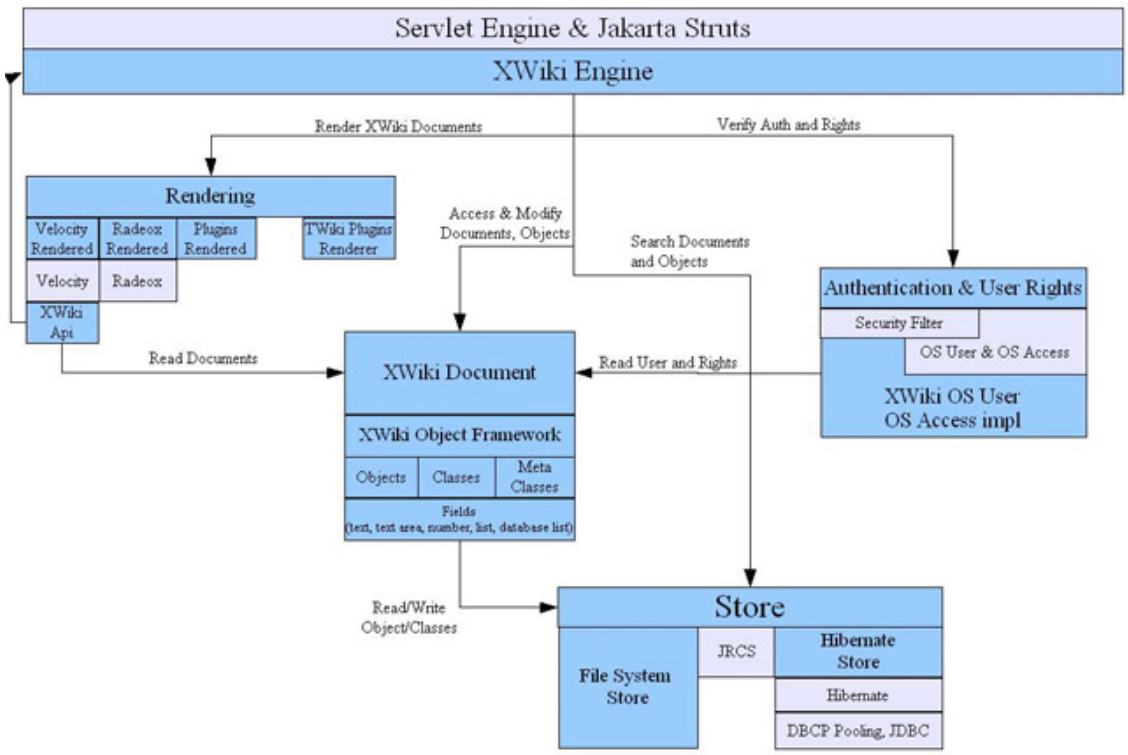


Figura 1.2: L'architettura XWiki

Andando ad analizzare singolarmente i componenti, possiamo spendere alcune parole sul core che è un singolo .jar che contiene diversi moduli che adempiono a molte features, tra le quali il syntax rendering, la gestione delle cache, localizations (ovvero la traduzione in vari linguaggi), exports (nei formati PDF, RTF, WAR), security (i servizi di autenticazione dell'utenza e gestione delle autenticazioni), user management;

i plugins sono anch'essi dei .jar che svolgono funzioni che vanno dal mail sender, al tagging, gestione delle skin, feed rss, upload di file, query, zip explorer e così via, sono tutti facilmente scaricabili tramite l'extensions wiki, sebbene il componente plugin sta per venire abbandonato in favore dei component.

I moduli offrono invece dei servizi equivalenti ai plugins, la loro differenza è che sono basati sul component orientend

development, sebbene XWiki utilizzi un metodo indipendente per gestire la modularità e non si appoggi ad OSGi, CDI o Guice, difatti utilizza un proprio manager chiamato lighweight component manager, scelta operata dal fatto che al momento non esiste ancora uno standard unico per java per questo tipo di sistema component orientend.

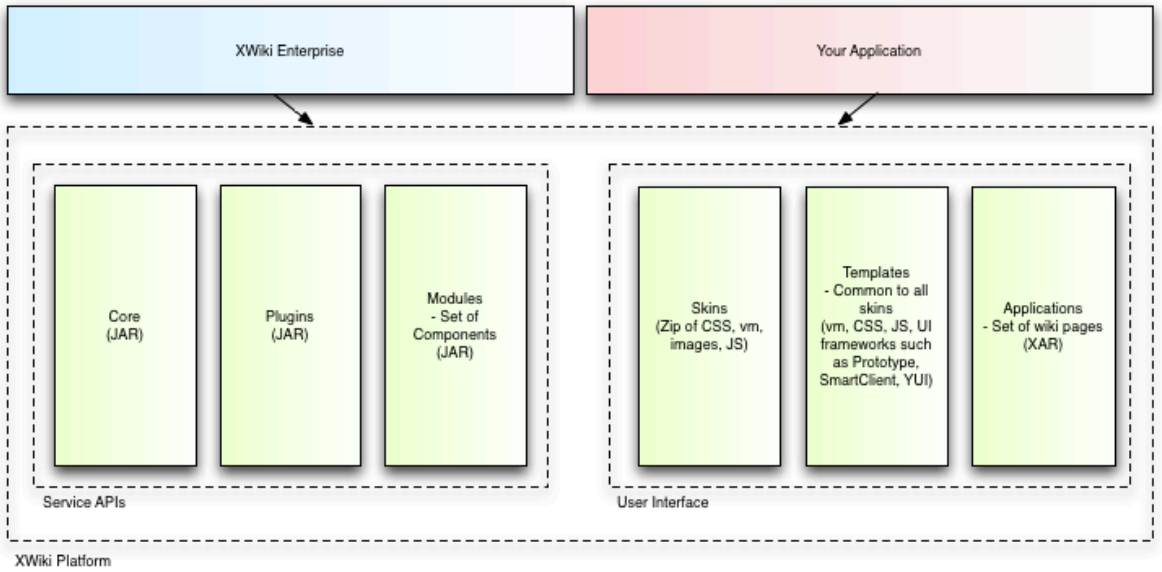


Figura 1.3: Architettura della piattaforma XWiki

Per quello che riguarda le application, come detto in precedenza, nelle pagine di XWiki è possibile inserire codice, combinando assieme varie pagine possiamo ricavare una vera e propria applicazione che è parte integrante del set di pagine da noi creato, che eventualmente possiamo esportare per poter importare in altri XWiki (riutilizzabilità delle applicazioni nella stessa piattaforma), le applicazioni in effetti sono un vero e proprio strumento che si ha a disposizione per rendere delle semplici pagine come componenti avanzati con enormi potenzialità, come esempi potremmo prendere bullettin board, calendari, form, pools.

