

Suomi.fi kehittäjille

Hyvät käytännöt ohjelmointirajapintojen suunnitteluun julkisessa hallinnossa

Liite 1

12.12.2024



Liite 1. Stripe kansainvälisenä esimerkkinä onnistuneesta API-ekosysteemeistä

Usein hyvänä kansainvälisenä esimerkkinä onnistuneesta ohjelmointirajapintojen avulla rakennetusta ekosysteemistä mainitaan Stripe. Vuonna 2009 perustettu Stripe (<https://stripe.com/>) toimii ohjelmisto palveluna -yrityksenä (engl. Software as a Service, SaaS). Sen liikeideana on tarjota digitaalisia palveluja, muun muassa luottokorttimaksujen vastaanottoa, ohjelmistoyrityksille. Käytännössä se tarjoaa helppokäyttöisiä ohjelmointirajapintoja, jonka avulla voi hoitaa esimerkiksi maksamisen. Maksamisen monimutkaisuus on piilotettu helppokäyttöisten ohjelmointirajapintojen taakse. Ohjelmistokehittäjän ei tarvitse miettiä maksamisen yksityiskohtia. Riittää, että hän hyödyntää yksinkertaista ohjelmointirajapintaa. Stripe veloittaa sen välittämän maksun kokonaissummasta 1,5 - 2,5 prosentin palkkion ja 0,25 euron maksun kerralta. Yhtiön ohjelmointirajapinnoista kannattaa ottaa esimerkkiä.

Miljardien arvoisen yrityksen menestyksen salaisuutena pidetään luotettavia, erittäin helppokäyttöisiä ja huolellisesti dokumentoituja ohjelmointirajapintoja, joiden käyttö on ratkaisevasti yksinkertaisempaa kuin monen kilpailijan ohjelmointirajapintojen. (Bu 2020). Stripe on myös investoinut vahvasti kehittäjäkokemukseen, eli siihen, miten ohjelmistokehittäjät kokevat ja käyttävät Stripe-tuotteita.

Stripen ohjelmointirajapinnat ja niiden dokumentointi

Stripe kuvaa menestystään hyvin analyttisesti ja yrityksen perustajat kertovat sekä yrityksen omassa blogissa että erilaisissa haastatteluissa (Armstrong 2018) mielellään siitä, kuinka todella yksinkertainen, seitsemän koodirivin mittainen luottokorttimaksun vastaanoton ohjelmointirajapinta oli olennainen heidän menestykselleen.

```
$ curl https://api.stripe.com/v1/charges \
-u vtUQeOtUnYr7PGCLQ96U14zqpDU04sOE: \
-d amount=400 \
-d currency=usd \
-d "description=Charge for site@stripe.com" \
-d "card[number]=4242424242424242" \
-d "card[exp_month]=12" \
-d "card[exp_year]=2012" \
-d "card[cvc]=123"
```

Create a new charge using curl

An API that gets out of your way

It's so easy, we've embedded a bunch of examples right here. Copy some of these requests into your terminal and check out what happens.

With wrappers in Ruby, PHP, Python and more, you can get started in minutes. [Learn more](#)

Stripe loi yksinkertaiset ohjelmointirajapinnat, jotka piilottavat maksunvälittämisen monimutkaisuuden. Yhtiön ohjelmointirajapintojen käyttö ei edellytä pankki- tai luottokorttimaksujärjestelmän monimutkaisuuden ymmärtämistä. Ohjelmointirajapintojen käyttö onkin kasvanut erittäin nopeasti.

Yrityksen ohjelmointirajapinnat ja näiden dokumentaatio (Stripe 2024a) ovat olennainen osa yrityksen markkinointia. Stripe.com-verkkosivuston etusivulla näytetään suoraan, kuinka palvelua kutsutaan omasta lähdekoodista ja kävijöitä rohkaistaan voimakkaasti tutustumaan yrityksen tarjoamien palvelujen dokumentaatioon.



DTY / Honkanen Mika (DVV)

```
1  const stripe = require('stripe')('sk_test_BQokikJOvBi2H146...
2
3  await stripe.paymentIntents.create({
4    amount: 2000,
5    currency: 'usd'
6  });
```

NORMAL server.js 100% 6/6 ln : 4

```
$ node server.js && stripe listen
> Ready! Waiting for requests...
2023-11-04 18:35:41 [200] payment_intent.created
2023-11-04 18:35:41 [200] charge.succeeded
2023-11-04 18:35:42 [200] payment_intent.succeeded
```

Ohjelmointirajapinnat on dokumentoitu tavalla, joka rohkaisee kokeiluihin. Onnistuneet kokeilut johtavat tuotantokäyttöön. Quickstart-osiosta löytyy valmiita esimerkkejä toteutuksesta useilla ohjelmointikielillä. Lisäksi yrityksen ohjelmointirajapintoja voi käyttää verkkosivuilta löytyvillä testiluottokorttien numeroilla, jotta maksuliikennettä voi simuloida ilman oikeaa luottokorttia.

