

Tiedonhallinta ja välitys -sovellus

Case: Adapter

LAHDEN
AMMATTIKORKEAKOULU
Tekniikan ala
Tietotekniikan koulutusohjelma
Ohjelmistotekniikka
Opinnäytetyö
Kevät 2017
Tomi Tanhuansuu

Lahden ammattikorkeakoulu
Tietotekniikan koulutusohjelma

TANHUANSUU, TOMI:

Tiedonhallinta ja välitys -sovellus
Case: Adapter

Ohjelmistotekniikan suuntautumisvaihtoehdon opinnäytetyö, 42 sivua

Kevät 2017

TIIVISTELMÄ

Opinnäytetyössä perehdytään Symfony PHP -kehysalustaan ja sen komponentteihin ja bundleihin sekä jQueryyn ja Material Design Lite -käyttöliittymäkomponenttikirjastoon. Perehtyminen on tekniikoiden laajuuden vuoksi varsin pintapuolinen. Tarkoituksena oli kuitenkin perehtyä kaikkein oleellisimpiin ominaisuuksiin. Opinnäytetyössä tekniikat esitellään ensin teoreettisesti, minkä jälkeen esitellään toteutettu sovellus käyttäen näitä tekniikoita.

Tavoitteena oli toteuttaa Mediatalo ESA Oy:lle ja Lahden Nuorisopalveluille Adapter-niminen hallintasoftware, jolla pystytään tuottamaan ja hakemaan päijäthämäläisille nuorille suunnattua sisältöä. Lisäksi Adapterilla pyritään mahdollistamaan nuorisokortin sähköistäminen. Opinnäytetyön tarkoituksena oli tutkia kuinka kyseinen sovellus toteutetaan annetuilla tekniikoilla.

Tuloksena syntyi hallintasoftware, jolla voidaan luoda monia itsenäisiä pienempiä sisäisiä sovelluksia. Sisäiset sovellukset ovat perusrungoltaan identtisiä, mutta jokaisella sovelluksella on omat käyttäjänsä, sisältönsä ja rajapintansa sekä liitännäisensä. Rajapintojen käyttäminen edellyttää tunnistautumista sovelluskohtaisilla oauth-avaimilla. Adapterin sisäisten sovellusten perusrunko on rakennettu ensisijaisesti täyttämään Nobiili-nimisen mobiilisovelluksen tarpeet.

Asiasanat: Symfony, PHP, JQuery, CSS, MDL

Lahti University of Applied Sciences
Degree Programme in information Technology

TANHUANSUU, TOMI: Information management and sharing
software
Case: Adapter

Bachelor's Thesis in Software Engineering, 42 pages

Spring 2017

ABSTRACT

This thesis is focused on the Symfony PHP framework and its components and bundles. It also deals with the jQuery and Material Design Lite user interface component library. Because of the wide scope of these techniques, the purpose is only to highlight the most important aspects. The techniques are first explained theoretically and after that implemented in practice.

The goal was to implement a management application named Adapter for Mediatalo ESA Oy and Lahti Youth Services. The idea was to produce and fetch content for young people in Päijät-Häme. Another aim was to enable electrification of the youth card. The purpose at the thesis was to research how to develop such an application with given techniques.

The result was software, which can create smaller internal applications. These internal applications are basically identical, but every application has its own users, content, interface and plugins. Using the interfaces requires application specific authentication with oauth keys. The base of the internal application is intended to fulfill the needs of the Nobiili -mobile application.

Key words: Symfony, PHP, JQuery, CSS, MDL

SISÄLLYS

1	JOHDANTO	1
2	ADAPTER HALLINTASOVELLUKSEN PÄÄTEKNIIKAT	2
2.1	Symfony	2
2.1.1	Symfonyn asentaminen	3
2.1.2	Symfony komponentit	5
2.1.3	Symfony Bundles	6
2.1.4	Doctrine	7
2.1.5	Twig	9
2.2	JQuery	10
2.3	Material Design Lite	12
2.4	Web API	14
3	ADAPTER HALLINTASOVELLUS	17
3.1	Käytetyt teknologiat ja komponentit	18
3.2	Käyttäjähallinnointi	20
3.2.1	Kutsuminen ja rekisteröityminen	21
3.2.2	Kirjautuminen ja salasanan palauttaminen	23
3.2.3	Käyttäjän profiilisivu	25
3.2.4	Käyttäjäoikeudet	26
3.3	Adapterin sisäiset sovellukset	26
3.3.1	Sisäisen sovelluksen luominen	27
3.3.2	Sisäiset sovellukset perusrunko	27
3.3.3	Sisäisten sovellusten käyttäjäoikeudet	31
3.4	Adapterin sisäisten sovellusten liitännäiset	32
3.4.1	ESSFEED -plugin	32
3.4.2	Nuokku -plugin	33
3.4.3	Nuoli -plugin	36
3.4.4	TapKal -plugin	37
4	YHTEENVETO	39
	LÄHTEET	41

1 JOHDANTO

Mobiililaitteiden kuten puhelimien ja tablettien räjähdymäisesti nousseen suosion vuoksi verkkosovellus kehittämisestä on tullut yksi suosituimmista ja tärkeimmistä osa-alueista ohjelmistokehityksessä. Myös monet yritykset ovat huomanneet sen ja pyrkineet hyödyntämään sitä liiketoiminnassaan.

Mediatalo ESA Oy on Päijät-Hämeen alueella toimiva mediayhtiö, joka on erikoistunut painettujen ja digitaalisten sisältöjen sekä mediapalveluiden tuottamiseen. Mediatalo ESA Oy:n historia ulottuu 1900 -luvun alkupuolelle, ja se työllistää yhteensä noin 600 - 700 henkilöä. Vuonna 2015 sen liikevaihto oli noin 26,3 miljoonaa euroa. Mediatalo ESA Oy siirtyi 2016 loppuvuodesta osaksi Keski-suomalainen -osakeyhtiötä.

Tämän opinnäytetyön tavoitteena oli toteuttaa Mediatalo ESA Oy:lle ja Lahden nuorisopalveluille Adapter -hallintasovellus. Sovelluksen avulla on tarkoitus pystyä tuottamaan ja hakemaan päijäthämäläisille nuorille suunnattua sisältöä. Lisäksi tavoitteena oli mahdollistaa nuorisokortin sähköistäminen. Opinnäytetyön alitavoitteena oli toteuttaa hallintasovellus niin, että sovelluksen sisään pystyy tekemään rajattoman määrän toisistaan riippumattomia sovelluksia omine toiminnallisuuksineen. Toinen alitavoite oli toteuttaa jokainen toiminnallisuus omana liitännäisenä.

Ongelmaksi jää tutkia, miten kyseinen hallintasovellus toteutetaan käyttäen Symfonia, jQuerya ja MDL -käyttöliittymäkomponenttikirjastoa. Opinnäytetyön teoreettinen osuus rajoittuu näiden tekniikoiden pintapuoliseen esittelyyn tuoden esille tärkeimpiä ominaisuuksia. Käytännön osuudessa puolestaan esitellään näillä tekniikoilla syntyneitä lopputulosta.

Opinnäytetyön tarkoituksena ei ole vertailla tai arvioida, mikä tekniikka soveltuu parhaiten verkkosovelluksen kehittämiseen vaan esitellä käytetyt tekniikat ja niillä syntynyt toteutus. Opinnäytetyö on toteutettu osana Lahden ammattikorkeakoulun työharjoitteluun liittyviä opintoja Korpimedia Oy:ssä yhdessä Mediatalo ESA:n ja Lahden nuorisopalveluiden edustajien kanssa.

2 ADAPTER HALLINTASOVELLUKSEN PÄÄTEKNIIKAT

2.1 Symfony

Symfony on PHP -komponenteista koostuva avoimen lähdekoodin ohjelmistokehys, joka on yksi johtavista ja käytetyimmistä PHP -ohjelmistokehyksistä. Se on suunnattu erityisesti verkkosivujen ja verkkosovellusten kehittämiseen. Sillä on yli 2000 contributoria eli henkilöä, jotka ovat edesauttaneet sen kehityksessä, yli 300 000 käyttäjää ja yli 29 000 000 kuukausittaista latauskertaa (Symfony 2017g). Symfony käyttää viimeisintä verkkoteknologiaa, ja se soveltuu yhtä hyvin tuotantoon tarkoitettujen kuin harrastusmielessä tuotettujen sovellusten kehittämiseen. (O'Dell 2017).

Symfony usein mielletään MVC eli Malli-Näkymä-Käsittelijä -ohjelmistokehykseksi, vaikka käytännössä se tarjoaa työkalut vain käsittelijöille ja näkymille. Symfonyssa on oletuksellisesti tiivis integraatio parhaiten tunnetuille ORM:ille eli relationaalisille oliokartoituksille, kuten Doctrine2:lle ja Propelille. Integraatio on kuitenkin täysin optionaalinen ja jättää kehittäjän päätettäväksi, käytetäänkö sitä projektissa vai ei. Symfonyn peruseriaatteet ovat keskittyneet HTTP -määrittelyn ympärille, ja näin ollen kyse on enemmän HTTP- kuin MVC -ohjelmistokehyksestä. (Potencier 2011.)

Muista ohjelmistokehyksistä poiketen Symfony ei rajoita tai aseta oletuksia, miten sovellusten tulisi toimia, vaan se on kustomoitavissa sovelluskohtaisten tarpeiden mukaan. Toisistaan riippumattomat komponentit ja oliosuuntautunut arkkitehtuuri mahdollistavat minkä tahansa osan laajentamisen, kustomoimisen, korvaamisen tai poistamisen ohjelmistokehyksestä. Symfonyyn on mahdollista integroida myös laaja joukko erilaisia tietokantatyyppejä, kuten esimerkiksi SQL, NoSQL, Avain-arvo tai kaaviotietokantoja. (O'Dell 2017.)

Symfony on suorituskyvyltään hyvä, mikä johtuu siitä, että se ei kuormita kehitettävää sovellusta tarpeettomasti vaan tarvittava tieto ladataan ja käsitellään vain tarpeen vaatiessa. Symfonystä löytyy myös kehittynyt riippuvuusinjektiosäiliö ja proxy -objekteja suorituskyvyn parantamiseksi. Lisäksi Symfony hyödyntää HTTP:n välimuistin standardeja ja tukee ESI eli Edge Side Includes -tekniikkaa, joka mahdollistaa edistyneemmän sivujen tallentamisen välimuistiin. Nopeaa suorituskykyä tukevat myös esipakatut front-end mallit ja esipakattu riippuvuusinjektiosäiliö sekä HTTP -ytimen arkkitehtuuri, joka mahdollistaa nopeat välimuistivastaukset. (O'Dell 2017.)

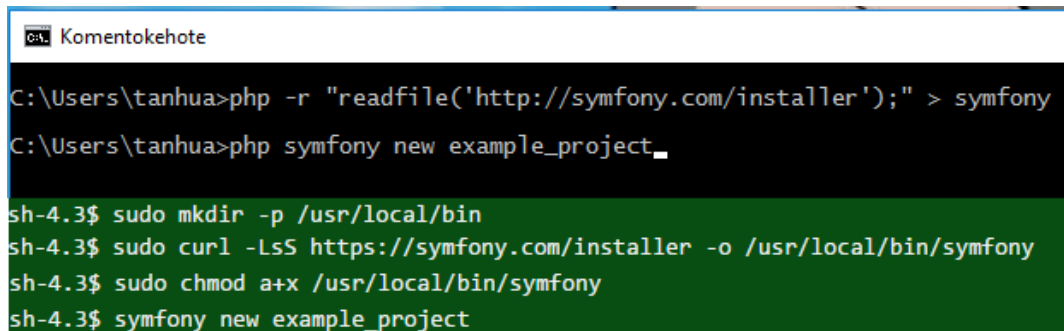
Joustavuuden ja suorituskyvyn lisäksi Symfonystä löytyy myös muita vahvuuksia. Symfonyn avulla sovelluskehittämisestä tulee nopeampaa muun muassa esirakennettujen komponenttien ansiosta. Sen sijaan, että kehittäjä joutuu käyttämään aikaansa yleisten komponenttien, kuten lomakkeiden validointiin tai istuntojen hallintaan, antaa Symfony mahdollisuuden keskittyä suoraan sovelluskohtaisten ominaisuuksien kehittämiseen. Symfonystä löytyy myös automaattinen suojaus nykypäivän tavallisimpiin verkkohaavoittuvuuksiin, kuten SQL -injektioon, cross-site skriptaukseen ja istuntokaappauksiin. (O'Dell 2017.)

