

**DISCOGS-MARKKINAPAIKAN INTEGROINTI
VERKKOKAUPPAAN: CASE LEVYKAUPPA ÄX**



Ammattikorkeakoulututkinnon opinnäytetyö
Tieto- ja viestintäteknikka, insinööri (AMK), Riihimäen kampus

Kevät, 2021

Anssi Ahola

TIIVISTELMÄ

Tämän opinnäytetyön tavoitteena oli selvittää Discogs-palvelun rajapinnan tarjoamat ominaisuudet ja kuinka ne voitaisiin integroida asiakkaan hallintajärjestelmän kanssa. Integroinnin tarkoituksena oli keskittää ja automatisoida asiakkaan verkkokaupan tuotteiden myynti ja hallinnointi Discogs-markkinapaikalla. Työn toimeksiantajana toimi Sovelluskontti Oy ja työn sisältö toteutettiin Levykauppa Äx Oy:lle.

Työn teoriaosuudessa tutustuttiin Discogs-palveluun, siellä toimivaan markkinapaikkaan sekä palvelun tarjoamaan avoimeen rajapintaan ja sen taustalla toimiviin toteutuksiin ja teknologioihin. Teoriaosuudessa käytiin myös läpi kilpailijoita sekä vaihtoehtoja Discogs-markkinapaikalle ja Discogsin tarjoamalle rajapinnalle.

Työn lopputuloksena syntyi selvitys Discogs-rajapinnan ominaisuuksista, kattava ohjelmakirjasto integrointia varten sekä alustava selvitys Discogs-markkinapaikan kautta tehtyjen tilausten automaattisesta käsittelystä. Työn lopussa toteutettiin myös pienimuotoinen ohjelma demonstroimaan ohjelmakirjaston ominaisuuksia asiakkaalle.

Avainsanat Discogs, http, restful

Sivut 36 sivua

ABSTRACT

The aim of this thesis was to examine the features of the Discogs API (Application Programming Interface) and how those features could be integrated with the client's management system. The goal of the integration was to centralize and automatize the selling and managing of the client's products in the Discogs marketplace. This thesis was commissioned by Sovelluskontti Oy and the contents of this thesis were made for Levykauppa Äx Oy.

The theoretical section of this thesis introduces Discogs, its marketplace, Discogs API and the underlying technologies. It also covers some of the competitors and alternatives for the Discogs marketplace and the Discogs API.

The result of this thesis was an examination of the features that the Discogs API provides, a comprehensive program library for integration with the client's management system, and a preliminary investigation of the automatic processing of orders placed through the Discogs marketplace. At the end of this thesis, a small-scale program was also made to demonstrate the program library features to the client.

Keywords Discogs, http, restful

Pages 36 pages

Sisällysluettelo

1	Johdanto	1
2	HTTP.....	2
2.1	Metodit	3
2.2	Otsakkeet	4
2.3	Tilakoodit.....	5
3	Rajapinnat ja Oauth.....	6
3.1	RESTful API	8
3.2	OAuth	10
4	Discogs.....	12
4.1	Rajapinnan tarjoamat markkinapaikan ominaisuudet	14
4.2	Autentikointi	16
4.3	Vaihtoehdot/Kilpailijat.....	19
4.3.1	Spotify	19
4.3.2	MusicBrainz	20
4.3.3	Ebay	21
4.3.4	CDandLP	21
5	Toteutus.....	22
5.1	Suunnitelma	22
5.1.1	Myynti-ilmoitusten saldojen päivitys.....	23
5.1.2	Tilauksen käsittely	25
5.2	Integroitava ohjelmisto.....	28
5.3	Demo	29
5.4	Lopputulos.....	32
6	Yhteenveto	32
	Lähteet.....	34

Kuvat, taulukot ja kaavat

Kuva 1. Selain noutaa dokumentteja useista lähteistä (Mozilla, n.d.-a).....	2
Kuva 2. GET-kutsu tools.ietf.org sivustolle.....	3
Kuva 3. Asiakasohjelmat keskustelevat eri palveluiden kanssa yhtenäisen rajapinnan kautta. (Hurbans & Van Der Westhuizen, n.d.)	9

Kuva 4. Korkean tason kuvaus OAuth:in toiminnasta (Richer & Antonio, 2017, s. 12). .	11
Kuva 5. Discogs:in historian kohokohdat (Discogs, 2020).	13
Kuva 6. Oauth request_token pyyntö.	18
Kuva 7. Oauth access_token pyyntö.	18
Kuva 8. Diagrammi myynti-ilmoitusten määrän päivityksestä.	25
Kuva 9. Diagrammi tilauksen käsittelystä.	27
Kuva 10. Demon 1. osa, tuotteet ja haku.	30
Kuva 11. Ponnahdusikkuna Discogs-tietojen lisäämiseen.	30
Kuva 12. Demon 2. osa, Discogs-tiedot.	31
Kuva 13. Demon 3. osa, myynti-ilmoitukset.	31
Taulukko 1. RFC 7231:n määrittelemät tilakoodit ja selitykset (IETF, 2014, s. 48).	6
Taulukko 2. API-luokittelu (Moilanen, 2018, s. 58).	8
Taulukko 3. Median kuntoluokitukset (Discogs.com, n.d.-c).	15
Taulukko 4. Discogs-palvelun autentikointitavat (Discogs.com, n.d.-a).	16

1 Johdanto

Työn tavoitteena oli tutkia miten asiakas voisi hyödyntää Discogs-palvelua omassa liiketoiminnassaan ja toteuttaa integraatio Discogs-palvelun tarjoaman avoimen rajapinnan ja asiakkaan hallintajärjestelmän välillä. Työn tarkoituksena on yhtenäistää ja keskittää fyysisen musiikin jälleenmyyntiä, käyttäen asiakkaan hallintajärjestelmää asiakkaan oman verkkokaupan kuin myös Discogs-markkinapaikan kautta tehtyjen tilausten käsittelyyn.

Työn toimeksiantajana toimi Sovelluskontti Oy. Työn varsinainen sisältö toteutettiin Levykauppa Äx Oy:n tarpeisiin. Levykauppa Äx Oy:n (jäljempänä asiakas) liiketoiminta koostuu fyysisen musiikkimedian myynnistä sekä käytettyjen levyjen ostamisesta ja niiden jälleenmyynnistä.

Asiakkaan tarpeena oli saada tuotteitaan enemmän ja helpommin myyntiin Discogs-palvelun markkinapaikalle, jotta niiden myynti sekä ulkomaanmyynti kasvaisi. Asiakas oli jo ennestään tehnyt koemyyntejä Discogs-markkinapaikalla, mutta myynti-ilmoitusten luominen ja ylläpitäminen oli tehty manuaalisesti Discogs-palvelun kautta.

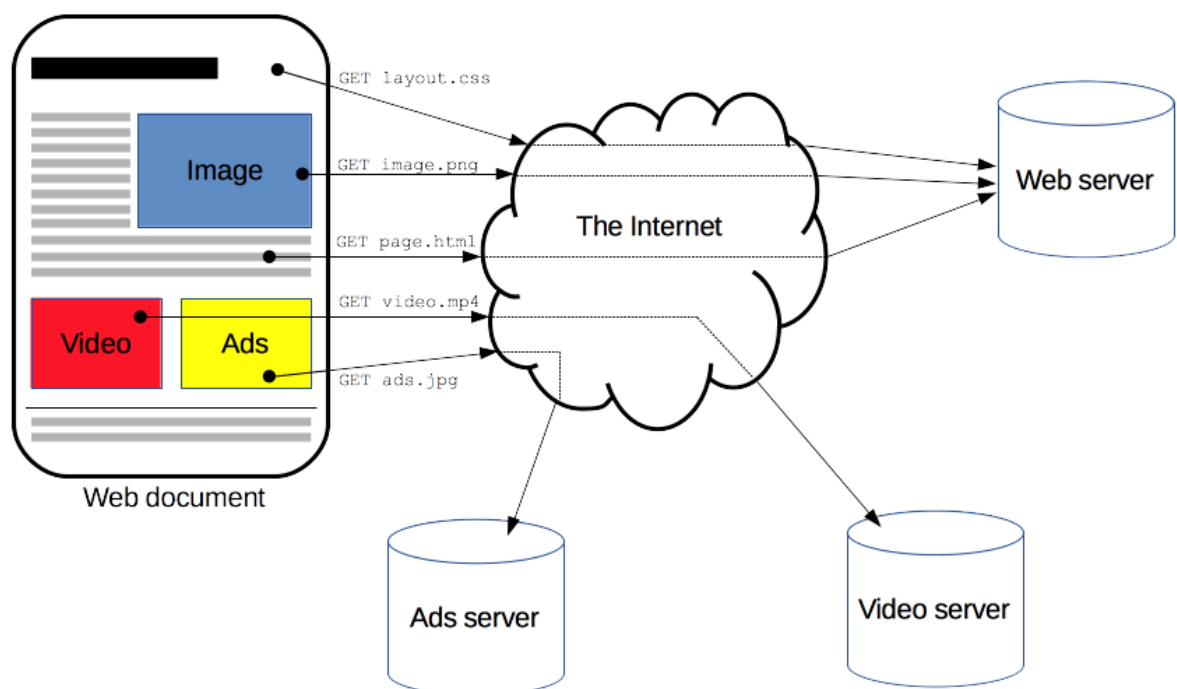
Työn lopputuloksena tulisi syntyä toimiva integraatio, jonka avulla asiakkaan varastossa olevista tuotteista luotaisiin automaattisesti myynti-ilmoitukset Discogs-palvelun markkinapaikaan ja ilmoituksien ylläpitäminen olisi ainakin osittain automatisoitu. Tämän lisäksi Discogs-palvelun kautta tulleet tilaukset tulisi käsitellä samalla tapaan kuin asiakkaan verkkokaupan kautta tulleet tilauksetkin. Tilauksen käsittelyn vaiheet, joita ei voida automatisoida, tulisi asiakkaan kyetä käsittelemään keskitetysti hallintajärjestelmästä.

2 HTTP

HTTP (Hypertext Transfer Protocol, suom. Hypertekstin siirtoprotokolla) on sovellustason tiedonsiirtoprotokolla hypermedia dokumenttien (kuten HTML) siirtoon, jossa asiakasohjelma avaa yhteyden palvelimelle ja lähettää pyynnön. Palvelin puolestaan lähettää asiakasohjelmalle vastauksen, joka sisältää tilakoodin, otsakkeet sekä mahdollisen vastausdatan. HTTP on tilaton (stateless) protokolla, eli palvelin ei ylläpidä tietoa asiakasohjelman tilasta. (Mozilla, n.d.-b)

HTTP-protokolla on webin tukipilari, jonka päällä suurin osa internetin palveluista toimii. Kuvassa 1, selain hakee HTTP:n avulla sivun rakenteen (HTML), tyylittelyt (CSS) ja kuvat web-palvelimelta, sivun sisäisen videon videopalvelimelta sekä mainokset mainospalvelimelta ja kokoaa niistä lopullisen sivun. (Mozilla, n.d.-a)

Kuva 1. Selain noutaa dokumentteja useista lähteistä (Mozilla, n.d.-a).



Asiakasohjelman (esim. selaimen) tekemä pyyntö pitää HTTP 1.1 -spesifikaation (IETF, 2014, ss. 32-46) mukaan sisällään:

1. HTTP-metodin, joka kertoo palvelimelle pyynnön tarkoituksen. Esim. GET-metodi hakee tietoa palvelimelta.
2. Resurssin URI (Uniform Resource Identifier) muodossa, esimerkiksi `/html/rfc7231`.
3. Pynnön otsakkeet, jotka antavat pyynnölle kontekstin, esimerkiksi "Host" otsake määrittelee resurssin osoitteen (esim. `tools.ietf.org`) ja "User-Agent" otsake kertoo asiakasohjelman metatiedoista.
4. Pynnön rungon (Body), joka sisältää mahdollisen lähetettävän datan. Esim. POST-pynnöissä.

Palvelimen lähettämä vastaus on pääosin hyvin samanlainen kuin lähetetty pyyntö, suurimpana erona HTTP-metodin sijaan vastauksessa on tilakoodi, joka kertoo, onnistuiko pyyntö (esim. 200 OK) vai tapahtuiko virhe (esim. 409 Conflict). (IETF, 2014)

2.1 Metodit

HTTP-metodit ovat suurin kirjaimin kirjoitettuja merkkijonoja, jotka määrittelevät minkä operaation palvelin tulisi suorittaa määritellylle resurssille (IETF, 2014). Esimerkkinä kuva 2, jossa GET-kutsu määrittelee asiakasohjelman pyytämään dataa osoitteen (Host) **tools.ietf.org** resurssista **/html/rfc2616** ja asiakasohjelman hyväksymään vastauksena palautetun datan HTML tai XML muodossa (Accept).