Ohjelmointirajapintojen toteutus

Stripe hyödyntää web-pohjaisia rajapintoja, jotka perustuvat HTTP-protokollaan ja JSON-formaattiin. Yhtiö käyttää RESTful-arkkitehtuuria, mikä tarkoittaa, että ohjelmointirajapinta muodostuu resursseista, jotka ovat tunnistettavissa yksilöllisten URI-osoitteiden avulla ja niihin voidaan kohdistaa toimintoja HTTP-metodien GET, POST, PUT tai DELETE avulla.

Uuden maksutapahtuman luominen Stripe-ohjelmointirajapinnan kautta tapahtuu lähettämällä HTTP-pyyntö, joka palauttaa vastauksena maksutapahtuman tiedot JSON-muodossa.

```
POST https://api.stripe.com/v1/charges { "amount": 2000, "currency": "eur", "source": "tok_visa", "description": "My first payment" }
```



DTY / Honkanen Mika (DVV)

Ohjelmointirajapinnan toiminnan voi arvata ohjelmointirajapintakutsua tarkemmin tutkimalla – 2000 euron maksu otsikolla “My first payment”.

Maksunvälityksen vastaus on monisanaisempi, mutta sekin on varsin helppolukuinen RESTful-ohjelmointirajapinnan mukaisesti. Tällä kertaa maksu oli onnistunut.

```
{ "id": "ch_1JQwXn2eZvKYlo2C8fP4nq0B", "object": "charge", "amount": 2000, "created": 1636477762, "currency": "eur", "description": "My first payment", "paid": true, "status": "succeeded", "source": { "id": "card_1JQwXn2eZvKYlo2CQgXGZyZ4", "object": "card", "brand": "Visa", "last4": "4242", "exp_month": 4, "exp_year": 2024 } }
```

Ohjelmointirajapintojen kutsuminen on varsin suoraviivaista, mutta yritys on luonut eniten käytetyille selain- ja mobiilialustoille valmiita ohjelmistokehityspaketteja, joilla ohjelmointirajapintojen käyttö on tehty vielä astetta suoraviivaisemmaksi.

Neljä asiaa tekevät Stripen ohjelmointirajapinnoista loistavia:

1. Hyödyntämisen yksinkertaisuus (helppoa, nopeaa)
2. Erinomainen dokumentaatio (erilaisille osajille)
3. Tuottavuus (yhdeällä ohjelmointirajapinnan kutsulla voi tehdä halutun asian)
4. Yhtiö hyödyntää itse omia ohjelmointirajapintojaan.

Yhtiön ohjelmointirajapintojen suunnittelu on vaatinut paljon palvelumuotoilua ja ajattelua sen suhteen, miten esimerkiksi RESTful-periaatteita sovelletaan. Loistavia rajapinnoista tekee niiden yksityiskohdat.

Vuosien varrella ohjelmointirajapintojen toteutus on muuttunut toistuvasti ja osa muutoksista teki järjestelmästä vaikeamman käyttää. Yksinkertaisen, mutta ilmaisuvoimaisen ohjelmointirajapinnan määrittely voi olla vaikeaa jopa tähän keskittyneelle organisaatiolle.

Kehittäjäkokemus

Kehittäjäkokemus on keskeinen tekijä ohjelmointirajapintojen menestyksessä. Se vaikuttaa siihen, miten helposti ja mielellään kehittäjät ottavat käyttöön yrityksen tarjoamat ratkaisut. Stripen dokumentaatiosta ja kehittäjäkokemuksesta huomaa, että kehittäjien kokemukseen on panostettu. Yhtiön internetsivut ovat interaktiivisia ja rohkaisevat käyttämään sekä tekstuaalista kuvausta että tutustumaan vaadittavaan lähdekoodiin ohjaamalla katseen juuri sinne minne pitääkin.

Kehittäjäkokemus näkyy erityisesti rajapintojen suoraviivaisuudessa, dokumentaation laadussa ja kattavuudessa sekä valmiiden ratkaisujen saatavuudessa moniin käyttötarkoituksiin ja ohjelmointiympäristöihin. Niin sanottu hiekkalaatikko, eli kehittäjien testialusta, on palvelussa kattavasti käytettävissä ja sovelluksia on helppo testata ensin hiekkalaatikossa, jossa ei tehdä virheitä oikeilla luottokorttitapahtumilla.

Hyvä esimerkki kehittäjäkokemuksesta on ns. Go-live tarkastuslista (Stripe 2024b). Luottokorttimaksujen onnistunut vastaanotto on Stripen asiakkaille elintärkeää ja sivustolta löytyy selkeä tarkastuslista auttamaan lähes valmiin maksunvastaanoton viemisessä tuotantoon.



DTY / Honkanen Mika (DVV)

Stripe on saanut tunnustusta kehittäjäkokemuksestaan, voittaen palkintoja ja saaden positiivista asiakaspalautetta. Yrityksen kehittäjille suunnattu dokumentaatio ja resurssit ovat saatavilla yhtiön internetsivustolla.