

TALLINNA ÜLIKOOL
DIGITEHNOLOOGIATE INSTITUUT

Sisuhaldussüsteemi Drupal 8 uued võimalused

Bakalaureusetöö

Autor: Sten Martmaa

Juhendaja: Inga Petuhhov

Autor: ,, ,, 2017

Juhendaja: ,, ,, 2017

Instituudi direktor: ,, ,, 2017

Tallinn 2017

Autorideklaratsioon

Deklareerin, et käesolev bakalaureusetöö on minu töö tulemus ja seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

..... (allkiri ja kuupäev)

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, Sten Martmaa (sünnikuupäev: 14.12.1991)

1. Annan Tallinna Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Sisuhaldussüsteemi Drupal 8 uued võimalused“, mille juhendaja on Inga Petuhhov, säilitamiseks ja üldsusele kättesaadavaks tegemiseks Tallinna Ülikooli Akadeemilise Raamatukogu repositooriumis.
2. Olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tallinnas,

..... (allkiri ja kuupäev)

Sisukord

Sõnastik.....	5
Sissejuhatus	6
1. Metoodika	7
2. Paigaldamine ja tehnilised andmed	8
3. Kasutajaliides.....	10
3.1. Mobiilne veeb.....	11
3.2. Juurdepääsetavus.....	14
4. Sisuhaldussüsteem	17
4.1. Arhitektuur	17
4.2. Struktuur	19
4.3. Twig teema mootor.....	21
4.4. Teemad.....	23
5. Moodulid	31
5.1. Tuuma moodulid	32
5.1.1. Views.....	33
5.1.2. CKEditor	36
5.1.3. Quick Edit	37
5.1.4. Mitmekeelsus.....	39
5.2. Kohandatud moodulid.....	46
Kokkuvõte	55
Summary.....	56
Kasutatud kirjandus	57
LISAD.....	59
Lisa 1	60

Sõnastik

Mõiste eesti keeles	Mõiste inglise keeles
Adapteeruv pilt	Responsive image
Eelkõige mobiililset veebi silmas pidades	Mobile-first
Kasutajaliidese arendaja	Front-end developer
Kasutajate poolt loodud moodulid	Contributed modules
Kasutuselevõtuks valmis	Out of the box
Kimp	Bundle
Klassifikatsioon (Kasutatakse sõnastike loomiseks Drupal sisuhaldussüsteemis. Selle abil saab sisu klassifitseerida)	Taxonomy
Kohandatud	Custom
Kohandatud moodulid	Custom modules
Laadileht	Stylesheet
Mall	Template
Mobiilisõbralik	Mobile-friendly
Olem	Entity
Pistikprogramm	Plugin
PDO (PDO on kerge ja järjepidev viis andmebaasidesse sisenemiseks. PDO on nagu andmetesse sisenemise kiht, mis kasutab ühtset rakendusliidest)	PHP Data Objects
Plokk (Kasutatakse sisu paigutamiseks Drupal sisuhaldussüsteemis)	Block
Raamistik	Framework
Rakendusliides	API
Silt	Tag
Teek	Library
Teema	Theme
Teema kiht	Theme layer
Teema mootor	Theme engine
Tuum	Core
Tuumamoodulid	Core modules
Vahemälu	Cache

Sissejuhatus

Tänapäeval on üpris tavaline, et veebilehe loomisel kasutatakse sisuhaldussüsteemi, mis võimaldab arendusprotsessi käigus loodud lahenduse sisu hiljem kliendil iseseisvalt hallata. Üheks tuntumaks sisuhaldussüsteemiks on Drupal. Selle edu on tingitud suured, viimasele tehnoloogiale ja tulevikule keskendunud uuendused, mida iga versiooniuuendus kaasa on toonud. Selle tulemusel on Drupal püsinud läbi aastate asjakohane (Buytaert, 2013).

Drupali värskem versioon on hetkel Drupal 8, mille kasutamissagedus püsib küll tõusuteel, kuid mis jääb kasutamise statistikas erinevatel põhjustel endiselt alla eelviimasele versioonile, milleks on Drupal 7. Sellest tulenevalt saab järeldada, et sisuhaldussüsteemi uue versiooni kasutuselevõtuks võib kuluda päris palju aega.

Käesoleva bakalaureusetöö eesmärgiks on luua eesti keeles põhjalik ülevaade Drupal 8 omadustest ja uutest võimalustest. Antud töö peaks tegema Drupal 8 uued võimalused kasutaja jaoks arusaadavamaks ja reaalse ülemineku Drupali uuele versioonile kiiremaks ning sujuvamaks. Seejuures peaks töö põhjendama ära uue versiooni vajalikkuse ning eelised vanema versiooni ees. Tegemine on ülevaatliku rakendusuringuga, mis põhineb olemasolevatel materjalidel ning mida toetavad töö käigus kogutud andmed.

Töö käigus kaardistan põhilised uuendused Drupali uues versioonis ning võrdlen neid eelmise versiooniga. Lisaks kirjeldan, kuidas erineb Drupal 8 kasutamine Drupal 7 omast ning demonstreerin, kuidas erineb valitud ülesannete lahendamine sisuhaldussüsteemi erinevatel versioonidel.

Töö eeldab teadmisi *HTML*, *CSS* ja *PHP* programmeerimiskeeltest ning sisuhaldussüsteemi Drupal baastadmisi.

Töö on suunatud eelteadmisi arvesse võttes eelkõige arendajatele, kuid see sobib lugemiseks ka kõigile neile, kes vajavad põhjalikku ülevaadet Drupali uute omaduste ja võimaluste kohta.

1. Metoodika

Töö jaoks informatsiooni kogumiseks ja võimalikult hea võrdluspildi saamiseks paigaldan Drupal sisuhaldussüsteemi mõlemad võrreldavad versioonid identsetesse keskkondadesse ja kasutan mõlemat versiooni nende võimaluste piires samaaegselt.

Seda tehes saan esile tuua muudatused, mis erinevate versioonide vahel kattuvad, ja ära kaardistada ka Drupal 8 kui sisuhaldussüsteemi uuema versiooni spetsiifilised uuendused.

Tööd alustan Drupal 8 paigaldamisega ja selle tehnilistest andmetest üldise ülevaate loomisega. Järgmiseks vaatlen kasutajaliidest ning keskendun visuaalselt nähtavatele uuendustele ja juurdepääsetavusele. Peale seda loon ülevaate sisuhaldussüsteemi sisemisest arhitektuurist ja muutustest struktuuris, tuginedes eelkõige olemasolevatele materjalidele. Seejärel uurin valikuliselt lähemalt just selles versioonis lisandunud konkreetseid moduleid, millest igaüks vastutab erinevate uute võimaluste eest, mida Drupal 8 pakub.

Valiku töös kajastatud informatsiooni kohta teostan niimoodi, et töös saaksid kirjeldatud vaid tähtsaimad uuendused sisuhaldussüsteemis Drupal 8, mille olulisusest lähtun olemasolevatest materjalidest tehtud järelduste põhjal (Byron, Everything You Need to Know About the Top Changes in Drupal 8, 2015). Lisaks tuginen ka isiklikule kogemusele ja keskendun sellele, et töös kirjeldatud teemad sisaldaks informatsiooni muudatustest sisuhaldussüsteemi erinevates kihtides ning et töö ei keskenduks vaid muudatustele sisuhaldussüsteemi ühes konkreetses piirkonnas. Sellest tulenevalt loob töö lisaks spetsiifiliste uuenduste ja muudatuste kirjeldamisele ka üldpildi sisuhaldussüsteemi uuest versioonist endast.

Selleks, et luua tähtsaimatest uuendustest põhjalik ülevaade, koostan nende kasutamisest ka elulisi näiteid ja rakendan neid nii, et need väljendaks komponentide üldlevinumaid kasutusjuhte. Näidete koostamisel lähtun sellest, et need oleksid võimalikult primitiivsed ja demonstreeriks lihtsustatud kujul kõigest komponendi põhifunktsionaalsust ning nende koostamisel lähtun eelkõige isiklikust kogemusest ja Drupal 8 sisuhaldussüsteemi ametlikust dokumentatsioonist (Drupal 8 documentation, kuupäev puudub).