Sarebbe giusto spendere delle parole finali nei confronti dell'estensione di XWiki, riassumendo come è possibile lavorare sulla piattaforma in questo senso nei vari modi offerti:

- Scrivendo script nelle pagine dell'XWiki
- Scrivendo applicazioni (Set di pagine)
- Scrivendo plugins
- Scrivendo moduli (Set di componenti)
- Progettando skin o facendo un lavoro di modding sulle esistenti
- Estendendo l'esistente services API (Facendo riferimento alla documentazione fornita)

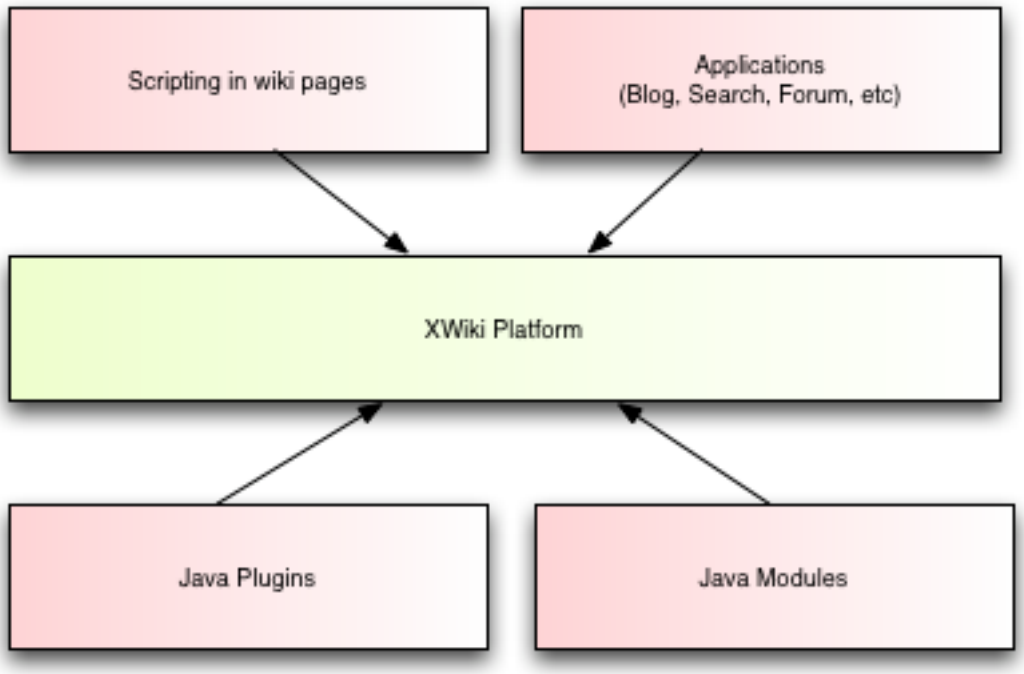


Figura 1.4: Estensione XWiki platform

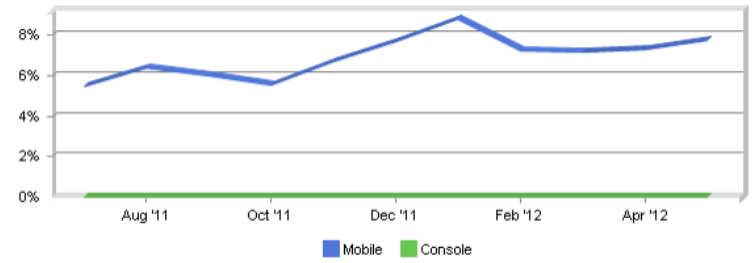
Capitolo 2: Architetture mobile Android e iOS lo stato dell'arte

2.1 Il mondo Mobile

Il mondo mobile è una realtà più che presente nel panorama del web odierno, l'affollarsi del mondo tecnologico di dispositivi portatili come smartphones e tablet ha fatto si che il mondo del web si debba adattare alle nuove architetture di questi dispositivi, in particolare alla loro capacità di calcolo ridotta (ovviamente paragonandoli ai sistemi desktop), la banda "limitata" delle connessioni mobili (per quello che riguarda la copertura e la velocità delle reti 3G la situazione italiana è migliorata nettamente negli ultimi tempi sebbene le nuove generazioni LTE o 4G, che fornirebbero grandi quantitativi di dati, stanno stentando a decollare), l'interfaccia di utilizzo touchscreen (che per molti siti richiede un approccio diverso per la navigazione, considerando le sue enormi potenzialità per la GUI ma ovviamente anche i limiti rispetto al mouse in quanto precisione) e il loro schermo di dimensioni ridotte (con dimensioni che variano dai 3,5"-4,2" degli smartphone ai 7"-10,5" dei tablet).

Browsing by Device Category Trend

July, 2011 to May, 2012



Month	Desktop	Mobile + Tablet	Console
July, 2011	94.13%	5.50%	0.02%
August, 2011	93.20%	6.37%	0.02%
September, 2011	93.65%	5.98%	0.02%
October, 2011	94.16%	5.52%	0.02%
November, 2011	92.92%	6.73%	0.02%
December, 2011	91.99%	7.67%	0.02%
January, 2012	90.84%	8.77%	0.02%
February, 2012	92.49%	7.24%	0.01%
March, 2012	92.51%	7.20%	0.01%
April, 2012	92.44%	7.30%	0.01%
May, 2012	91.96%	7.76%	0.00%

Figura 2.1: Browsing da tipologie di dispositivi diversi e trend globale

Se scrivere o far visualizzare un sito web ad un PC può essere relativamente facile in quanto questi dispositivi possiedono generalmente hardware che permettono di sfruttare facilmente anche le tecnologie più pesanti (ad esempio Flash o Javascript complessi) e generalmente hanno alle loro spalle connessioni da almeno 6-7 megabit stabili (sempre considerando un contesto di connessioni mobili con segnale da rete eterea) che supportano grandi volumi di download/upload, lo stesso non si può dire degli smartphones o tablet(almeno per il momento) in quanto in pochi dispositivi abbiamo processori doppio core, ram superiori ai 512mb e connessioni 3G o superiori degne di questo nome a livello di marketshare.

