Hola a tothom, som en Salva Lorca i aquesta és una presentació de la part més visible de l’aplicació. Estam parlant de la part Client.

SENCHA TOUCH

Per aquest projecte d’aplicacions per a mòbils usant HTML5 hem fet servir el framework Sencha Touch. Aquest utilitza els estàndards web d'HTML5, CSS3 i Javascript per desenvolupar aplicacions web per a dispositius mòbils amb l'aparença de ser natius dels sistemes iOS, Android i Blackberry. De fet, Sencha Touch va ser el primer framework d'aplicacions web complet i específicament desenvolupat per aprofitar la major potència i flexibilitat d'HTML5, CSS3 i Javascript.

ESTRUCTURA DE L’APLICACIÓ

Us explicarem una mica com està estructurada la nostra aplicació, i ho farem des de l’entorn de desenvolupament que hem escollit, el Sencha Architect. La base de l’aplicació conté dos models, dos magatzem de dades (o stores), 10 vistes (o views), un nom d’aplicació (IdiomesApp) i un controlador. A més l’aplicació executa un component inicial, i en el nostre cas hem escollit la vista “menuPanel”. Quan repassem les vistes de l’aplicació vorem com és aquesta vista.

CONTROLADOR

En primer lloc parlarem del controlador. Aquesta instància maneja les interaccions d’algunes parts de l’aplicació. Nosaltres l’hem usat per tractar el comportament de l’acció “tornar enrere”.

VISTES

En total tenim 10 vistes, i aquesta és la vista inicial “menuPanel”. El menú panel conté un component de pestanyes (o tabs) que tindran un comportament de menú. El “menuPanel” es compon de 3 contenidors que corresponen als tres mòduls de l’aplicació. Aquests són Diccionari, Llistes d’Estudi i Flashcards. Quan canviam d’opció de menú es disparen uns events recollits en la funció “onMenuPanelActiveItemChange”. Un altre component molt important d’aquesta vista principal és la barra superior, que inclou el títol de la pantalla actual i els botons per executar diverses accions, com per exemple tornar a la pantalla anterior, crear una nova paraula, etc.

Les següents 3 vistes són llistes d’elements que estan enllaçades amb els 3 contenidors de la vista principal. La primera és un panell amb una llista de paraules recollides del magatzem de dades. Un listener esperarà la pulsació d’un element de la llista i ens mostrarà la vista de detall de la paraula en concret. La segona i tercera són panells amb una llista de llistes d’estudi. També tenen els seus listeners que donaran comportament a les accions de l’usuari.

La vista de detall d’una paraula és un contenidor senzill amb una plantilla i uns estils per mostrar la paraula d’una manera aclaridora.

Un altre tipus diferent de vistes són les que hereten dels formularis. En concret disposam dels formularis per donar d’alta una paraula, per editar una paraula i per donar d’alta una llista d’estudi. Quan premem el botó que acompanya al formulari s’activa la funció que envia les dades al servidor mitjançant una petició Ajax amb paràmetres que interpretarà el servidor i que prèviament hem recollit del formulari.

També disposam d’una vista de tipus “actionSheet” que s’encarrega de mostrar tres botons extra a la pantalla del detall d’una paraula. Aquesta vista és molt útil com a menú contextual d’una altra vista. L’esborrat de paraules avisa abans amb un missatge de confirmació i en funció de la resposta farà una petició al servidor de tipus Ajax enviant l’identificador de paraula. L’edició de paraules ens portarà a la vista formulari d’edició per un posterior ús.

L’última vista agrupa varis components, entre ells un carrusel que presentarà les paraules de l’exercici Flashcards en format de tarjeta amb dos contenidors (Pregunta i Resposta), i un contenidor amb una barra de comptadors de paraules pendents i encertades. A més tenim dos botons que es mostraran només quan l’usuari estigui visualitzant la resposta de la pregunta i que la seva funció és incrementar els comptadors i demanar al servidor la següent paraula disponible (en el cas que faltin paraules per sortir).

Una vegada hem vist les vistes de l’aplicació explicarem els magatzems de dades que fa servir i els models associats. Com veiem, “paraulaJson” requereix el model “paraulaModel” mentre “llistaJson” fa servir el model “llistaModel”. Ambdós magatzems presenten manejats d’events en el cas que s’insertin, carreguin o esborrin registres, una agrupació per fer que el llistat d’elements es visualitzi d’una forma molt senzilla, i un Proxy configurat de tipus Ajax que llegirà del servidor les paraules carregades i d’un fitxer local les llistes d’estudis. Dels models podem dir que contenen els camps necessaris per cada entitat i una validació de presència per cada un d’ells.

Ja per acabar veiem que a l’apartat de recursos disposam del fitxer on definim tots els estils de les vistes.

Per finalitzar aquesta presentació, hem de dir que tot això no serviria si des del navegador no executam el fitxer “index.html”. Aquest s’encarrega de llançar l’aplicació.

Ha estat una experiència molt positiva haver desenvolupant una aplicació amb aquest marc de treball i espero que us hagi agradat aquesta presentació.