Julkaisupäivästä alkaen Symfony on osoittanut olevansa luotettava ja laadukkaalla koodilla toteutettu ohjelmistokehys kustomoitujen sovellusten kehittämiseen. Pääversioille Symfony tarjoaa noin 3 - 4 vuoden ajan täyden tuen mukaan lukien virheiden ja haavoittuvuuksien korjaamiset. Lisäksi Symfony pyrkii säilyttämään yhteensopivuuden kolmannen osapuolen kirjastoihin päivitysten yhteydessä. (O'Dell 2017.)

2.1.1 Symfonyn asentaminen

Symfonyn asentaminen on helppoa ja vaivatonta niin Windows ja MacOS kuin Linux -käyttöjärjestelmillä. Symfonyn voi asentaa muutamalla eri tavalla, joista käytetyimpiä tapoja lienevät Symfonyn tarjoaman Symfony Installerin tai Composer -sovelluksen käyttäminen. Symfony Installer on sovellus Symfony -sovellusten luomiseen ja Composer on sovellustason

pakettienhallintaohjelma. Kuvio 1 on kuvankaappaus Symfony Installerin asentamiseen vaadittavista komennoista, joissa mustalla pohjalla olevat komennot havainnollistavat Windows -ympäristön komentoja ja vihreällä pohjalla olevat puolestaan Linux ja MacOS -ympäristöjä. Komento "php symfony new example_project" luo uuden "example_project" -nimisen Symfony sovelluksen ja komennot ennen tätä asentavat Symfony Installerin. (Symfony 2017c.)

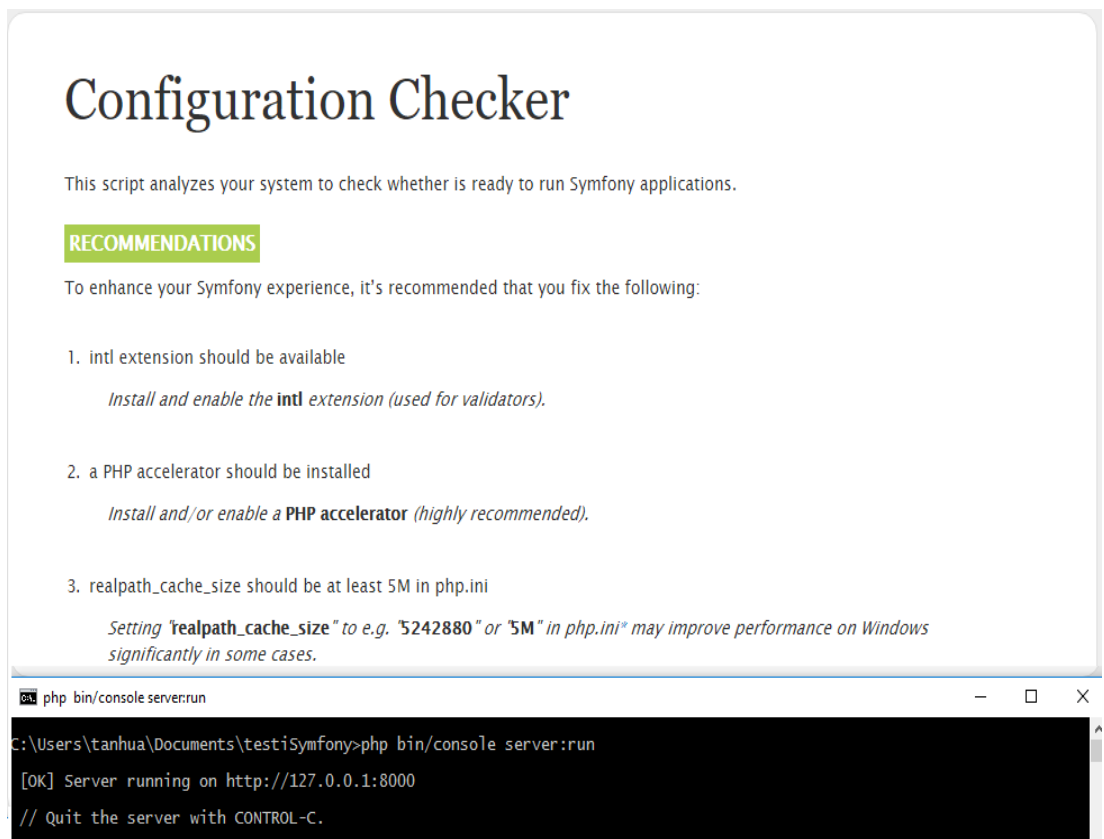


```

C:\Users\tanhua>php -r "readfile('http://symfony.com/installer');" > symfony
C:\Users\tanhua>php symfony new example_project_

sh-4.3$ sudo mkdir -p /usr/local/bin
sh-4.3$ sudo curl -Ls https://symfony.com/installer -o /usr/local/bin/symfony
sh-4.3$ sudo chmod a+x /usr/local/bin/symfony
sh-4.3$ symfony new example_project
  
```

KUVIO 1. Symfony Installerin asentaminen



Configuration Checker

This script analyzes your system to check whether is ready to run Symfony applications.

RECOMMENDATIONS

To enhance your Symfony experience, it's recommended that you fix the following:

- intl extension should be available
Install and enable the intl extension (used for validators).
- a PHP accelerator should be installed
Install and/or enable a PHP accelerator (highly recommended).
- realpath_cache_size should be at least 5M in php.ini
Setting 'realpath_cache_size' to e.g. '5242880' or '5M' in php.ini may improve performance on Windows significantly in some cases.*

```

php bin/console server:run
C:\Users\tanhua\Documents\testiSymfony>php bin/console server:run
[OK] Server running on http://127.0.0.1:8000
// Quit the server with CONTROL-C.
  
```

KUVIO 2. Symfony:n käynnistäminen ja tarkastaminen

Jos Symfony -sovellusta ei ole pystytetty Apachen HTTP:lle tai muulle vastaavanlaiselle palvelimelle, voidaan sovellus tarvittaessa käynnistää kirjoittamalla komentokehoitteeseen komento "php bin/console server:run", kuten kuvion 2 alareunassa näkyvässä komentokehoitteessa on tehty. Tämä käynnistää sovelluksen paikallisesti portissa 8000 (Symfony 2017c). Symfony -sovelluksen suorittaminen ei aseta ympäristölleen kovin montaa teknistä vaatimusta. Minimivaatimuksia Symfony -sovelluksen suorittamiseen ovat PHP 5.5.9 tai sitä uudempi versio, JSON:n ja ctype:n täytyy olla käytettävissä sekä php.ini -tiedostossa täytyy olla määritettyinä date.timezone -asetus. Symfonylla on myös kymmenkunta valinnaista asetusta ja joitakin suosituksia, joista kolme on nähtävissä kuviossa 2. Vaatimukset ja suositukset ovat nähtävillä sovelluksen käynnistämisen jälkeen sovelluksen URL:n config.php -polussa eli esimerkiksi "localhost:8000/config.php" tai komentokehoitteesta komennolla "php bin/symfony_requirements". (Symfony 2017e.)

2.1.2 Symfony komponentit

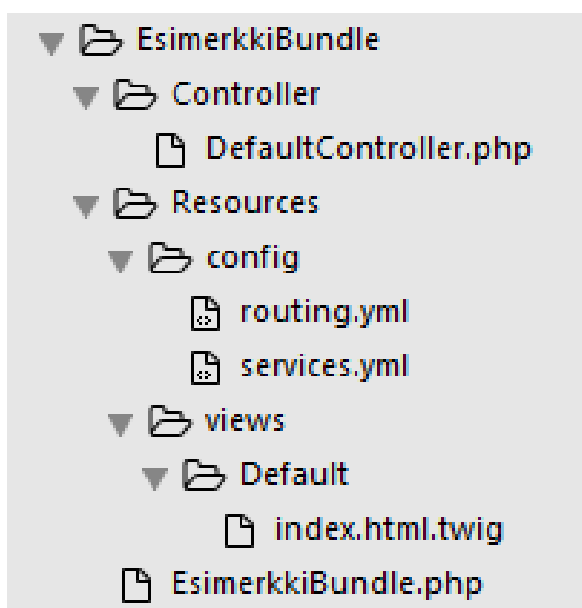
Symfony komponentit ovat itsenäisiä ja uudelleen käytettävissä olevia komponentteja, joita on käytetty myös useissa suosituissa PHP -sovelluksissa, kuten Drupal, phpBB ja eZ Publish (Symfony 2017g). Näiden komponenttien tarkoitus on tarjota tavallisimpia ominaisuuksia ja työkaluja, joita verkkosovellusten kehittämiseen tarvitaan, sekä ratkaista niihin liittyviä yleisimpiä ongelmia. Lisäksi näiden komponenttien koodi on laadukasta ja niihin on lisätty turvallisuutta edistäviä ominaisuuksia. (Potencier 2011.)

Itsenäisillä komponenteilla tarkoitetaan sitä, että yksittäisiä komponentteja voidaan käyttää eri PHP -sovelluksissa ilman, että koko sovellusta tarvitsee rakentaa Symfonyn varaan. Tämä tarkoittaa myös sitä, että Symfonyn standardista ohjelmistokehyksestä voidaan tarvittaessa poistaa yksittäisiä tarpeettomia komponentteja. Komponentteja on myös helppo laajentaa omalla koodilla, koska ne käyttävät toteutuksissaan laajalti rajapintoja (Potencier 2011). Eräs tärkeimmistä ja muissa PHP -

sovelluksissa käytetyimmistä Symfony komponenteista on HttpFoundation -komponentti, joka on eräänlainen työkalu HTTP -pyyntöjen ja -vastausten käsittelyyn. (Symfony 2017b.)

2.1.3 Symfony Bundles

Bundlet ovat jäsenelty joukko tiedostoja hakemistossa, joiden tarkoitus on lisätä yksittäinen ominaisuus liitettävään Symfony -sovellukseen. Bundle hakemisto pitää sisällä kaikki tähän ominaisuuteen liittyvät tiedostot kuten testit, php-, ja JavaScript -tiedostot (Ribeiro 2016). Toisin sanoen Bundlet ovat kuin muiden sovellusten pluginit eli liitännäiset. Bundlejen ja liitännäisten välillä ero piilee siinä, että Symfonyssa kaikki, mukaan lukien ydin toiminnallisuudet ja yksittäisiä sovelluksia varten kirjoitettu koodi koostuu bundleista. Tämän ansiosta esirakennettujen kolmannen osapuolen pakettien hyväksi käyttäminen ohjelmistokehityksessä on joustavaa. (Symfony 2017f.)



KUVIO 3. Oletus Bundle -pohja

Symfonyn bundleilla ei varsinaisesti ole rakenteellista vaatimusta, vaan ne voidaan rakentaa tarpeiden ja vaatimusten mukaan. Oletuksellisesti bundlet kuitenkin noudattavat kuvion 3 mukaista kansiorakennetta (Symfony 2017f). Kuviossa 3 esillä oleva EsimerkkiBundle niminen bundle -pohja on generoitu Symfonyn konsoli -komponentilla

komentorivikomennolla "php bin/console generate:bundle". Kansioden nimistä on loogisesti pääteltävissä mitä niiden on tarkoitus pitää sisällään; Controller -kansioon käsittelijät, Resources -kansioon muunmuassa konfigurointiin liittyvät tiedostot kuten reititykset ja palvelut ja Views -kansioon näkymät sekä bundlen PHP -tiedosto.

2.1.4 Doctrine

Symfonyssa ei ole ainuttakaan integroitua komponenttia tietokantojen kanssa työskentelyyn, sen sijaan Symfony on tiiviisti integroitu Doctrine -nimisen kolmannen osapuolen kirjaston kanssa (Symfony 2017a). Doctrine on pääasiallisesti tietokanta tallentamiseen ja olio kartoitukseen keskittynyt joukko PHP -kirjastoja. Sen lisäksi, että sillä on laadukas koodipohja, se pitää sisällään joustavia ja tehokkaita ominaisuuksia juurikin näihin olio kartoituksiin sekä tietokanta kutsuihin. (Doctrine 2016.)

```
1  <?php
2
3  namespace EsimerkkiBundle\Entity;
4
5  use Doctrine\ORM\Mapping as ORM;
6
7  /**
8   * @ORM\Entity
9   * @ORM\Table(name="example_orm")
10  */
11  class Esimerkki
12  {
13      /**
14       * @ORM\Column(type="integer")
15       * @ORM\Id
16       * @ORM\GeneratedValue(strategy="AUTO")
17      */
18      protected $id;
19
20      /**
21       * @ORM\Column(type="string")
22      */
23      protected $exampleColumn;
24  }
```

KUVIO 4. Symfony Entity -luokka

Kuviossa 4 on esitelty yksinkertainen Symfony Entity -luokka, johon on lisätty ORM merkinnät käyttäen Doctrinea. Entity -sanalla viitataan perusluokkaan, joka pitää sisällään tietoa. Jos luokasta poistettaisiin ORM -merkinnät, ei luokkaa pystyisi tallentamaan tietokantaan, koska silloin se olisi vain perinteinen PHP -luokka. Merkintöjen ansiosta Doctrine tietää täsmälleen, miten kuvion 4 Esimerkki -luokka ja sen ominaisuudet tulee tallentaa tietokantaan. Lisäksi Doctrine mahdollistaa haluttujen objektien hakemisen tietokannasta sen sijaan, että haettaisiin suuria määriä rivejä taulusta. (Symfony 2017a.)

