

## Paytrail-moduuli Drupalille

Joona Viertola



<b>Tekijä(t)</b> Joona Viertola	
<b>Koulutusohjelma</b> Tietojenkäsittelyn koulutusohjelma	
<b>Opinnäytetyön otsikko</b> Paytrail moduuli Drupalille	<b>Sivu- ja liitesivumäärä</b> 18 + 4
<b>Opinnäytetyön otsikko englanniksi</b> Paytrail module for Drupal	
<p>Opinnäytetyön tavoitteena on kuvata toimeksiantajalle tehtyä projektia, jonka lopputuloksena on Paytrail-maksupalvelua käyttävä Drupal-moduuli. Moduulin tarkoitus on mahdollistaa ostotapahtuma verkkosivun vierailijalle Paytrail-maksupalvelussa. Myös moduulin ylläpitotoiminnot ovat tärkeä osa projektia.</p> <p>Opinnäytetyö kertoo projektista, jonka lopputulos moduuli on. Kuvaus kertoo projektin alussa alkaneesta Drupaliin ja Paytrailiin tutustumisesta ja integroinnin suunnittelusta. Projektin aikaisia päätöksiä ja käytäntöjä tuodaan esille kuvauksessa työn edetessä. Päätöksiä ja käytäntöjä perustellaan kehittämisen ja jatkokäytön kannalta. Myös suunnittelussa että kehittämisessä käytettyjä työkaluja ja käytäntöjä tuodaan esille.</p> <p>Opinnäytetyön lopussa käydään projektin päättymisen jälkeisiä pohdintoja, millä tavoin projekti onnistui ja missä kohdin olisi voitu parantaa. Oppimiskokemuksissa kerrotaan omia henkilökohtaisia projektin aikana opittuja käytäntöjä. Kirjoitelma sisältää myös ensivaikutelma Drupalista ja Paytrailista uudelle käyttäjälle.</p> <p>Opinnäytetyön kirjoittaja on verkkokehittämiseen osaamistaan keskittänyt ohjelmoija. Koke-musta löytyy palvelinympäristöstä, tietokannoista ja verkkosivujen frontend- ja backend-kehittämisestä. Drupal ja Paytrail projektin alussa olivat kirjoittajalle uusia käsitteitä ja projekti toimi ensikosketuksena näiden käyttöön.</p>	
<b>Asiasanat</b> Integraatio, projekti, maksupalvelu, sisällönhallintajärjestelmä, Drupal, Paytrail	

<b>Author(s)</b> Joonas Viertola	
<b>Degree programme</b> Business Information Technology	
<b>Report/thesis title</b> Paytrail Module for Drupal	<b>Number of pages and appendix pages</b> 18 + 4
<p>The purpose of the thesis is to describe a project made for an employer which has resulted in configuring a Drupal module using the Paytrail payment service. The purpose of the module is to enable payment functions for a visitor in Paytrail service. Also the module administration side is an important part of using it.</p> <p>The thesis describes the project where the module serves as the end result. The thesis reflects on learning the basics of Drupal and Paytrail for implementing the module. Decisions and practices are brought out to describe the work. Decisions and practices are explained in their continuous development and usage. The tools and practices used for planning and developing are in the focus of attention.</p> <p>At the end of the thesis there is a reflection whether the project has been a success and what could have been done in another way. The learning outcomes include some improvement of the techniques and practices. The thesis describes the beginner experience of a user unacquainted with Drupal and Paytrail of an inexperienced user.</p> <p>The author of this thesis is a developer focused on a web based developing. His overall experience includes the server side, databases and website frontend and backend developing. Drupal and Paytrail were the new things at start of the project. The project served as the first attempt to integrate these software solutions.</p>	
<b>Keywords</b> Integration, project, payment service, content management platform, Drupal, Paytrail	

## Sisällys

1 Johdanto .....	1
2 Suunnittelu .....	2
3 Moduulin rakenne .....	3
3.1 Drupal moduulin osat .....	3
3.2 Paytrail rajapintoihin liittyvät osat .....	3
4 Toteutustavat ja työkalut .....	5
5 Toteutus .....	7
5.1 Paytrailin rajapinta .....	7
5.2 Drupal moduuli .....	8
6 Testaus .....	11
7 Julkaisu .....	12
8 Tulokset .....	13
9 Johtopäätökset .....	14
10 Yhteenveto .....	15
10.1 Jatkotoimenpiteet .....	15
10.2 Oppimiskokemukset .....	15
11 Käsitteet .....	17
Lähteet .....	18
Liitteet .....	20
Liite 1. Connect API kuvaus 1 .....	20
Liite 2. Connect API kuvaus 2 .....	21
Liite 3. REST Interface kuvaus .....	22
Liite 4. Form interface kuvaus .....	23

# 1 Johdanto

Tämän opinnäytetyön tavoitteena on kertoa toimeksiantajalle tehdystä projektista, jonka lopputuloksena oli Drupalin ja Paytrail-maksupalvelun yhdistäminen toimivaksi moduuliksi. Kertomuksessa käydään Drupalin ja Paytrailin ominaisuuksien käyttämistä ja oppimista, projektin aikaisista työkaluista sekä kehittämisen aikana tehdyistä päätöksistä.

Drupal on avoimen lähdekoodin PHP-pohjainen sisällönhallintajärjestelmä verkkosivuja varten. Sen suuri yhteisö pitää sisällään yli miljoona kehittäjää ja käyttäjää, jotka ylläpitävät ja kehittävät Drupalia ja sen moduuleja. Drupal sopii moneen monimutkaisempaan hankkeeseen sen moduulirakenteen vuoksi. Se tarjoaa dynaamisen verkkosivuston rakentamisen yhteisesti sovittujen standardien kautta nopeasti.

Paytrail tarjoaa yhteisen rajapinnan Suomalaisia pankkeja ja muita maksutapahtumia varten. Palvelussa verkkokaupan omistajalla on oma tilinsä, josta maksetaan maksua perustuen aikaan ja transaktioiden määrään. Palvelu tarjoaa kehittäjille ilmaisen integraatiooppaan ja testitunnukset osaan rajapinnoista. Integraatio-opas pitää sisällään maksupalveluiden lisäksi myös esimerkiksi kirjanpitoon liittyviä rajapintoja.

Drupal.org sivusto pitää sisällään käyttäjien lähettämiä moduuleja laajentamaan Drupal-asennuksen toimintoja. Aloitushetkellä se ei sisältänyt kunnollista Paytrailin Connect API:n, REST ja Form Interface -rajapintoja kattavaa moduulia. Toimeksiantaja Druid Oy näki tarvetta tälle tulevaisuudessa, jolloin moduulia lähdettiin toteuttamaan.

## 2 Suunnittelu

Moduulin suunnittelun pohjana käytettiin Paytrailin kehittäjille suunnattua integraatio-opasta. (Paytrail 2014.) Integraatio-opas sisälsi dokumentaatiota tekstimuodossa että kuvioina. Kuvioista esimerkkinä sekvenssikaaviot. Tärkeintä moduulin suunnittelussa oli loppukäyttäjän, verkkokaupan ja Paytrailin välinen viestinkulku. Tämä viestinkulku vaikuttaa moduulissa sen rakenteeseen, käyttöliittymään, tietomuotoihin ja käytettävyyden suunnitteluun.