Kuva 2. GET-kutsu tools.ietf.org sivustolle.

```
GET /html/rfc2616 HTTP/1.1
Host: tools.ietf.org
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:85.0) Gecko/20100101 Firefox/85.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
DNT: 1
Connection: keep-alive
```

HTTP 1.1 -spesifikaatiossa (IETF, 2014, ss. 20-32) on määritelty yhdeksän metodia:

1. GET. Pyytää palvelinta palauttamaan resurssilla osoitettua dataa.
2. HEAD. Muilla tavoin identtinen GET-metodin kanssa, mutta estää varsinaisen datan palautuksen vastauksessa.

3. POST. Pyytää palvelinta vastaanottamaan viestin rungon ja suorittamaan sen itse määrittelemän funktion tälle. Esimerkiksi **POST /foorumi/2/viesti** voisi pyytää palvelinta luomaan uuden viestin foorumin alaosion 2 alle.
4. PUT. Pyytää palvelinta luomaan tai muokkaamaan kutsussa määriteltyä resurssia riippuen siitä onko resurssi jo olemassa.
5. PATCH. Muilla tavoin identtinen PUT-metodin kanssa, mutta jos resurssi on jo olemassa, tästä luodaan uusi versio käyttäen viestissä määriteltyjä muokkauksia (IETF, 2010a).
6. DELETE. Pyytää palvelinta poistamaan kutsussa määritellyn resurssin.
7. CONNECT. Luo kaksisuuntaisen kommunikaation (tunnelin) palvelimen kanssa.
8. OPTIONS. Pyytää palvelimelta sallitut asetukset ja tavat, joilla sen kanssa voi kommunikoida.
9. TRACE. Tarkastaa reitin resurssille. Palvelin palauttaa asiakasohjelmalle viestin tuloksista jättäen pois kaiken arkaluonteisen datan.

2.2 Otsakkeet

HTTP-viestejä lähettäessä sekä vastaanottaessa niiden mukana kulkee aina otsakkeita, jotka muun muassa antavat lisää kontekstia viestille, resurssille, viestin mukana tulevalle datalle sekä lisätietoa viestin lähettäjistä. Firefox nettiselaimestaan tunnetun Mozillan mukaan (Mozilla, n.d.-c) HTTP-otsakkeet voidaan jakaa neljään eri kategoriaan näiden kontekstin perusteella

1. Yleiset otsakkeet. Koskevat pyyntöjä kuin vastauksiakin, eivätkä viittaa millään tapaa viestin sisältöön. Esimerkiksi "Cache-Control".
2. Pyyntö otsakkeet. Koskevat nimensä mukaisesti vain pyyntöjä ja sisältävät lisätietoa pyynnössä määriteltyä resurssia sekä pyynnön lähettänyttä asiakasohjelmaa. Esimerkiksi "User-Agent", "Accept-*" otsakkeet ja "Authorization".
3. Vastaus otsakkeet. Sisältävät lisätietoa palvelimen lähettämästä vastauksesta pyyntöön, esimerkiksi pyynnössä määritellyn resurssin sijainnin palvelimella tai lisätietoa palvelimesta. Esimerkiksi "Age", "Location" tai "Server".

4. Entiteetti otsakkeet. Sisältävät metatietoja viestin mukana tulleesta datasta, kuten "**Content-Length**" tai "**Content-Type**".

2.3 Tilakoodit

Palvelimen lähettämät vastaukset asiakasohjelman pyyntöihin sisältävät aina tilakoodin (Status Code), joka kertoo, onnistuiko pyyntö. HTTP-spesifikaation mukaan (IETF, 2014, ss. 46-63) tilakoodit voidaan lajitella viiteen kategoriaan:

1. Informatiiviset koodit (1xx) ovat tilapäisviestejä tai koodeja, jotka kertovat yhteyden tai pyynnön tilasta. Esimerkiksi **100 Continue**, pyyntö oli hyväksyttävä ja asiakasohjelma voi jatkaa pyyntöjen tai viestien lähettämistä (esim. suuren tiedoston lähetys).
2. Onnistuneiden pyyntöjen koodit (2xx) kertovat palvelimen vastaanottaneen, ymmärtäneen ja käsitelleen pyynnön onnistuneesti. Esimerkiksi **201 Created**, asiakasohjelman lähettämän pyynnön prosessointi on onnistunut ja resurssi on luotu.
3. Uudelleenohjauksen koodit (3xx) kertovat, että asiakasohjelman tulee tehdä lisätoimenpiteitä pyynnön suorittamiseksi. Esimerkiksi **301 Moved Permanently**, pyynnössä määritelty resurssi on siirretty uuteen osoitteeseen (URI) pysyvästi ja tulevaisuudessa kaikki samaan resurssiin viittaavat pyynnöt tulee tehdä tähän uuteen osoitteeseen.
4. Asiakasohjelmavirhekoodit (4xx) kertovat, että asiakasohjelman tekemä pyyntö ei onnistunut ja syy tähän on asiakasohjelman puolella. Esimerkiksi **405 Method Not Allowed**, pyynnössä määritelty metodi ei ole tuettu määritellylle resurssille.
5. Palvelinvirhekoodit (5xx) kertovat, että palvelimen puolella on tapahtunut virhe tai palvelin ei kykene suorittamaan pyydettyä metodia. Esimerkiksi **501 Not Implemented**, palvelin ei tue kyseistä toiminnallisuutta suorittaakseen pyynnössä määritellyn toiminnan.

Tilakoodin mukana tulee aina myös lyhyt selitys sen tarkoituksesta (Reason-Phrase), esimerkiksi tilakoodin 405 selitys on "Method Not Allowed". HTTP 1.1 -spesifikaatiossa on

määritelty lista varattuja tilakoodeja ja niiden selityksiä (Taulukko 1). Näiden tilakoodien lisäksi on määritelty muitakin tilakoodeja eri spesifikaatioissa, joista listaa ylläpitää Internet Assigned Numbers Authority (IANA) (IETF, 2014, s. 49; IANA, n.d.). Spesifikaatiossa määritellyt tilakoodien selitykset ovat vain suosituksia ja niiden muokkaaminen ei vaikuta protokollan toimivuuteen (IETF, 2014, s. 47).

Taulukko 1. RFC 7231:n määrittelemät tilakoodit ja selitykset (IETF, 2014, s. 48).

Code	Reason-Phrase	Defined in...
100	Continue	Section 6.2.1
101	Switching Protocols	Section 6.2.2
200	OK	Section 6.3.1
201	Created	Section 6.3.2
202	Accepted	Section 6.3.3
203	Non-Authoritative Information	Section 6.3.4
204	No Content	Section 6.3.5
205	Reset Content	Section 6.3.6
206	Partial Content	Section 4.1 of [RFC7233]
300	Multiple Choices	Section 6.4.1
301	Moved Permanently	Section 6.4.2
302	Found	Section 6.4.3
303	See Other	Section 6.4.4
304	Not Modified	Section 4.1 of [RFC7232]
305	Use Proxy	Section 6.4.5
307	Temporary Redirect	Section 6.4.7
400	Bad Request	Section 6.5.1
401	Unauthorized	Section 3.1 of [RFC7235]
402	Payment Required	Section 6.5.2
403	Forbidden	Section 6.5.3
404	Not Found	Section 6.5.4
405	Method Not Allowed	Section 6.5.5
406	Not Acceptable	Section 6.5.6
407	Proxy Authentication Required	Section 3.2 of [RFC7235]
408	Request Timeout	Section 6.5.7
409	Conflict	Section 6.5.8
410	Gone	Section 6.5.9
411	Length Required	Section 6.5.10
412	Precondition Failed	Section 4.2 of [RFC7232]
413	Payload Too Large	Section 6.5.11
414	URI Too Long	Section 6.5.12
415	Unsupported Media Type	Section 6.5.13
416	Range Not Satisfiable	Section 4.4 of [RFC7233]
417	Expectation Failed	Section 6.5.14
426	Upgrade Required	Section 6.5.15
500	Internal Server Error	Section 6.6.1
501	Not Implemented	Section 6.6.2
502	Bad Gateway	Section 6.6.3
503	Service Unavailable	Section 6.6.4
504	Gateway Timeout	Section 6.6.5
505	HTTP Version Not Supported	Section 6.6.6

3 Rajapinnat ja Oauth

Ohjelmointirajapinta eli API (Application Programming Interface) on käyttöliittymä sovellusten tai tietokoneiden välille, jotta ne voisivat ymmärtää toisiaan (Berlind, 2015a).

Ohjelmointirajapinnan tehtävänä on mahdollistaa kommunikointi palvelujen ja ohjelmien välillä dokumentoidun rajapinnan kautta, ilman vaatimusta tietää yksityiskohtia rajapinnan takana olevan palvelun toteutuksesta. (IBM, 2020)

ProgrammableWeb-artikkelissaan David Berlind vertaa ohjelmointirajapintaa pistorasiaan. Kodinkoneet kiinnitetään sähköverkkoon pistorasian kautta olettaen, että sitä kautta saadaan tietyt vaatimukset täyttävää energiaa, välittämättä miten energia on tuotettu. (Berlind, 2015b)

Suosituksen rajapintojen kehitystyökalun, Postmanin blogisivulla Kin Lane käyttää seuraavaa metaforaa kuvaamaan rajapintoja. Ravintolassa on asiakas, tarjoilija sekä keittiö. Asiakas puhuu tarjoilijan kanssa ja esittää hänelle mitä hän haluaa ruoaksi. Tarjoilija kirjaa asiakkaan toiveet ylös ja välittää ne keittiölle. Keittiö valmistaa tilatun ruoan ja lopulta tarjoilija tuo valmiin aterian asiakkaalle. Tässä metaforassa asiakas on ohjelmisto tai käyttäjä, joka tekee pyynnön. Tarjoilija (rajapinta) välittää pyynnön keittiöön (palvelin). Lopuksi palvelin palauttaa pyydetyn tiedon siinä muodossa, jonka ohjelmistorajapinnan spesifikaatio määrittelee. (Lane, 2020)

Rajapinnat voidaan luokitella neljään kategoriaan (Taulukko 2), avoimen datan rajapintoihin, julkisiin ja kumppanirajapintoihin sekä sisäisiin rajapintoihin. Avoimen datan rajapinnat ovat rajapintoja, joita voi käyttää vapaasti kuka tahansa ja niiden tarjoama data on myös avointa ja käytettävissä siihen liitetyn lisenssin mukaisesti. Julkiset rajapinnat ovat hyvin samankaltaisia avoimen datan rajapintojen kanssa, mutta sisältävät mahdollisesti eri hintatasoja ilmaisesta lähtien, joissa eri tasoilla on eri oikeuksia tai rajoituksia. Julkisissa rajapinnoissa myöskään data ei ole aina avointa. Kumppani rajapinnat ovat suljettuja rajapintoja, jotka eivät ole julkisesti saatavilla eikä niiden tarjoama data ole avointa. Kumppani rajapinnat ovat usein asiakaskohtaisesti tuotettuja rajapintoja. Sisäiset rajapinnat ovat useasti yrityksen tai organisaation sisäverkkoon luotuja rajapintoja. (Moilanen, 2018, ss. 57-59)

Taulukko 2. API-luokittelu (Moilanen, 2018, s. 58).

