API:

API står för Application Programming Interface och är en ihopsättning av regler som kontrollerar hur olika programvaror kommunicerar. Reglerna är allt som oftast olika funktionsanrop som ser till att man på ett kontrollerat sätt kan använda en annan programvaras funktionalitet. Kända siter som har API är Facebook, Twitter och YouTube.

Bibliotek(Lib):

Innehåller förskriven kod av funktioner, klasser med mera.

Ramverk:

Används för att effektivisera arbetet av en programvara. Ramverk gör så programmeraren kan fokusera på de mer unika och avancerade delarna i utvecklingen istället för att behöva bygga grunden. I ett ramverk kan man använda färdiga funktioner som gör mycket av grundjobbet. Kända PHP-ramverk är CodeIgniter, Zend och CakePHP.

Interface:

Interface är den del där olika komponenter kommunicerar. Det kan vara klienten och en människa. Då är utseendet på sidan interfacet.

Applikation:

En programvara som är utvecklad för att hjälpa användaren utföra en handling. Det kan vara att köpa saker från internet, då är webbshoppen en applikation.

CMS:

Betyder Content Management System. Det används för att hantera innehåll på en sida. Det kan vara att publicera nya saker eller redigera innehåll. Allt detta sker via ett gränssnitt som är webbaserat. Alla bloggsidor har ett CMS. CMS används för att människor utan kunskap om HTML ska kunna driva webbsidor. Det mest populära CMS:et är Wordpress och även Joomla är populärt.

Komponent:

En del av ett system.

Klass:

Del av ett system som man kan skapa flera objekt av. Klasser innehåller attribut och metoder.

Hur bör er komponent dokumenteras?

Det som borde finas med i dokumenteringen är instruktioner till hur man ska installera den. Detta är den viktigaste delen. Dessutom borde instruktioner till hur man ändrar vilket element som ska skrapas med så en annan person kan modifiera komponenten till sin egna sida och syfte.

Hur skapar vi kod som använder ett api men inte är beroende av det? Hur skyddar vi oss från förändringar i api’t?

För att inte vara beroende av det måste man väl se till att det finns back up ifall det misslyckas. Kanske nyttja ett annat liknande api eller skriva något eget, lite enklare som kan fungera som en tillfällig lösning. I detta fall är de nerladdade API:erna cURL och SimpleTest nerladdade och uppdateras därför inte utan att jag vet om det. På så sätt kan jag uppdatera och se till att det fungerar och behöver inte oroa mig för att det uppdateras när som.

Hur installeras det valda API’et?

cURL installerade jag via easyphp. Då gick jag bara in i php:s config-fil och tog bort kommenteringen av cURL. Efter en omstart av easyPHP fungerade cURL.

SimpleTest laddade jag ner och la mappen i mitt projekt. I min test-fil skrev jag ”require\_once(”simpletest/autorun.php”) vilket inkluderar SimpleTest i mitt projekt.

Hur installeras er komponent?

Jag gjorde en liknande lösning som SimpleTest. För att installera laddar man ner min mapp som är döpt till webscraper och lägger den i sin mapp. Sedan skriver man, förslagsvis i index.php, require\_once(”webscraper/webscraperController.php”). Efter detta anropar man funktionen DoControll och ekar sedan ut resultatet ur denna funktion på valfritt ställe. För att sökningen ska fungera måste man ha ett sökfält där name=”searchField” och en submit-knapp med name=”searchButton”.

Hur hanterar man sökvägar i er komponent?

Jag använder Simple HTML Dom för att söka sig till rätt element. Detta var väldigt enkelt att jobba med och fungerar likadant som sökvägar i bla jQuery och xPath. Man kan ändra vilket element man vill plocka ut orden ur via en sessions-variabel som finns i kontrollen. Detta för att det ska vara enkelt för andra att anpassa komponenten till sin sida och syfte.

Hur skapas bra automatiska testfall för en kodskrapa?

Jag gjorde så att jag skapade en testsida med HTML, liknande den som det är tänkt man ska skrapa från sen, vilket är blogg.se. I testfilen gjorde jag sedan en lång textsträng innehållandes samma källkod som testsidan. När man sedan skrapar testsidan ska resultatet från skrapningen vara samma som textsträngen. Då vet man att man fick med alla element från skrapningen.

Hur hanterar man fel vid skrapning av data?

Vid användning av curl kan man använda CURLOPT\_FAILONERROR vilket innebär att om sidan man försöker skrapa returnerar en felkod på 400 eller högre så stängs skrapningen ner och ens sida visas som normalt för användaren. Jag gjorde så att om användaren försökte skrapa en sida som inte fanns så kollade jag i kontrollen om funktionen returnerade en tom sträng så avbröt jag övriga funktioner och skickade tillbaka ett felmeddelande.