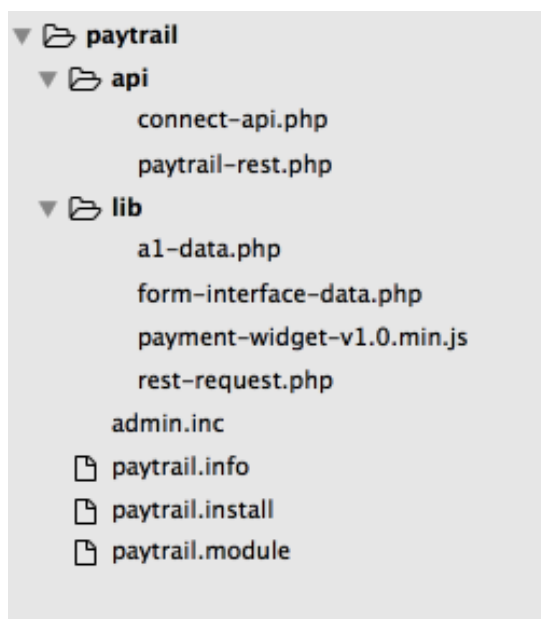
Paytrailin integraatio-opas sisältää erilaisia maksutapoja, maksun tarkistusrajapintoja, maksu-upotusrajapinnan, maksutapakuvituksia ja tietomuotoja. Eri tekniikat on myös opassa selitetty eri tavoilla, jolloin tärkeimmäksi aluksi ennen varsinaista moduulin suunnittelua oli luoda selkeä kuva itselle tarvittavista ominaisuuksista. Suunnittelun aikana luotiin sekvenssikaavioita maksutapojen viestinkulusta. (Liitteet 1, 2, 3 ja 4.)

Heti suunnittelun alusta lähtien haluttiin pitää Paytrailin maksupalveluun yhteyttä ottava lähdekoodi erillään jatkokäyttöä varten. Tämä valinta auttaa myös lähdekoodin selkeydessä, että on tutumpi käytäntö ohjelmoinnissa.

Moduulia lähdettiin selkeästi rakentamaan Drupal 7 –versiolle sen ollessa uusin julkaistu versio. Version tuki on pidempi ja näin myös moduulin käyttöaika (Drupal 2015a). Drupalin puoli moduulista vaati paljon suunnittelua. Maksamisen pitää olla selkeää ja yksinkertaista, joka tarkoittaa käytettävyyteen panostamista. Tämä on suorassa yhteydessä Paytrailin rajapintojen kanssa. Paytrailin integraatio-opasta käytettiin apuna maksamisessa ja moduulin asetus-sivun luomisessa.

### 3 Moduulin rakenne

Moduulin suunnittelun aikana luotu ajatus Paytrailin maksurajapintoihin yhteyttä ottavien osien erillään pitämisestä oli tärkeä rakenteen suunnittelun osa. Tämä loi lisätyötä REST-pyyntöjen kanssa. Drupal 7 tarjoaa valmiin toiminnallisuuden kyseisiä pyyntöjä varten, mutta riippuvuuden aiheuttamasta syystä nähtiin paremmaksi ideaksi luoda oma tuki, jonka voi myös vaihtaa tarpeen mukaan



Kuvio 1. Tiedostorakenne

#### 3.1 Drupal moduulin osat

Moduulin rakenne pohjautuu Drupal 7:n standardin mukaiseen moduulirakenteeseen (Drupal 2012; Drupal 2013). Moduulin kansion nimi toimii moduulin nimenä. Tiedosto `paytrail.info` pitää sisällään metatietoa moduulista. Tiedosto `paytrail.module` on päätiedosto moduulin lähdekoodille. Se pitää sisällään Drupal 7:n standardin mukaista koodia ja sitä voidaan myös eriyttää tarpeen mukaan. Tiedosto `admin.inc` on esimerkki eriytetystä osasta. Se pitää sisällään ylläpitäjälle tarpeellisia toiminnallisuuksia. Se ladataan vain jos sille on tarvetta. Tämä siis nopeuttaa että selkeyttää lähdekoodia.

#### 3.2 Paytrail rajapintoihin liittyvät osat

Muut tiedostot ja kansiot rakenteessa liittyvät Paytrailin rajapintojen viestintään ja tietomuotoihin. Kansio `api` pitää sisällään rajapintoihin yhdistämiseen liittyviä päätiedostoja. Kansio `lib` taas pitää sisällään muita tarvittavia toimintoja kuten tietomuotoja, yhteisiä toimintoja ja täydentäviä ominaisuuksia.

### **3.2.1 Api tiedostot**

Tiedosto connect-api.php sisältää nimensä mukaisesti Paytrailin Connect API maksurajapintaan yhdistämistä varten olevia toimintoja. Tiedosto paytrail-rest.php pitää sisällään taas Paytrailin REST-rajapinnan toimintoja. Tämä tiedosto on Paytrailin itse tarjoama, eikä se sisällä projektin aikana tehtyjä muutoksia.

### **3.2.2 Lib aputiedostot**

Tiedostot a1-data.php ja form-interface-data.php ovat tietomuotoja, joita käytetään maksurajapinnan vaatimaan maksun muodostukseen. Tiedosto a1-data.php on tarkoitettu Paytrailin Connect API:lle ja form-interface-data.php taas Paytrailin lomakerajapinnalle.

Tiedosto payment-widget-v1.0.min.js on tarkoitettu maksutapavalinnan upottamiseen REST- ja lomakerajapinnoille. Tiedosto on Paytrailin itse tarjoama, eikä se sisällä projektin aikana tehtyjä muutoksia.

Tiedosto rest-request.php pitää sisällään standardin mukaisten REST-pyyntöjen lähettämiseen tarkoitettua tukea. Toiminto on eriytetty riippuvuussyistä ja mahdollista jatkokehityksen helpottavaksi tekijäksi.



## 4 Toteutustavat ja työkalut

Projektin tekemiseen käytettiin Applen MacBookkia, jonka käyttöjärjestelmänä toimi OS X Yosemite. Joitain kehitystyökaluja ei löydy muilta käyttöjärjestelmiltä. Suuri osa kehitystyökaluista ovat silti työpöytäkäytössä järjestelmäriippumattomia ja kaikille löytyy korvaava sovellus.

Taulukko 1. Kehittämistyökalut

Ohjelman nimi	Käyttötarkoitus
Sublime 3	tekstieditori
iTerm	pääte-emulaattori
Google Chrome	verkkoselain
Drush	komentorivityökalu Drupalille
Vagrant	komentorivityökalu virtuaalikoneille
Vim	komentorivitekstieditori
Sequel Pro	tietokannanhallintatyökalu

Kehittämistyökaluista komentorivityökaluja käytettiin virtuaalikoneen sisällä. Virtuaalikoneessa käyttöjärjestelmänä käytettiin Ubuntu 14.04 LTS –julkaisua. Tiedostoja pystyttiin muokkaamaan virtuaalipalvelimen sisällä joko Vimillä tai isäntäkoneen kanssa jaetun hakemiston kautta Sublimella.

Tiedonhaku tapahtui internetin dokumentaatioista, keskustelupalstoista ja monista yksittäisistä henkilökohtaisista blogeista hakukoneen avulla. Erikoisissa ohjelmointitarpeissa jouduttiin monesti käyttämään monen lähteen yhdistelmää. Paytrailin kanssa sen oma integraatio-opas oli ainoa lähde. Drupalin kanssa työskennellä taas jouduttiin monesti lukemaan ohjeita kehittäjien omilta verkkosivuilta. Drupal.org-sivun omat dokumentaatiot olivat monesti puutteellisia, eivätkä antaneet kattavaa kuvausta.