2. Paigaldamine ja tehnilised andmed

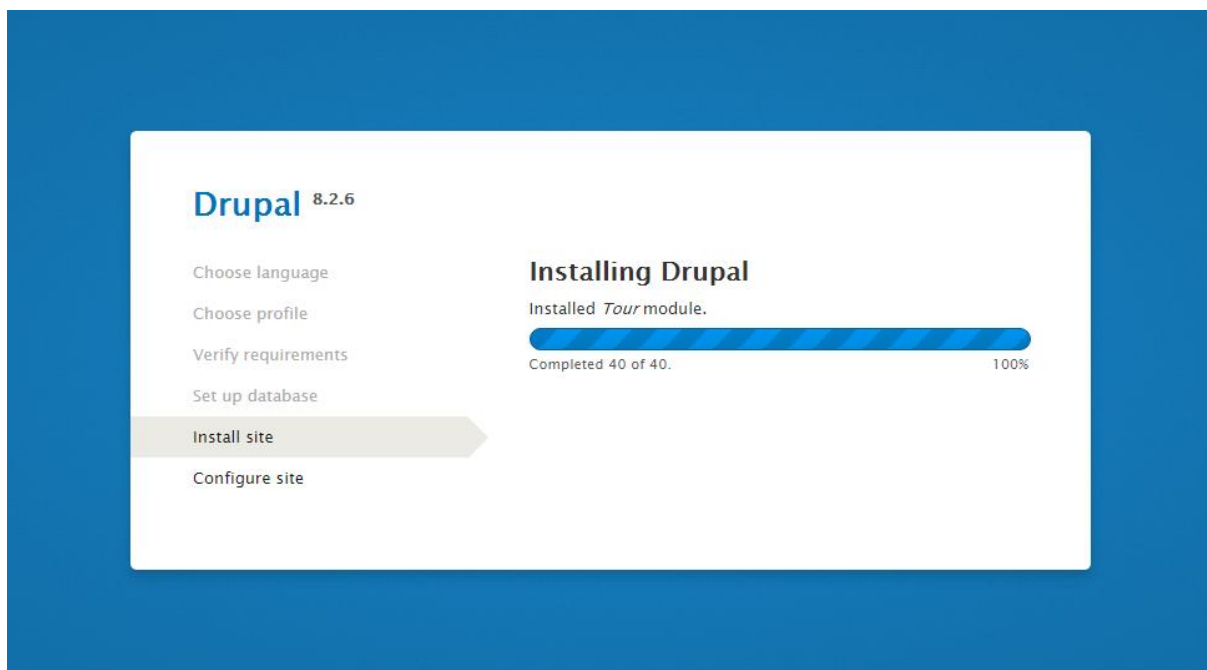
Drupal sisuhaldussüsteemi kasutamiseks tuleb see kõigepealt paigaldada. Antud bakalaureusetöös kasutan 05.02.2017 seisuga viimast saadaval olevat Drupali versiooni, mis Drupal 8 puhul on 8.2.6 ja Drupal 7 puhul 7.54.

Drupal 8 sisuhaldussüsteemi saab sarnaselt oma eelkäijale paigaldada igasse veebiserverisse, kus on olemas PHP. Drupal 8 toimimiseks on vajalik PHP versioon 5.5.9 või uuem. Drupal 7 puhul on nõutud PHP versioon 5.2.5, kuid soovituslik on 5.4 (vt tabel 1) (Overview, 2016).

Tabel 1. Tehnilised nõuded Drupali paigaldamiseks veebiserverisse

	Drupal 8	Drupal 7
Andmebaas	<ul style="list-style-type: none">• MySQL 5.5.3/MariaDB 5.5.20/Percona Server 5.5.8 või uuem koos PDO toega ja InnoDB-ühilduvusega• PostgreSQL 9.1.2 või uuem koos PDO toega• SQLite 3.7.11 või uuem	<ul style="list-style-type: none">• MySQL 5.0.15/MariaDB 5.1.44/Percona Server 5.1.70 või uuem koos PDO toega• PostgreSQL 8.3 või uuem koos PDO toega• SQLite 3.3.7 või uuem
PHP	5.5.9 +	5.2.5 + (soovituslik 5.4)

Drupali paigaldamise protsess on läbi erinevate versioonide jäänud ülesehituselt sarnaseks – tuleb luua andmebaas, täita vormis ära paigaldamiseks nõutud väljad ja algatada paigaldamine. Paigaldamise käigus seatakse üles Drupali tuum (ingl k *core*) ja selle juurde kuuluvad moodulid (vt joonis 1).



Joonis 1. Drupal 8 paigaldamine

Drupal 8 paigaldamine võtab küll mõnevõrra rohkem aega kui Drupal 7 paigaldamine, kuid see on ka põhjendatud, sest Drupal 8 kogumaht ja paigaldatavate failide hulk on oluliselt suurem (vt tabel 2).

Tabel 2. Drupal 8 mahu suuruse ja failide koguse võrdlus versiooniga Drupal 7

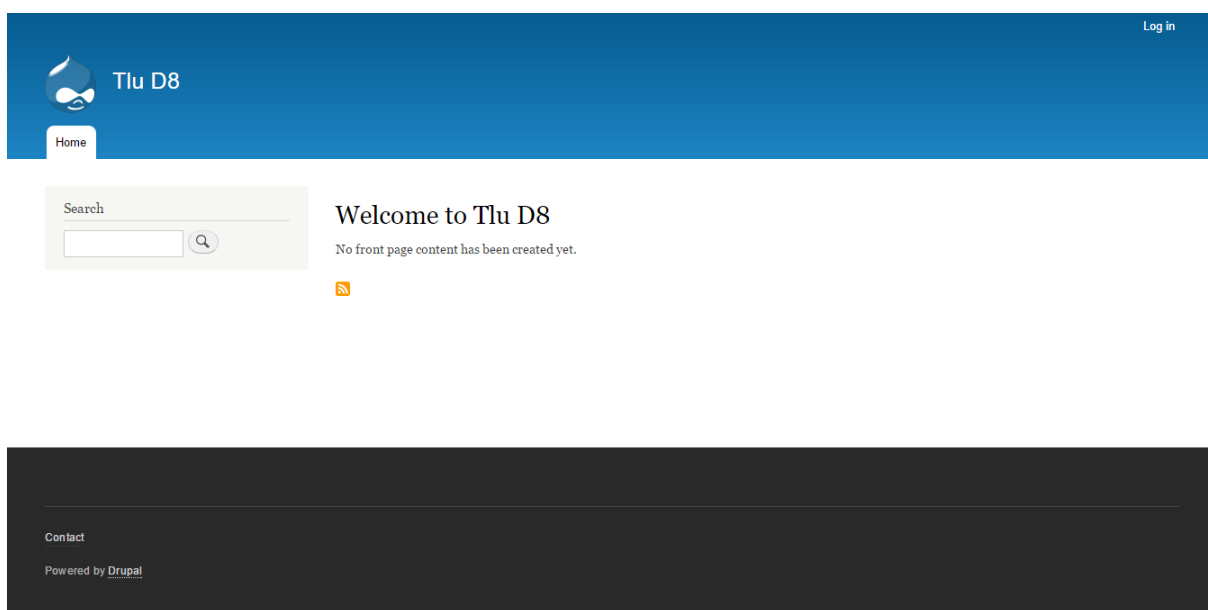
	Drupal 8	Drupal 7
Suurus kokku pakitud kujul enne paigaldamist	11.5 MB	3.12 MB
Suurus kettal	72.4 MB	13.5 MB
Failide hulk	12 628 faili, 3820 kausta	1110 faili, 152 kausta

Suurem failide hulk tuleneb sellest, et mitmed moodulid ja rakendusliidesed (ingl k *API*), mis on Drupal sisuhaldussüsteemiga veebirakenduse arendamisel vajalikud ja mis tuli Drupal 7 puhul eraldi lisada, on nüüd Drupal 8 puhul juba vaikimisi tuumas olemas.

Kokkuvõttev koondtabel kõigist töö raames käsitletud erinevustest sisuhaldussüsteemi võrreldavate versioonide vahel on lisades (vt lisa 1).

3. Kasutajaliides

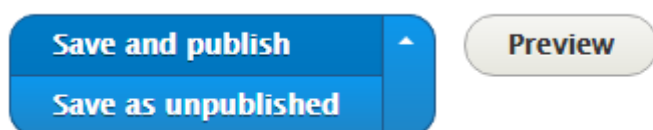
Peale Drupal sisuhaldussüsteemi paigaldamist on mõlema versiooni puhul nähtav Drupali vaikimisi avaleht. Drupal on säilitanud oma üldise stiili, mis on juba varasematest versioonidest tuntuks saanud. Tänu minimaalsetele muudatustele kasutajaliideses on Drupal 8 visuaalne pool korralikum ja viisakam (vt joonis 2).



Joonis 2. Drupal 8 vaikimisi avaleht

Ka sisuhalduse pool on omandanud peamiselt graafilisi uuendusi, mille juures sisu haldamise loogika on jäänud samaks.