Hur hanterar man cache av skrapat data?

Bästa är väl att spara ner sin skrapningar i en databas och sätta en tidsgräns på hur länge den cachade datan ska vara giltig. När man sedan gör en sökning kollar man i databasen innan man skrapar om det finns någon giltig data där och använder i så fall den istället för att skrapa ny.

Vad är och hur fungerar JSON? Finns det andra tekniker för samma sak?

JSON står för JavaScript Object Notation. Det är ett textbaserat format som används för att utbyta data mellan olika språk, och är designat så att det är läsligt för en människa. Tack vare designen blir det lätt att jobba med och lätt att förstå hur det fungerar. JSON används ofta när man vill hämta information från en annan databas så att man i sin egna applikation kan ta in och hantera datan. Ett alternativ till JSON är XML.

Hur hanteras inloggning via Facebook?

Det finns två varianter man kan använda när man jobbar med inloggning via Facebook. Den ena är inloggning via server och den andra via klient. Båda dessa använder OAuth-protokollet för att sköta inloggningen. Inloggning via server används när man anropar Facebooks API från servern och inloggning via klient när man anropar från klienten, logiskt. Inloggningen går till i tre steg. Först kollar OAuth om användaren som ska logga in är den användare som den uppges sig vara. Om detta stämmer får användaren tillåta att han eller hon vill använda den här appen och får information om vad för information som blir tillgänglig denna appen. När detta har godkänts så kollas det så att denna information som ges ut av användaren verkligen går till appen i fråga och att det inte är någon annan som ligger och stjäl denna information. När de här tre stegen har gått igenom på ett lyckat sätt får appen en så kallad user access token som är en nyckel så man kan använda informationen från användaren.

Vilka risker finns det för användaren som använder er appliaktion?

När man skapar en applikation kan man begära olika informationer från användaren. Något som skulle vara riktigt elakt är att begära informationen till användarens vänner. Sedan kan man ta deras information och använda till något helt annat än vad appen sa att man skulle ha det till. På så sätt blir vännerna smutskastade eller utsatta av något slag och blir därmed arga på killen som tillät att deras information användes av appen.

På vilka grundtekniker vilar Facebooks api?

Javascript, PHP, JSON. Sedan finns det verktyg för utveckling till Android och iOS. OAuth för autentisering.

Hur dokumenteras Facebooks api?

Facebook har sin dokumentation uppdelad i flera delar på <http://developers.facebook.com/docs/>. De bygger på kodexempel för hur man göra grunderna och instruktioner för hur man ändrar de lite mer avancerade delarna. Som komplement till detta får man screenshots på förväntade resultat. Detta gör att man kan kopiera koden därifrån och veta att man får en bra grund. Det finns dessutom videos. Om man vill använda sig av ett plugin finns det kodgeneratorer så man ställer in lite inställningar och sedan får man genererad kod som man bara kan klistra in i sin egen kod. Facebooks kod är ganska avancerad och det är därför de ger ut mycket exempel som man bara kan kopiera. Alla metoder som går att använda finns upplistade och man kan gå in och läsa på om vad deras funktion är. Överlag kan man sammanfatta att det är en väldigt bra dokumentation och om man använder detta minskar det arbetet för en själv rejält då man kan klippa och klistra mycket.

Hur delas och uppdateras Facebooks api?

Delningen sker på olika sätt beroende på om man använder Javascript, PHP, iOS eller Android. Med Javascript hämtar man biblioteket direkt från webben utan att behöva ladda ner något. Med de övriga måste man ladda ner och sedan lägga in mappen på önskat ställe. Uppdateringar sker en gång i veckan då löser buggar som finns. De har en blogg och en changelog man bör ha ögonen på om man är beroende av Facebooks api för att se vilka uppdateringar som gjorts och som är planerade. Märker man att ens app slutar fungera gör man bäst i att gå in och kolla vad som ändrats i api:et och sedan gå in och ändra själv för att få det att fungera. Man kan också ställa in att ens app prenumererar på uppdateringar från Facebook så man slipper uppdatera själv. Denna funktion heter Real-time Updates.

Hur plockar vi bort funktionalitet från ett api? Hur har facebook gjort?

Hur hanteras felmeddelanden i facebooks api?

De är 400 http-errors och levereras i JSON-format. Med en typ och ett meddelande. För att man lätt ska veta vad för typ felet är.