Taulukko 2. Tiedonhakuverkkosivut

Sivun nimi	Tyyppi
Drupal.org	yhteisön dokumentaatio
Drupalcontrib.org	moduulidokumentaatio
Stack Overflow	kysy ja vastaa foorumi
Paytrail.com	Integraatio-opaat
Php.net	PHP–dokumentaatio

Ohjelmointitapoihin ja käsitteisiin lähdettiin yleisten käytäntöjen ja standardien kautta. Drupalin ohjelmointistandardit (Drupal 2015b) määrittivät paljon moduulin Drupalin puolen osien ohjelmoinnista. Paytrailin osia ohjelmoitiin PHP:n olio-ohjelmoinnilla. Olio-ohjelmointiin syynä lähdekoodin pysyminen yksinkertaisena ja osien jaottelun järkevä lähestymistapa. Koska olio-ohjelmointi ei määrittele esimerkiksi olion parametreja käsitteleville metodeille pakollista standardia on tukeuduttu yleisiin käytäntöihin. Käytännöistä löytyy keskustelua puolesta ja vastaan internetin keskustelupalstoilla.

## 5 Toteutus

### 5.1 Paytrailin rajapinta

Ohjelmointiin lähdettiin Paytrailin rajapinnan osista. Tähän syynä testattavuus. Kyseiset rajapinnat voitiin testata ilman käyttöliittymää, luomalla erillisiä pieniä koodinpätkiä. Käyttöliittymä ja tiedon kuljettaminen ovat taas vahvasti rajapintakutsuista riippuvaisia.

#### 5.1.1 Connect API

Toteutukseen lähdettiin Connect API:n toiminnallisuuksien toteuttamisella. Heti ensimmäisenä ongelmatilanteena nähtiin REST-toimintojen kanssa. PHP ei tarjoa suoraan helppokäyttöisiä funktioita tai olioita. Drupal sisältää HTTP-pyyntöjen lähettämiseen tarkoitetun funktion (Drupal 2011a), mutta sen käyttäminen loisi heti toimintojen käyttöön riippuvuuden Drupalista.

Connect API:n monimutkaisuuden vuoksi se vei toteuttamisessa eniten aikaa. Toteutus tehtiin toki suunnittelumateriaalin kanssa, mutta eri kutsujen ja palautusten JSON-datan ymmärtäminen oli hidasta. REST-hakujen header-osio oli ainoa jota pystyttiin automatisoimaan osaksi omaan funktioonsa. REST-pyyntöjen sisällä tieto kulkeva tieto on voimakkaasti käyttäjän tekemien valintojen luomaa. Rajapintaa käyttävien rajapintaluokkien rakennetta ohjelmoijan näkökulmasta oli tässä kohtaa vielä vaikea pohtia. Rajapintaluokkien pitää olla mahdollisimman monikäyttöisiä. Rakenteen tarve on voimakkaasti riippuvainen verkkokaupan asiakkaan käyttöliittymästä. Tästä syystä oli hyvä miettiä yleispäteviä ja ennen kaikkea helposti ymmärrettäviä rakenteita. Connect API koostuu tapahtumista, joita käytettiin tiedoston rajapintaluokkien runkona. Ne ovat Paytrailin dokumentaation avulla ymmärrettävissä. Connect API rajapintaan liittyvät toiminnot ohjelmoitiin olio-ohjelmoinnilla. Tämä oli mieluinen vaihtoehto selkeyttävänä tekijänä, mutta se on myös luonnollisemman tuntuinen kun luodaan yhteyttä. Olion ilmentymä on yhteys, jota hallinoidaan metodien avulla.

Metodit Connect API:ssa ottavat parametreina vastaan vain JSON-tietoa, joka on tarkoitettu REST-pyyntöjen body-osioksi. Muissa tapahtuma-metodeissa, paitsi itse ostoksen tiedot tekevässä chargePayment-metodissa ohjelmoija on täysin vastuussa JSON-rakenteen toimivuudesta. Metodissa chargePayment on Paytrailin A1-määritelmän mukaista JSON-tietoa. Tämä rakenne on liian monimutkainen nopeasti opittavaksi tai käsin rakennettavaksi, joten sitä varten tehtiin oma olionsa. Tämäkin tehtiin olio-ohjelmoinnin avulla käyttäen gettereitä ja settereitä, yleisiin käytäntöihin nojaten.

### 5.1.2 REST Interface

Paytrail tarjoaa REST Interface –rajapinnalle valmiin tiedoston PHP-olioilla sen käyttämistä varten. Sama tiedosto sisältää myös S1- ja E1-määritelmät maksujen luomiselle. Tämä nopeutti voimakkaasti ohjelmointia, vaikka loi myös monimutkaisuutta, sillä olion rakenne varsinkin maksun tiedon luomisessa on eriävä gettereiden ja settereiden käytöstä.

### 5.1.3 Form Interface

Form Interface on lomakerajapinta, joka lähetetään asiakkaan koneelta Paytrailin palvelimille. Rajapinnan toimintaperiaate teki tästä rajapintaluokasta muihin nähden suuresti eroavan. Uudelleenkäytettävyys ja ymmärrettävyys etusijalla, lähdettiin tutun kautta liikkeelle ja luotiin ainoastaan S1- ja E1-määritelmän mukaiset tietomuoto-oliot. Oliot toimivat A1-määritelmässä tutulla kaavalla gettoreiden ja settereiden avulla. Olioista pystytään palauttamaan taulukko avain ja arvo pareilla. Tällaisen taulun avulla kehittäjä pystyy helposti asettamaan tiedot lomakkeen kenttiin.

Tämä rajapinta olisi voinut tarvita myös oman dynaamisen olionsa HTML-lomakkeen luontiin. Tämä kuitenkin jätettiin pois luottaen sen olevan turha Drupal 7:en mahdollisuuksien ohella. Muihin projekteihin kyseinen toiminto olisi tosin voinut olla järkevä lisä.

## 5.2 Drupal moduuli

Paytrail-moduulin Drupalin osia lähdettiin ohjelmoimaan ylläpitoasetuksien puolelta, testattavuuden vuoksi. Asetukset tallentuvat aina tietokantaan ja ovat näin tarkasteltavissa ja testattavissa. Verkkosivun käyttäjän puoli on vahvasti asetuksista riippuvainen ja eri toiminnallisuuksien testattavuus on helpompaa ylläpitoasetuksien ollessa valmiit. Myös tuotteiden asettaminen oli tärkeä ominaisuus ennen verkkosivun käyttäjäpuolen toteuttamista.

### 5.2.1 Ylläpitoasetukset

Verkkosivun ylläpitoasetuksilla oli selkeät tarkoitukset. Ne mahdollistavat Paytrail rajapintatiedostojen ominaisuuksien käytön vaihtelut ja maksettavien tuotteiden asettamisen oston varten. Myös asetusten selkeys käytön helpottamiseksi oli yksi tärkeä periaate, josta haluttiin pitää kiinni.

Tässä kohtaa alettiin ensimmäistä kertaa ohjelmoimaan Drupal moduulin ominaisuuksia. Moduulin pohjaa lähdettiin rakentamaan Drupalin kotisivun moduulin kehitysoppaalla. (Drupal 2011b). Selkeyden vuoksi heti lähdettiin miettimään selkeää tapaa jakaa eri toiminnallisuudet eri välilehtiin moduulin asetuksissa. Eri rajapinnat vaativat omat välilehten-

sä asetusmäärästä johtuen. Tuotteiden syöttäminen oli myös hyvä eriyttää yksittäisistä maksutavoista. Myös välilehti yhteisillä asetuksille tai muihin kuulumattomille toiminnoille oli tarpeellinen. Välilehtien sisältö luotiin käyttäen Drupalin Form API:a (Drupal 2011b).