2.2 Sviluppo di XWiki per il mondo mobile

Quindi visualizzare e partecipare alla piattaforma XWiki dal mondo mobile può avere a mio avviso due possibili strade:

1. Ottimizzazione con dei servizi specifici per dispositivi portatili.
2. Una applicazione lato client.

Considerando il primo punto ci può venire in aiuto lo skin extensions
ovvero la possibilità di cambiare e customizzare il layout del wiki, sia solo di alcune pagine che dell'intero wiki senza cambiare il template della skin o i fogli di stile. Creare una extension mobile per il layout delle pagine potrebbe essere la soluzione ideale, non richiederebbe registrazione, ma solo la connessione al wiki e un dispositivo mobile con browser (Safari, android, opera, etc). E' possibile usare una skin extension on demand con questo codice velocity

```
{{velocity}}
$XWiki.ssx.use("XWiki.MyFirstStylesheetExtension")
{{/velocity}}
```

Dove al posto di "XWiki.Qualcosa" si mette il nome della extension creata. Potrebbe essere utile usare degli script per ottimizzare il contenuto per la pagina, ad esempio che ridimensiona il testo, diminuisce la larghezza della pagina a seconda di come vogliamo rappresentarla su un dispositivo mobile, in effetti questo plug-in permette ai componenti di interfaccia di prendere i CSS e i Javascript che necessitano a seconda di cosa gli viene richiesto e permette di non dover inserire in stili e pagine di script che magari

verrebbero utilizzate esclusivamente in qualche pagina. Una skin è composta da:

- Template in Velocity
- CSS file
- Javascript file
- Immagini

Questi componenti si trovano nel server dove è stata inizializzata l'istanza di XWiki. Ogni Skin può essere utilizzata a vari livelli, ovvero per tutta l'istanza del nostro Wiki, oppure possiamo calibrarla per far si che copra solo uno space, un utente particolare o per qualunque altra nostra necessità (ad esempio gruppi di utenti, o nel nostro caso, browser o dispositivi specifici).
Attualmente è disponibile tramite il sito di XWiki una mobile skin, denominata "Simple Mobile Skin", al momento è un prototipo e permette il login/logout, la visualizzazione veloce dei link, l'edit del wiki e un editor wysiwyg.

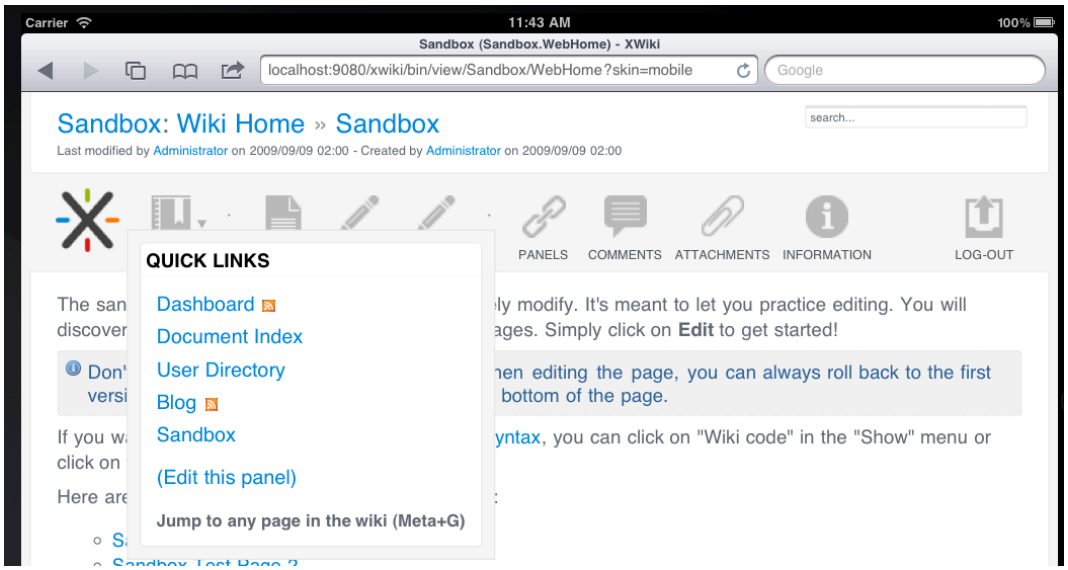


Figura 2.2: Simple mobile skin dropdown menù

L’utilizzo dei dropdown menù permette una interfaccia più snella per la fruizione dei testi e della GUI propria della piattaforma XWiki.

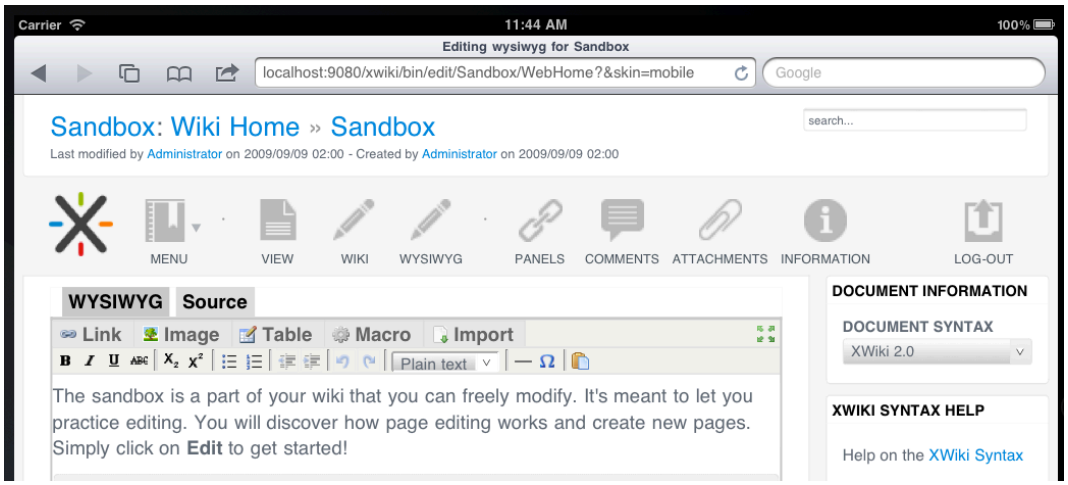


Figura 2.3: Simple mobile skin editor wysiwyg

Per quello che riguarda lo sviluppo di una applicazione dedicata, conviene anzitutto chiedersi:
Per quali dispositivi può valerne la pena?

Secondo l’attuale market share abbiamo che la maggior parte dei dispositivi mobili sono rappresentati da iOS, in seconda istanza abbiamo Android, seguono Java ME, Blackberry RIM, Symbian e Windows Phone.

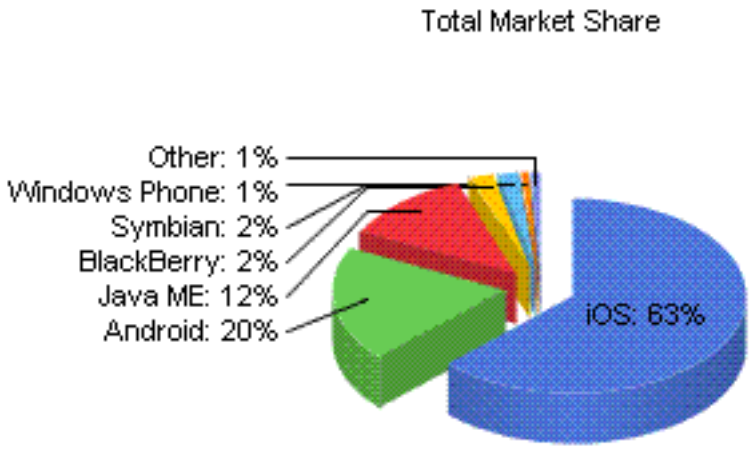


Figura 2.4: Market share dispositivi mobili

Lo sviluppo di una app per iOS e per terminali Android potrebbe avere un riscontro positivo sul mercato.
Un Applicazione permette oltre che a dimensionare correttamente i contenuti “on screen”, ad avere più servizi rispetto ad una ordinaria web based interface, senza considerare l’adattabilità della GUI alle dimensioni dello schermo e agli elementi dei menù user friendly per quello che riguarda il click con il dito.