Eraldi tasub ära märkida, et Drupal 8 sisuhaldussüsteemis on ära kaotatud eelmise versiooni puhul kasutuses olnud valikukast sisu avalikustamise seadistamiseks, millega seoses on sisu avalikustamine toodud koos sisu salvestamisega ühe nupu alla. See väike muudatus kasutajaliideses teeb sisu salvestamise hetkel paremini arusaadavaks selle, kas antud sisu on ka peale salvestamist avalikult nähtav. Sama nupu alt on endiselt võimalik sisu salvestada ka mitteavalikuna (vt joonis 3).



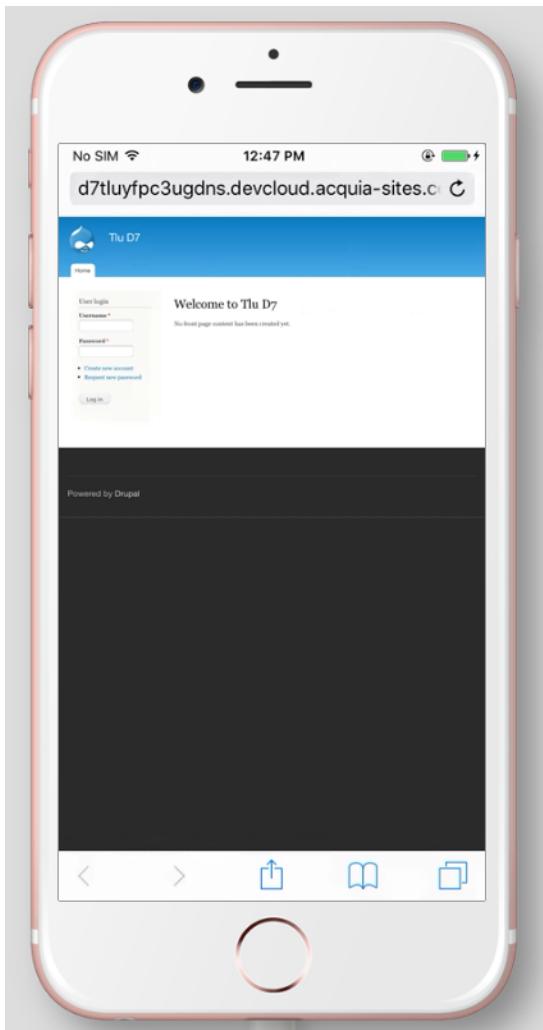
Joonis 3. Drupal 8 puhul toimub sisu salvestamine ja avalikustamine sama nupu alt

Mitmed visuaalsed muudatused on keskendunud sisuhaldussüsteemi kasutusmugavusele ning tingitud eesmärgist teha Drupal kaasaegsemaks, pakkudes tuge ka mobiilsetele seadmetele kohanduvale veebile ja juurdepääsetavusele.

3.1. Mobiilne veeb

Nüüd, kus Drupal 7 välja tulekust on möödas mitmeid aastaid, pole mobiilsele seadmele kohanduv veebisait enam lihtsalt trendikas lahendus, vaid on saanud veebiarenduses üldiselt levinud standardiks.

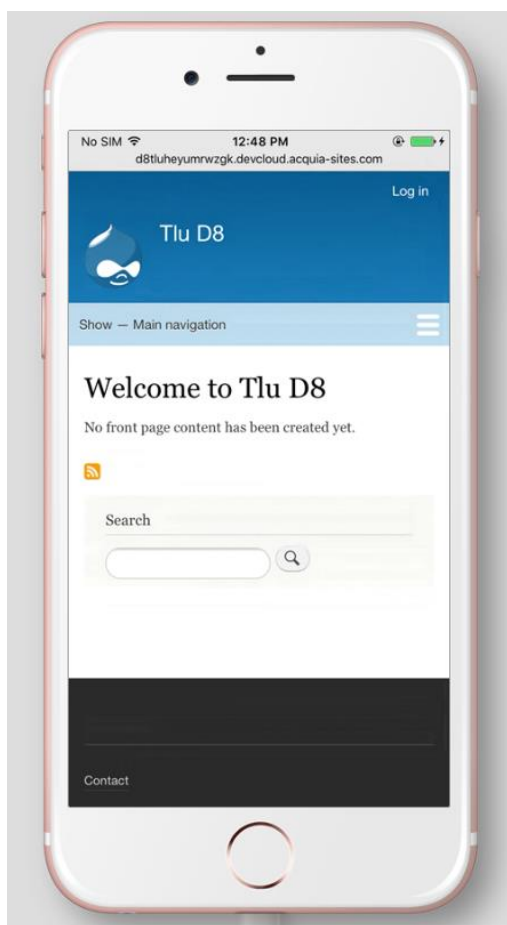
Drupal 7 vaikimisi teemad (ingl k *theme*) ja haldusliides polnud vaikimisi mobiilile kohanduvad (vt joonis 4).



Joonis 4. Drupal 7 avaleht mobiilis

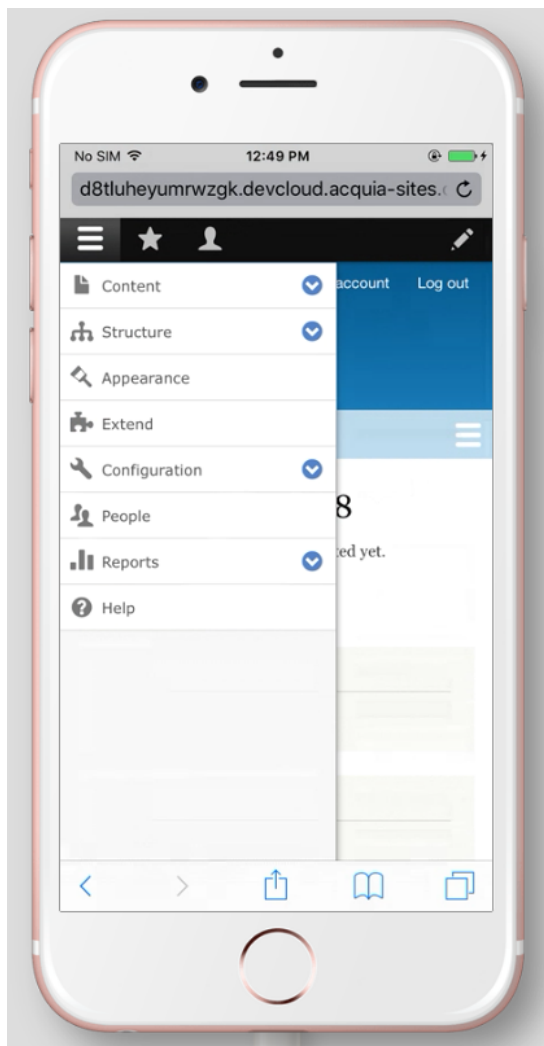
Drupal 8 on disainitud täies ulatuses mobiilset veebi silmas pidades. Drupal 8 pole mitte lihtsalt mobiilisõbralik (ingl k *mobile-friendly*), vaid on vastavalt tänapäevasele heale tavale disainitud eelkõige mobiililset veebi silmas pidades (ingl k *mobile-first*).

Kõik tuumas olevad teemad on mobiilile kohanduvad ja paigutavad väiksema ekraani korral automaatselt ringi lehel leiduvad elemendid nagu menüüd ja plokid nii, et need mahuks ekraanile ära. Kui ekraani suurus on üksteise kõrval olevate elementide kuvamiseks liiga kitsas, paigutuvad muidu horisontaalselt kuvatud elemendid vertikaalselt üksteise alla (Byron, The Ultimate Guide to Drupal 8, kuupäev puudub) (vt joonis 5).



Joonis 5. Drupal 8 avaleht mobiilis

Ka kõik uued võimalused nagu näiteks sisu haldamine läbi visuaalse kasutajaliidese ja uuenduskuuri läbinud haldusmenüü on disainitud toimima ka väikestel ekraanidel (vt joonis 6). See tähendab, et Drupal 8 sisuhaldussüsteemil põhineva veebilehe sisu haldamine on juba vaikimisi, ilma seda spetsiaalselt arendamata, lõppkasutaja jaoks mugavam, sest see on kohe valmis kasutamiseks ka nutiseadmetes ja väiksemate ekraanisuuruste korral.



Joonis 6. Drupal 8 haldusmenüü mobiilis

Tänu sisse ehitatud adapteeruvate piltide (ingl k *responsive images*) toele alaneb kõikide piltide suurus ekraani suuruse muutumisel järk-järgult automaatselt väiksemaks, et mahtuda ka tahvelarvuti või nutitelefoniga ekraanile (Byron, *The Ultimate Guide to Drupal 8*, kuupäev puudub).