Connect API:n välilehdelle ylläpitäjän asetettavaksi mahdollistettiin käyttäjältä pyydettävät oikeudet, lokalisointi ja tilausnumero. Käyttöoikeuksissa pyydetään maksutus- ja toimitusosoitetta, jotka hyväksymällä käyttäjä antaa Paytrail-palvelulle luvan luovuttaa tiedot verkkokaupalle. Lokalisointi pitää sisällään Paytrailin kieliasetukset, jos käyttäjä siirtyy Paytrailin sivuille maksamaan. Tämän valinnan voi muuttaa myös Paytrailin sivuilla. Tilausnumerosta tehtiin tässä moduulissa staattinen.

REST ja Form Interface –asetukset päädyttiin laittamaan saman välilehden alle, niiden samankaltaisuuden vuoksi. Maksutapahtuma sisältää tilausnumeron, viitenumeron, tilauksen kuvauksen, lokalisoinnin ja mahdollisuuden maksutapaupotukseen. Tilausnumero, viitenumero ja tilauksen kuvaus ovat molemmat staattisia. Paytrail-palvelun lokalisointi on mahdollista ylläpidon puolelta, vaikka käyttäjä voi myös muuttaa sen itse Paytrailin sivuilla. Maksutapaupotus on valinnainen ja mahdollistaa käyttäjän valita maksutapansa jo verkkokaupan puolella ennen kuin siirtyä maksamaan.

REST ja Form Interface -asetukset pitävät myös sisällään maksutavan version S1 tai E1 väliltä valitsemisen. Nämä eroavat toisistaan sillä onko maksussa kyseessä tuotteita ja toimitustiedot vai kiinteä maksu. Näiden versioiden eri asetukset eriteltiin omiin alueisiinsa selkeyttämään asetuksia.

Tuotevälilehti pitää sisällään tuotteiden hallinnan. Se ilmoittaa jos tuotteita ei ole asetettu tyhjällä taululla. Tuotteita pystyy helposti lisäämään Create new product -painiketta klikkaamalla. Tuotteen lisäyksessä sille valitaan maksutavan versio, tuotekoodi, nimi, määrä, yksikköhinta, vero, alennusprosentti ja tuotteen tyyppi.

Tuotteelle valitaan maksutapaversio, koska halutaan varmistaa ettei tuotteelle anneta väärää tietoa. Alennusprosenttia ja tuotteen tyyppiä käytetään vain jos E1-maksutapaversio on käytössä, muulloin nämä kaksi tietoa jätetään käyttämättä maksua luodessa. Ne säilyvät asetusten tietokannassa, mutta eivät muutu maksutapaversiota valittaessa. Tuote tallennetaan tietokannassa tuotteita sisältävään tauluun. Tämä nähtiin parhaimmaksi lähestymiskohdaksi monien tuotteiden sisältäessä monta tuotekohtaista tietoa.

Rajapinta- ja tuotevälilehtien lisäksi ylläpitoasetukset pitävät sisällään vielä yhden välilehden yleisille asetuksille. Välilehti pitää sisällään muihin välilehtiin kuulumattomat yleiset

asetukset. Näitä ei kertynyt paljoa, mutta ovat silti järkeviä erillään. Välilehdeltä löytyvät myyjän tunnus- ja salasana-kentät sekä valinta mitä rajapintaa käytetään. Tämä välilehti on ensimmäisenä auki ylläpitoasetuksissa sen sisällön vuoksi.

### **5.2.2 Ostotapahtuman käyttöliittymä**

Ostotapahtuman käyttöliittymä luotiin suoraviivaisiksi tiedon lähettämiseksi. Tämä tarkoittaa suoraviivaista toimintaa miten käyttäjä lisää tietonsa ja hyväksyy maksun. Eri rajapinnat toimivat eri tavoin ja tästä syystä käyttökokemus on erilainen riippuen minkä rajapinnan verkkokaupan ylläpitäjä on asettanut päälle. Tarvittaessa käyttäjä saa myös linkin, jonka avulla hän pääsee siirtymään Paytrail-palvelun verkkosivuille antamaan lisätietoa. Käyttöliittymä ilmoittaa kun maksaminen on onnistunut.

## 6 Testaus

Testausta tehtiin kahdessa osassa. Rajapintaluokat testattiin ennen Drupalin osiin siirtymistä. Näin virheiden etsintää saadaan rajattua Drupalin osia testatessa. Myös kun Paytrailin rajapinnat olivat vielä lähimuistissa, oli järkevämpää testata ne ensin. Rajapintaluokien toimivuutta testattiin erillisillä staattisilla koodinpätkillä. Staattisilla tässä tilanteessa tarkoitetaan tilaustietojen antamista koodin sisältämänä listana.

REST ja Form Interface sisälsivät valmiit julkiset testitunnukset, toisin kuin Connect API. Tunnuksen luominen vain testausta varten katsottiin liian kalliiksi, joten otettiin yhteyttä Paytrailin tekniseen asiakaspalveluun. Tätä kautta saatiin testitunnus käyttöön projektia varten.

Toinen testaus tehtiin Drupalin osien valmistuttua. Näissä testattiin asetusten tallentamista, käyttöliittymän toimivuutta, tietojen lähettämistä ja maksutapahtuman onnistumista odotetulla tavalla. Asetusmuutokset pystyttiin tarkastamaan käsin tietokannasta. Työkaluina tähän käytettiin Sequel Pro -tietokannanhallintatyökalua. Toinen tapa tarkistaa asetuksia ja niiden siirtymistä maksutapahtumaan asti oli käyttöliittymään tietojen tulostaminen.

Ostotapahtuman käyttöliittymää testattiin käyttötapausten avulla (Liite 5). Näissä odotettiin oikeanlaista vastausta Paytrail-palvelulta ja käyttöliittymän reagointia. Lopullinen maksun onnistuminen selvisi pelkästään Paytrailin lähettämästä vastauksesta, mutta se tarkistettiin myös Paytrailin omien verkkosivujen kautta. Ostotapahtumat tallentuivat käyttäjätilin tietoihin, jonka kortilla ostokset tehtiin.

Kun moduuli toimi odotusten mukaisesti oli se valmis julkaisua varten. Seuraavaa vaihetta edelsi vielä koodin refaktorointia, joka koostui pääosin kommentoidun käyttämättömän koodin poistamisesta.

## 7 Julkaisu

Moduulin oli alusta lähtien tarkoitus julkaista Drupal.org:issa. Syynä että se on Drupalin käyttäjäkunnan kesken yleisin julkaisutapa. Se helpottaa moduulin löytämistä ja mahdollistaa moduulin lataamisen Drush-komentorivityökalulla. Drupal.org:issa muiden käyttäjien on myös mahdollista ilmoittaa moduulin ongelmista ja lähettää korjausehdotuksia lähdekoodiin.

Moduulin lähdekoodi ladattiin Drupal.org:issa käyttäjätilin yksityiseen versionhallintaan. Versionhallinnassa luotiin moduulille oma Drupal 7:lle tarkoitettu haara. Paytrailin omat alkuperäiset tiedostot jätettiin tässä vaiheessa vielä lisäämättä versionhallintaan. Syynä tähän Drupal.org:in vaatimus jossa kaiken ladatun lähdekoodin tulee olla GPLv2-lisenssin tai uudemman alaista. Kyseisistä tiedostoista ei löytynyt lisenssistä mainintaa.

Moduulin lähdekoodia ei julkistettu Drupal.org:issa. Tähän syynä moduulin kokonaisuus. Tarkempaa tietoa päätöksestä löytyy Tulokset-osiossa. Yksityistä versionhallintaa on silti mahdollista jakaa muiden käyttöön linkin avulla. Näin lähdekoodia pystyttiin tarkastelemaan Druidin sisällä ja keskustelemaan parannusehdotuksista.