2.1.5 Twig

Twig on nykyaikainen template -moottori PHP:lle, jonka yksi alkuperäisistä kehittäjistä on Symfony'n luoja Fabien Potencier. Twigin vahvuuksia ovat sen nopeus, joustavuus ja turvallisuus. Nopeus perustuu moottorin tapaan optimoida vaadittavan PHP -koodin määrä mahdollisimman pieneksi, joka on myös sen vahvuus verrattaessa tavalliseen PHP -koodiin. Twigin joustavuus puolestaan perustuu siihen, että se suoritetaan joustavassa jäsentimessä ja lexerissä eli sovelluksessa, joka suorittaa leksikaalista analyysiä. Nämä yhdessä antavat kehittäjälle mahdollisuuden lisätä omia tageja ja filtereitä sekä mahdollistavat oman DSL:n (Domain specific language) eli täsmäkielen luomisen. Turvallisuutta puolestaan edes auttavat "automaattinen karkaaminen ulostulosta" ja Sandbox - ominaisuus. (Twig 2017.)

`"This is an \"escaped\" string";`

KUVIO 5. Escape esimerkki

Automaattinen karkaaminen ulostulosta on suomennos termistä Automatic output escaping. Kuviossa 5 on esimerkki siitä, jossa kenoviivojen avulla sisäisemmät lainausmerkit luetaan lainausmerkkeinä sen sijaan, että ne toimisivat jonon päättäjinä. Sandbox -ominaisuudella pystytään puolestaan arvioimaan epäluotettava koodi hiekkalaatikkoimaisessa ympäristössä, jossa voidaan määrittää sallitut tagit, filterit, ominaisuudet tai metodit. (Twig 2017.)

<pre> \$text = 'thisIsVerySimpleExample'; \$intention = 'string123'; 1 {{ absolute_url('/example.txt') }} 2 {{ text humanize }} 3 {{ csrf_token(intention) }} </pre>	<pre> http://localhost:8000/example.txt This is very simple example c5hP5a5RHdrhxQ5DRcWLLdMBe9-Pqaakrlclnndbmqw </pre>
---	--

KUVIO 6. Esimerkkejä Symfony'n Twig laajennuksista

Twig on yksi Symfonyn oletus laajennuksista. Symfony lisää omia laajennuksia Twigin päälle integroidakseen joitakin komponentteja Twigin templateihin. Nämä laajennukset pitävät sisällään muunmuassa funktioita, filttereitä, tageja ja testejä, jotka ovat käytettävissä käytettäessä Symfonyn ydinohjelmistokehystä (Symfony 2017d). Kuviossa 6 on esiteltynä muutama näistä Symfonyn tarjoamista Twig -laajennuksista. Mustalla pohjalla on ensin kaksi PHP -muuttujaa ja näiden alapuolella twig -laajennukset, joihin näitä muuttujia on käytetty sekä valkoisella pohjalla on näkyvillä, miten nämä näkyvät selaimessa käyttäjälle.

2.2 JQuery

JQuery on 2006 John Resigin julkaisema alustariippumaton JavaScript -kirjasto, joka on tarkoitettu helpottamaan client puolen ohjelmointi ongelmia ja HTML -skriptausta (McFarlin 2016). Se on nopea ja ominaisuuksiltaan rikas, sekä siinä on yhdistelmä monipuolisuutta ja laajennettavuutta. Sen lisäksi, että se on helppokäyttöinen, löytyy siitä myös laaja ja kattava dokumentaatio. JQuery on yksi suosituimmista JavaScript -kirjastoista, ja se on muuttanut miljoonien ihmisten tapaa koodata JavaScriptillä. (JQuery 2017.)

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>
5       Example DOM
6     </title>
7   </head>
8   <body>
9     <h1>Title</h1>
10    <div>Content</div>
11  </body>
12 </html>
```

KUVIO 7. Esimerkki DOM:sta

DOM:lla eli dokumentti objektimallilla tarkoitetaan tapaa kuvata rakenteista dokumenttia, kuten HTML:llä tai XML:llä rakenne puuna. Kuviossa 7 on havainnollistamis esimerkki DOM:sta. Käsitteellisesti sitä voitaisiin kuvata puumaisena tietorakenteena, jossa jokaisella nodella eli HTML -elementillä on omat oksansa ja lehtensä. Kuvion 7 tapauksessa puun runkona toimii html -elementti, oksina head- ja body -elementit. Lisäksi body -elementillä on lehtinä elementit h1 ja div sekä head -elementillä title -elementti. (McFarlin 2016.)

```
1 <body>
2   <script type="text/javascript">
3     $("div").each(function(){
4       $(this).addClass("example");
5     })
6   </script>
7 </body>
```

KUVIO 8. Esimerkki DOM:n läpikäynnistä ja manipuloinnista

JQueryn suurimpiin vahvuuksiin lukeutuu sen kyky käydä läpi ja manipuloida tätä DOM -puuta. Kuviossa 8 on havainnollistamis esimerkki näistä molemmista vahvuuksista. Kuvion 8 rivillä 3 löytyvä "\$("div")" määrittelee jQueryn kohde elementit ja siihen ketjutettu "each()" -funktio puolestaan, mitä näille kohde elementeille tehdään. Tässä tapauksessa läpi käydään jokainen HTML:n div -elementti. Rivillä 4 löytyvä koodi taas puolestaan manipuloi kulloinkin vuorossa olevaa div -elementtiä lisäten niihin example -luokan. (McFarlin 2016.)

```

9      <input type="button" id="btn" value="Nappula" />
10     <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
11     <script type="text/javascript">
12         $("#btn").on("click", function(){
13             $.ajax({
14                 url: "example.html",
15                 context: this
16             }).done(function(){
17                 $(this).fadeOut("slow");
18             })
19         })
20     </script>

```

KUVIO 9. JQuery esimerkkejä

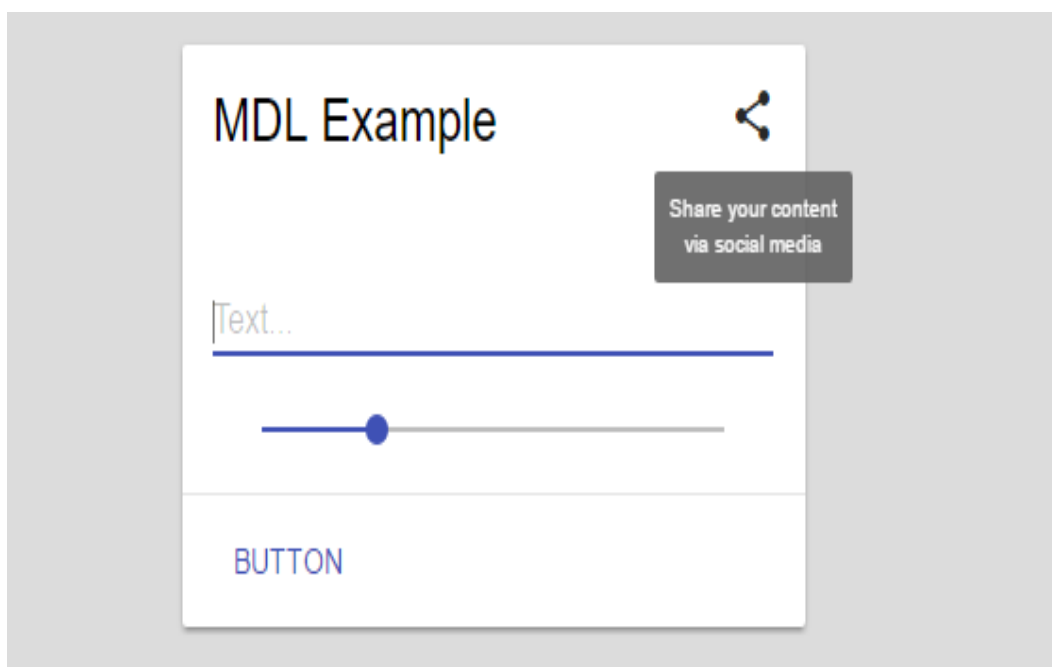
JQueryn muihin vahvuuksiin lukeutuu myös Ajax -kutsut, tapahtumien käsittely sekä animaatiot. Ajax -kutsu on pohjimmiltaan tapa, jolla verkkosivu voi tehdä kutsuja palvelimelle, käsitellä vastaukset ja päivittää tarvittavaa osaa sivustosta ilman koko sivuston uudelleen lataamista. Tapahtumakäsittelijöitä tarvitaan puolestaan käsittelemään esimerkiksi napin, näppäimistön tai hiiren painikkeen painallukset tai animaatioiden kuten esimerkiksi valonvälähdyksen luomiseen verkkosivulla. Kuviossa 9 on yksinkertaiset esimerkit havainnollistamaan näitä kolmea ominaisuutta. Rivillä 9 on input -elementti, johon on liitetty rivillä 12 tapahtuma. Tätä elementtiä painettaessa laukaistaan Ajax -kutsu, jonka koodi on nähtävissä riviltä 13 riville 18 lukuun ottamatta rivi 17, jolla on määritelty haihtumis -animaatio, joka suoritetaan kun Ajax -kutsu on suoritettu. (McFarlin 2016.)

2.3 Material Design Lite

MDL eli Material Design Lite on käyttöliittymäkomponenttikirjasto. Se on kehitetty Googlen 2014 julkaiseman Material Design -nimisen muotokielen pohjalta. Material Designin tarkoitus on antaa suuntaviivat hyvälle ja kauniille käyttöliittymäsuunnittelulle mobiililaitteilla riippumatta laitteen näytön koosta. MDL:n tarkoitus on taas tuoda nämä samat suuntaviivat verkkosivuille ja -sovelluksille. MDL:n tekemiseen on käytetty BEM eli Block Modifier -nimistä työkalua ja sen lähdekoodi on kirjoitettu Sass -

tyylikielillä. BEM:in tarkoitus on auttaa uudelleen käytettävien komponenttien tekemisessä ja niiden koodin jakamisessa front-end kehityksessä. Pohjimmiltaan MDL on kuitenkin perinteistä HTML:ää, CSS:ää ja JavaScriptiä, joka on jalanjäljeltään minimaalinen. (Osmani 2015.)

Material Designin perään liitetty Lite -sana kuvastaa muutamaa suunnittelun tavoitetta. MDL on muista ulkoisista JavaScript -kirjastoista riippumaton ja sillä on vain muutama muu riippuvuus sekä sen asentaminen ja käyttäminen on helppoa. Lisäksi se on ohjelmistokehys agnostikko, joka tarkoittaa sitä, että sen käyttäminen on mahdollista myös nopeasti muuttuvien ja kehittyvien front-end -kirjastojen, työkalujen ja ympäristöjen kanssa. MDL optimoi sisällöllisesti raskaimmatkin sivustot ja se toimii kaikilla nykyaikaisilla niin sanotuilla ikivihreillä selaimilla sekä Safarilla. Ikivihreällä -termillä tarkoitetaan selaimia, jotka päivittävät itseään ilman, että käyttäjä tiedostaa sitä ja näin ollen käyttäjä ei voi olla varma käyttämästään selaimen versiosta. Tällaisia selaimia ovat: Google Chrome, FireFox, Opera ja Edge. (Osmani 2015.)

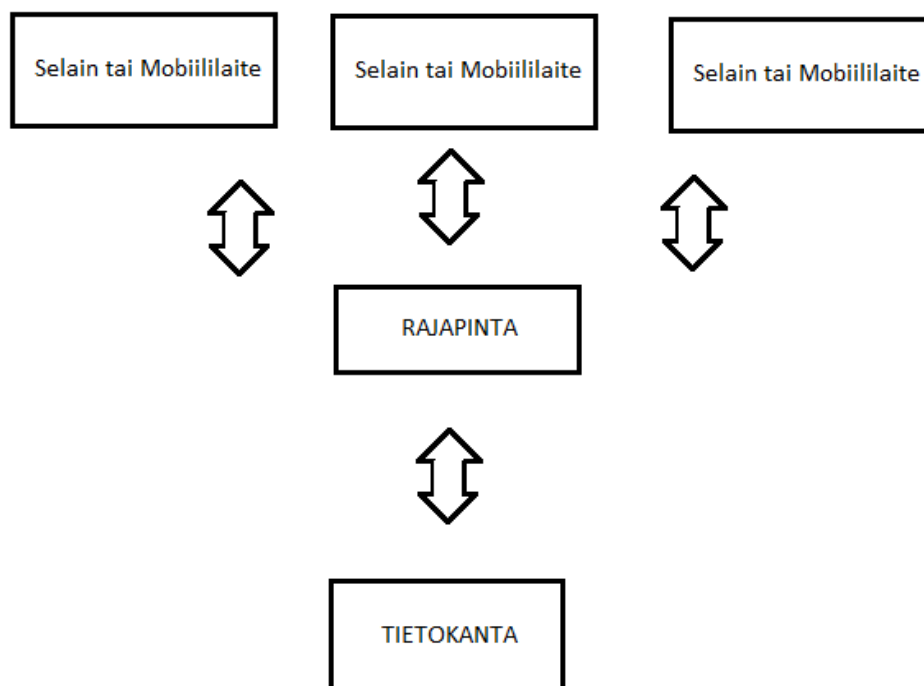


KUVIO 10. MDL komponentti esimerkkejä

MDL pitää sisällään paljon erilaisia komponentteja joihin lukeutuu muunmuassa erilaiset painikkeet, tekstikentät, vihjelaput ja korttipohjat. Sen lisäksi, että MDL:n kaikki komponentit ovat responsiivisia, löytyy siitä myös responsiivinen ruudukko -elementti. Kuviossa 10 on muutamia poimintoja MDL -komponenteista. Taustalla näkyvä valkoinen komponentti on MDL:n korttipohja, oikeassa yläkulmassa ikoni, johon on liitetty vihjelappu sekä vasemmassa alalaidassa puolestaan "Button" -tekstillä varustettu painike. Lisäksi kortin keskellä on tekstikenttä ja tämän alla liu'utin -komponentti. (Osmani 2015.)

2.4 Web API

Termillä API eli Application Programming Interface yleisesti tarkoitetaan rajapintaa, joka määrittelee miten ohjelmistokomponentit kommunikoivat keskenään. Verkkokehityksessä API on synonyymi online web servicelle eli verkkopalveluille, josta voidaan noutaa tietoa tai päivittää sitä. Vuosien saatossa näillä verkkopalveluilla on ollut muutamia muotoja ja nimiä, mutta tunnetuimpia ja käytetyimpiä ovat SOAP (Simple Object Access Protocol) API eli XML -pohjainen protokolla tietokoneiden väliseen tietojen vaihtamiseen ja tällä hetkellä suosituin REST (Representational State Transfer) API. REST API on verkko standardeihin pohjautuva arkkitehtuuri ja se käyttää HTTP -protokollaa kommunikoimiseen. (Kearn 2015.)



KUVIO 11. Rajapinta esimerkki

Kuviossa 11 on havainnollistamis esimerkki perinteisen rajapinnan arkkitehtuurista. Tieto haetaan normaaliin tapaan tietokannasta, joka välittyy rajapintaan. Rajapinnan ansiosta sama tieto voidaan välittää useammalle sovellukselle samanaikaisesti ilman, että jokaiselle sovellukselle tarvitsee ohjelmoida omaa sulautettua toiminta logiikkaa keskustelemaan tietokannan kanssa. Tämän lisäksi myös muutosten tekeminen rajapintaan on vaivattomampaa, sillä muutoksia ei tarvitse tehdä kuin yhdessä paikassa (Kearn 2015.) Rajapintojen tieto voidaan esittää monessa muodossa, mutta selvästi muita suosituimpia esitystapoja ovat XML ja JSON. (Wagner 2011).