		Julkisesti saatavilla ⁷⁹	Käyttöönoton tuki julkista ⁸⁰	Käyttö maksaa	Data avointa
Avoimen datan API (Open Data Interface)		Kyllä	Kyllä	Yleensä aluksi ilmaista, mutta vaatimusten kasvaessa saattaa olla maksullista.	Kyllä, sisältöön liitetty lisenssi määrittää oikeudet.
Avoin API (Open API)	Julkinen API (Public API)	Kyllä	Kyllä	Useasti kyllä. Voi sisältää myös ilmaiskerroksen (freetier) muodossa tai toisessa.	Ehkä
	Kumppani API (Partner API)	Ei	Kyllä	Ei, "kuuluu usein pakettiin".	Ei
Sisäiset API:t (private API eli julkisessa verkossa oleva sisäinen käyttöön tai internal API sisäverkossa sisäiseen käyttöön)		Ei	Ei	Ei	Ei

3.1 RESTful API

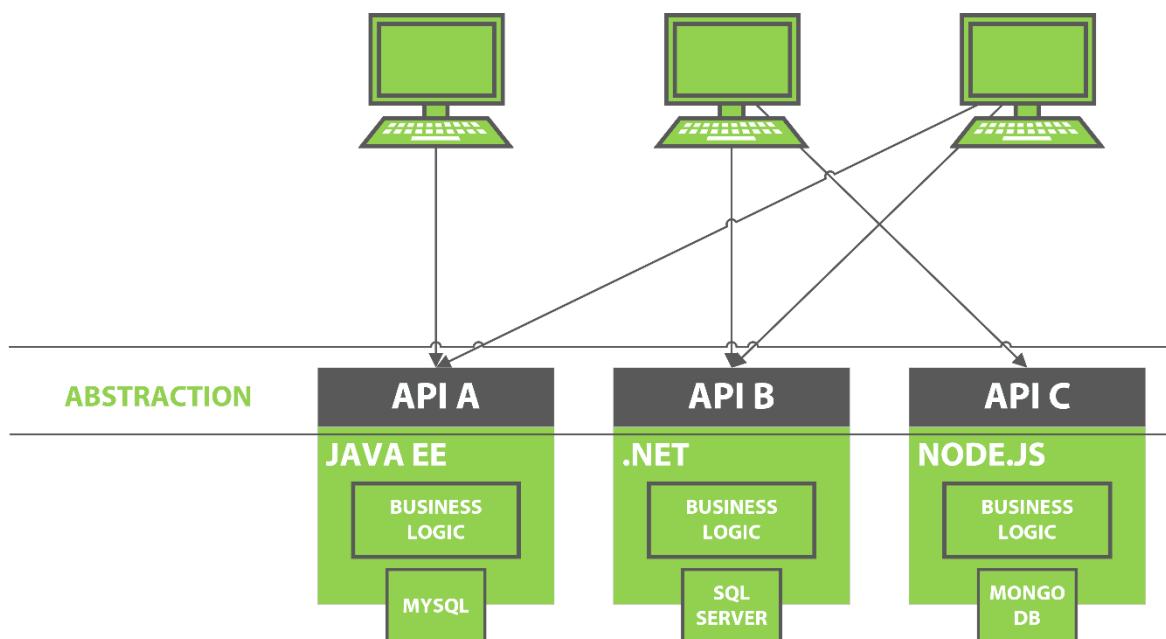
Termi REST tulee sanoista REpresentational State Transfer, joka konseptina tarkoittaa arkkitehtuurityyliä hajautettuihin hypermediajärjestelmiin, termin REST on alun perin esitellyt Roy Thomas Fielding väitöskirjassaan "Architectural Styles and the Design of Network-based Software Architecture".

Väitöskirjassaan Fielding (Fielding, 2000, ss. 46, 76–85) määrittelee arkkitehtuurille kuusi ohjaavaa periaatetta, jotka määrittelevät onko rajapinta RESTful:

1. Client–Server (Asiakas–Palvelin). Erottamalla asiakasohjelman (käyttöliittymän) vastuut palvelimen (tietokannan) vastuista, käyttöliittymän "kannettavuus" paranee ja mahdollistaa tämän käytön useammalla alustalla. Myös palvelimen osat (komponentit) yksinkertaistuvat, jolloin palvelimen skaalautuvuus paranee ja komponentit pääsevät kehittymään itsenäisesti, eivätkä ole riippuvaisia toisistaan.
2. Stateless (Tilaton). Jokainen asiakasohjelman lähettämä pyyntö tulee sisältää kaikki tarpeellinen tieto liittyen pyyntöön, jonka palvelin tarvitsee ymmärtääkseen vastaanotetun pyynnön. Näin ollen istunnon (session) tila pysyy vain asiakasohjelman tiedossa.
3. Cache (Välimuisti). Palvelimen lähettämä vastaus tulee sisältää tieto siitä, onko vastaus varastoitavissa välimuistiin vai ei. Jos vastaus on varastoitavissa, asiakasohjelman tehdessä samankaltaisia pyyntöjä, voi se käyttää välimuistissa säilytettyä versiota palvelimen vastauksesta.

4. Uniform Interface (Yhtenäinen Käyttöliittymä). Määrittelee yhtenäisen tavan keskustella palvelimen kanssa ja luo abstraktion toteutuksen ja käyttöliittymän välillä. Näin ollen, asiakasohjelma kykenee keskustelemaan kaikkien järjestelmän palveluiden kanssa yhtenäisten käyttöliittymien kautta (Kuva 3) (Hurbans & Van Der Westhuizen, n.d.).
5. Layered System (Kerroksinen Järjestelmä). Järjestelmä koostuu hierarkkisista kerroksista, joissa komponentit pystyvät kommunikoimaan vain hierarkiassa suoraan ylä- tai alapuolella olevien komponenttien kanssa. Kerroksen komponentti tarjoaa palveluita vain hierarkiassa suoraan yläpuolella oleville komponenteille ja käyttää vain hierarkiassa suoraan alapuolella olevien komponenttien palveluita. Esimerkiksi asiakasohjelma ei voi tietää onko tämä yhteydessä suoraan päätepalvelimeen vai kuormantasaukseen tarkoitettuun välipalvelimeen (Fredrich, n.d.).
6. Code-On-Demand. Mahdollistaa asiakasohjelman toimintojen laajentamisen lataamalla ja ajamalla ohjelmakoodia sovellusten tai skriptien avulla. Tämä periaate/rajoitus on vapaaehtoinen.

Kuva 3. Asiakasohjelmat keskustelevat eri palveluiden kanssa yhtenäisen rajapinnan kautta. (Hurbans & Van Der Westhuizen, n.d.)



3.2 OAuth

OAuth on protokolla, jonka avulla voi valtuuttaa ohjelmiston käsittelemään tai muokkaamaan käyttäjän hallinnoimaa resurssia palvelussa ilman ohjelmiston tarvetta esittää käyttäjää. Protokolla alun perin luotiin pienen webkehittäjäyhteisön toimesta ratkaisemaan yleinen ongelma, jossa täytyi antaa valtuudet johonkin suojattuun resurssiin (IETF, 2010b).

Esimerkki OAuth:in toiminnasta on ohjelma, jonka avulla voi tulostaa valokuvia suoraan pilvipalvelusta. Käyttämällä OAuth-protokollaa, ohjelma voi todentaa tällä olevan oikeus käyttää kyseisen käyttäjän resursseja (valokuvia) pilvipalvelussa, ilman käyttäjän tarvetta luovuttaa pilvipalvelun tunnuksia ohjelmalle. OAuth mahdollistaa myös käyttöoikeuksien hienosäätämisen, esimerkiksi vain lukuoikeudet. (Richer & Antonio, 2017, ss. 3-6)

Ennen OAuth:in kehitystä, näiden valtuuksien ja käyttöoikeuksien antaminen ohjelmistoille tapahtui useasti hyvin tietoturvattomasti jakamalla resurssin omistajan (käyttäjän) tunnukset tai muun pääsy tiedon asiakasohjelmalle (Client). Asiakasohjelma käytti tunnuksia esittääkseen käyttäjää ja päästäkseen käsiksi käyttäjän omistamaan suojattuun resurssiin (Esim. pilvitalennustila).

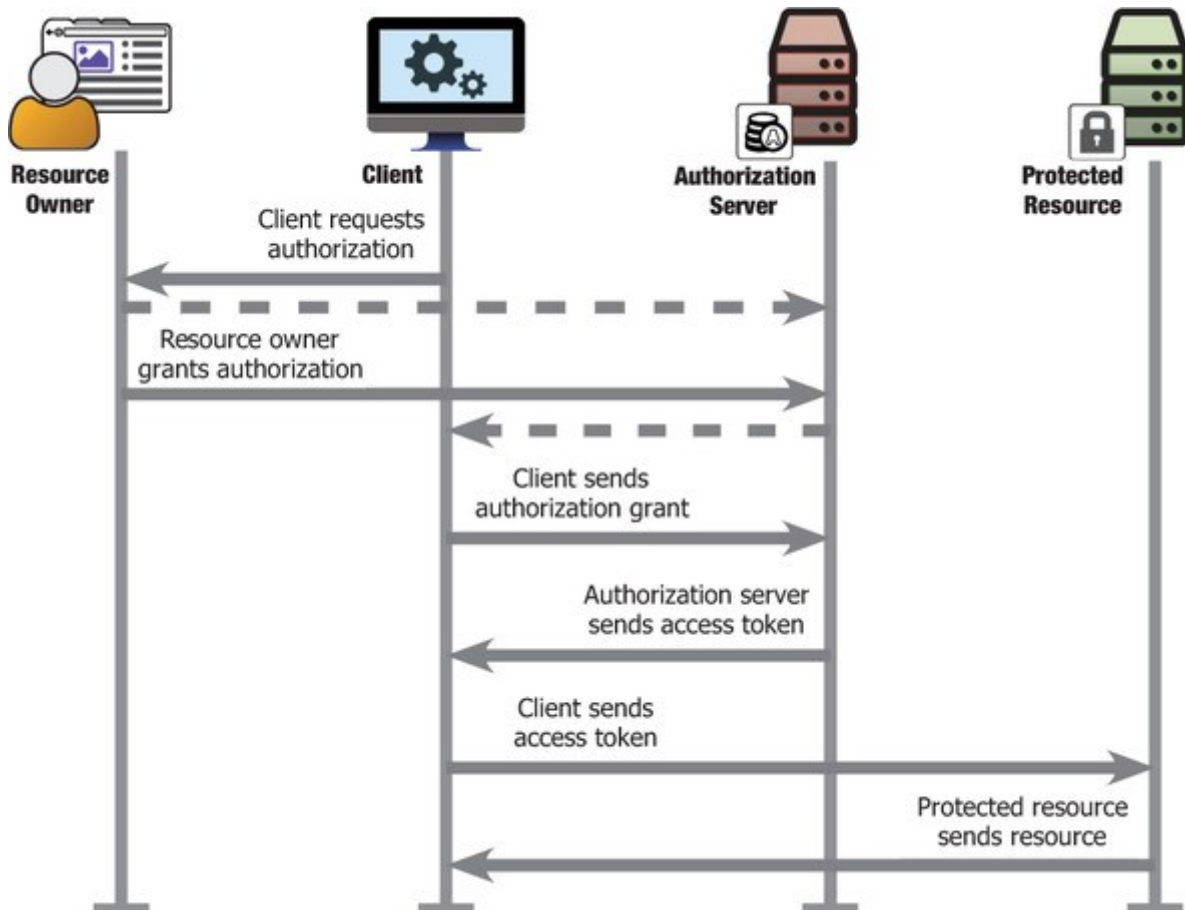
Näitä tapoja jakaa käyttäjän tunnukset olivat muun muassa tunnusten kopioiminen tai pyytäminen käyttäjältä ja niiden esittäminen resurssille. Kumpikin näistä tavoista altistaa käyttäjän tunnukset mahdollisesti epäluotettavalle sovellukselle, asiakasohjelmalle. Kummassakin tavassa myös esitetään käyttäjää, eikä käyttöoikeuksia voi hienosäätää millään tavalla, jolloin tunnukset haltuunsa saanut asiakasohjelma kykenee käyttämään resurssia kuin haluaa, täysin käyttäjän tietämättä. (Richer & Antonio, 2017, ss. 7-8)

Yksi valtuutustavoista on, käyttäjän tunnusten sijaan, antaa asiakasohjelmalle niin kutsuttu yleisavain, jota se käyttää kaikissa kutsuissa suojatulle resurssille, riippumatta siitä kuka asiakasohjelmaa käyttää. Tämä tapa estää käyttäjän tunnusten paljastamista asiakasohjelmalle, mutta se antaa ohjelmalle hyvin korkean tason valtuudet esittää ketä tahansa käyttäjää. Tämä mahdollistaisi asiakasohjelman käyttämään kaikkien käyttäjien resursseja niillä valtuuksilla, jotka kyseiselle yleisavaimelle on suotu. (Richer & Antonio, 2017, ss. 9-10)

Näiden lisäksi on vielä yksi tapa, jossa käyttäjällä on erikoissalasana. Tätä erikoissalasanaa käyttäjä käyttää vain kyseisen asiakasohjelman tai kolmannen osapuolen asiakasohjelmien kanssa. Näin käyttäjä ei luovuta päätunnuksiaan eikä asiakasohjelma saa niin sanotusti vapaita käsiä suojattuun resurssiin. Tämän tavan ongelma kuitenkin on käyttäjän tarve hallinnoida päätunnuksien lisäksi erikoissalasanoja. Tämän lisäksi asiakasohjelman ja valtuuksien välillä ei ole mitään yhdistävää tekijää, joten oikeuksien poistaminen tietyltä asiakasohjelmalta on hankalaa. (Richer & Antonio, 2017, s. 10)

OAuth:issa mukaan tulee valtuutuspalvelin (Authorization Server), joka jakaa tokeneita (OAuth Access Token) asiakasohjelmille (Kuva 4). Tokenin saadakseen asiakasohjelma ohjaa käyttäjän valtuutuspalvelimelle, jossa käyttäjä valtuuttaa asiakasohjelmalle tietyt oikeudet (Esim. luku- tai kirjoitusoikeus) resurssiin. Kun käyttäjä on antanut asiakasohjelmalle valtuudet, asiakasohjelma pyytää valtuutuspalvelimelta tokenin, jota se käyttää päästäkseen käsiksi suojattuun resurssiin. (Richer & Antonio, 2017, ss. 11-13)

Kuva 4. Korkean tason kuvaus OAuth:in toiminnasta (Richer & Antonio, 2017, s. 12).