Lisaks on oluliselt paranenud sisuhaldussüsteemi kasutajaliidese jõudlus, et positiivne kasutajakogemus püsiks ka mobiilsetel seadmetel. Erinevad suurt ressursi nõudvad JavaScriptil põhinevad lahendused on asendunud mobiilisõbralike alternatiividega. Näiteks terve Drupal 7 tuumas olnud *Overlay* mooduli funktsionaalsus, mis tõi haldusvaate sisu läbi modaalakna veebilehe normaalvaate kohale, on asendatud Drupal 8 puhul lihtsa lingiga haldusmenüüs, mis viib kasutaja haldusvaadetest tagasi veebilehe normaalvaatesse (Byron, *The Ultimate Guide to Drupal 8*, kuupäev puudub).

3.2. Juurdepääsetavus

Tänapäeval on üha tähtsam, et internet oleks kasutatav ka erivajadustega inimeste poolt. Drupal 8 hõlmab endas mitmeid erinevaid uuendusi, mis kindlustavad seda, et Drupal sisuhaldussüsteemil põhinevad veebileheküljed oleks nii universaalsed kui ka ligipääsetavad.

Kui Drupal 7 kasutas XHTML koodi, siis kogu Drupal 8 on ümber kirjutatud kasutades HTML5 koodi (Theming differences between Drupal 6, 7 & 8, 2017). Selle tulemusena on Drupali vaikimisi HTML kood tänapäevasem ja kasutab ära HTML5 poolt pakutavaid uusi võimalusi. Need võimalused on abiks ka veebi juurdepääsetavamaks muutmisel, mis on tänapäeval tihti nõutud ka WCAG (*Web Content Accessibility Guidelines*) standardite järgi. WCAG standardid hõlmavad endas mitmeid spetsiifilisi nõudeid ja soovitusi, mis on jaotatud nende ranguse järgi erinevatele tasemetele ja mida rakendades on võimalik veebilehekülge paremini ligipääsetavamaks muuta.

Drupali vaikimisi teema nimega *Bartik* on kasutajaliidese visuaalsete muutuste käigus omandanud kontrastsemad värvid teksti ja tausta vahel, mis teevad veebilehekülje sisu paremini loetavamaks ka nägemisvaegustega inimestele ja värvipimedatele (vt joonis 7).



Joonis 7. Vasakul on kujutatud Drupal 7 ja paremal pool Drupal 8 vaikimisi teema päist

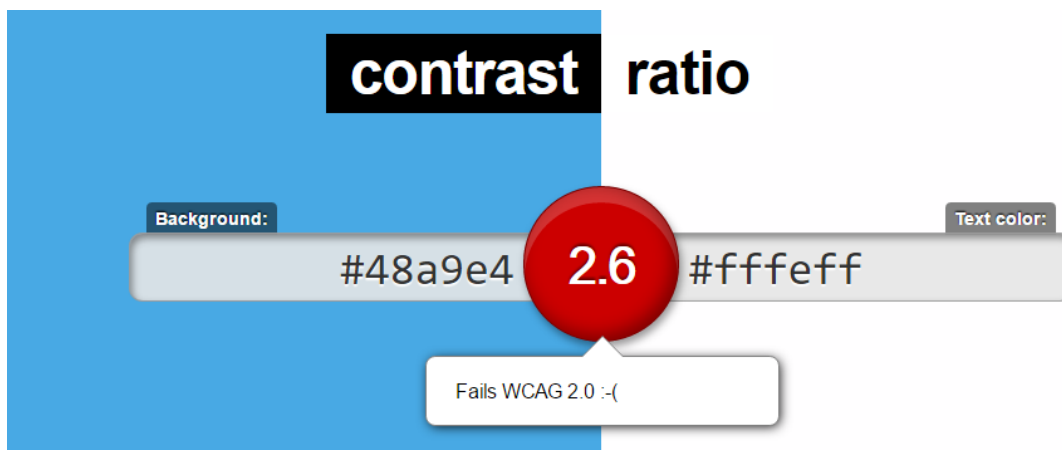
Kontrollin, kas eelmisel joonisel (vt joonis 7) näha olevate värvide kontrastsus vastab ka WCAG nõuetele ja kasutan värvide täpse kontrastsuse kontrollimiseks veebipõhist rakendust *Contrast Ratio*¹.

Selleks, et antud näites kasutatud värvid vastaks WCAG 2.0 ligipääsetavuse nõuetele AA tasemel, peab värvide kontrastsuse suhe olema vähemalt 3:1. Kontrastsuse suhe arvutatakse välja valemiga $(L1 + 0.05) / (L2 + 0.05)$, kus $L1$ on heledama värvi relatiivne erendus ja kus $L2$ on tumedama värvi relatiivne erendus (W3C, kuupäev puudub).

¹ <http://leaverou.github.io/contrast-ratio/>

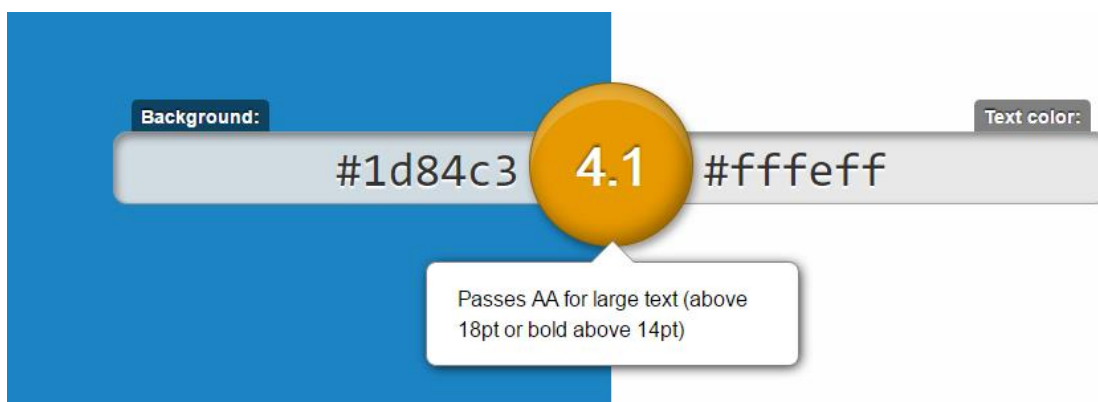
Kui antud näites oleks tegemist väiksema tekstiga, siis peaks minimaalseks kontrastsuse suhteks olema 4,5:1, sest väiksemat teksti on raskem lugeda ja seega on ka nõue selle kontrastsusele rangem (W3C, kuupäev puudub).

Kontrastsuse test andis antud näites kasutatud värvide puhul vastuseks, et Drupal 7 sisuhaldussüsteemi puhul on teksti ja tausta värvide vahel liiga väike kontrastsus ja seega ei vasta Drupal 7 vaikimisi teema puhul lehe päises kasutatud värvid WCAG 2.0 tasemel juurdepääsetavuse nõuetele (vt joonis 8).



Joonis 8. Drupal 7 vaikimisi teema värvid ei vasta WCAG 2.0 juurdepääsetavuse nõuetele


Tänu väikestele muudatustele on Drupal 8 puhul saavutatud aga olukord, kus päises olev pealkiri vastab WCAG 2.0 juurdepääsetavuse nõuetele ja seega on Drupal sisuhaldussüsteemi uuem versioon juba vaikimisi oluliselt paremini kasutatav ka nägemisvaegusega inimeste jaoks (vt joonis 9).



Joonis 9. Drupal 8 vaikimisi teema värvid vastavad WCAG 2.0 juurdepääsetavuse nõuetele

Lisaks nägemisvaegusega inimestele on parandatud veebi kasutatavust ka täiesti pimedate kasutajate jaoks. Drupal 8 pildiväljad nõuavad koos pildiga ka alternatiivse teksti sisestamist, mis on eriti oluline pimedate kasutajate jaoks, sest erinevad ekraanilugejad suudavad teksti pimedale kasutajale inimkeeles ette lugeda ja tänu sellele saavad ka nemad omandada pildil kujutatud informatsiooni (vt joonis 10).


Image



Alternative text *

Please fill out this field.

This text will be used by screen readers, search engines, or when the image cannot be loaded.

 wallhaven-194154.jpg (319.55 KB) [Remove](#)

Joonis 10. Koos pildiga tuleb sisu salvestamisel täita ka pildi juurde kuuluva alternatiivse teksti väli

Juhul, kui arendaja siiski ei soovi mingil põhjusel kasutaja jaoks tekstivälja täitmist kohustuslikuks määrata, on võimalik seda kontrolliv seade *CKEditor* mooduli või pildivälja haldusseadetes siiski ka välja lülitada (Drupal.org, 2016).