```
1 <RootXML>
2   <Sample>Example</Sample>
3   <Leaf attr="helloworld">Example2</Leaf>
4   <Adapter>
5     <Plugin name="TapKal">Tapahtumakalenteri</Plugin>
6     <Plugin name="Nuoli">Nuoriso -plugin</Plugin>
7   </Adapter>
8 </RootXML>
```

KUVIO 12. XML esimerkki

XML (Extensible Markup Language) on tekstipohjainen merkintäkieli, joka muistuttaa ulkoisesti ja rakenteellisesti HTML -kieltä. Kuviossa 12 on esimerkki XML:sta. Kuviossa 12 tiedot on määritelty "<RootXML>" -tagin sisälle ja sen sisällä olevien tagien sisälle varsinainen tieto, kuten esimerkiksi Sample -tagin sisällä oleva Example -teksti. Tageille on mahdollista määrittellä attribuutteja mielensä mukaisesti, kuten Leaf -tagille on määritelty attr -attribuutti "helloworld" -arvolla. (Tutorialpoint 2017b.)

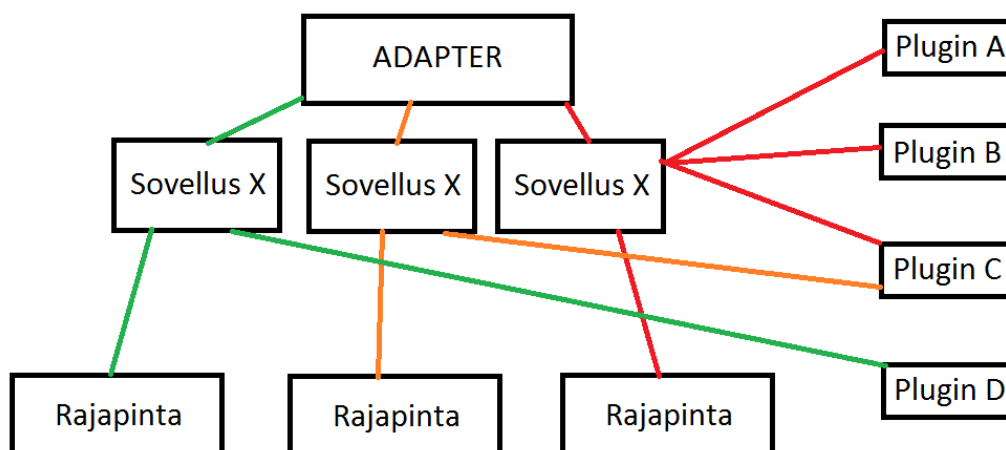
```
11 {
12   "example":
13     [
14       {
15         "id": 1,
16         "name": "Hello"
17       },
18       {
19         "id": 2,
20         "name": "World"
21       }
22     ]
23 }
```

KUVIO 13. JSON esimerkki

JSON (JavaScript Object Notation) on tekstipohjainen merkintätapa, jonka tarkoitus on olla mahdollisimman helposti luettavaa. Kuviossa 13 on esimerkki JSON:sta, jossa ylimpänä arvona on "example" ja sille on määritelty ryhmä objekteja. (Tutorialspoint 2017a.)

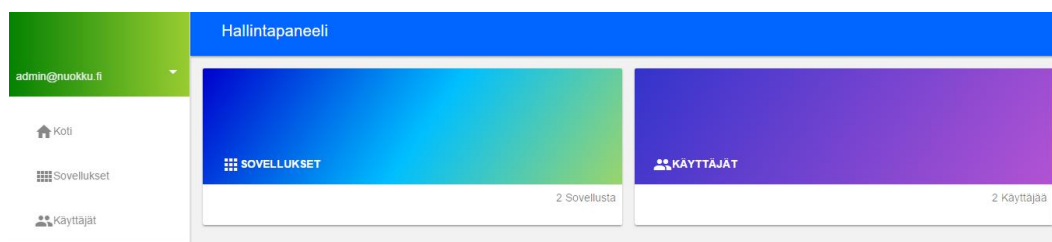
3 ADAPTER HALLINTASOVELLUS

Opinnäytetyön tarkoituksena oli rakentaa Mediatalo ESA:lle ja Lahden Nuorisopalveluille Adapter -hallintapuolen sovellus. Adapteri on eräänlainen hallintasovellus, jonka sisälle on mahdollista luoda rajaton määrä toisistaan riippumattomia pienempiä sisäisiä sovelluksia. Sisäisten sovellusten ensisijainen toimenkuva on tuottaa sisältöä rajapinnoille muiden sovellusten käytettäväksi. Näiden sovellusten perusrunko on hyvin yksinkertainen, mutta niiden toiminnallisuuksia on mahdollista lisätä niihin liitettävien liitännäisten avulla.



KUVIO 14. Adapterin havainnoillistamiskuva

Kuvio 14 on havainnoillistamiskuva Adapterista. Adapteri toimii nimityksenä koko sovellukselle ja sen sisälle luotavat sovellukset voidaan nimetä halutulla tavalla. Näihin sovelluksiin voidaan liittää olemassa olevista liitännäisistä halutut ja jokaisella sisäisellä sovelluksella on toisistaan riippumaton rajapinta.



KUVIO 15. Adapterin Koti -näkyvä

Adapterissa on kolme päänäkymää: koti, sovellukset ja käyttäjät -näkyvä. Näistä näkymistä koti -näkyvä on nähtävillä kuviossa 15. Koti -näkyvässä on MDL:n korttipohjilla toteutetut painikkeet kahteen muuhun päänäkymään. Sovellukset -näkyvässä on koti -näkyvän korttipohja tyylisesti luettelo käyttäjän sovelluksista ja käyttäjät -näkyvässä kaikki Adapterin käyttäjät on listattu tauluun.

3.1 Käytetyt teknologiat ja komponentit

Adapter-hallintasoftware on kehitetty Windows 7 -käyttöjärjestelmän työympäristössä käyttäen Symfonyn standardia ohjelmistokehystä, jQueryä ja Material Design Lite -käyttöliittymäkomponenttikirjastoa. Kehityksessä on myös käytetty apuna XAMPP -nimisen ohjelman Apache HTTP -palvelinta ja MySQL -tietokantaa. Symfonyn standardi ohjelmistokehys ja muut palvelimen puolella käytetyt komponentit on asennettu käyttäen Composer -pakettienhallintaohjelmaa ja puolestaan client -puolella käytetyt kirjastot on asennettu käyttäen bower -nimistä pakettienhallintaohjelmaa.

```
19     "require": {
20         "php": ">=5.5.9",
21         "symfony/symfony": "3.0.*",
22         "doctrine/orm": "^2.5",
23         "doctrine/doctrine-bundle": "^1.6",
24         "doctrine/doctrine-cache-bundle": "^1.2",
25         "symfony/swiftmailer-bundle": "^2.3",
26         "symfony/monolog-bundle": "^2.8",
27         "sensio/distribution-bundle": "^5.0",
28         "sensio/framework-extra-bundle": "^3.0.2",
29         "incenteev/composer-parameter-handler": "^2.0",
30         "friendsofsymfony/user-bundle": "~2.0@dev",
31         "jms/translation-bundle": "dev-master",
32         "friendsofsymfony/oauth-server-bundle": "^1.5",
33         "gedmo/doctrine-extensions": "^2.4",
34         "symfony/yaml" : "^3.0.4",
35         "symfony/finder" : "^3.0.4",
36         "symfony/assetic-bundle": "^2.8"
37     },
38     "require-dev": {
39         "sensio/generator-bundle": "^3.0",
40         "symfony/phpunit-bridge": "^3.0",
41         "doctrine/doctrine-fixtures-bundle": "^2.3",
42         "doctrine/doctrine-migrations-bundle": "^1.1"
43     },
```

KUVIO 16. Adapter hallintasovelluksen palvelinpuolen komponentit

Kuvio 16 on kuvankaappaus Adapterissa käytetyistä palvelinpuolen komponenteista, jotka on määritelty composer.json -nimiseen tiedostoon. Kuvion 16 riviltä 20 riville 29 ja riviltä 39 riville 40 ovat Symfonyn standardin ohjelmistokehityksen mukana tulleita komponentteja. Nämä asentuvat automaattisesti, jos Symfony asennetaan Composeria käyttäen, mutta tarvittaessa ne voidaan asentaa komennolla "composer install", kuten kaikki muutkin komponentit, jotka eivät tule Symfonyn mukana.

