

Rapport TIG167 - Snutnytt

Erik Berggren

Oscar Hedendahl



Contribution

Vi har skrivit mer eller mindre all kod tillsammans. Erik fokuserade dock mer på själva Android-applikationen medan Oscar fokuserade mer på hämtning av data, parsing av JSON och servleten.

Tools

Vi använde oss främst av Android Studios, Atom, Github samt Cygwin/Bash. Klienten har i stort sett skrivits helt i Android Studios, vilket har fungerat bra. Stödfunktionerna i Android Studios upplevde vi var till stor hjälp. Med stödfunktioner menar vi till exempel hjälp med importsatser eller meddelanden om att fel objekttyp används.

Att använda Github för versionshantering av projektet har gått hyfsat bra, framförallt efter att vi insåg att det är bäst att inte ändra på samma fil samtidigt. Vi saknade därmed ett verktyg för att koda samtidigt på distans och se ändringar, att arbeta som i ett Google-dokument fast för programmering. Vi har inte gjort några större efterforskningar för att se om sådana verktyg finns. En annan funktion vi upptäckte ganska sent i projektet är att Android Studios har automatisk Git-integration, i form av att ändrade filer blåmarkeras och osynkade filer är grönmarkerade och så vidare.

Att använda Avd har mestadels gått bra. Fick ändra en inställning i BIOS på en Windows-dator för få det att fungera och en Mac-dator har haft lite ljudbuggar.

API:s

Vi använde oss framförallt av Polisens API som listar utvalda händelser i olika städer. Det var ett simpelt API som var enkelt att använda då endast ett fåtal parametrar kunde användas. Vi hade dock problem med att vår GET-request till deras api, till synes oregelbundet, valde att timea ut, vilket ledde till att vi inte fick något svar. Detta kunde åtgärdas genom att sätta en "retrypolicy" på vår request, vilket åtminstone gjorde så att man slapp trycka på "uppdatera-knappen" flera gånger i applikationen.

Vi använde oss också av Trafikverkets API. Detta api:t skilde sig från polisens då den ville ha ett POST-anrop (se <https://api.trafikinfo.trafikverket.se/API/>). Till en början förstod vi inte alls hur vi skulle gå tillväga för att göra en sådan request. Efter lite kikande på Stackoverflow så förstod vi att requesten skulle läsas från en XML-fil där vi specificerat våra parametrar. Trafikverkets API är mycket väldokumenterat men vi hade (och har) problem med strukturera upp POST-anropet samt att parse svaret för att få det precis som vi vill ha det. Vi fick det att fungera relativt bra i Atom/Bash men lyckades inte implementera det i appen (se Trafikverket-mappen i klienten).

Developer documentation

Så fort vi stötte på problem så använde vi Juneday eller Google (om det inte fanns beskrivit i wikin). Detta ledde till att vi ofta hamnade hos Oracle eller på Stackoverflow. När vi försökte använda oss av guider och tips från Stackoverflow så hade vi en tendens till att överarbeta problemet. Det var oftast mer simpelt än vad vi ursprungligen fick uppfattningen om.

Vi upplevde också handledningstillfällena som givande, då det var bra att diskutera koden och generellt få tips och riktlinjer.

Android platform

Det var helt nytt för oss att arbeta med Android som plattform. Det har ändå gått hyfsat smidigt. Kanske var det lite roligare att utveckla för en mer begränsad plattform med touch-kontroller, dvs. Android, jämfört med att utveckla för till exempel PC. Ställer kanske lite större krav på appens utformning och funktionalitet.

När det gäller det grafiska gränssnittet var det helt ok att arbeta med. Det krävdes en viss inlärningsperiod för att förstå Layout och Views. Vi upplevde även att det ibland blev problem när vi använde drag and drop funktionen i Android Studios. De värden som då skrevs automatiskt kunde vara oklara eller skapa problem senare. Vi undvek därmed att använda oss av det och skrev det mesta för hand i XML-filerna istället.

En annan reflektion är att vi kanske skulle skrotat vissa delar vi gjorde tidigt i projektet innan vi lärt oss allt. Ett exempel är den så kallade Expandable List View som vi importerade tidigt. Vi hade mindre kontroll över denna View senare i projektet.

En annan del vi arbetade med var lokalisering/resurser. Vi märkte att det var ett bra verktyg för att hantera samma strängar i olika delar av applikationen. När de var på plats var det även väldigt smidigt att översätta dessa och därmed ha stöd för flera olika språk. Ett problem var dock att staden i anropets URL behövde hårdkodas till de svenska varianterna, då polisens API kräver det. I nuläget har applikationen stöd för svenska, engelska och tyska.

En annan positiv erfarenhet var SharedPreferences som ett sätt att spara användarens inställningar. En fundering var dock om vi ska använda variabler som nycklar istället för text i de olika aktiviteterna. Resources verkade onödigt då de inte ska översättas. Vi lämnade de som text tills vidare.

Json

Vi använder inte vår servlet för att skicka data till vår Android-applikation (kommer i version 3.0), så vi kan inte reflektera över det. Att parsa själva Json:en var dock ganska simpelt om datan man behövde inte var alldeles för nästlad vilket den tyvärr var för datan vi hämtade från Trafikverket. Här behövde vi göra for-loopar i for-loopar i for-loopar för att få ut den datan vi ville åt. Detta kändes inte optimalt men det kanske är så man behöver göra.

Java servlet

Då vi inte använder servleten genom vår applikation så blir det svårt för oss att utvärdera den som en "backend-plattform". I nästa version av applikationen skulle vi kunna använda servleten för att cacha svaret från Polisens API, då det brukar ta ganska lång tid att få svar. För tillfället används servleten som ett PC-alternativ för de som inte har en Android-telefon men som ändå vill ta del av denna underbara applikation.

Av någon anledning så visas inte ÅÄÖ osv på vår localhost. Vi felsökte detta men hittade ingen lösning. Hittade denna [länk](#) där en annan person har samma problem: *"It looks like, by default, serving static HTML from winstone doesn't seem to work unless you explicitly specify the character encoding as UTF-8. A simple web-filter applied to all URLs helps us here"*. Vi lade till ett filter i servleten (se CustomCharacterEncodingFilter) men det verkar inte hjälpa. Vi valde att inte lägga mer tid på det då vi tror att det är Winstone som är boven.

Misc

Vi fokuserade mycket på frontend/applikationen i vårt projekt. Det blev naturligt då vi inte skulle göra så mycket mer än att visa händelserna vi fick från polisen. I ett nästa steg hade vi gärna implementerat fler API:er och utökat funktionaliteten.

Vidare har vi inte skrivit kod för landskapsläge, onResume och liknande situationer i Android. Vi har inte heller testat på nyare eller äldre APK, eller olika typer av telefoner. Vi inser att det kanske behövs men anser att projektets begränsade storlek kanske gör att det inte behövs just nu.

Betygsmässigt tror vi att applikationen/frontend är på VG-nivå, men servlet och backend kanske inte riktigt når upp dit. Om det inte är alltför långt för oss till VG får ni gärna säga till vad som behövs. Annars nöjer vi oss med ett G.

Slutligen anser vi att det kanske hade varit bra att under kursens gång fått se ett eller två projekt 'live' från förra året, för att förstå hur det ska fungera och hänger ihop samt vilka förväntningar ni har på våra egna projekt.