Ekraanilugejatega on Drupal 8 sisuhaldusplatvormil põhinev veebilehekülg paremini kasutatav ka tänu semantikale. HTML5 võimaldab kasutada erinevaid elemente, mis lisavad koodile tähendust ning teevad seega koodi sisu paremini arusaadavamaks ja kergemini loetavamaks. Sellisteks elementideks on näiteks `<header>`, `<section>` ja `<nav>`, mis aitavad ekraanilugejal teha lehekülje sisu asukoha kasutaja jaoks arusaadavamaks ja näiteks ``, `<video>`, `<audio>` ning teised multimeedia kuvamiseks kasutatavad elemendid, mis aitavad paremini selgeks teha edasi antava sisu olemuse.

Lisaks on võimalik kasutada tuumas olevat eksperimentaalset moodulit nimega *Inline Form Errors*, mis lubab kõikidel vormidel veateadete kuvamise kohe vastava välja kõrval. Selle mooduli rakendamine teeb kasutaja jaoks selgemaks selle, milline väli vormi kasutamisel täpsemalt veaolukorda esile kutsub ja võimaldab vormi täitval kasutajal keskenduda just konkreetsel väljal oleva informatsiooni parandamisele.

4. Sisuhaldussüsteem

Drupal 8 on võrreldes sisuhaldussüsteemi eelmise versiooniga peaaegu täies mahus ümber kirjutatud ja kasutusele on võetud täiesti uus arhitektuur. Versiooniuuendus on toonud kaasa suured muutused ka struktuuri ja Drupali teema kihti (ingl k *theme layer*).

Paljud süsteemsed uuendused on tingitud soovist lihtsustada Drupal sisuhaldussüsteemi kasutamist ning lühendada selle kasutama õppimiseks kuluvat aega, samal ajal säilitades selle paindlikust.

4.1. Arhitektuur

Üks suurimaid probleeme, mida Drupal 8 loomisega lahendada üritati, oli see, et paljudel firmadel oli raske leida sobivat isikut, kes tunneks Drupalit piisavalt, et sellega igapäevaselt töötada. Drupal 7 oli isikutele, kes Drupal sisuhaldussüsteemiga varem kokkupuudet ei omanud, alguses väga keeruline ja sellest aru saamine koos selle kasutamiseks vajalike oskuste omandamisega võttis tihti liiga palju aega. Selle probleemi lahendamiseks on Drupali uues versioonis kasutatud laiemalt tuntud ja standardiseeritumaid tehnoloogiaid, mis vähendavad Drupali spetsiifiliste teadmiste vajaduste osakaalu, et osata sisuhaldussüsteemi nõutud tasemel kasutada (Buytaert, 2013).

Lahendusena on Drupal 8 läbinud arhitektuuriliselt mitmed muutused ja selle tuumas kasutatakse nüüd modernseid PHP kontsepte ja standardeid nagu Twig, objektorienteeritud programmeerimist ja Symfony raamistikku (ingl k *framework*) (Buytaert, 2013).

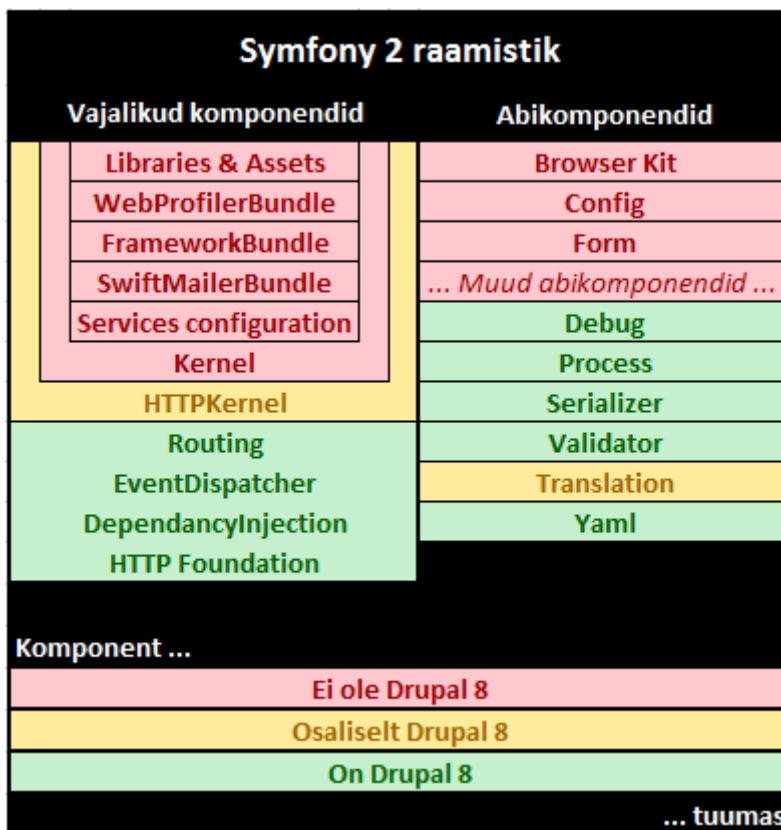
Drupal 8 kasutab täpsemalt öeldes Symfony 2 raamistikku. Symfony on oma olemuselt kogumik PHP komponente ja see on disainitud nii, et see lubaks arendajal efektiivselt ehitada vastavalt vajadusele kohandatud (ingl k *custom*) veebirakendusi.

Kuna Symfony ise ei ole sisuhaldussüsteem, siis kasutab Drupal 8 ainult Symfony tuuma ja laiendab seda nii, et see toetaks Drupali mooduleid. Sellest tulenevalt on Drupal 8 tasakaalustatud süsteem, mis kasutab suures osas Symfony tuuma ja laiendab seda samal ajal paindliku sisuhaldussüsteemiga (Cipix, kuupäev puudub).

Symfony raamistik koosneb mitmest erinevast komponendist, millest osad on süsteemi toimimiseks vajalikud, teised aga mitte. Kogu süsteemi keskmeks olevaks komponendiks on

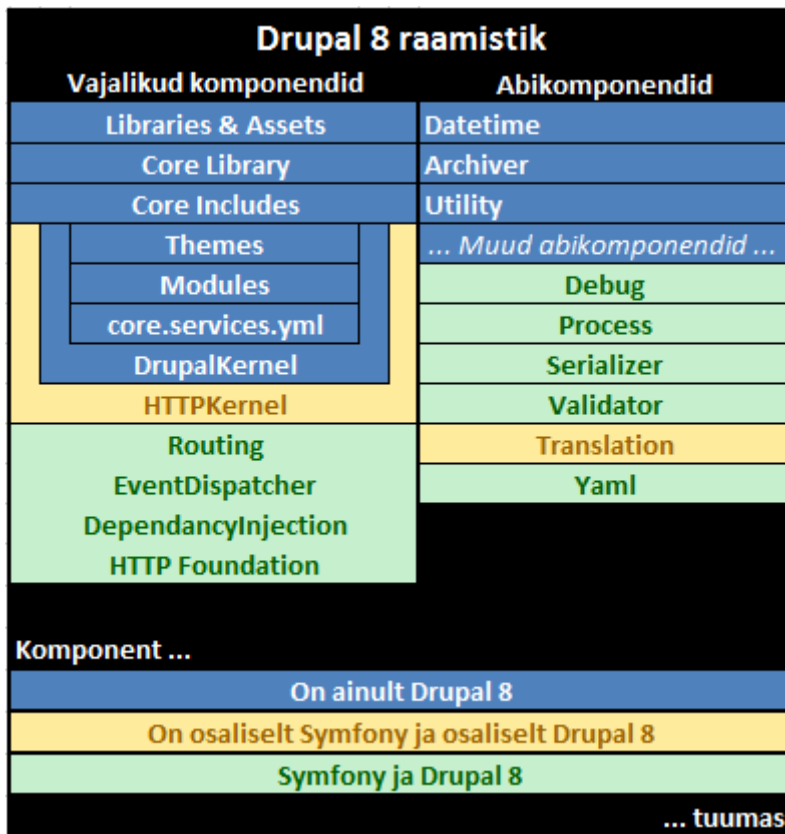
Kernel. Kernel on oma olemuselt põhiline klass, mis kogu keskkonda juhib ja mis haldab HTTP päringut. Arendaja saab Kernelit laiendades kogu süsteemi laiendada ning lisada sinna kohandatud kimpe (ingl k *bundles*). Neid kimpe saab kasutada selleks, et luua funktsionaalsust sarnaselt sellele, kuidas moodulid toimivad Drupalis (Cipix, kuupäev puudub).