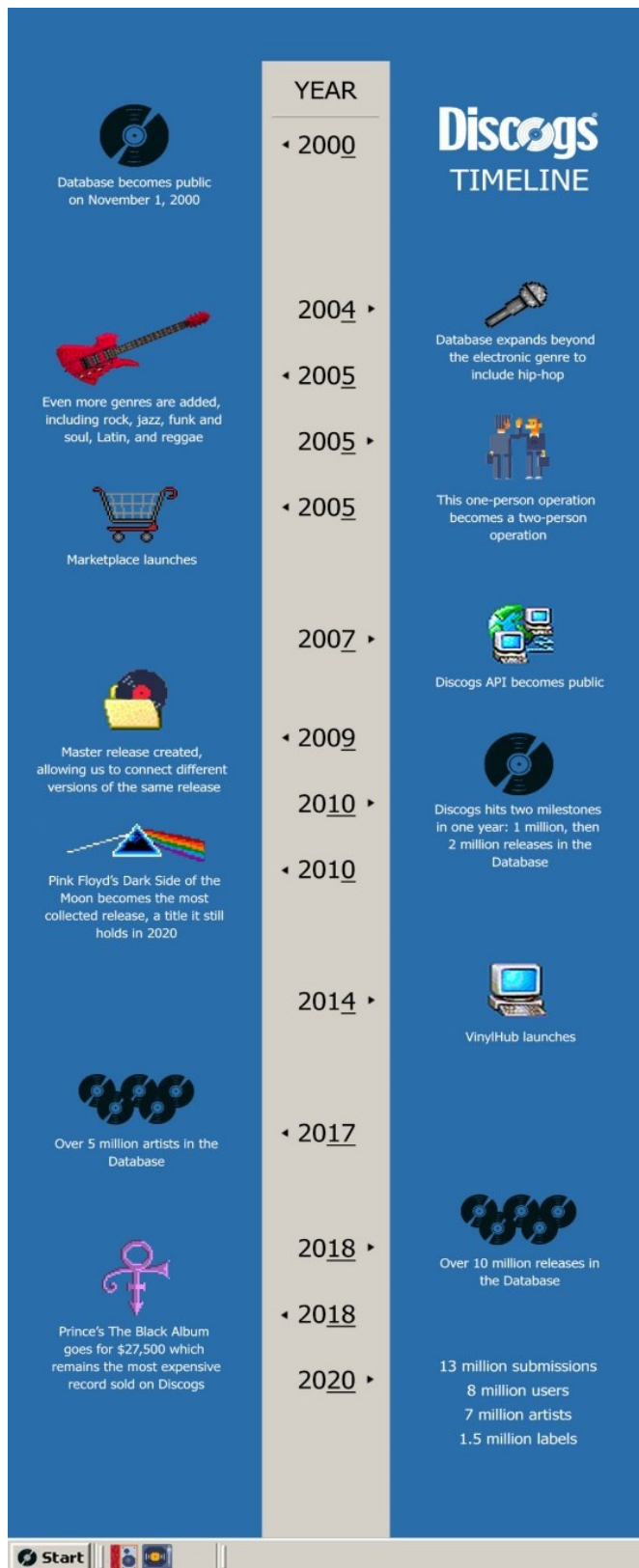
4 Discogs

Discogs on maailmanlaajuinen joukkoistettu tietokanta, joka sisältää tietoa musiikkijulkaisuista ympäri maailmaa. Tietokanta sisältää virallisten julkaisujen lisäksi kaikkea epävirallisista julkaisuista haastatteluihin. Discogs toimii myös markkinapaikkana, jossa voi myydä ja ostaa musiikkia. Tietokannan ylläpidosta eli julkaisujen muokkaamisesta ja uusien julkaisujen lisäämisestä huolehtii yhteisö. (Saunders, 2020)

Discogs:in perustaja ja toimitusjohtaja Kevin Lewandowski julkaisi Discogs.com -sivuston marraskuun 1. päivä vuonna 2000. Lewandowskin mukaan tietokanta lähti liikkeelle harrastuksena, eikä hän osannut odottaa sen tulevan kantautumaan näinkin pitkälle. Tietokannan tarkoituksena oli alun perin sisältää vain elektronista musiikkia, mutta yhteisön palautteen johdosta tietokantaan alettiin hyväksymään myös hip-hoppiä, rokkia, jazzia ja lopulta kaikkea muutakin. (Discogs, 2020)

Discogs:in 20 vuoden historiaan on mahtunut monenmoista. Kuvasta 5 näkee Discogs:in henkilökunnan valitsemat kohokohdat, kuten markkinapaikan avauksen vuonna 2005 ja rajapinnan ensimmäisen version julkaisu kaksi vuotta myöhemmin. Vuonna 2018, Discogsin kallein myyty julkaisu oli Princen ”The Black Album” 27 500 dollarilla (Discogs, 2020). Kunes vuoden 2020 joulukuussa Scaramanga Silk:in albumi ”Choose Your Weapon” myytiin hintaan 41,095.891 dollaria (Raney, 2021).

Kuva 5. Discogs:in historian kohokohdat (Discogs, 2020).



4.1 Rajapinnan tarjoamat markkinapaikan ominaisuudet

Discogs-palvelun tarjoama avoin rajapinta on REST pohjainen rajapinta, joka tarjoaa rajapinnan dokumentoinnin mukaan (Discogs.com, n.d.-b) kattavan valikoiman erilaisia ominaisuuksia, kuten

- hakujen tekeminen tietokantaan hakusanoin,
- artistien, levy-yhtiöiden sekä julkaisujen tietojen noutaminen,
- käyttäjien julkisesti saatavilla olevien tietojen noutaminen,
- autentikoidun käyttäjän tietojen noutaminen ja muokkaaminen,
- myynti-ilmoituksen luominen ja sen tietojen noutaminen sekä muokkaaminen,
- tilauksen tietojen noutaminen ja muokkaaminen myyjäksi autentikoituna.

Markkinapaikan pääominaisuuksia ovat myynti-ilmoitusten ja tilausten noutaminen sekä muokkaus ja uusien myynti-ilmoitusten luominen. Näiden lisäksi rajapinnan kautta voi ladata ja lähettää inventaarion sisältöä kokonaisuudessaan käyttäen CSV-tiedostoja, noutaa tiedon myynnin yhteydessä syntyvistä kuluista myyjälle, hakea suositushinnan myyntikohteelle sekä noutaa myyntitilastot tietyille julkaisulle. (Discogs.com, n.d.-c)

Uuden myynti-ilmoituksen (Listing) luominen tapahtuu lähettämällä POST-pyyntö rajapintaan. Pyyntönsä otsakkeiden tulee sisältää käyttäjän autentikointiin tarvittavat tiedot (Authorization), asiakasohjelman nimen (User-Agent) ja sisältötyypin (Content-Type) ”application/json”. Näiden lisäksi pyynnön runkona (Body) tulee olla luotavan myynti-ilmoituksen tiedot JSON muodossa. Uusi myynti-ilmoitus lisätään autentikoidun käyttäjän inventaarioon (Inventory), joka sisältää kaikki käyttäjän omat myynti-ilmoitukset. Discogs-rajapinnan dokumentointi määrittelee (Discogs.com, n.d.-c) uuden myynti-ilmoituksen tiedoiksi

1. release_id (Pakollinen), tunnusnumero, joka viittaa kyseisen julkaisun Discogs-sivuun,
2. condition (Pakollinen), kuntoluokitus medialle (Taulukko 3),
3. sleeve_condition, kuntoluokitus median kansille (Taulukko 3),
4. price (Pakollinen), myytävän tuotteen hinta myyjän asettamassa valuutassa,

5. comments, myynti-ilmoituksen ohessa näytettävä viesti (esim. lisähuomio tuotteesta,
6. allow_offers, voivatko käyttäjät tehdä ostotarjouksia myynti-ilmoituksesta (vaihtoehtoina "true" ja "false"),
7. status, mihin tilaan myynti-ilmoitus asetetaan (vaihtoehtoina "Draft" ja "For Sale"),
8. external_id, vapaamuotoinen yksityinen kenttä myyjän käyttöön (esim. myytävän tuotteen tunnusnumero myyjän omassa järjestelmässä),
9. location, vapaamuotoinen yksityinen kenttä myyjän käyttöön (esim. tuotteen sijainti myyjän omassa varastossa),
10. weight, myytävän tuotteen paino grammoina (arvolla "auto", paino arvioidaan automaattisesti),
11. format_quantity, kuinka monena tuotteena tuote lasketaan toimituskustannuksia laskettaessa (arvolla "auto", määrä arvioidaan automaattisesti).

Taulukko 3. Median kuntoluokitukset (Discogs.com, n.d.-c).

Mint (M)	
Near Mint (NM or M-)	
Very Good Plus (VG+)	
Very Good (VG)	
Good Plus (G+)	
Good (G)	















Fair (F)	
Poor (P)	
Generic	Vain sleeve_condition
Not Graded	Vain sleeve_condition
No Cover	Vain sleeve_condition

4.2 Autentikointi

Discogs-rajapintaa voi käyttää vain rajoitetusti ilman autentikointia (Taulukko 4), jolloin myös pyyntöjä rajapintaan voi suorittaa vain 25 kappaletta minuutissa (Low tier).

Autentikointitapoja on kolme, kaikissa pyyntöjen määrä nousee 60:een pyyntöä minuutissa (High tier). Tämän lisäksi kaikilla kolmella tavalla on mahdollista saada pyyntöjen mukana kuvien osoitteet. Autentikointitavoista vain kaksi mahdollistaa käyttäjäksi autentikoinnin. (Discogs.com, n.d.-a)

Taulukko 4. Discogs-palvelun autentikointitavat (Discogs.com, n.d.-a).