## 8 Tulokset

Projektin lopputuloksena syntyi alkuperäiseen vaatimukseen sopiva moduuli Drupalille, jonka avulla pystytään suorittamaan maksu Paytrailin palveluun odotetuin tuloksin. Koska Paytrailin rajapintaluokat pidettiin erillään on lopputuloksena myös muihin tarkoituksiin sopivaa lähdekoodia Paytrail-integraatiota varten.

Moduulin asentaminen on helppoa, eikä sillä ole riippuvuuksia muista moduuleista. Moduulin käyttäminen vaatii Paytrailiin tarkempaa tutustumista. Käyttäjätunnuksen asettaminen onnistuu vaivatta, mutta maksurajapintojen erovaisuuksien ymmärtämiseen tarvitsee käyttäjän lukea Paytrailin integraatio-opasta.

Projektin aikana selvisi että Drupal vaatii ohjelmoijalta paljon tietoa sen omista standardeista ja tavoista tehdä asioita. Drupaliin tutustumattomalla ohjelmoijalla on siis paljon opetettavaa ennen myös varsinaista toteuttamista.

Projektin lopputulos jätettiin julkaisematta Drupal.org:issa. Vaikka moduuli vastasi alkuperäisiä vaatimuksia, oli sen kokonaisuus silti väärä. Kaikkien maksutapojen mahdollistaminen ylläpitäjän valitsemaa maksua varten oli turhaa. Moduulin toimintaan olisi riittänyt pelkästään lomakerajapinnan tai REST-rajapinnan S1-määritelmän mukainen maksutapah-tuma. Kaikkien rajapintojen käyttö on tarpeetonta ja lähinnä vaikeuttaa moduulin käyttöön ottamista. Moduuli ei myöskään tarjoa kauppiaille tärkeää tietokantaa maksetuista ostokista. Tällöin kauppiaan on mentävä Paytrail-tilille nähdäkseen nykyisen tilanteen. Paras hyöty Paytrail-rajapintaluokista saataisiin liitettynä muitakin verkkomaksupalveluita tarjoavaan moduuliin, kuten Payment-moduuli.

## 9 Johtopäätökset

Moduulin ohjelmoiminen oli iso projekti vaadittavan suuren tietopohjan vuoksi. Drupal on tunnetusti sisällönhallintajärjestelmä jonka opettelu vie aikaa. Vaikka projektin aikana saatiinkin apua ja neuvoja, niin kokonaiskuva varsinkin alussa oli suunnitelmista riippumatta puutteellinen. Paytrailin integraatio-opas oli myös välillä tulkinnanvarainen ja tarvitsi paljon aikaa paneutua.

Moduulin osien erottelu oli loistava ratkaisu. Näin rajapintaluokkien käyttö on mahdollista muissakin projekteissa. Myös niiden julkaisu erillään on hyvä tapa tuoda omaa osaamistaan esille ja käyttöön myös Drupalin käyttäjien ulkopuolelta. Erikoiseksi rajapintaluokat tekee se että koska Connect API -rajapinta ei sisältänyt testitunnuksia, niin se puuttuu monien kolmannen osapuolen tekemistä rajapintakokoonpanoista.

Drupalin osat moduulista sisältävät toimivia kokoonpanoja ja paljon tärkeitä yksittäisiä osia. Ylläpidon hallinta-asetukset ja tietokantarakenteet ovat hyvin toimivia ja uudelleenkäytettäviä. Moduulin ominaisuuksissa on silti parantamisen varaa. Staattisesti asetetulla ostoksella ei ole järkeä käyttää enemmän kuin yhtä maksurajapintaa. Myös tarve tuotteiden erittelylle moduulissa staattisella ostoksella on tarpeeton.

Moduulin alkuperäisenä ajatuksena ollut helppokäyttöisyys ei toteutunut. Rajapinnat ovat valmiiksi sen verran monimutkaisia että tarvitsevat perehtymistä varsinaiseen Paytrailin integraatio-oppaaseen. Tämä ongelma olisi ratkaistavana ylimääräisten rajapintaluokkien poistolla käytöstä.

Maksutapahtuma mahdollistaa maksun ja odotetulla tavalla. Käyttöliittymässä tosin on parantamisen varaa käyttömukavuuden kannalta. AJAX:in käyttö olisi käyttäjäystävällisempi ratkaisu kuin kokonaan uuden sivun lataaminen. Tällöin ostotapahtuma olisi käyttäjälle nopeampi ja mukavampi kokemus. Myös Paytrailin vaatima ostotapahtuman mutkatomuus tulisi täytettyä.

## 10 Yhteenveto

Projektin tavoitteena oli luoda moduuli Drupal 7:lle, joka mahdollistaa verkkosivulla ylläpitäjän päättämän summan maksamisen. Maksupalveluna käytetään Paytrailin maksurajapintoja ja kaikki maksurajapinnat tulee olla toteutettuna.

Moduulin toteuttamista edelsi suunnittelu millä tavoin se pystytään toteuttamaan sisältäen Drupalin moduulirajapintojen ja Paytrailin maksurajapintojen tarkastelun. Paytrailin maksurajapintaoppaan selkeyttämiseksi luotiin sekvenssikaavioita millä tavoin viestintä Paytrailin kanssa toimii. Moduulia aloittaessa ohjelmoimaan päätettiin maksurajapintojen ja Drupalin osien olevan erillä toisistaan ja riippumattomia jatkokehitystä ja selkeyttä varten. Maksurajapintojen toteuttamisen jälkeen niitä testattiin testitunnuksilla ja siirryttiin tämän jälkeen Drupalin osiin. Moduulia aloitettiin tekemään ylläpitopuolelta sisältäen tietokantasuunnitelman tuotteita varten. Ylläpitäjän puolen ja maksutapahtumapuolen jälkeen testattiin kokonaisuutta.

Projektin alkuperäinen idea oli julkaista moduuli Drupal.org-palvelussa, mutta tämä jätettiin tekemättä johtuen kokonaisuuden käytännöllisyydestä. Moduulille tehtiin yksityinen versiohallinta palveluun, jotta sen voi jakaa halukkaille.

### 10.1 Jatkotoimenpiteet

Moduulin maksurajapintojen erittelystä ja riippumattomuudesta johtuen jatkokehitys on tavoitteena. Maksurajapinnat julkaistaan GitHub-palvelussa ja parannellaan, jotta niistä olisi apua muillekin.

Uusi moduuli on mahdollinen toteutettava projektin aikana kertyneen kokemuksen ja käytäntöjen vuoksi. Mahdollista on myös rajapintojen integroiminen valmiisiin maksumoduuleihin jotka mahdollistavat Paytrailin kautta tehtävä maksuja Drupal Commerce -alustalla.

### 10.2 Oppimiskokemukset

Projekti tarjosi laajalti uutta opeteltavaa. Itsessään Drupalin peruskäyttö on jo suuri oppimispaketti monelle. Varsinaisessa moduulin ohjelmoinnissa ei Drupalin perinteistä ylläpitopuolta käyty läpi. Peruskäsitteet kuten node tulivat hyvin selväksi ohjelmoinnin aikana. Myös schema-käsite oli valaiseva ja yleishyödyllinen Drupalin kanssa jatkotyöskentelyä ajatellen. Drush-työkalun käyttäminen oli myös nopeuttava tekijä työskentelyssä ja testaamisessa.

Oppimiskokemus sisältää myös Drupalin ulkopuolelta tekniikoita. Projektissa käytettiin virtualisoituja testialustoja Vagrant-työkalulla, joka helpotti runsaasti testaamista oikeaa tilannetta varten. Drupalia pidetään yleensä Linux-pohjaisissa ympäristöissä, jolloin OS X:llä testaaminen olisi voinut tuoda eroja.

