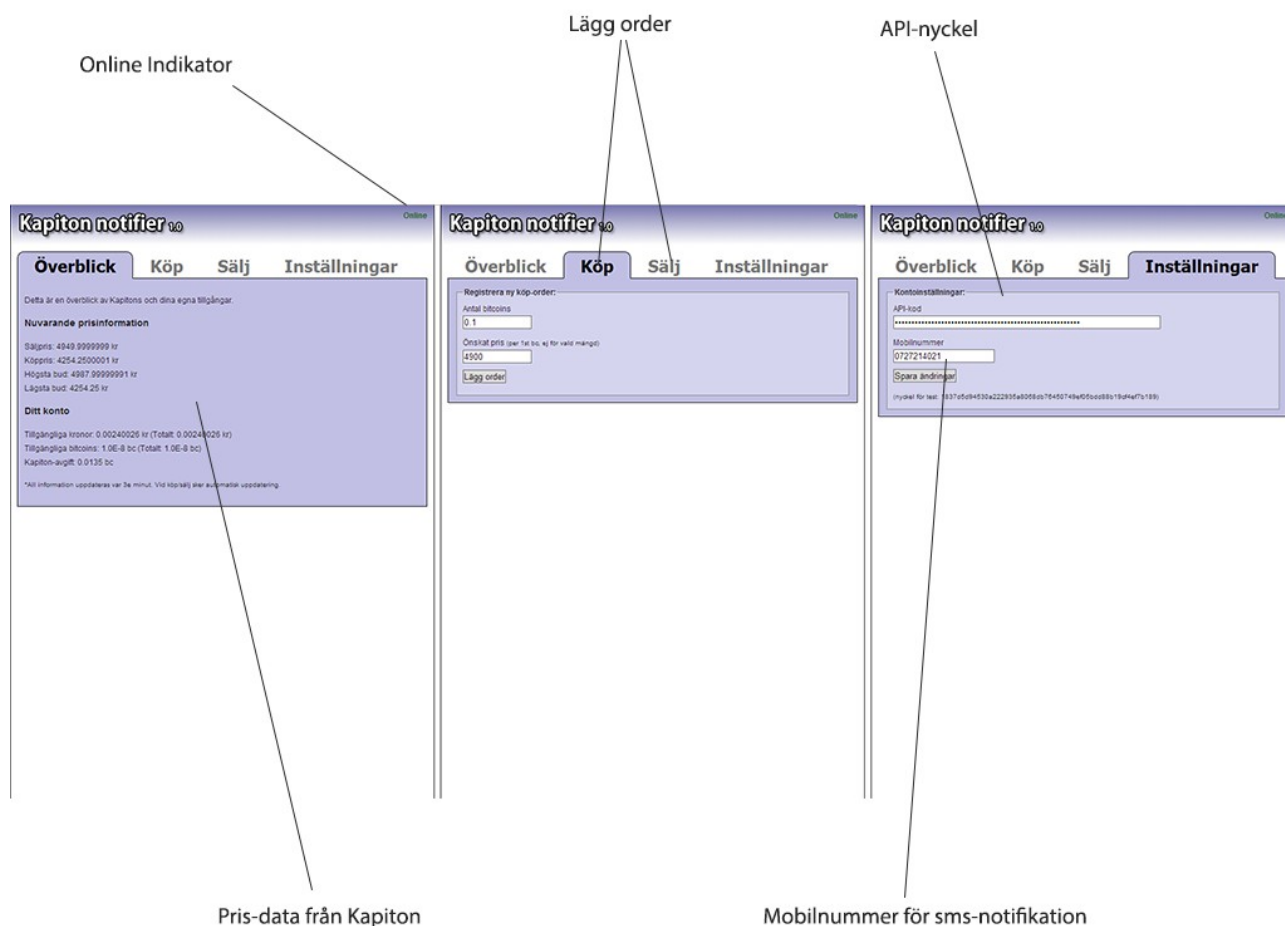


# Kapiton Notifier

## Inledning

Jag har gjort en applikation som med hjälp av Kapiton-API och Mosms-API kan utföra valutaväxlingar mellan Bitcoins och svenska kronor. Utöver detta ska applikationen hålla reda på de ordrar man lagt för att vid lyckat köp/sälj meddela användaren via sms att köpet/säljet ägt rum.

Jag sysslar själv mycket med att köpa/sälja bitcoins, men upptäckte fort att det kräver mycket tid framför skärmen, endast för att hålla reda på om en order gått igenom eller inte. Det kan ta allt från 1 sekund till 1 år innan en order går igenom på önskat pris. Därför kom jag på denna applikation, som skulle kunna användas på en mobil enhet och informera mig direkt så en transaktion är genomförd.



# Serversidan

Jag har valt att använda mig av php under detta projekt utan att använda mig av några ramverk utöver detta. Serversidan består av 2st huvuddelar. Den första delen laddar applikationens basstruktur och klient-funktionallitet. Den andra delen är gjord för ajax-anrop endast.

Ajax-anrop görs antingen som en "save" eller en "getContent". Serversidan identifierar vilken av de 4 sektionerna som det handlar om och vilket utförande man vill göra. Sedan anropas cashad-data eller direkt data från en webservice.

All inkommande data (tas från querystrings/cookies) säkras från injections innan de används. Det som inte säkras på servern är inkommande data från Kapiton, som jag i detta fall litar på då de inte består av någon data som inte är direkt server-producerad. Dvs ingen användare har något inflytande på det data kapiton levererar.