Attualmente non è disponibile nessuna versione per iOS, mentre esiste una versione sperimentale per Android, realizzata da Chamika Weerasinghe e il cui codice è reperibile tramite il sito delle extensions di XWiki, per il suo sviluppo sono state utilizzate:

- REST model for simple-xml
- REST API for Android
- User interface components for Android
- Sample Application which explains above libraries

Attualmente le XWiki RESTful APi sono basate sul modello JAXB, ma per via delle limitatezze dei dispositivi mobile per lo storaging è stato sfruttato un modello chiamato “simple-xml” che non è altro che un parser leggero xml. L’interfaccia non è sicuramente delle migliori e si possono notare degli impuntamenti nell’utilizzo in alcune pagine e spaces, è una versione embrionale per l’utilizzo di XWiki e la renderizzazione delle pagine non è fedele all’aspetto originario e lo stile di impaginazione, in quanto notiamo dalle immagini dell’app che il modulo “page viewer” contiene le informazioni della pagina (e spesso quando ci troviamo a navigare dentro crasha) e in fondo il contenuto della stessa (senza alcun tipo di formattazione) e soprattutto il problema è che è solo un visualizzatore non possiamo apportare le nostre modifiche alle pagine o amministrare l’XWiki in nessun modo. Come ultima note aggiungo che richiede che l’utente sia già registrato (non offrendo nessun modo per farlo).

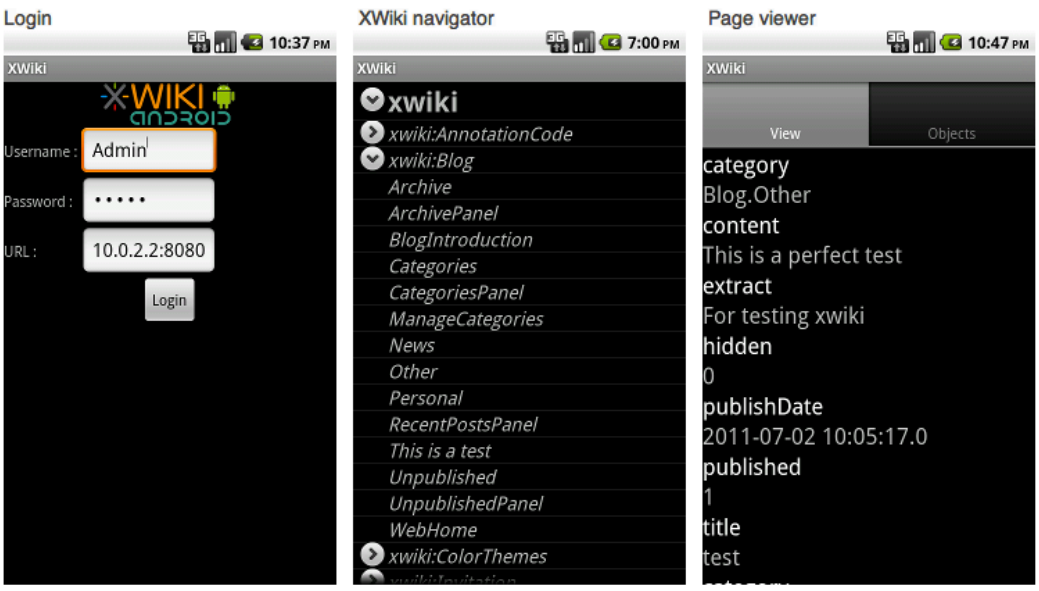


Figura 2.5: Interfaccia app Android

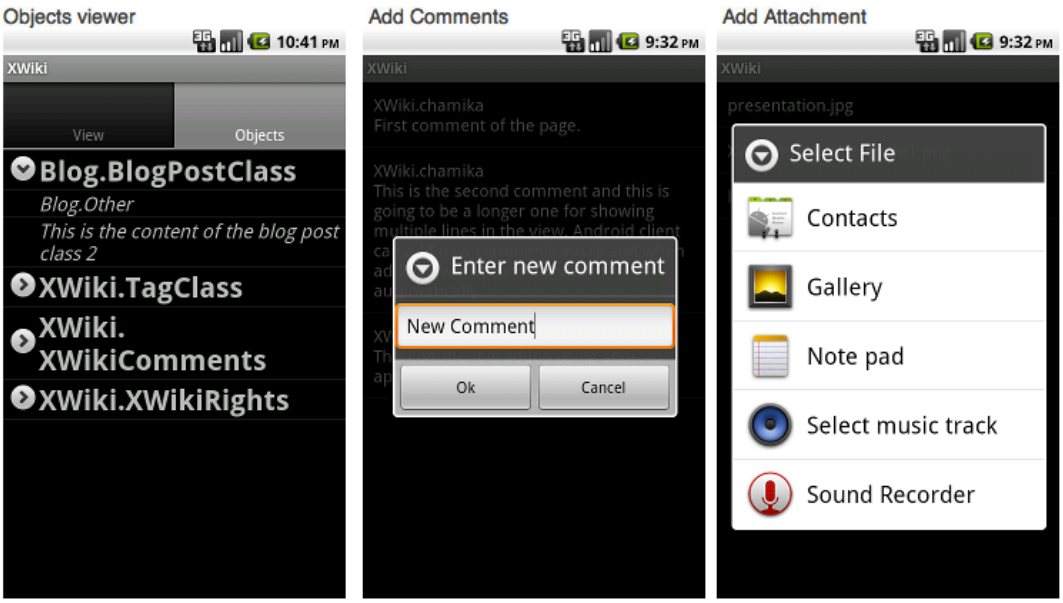


Figura 2.6: Interfaccia app Android

Capitolo 3: Customizzazione di XWiki

3.1 Lo Skin extension plugin

Le potenzialità di XWiki si esprimono ogni qualvolta si sviluppa qualcosa con essa, in particolare andremo a vedere l'utilizzo dello strumento XWiki Skin Extension e come potrebbe essere utilizzato per la creazione di una Skin incentrata per il mondo mobile web.