PHP:n olio-ohjelmointi oli keskeinen osa maksurajapintoja. Olio-ohjelmointi ei tuonut uusia asioita itsessään, mutta yleiset käytännöt esimerkiksi olion parametrien muuttamiseen tarvitsi tutkimista. Myös PHP:n käyttäminen REST-rajapintoja varten oli uusi asia. Tätä varten tarvitsi käyttää Curl-kirjastoa, joka tarjoaa paljon ominaisuuksia.

Opinnäytetyön kirjoittamisen vaikeudeksi koitui yrittää pitää teksti ymmärrettävällä tasolla. Projekti piti sisällään monia erilaisia tekniikoita, joiden aukaiseminen olisi pidentänyt merkittävästi kirjoitelman pituutta ja samalla vienyt huomiota pois itse projektista. Kirjoitus sisältää paljon käsitteitä, jotka vaativat perehtymistä.

## 11 Käsitteet

AJAX	Tekniikka sisällön lataamiseksi palvelimelta ilman koko sivun uudelleen hakemista. (Mozilla 2015.)
Curl-kirjasto	Kirjasto tiedonsiirtoon URL-syntaksin avulla (Curl 2015a). Projektissa käytettiin apuna PHP-toteutusta (Curl 2015b).
Drupal	PHP-pohjainen sisällönhallintajärjestelmä verkkosivuja varten.
Drupal Commerce	Drupal-kokonaisuus joka sisältää verkkokaupan toimintaan tärkeitä moduuleja. Sisältää oman erikoistuneen yhteisön, joka tarjoaa tukea toiminnalle. (Drupal Commerce 2015.)
moduuli	Lisäosa Drupalia varten toimintojen lisäämistä varten. (Drupal 2005.)
Payment-moduuli	Moduuli joka mahdollistaa tuen muille moduuleille eri verkkomaksupalveluita varten. (Payment 2015.)
Paytrail	Verkkomaksupalvelu, joka mahdollistaa yleisimmät Suomessa käytetyt maksutavat.
PHP	Suosittu yleiskäyttöinen komentosarjakieli, joka on erityisesti tarkoitettu verkkopalveluiden kehittämiseen. (PHP 2015.)
REST	HTTP-protokollaan perustuva rajapinta-arkkitehtuuri. (IBM 2015.)
Sandbox	Drupal.org-palvelussa yksityinen versionhallinta. (Drupal 2014.)

## Lähteet

3scale 2011. What is an API? Your guide to the Internet Business (R)evolution. Luettavissa: <http://www.3scale.net/wp-content/uploads/2012/06/What-is-an-API-1.0.pdf>. Luettu: 21.5.2015.

Curl 2015a. curl and libcurl. Luettavissa: <http://curl.haxx.se>. Luettu: 29.6.2015.

Curl 2015b. PHP: cURL – Manual. Luettavissa: <http://php.net/manual/en/book.curl.php>. Luettu: 29.6.2015.

Drupal 2005. General concepts | Drupal.org. <https://www.drupal.org/node/19828>. Luettu: 2.7.2015.

Drupal 2011a. drupal\_http\_request | common.inc | Drupal 7 | Drupal API. [https://api.drupal.org/api/drupal/includes%21common.inc/function/drupal\\_http\\_request/7](https://api.drupal.org/api/drupal/includes%21common.inc/function/drupal_http_request/7). Luettu: 21.4.2015.

Drupal 2011b. Creating modules - a tutorial: Drupal 7.x | Drupal.org. <https://www.drupal.org/node/1074360>. Luettu: 23.4.2015.

Drupal 2011c. Form API Reference | Drupal 7 | Drupal API. [https://api.drupal.org/api/drupal/developer!topics!forms\\_api\\_reference.html/7](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7). Luettu: 23.4.2015.

Drupal 2011d. variable\_set | bootstrap.inc | Drupal 7 | Drupal API. [https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/variable\\_get/7](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/variable_get/7). Luettu: 26.4.2015.

Drupal 2011e. variable\_get | bootstrap.inc | Drupal 7 | Drupal API. [https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/variable\\_set/7](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/variable_set/7). Luettu: 26.4.2015.

Drupal 2012. Getting started | Drupal.org. <https://www.drupal.org/node/1074362>. Luettu: 29.3.2015.

Drupal 2013. Telling Drupal about your module | Drupal.org. <https://www.drupal.org/node/1075072>. Luettu: 29.3.2015.

Drupal 2014. Creating a sandbox (experimental) project | Drupal.org.  
<https://www.drupal.org/node/1011196>. Luettu 21.6.2015.

Drupal 2015a. Choosing a Drupal version | Drupal.org.  
<https://www.drupal.org/documentation/version-info>. Luettu: 28.3.2015.

Drupal 2015b. Coding standards | Drupal.org. <https://www.drupal.org/coding-standards>.  
Luettu: 1.4.2015.

Drupal Commerce 2015. Core features | Drupal Commerce.  
<https://drupalcommerce.org/features>. Luettu: 21.6.2015.

IBM 2015. RESTful Web services: The basics.  
<http://www.ibm.com/developerworks/library/ws-restful>. Luettu: 28.3.2015.

Mozilla 2015. Ajax | MDN. <https://developer.mozilla.org/en-US/docs/AJAX>. Luettu  
21.6.2015.

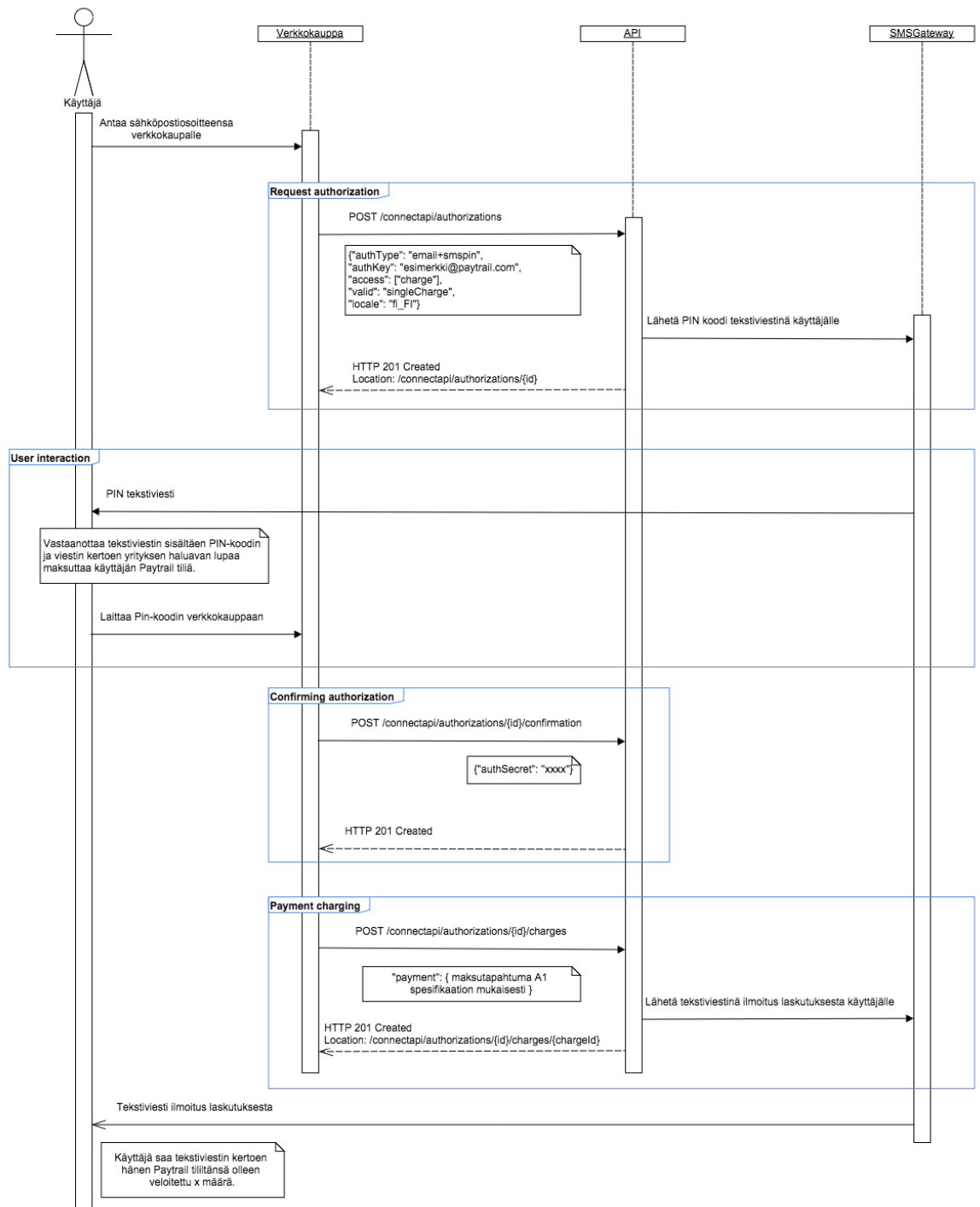
PHP 2015. PHP: Hypertext Preprocessor. <http://php.net>. Luettu: 21.5.2015.