Kuna Drupalis peab aga süsteemi tuum toimima veidi teistmoodi, siis kasutab Drupal 8 vaid osaliselt Symfony komponente ja mitmed Symfony tuumas vaikimisi olevad komponendid pole seega Drupal 8 puhul kasutuses (vt joonis 11).



Joonis 11. Symfony raamistik ja Drupal 8 sisuhaldussüsteemi raamistikuga kattuvad komponendid (Cipix, kuupäev puudub)

Drupal 8 ei laienda Kernelit nagu tavaline Symfony veebirakendus seda teeks. Sama funktsionaalsust pakub Drupal aga rakendades Kerneli kasutajaliidest. Drupal kasutab selle asemel oma enda komponenti – DrupalKernel, mis laeb kogu keskkonna koos saadaval olevate teenuste ja moodulitega veidi testmoodi, kuid teostab päringute käsitlemist samamoodi nagu Symfony Kernel. Lisaks kasutab Drupal 8 omaenda kolmanda osapoole komponente ja tuuma (vt joonis 12) (Cipix, kuupäev puudub).



Joonis 12. Drupal 8 sisuhaldussüsteemi raamistik ja Symfony raamistikuga kattuvad komponendid (Cipix, kuupäev puudub)

4.2. Struktuur

Drupal 8 failistruktuur on loodud nii, et see oleks tunduvalt arusaadavam ka uutele kasutajatele. Kui vaadelda failistrukture võrreldavate Drupali versioonide puhul, siis on kohe näha, et Drupal 8 puhul on failistruktuur kompaktsem ja kogu tuum eksisteerib korralikult eraldi kaustas nimega „core“ (vt joonis 13).

Name	Date modified	Size
core	04.03.2017 12:52	
files	08.02.2017 14:25	
modules	04.03.2017 12:52	
profiles	04.03.2017 12:52	
sites	04.03.2017 12:52	
themes	04.03.2017 12:52	
vendor	04.03.2017 12:52	
.csslintrc	04.03.2017 12:50	1 KB
.editorconfig	04.03.2017 12:50	1 KB
.eslintignore	04.03.2017 12:50	1 KB
.eslintrc	04.03.2017 12:50	1 KB
.htaccess	04.03.2017 12:50	4 KB
.htaccess	04.03.2017 12:50	8 KB
autoload	04.03.2017 12:50	1 KB
composer.json	04.03.2017 12:50	2 KB
composer.lock	04.03.2017 12:50	146 KB
example	04.03.2017 12:52	2 KB
index	04.03.2017 12:52	1 KB
LICENSE	04.03.2017 12:52	18 KB
README	04.03.2017 12:52	6 KB
robots	04.03.2017 12:52	2 KB
update	04.03.2017 12:52	1 KB
web.config	04.03.2017 12:52	5 KB

Name	Date modified	Size
files	09.02.2017 9:49	
includes	04.03.2017 12:49	
misc	04.03.2017 12:49	
modules	04.03.2017 12:49	
profiles	04.03.2017 12:49	
scripts	04.03.2017 12:49	
sites	04.03.2017 12:49	
themes	04.03.2017 12:49	
.editorconfig	04.03.2017 12:49	1 KB
.htaccess	04.03.2017 12:49	6 KB
authorize	04.03.2017 12:49	7 KB
CHANGELOG	04.03.2017 12:49	109 KB
COPYRIGHT	04.03.2017 12:49	2 KB
cron	04.03.2017 12:49	1 KB
index	04.03.2017 12:49	1 KB
INSTALL.mysql	04.03.2017 12:49	2 KB
INSTALL.pgsql	04.03.2017 12:49	2 KB
install	04.03.2017 12:49	1 KB
INSTALL.sqlite	04.03.2017 12:49	2 KB
INSTALL	04.03.2017 12:49	18 KB
LICENSE	04.03.2017 12:49	18 KB
MAINTAINERS	04.03.2017 12:49	9 KB
README	04.03.2017 12:49	6 KB
robots	04.03.2017 12:49	3 KB
update	04.03.2017 12:49	20 KB
UPGRADE	04.03.2017 12:49	10 KB
web.config	04.03.2017 12:49	3 KB
xmlrpc	04.03.2017 12:49	1 KB

Drupal 8

Drupal 7

Joonis 13. Drupal sisuhaldussüsteemi failistruktuur

Muudatuse põhjuseks on soov vältida reaalsuses väga tihti esinenud olukorda, kus arendajad hakkavad instinktiivselt oma mooduleid ja teemasid lisama sisuhaldussüsteemi tuuma failide juurde, mis pole Drupali arendamisel aktsepteeritud teguviis. Seega on kogu tuuma eraldi kausta viimine teinud oluliselt loogilisemaks selle, kuhu lisada kohandatud mooduleid või teemasid.

Lähemalt sellest, kuhu ja kuidas täpsemalt paigutada kohandatud teemasid ja mooduleid, saab lugeda järgmistes peatükkides.

4.3. Twig teema mootor

Väljastpoolt paistab, et erilist muutust uues Drupali versioonis teemade valdkonnas toimunud pole – endiselt on Drupal 8 vaikimisi teemaks *Bartik* ja ka visuaalselt näeb see välja üpris sarnane sellele, milline ta oli sisuhaldussüsteemi eelmises versioonis. Ka teemade paigaldamise loogika on jäänud samaks.

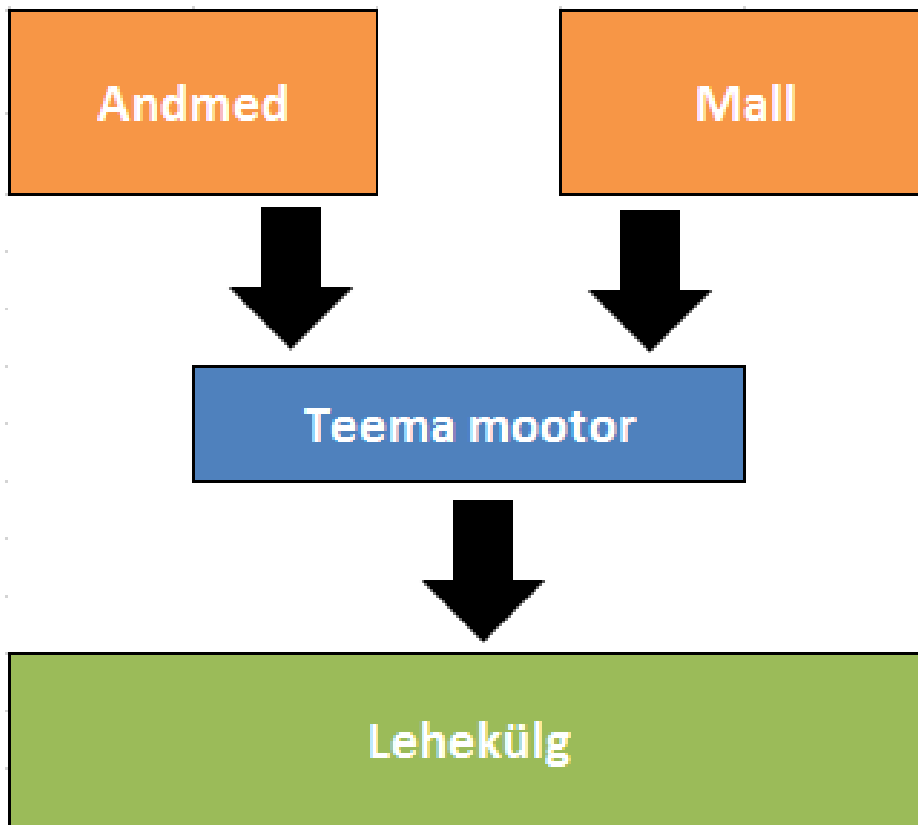
Siiski on Drupal 8 teema kihis toimunud suured uuendused, mis esmapilgul välja ei paista. Need uuendused olid vajalikud, sest sisuhaldussüsteemi eelmise versiooni teema kihis olid mitmed puudused, mis tegid uue loogika kasutusele võtmise hädavajalikuks.

- Kasutajaliidese arendajate (ingl k *front-end developers*) jaoks oli teemade loomine liiga keeruline ja nõudis efektiivseks teostamiseks mitmete spetsiifiliste teadmiste olemasolu.
- Malle (ingl k *templates*) oli väga palju ja sisu kuvamiseks läbi mallide oli liiga palju erinevaid võimalusi. Tänu sellele oli PHP koodiga vähem tuttavatel isikutel väga raske aru saada, kuidas ja millist malli kasutades sisu lehele välja kuvada.
- PHPTemplate teema mootoriga (ingl k *theme engine*) oli võimalik mallides käivitada andmebaasipäringuid ja sellega seoses kustutada ka terve andmebaas. Selline võimalus tegi selle teema mootori ja sellega koos ka kogu teema kihi äärmiselt ebaturvaliseks.