Lo skin extension è un meccanismo ideato per la personalizzazione del layout della piattaforma, possiamo applicare la skin creata solo ad alcune parti come pagine o spaces oppure estendere a tutta la piattaforma senza la necessità di cambiare gli skin template o i fogli di stile. Lo Skin Extension Plugin (disponibile nella versione Enterprise di XWiki) permette di utilizzare file diversi (Css o javascripts) per la visualizzazione a browser che non sono parte integrante della skin attuale del wiki, queste estensioni le possiamo chiamare wiki objects.

I wiki objects non sono altro che istanze uniche di una classe con valori unici definiti per ogni proprietà che compone la classe stessa, ogni oggetto è allegato ad una pagina specifica ed ogni pagina può avere può oggetti allegati ad essa, l'utilizzo degli oggetti permette l'inserimento strutturato di informazioni nel wiki, mentre in un wiki tradizionale possiamo inserire solo informazioni non strutturate all'interno di esso (come semplice testo).

3.2 Creazione di una skin extension

Le skin extension essendo veri e propri oggetti xwiki possono essere create attraverso browser, e le possiamo creare di due tipi:

- JavaScript extensions
- StyleSheet extensions

Quindi iniziamo creando il vero e proprio oggetto (che è il primo step per la creazione di entrambi i tipi di extension).
Per la creazione dell'oggetto ci dobbiamo posizionare nella pagina a cui vogliamo allegarlo ed editarla con l'object editor, la pagina può essere una qualunque pagina del nostro xwiki. Dal menù Edit selezionando Objects potremo aggiungere un nuovo oggetto alla nostra pagina, in questo caso sceglieremo un oggetto JavaScriptExtension

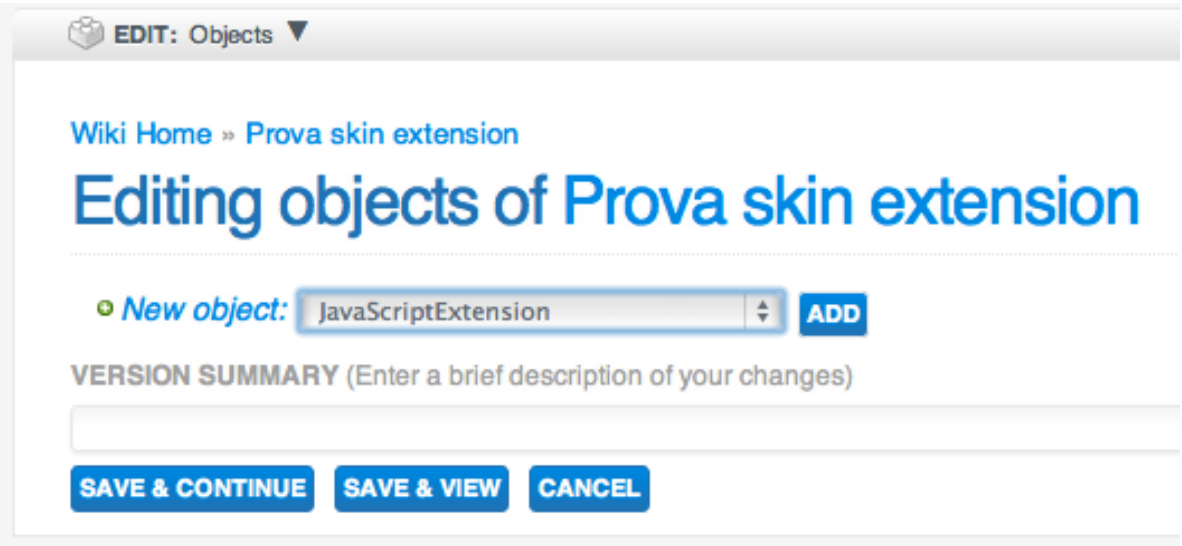


Figura 3.1 Aggiunta di un oggetto ad una pagina XWiki

Ora che l'oggetto è disponibile nella pagina possiamo iniziare a scrivere il vero e proprio codice dell'estensione ed iniziando a riempire i campi dell'oggetto

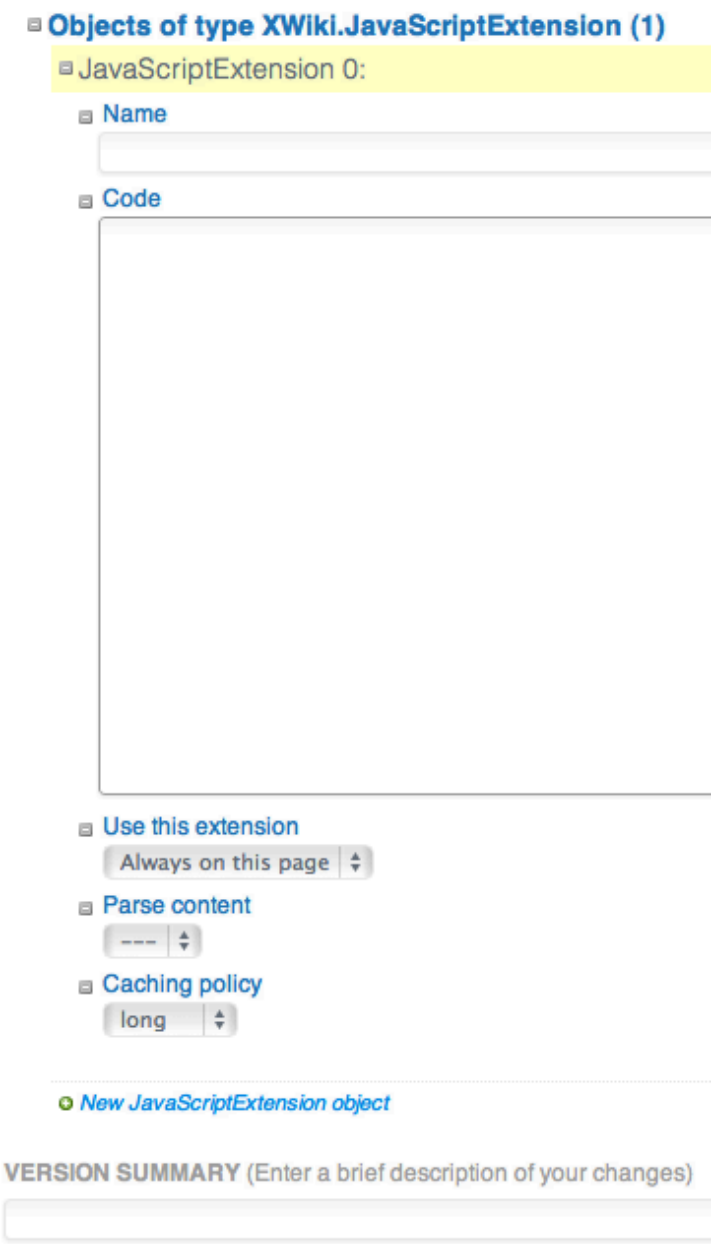


Figura 3.2 Campi dell'oggetto

Il primo riguarda il nome della nostra extension, mentre il secondo riguarda il vero e proprio codice della stessa, qui potremo scrivere il codice javascript che verrà eseguito dall'oggetto, potremmo inserire codici semplici come un "alert" oppure qualcosa di più complesso, tutto quello che ci è permesso fare con il linguaggio javascript. L'uso dell'extension può essere di tre tipi