```
15  "dependencies": {
16    "jquery-ui": "^1.11.4",
17    "material-design-lite": "1.1.2",
18    "material-design-icons": "*",
19    "material-modal": "*",
20    "mdl-selectfield": "^1.0.2"
21  }

1  {
2    "directory" : "web/assets/contrib/"
3  }
4
```

KUVIO 17. Adapter hallintasovelluksen client puolen paketit

Kuvio 17 yläreuna on puolestaan kuvankaappaus Adapterin bower.json tiedostosta, josta näkyy client puolella käytetyt paketit. Lisäksi kuvion 17 alareunassa on kuvankaappaus .bowerrc -tiedostosta, jossa on määritelty polku, mihin paketit asennetaan. Paketit asennetaan "bower install" -komennolla. Periaatteessa kaikki voitaisiin asentaa pelkästään käyttäen joko Composeria tai Boweria, mutta pelkästään jo parhaan käytännön mukaan palvelin ja client puolen komponentit ja kirjastot on hyvä pitää erillään.

3.2 Käyttäjähallinnointi

Adapterin käyttäjähallinnointi on rakennettu kolmannen osapuolen FOSUserBundle -nimisen bundlen pohjalta. Kyseinen bundle on yksi Symfonyn suosituimmista ja käytetyimmistä bundleista ja käytännössä se tarjoaa kaiken, mitä tavallisessa käyttäjähallinnoinnissa saattaa tarvita, kuten esimerkiksi rekisteröitysen, kirjautumisen ja salasanan palautuksen.

3.2.1 Kutsuminen ja rekisteröityminen



KUVIO 18. Sähköpostikutsu Adapter -hallintasovellukseen

Adapteri on tarkoitettu vain Mediatalo ESA:n ja Lahden Nuorisopalvelun käytettäväksi, joten sovellukseen ei ole mahdollista rekisteröityä itsenäisesti. Adapteriin rekisteröityminen tapahtuu aina sisäisen sovelluksen kautta tehdystä kutsusta. Kutsuttaessa sovellus automaattisesti luo käyttäjälle tunnukset annetun sähköpostin perusteella ja lähettää siitä viestin kutsuttavan sähköpostiin. Kuvio 18 on kuvankaappaus sähköpostiin ilmestyvästä viestistä, joka sisältää sovelluksen osoitteen, käyttäjätunnuksen ja linkin salasanan vaihtamiseen. Mikäli kutsuttavan sähköposti eli käyttäjätunnus löytyy jo sovelluksen tietokannasta ei sisäisen sovelluksen kutsusta lähetetä erillistä ilmoitusta vaan sovellus tulee näkyviin Adapterin sovellusluettelo näkymään. Uusien käyttäjien on välttämätöntä vaihtaa salasana ensimmäisellä kerralla sisäänkirjautuakseen, sillä sovellus ei ilmoita käyttäjälle automaattisesti generoitua salasanaa.



Vaihda salasana

Uusi salasana

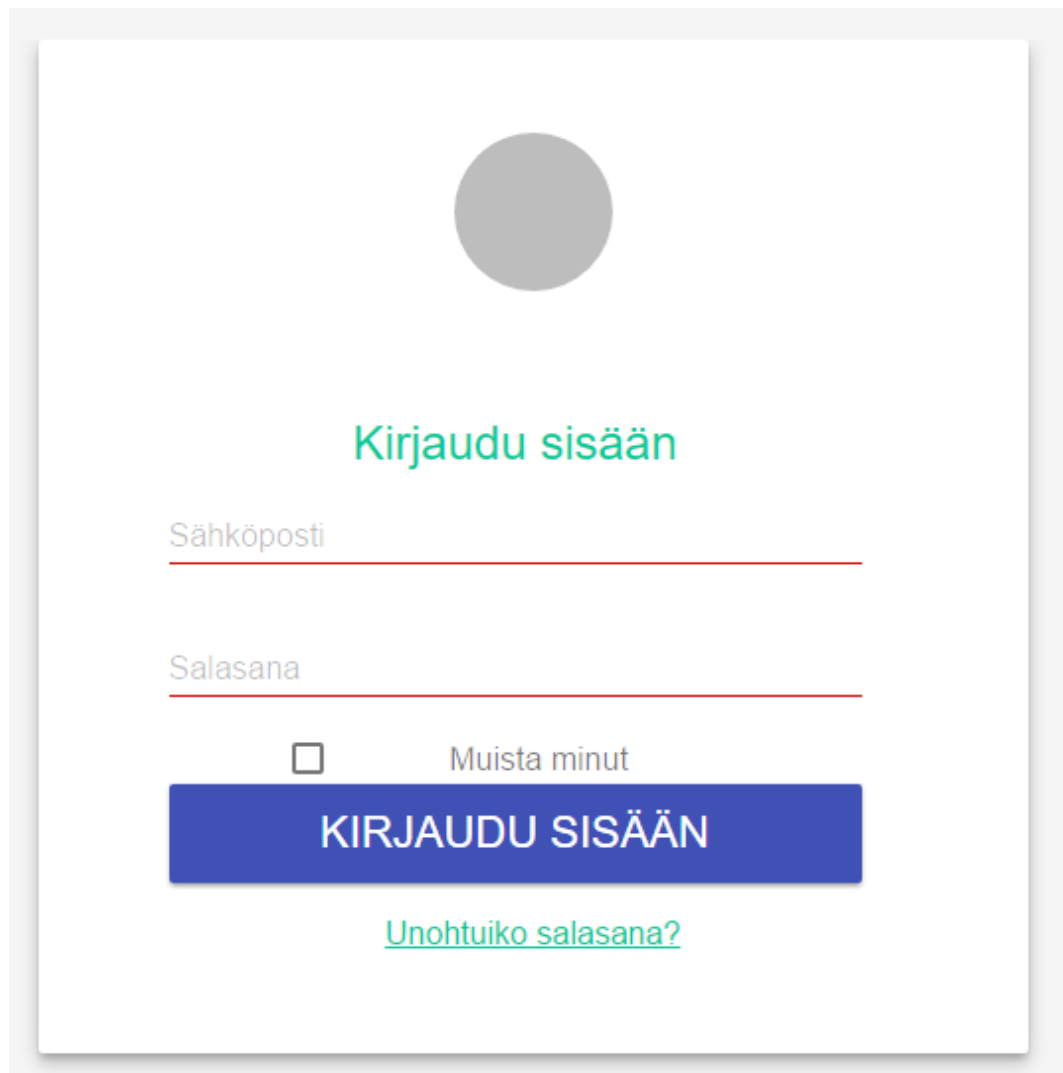
Varmista uusi salasana

VAHVISTA

KUVIO 19. Adapterin vaihda salasana -näkyvä

Kuvio 19 on kuvankaappaus Adapterin vaihda salasana -näkökuvasta. Näkökuvassa salasanan palautus lomake on toteutettu MDL:n korttipohjalla, johon on sijoitettu kaksi MDL:n tekstin syöttökenttää ja Vahvista -painike.

3.2.2 Kirjautuminen ja salasanan palauttaminen



The image shows a login form with a white background and a light gray border. At the top center is a gray circular placeholder for a profile picture. Below it, the title "Kirjaudu sisään" is displayed in a teal color. There are two input fields: "Sähköposti" (Email) and "Salasana" (Password), both with red horizontal lines below them. Below the password field is a checkbox labeled "Muista minut" (Remember me). A large blue button with white text "KIRJAUDU SISÄÄN" is centered below the checkbox. At the bottom, there is a teal link "Unohtuiko salasana?" (Forgot password?).

KUVIO 20. Sisäänkirjautuminen

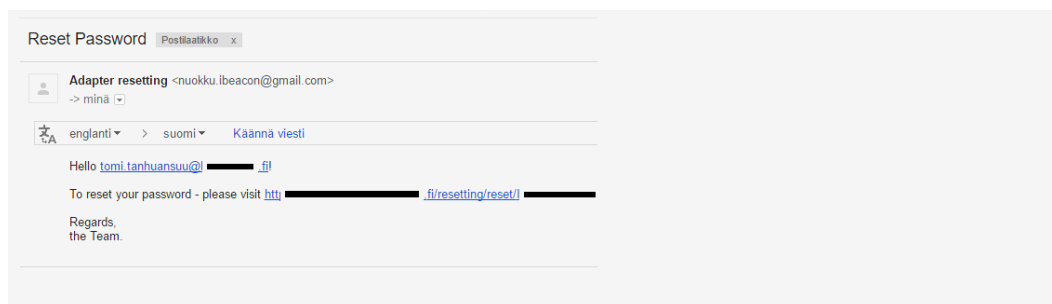
Kuviossa 20 on kuvankaappaus sisäänkirjautumisen näkymästä. Lomake on rakennettu MDL:n korttipohjan päälle, johon on sijoitettu harmaa pyöreä ikoni odottamaan mahdollista Adapter -ikonin. Tämän alle on sijoitettu Sähköposti- ja Salasana -kentät sekä Muista minut -valintaruutu, jonka avulla käyttäjätunnus pystyy tallentamaan seuraavaa sisäänkirjautumista varten. "KIRJAUDU SISÄÄN" -nappia painettaessa käyttäjä kirjautuu sisään mikäli sähköposti ja salasana täsmäävät tietokannassa olevaan käyttäjään. Lisäksi napin alapuolella on teksti "Unohtuiko salasana?", jota painamalla pääsee salasanan palauttamis näkymään.



The image shows a web form for returning a password. At the top, the title 'Palauta salasana' is displayed in a large, dark grey font. Below the title, the text 'Anna sähköpostiosoite' is centered in a smaller, grey font. A thin red horizontal line is positioned below this text. At the bottom of the form area, there is a prominent blue rectangular button with the text 'PALAUTA SALASANA' in white, uppercase letters.

KUVIO 21. Salasanan palauttaminen

Kuvio 21 on kuvankaappaus salasanan palauttamislomakkeesta. Myös kyseinen lomake on rakennettu MDL:n korttipohjan päälle. Lomakkeesta löytyy Anna sähköpostiosoite -kenttä, johon voidaan määrittää palautettavan käyttäjän sähköpostiosoite. Tämän jälkeen ”PALAUTA SALASANA” -nappia painamalla sähköpostin löytyessä tietokannasta käyttäjälle lähetetään kuvion 22 mukainen sähköposti viesti.



KUVIO 22. Salasanan palauttamissähköpostiviesti

Sähköpostiviestissä, joka lähetetään salasanan palauttamisen yhteydessä, pitää sisällään vastaanottajan sähköpostiosoitteen ja osoitteen salasanan palauttamiseen. Osoitetta painamalla pääsee määrittämään uuden salasanan. Jos salasana palauttamisviestin jättää kokonaan huomioitta, pysyy vanha salasana voimassa.

3.2.3 Käyttäjän profiilisivu

Käyttäjätunnus	Nimi	Muokkaa	Vaihda salasana
tomi.tanhuansuu@..._fi		MUOKKAA	VAIHDA

KUVIO 23. Adapterin käyttäjän profiilisivu

Adapterin profiilisivu on yksinkertainen ja käytännössä kaikesta turhasta karsittu. Kuviossa 23 on nähtävillä kuvankaappaus käyttäjän profiilisivusta. Kuvion ylempi sarake on rivinimi ja alempi sen arvo ja sitä vasemmalta oikealle luettaessa se sisältää käyttäjätunnuksen, nimen ja muokkaus- sekä vaihda salasana -toiminnot. Muokkaus -toiminnolla käyttäjän on mahdollista muokata käyttäjätunnustaan ja nimeään. Käyttäjätunnus on pakollinen ja sen on oltava sähköpostiosoite, mutta nimi on optionaalinen ja sen on tarkoitus auttaa muita käyttäjiä tunnistamaan yksittäisiä käyttäjiä, joiden käyttäjätunnus eli sähköpostiosoite ei suoranaisesti paljasta käyttäjän henkilöllisyyttä. Vaihda salasana -toiminnon takaa nimensä mukaisesti käyttäjän on mahdollista vaihtaa salasanansa. Molemmat toiminnot aukeavat omiin material-modal -nimisellä kirjastolla toteutettuihin ruudulle aukeaviin pikkuikkunoihin. Material-modal -kirjasto on yksi MDL:n

tyyliä mukaileva komponenttikirjasto, joka pitää sisällään juurikin ruudulle aukeavia pikkuikkunoita.

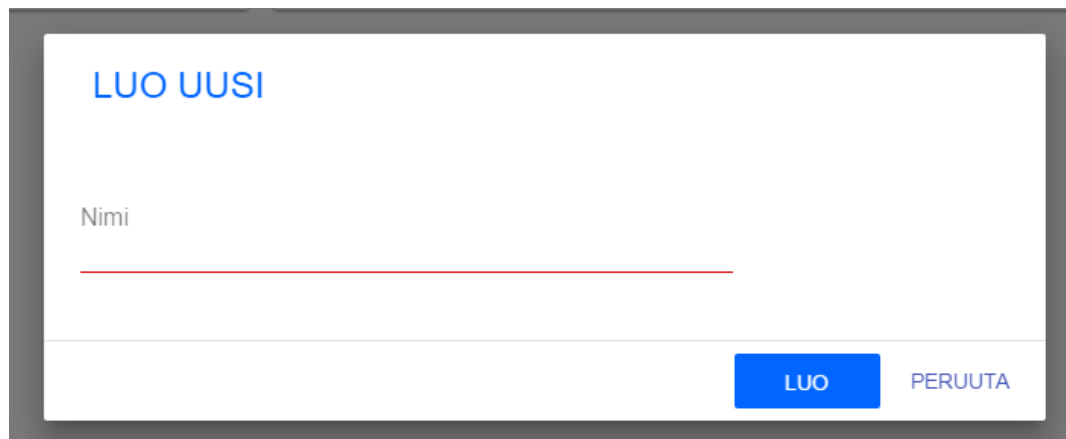
3.2.4 Käyttäjäoikeudet

Adapterissa käyttäjien oikeudet on jaettu perus- ja admin -käyttäjiin. Peruskäyttäjillä ei juurikaan ole minkäänlaisia toiminnallisia mahdollisuuksia Adapterin päätason sovelluksessa oman profiilinsa hallinnoinnin lisäksi. Oman profiilinsa ja sen hallinnoinnin lisäksi he näkevät kaikki kolme päänäkymää sekä pystyvät käyttämään sisäisiä sovelluksia, joihin heidät on lisätty. Admin -käyttäjät puolestaan pystyvät tekemään lähes kaiken mahdollisen Adapterin päätason sovelluksessa. He voivat luoda uusia sisäisiä sovelluksia, hallinnoida käyttäjiä muuttamalla heidän Adapterin roolia tai poistamaan heidät kokonaan. Heidän oikeuksiensa ulkopuolelle on kuitenkin rajattu sisäisten sovellusten poistaminen ja liittyminen muiden admin -käyttäjien luomiin sisäisiin sovelluksiin ilman näiden kutsua.