(Tirez, 2014)

Drupal 8 omab täiesti uut teema kihti, mille kõige suuremaks uuenduseks on Twig teema mootori kasutuselevõtt. Teema mootor on lihtsustatult öeldes tarkvara osa, mis kombineerib andmed teema mallidega ja näitab seejärel kasutajale lõplikku HTML väljundit (vt joonis 14)

(Tirez, 2014).



Joonis 14. Teema mootor kombineerib andmed ja malli kokku üheks leheküljeks (Tirez, 2014)

Twig on lihtsustatult moderne PHP koodil põhinev mallide loomise mootor, mis on ka osa Symfony raamistikust.

Kõik PHPTemplate mootoril põhinevad mallid, mis on **.tpl.php* laiendiga failid, on täies ulatuses asendatud Twig mallidega. Twig mallide uueks faililaiendiks on **.html.twig*.

Twigil on mitmeid eeliseid PHPTemplate ees, mis loovad Drupal 8 puhul ka selge eelise Drupal sisuhaldussüsteemi eelmise versiooni ees.

- Twig on palju turvalisem. See on rohkem piiritletud ja seega ei võimalda see teostada käske, mis olid PHPTemplate puhul potentsiaalseteks turvariskideks.
- Twigi puhul on kindlalt eristatud loogika, kuidas vaade kokku pannakse ja vaade ise. See tähendab, et PHP koodi enam mallifailides ei ole.
- Twigi süntaks on paremini loetav, mis teeb kogu koodi paremini arusaadavaks.
- Kuna Twig on populaarsem, siis on see tuntud ka enamuste arendamiseks kasutatavate rakenduste poolt.

- Twig on kasutusel ka mujal, mitte ainult Drupal sisuhaldussüsteemis. Seega nõuab Twig'i kasutamine vähem Drupali spetsiifilisi teadmiseid ja kuna see on laiemalt kasutatud, siis on see ka palju paremini dokumenteeritud.

(Tirez, 2014)

Twigiga tuli kaasa ka tööriist nimega *Twig Debug*. See võimaldab arendajal aru saada, millisest mallist sisu veebilehele kuvatakse. Funktsionaalsuse sisse lülitamiseks tuleb avada „*services.yml*“ fail, mille relatiivne tee on „*sites/default/services.yml*“ ning seal määrata muutuja „*debug*“ väärtuseks „*true*“ (vt koodinäide 1) (Debugging Twig templates, 2017).

```
parameters:
  twig.config:
    debug: true
```

Koodinäide 1. „*services.yml*“ failis tuleb *Twig Debug* tööriista sisse lülitamiseks määrata „*debug*“ muutuja väärtuseks „*true*“

Täpsemat infot Twig'i mallide kasutamisest ja *Twig Debug* tööriista rakendamisest saab lugeda järgmises peatükis.

4.4. Teemad

Teema on kogum failidest, mille põhjal saab Drupal sisuhaldussüsteemi kasutatav veebilahendus oma välimuse.

Muudatused arhitektuuris ja struktuuris mõjutavad otseselt teema loomist Drupal 8 sisuhaldussüsteemile. Teemas kasutatavad failid ja nende kasutamine toimub mitmel viisil teistmoodi, kui sisuhaldussüsteemi eelmise versiooni puhul.

- Kohandatud teemad paigutatakse nüüd „*themes*“ kausta, mitte „*sites/all/themes*“ kausta.
- Info failid on nüüd „*info.yml*“ failid, mis kasutavad YAML süntaksit.
- Teema mootoriks on nüüd Twig, millega seoses on ka kõik mallid laiendiga **.html.twig*, mitte **.tpl.php*.
- „*Template.php*“ faili asemel on nüüd fail, mille nimi on „*teema_nimi.theme*“.

- Laadilehed (ingl k *stylesheets*) ja skriptid defineeritakse nüüd koos teekidega (ingl k *libraries*).

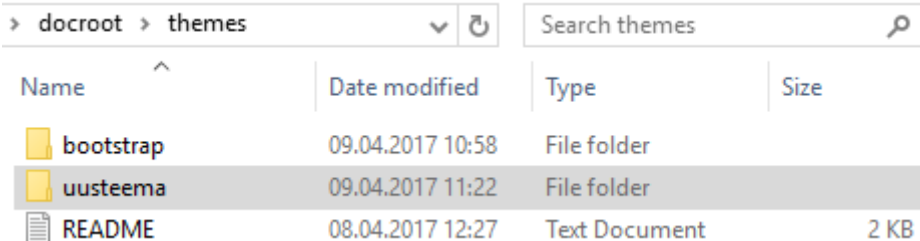
Kuna muudatusi on mitmeid, siis nende toimimisest paremini aru saamiseks loon põgusa ülevaate Drupal 8 sisuhaldussüsteemile teema loomise üldpõhimõtetest, mis aitab paremini aru saada versiooniuuendusega kaasnenud muudatuste mõjust Drupali teema kihile.

Antud näites loon uue alamteema Drupal 8 teemale *Bootstrap*, mis tähendab, et uus loodav teema tugineb olemasolevale teemale nimega *Bootstrap*. Tegemist on populaarse viisiga uue teema loomiseks, sest nii saab taaskasutada juba olemasolevaid komponente ja säästa selle arvelt aega.

Esimese sammuna paigaldan Drupal sisuhaldussüsteemi *Bootstrap* teema², mille alamteemana ma uue teema loon. *Bootstrap* teema paigaldamine toimub samamoodi nagu iga teise teema paigaldamine ja see toimub läbi Drupal sisuhaldussüsteemi haldusliidese.

Kui *Bootstrap* on paigaldatud, kopeerin uue alamteema loomise alustamiseks vajalikud põhifailid eraldi teema kausta. Need failid asuvad *Bootstrap* teema puhul selle kaustas olevas alamkaustas nimega „*starterkits*“. Kopeerin „*starterkits*“ kaustas olevad alamteema failid Drupal sisuhaldussüsteemi teemade kausta nimega „*themes*“ nii, et mu loodava teema kausta relatiivne tee oleks „*themes/uusteema*“, kus „*uusteema*“ on minu poolt loodava teema kausta nimi.

Drupal sisuhaldussüsteemi „*themes*“ kaustas peaks olema nüüd nii varasemalt paigaldatud *Bootstrap* teema, kui ka sinna kopeeritud alamteema kaust, mille nimeks määrasin „*uusteema*“ (vt joonis 15).

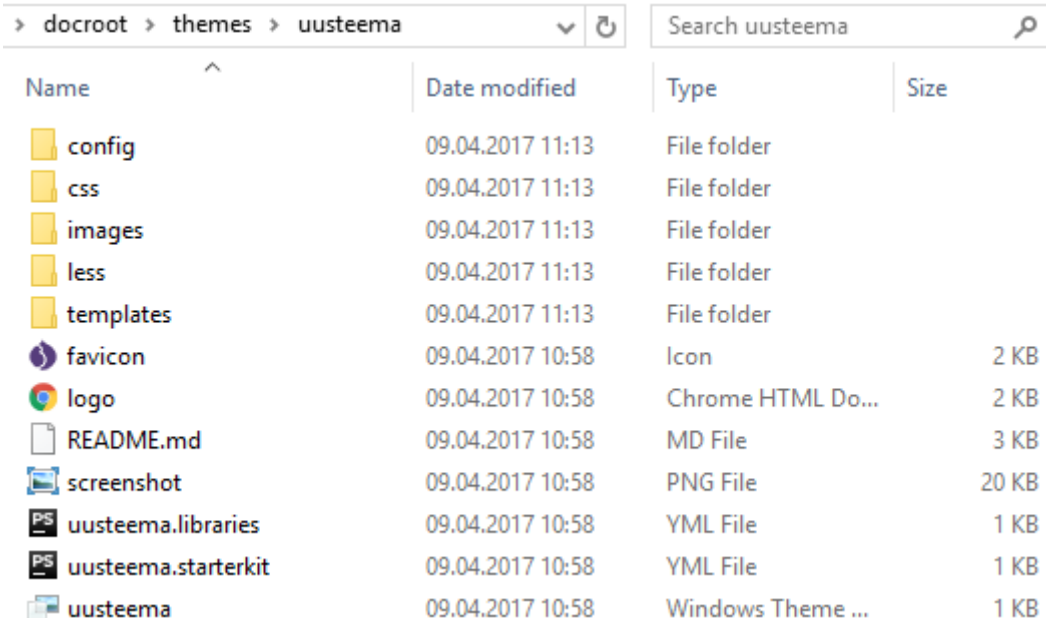


Name	Date modified	Type	Size
bootstrap	09.04.2017 10:58	File folder	
uusteema	09.04.2017 11:22	File folder	
README	08.04.2017 12:27	Text Document	2 KB

Joonis 15. Drupal sisuhaldussüsteemi *themes* kaust sisaldab nüüd nii *Bootstrap* põhiteemat, kui ka minu poolt loodavat uut teemat

² <https://www.drupal.org/project/bootstrap>

Lisaks kausta nimele, tuleb ümber nimetada ka kõik failid, mis on uue alamteema loomiseks vajalikud. Nende failide nimedes tuleb asendada „THEMENAME“ sõnaga „uusteema“, mis on antud näites loodava teema nimeks (vt joonis 16).