Då huvud-api är Kapitons api så cashas ingen "flik-data" data mer än i 3 minuter som den cashas på klientsidan. Detta då datan konstant uppdateras med ibland bara sekunders mellanrum. Där emot cashas ordrar som läggs av användaren i en mysql-databas. Detta så att applikationen kan identifiera order som gått igenom och meddela användaren vid det angivna mobilnummer som finns bundet till varje order.

Felhanteringen finns för de lägen då webservice inte svarar, ger felaktig data eller inte kan genomföra en order. Utöver detta finns felhantering för felaktigt inmatad data från användaren. Alla felmeddelanden visas i respektive View, men stoppar inte applikationen från att fortsätta. Dvs ingen uppdatering av applikationen behövs i något läge.

# Klientsidan

På klientsidan används JQuery som enda ramverk. Ajax-anrop sker vid all navigering och cashas lokalt med hjälp av javascript i 3 minuter, eller tills dess att en save/post har gjorts, då cashen töms.

Applikationen cashar även alla bas-resurser som .js-filer, bilder och .css filer med hjälp av HTML5 (Manifest). Detta gör att laddningstiderna minskar efter första besöket.

Utöver cashning, används även javascript-minifier för att förminska storleken på en annars ganska stor javascript-fil. Orginalet finns dock med i samma katalog. All javascript och all css är helt placerade i externa filer och så mycket som möjligt av duplicerad kod har tagits bort.

Applikationen reagerar på om internet försvinner genom att visa en "online/offline" indikator i övre högra hörnet. Denna reagerar i realtid om en anslutning försvinner. Vid ett sådant tillfälle sparas cashade sektioner permanent tills dess att nätverket åter finns tillgängligt. Alla anrop till servern resulterar i att användaren får ett meddelande om att anropet inte går att genomföra i offline-läge.

Designen är ganska enkel men då den är menad att vara för en mobil enhet finns vissa punkter som är värda att peka ut. Jag har valt att göra knappar för varje "flik" något större än vad man väntar sig av en vanlig applikation. Detta då det ska vara lätt att komma åt. Jag har även valt att göra applikationen i 960 \* 640 som standard då tanken var att senare göra applikationen responsive. Detta har jag dock inte med.

# Egen reflektion

Till en början gick arbetet med applikationen mycket bra. Men mot mitten av perioden så uppstod stora problem när Kapitons VD försvann med stora mängder pengar och med en ej funktionellt Api som resultat. Inga transaktioner kan göras varken ut eller in till Kapiton. Därmed försvann en enorm del av syftet med min applikation.

Efter diskussioner med John kom vi fram till att det inte fanns mycket att göra åt detta och att jag skulle utveckla vad som gick, med detta som utgångspunkt. Det resulterade i att min applikation kan hämta data i form av valuta-information, samt användarens egna status ifrån kapiton. Inga transaktioner kan dock göras, utan detta simuleras istället.

Det resulterade också i att jag inte kan göra en check av ”utförda ordrar” ifrån Kapiton, vilket var utgångspunkten med att använda mig av MOSMS-api. Funktionalliteten finns i bakgrunden (/ajax/model/MmoWebservice.php), men inga skarpa utskick sker ifrån MOSMS iom detta.

Det finns mycket implementation som jag inte kunnat utföra. Ännu värre finns det implementation som jag utfört som inte är en del av betygskraven för betyg 3, utan för högre nivåer. Detta i tron att jag skulle hinna med en högre betygsnivå. Exempel på detta är offline-funktionallitet.

I dagsläget ser jag inga möjligheter att jobba vidare med min applikation då Kapiton var ensamma med att tillhandahålla ett API för sin tjänst. Men i framtiden hoppas jag se flera aktörer på marknaden som öppnar upp för detta.

## Risker

Det finns en del risker med min applikation. Då den är utvecklad för mobila enheter så reflekterade jag till en början kring hur vida jag behövde ha en autentisering över huvud taget. Det fanns enligt mig inga fördelar med detta då en användare inte får ut någonting av ett medlemskap, utan all funktionallitet kretsar kring användarens api-nyckel som kommer ifrån kapiton och som sparas i användarens cashe.

Detta skapar ett litet problem med säkerhet kring api-nyckeln. Vem som helst kan ange en api-nyckel utan att på annat sätt bevisa sin identitet, och på så sätt utföra transaktioner. Detta hade kunnat hindras om ett eget medlemskap fanns där api-nyckeln kunnat lagras krypterat och där endast en användare får möjligheten att använda 1<sup>st</sup> unik api-nyckel. Detta är något jag inte har åtgärdat i slutskedet men som jag kommer att tänka mer på framöver.

I sin helhet är jag ganska missnöjd med applikationen då jag velat så mycket mer än vad resultatet gav. Inte enbart för studierna utan av eget intresse. Jag tror att med lite mer tid och ett framtida fungerande api, kan applikationen fylla ett riktigt syfte för de som idag handlar med bitcoins.

# Betygshöjande

Jag anser mig inte berättigad för ett högre betyg då hela iden med min applikation föll lite över stupet när Kapitons VD försvann. Detta i samband med ett stressigt projekt i en annan kurs gjorde att jag inte riktigt kom tillbaka i den rytm jag ville och projektet blev rörigt och ostrukturerat mot sitt slut.

Då jag kan känna att min applikation brister lite i vissa punkter finns det dock viss implementation som är vad menad att vara betygshöjande som jag ändå hoppas kan ha en viss påverkan på hur applikationen ses i sin helhet.