- OnDemand
- Always on this page
- Always on this wiki

A seconda dell'utilizzo che vogliamo del nostro oggetto dobbiamo scegliere quando utilizzarlo, ad esempio se vogliamo che ogni volta che qualcuno visita la pagina riceva un warning selezioneremo "Always on this page", mentre per l'opzione "OnDemand" dovremo noi nello specifico richiedere che l'oggetto venga eseguito, utilizzando una call con del codice in velocity

```
{{velocity}}
$xwiki.jsx.use("XWiki.MyFirstJavascriptExtension")
{{/velocity}}
```

Questo codice va scritto nella pagina in modalità Wiki. In ultimo possiamo decidere la politica di caching con il quale saranno generati gli header HTTP che saranno utilizzati per i file javascript generati.

Questo è quello che riguarda gli oggetti javascript, nel nostro caso volendo modificare l'apparenza del layout delle pagine, dovremo utilizzare le StyleSheet Extension.

Utilizzando la stessa metodologia per la creazione di un oggetto possiamo creare una StyleSheet extension semplicemente selezionando un oggetto di classe XWiki.StylesheetExtension, utilizzando come codice, ad esempio,

```
#xwikicssprova {
   background-color: blue;

}
```

Questo andrà a modificare il background color della pagina, in accordo con il linguaggio CSS utilizzato.

3.3 Creazione di una Skin extension mobile, primi passi

La creazione di oggetti StyleSheetExtension (SSX) e JavaScriptExtension (JSX), quindi, potrebbe essere il nostro punto focale sulla quale sviluppare la nostra skin extension, per quello che riguarda il tipo sicuramente dovremo basarci sul ”OnDemand”, ovvero quando il nostro oggetto principale, che potrebbe benissimo essere uno script scritto in velocity, riconoscerà il dispositivo come un mobile dovrà richiamare l’oggetto o gli oggetti in questione che avremo creato per l’occasione, quindi questo ci permette di poter creare oggetti diversi per vari dispositivi con schermo di dimensioni più o meno grandi e di potenza che può variare dal single core al multicore, ad esempio potremo creare un oggetto JSX che ridimensiona la pagina in un modo a seconda che il dispositivo venga riconosciuto come tablet o come smartphone, o semplicemente potrebbe attivare la vista mobile web con uno stile completamente diverso per menù e

visualizzazione delle pagine, impostato tramite CSS, panels e tutti gli elementi della nostra UI.

Il nostro case study, ovvero il sito di APICe ha una skin personalizzata rispetto alla base fornita con il portale XWiki, potrebbe essere una soluzione ottimale creare una custom skin mobile che si attiva quando un dispositivo mobile viene riconosciuto dai nostri script, sia per quello che riguarda la parte amministrativa sia per quello che riguarda la consultazione o l'aggiornamento collaborativo delle pagine, quindi nel prossimo capitolo tratteremo lo sviluppo di una skin mobile partendo da una base che è la simple mobile skin di cui abbiamo parlato nel capitolo 2.

Capitolo 4: Skin per APICe

4.1 Progettazione skin per APICe

In questo capitolo verranno affrontate le tematiche di progettazione di una skin e verrà progettata la modifica da effettuare alla Simple Mobile Skin fornita da xwiki.org e sviluppata da Ludovic Dubost ed Ecaterina Moraru.

Le modifiche che andremo ad apportare saranno riguardanti i quicklinks panel, l'header e il footer, la skin già di suo fornisce molti spunti utili per realizzare un efficiente sito mobile e si può adattare ad ogni necessità, basta saper usare velocity. Come primo passo parliamo di come la skin entra in funzione.

Essendo la skin OnDemand la sua richiesta viene effettuata controllando l'user agent che si collega al sito, una volta identificata procede alla richiesta della stessa aggiungendo all'URL della nostra wiki una query request php "?skin=mobile"; questo è il listato di velocity che effettua questo controllo

```
{{velocity}}
#set($mobile = $request.getCookie
("xwikimobile").getValue())
#if("$mobile"!="off")
#set($ua = $request.getHeader("User-agent"))
#if($ua.contains("iPhone")||$ua.contains("iPad")||
$ua.contains("Android")||$ua.contains("IEMobile"))
 #if("$!request.skin"=="")
  $response.sendRedirect($doc.getURL("view", "$!
request.getQueryString()&skin=mobile"))
 #end
```

```
#end
#end
{{/velocity}}
```

Listato 4.1

Questo ci permette di utilizzare la nostra skin base per il sito versione desktop e di far visualizzare correttamente ai nostri visitatori mobile web il sito con un'interfaccia più mobile like, problema di cui abbiamo parlato ampiamente nei capitoli precedenti.

4.2 Analisi di APICe e quick links panel

Analizzando il sito APICe (http://apice.unibo.it) notiamo una side bar nella quale vengono raggruppati i personal spaces più importanti, i product spaces, gli eventi imminenti e i progetti in corso.

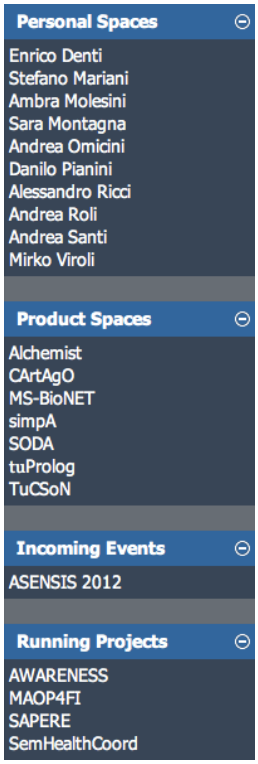


Figura 4.1: Side bar sito APICe

un’altro elemento importante è la side bar sotto il logo di APICe, che raccoglie tutti gli spaces del sito

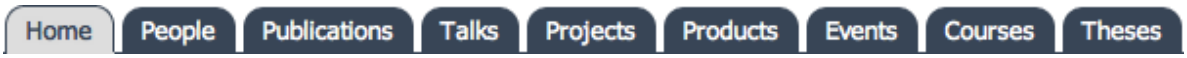


Figura 4.2: Tab bar sito APICe

difatti APICe è composto da molti spaces che comprendono i collaboratori, nei quali sono compresi professori, studenti e tecnici, le pubblicazioni, i seminari, i progetti in corso ed ultimati, eventi, corsi e tesi, essendo una grande quantità di link a spaces, verosimilmente dovremmo utilizzare un drop down menù per inserirli tutti in modo meno invasivo possibile, quindi inseriremo tutto utilizzando uno script in velocity come link a tutti i nostri Spaces.

```
{{velocity}}
#panelheader($msg.get('xe.panels.spaces'))
#set ($spaces = $xwiki.spaces)
#foreach ($space in $spaces)
  #if ($hasAdmin || ($xwiki.hasAccessLevel('view',
"${space}.WebHome") && !$blacklistedSpaces.contains
($space)))
    #if ($foreach.index > 0)
      (% class="pitemseparator" %) ~| (%%)##
    #end
    (% class="panelitem#if ($space == $doc.space)
currentspace#end" %)[[$space>>${space}.WebHome]](%%)
##
  #end
#end
#panelfooter()
{{/velocity}}
```

Listato 4.2

Questo script visualizzerà un drop down menù (visibile nella prossima pagina) con tutti i link ai nostri spaces a portata di “dito” per una navigazione veloce in tutto il sito da qualunque parte il visitatore si trovi.