Paytrail 2014. Paytrail – Integration guide. <http://docs.paytrail.com/en/index-all.html>. Luet-  
tu: 28.3.2015.

Payment 2015. Payment | Drupal.org. <https://www.drupal.org/project/payment>. Luettu:  
29.6.2015.

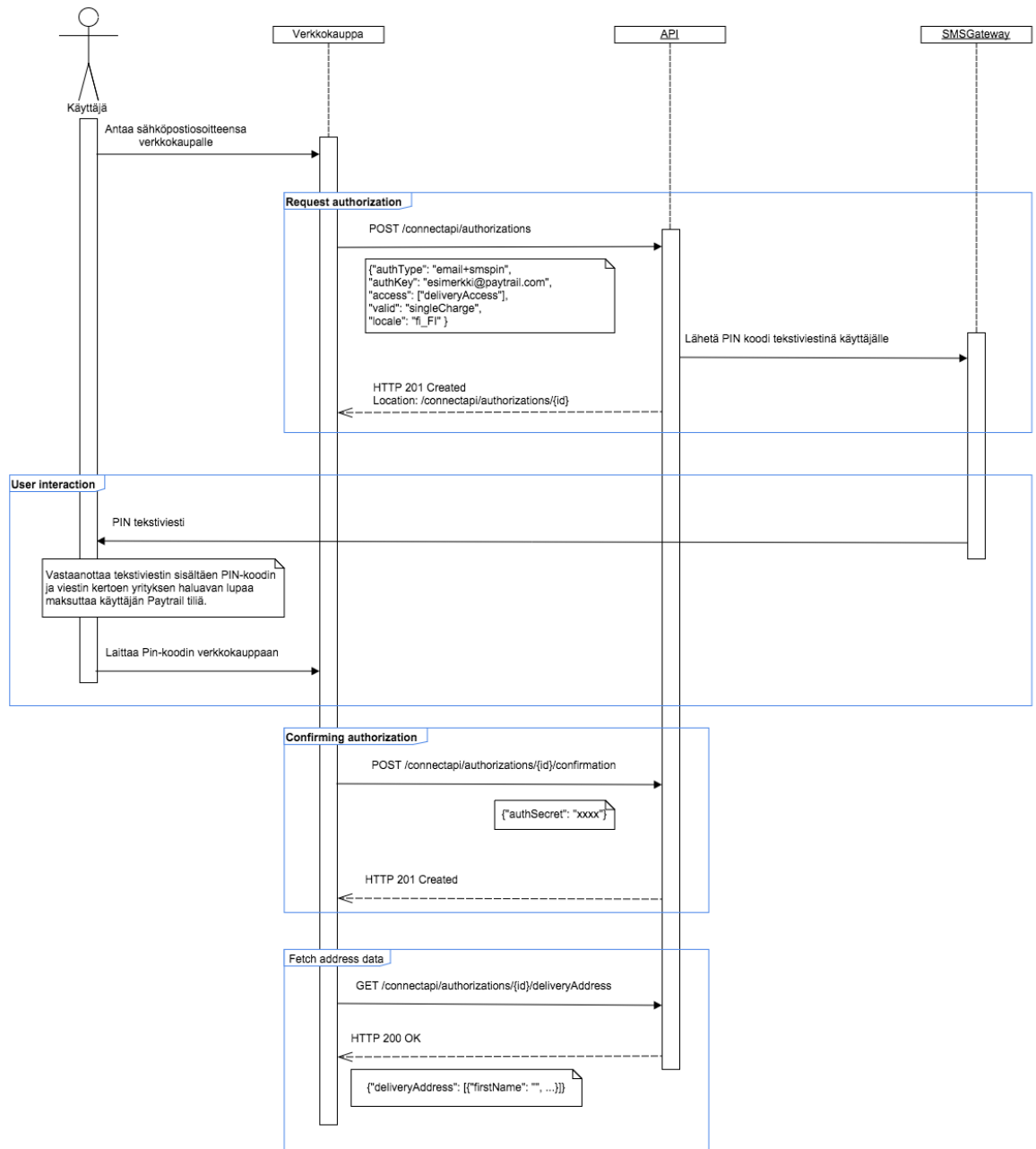
# Liitteet

## Liite 1. Connect API kuvaus 1

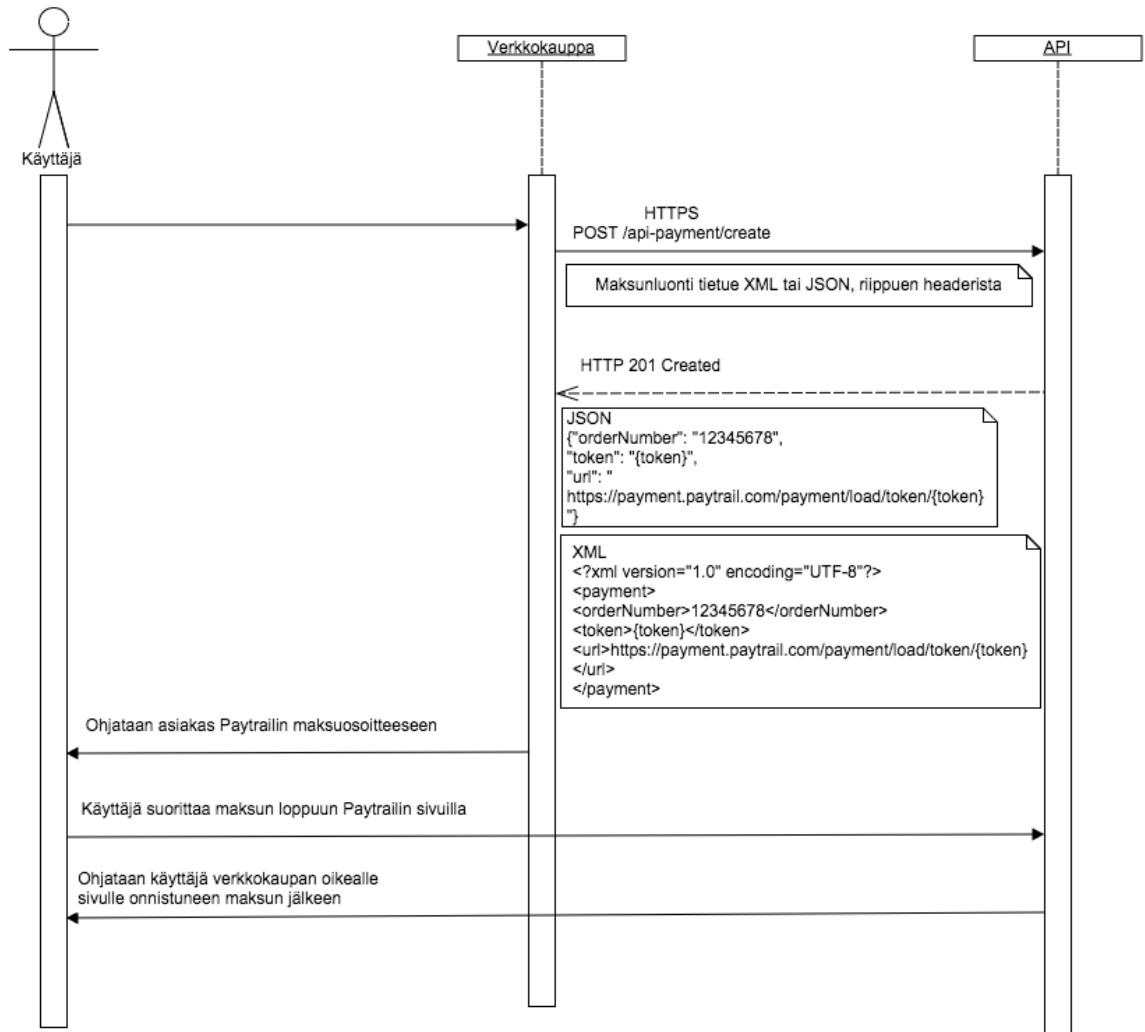




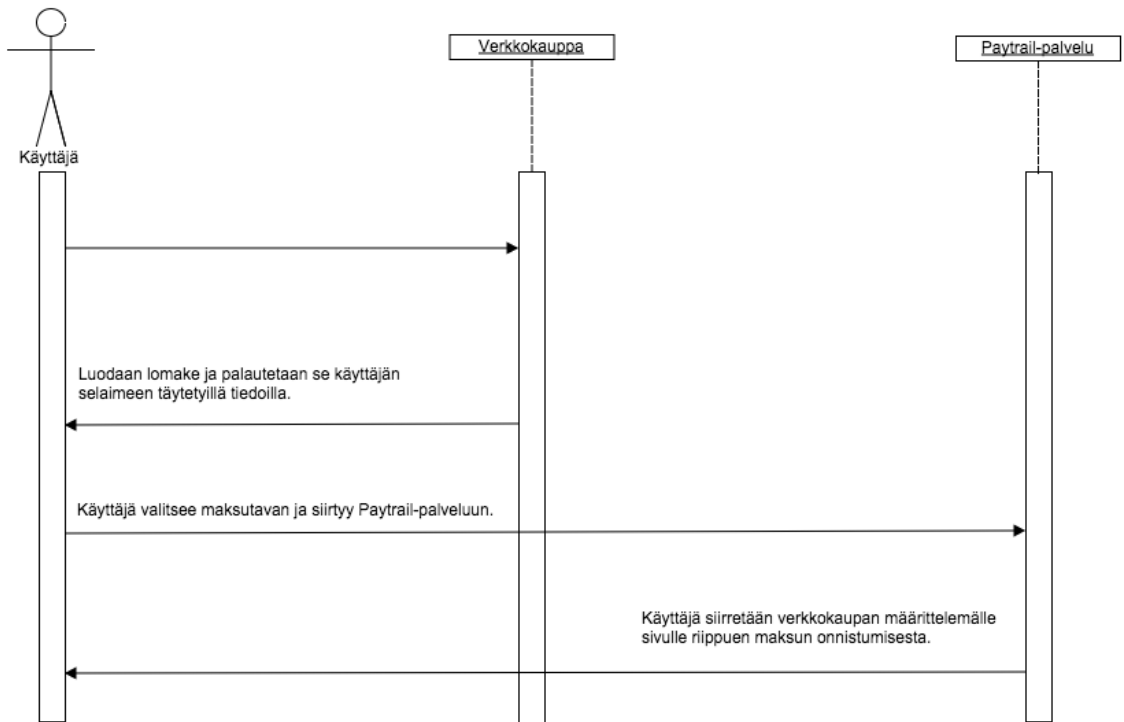
## Liite 2. Connect API kuvaus 2



### Liite 3. REST Interface kuvaus



## Liite 4. Form interface kuvaus



## Liite 5. Testauskaavio

Testauskaavio		Tulos*	Tulos:
Metodi	Maksuversio	Tehävä	
Connect API	A1	Maksun suorittaminen onnistuu yhdellä tuotteella	
Connect API	A1	Maksun suorittaminen onnistuu monella tuotteella	
Connect API	A1	Käyttäjän tietojen puuttuessa ohjelma osaa ilmoittaa virheestä ja palauttaa asianmukaisesti maksun valmistelussa	Toinni
Connect API	A1	Käyttäjän tietojen puuttuessa ohjelma osaa ilmoittaa virheestä ja palauttaa asianmukaisesti maksun lähetyksessä	Toinni, mutta virheellä
Connect API	A1	Verkkosivu antaa oikean linkin maksun suorittamiseen Paytrailin verkkosivulla	Epäonnistui
Connect API	A1	Maksun tiedot ovat oikeassa muodossa Paytrail-palvelussa	
Connect API	A1	Pyydettäessä käyttäjän toimitustietoja, ne palautetaan	
Connect API	A1	Käyttäjän toimitustietojen haku toimii yhdessä makuttamisen kanssa	
Connect API	A1	Maksun onnistuessa siirrytään onnistumisen sivulle	
Connect API	A1	Maksun onnistuessa siirrytään epäonnistumisen sivulle	