3.3 Adapterin sisäiset sovellukset

Adapterin sisäiset sovellukset ovat siis toisistaan riippumattomia sovelluksia omine käyttäjiineen, rajapintoineen ja toimintoineen sekä yksittäisiä sovelluksia on mahdollisuus laajentaa liitännäisten avulla. Käyttäjien on mahdollista käyttää sisäisiä sovelluksia, joihin he kuuluvat ja rajapinnan käyttäminen vaatii sisäisen sovelluksen uniikit oauth -avaimilla tapahtuvan tunnistautumisen.

3.3.1 Sisäisen sovelluksen luominen

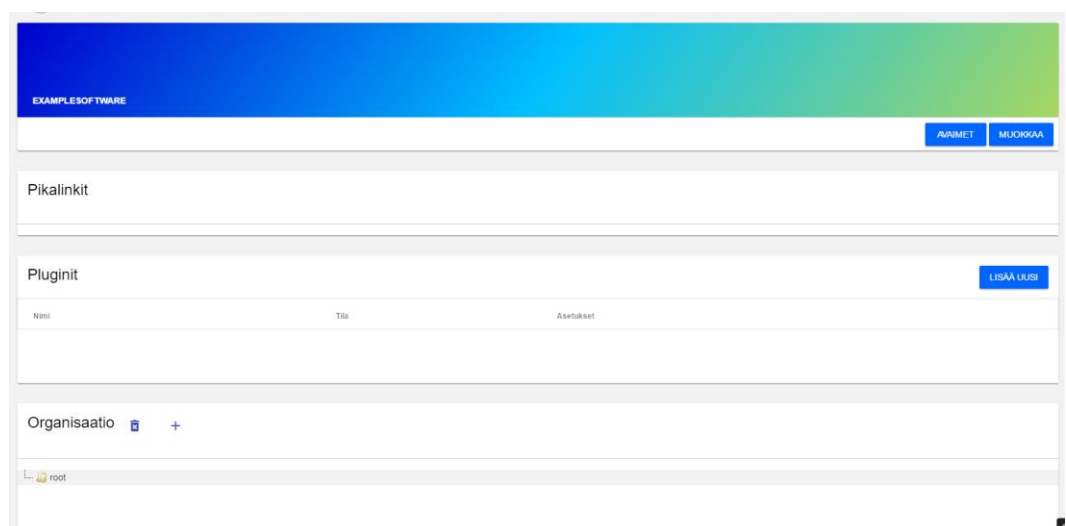


The screenshot shows a web form titled "LUO UUSI" in blue text. Below the title is a text input field labeled "Nimi" with a red underline. At the bottom right of the form, there are two buttons: a blue "LUO" button and a grey "PERUUTA" button.

KUVIO 24. Adapterin uuden sisäisen sovelluksen luominen

Adapteriin on mahdollista luoda rajaton määrä sisäisiä sovelluksia. Uuden sisäisen sovelluksen luominen on mahdollista sovellukset -päänäkymästä Admin -käyttäjän toimesta painamalla "Luo uusi" -painiketta. Tätä painiketta painettaessa aukeaa kuvion 24 mukainen ponnahdusikkuna, johon tarvitsee antaa vain uuden sisäisen sovelluksen nimi. Luomisen jälkeen Adapteri generoi uudelle sisäiselle sovellukselle perusrungon ja ohjaa käyttäjän juuri luomaansa sovellukseen.

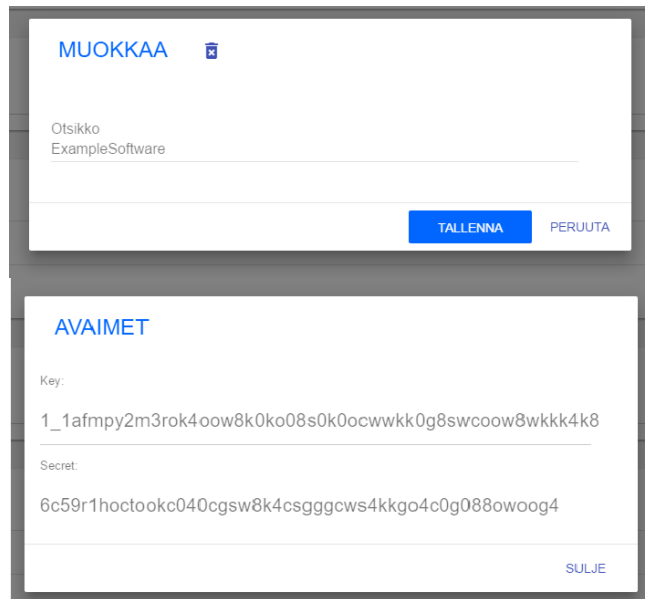
3.3.2 Sisäiset sovellukset perusrunko



The screenshot shows a web interface for managing internal applications. At the top, there is a blue header with the text "EXAMPLESOFTWARE" and two buttons: "AVAIMET" and "MUOKKAA". Below the header, there are several sections: "Pikalinkit", "Pluginit" (with a "LISÄÄ UUSI" button), and "Organisaatio" (with a "+" button). The "Organisaatio" section shows a tree view with a "root" node.

KUVIO 25. Osa Adapterin sisäisen sovelluksen perusrunko näkymästä

Kuvio 25 on kuvankaappaus juuri luodusta sisäisestä sovelluksesta. Kuvankaappauksessa on nähtävillä noin puolet perusrunkoon sisältyvistä toiminnoista. Toiminnot on eritelty toisistaan käyttäen MLD:n korttipohjia ja näitä korttipohjia on kuviossa 25 nähtävillä neljä kappaletta. Ylimmässä korttipohjassa vasemmalla sinisen pohjaväriin alakulmassa valkoisella fontilla on nähtävissä sovelluksen nimi ja saman korttipohjan oikeassa alalaidassa puolestaan kaksi painiketta ”Avaimet” ja ”Muokkaa”.



KUVIO 26. Sisäisen sovelluksen muokkaaminen ja avaimet - ponnahdusikkunat

Muokkaa -painiketta painettaessa avautuu kuvion 26 yläkulmassa näkyvän näköinen ponnahdusikkuna, josta on mahdollista muokata sovelluksen nimeä. Lisäksi ”MUOKKAA” -otsikon oikealla puolella näkyvästä roskakori -ikonista yksittäinen sisäinen sovellus on mahdollista poistaa. Sovelluksen poistamiseen on liitetty varmitustoiminto ehkäisemään vahinkossa poistamisia, sillä poistamisen jälkeen sovellukseen liittyvää tietoa ei ole enää mahdollista palauttaa.

Kuvion 26 alempi ponnahdusikkuna puolestaan pitää sisällään sovelluskohtaiset oauth -avaimet. Sovellusten rajapinnat eivät ole siis vapaasti käytettävissä vaan niiden käyttämiseen tarvitaan oauth -avaimilla suoritettava tunnistautuminen. Avaimet generoidaan automaattisesti sovelluksen luomisen yhteydessä ja siihen käytetään kolmannen

osapuolen FOSOAuthServerBundle nimistä bundlea. Sen lisäksi, että näillä avaimilla tunnustaudutaan, niiden avulla osataan hakea tietokannasta haluttu data.

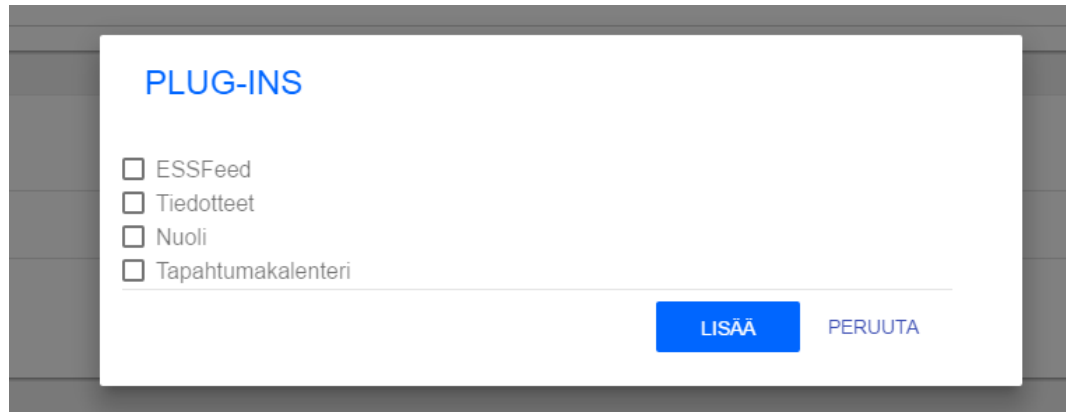
Kuviossa 25 toisiksi ylin korttipohja ”Pikalinkit” -otsikolla on oletuksellisesti muuten tyhjä. Joillakin liitännäisistä on ominaisuuksia, joiden käyttämistä voidaan nopeuttaa niin sanotuilla suorilla linkityksillä. Nämä ominaisuudet on erikseen määritelty ja ne lisätään automaattisesti tähän korttipohjaan liitännäisen päälle kytkemisen yhteydessä. Ulkonäöllisesti linkitykset ovat samannäköisiä sinisiä painikkeita, joita on nähtävillä esimerkiksi kuviossa 25.

KUVIO 27. Uuden organisaation luominen

Kuvion 25 alin Organisaatio -otsikolla varustettu korttipohja on puolestaan ominaisuus, jolla voidaan rakentaa sovelluksen sisäinen organisaatio. Oletuksellisesti organisaatio koostuu vain root -nimisestä haarasta, mutta siihen on mahdollista lisätä rajaton määrä uusia haaroja. Uuden haaran lisääminen tapahtuu kuviossa 25 nähtävästä ”+” -painikkeesta, josta aukeaa kuvion 27 mukainen ponnahtusikkuna. Uusi haara vaatii nimi-, linkin nimi-, linkin-, ja organisaation osoite -kenttien täyttämisen. Nimi -kenttä on organisaation nimi, linkki -kenttä esimerkiksi organisaation verkkosivuston osoite ja linkki nimi -kenttään voidaan määritellä linkille erillinen nimi ellei haluta käyttää organisaation nimeä. Lisäksi organisaation osoite -kenttään määritellään organisaation osoite.

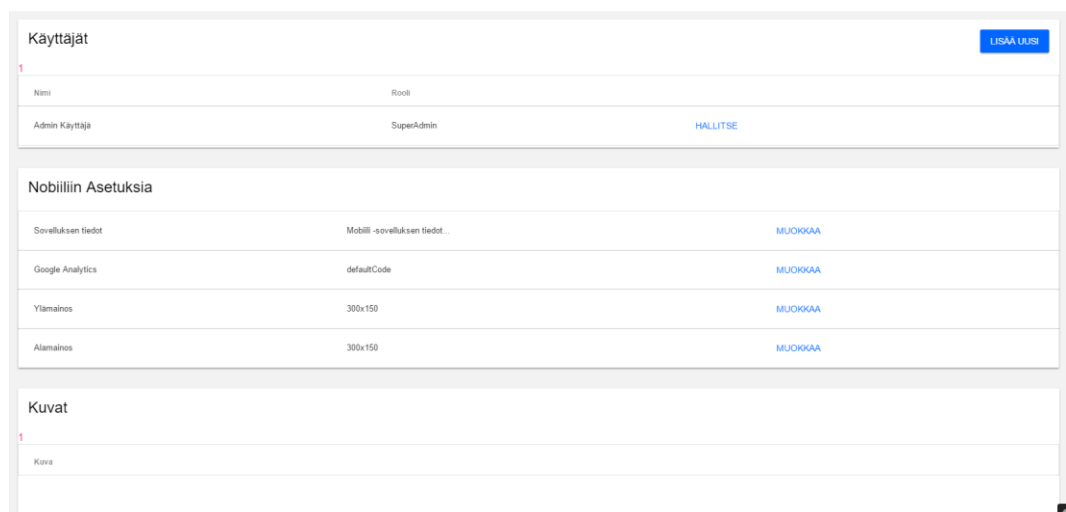
Organisaation haaroja voidaan puolestaan poistaa valitsemalla organisaatio -puusta poistettava organisaatio ja tämän jälkeen painamalla kuviossa 25 näkyvää roskakori -ikonia. Jos organisaatioon halutaan

myöhemmin tehdä rakenteellisia muutoksia, on se mahdollista ”Drag and drop” -menetelmällä. Tämä tarkoittaa siis sitä, että hiiren painikkeella voidaan raahata haaroja puussa uusiin haaroihin tai vaihtaa haara tasoja toistensa paikoille.



KUVIO 28. Pluginien eli liitännäisten lisääminen

Kuvion 25 Pluginit -otsikolla varustetulla korttipohjalla on taas listattuna sovellukseen liitetyt liitännäiset. Oletuksellisesti sovelluksiin ei ole lisätty liitännäisiä, mutta korttipohjan oikeassa ylä laidassa ”Lisää uusi” -painikkeesta painettaessa avautuu kuvion 28 näköinen ponnahdusikkuna, josta voidaan valintaruutuja valitsemalla asentaa halutut liitännäiset sovellukseen.



KUVIO 29. Adapterin sisäisten sovellusten toinen osa perusrungosta

Kuviossa 29 on esillä loput sisäisten sovellusten perusrungosta. Kuvion kolmesta korttipohjasta alin Kuvat -otsikolla varustettu on oletuksellisesti

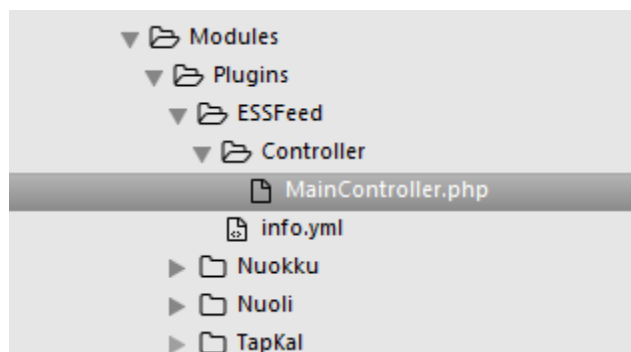
tyhjä, mutta jos liittännäisistä löytyy ominaisuus, jolla voidaan lisätä sovellukseen kuvia, niin ne lisätään tähän korttipohjaan, josta niiden hallinnointi on helppoa. Lisäksi tämän yläpuolella on korttipohja, joka on nimetty toimeksiannon puolesta hieman harhaan johtavasti. Kortissa näkyvien ominaisuuksien avulla voidaan kuitenkin määrittää sovelluskohtaiseen rajapintaan kaksi mainosta, google analytics -koodi ja vapaamuotoinen teksti sovelluksen tiedoista.