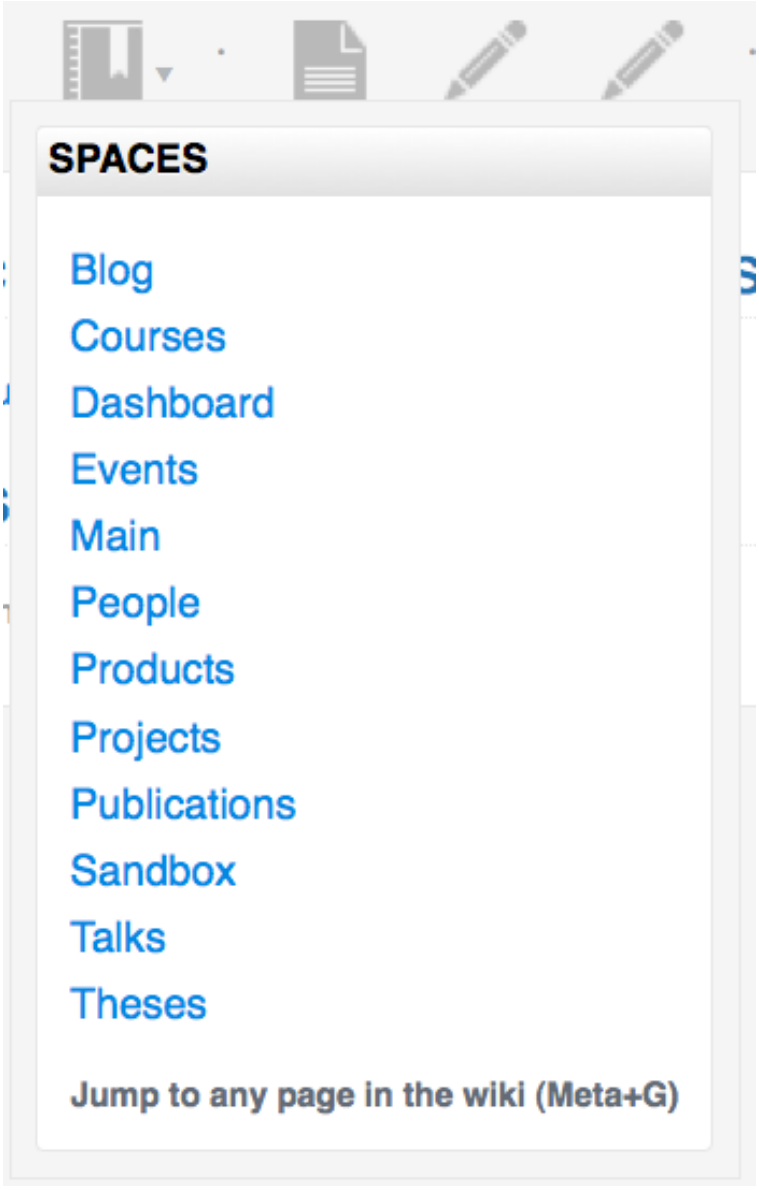


Figura 4.3: Drop down menù

Apporteremo anche delle modifiche all’header di modo che sia presente il logo APICe in cima e al footer facendo si che

combaci con quello presente nel sito. Essendo entrambi dei listati in velocity dovremo modificare i file header.vm e footer.vm per far si che la nostra skin venga modificata.