Credentials in request	Rate limiting?	Image URLs?	Authenticated as user?
None	 Low tier	 No	 No
Only Consumer key/secret	 High tier	 Yes	 No
Full OAuth 1.0a with access token/secret (see below)	 High tier	 Yes	 Yes, on behalf of any user 
Personal access token	 High tier	 Yes	 Yes, for token holder only 

Osa Discogs-rajapinnan ominaisuuksista on käytettävissä vain käyttäjäksi autentikoituna, kuten kaikki ominaisuudet, jotka muokkaavat jotain resurssia. Autentikointia tarvitsee myös kaikki käyttäjän yksityisiin tietoihin liittyvät ominaisuudet, kuten nimen tai nimimerkin

muokkaaminen. Jos käyttäjä on asettanut joitain tietoja yksityiseksi, kuten toivelistan, tämän noutaminen vaatii myös autentikoinnin kyseisen käyttäjän tunnuksilla. (Discogs.com, n.d.-b)

Käyttäessä avain–salaisuus paria tai henkilökohtaista tokenia autentikointiin, voidaan nämä lähettää jokaisen kutsun mukana joko kyselyosoitteen (query) perässä muodossa **"&key=avain&secret=salaisuus"** tai **"&token=tokeni"** tai "Authorization" otsakkeessa muodossa **"Discogs key=foo123, secret=bar456"** tai **" Discogs token=tokeni"**. (Discogs.com, n.d.-a)

Discogs-rajapinnan dokumentoinnin mukaan (Discogs.com, n.d.-a) autentikointi käyttäen OAuth 1.0a -protokollaa tapahtuu viidessä vaiheessa

1. Kuluttaja avaimen ja salaisuuden (Consumer key ja Consumer secret) hakeminen Discogs-palvelun käyttäjäasetuksista, "developers"-välilehden alta.
2. GET-pyyntö lähettäminen rajapinnan resurssiin **"/oauth/request_token"**. Pyyntö otsakkeissa on määritelty "Content-Type", "Authorization" sekä "User-Agent" (Kuva 6). Authorization-otsakkeen kohtiin **oauth_consumer_key** ja **oauth_signature**, syötetään ensimmäisessä vaiheessa haetut avain ja salaisuus. Onnistuneen kutsun vastaus sisältää tokenin (**oauth_token**) ja tokeni salaisuuden (**oauth_token_secret**).
3. Käyttäjän ohjaaminen valtuutusivulle (Authorization page), joka on osoitteessa **"https://discogs.com/oauth/authorize?oauth_token=<oauth_request_token>"**, jossa **oauth_request_token**:in tilalle syötetään edellisessä vaiheessa saatu **oauth_token**. Sivulla käyttäjä voi valtuuttaa käyttämänsä asiakasohjelman ja saa tästä vahvistusavaimen (**verifier_key**). Jos asiakasohjelman kehittäjä on asettanut ohjelmalle Callback URL:n, hyväksymisen jälkeen tapahtuu uudelleenohjaus kyseiseen osoitteeseen, jonka vastauksesta voi poimia vahvistusavaimen.
4. POST-pyyntö lähettäminen rajapinnan resurssiin **"/oauth/access_token"**, jonka otsakkeissa on määritelty "Content-Type", "Authorization" sekä "User-Agent" (Kuva 7). Otsakkeet ovat lähestulkoon samat kuin kohdassa 2 määritellyt, poikkeuksena uusi kenttä **oauth_token**, jonka arvoksi tulee kohdan kaksi vastauksessa saatu **oauth_token**, sekä **oauth_callback**:in tilalla on

oauth_verifier. Jälkimmäisen arvoksi tulee edellisessä vaiheessa saatu vahvistusavain (**verifier_key**). Onnistuneen pyynnön vastaus sisältää tokenin sekä salaisuuden, joita asiakasohjelma voi käyttää päästäkseen käsiksi käyttäjän omistamiin resursseihin (esim. myynti-ilmoitukset).

5. Jatkossa tehtävien pyyntöjen otsakkeet tulevat sisältää edellisessä vaiheessa saadun tokenin sekä salaisuuden.

Kuva 6. OAuth request_token pyyntö.

```
GET https://api.discogs.com/oauth/request_token

Content-Type: application/x-www-form-urlencoded
Authorization:
    OAuth oauth_consumer_key="your_consumer_key",
    oauth_nonce="random_string_or_timestamp",
    oauth_signature="your_consumer_secret&",
    oauth_signature_method="PLAINTEXT",
    oauth_timestamp="current_timestamp",
    oauth_callback="your_callback"
User-Agent: some_user_agent
```

Kuva 7. OAuth access_token pyyntö.

```
POST https://api.discogs.com/oauth/access_token

Content-Type: application/x-www-form-urlencoded
Authorization:
    OAuth oauth_consumer_key="your_consumer_key",
    oauth_nonce="random_string_or_timestamp",
    oauth_token="oauth_token_received_from_step_2"
    oauth_signature="your_consumer_secret&",
    oauth_signature_method="PLAINTEXT",
    oauth_timestamp="current_timestamp",
    oauth_verifier="users_verifier"
User-Agent: some_user_agent
```

4.3 Vaihtoehdot/Kilpailijat

Koska Discogs-palvelu tarjoaa kaikille avoimen markkinapaikan, hyvin kattavan tietokannan musiikkijulkaisuista sekä avoimen rajapinnan, jonka kautta pystyy käyttämään lähes kaikkia palvelun ominaisuuksia, on suoran kilpailijan tai vaihtoehdon löytäminen haastavaa.

Yksittäisillä osa-alueilla kuitenkin löytyy kilpailijoita.

Fyysisten musiikkijulkaisujen markkinapaikkoja ovat esimerkiksi Amazon, Ebay, Vinylom sekä cdandlp. Näistä neljästä Ebay:llä sekä cdandlp:llä on olemassa rajapinta, jonka kautta voi käyttää ainakin osaa palveluiden ominaisuuksista.

Kattavia musiikkijulkaisujen tietokantoja, jotka tarjoavat myös avoimen rajapinnan, on olemassa esimerkiksi Spotify ja Musicbrainz, myös edellä mainittu Vinylom tarjoaa hyvin rajallisen tietokannan vinyyljulkaisuista ja niiden historiasta, mutta ei rajapintaa niiden noutamiseen.

Tässä työssä keskitytään tietokantojen osalta vaihtoehtoihin Spotify sekä Musicbrainz ja markkinapaikkojen osalta vaihtoehtoihin Ebay sekä cdandlp. Nämä vaihtoehdot valittiin koska ne tarjoavat avoimen rajapinnan kattavilla ominaisuuksilla.

4.3.1 Spotify

Spotify on suosittu musiikin suoratoistopalvelu, jota käyttää noin 345 miljoonaa käyttäjää, joista noin 155 miljoonaa on maksavia tilaajia. Spotify sisältää yli 70 miljoonaa kappaletta sekä yli 2.2 miljoonaa podcastia. (Spotify, n.d.-a)

Spotify tarjoaa myös avoimen REST rajapinnan kappaleiden, albumien sekä artistien tietojen hakemiseen. Rajapinnan kautta pääsee käsiksi myös kappaleiden analyysidataan, joka sisältää muun muassa kappaleen tunnelman, tahdin, sävellajin sekä sen kuinka hyvin kappaleen tahdissa voi tanssia. (Spotify, n.d.-c)

Spotifyn rajapinnan käyttö vaatii käyttäjätunnukset ja pyynnöissä, jotka hakevat tai muokkaavat yksityistä dataa, tarvitaan myös OAuth autentikointi, kuten käyttäjäprofiilin tai soittolistan noutamisessa. Spotify rajoittaa rajapintaan tehtyjen pyyntöjen määrää

asiakasohjelman tunnuksen (Client ID) perusteella, riippumatta kuinka monta käyttäjää kyseistä asiakasohjelmaa käyttää samanaikaisesti. Pyyntöjen maksimimäärää ei ole erikseen määritelty, vaan rajapinta palauttaa HTTP vastauksessa tilakoodin 429 ja sisältää otsakkeen "Retry-After", jonka arvo viittaa siihen, kuinka monta sekuntia asiakasohjelman tulee odottaa ennen seuraavan pyynnön lähettämistä (Spotify, n.d.-b)

Spotifyn tarjoamaa metadataa julkaisuista ei voida vaihtaa suoraan Discogsin tarjoaman metadatan kanssa, sillä Discogs-markkinapaikan myynti-ilmoitukset käyttävät hyödykseen Discogsin omassa tietokannassa olevaa metadataa, eikä tätä pysty korvaamaan Discogsin ulkopuolisella datalla. Spotifyn käyttäminen vaihtoehtona edellyttäisi siis toteutuksessa käytettävän jotain korvaavaa markkinapaikkaa, Discogs-markkinapaikan sijaan.

4.3.2 MusicBrainz

MusicBrainz on vuonna 2000, Robert Kaye:n toimesta julkaistu tietokanta, joka sisältää tietoa muun muassa artisteista, albumeista sekä kappaleista. MusicBrainz on Discogs:in tavoin joukkoistettu, yhteisön ylläpitämä tietokanta, jonka sisällön lisäämiseen ja muokkaamiseen kuka tahansa voi osallistua. (MetaBrainz Foundation, n.d.-b)

MusicBrainz sisältää metatietoa yli 1.8 miljoonasta artistista, yli 2.8 miljoonasta julkaisusta (albumi, single, jne.) sekä yli 33 miljoonasta kappaleesta. (MetaBrainz Foundation, n.d.-c) MusicBrainz tarjoaa avoimen REST rajapinnan näiden metatietojen noutamiseen. Rajapinnan kautta voi myös lisätä tietoja, mutta hyvin rajoitetusti, kuten esimerkiksi käyttäjäarvosanan, julkaisun viivakoodin sekä tunnisteiden tai tyylilajin. (MetaBrainz Foundation, n.d.-a)

Rajapinnan peruskäyttö vaatii ainoastaan "User-Agent" otsakkeen pyyntöjen mukana, jonka avulla MusicBrainz-palvelun ylläpitäjät voivat ongelmatilanteissa ottaa sovelluksen ylläpitäjään yhteyttä, esimerkiksi "Sovelluksen nimi/<versio> {sähköpostiosoite}". Rajapintapyynnot, jotka käsittelevät käyttäjätietoja tai metadatan lisäystä tietokantaa, tarvitsevat myös Digest autentikoinnin. (MetaBrainz Foundation, n.d.-a)

Rajapintapyyntöjen määrä on rajoitettu yhteen pyyntöön sekunnissa per IP-osoite ja jos pyyntöjen otsakkeissa määritelty "User-Agent" ei ole edellä mainitussa formaatissa, rajoitetaan pyyntöjen määrä 50 pyyntöön sekunnissa koko sovellukselta, riippumatta kuinka

monta käyttäjää kyseisellä sovelluksella on. Tämän lisäksi, jos MusicBrainz-palvelun palvelimien kuorma on liian suuri, pyynnöt hylätään. Jos jokin edellä mainituista rajoituksista ylittyy tai täyttyy, pyynnöt hylätään. Rajapinnan käyttö kaupallisessa käytössä vaatii erillisen sopimuksen MetaBrainz Foundationin kanssa. (MetaBrainz Foundation, n.d.-a)

4.3.3 Ebay

Ebay (alun perin AuctionWeb) on vuonna 1995, Pierre Omidyarin toimesta perustettu maailmanlaajuinen markkinapaikka uuden ja käytetyn tavaran ostamiseen ja myymiseen. Ebay-markkinapaikka sisältää 1.7 miljardia myynti-ilmoitusta erilaisista tuotteista ympäri maailmaa. (eBay Inc., n.d.-b) Viimeisen vuoden aikana markkinapaikalla on toiminut 20 miljoonaa myyjää, ostajia puolestaan on ollut 187 miljoonaa (eBay Inc., 2021).

Ebay tarjoaa kattavan valikoiman erilaisia rajapintoja eri tarkoituksiin, esimerkiksi myyntirajapinta tuotteiden myymiseen, ostorajapinta ostamiseen, kaupankäyntirajapinta, jota on tarkoitus käyttää osto- ja myyntirajapintojen kanssa. (eBay Inc., n.d.-a)

4.3.4 CDandLP

CDandLP on vuonna 2001 perustettu CD- ja Vinyylilevyjen markkinapaikka, jossa käyttäjät voivat myydä uusia kuin käytettyjäkin levyjä. CDandLP markkinapaikka toimii välittäjänä myyjien ja ostajien välillä. (CDandLP, n.d.-a)

CDandLP tarjoaa suljetun rajapinnan, jonka käyttäminen vaatii käyttäjätunnuksen palvelussa sekä erillisen luvan palvelun ylläpitäjiltä. Luvan saamiseksi, tulee CDandLP:n ylläpitäjille lähettää rajapintaa käyttävän palvelimen URL tai IP-osoite. (CDandLP, n.d.-b)

5 Toteutus

Ennen opinnäytetyötä asiakas oli myynyt kokeiluna Discogs-markkinapaikan kautta useita tuotteita, pääosin uusia julkaisuja. Myynti-ilmoitukset tehtiin manuaalisesti ja myydyistä tuotteista tehtiin asiakkaan omassa järjestelmässä uusi tilaus. Tilaus tehtiin tuotteen ostaneen Discogs-käyttäjän tiedoilla, jotta tilaus käsiteltäisiin oikein ja samalla tavalla kuin asiakkaan oman verkkokaupan kautta tehdyt tilaukset.

Integroinnin kohdejärjestelmä oli asiakkaan PHP 5.4 pohjainen tilausten ja inventaarion hallinnointijärjestelmä. Järjestelmästä ei löytynyt ennestään mitään oleellista tietoa integroinnin osalta, kuten median tai kansien kuntoluokituksia tai tiettyyn musiikkijulkaisuun viittaavaa Discogs-tunnusnumeroa.