Lisäksi kuvion 29 ylimpään korttipohjaan on listattu sovellukseen kuuluvat käyttäjät, heidän roolinsa ja hallitsemisominaisuus sekä uusien käyttäjien lisääminen sinisestä painikkeesta kortin oikeassa ylä laidassa. ”Hallitse” - painiketta painettaessa käyttäjälle avautuu näkymä, josta voi hallita valitun käyttäjän roolia tai organisaatio sekä poistaa käyttäjä kyseisestä sovelluksesta.

3.3.3 Sisäisten sovellusten käyttöoikeudet

Sisäisten sovellusten roolitus on jaettu kolmeen osaan: super admineihin, admineihin ja peruskäyttäjiin. Huomion arvoista on, että Adapterin päätason sovelluksen roolilla ei ole merkitystä sisäisissä sovelluksissa eli myös Adapterin pääkäyttäjä voi olla sisäisen sovelluksen peruskäyttäjä. Sisäisissä sovelluksissa peruskäyttäjän rooli on varsin marginaalinen, sillä sisäisen sovelluksen perusrunko näkymään peruskäyttäjällä ei ole lainkaan pääsyä. Tämä tarkoittaa sitä, että jos sovellukseen ei ole lisätty liittännäisiä, jotka tarjoaisivat peruskäyttäjälle toiminnallisuuksia, ei peruskäyttäjä voi lainkaan käyttää sisäistä sovellusta. Admin -käyttäjän rooli poikkeaa peruskäyttäjistä toiminnollisesti vain hieman. Admin -käyttäjällä on pääsy sovelluksen perusrunko näkymään ja mahdollisuus nähdä organisaatio -puu, asennetut liittännäiset, käyttäjät ja kuvat, mutta Admin -käyttäjällä on oikeudet ainoastaan organisaatio -puun hallinnointiin. Super admin -käyttäjä on puolestaan täysi vastakohta peruskäyttäjälle, jolla on oikeudet kaiken mahdollisen hallinnointiin.

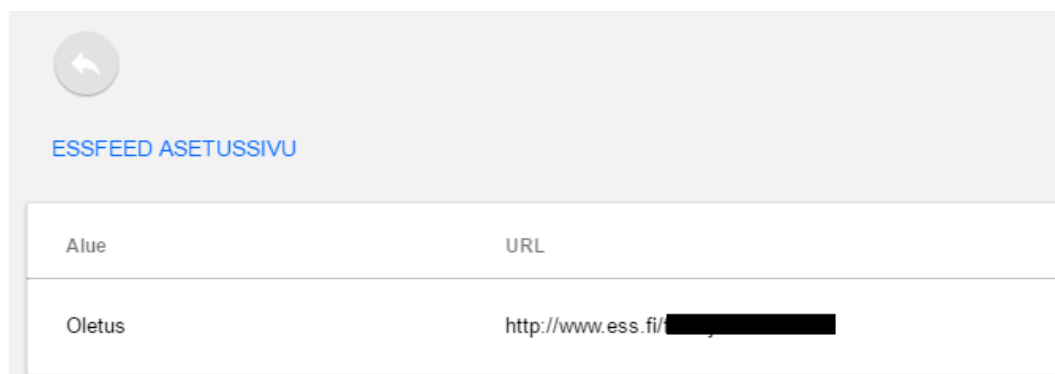
3.4 Adapterin sisäisten sovellusten liitännäiset



KUVIO 30. Adapterin ensimmäisen version liitännäiset ja kansiorakenne

Kuviossa 30 on nähtävillä Adapterin ensimmäisen version mukana olleet liitännäiset ja niiden minimaalisin vaadittu tiedosto- ja kansiorakenne. Joka kerta sisäänkirjautumisen yhteydessä Adapteri tarkastaa Symfonyn src -kansion bundlet etsien niistä mahdollista Plugins -nimistä kansiota. Tämän löytyessä Adapteri yrittää etsiä Plugins -nimisen kansion sisältä kaikki info.yml -nimiset tiedostot ja mikäli yml -tiedostossa on määriteltynä "machineName" ja "humanName" -kentät, olettaa sovellus kyseessä olevan liitännäinen.

3.4.1 ESSFEED -plugin



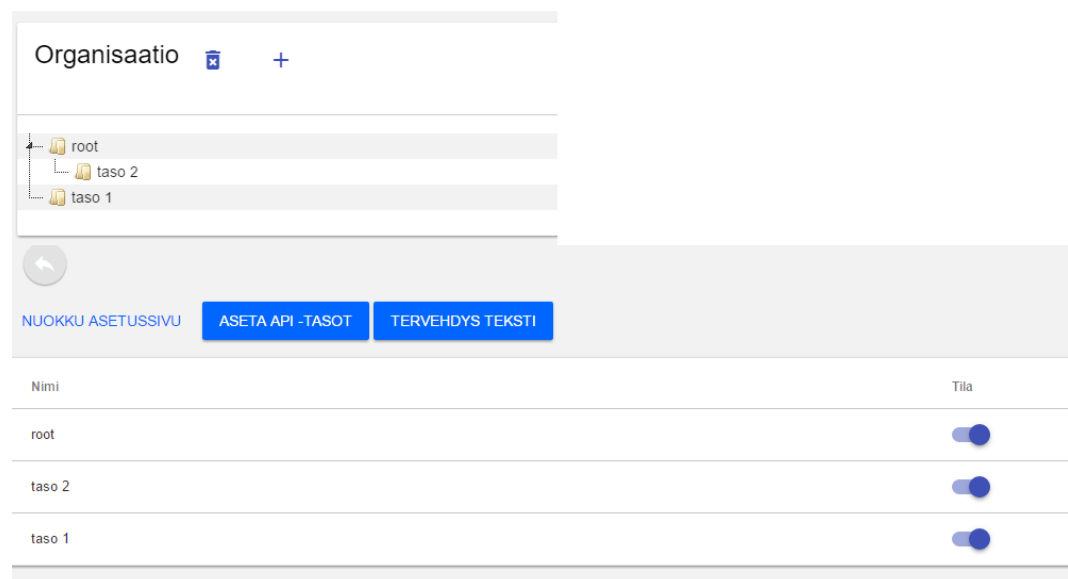
KUVIO 31. EESFEED -pluginin asetussivu

ESSFeed -plugin on liitännäinen, joka hakee Mediatalo ESA:n rajapinnasta nuorille suunnattuja uutisia. Adapteri poimii kyseisestä rajapinnasta muunmuassa tapahtuma tunnisteet, kuvaukset, kuvat ja kirjoittajan tiedot sekä osoitteen uutiseen. Kuviossa 31 on nähtävillä

kuvankaappaus liitännäisen asetussivusta. Sisäiseen sovellukseen liittämisen jälkeen asetussivulle on mahdollista päästä kyseisen sovelluksen Plugins -korttipohjalle ilmestyvän painikkeen kautta. Rajapintaa, josta uutiset haetaan on mahdollista vaihtaa, mutta rajapinnan täytyy olla rakenteellisesti sama kuin oletus rajapinnan.

3.4.2 Nuokku -plugin

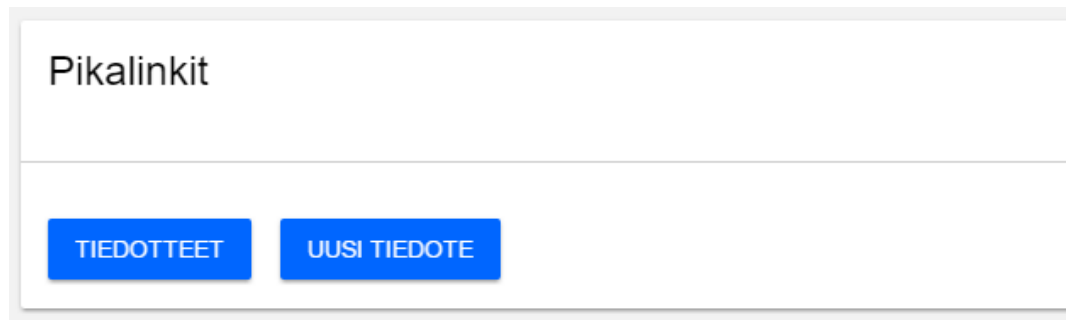
Nuokku -plugin on eräänlainen sisällön tuottamiseen tarkoitettu liitännäinen, jonka avulla pystytään luomaan tiedotteita organisaatiokohtaisesti. Tämä liitännäinen on ainoa alkuperäisistä liitännäisistä, joka tarjoaa toiminnallisuuden peruskäyttäjille ja näin ollen mahdollisuuden käyttää sisäistä sovellusta. Nuokku -plugin käyttäähyväkseen siis perusrungon organisaatorakennetta. Sen avulla käyttäjät voidaan rajata niin, että he voivat lähettää tiedotteita vain organisaatiohaaraan, johon he kuuluvat.



KUVIO 32. Nuokku -plugin asetussivu ja organisaatio -puu

Kuvion 32 kuvankaappauksessa on nähtävillä ylempänä organisaatio -puu ja alempana Nuokku -pluginin asetussivu. Kuten kuvioista voi havaita, organisaatio on taulukoitu asetussivun tauluun. Taulukosta pystytään rajaamaan yksittäisiä organisaatiohaaroja, joihin ei voi lähettää tiedotteita

kytkemällä oikeassa reunassa näkymä kytkin pois päältä. Mikäli halutaan sallia tiedotteiden lähettäminen vain tietyn tason organisaatioille pystytään se määrittämään "ASETA API -TASOT" -painikkeen takaa löytyvästä ponnahdusikkunasta. "TERVEHDYS TEKSTI" -painikkeen takaa niin ikään voidaan määrittää tervehdys teksti, jota voidaan esimerkiksi käyttää tiedotteiden listauksen yhteydessä mobiilisovelluksessa.



KUVIO 33. Nuokku -pluginin pikalinkit

Kun Nuokku -plugin liitetään sisäiseen sovellukseen, lisätään perusrungon pikalinkit korttipohjaan kaksi uutta painiketta "TIEDOTTEET" ja "UUSI TIEDOTE", kuten kuviossa 33 on nähtävillä. Tiedotteet -painikkeen takaa käyttäjän on mahdollista nähdä taulukoituna kaikki ne tiedotteet, jotka hänen olisi ollut mahdollista itse lähettää eli tarkoittaen sitä, että kaikki tiedotteet, jotka on lähetetty kyseisen käyttäjän organisaatiohaaralle. Lisäksi käyttäjällä on mahdollista muokata ja poistaa kyseisiä tiedotteita.

LUO TIEDOTE

Otsikko

Liitä kuva...

Tai valitse kuva galleriasta

Lisää Youtube -video linkki

Vastaanottajat

- root
- taso 2

Lähetä kaikille

Kyllä Ei

Alkamispäivämäärä

03/10/2017	11:59 AM
------------	----------

Loppumispäivämäärä

03/10/2017	11:59 AM
------------	----------

Paikka/Lähettäjä

KUVIO 34. Uuden tiedotteen luominen Nuokku -pluginilla

Kuviossa 34 on kuvankaappaus uuden tiedotteen luomisen näkymästä. Uuden tiedotteen luominen vaatii minimissään otsikon, sisällön vastaanottajan, alkamis- ja loppumispäivämäärän sekä paikan tai lähettäjän nimen. Kuviossa 34 sisällön tekstikenttä on rajattu pois näkyvistä. Tiedotteeseen on lisäksi mahdollista lisätä kuva ja youtube -linkki. Kuvan voi ladata omalta koneeltaan tai vaihtoehtoisesti valita sovelluksen galleriasta, joka koostuu jo aiemmin lisätystä kuvista. Kuviossa 34 näkyvä "Lähetä kaikille" -valinta ominaisuus mahdollistaa tiedotteen lähettämisen kaikille organisaatihaaroille lisäen siihen

eräänlaisen merkinnän, joka korostaa sen tärkeyttä. Esimerkiksi jos sovelluksessa, joka hakee kyseisiä tiedotteita rajapinnasta on ominaisuus että käyttäjä voi rajata ja vastaanottaa vain haluamiansa tiedotteita niin tällä pystytään helposti yliajamaan käyttäjän valinnat ja näyttämään tiedote valinnoista riippumatta. Kyseinen lähetä kaikille ominaisuus on käytettävissä vain sisäisen sovelluksen superkäyttäjille ja se on piilotettu muilta.

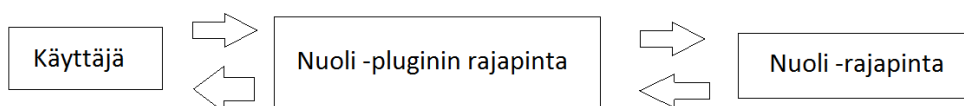


KUVIO 35. CKeditor

Nuokku -pluginin sisällön tekstikenttä on toteutettu kuvion 35 mukaisella CKeditor -tekstieditorilla. Tämän avulla käyttäjän on helppo muotoilla tuotettavaa sisältöä esimerkiksi lihavoinnilla tai kursivoinnilla. Lisäksi luetteloiden tekeminen on helpompaa kuin perinteisessä tekstikentässä.

3.4.3 Nuoli -plugin

Nuoli -plugin on liitännäinen, joka hakee Lahden Nuorisopalveluiden Nuoli -sovelluksen rajapinnasta pyydettyä yksittäisten nuorten tiedot ja välittää ne omaan rajapintaansa. Tämän liitännäisen tarkoitus on sähköistää nuorisokortti. Henkilötietojen suojaamiseksi sisäinen sovellus ei tallenna tai sisällä henkilöiden tietoja.



KUVIO 35. Nuoli -plugin toimintamalli