Connect API	A1	Maksun onnistuessa siirrytään kiinteällä hinnalla.	
REST Interface	S1	Maksun tiedot ovat oikeassa muodossa Paytrail-palvelussa	
REST Interface	S1	Maksun onnistuessa siirrytään onnistumisen sivulle	
REST Interface	S1	Maksun onnistuessa siirrytään epäonnistumisen sivulle	
REST Interface	S1	Maksun onnistuessa siirrytään kiinteällä hinnalla.	
REST Interface	S1	Maksun onnistuessa siirrytään onnistumisen sivulle	
REST Interface	S1	Maksun onnistuessa siirrytään epäonnistumisen sivulle	
REST Interface	S1	Maksun onnistuessa siirrytään kiinteällä hinnalla.	
REST Interface	S1	Maksun onnistuessa siirrytään onnistumisen sivulle	
REST Interface	S1	Maksun onnistuessa siirrytään epäonnistumisen sivulle	
REST Interface	S1	Maksun onnistuessa siirrytään kiinteällä hinnalla.	
REST Interface	S1	Maksun onnistuessa siirrytään onnistumisen sivulle	
REST Interface	S1	Maksun onnistuessa siirrytään epäonnistumisen sivulle	
Form Interface	E1	Maksun suorittaminen onnistuu yhdellä tuotteella	
Form Interface	E1	Maksun suorittaminen onnistuu monella tuotteella	
Form Interface	E1	Käyttäjän tietojen puuttuessa ohjelma osaa ilmoittaa virheestä ja palauttaa asianmukaisesti	
Form Interface	E1	Maksun tiedot ovat oikean muotoiset Paytrail-palvelussa	
Form Interface	E1	Maksun onnistuessa siirrytään onnistumisen sivulle	
Form Interface	E1	Maksun onnistuessa siirrytään epäonnistumisen sivulle	
Form Interface	E1	Maksutavan valinnan upotus toimii	