Integroinnissa Discogs-palvelun rajapinnan kanssa kommunikointiin oli vaihtoehtoina kaksi kolmannen osapuolen kirjastoa, Python- ja PHP-kirjastot. Vaihtoehtoista Python-kirjasto oli ajantasaisempi verrattuna PHP-kirjastoon ja näin ollen mahdollisesti tietoturvasempi, mutta PHP-kirjasto oli kirjoitettu samalla ohjelmointikielellä ja versiolla kuin kohdejärjestelmäkin.

5.1 Suunnitelma

Toteutuksen suunnitteluvaihe lähti liikkeelle pohdinnalla siitä mitä tietoa asiakkaan järjestelmässä tulisi olla, jotta uusien myynti-ilmoitusten luominen sekä vanhojen poistaminen onnistuisi mahdollisimman vaivattomasti. Toimeksiantajan kanssa keskusteltaessa päädyttiin tulokseen, jossa kohdejärjestelmässä olevilla tuotteilla tulisi olla ainakin Discogs-tunnusnumero sekä kuntoluokitukset. Tunnusnumero viittaa kyseessä olevaa tuotetta vastaavan musiikkijulkaisun Discogs-sivuun. Discogs-tunnusnumero ja kuntoluokitukset ovat ainoat puuttuvat tiedot mitä tarvitaan uuden myynti-ilmoituksen tekemiseen Discogs-markkinapaikalla.

Discogs-tunnusnumeron lisääminen tulisi tehdä mitä todennäköisimmin manuaalisesti, sillä Discogs-palvelu sisältää useasti monta eri versiota yhdestä julkaisusta, esimerkiksi eriväriset vinylilevyt, joilla jokaisella on eri tunnus. Tämä vaikeuttaa huomattavasti tunnusnumeron

hakemista automatisoidusti ja vaatisi todennäköisesti manuaalista hakemista. Manuaalista hakemista auttamaan, osaksi integrointia voisi tehdä hakukoneen niiden Discogs-julkaisujen hakemiseen, jotka täsmäävät mahdollisimman hyvin järjestelmän tuotteen kanssa. Tämä helpottaisi huomattavasti oikean julkaisun valitsemista.

Kohdejärjestelmän tuotteiden kuntoluokitusten määrittely voitaisiin tehdä ainakin osittain automatisoidusti. Uusille ja käyttämättömille tuotteille voitaisiin asettaa kuntoluokitukseksi automaattisesti "Mint (M)" eli paras mahdollinen. Käytettyjen tuotteiden kuntoluokitukset tulisi kuitenkin tarkistaa manuaalisesti ja lisätä järjestelmään.

Toimeksiantaja toivoi myös järjestelmästä löytyvän kenttä, jonka avulla voi estää kyseisen tuotteen myynnin Discogs-palvelun kautta. Tämän avulla asiakas voisi halutessaan esimerkiksi lisätä tuotteelle tietoja, ilman tarvetta asettaa ne heti myyntiin.

Suunnitteluvaiheessa ei ollut vielä selvää mitä ohjelmointikieltä tai kolmannen osapuolen ohjelmakirjastoa toteutuksessa tultaisiin käyttämään. Jotta toteutus tulisi sujumaan mahdollisimman hyvin ja sujuvasti, riippumatta valituista teknologioista, päivitettiin kummatkin vaihtoehdoista ajan tasalle sekä lisättiin puuttuvat ominaisuudet. PHP-kirjasto päivitettiin PHP-kielen versioon 7.3, sillä tässä vaiheessa ei ollut vielä selvillä yksityiskohtia kohdejärjestelmästä, kuten kielen versio.

Python-kirjastoon lisättiin puuttuvia ominaisuuksia, kuten myynti-ilmoitusten luominen ja muokkaaminen sekä korjattiin muutama ohjelmointivirhe, jotka ilmenivät ominaisuuksien toteutuksen lomassa. Molempien ohjelmakirjastojen päivitys ja täydennys helpotti integroitavan ohjelmiston suunnittelun aloittamista, siitä huolimatta kumpi kirjasto lopulta valittaisiin. Ohjelmakirjastojen päivitys mahdollisti myös syvemmän tutustumisen Discogs-rajapintaan ja sen toimintaan ohjelmakoodien ja esimerkkien kautta.

5.1.1 Myynti-ilmoitusten saldojen päivitys

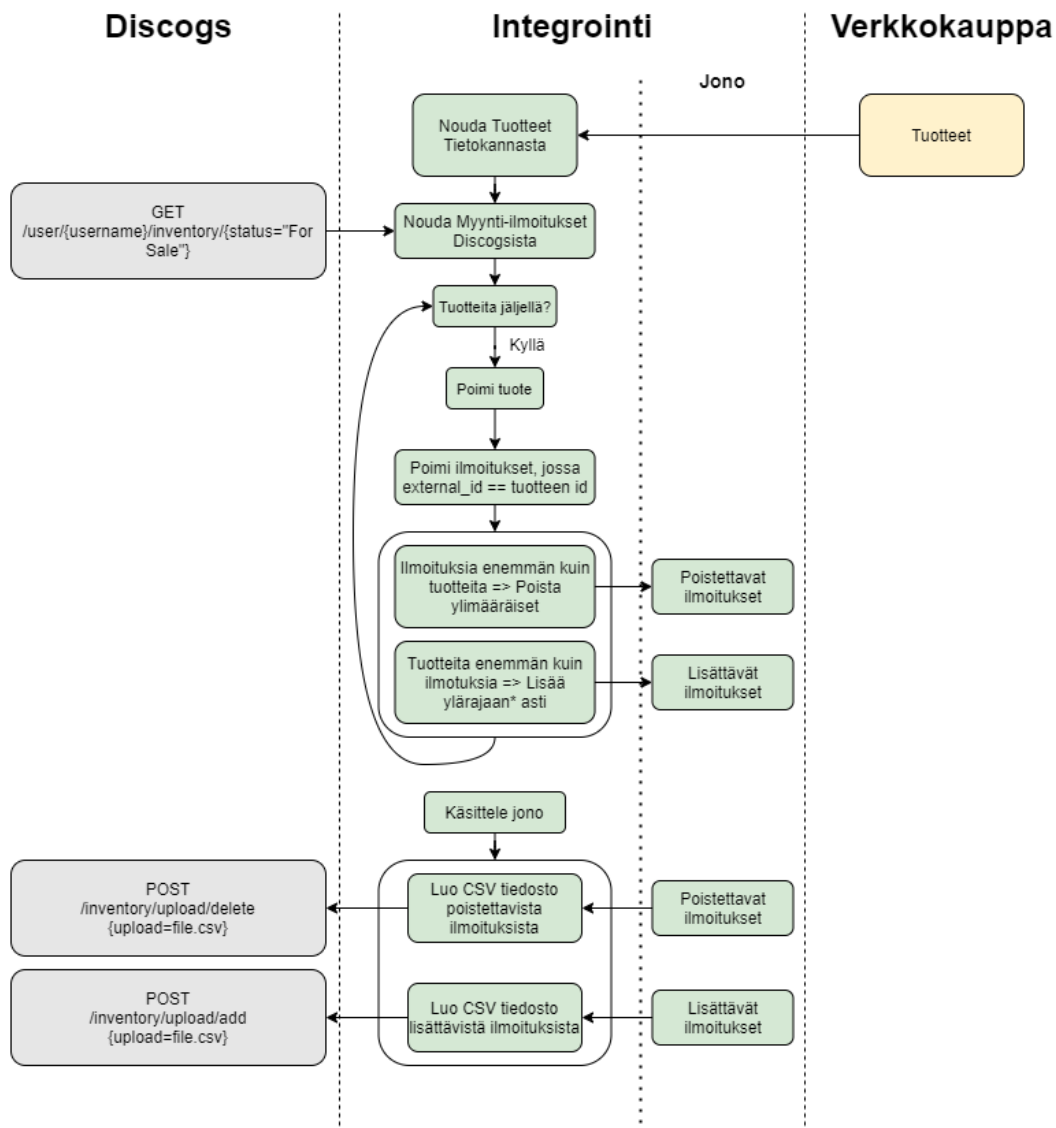
Seuraavaksi tuli suunnitella kuinka reagoidaan kohdejärjestelmässä tapahtuneeseen tuotteen saldon muutokseen. Esimerkiksi jos tuotetta ei ole alun perin varastossa yhtään ja tuotetta saapuu varastoon kaksikymmentä kappaletta, tuleeko kaikki tuotteet lisätä heti Discogs-markkinapaikalle myyntiin vai asetetaanko jokin raja, jonka yli myynti-ilmoituksia ei

tehdä? Toisena esimerkkinä, kuinka toimia, jos tuotteen saldo laskee ja Discogs-markkinapaikalla on enemmän kyseistä tuotetta myynnissä kuin varastossa on tavaraa?

Usean suunnittelupalaverin jälkeen päädyttiin ratkaisuun, jossa tuotteen saldon muuttuessa varastossa, tulee kyseisestä tuotteesta tehtyjen myynti-ilmoitusten määrä Discogs-markkinapaikalla tarkistaa ja määrä päivittää. Tämä estää niiden tuotteiden myymisen, joita ei ole saatavilla tai tarpeeksi varastossa. Kuvassa 8 kuvataan aliohjelman toimintaa seuraavasti

1. Haetaan tuotteet asiakkaan hallintajärjestelmästä.
2. Haetaan asiakkaan myynti-ilmoitukset Discogs-markkinapaikalta.
3. Poimitaan tuotteista yksi ja verrataan sen saldoa myynti-ilmoitusten määrään, joilla on sama "external_id" kuin tuotteen tunnusnumero.
4. Jos myynti-ilmoituksia on enemmän kuin tuotetta saldossa, lisätään ylimenevä määrä myynti-ilmoituksia jonoon poistettavaksi. Jos myynti-ilmoituksia on vähemmän kuin tuotetta saldossa, lisätään tiedot uusista myynti-ilmoituksista jonoon ylärajaan asti. Yläraja on määritelty tuotekohtaisesti asiakkaan järjestelmässä.
5. Jos tarkistettavia tuotteita on vielä jäljellä, hypätään takaisin vaiheeseen 3.
6. Luodaan CSV-tiedosto poistettavista myynti-ilmoituksista ja lähetetään tiedosto Discogsiin rajapinnan avulla. Tämä poistaa kaikki myynti-ilmoitukset, joiden tunnusnumeron CSV-tiedosto sisältää.
7. Luodaan CSV-tiedosto lisättävistä myynti-ilmoituksista ja lähetetään tiedosto Discogsiin rajapinnan avulla. Tämä lisää myynti-ilmoitukset Discogs-markkinapaikalle.

Kuva 8. Diagrammi myynti-ilmoitusten määrien päivityksestä.