4.3 Header

Lo header conterrà il codice per la menù bar, la search bar, il puntatore alla pagina dove il visitatore di trova e le last modifications. Il modo migliore di modificarla potrebbe essere quello di inserire il logo di APICe in testa oltre che ad eliminare il link panels presente nella skin che non è altro che una ripetizione dei quick links, nel nostro caso, ma senza drop down menù (cosa che reputo molto scomoda). Le last modifications verrano eliminate dall'header perchè non sono informazioni da visualizzare in prima battuta, quindi di seguito il listato dello header

```
{{velocity}}
$context.setLinksQueryString("skin=${xwiki.skin}")
<div id="quicklinks" style="display: none;">
###
###    Display Panel
###
#macro(iosdisplaypanel $name)
  #set ($pobj = "")
  #set ($paneldoc = "")
  #set ($paneldoc = $xwiki.getDocument($name))
  #if ($paneldoc != "")
    #set ($pobj = $paneldoc.getObject
("Panels.PanelClass"))
    #if (!$pobj)
      ## discarded
    #else
      $!doc.display("content", "view", $pobj)
    #end
```

```
  #end
#end
#iosdisplaypanel("Panels.QuickLinks")
</div>
<script type="text/javascript">
function showQuickLinks() {
  $("quicklinks").toggle();
}
</script>
#set($parents = $util.arrayList)
#set($discard = $parents.add($doc.fullName))
#macro(breadcrumb $doc $string $level)
  #set($parent = $doc.parent)
  #if(($parent != "") && ($level < 6) && (!
$parents.contains($parent)))
    #set($discard = $parents.add($parent))
    #set($pdoc = $xwiki.getDocument
($parent).getTranslatedDocument())
    #set($pdocurl = $pdoc.getURL("view"))
    #set($nstring = "<a href='$pdocurl'>
$xwiki.getXMLEncoded(${pdoc.getRenderedTitle('plain/
1.0')})</a> <span class='separator'>&raquo;</span>
$string")
    #set($level = $level + 1)
    #breadcrumb($pdoc $nstring $level)
  #else
    $string
  #end
#end
#if($context.getMode()==0) ## Visible only in a page
<div id="headerspace" class="layoutsection">
 <img src="http://thelaw.altervista.org/immagini/
newlogo.jpeg">

    <div class="ios" id="search">
      <form action="$xwiki.getURL
('Main.WebSearch')">
        <input type="hidden" name="skin" value="$
{xwiki.skin}" />
```

```
          <label class="hidden"
for="headerglobalsearchinput">$msg.get
('panels.search.inputLabel')</label><input
class="globalsearchinput withTip"
id="headerglobalsearchinput" type="text" name="text"
value="$msg.get('panels.search.inputText')"
size="10"/>
      </form>
    </div>
    <div class="ios" id="languages">
    #if($xwiki.isMultiLingual())
      <span class="glink" id="headerlanguages">
      #set($defaultLanguage = "$!
{doc.getDefaultLanguage()}")
      #if($defaultLanguage == '')
        #set($defaultLanguage = "$!{doc.getLanguage
()}")
      #end
      #if($defaultLanguage == '')
        #set($defaultLanguage = "default")
      #end
      <a href="$!doc.getURL("view", "language=
$defaultLanguage")" class="language-default#if
($tdoc.realLanguage == $defaultLanguage ||
($defaultLanguage == 'default' && $tdoc.realLanguage
== '')) language-current#end">$defaultLanguage</a>
      #set ($wikiSettingsLanguages = $!
xwiki.getXWikiPreference('languages').trim().split
('\s*[,| ]\s*'))
      #set ($hasLanguagesSet =
$wikiSettingsLanguages.size() > 1 || "$!
wikiSettingsLanguages.get(0)" != '')
      #foreach($lang in $doc.translationList)
        #if(!$hasLanguagesSet ||
$wikiSettingsLanguages.contains($lang))
          <a href="$!doc.getURL("view", "language=
$lang")" class="language-translation#if
($tdoc.realLanguage == $lang) language-current#end">
$!lang</a>
```

```
        #end
      #end
      </span>
    #end
    </div>
  </div>
  <div class="ios" id="menu">
    #set($editaction = $doc.getDefaultEditMode())
    <a class="logo" href="$!xwiki.getURL
('Main.WebHome')" title="Home" rel="home"></a>
    <a class="action other" href="$doc.getURL
("view", "viewer=panels")" onclick="showQuickLinks
(); return false;"><span class="menuarrow">&#9660;</
span>$msg.get('XWiki.XWikiPreferences_menu')</a>
    <span class="separator">&#183;</span>
    <a class="action view" href="$doc.getURL("view",
"")">$msg.get('view')</a>
    <a class="action syntax" href="$doc.getURL
("edit", "editor=wiki&$!languageparams")">$msg.get
('core.menu.edit.wiki')</a>
    <a class="action edit" href="$doc.getURL
($editaction, $!languageparams)">$msg.get
('core.menu.edit.wysiwyg')</a>
    <span class="separator">&#183;</span>
    <a class="action comment" href="$doc.getURL
("view", "viewer=comments")">$msg.get
('docextra.comments')</a>
    <a class="action attach" href="$doc.getURL
("view", "viewer=attachments")">$msg.get
('docextra.attachments')</a>
    <a class="action info" href="$doc.getURL("view",
"viewer=information")">$msg.get
('docextra.information')</a>
      #if (!$xcontext.action.startsWith('login'))
        #if ($isGuest)
          #set ($loginurl = $xwiki.getURL
('XWiki.XWikiLogin', 'login', "xredirect=
$escapetool.url($xwiki.getRequestURL())"))
```

```
          <a class="action login" href="$!loginurl">
$!msg.get('login')</a>
        #else
          #set ($logouturl = $xwiki.getURL
('XWiki.XWikiLogout', 'logout', "xredirect=
$escapetool.url($xwiki.getRequestURL())"))
          <a class="action login" href="$!
logouturl">$!msg.get('logout')</a>
        #end
      #end
    <div class="clearfloats"></div>
  </div>
</div>
#end
{{/velocity}}
```

Listato 4.3

Che verrà visualizzato in questo modo

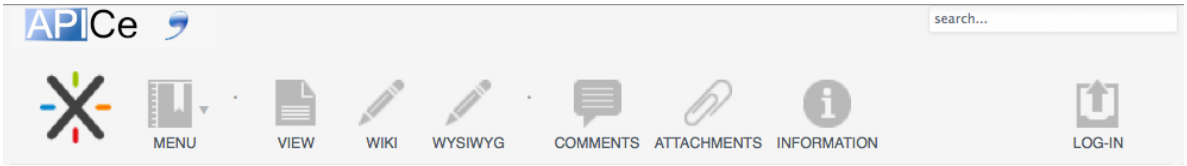


Figura 4.4 Header completo

4.4 Footer

Il footer conterrà le last modifications sulla pagina, il copyright e lo switch alla desktop version, di seguito il listato

```
{{velocity}}
 <div id="document-info-mobile">
     <span class="xdocLastModification"> ## Last
modification
      $msg.get('core.footer.modification',
[$xwiki.getUserName($tdoc.author), $xwiki.formatDate
($tdoc.date)])
```

```
      </span>
      <span class="xdocSeparator">
      -
      </span>
      <span class="xdocCreation">
      $msg.get('core.footer.creation',
[$xwiki.getUserName($doc.creator), $xwiki.formatDate
($doc.creationDate)])
       #if($tdoc.realLanguage != $doc.realLanguage)
         $msg.get('core.footer.translationCreation',
[$tdoc.realLanguage, $xwiki.getUserName
($tdoc.creator), $xwiki.formatDate
($tdoc.creationDate)])<br/>
       #end
      </span>

      <span class="xdocCreation">Copyright © 2011
aliCE Research Group @ DEIS
    </span>
    </div>
    $context.unsetLinksQueryString()
    <div class="minwidth">
    <a href="$doc.getURL("view")"
onclick="createCookie('xwikimobile','off',
0);">Desktop version</a></div>
    </div>
$context.unsetLinksQueryString()
{{/velocity}}
```

Listato 4.4

visualizzato in browser

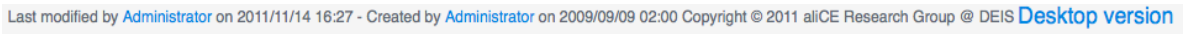


Figura 4.5 Footer

Conclusioni

Il mondo mobile sta avendo una rilevanza sempre maggiore nel mondo delle telecomunicazioni e questo aumento continuo di market share non può lasciare inalterati gli sviluppatori di software e di siti web che devono adattare le loro architetture sempre di più verso questo mercato in rapida ascesa.

XWiki, che si propone come piattaforma sempre più importante per aziende e team di sviluppo sicuramente non è al passo con i tempi, in questo campo, in quanto dispone solo di una skin mobile a livello embrionale e di una app per android sviluppata malamente, il mercato potrebbe richiedere lo sviluppo di una app iOS e una app android funzionali e di una skin per i restanti dispositivi e/o per la vista web nel caso alcune risorse non siano disponibili nelle app, sicuramente una evoluzione in questo senso sarebbe auspicata per l'aumento dell'utilizzo di una piattaforma potete ed estremamente versatile come XWiki.

Bibliografia

Sitografia

- http://www.xwiki.org
- http://jira.xwiki.org
- http://xwiki.475771.n2.nabble.com
- http://www.mokabyte.it
- http://apice.unibo.it