Kuviossa 35 on havainnollistettu aiemmin selitettyä toimintamallia. Käyttäjä lähettää Nuoli -pluginin rajapintaan ensimmäisenä hakupyynnön, joka sisältää henkilökohtaisen tunnistautumis -koodin ja uniikin

laitekohtaisen id:n sekä toiminnon, mitä halutaan tehdä. Toiminto vaihtoehtoja ovat: tunnistautumisen, päivittäminen ja poistaminen. Nuoli - plugin tarkistaa ensin, että kaikki tarvittava tieto on saatavilla ja sen jälkeen välittää sen Nuolen rajapintaan, joka palauttaa toimintokohtaisen tiedon. Tämän jälkeen tieto palaa päinvastaisessa suunnassa takaisin käyttäjälle käytettäväksi.

3.4.4 TapKal -plugin

Kategoria	Otsikko	Kuvaus
Seniorit	Marja ja Reijo Lundberg esiintyvät	
Elokuvat	Kino Lehtioja: Truth - totuus	Cate Blanchett on Mary. Mapes ja Robert Redford on ...
Tapahtumat	Kirjasto on hiljaisuutta ja omaa rauhaa	uusitun aikuisten alueen ryhmähuoneessa Mindfulness...
Näyttelyt	Lahti kansainvälinen hiihtokaupunki/ Lahti International Ski City	Hiihtomuseon Lahti kansainvälinen hiihtokaupunki -...
Elokuvat	Saattokeikka	Komedia, draama, kotimainen. Kesto 1 t 18 min. Ohj...
Tapahtumat	Perhepuistotoimintaa Puutarhurinkadulla	Perhepäivän viettoa eläinleikkien ja -tarinoiden m...
Elokuvat	Vaijana 2D dub. (7)	
Näyttelyt	Taidenäyttely	Wellamo-opiston Lahden torstain kuvataideryhmiän 2...
Elokuvat	Lego Batman elokuva 2D dub. (7)	
Elokuvat	La La Land (S)	

KUVIO 36. TapKal -pluginin asetussivu

Tapkal -plugin on tapahtumakalenteri liitännäinen, joka hakee Mediatalo ESA:n menoinfoon lisäämiä tapahtumia. Menoinfo on siis sivusto, johon voi lisätä tapahtumia. Kuviossa 36 on nähtävillä kuvankaappaus liitännäisen asetussivusta, joka on saman tyyppinen muiden liitännäisten asetussivujen kanssa. Tekstikentässä on näkyvillä sensuroitu osoite parametreineen, josta tapahtumat haetaan eli tässä tapauksessa osoite Mediatalo ESA:n menoinfoon. Osoitetta voi muokata suoraan kirjoittamalla tekstikenttään ja painamalla "TALLENNA" -painiketta. "OLETUS" -painiketta painettaessa palautetaan nimensä mukaisesti oletus osoite.

Painikkeiden alapuolelle on taulukoitu kaikki tällä hetkellä haetut tapahtumat, jotka välitetään rajapintaan.

4 YHTEENVETO

Tässä opinnäytetyössä oli tarkoitus toteuttaa Mediatalo ESA:lle ja Lahden Nuorisopalveluille Adapter -niminen hallintasovellus, jolla voidaan hakea ja tuottaa sisältöä rajapinnoille sekä tarjota mahdollisuus sähköistää nuorisotaloilla käytettävä nuorisokortti. Sovellus toteutettiin osana koulun opintoihin liittyvää työharjoittelua Korpimedia Oy:ssä opiskelijoiden toimesta. Päätekniikoina oli tarkoitus käyttää Symfony PHP kehitysalustaa, jQueryä ja Material Design Lite -käyttöliittymäkomponenttikirjastoa. Apuna toteutuksessa käytettiin valmiita kolmannen osapuolen tarjoamia Symfony Bundleja sekä muita hyviä frontend -puolen komponentteja ja kirjastoja.

Sovellukselle asetetut päätavoitteet saavutettiin hyvin. Lopputuloksena syntyneellä Adapter -hallintasovelluksella on mahdollista tuottaa ja hakea sisältöä sekä nuorisokortin sähköistämisen mahdollistava ominaisuus. Myös alitavoitteet saavutettiin vähintäänkin tyydyttävästi. Sisäisiä sovelluksia pystytään tuottamaan rajaton määrä ja ne ovat toisistaan täysin riippumattomia. Toiminnallisuuksien toteuttaminen täysin liitännäisinä jäi kuitenkin osittain toteutumatta. Toiminnallisesti kaikki toimii, mutta osa liitännäisten ominaisuuksista on sijoitettu sisäisten sovellusten perusrunkoon, mikä ei täysin täsmää alitavoitteen kanssa.

Tällä hetkellä sovellus on aktiivisessa käytössä Lahden Nuorisopalveluilla ja teknisesti sovellus on toiminut lähes virheettömästi. Jos alkaisin toteuttamaan kyseistä sovellusta uudelleen, niin valitsisin pääteknologioiksi ReactJS:n ja NodeJS:n ja suunnittelisin toteutuksen paremmin. Myös testaukseen panostaisin enemmän, vaikka suurempia ongelmia ei missään vaiheessa ilmennytkään.

Sovellusta on tarkoitus jatkokehittää käytettäväksi myös Päijät-Hämeen ulkopuolella. Jatkokehitys kuitenkin vaatii jonkin verran rakenteellisia muutoksia toteutuakseen. Lisäksi jatkokehityksessä pyritään parantamaan ja oikomaan ensimmäisen version koodia ja tietokantarakenteita. Koodista löytyy paikoittain turhaa toistamista, joka on nykyaikaisen ohjelmoinnin DRY (Don't repeat yourself) eli "älä toista itseäsi" -periaatteen vastaista.

Tietokantarakenteet puolestaan sisältävät paikoittain ylimääräisiä tauluja, joita on mahdollista optimoida.

LÄHTEET

Doctrine 2016. Why use Doctrine? [viitattu 11.2.2017]. Saatavissa: <http://www.doctrine-project.org/>

JQuery 2017. [viitattu 11.2.2017]. Saatavissa: <https://jquery.com/>

Kearn, M. 2015. [viitattu 25.3.2017]. Saatavissa: <https://blogs.msdn.microsoft.com/martinkearn/2015/01/05/introduction-to-rest-and-net-web-api/>

McFarling, T. 2016. What is jQuery. Envato Tuts+ [viitattu 4.2.2017]. Saatavissa: <https://code.tutsplus.com/tutorials/what-is-jquery--cms-26232>

O'Dell, C. 2017. What is Symfony. Unleashed Technologies, LLC. [viitattu 20.1.2017]. Saatavissa: <https://www.unleashed-technologies.com/blog/2017/01/31/introduction-symfony>

Osmani, A. 2015. Medium [viitattu 4.2.2017]. Saatavissa: <https://medium.com/google-developers/introducing-material-design-lite-3ce67098c031#.m39crgn0k>

Potencier, F. 2011. What is Symfony2 [viitattu 20.1.2017]. Saatavissa: <http://fabien.potencier.org/what-is-symfony2.html>

Ribeiro, D. 2016. Yes, you can have low coupling in a Symfony application! Medium. [viitattu 20.1.2017]. Saatavissa: <https://medium.com/@drgomesp/yes-you-can-have-low-coupling-in-a-symfony-standard-edition-application-41c5a34aaf94#.dzlpo5pnw>

Symfony 2017e. Databases and the Doctrine ORM [viitattu 22.1.2017]. Saatavissa: <http://symfony.com/doc/current/doctrine.html>

Symfony 2017b. HttpFoundation (Symfony Components) [viitattu 21.1.2017]. Saatavissa: <http://symfony.com/components/HttpFoundation>

Symfony 2017f. Installing & Setting up the Symfony Framework [viitattu 22.1.2017]. Saatavissa: <https://symfony.com/doc/master/setup.html>

- Symfony 2017d. Symfony Twig Extensions [viitattu 22.1.2017].
Saatavissa: http://symfony.com/doc/current/reference/twig_reference.html
- Symfony 2017g. Requirements for Running Symfony [viitattu 23.1.2017].
Saatavissa: <http://symfony.com/doc/master/reference/requirements.html>
- Symfony 2017c. The Bundle System [viitattu 21.1.2017]. Saatavissa:
<http://symfony.com/doc/current/bundles.html>
- Symfony 2017a. What is Symfony [viitattu 21.1.2017]. Saatavissa:
<https://symfony.com/what-is-symfony>
- Tutorialspoint 2017a. JSON Tutorial [viitattu 25.3.2017]. Saatavissa:
<https://www.tutorialspoint.com/json/>
- Tutorialspoint 2017b. XML Overview [viitattu 25.3.2017]. Saatavissa:
https://www.tutorialspoint.com/xml/xml_overview.htm
- Twig 2017. [viitattu 4.2.2017]. Saatavissa: <http://twig.sensiolabs.org/>
- Wagner, J. 2011. [viitattu 25.3.2017]. Saatavissa:
<https://code.tutsplus.com/articles/the-increasing-importance-of-apis-in-web-development--net-22368>