* Yläraja on määritelty tuotekohtaisesti järjestelmässä

5.1.2 Tilauksen käsittely

Työn lopussa jäi vielä aikaa tehdä alustava suunnitelma Discogs-markkinapaikan kautta tehtyjen tilausten käsittelystä. Kuvassa 9 kuvataan tilauksen käsittelyn vaiheita kahdessa osassa, ensimmäisessä osassa reagoidaan uuteen tilaukseen seuraavasti

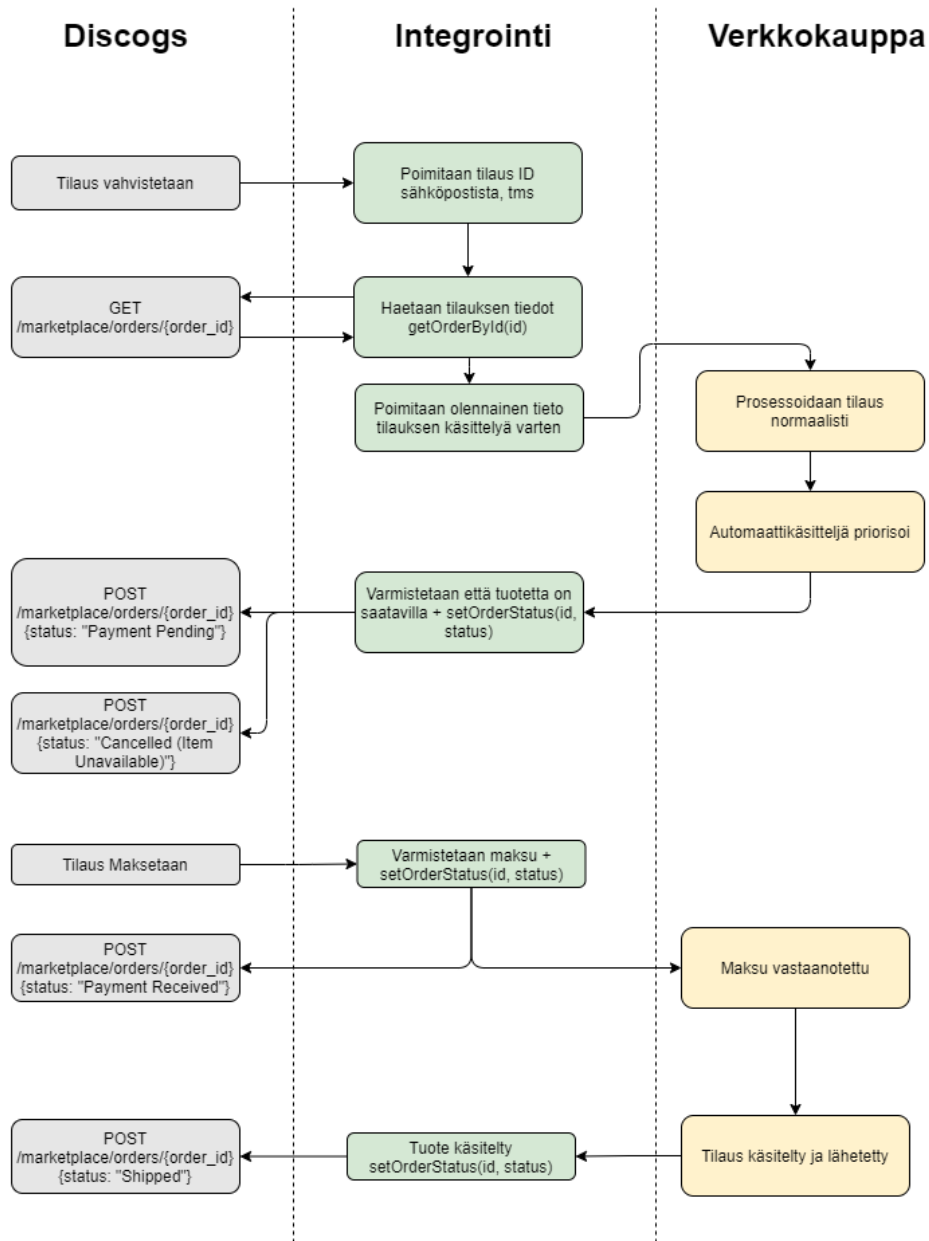
1. Saadaan tieto tilauksen vahvistuksesta esimerkiksi sähköpostiin, josta poimitaan ohjelmallisesti tilauksen tunnus (id). Tämän tunnuksen avulla haetaan tilauksen tiedot Discogs-rajapinnasta. Haetuista tiedoista poimitaan kaikki olennainen tieto tilauksen käsittelyyn asiakkaan järjestelmässä.
2. Poimittu tieto syötetään asiakkaan järjestelmään, jossa näiden tietojen perusteella luodaan uusi tilaus. Tilaus prosessoidaan ja priorisoidaan normaalisti muiden asiakkaan verkkokaupan kautta tulleiden tilausten tavoin.
3. Varmistetaan tilauksessa ostetun tuotteen tai tuotteiden saatavuus. Jos saatavuutta on, Discogs-rajapinnan avulla päivitetään Discogs-tilauksen tilaksi "Payment Pending" eli odotetaan maksua. Jos saatavuutta ei ole, tilaksi muutetaan "Cancelled (Item Not Available)" eli perutaan tilaus.

Toisena osana tilauksen käsittelyä reagoidaan tilauksen maksuun seuraavasti

1. Saadaan tieto tilauksen maksusta esimerkiksi sähköpostista ohjelmallisesti tai käyttäen jotain muuta tapaa, kuten PayPal-maksupalvelulla maksettaessa, tämän tarjoamalla rajapinnalla.
2. Varmistetaan maksu esimerkiksi yllä mainitun PayPal:in rajapinnan kautta ja ilmoitetaan maksun saapumisesta ja hyväksymisestä tilaukseen Discogs-markkinapaikalla kuin myös asiakkaan järjestelmässäänkin.
3. Kun tilaus on asiakkaan järjestelmässä käsitelty kokonaisuudessaan ja lähetetty tilauksen tekijälle, päivitetään Discogs-markkinapaikan tilauksen tilaksi "Shipped" eli tilaus on lähetetty.

Koska moni asia tilauksen käsittelystä ei ole vielä täysin selvillä, kuten tilauksen tunnuksen poimiminen ja maksun varmistus, Kuvassa 9 esitetyt vaiheet voivat vielä muuttua tulevaisuudessa. Vaiheet on kuitenkin käyty alustavasti läpi toimeksiantajan kanssa ja palautteen perusteella tästä on hyvä lähteä jatkokehittämään.

Kuva 9. Diagrammi tilauksen käsittelystä.



Uuden tilauksen (ID) poimiminen sekä maksun varmistamisen vielä auki

Tässä esimerkissä discogsiin ilmoitetaan vain jos tuote ei ole ollenkaan saatavilla tai on käsitelty ja lähetetty

Tässä esimerkissä ei ole otettu huomioon osittaisia puutteita tilauksen tuotteissa, esim. vain toinen tuote saatavilla

5.2 Integroitava ohjelmisto

Varsinainen toteutusvaihe lähti liikkeelle valitsemalla integroinnissa käytettävä kolmannen osapuolen ohjelmakirjasto Discogs-palvelun rajapinnan kanssa keskusteluun. Vaihtoehtoista päädyttiin PHP-kirjastoon, sillä se oli kirjoitettu samalla ohjelmointikielellä kuin kohdejärjestelmäkin.

Jotta integroitavan ohjelmiston voisi räätälöidä kohdejärjestelmää varten, toimeksiantaja toimitti kopion kohdejärjestelmän tietokannasta, joka sisälsi vain integroinnin kannalta oleelliset tiedot kuten järjestelmän tuotteet ja niiden tiedot.

Ohjelmisto alkoi rakentumaan aluksi prototyypin muodossa, jossa kehitettiin hyvin yksinkertainen ohjelmaskripti. Skriptissä oli ennalta määritelty tuotetunnus, jonka avulla haettaisiin tuote tietokannasta. Skriptissä oli myös määritelty Discogs-tunnus, joka vastaa kyseessä olevaa tuotetta Discogs-palvelussa. Näitä tunnuksia käytettäisiin myynti-ilmoituksen luonnissa. Ohjelmaskriptiä suoritettaessa, haettiin ensin tuote tietokannasta sekä sille asetettu hinta. Tämän jälkeen luotiin uusi myynti-ilmoitus Discogs-markkinapaikalle.

Prototyyppiin lisättiin seuraavaksi ominaisuus, joka tarkisti, onko Discogs-markkinapaikalla myynti-ilmoituksia tästä tietystä tuotteesta enemmän kuin on kyseistä tuotetta varastossa. Tarkistuksen jälkeen skripti poisti ylimääräiset tai lisäsi puuttuvat myynti-ilmoitukset Discogs-markkinapaikalle, tietokannasta löytyvän saldon mukaan.

Tässä vaiheessa ei kuitenkaan ollut määritelty ylärajaa myynti-ilmoitusten määrälle ja testauksessa käytetyn tuotteen saldo sattui olemaan useita kymmeniä, jolloin myynti-ilmoituksia tehtiin saman verran. Tämän seurauksena Discogs-rajapinnan pyyntömäärien raja tuli nopeasti vastaan, joka nosti huolen siitä, kuinka ohjelmisto toimisi kymmenillä tai jopa sadoilla tuotteilla, vaikka niiden myyntimäärät olisikin rajoitettu muutamaan kappaleeseen kerralla.

Tässä vaiheessa toteutusta ohjelmaskripti pilkottiin pienempiin osiin, jotta jatkossa uusien ominaisuuksien lisääminen onnistuisi helpommin ja ohjelmakoodi olisi paremmin luettavissa ja ymmärrettävissä. PHP-kirjaston ympärille luotiin luokka, joka huolehtisi kaikesta

kommunikoinnista Discogs-rajapinnan kanssa sekä tämän palauttaman datan prosessoinnista. Samalla luotiin myös luokka, joka hoitaisi kommunikoinnin tietokannan kanssa.

Ohjelmaskriptiin jäi jäljelle erilaisia funktioita ja koodinpätkiä, joista koostui logiikka esimerkiksi päättämään, lisätäänkö vai poistetaanko myynti-ilmoituksia. Nämä siirrettiin myös erilliseen luokkaan sillä, niitä tulotaisiin todennäköisesti käyttämään muuallakin, esimerkiksi toisissa ohjelmaskripteissä.

Toimeksiantajan ehdotuksesta ohjelmistoon lisättiin vaihtoehto lisätä ja poistaa myynti-ilmoituksia CSV-tiedostoilla, jota rajapinta tukee, jolloin pyyntöjä syntyisi vain muutamia. Koska tuotteita voi mennä kerralla myyntiin kymmeniä tai jopa satoja ja lisäysten lisäksi samalla kertaa voidaan poistaa myynti-ilmoituksia, tulisi rajapinnan pyyntörajoitus hyvin nopeasti vastaan. Vaihtoehtona ohjelmistossa olisi mahdollista käyttää eräänlaista rajoitinta joka pyyntörajoituksen täytyessä jäisi odottamaan pyyntöjä vapautumista. Tämä kuitenkin hidastaisi huomattavasti ohjelmaskriptin suorittamista suurilla pyyntömäärillä.

5.3 Demo

Toteutusvaiheen lopussa toteutettiin pienimuotoinen ohjelma demonstroimaan asiakkaalle opinnäytetyön aikana syntyneen ohjelmakirjaston ominaisuuksia sekä alustava esimerkki siitä, kuinka Discogs-tietojen lisääminen verkkokaupan tuotteisiin tulisi tapahtumaan varsinaisessa integroinnissa.

Demo-ohjelman etusivu jaettiin kolmeen osaan, jotka esittivät prosessiin eri vaiheita. Ensimmäisenä oli lyhyt lista verkkokaupan tuotteita (Kuva 10), joille pystyi lisäämään Discogs-tietoja ”Lisää Tiedot” -painikkeesta. Tästä painikkeesta avautui ponnahdusikkuna, jonka kautta voitiin lisätä tuotteelle Discogs:iin liittyviä tietoja (Kuva 11). Jokaisella tuotteella oli myös ”Hae”-painike, jonka avulla voitiin hakea tuotetta vastaava Discogs-julkaisu.

Kuva 10. Demon 1. osa, tuotteet ja haku.

Discogs Haku

Discogs ID	Title	Formaatti	Vuosi	Genre	Linkki
1597994	Accept - Balls To The Wall	CD	2002	Rock	<input type="button" value="->"/>

Verkkokaupan tuotteet

Tuote ID	Title	Saldo	Formaatti	Vuosi	Koodi	Hinta	Muokkaa	Haku
521889	Keziah Lilith Medea (Chapter X)	2	cd	2017		15.99	<input type="button" value="Lisää Tiedot"/>	<input type="button" value="Hae"/>
380942	Raised In Rain	1	cd	2013	5099992836822 50999936822	21.99	<input type="button" value="Lisää Tiedot"/>	<input type="button" value="Hae"/>
3249	Kroketti	2	cd	1997	1000230842 6417734230841	10.99	<input type="button" value="Lisää Tiedot"/>	<input type="button" value="Hae"/>
2872	Doggystyle	1	cd	1993/2001	DROW 116 728706300223	21.99	<input type="button" value="Lisää Tiedot"/>	<input type="button" value="Hae"/>
486591	Agonia	4	cd	2016	INV235 6430015104761	13.99	<input type="button" value="Lisää Tiedot"/>	<input type="button" value="Hae"/>
20	Balls to the wall -remastered-	1	cd	2002	0743219321428	8.99	<input type="button" value="Lisää Tiedot"/>	<input type="button" value="Hae"/>
14019	Singles 93-03	1	cd	2003	0724359314228 724359283227	22.99	<input type="button" value="Lisää Tiedot"/>	<input type="button" value="Hae"/>
41599	Boustrophedon	1	cd		ECM 1873 602517500549	19.99	<input type="button" value="Lisää Tiedot"/>	<input type="button" value="Hae"/>
522617	Tack för kaffet	5	cd	2017	SUB119CD 7393210134645	15.99	<input type="button" value="Lisää Tiedot"/>	<input type="button" value="Hae"/>
330911	2	2	cd	2012	CT164CD 0817949016425	15.99	<input type="button" value="Lisää Tiedot"/>	<input type="button" value="Hae"/>

Kuva 11. Ponnahdusikkuna Discogs-tietojen lisäämiseen.

