

Genomförda moment:

- Server
 - Databas med studenter och kurser
 - Servlet för att skicka data via HTTP
 - Presenteras via API för XML & JSON
- Klient
 - Android app
 - Tar emot data från våran server
 - Visar studenter och deras kurser
 - Kan lagra data i en lokal databas

Verktyg för utveckling:

Eclipse: Jag började skriva min server i eclipse och använde delvis några tips ifrån vårt crud projekt som vi har skapat i en annan kurs. Samt få lite hjälp i en IDE för att se lite mönster och få en överblick över projektet och snabbt kunna testa anrop till min servlet.

Winstone: När jag kände att helheten var klar med servern så tog jag och migrera över den till att kunna köra direkt i en Winstone server och startade den på en lokal dator hemma för att testa dela upp server och klient.

Postman: Innan jag skrev så mycket kod på android appen så använde jag mig utav en REST klient som heter Postman som är utvecklad utav Google för att kunna göra massa olika anrop till en server och få tillbaka data. **Ps.** REKOMENDERAR STARKT!

Vad kan jag göra bättre?:

Server: På min server så har jag delat upp det ganska rent med StorageManager och bra struktur genom hela projektet men vissa delar hade jag velat lägga mera tid på och snygga till så att det bli mera CLEAN CODE.

Min **Servelet** hade jag velat dela upp till flera servletar så som Hendrik och Rikards exempel med en som hanterar JSON och en som hanterar XML och en som tar emot svar.

Så som min servlet ser ut just nu är att den skickar parametrar om studenter och id till en klass som heter **Formatter.java** som i sin tur säger till **JSONFormater** eller **XMLFormater** att sätta ihop data och skicka tillbaka till min servlet som skickar den till användaren. När man läser koden så ser man vad den gör ganska snabbt men det kan bli annu tydligare !

Klient: På min android app så är jag väldigt nöjd med utseende och hur jag hanterar all data. Finns dock flera minus på Androids API för att jobba med vissa komponenter så som ListView och SubItems. Detta satt jag i 2-3 dagar och googlade för att hitta en snygg lösning utan något resultat. Men detta problemet skötte jag via ett/många fulhack.

API:s

Jag är väldigt nöjd överlag med dessa api:er som vi/ni har valt att vi ska jobba med. Det är ju väldigt inne just nu att jobba på detta sätt. Sen har vi skrivit mycket kod som det finns färdiga ramverk som löser mycket åt oss men det är ett stort plus att kunna skriva en egen JSON/XML formaterare osv, bra upplägg! Har inga negativa tillägg här.

Utvecklings dokumentation

Jag har kollat på en massa manualer, api:er, guider men jag hade tänkt sammanfatta dem viktigaste som jag gillar.

Android appen:

- okhttp3
 - OkHttpClient // Enkelt skapa kontakt till server
 - Cache
 - Timeouts
 - Connections Pools
- Retrofit
 - Call // För att hantera call och callbacks
 - CallBack // Enkelt med REST anrop !
 - Response // Innehåller converter factory
 - HTTP.POST / GET
 - Converter -> Gson/XML
- GSON
- Org.json // För att enkelt hämta datan via meta namn.
- Butterknife // För att snabbt hämta komponenter man har,
 - Bind // skapat i android.xml och för att skapa onclick,
 - OnClick // funktioner enkelt i koden.

Blir väldigt grötigt med länkar till varje API så jag hänvisar till google och deras Github sida för att hitta exempel, det finns massa utav dem.

Server side:

- Har inga coola api:er över huvudtaget...

Något jag har missat?

Skriv gärna ett mail asap så fyller jag på med information.

Android platform punkten skrev jag snabbt om i **Verktyg för utveckling**.

Men kan tillägga att jag gillar Android överlag, men lite kompricerat att sätta sig in i. Border finnas lättare sätt att skriva koden, men vem vet Swift och iOS kanske är annu svårare. Får blir Xamarin och C# annars och försöka bygga så native det går ;)