Lisää Discogs Tiedot ✕

Tuote ID

Discogs ID

Median kunto

Kannen kunto

Discogsissa myytävä maks. määrä

Hinta Discogsissa

Kommentti

Ei Saa Myydä Discogsissa?

Demon toisessa osassa oli lista tuotteista, joille oli edellisessä vaiheessa asetettu Discogs-tiedot. Listan tuotteilla oli myös painikkeet näiden tietojen ja varastosaldon muokkaamiseen (Kuva 12). Saldon muokkaamiseen tarkoitettu painike lisättiin tähän vaiheeseen, jotta voitaisiin demonstroida kuinka automatisointi reagoi saldojen muutoksiin. Integroinnissa ja kohdejärjestelmässä nämä saldot muuttuisivat automaattisesti myyntien sekä varaston täydennysten mukaan.

Kuva 12. Demon 2. osa, Discogs-tiedot.

Verkkokaupan tuotteet Discogs tiedoilla [Päivitä](#)

Tuote ID	Title	Saldo	Max Määrä (Discogs)	Media	Kansi	Hinta	Hinta (Discogs)	Myy Discogsissa?	Muokkaa
20	Balls to the wall -remastered-	1	1	Mint (M)	Mint (M)	8.99	12.99	Kyllä	Tiedot Saldo
370779	Marcando La Historia	3	1	Mint (M)	Mint (M)	19.99	19.99	Kyllä	Tiedot Saldo

Demon kolmannessa osassa oli lista tuotteista tehdyistä myynti-ilmoituksista Discogs-markkinapaikalla (Kuva 13). Kun ”Päivitä Määrät”-painiketta painetaan, ajetaan taustalla suunnitteluvaiheessa syntyneitä diagrammeja (Kuva 8) noudattava ohjelmaskripti, joka päivittää myynti-ilmoitusten määrät Discogs-markkinapaikalla. Varsinaisessa integroinnissa tämä skripti ajettaisiin automaattisesti tietyin väliajoin.

Kuva 13. Demon 3. osa, myynti-ilmoitukset

Discogs Ilmoitukset [Päivitä](#) [Päivitä Määrät](#)

Tuote ID	Discogs ID	Artisti	Albumi	Media	Kansi	Hinta
20	1597994	Accept	Balls To The Wall	Mint (M)	Mint (M)	12.99
3249	385261	Ultra Bra	Kroketti	Mint (M)	Mint (M)	12.99

5.4 Lopputulos

Opinnäytetyön tuloksena syntyi selvitys Discogs-palvelun ja tämän tarjoaman avoimen rajapinnan ominaisuuksista ja näiden toiminnasta sekä selvitys Discogs-markkinapaikan myynti-ilmoitusten ylläpitämisestä automatisoidusti. Lisäksi syntyi ohjelmakirjasto, jonka avulla voidaan siirtää selvityksen tulokset käytäntöön eli rajapinnan kanssa keskusteleminen ja myynti-ilmoitusten ylläpitämisen automatisointi.

Lopputuloksena syntyi myös alustava selvitys Discogs-markkinapaikan tilausten rekisteröimisestä sekä prosessoinnista asiakkaan järjestelmässä mahdollisimman automatisoidusti. Toteutuksen lopussa toteutettiin demo-ohjelma, jonka avulla demonstroitiin asiakkaalle opinnäytetyön aikana tehdyn ohjelmakirjaston ominaisuuksia.

Työn suunnitteluvaiheessa PHP-kirjasto päivitettiin käyttämään ohjelmointikielen versiota 7.3 ja siihen lisättiin puuttuvat ominaisuudet. Kuitenkin toteutusvaiheessa kävi ilmi, kohdejärjestelmän käyttävän PHP-versiota 5.4. Tämä ei kuitenkaan muodostunut suureksi ongelmaksi sillä toteutuksessa syntynyt ohjelmakirjasto on ainoastaan riippuvainen kohdejärjestelmän tietokannasta.

6 Yhteenveto

Opinnäytetyön alussa määriteltiin työn tavoitteiksi selvitystyön lisäksi integrointivaihe, jonka tarkoituksena oli toteuttaa integrointi työn tuloksena syntynyt ohjelmakirjasto asiakkaan hallintajärjestelmän kanssa. Työn tuloksena syntyi selvitys Discogs-palvelun rajapinnan ominaisuuksista ja kuinka niitä voi hyödyntää integroinnissa myynti-ilmoitusten päivittämiseen ja alustava selvitys Discogs-tilausten käsittelystä hallintajärjestelmässä. Itse varsinaista integrointia ei toteutettu tämä opinnäytetyön aikana, aikataulun sekä työmäärän takia.

Jatkokehittävänä jäi integroinnin lisäksi tarkempi selvitystyö Discogs-tilausten käsittelystä sekä automatisoinnista asiakkaan järjestelmässä. Esimerkiksi kuinka toimitaan, kun tilattuja tuotteita on useampi kappale ja yhtä tuotteista ei ole enää saatavilla? Mitä jos tilaus perutaan asiakkaan toimesta? Tuleeko nämä tapaukset käsitellä manuaalisesti tai voiko

joihinkin tapauksiin kenties reagoida luotettavasti automatiikan avulla? Ensimmäisen esimerkin tapauksessa ilmeisin ratkaisu olisi poistaa kyseinen tuote tilauksesta automaattisesti, mutta Discogs-rajapinnan puutteiden vuoksi tätä ei voi tehdä automaattisesti eikä edes keskitetysti asiakkaan hallintajärjestelmän kautta. Muissa poikkeustapauksissa, joissa automatisointi on jollain tapaa liian monimutkaista tai jopa mahdotonta, voidaan näiden hallinta manuaalisesti keskittää asiakkaan hallintajärjestelmään, olettaen Discogs-rajapinta tukee kyseistä ominaisuutta.

Opinnäytetyö kokonaisuudessaan osoittautui erittäin opettavaiseksi projektiksi, teoriaosuudessaan syvennyttiin rajapintoihin sekä niiden taustalla toimiviin teknologioihin, mikä avarsi ymmärrystä merkittävästi webin rakenteesta ja toiminnasta. Lisäksi työ opetti suunnitelmallisuuden merkityksen ja sen tärkeyden sovelluskehityksessä ja sen vaikutuksiin työn sujumuudessa sekä toteutuksessa.

Opinnäytetyön aikana syntyneen ohjelmakirjasto tulee asiakkaan yksityiseen käyttöön, minkä takia ohjelmakirjaston lähdekoodia ei saa julkaista, mutta työn aikana käytetyt kolmannen osapuolen kirjastot ovat avoimesti saatavilla seuraavista osoitteista

- Python-kirjasto: https://github.com/joalla/discogs_client
- Alkuperäinen PHP 5.4 -kirjasto: <https://github.com/ricbra/php-discogs-api>
- Päivitetty PHP 7.3 -kirjasto: <https://github.com/AnssiAhola/php-discogs-api>.

Lähteet

Berlind, D. (3.12.2015a). *APIs Are Like User Interfaces--Just With Different Users in Mind.*

<https://www.programmableweb.com/news/apis-are-user-interfaces-just-different-users-mind/analysis/2015/12/03>

Berlind, D. (3.12.2015b). *What is an API, Exactly?*

<https://www.programmableweb.com/news/what-api-exactly/analysis/2015/12/03>

CDandLP. (n.d.-a). *About Us.* Haettu 26.4.2021 osoitteesta

<https://www.cdandlp.com/en/about-us/>

CDandLP. (n.d.-b). *API CDandLP (Version 3.0).* Haettu 26.4.2021 osoitteesta

<https://www.cdandlp.com/en/api-developers/>

Discogs. (29.10.2020). *Discogs Blog: Thank You for 20 Years of Discogs.*

<https://blog.discogs.com/en/20-years-of-discogs-2020/>

Discogs.com. (n.d.-a). *Authentication: Discogs API Documentation.* Haettu 16.3.2021

osoitteesta <https://www.discogs.com/developers/#page:authentication>

Discogs.com. (n.d.-b). *Discogs API Documentation.* Haettu 16.3.2021 osoitteesta

<https://www.discogs.com/developers/>

Discogs.com. (n.d.-c). *Discogs API Documentation: Marketplace.* Haettu 16.3.2021

osoitteesta <https://www.discogs.com/developers/#page:marketplace>

eBay Inc. (28.4.2021). *eBay Inc. Reports Better Than Expected First Quarter 2021 Results.*

<https://www.ebayinc.com/stories/news/ebay-reports-better-than-expected-first-quarter-2021-results/>

eBay Inc. (n.d.-a). *All API Documentation.* Haettu 3.5.2021 osoitteesta

<https://developer.ebay.com/docs>

eBay Inc. (n.d.-b). *Our Company.* Haettu 3.5.2021 osoitteesta

<https://www.ebayinc.com/company/>

Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures.* [väitöskirja, University of California]

https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf

Fredrich, T. (n.d.). *REST API Tutorial: What Is REST?* Haettu 22.4.2021 osoitteesta

<https://www.restapitutorial.com/lessons/whatisrest.html#>

Hurbans, R.;& Van Der Westhuizen, M. (n.d.). *The REST Architecture Style.* Haettu 29.4.2021

osoitteesta <https://github.com/rishal-hurbans/The-REST-Architectural-Style#uniform-interface>

- IANA. (n.d.). *Hypertext Transfer Protocol (HTTP) Status Code Registry*. Haettu 31.3.2021 osoitteesta <https://www.iana.org/assignments/http-status-codes/http-status-codes.xhtml>
- IBM. (19.8.2020). *IBM Cloud Learn Hub*. <https://www.ibm.com/cloud/learn/api>
- IETF. (Maaliskuu 2010a). *PATCH Method for HTTP*. <https://tools.ietf.org/html/rfc5789>
- IETF. (Huhtikuu 2010b). *The OAuth 1.0 Protocol*. <https://tools.ietf.org/html/rfc5849>
- IETF. (Kesäkuu 2014). *Hypertext Transfer Protocol (HTTP/1.1)*.
<https://tools.ietf.org/html/rfc7231>
- Lane, K. (5.10.2020). *Intro to APIs: What Is an API?: Postman Blog*.
<https://blog.postman.com/intro-to-apis-what-is-an-api/>
- MetaBrainz Foundation. (n.d.-a). *MusizBrainz: MusizBrainz API*. Haettu 23.4.2021 osoitteesta
https://musicbrainz.org/doc/MusicBrainz_API
- MetaBrainz Foundation. (n.d.-b). *MusizBrainz: About*. Haettu 23.4.2021 osoitteesta
<https://musicbrainz.org/doc/About>
- MetaBrainz Foundation. (n.d.-c). *MusicBrainz: Statistics*. Haettu 23.4.2021 osoitteesta
<https://musicbrainz.org/statistics>
- Moilanen, J., Niinioja, M., Seppänen, M., & Honkanen, M. (2018). *API-Talous 101*. Alma Talent Oy.
- Mozilla. (n.d.-a). *MDN Web Docs: An overview of HTTP*. Haettu 15.4.2021 osoitteesta
<https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>
- Mozilla. (n.d.-b). *MDN Web Docs: HTTP*. Haettu 2.4.2021 osoitteesta
<https://developer.mozilla.org/en-US/docs/Web/HTTP>
- Mozilla. (n.d.-c). *MDN Web Docs: HTTP Headers*. Haettu 23.3.2021 osoitteesta
<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers>
- Raney, N. (3.2.2021). *Discogs Blog: The 100 Most Expensive Records Ever Sold on Discogs*.
<https://blog.discogs.com/en/discogs-top-100-most-expensive-records/>
- Richer, J. & Antonio, S. (2017). *OAuth 2 in Action*. Manning Publications Co.
- Saunders, L. (15.4.2020). *Discogs: what is it, where it came from, and how to use it*.
<https://happymag.tv/discogs-what-where-how/>
- Spotify. (n.d.-a). *For The Record: Company Info*. Haettu 22.4.2021 osoitteesta
<https://newsroom.spotify.com/company-info/>
- Spotify. (n.d.-b). *Web API*. Haettu 22.4.2021 osoitteesta
<https://developer.spotify.com/documentation/web-api/>

Spotify. (n.d.-c). *Web API Reference: Tracks API*. Haettu 8.4.2021 osoitteesta

<https://developer.spotify.com/documentation/web-api/reference/#category-tracks>