Name	Date modified	Type	Size
config	09.04.2017 11:13	File folder	
css	09.04.2017 11:13	File folder	
images	09.04.2017 11:13	File folder	
less	09.04.2017 11:13	File folder	
templates	09.04.2017 11:13	File folder	
favicon	09.04.2017 10:58	Icon	2 KB
logo	09.04.2017 10:58	Chrome HTML Do...	2 KB
README.md	09.04.2017 10:58	MD File	3 KB
screenshot	09.04.2017 10:58	PNG File	20 KB
uusteema.libraries	09.04.2017 10:58	YML File	1 KB
uusteema.starterkit	09.04.2017 10:58	YML File	1 KB
uusteema	09.04.2017 10:58	Windows Theme ...	1 KB

Joonis 16. Nimetan ümber failid, mis sisaldavad nimes teksti „THEMENAME“ nii, et need sisaldaksid selle asemel loodava teema nime "uusteema"

Ka eelneval joonisel (vt joonis 16) näha olevas „config“ kaustas on kaks faili, mille nimedes on „THEMENAME“ tekst tuleb ümber nimetada.

Järgmiseks tuleb ümber nimetada „uusteema.starterkit.yml“ fail nii, et selle uus nimi oleks „uusteema.info.yml“. Täpsemalt teema *.info.yml faili vajalikusest saab lugeda edasises töös.

Kui kõik mainitud failid on ümber nimetatud, siis tuleks asendada ka „uusteema.info.yml“ faili sees kõik tekstid, mis viitavad varem ümber nimetatud failidele (vt joonis 17).

```
uusteema.info.yml x
core: 8.x
type: theme
base theme: bootstrap

name: 'UUS TEEMA'
description: 'See on uus teema, mis tugineb teemale Bootstrap.'
package: 'Bootstrap'

regions:
  navigation: 'Navigation'
  navigation_collapsible: 'Navigation (Collapsible)'
  header: 'Top Bar'
  highlighted: 'Highlighted'
  help: 'Help'
  content: 'Content'
  sidebar_first: 'Primary'
  sidebar_second: 'Secondary'
  footer: 'Footer'
  page_top: 'Page top'
  page_bottom: 'Page bottom'

libraries:
  - 'uusteema/global-styling'
  - 'uusteema/bootstrap-scripts'
```

Joonis 17. „uusteema.info.yml“ faili sisu

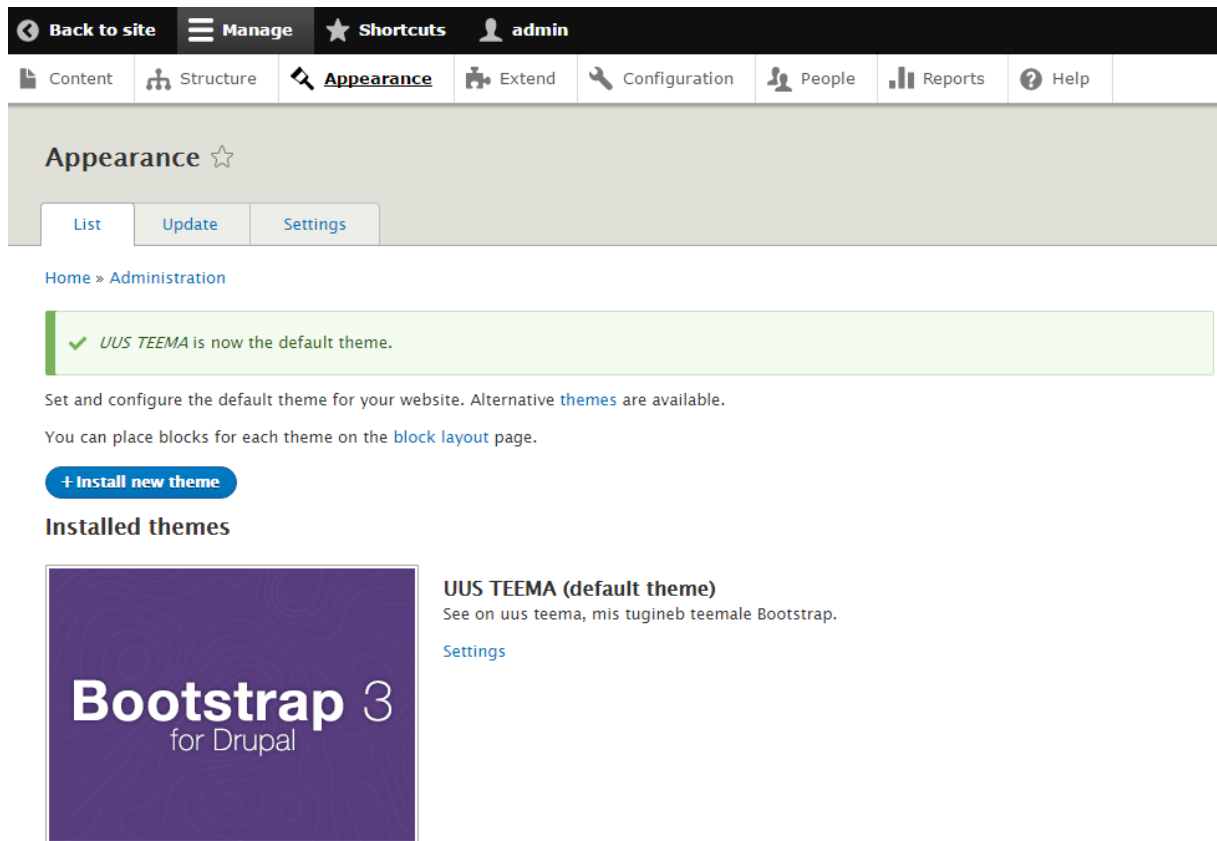
Teema **.info.yml* fail on põhiline fail, mille kaudu Drupal saab informatsiooni teema olemasolu ja selle andmete kohta. Faili kaudu määratakse teema nimi ja omadused ning selle kaudu tehakse teema Drupali poolt äratuntavaks. Teema **.info.yml* failis defineeritakse ka regioonid, kuhu saab paigutada sisu läbi Drupali sisuhaldusliidese. Lisaks on selles failis kirjeldatud ka relatiivne tee teekideni, mida antud teema kasutab.

Bootstrap võimaldab alamteema loomisel oma faile kasutada mitmel eri moel. Võimalik on kasutada *Bootstrap* teema faile läbi võrgu või kasutada neid kohalikest kaustast. Antud näites kasutan faile lokaalselt. Selleks laen alla ka kõige värskemad *Bootstrap* teema failid³ ja paigutan need eraldi kausta nimega „*bootstrap*“, mille loon enda loodava teema kausta nimega „*uusteema*“.

³ <http://getbootstrap.com/getting-started/#download>

Nii saan *Bootstrap* teema faile uuendada ka tulevikus ja nii ei lähe nende failide uuendamisega kaduma spetsiaalselt minu loodavas alamteemas tehtud muudatused.

Nüüd saan oma teema Drupal sisuhaldussüsteemis sisse lülitada. Selleks navigeerin välimuse haldamise vaatesse ja aktiveerin oma teema klakkides „*install and set as default*“. Peale seda peaks teema paigaldatud olema (vt joonis 18).



Joonis 18. UUS TEEMA on paigaldatud ja määratud vaikimisi teemaks

Nüüd saab hakata teemat edasi arendama. Esimese sammuna tuleb valida mall, mida kujundama hakata. Twig mallid asuvad eraldi kaustas, mille nimi on „*templates*“.

Selleks, et aru saada, millist malli konkreetse lehekülje kujundamiseks kasutada, on mõistlik sisse lülitada *Twig Debug*, mis tekitab lehekülje lähtekoodi HTML kommentaaride kujul info kasutatud mallide kohta.

Antud näites kujundan artikli sisutüübi detailivaadet. Selleks loon testsisuna ühe artikli ja peale *Twig Debug* tööriista sisse lülitamist uurin selle lähtekoodi (vt joonis 19).

```

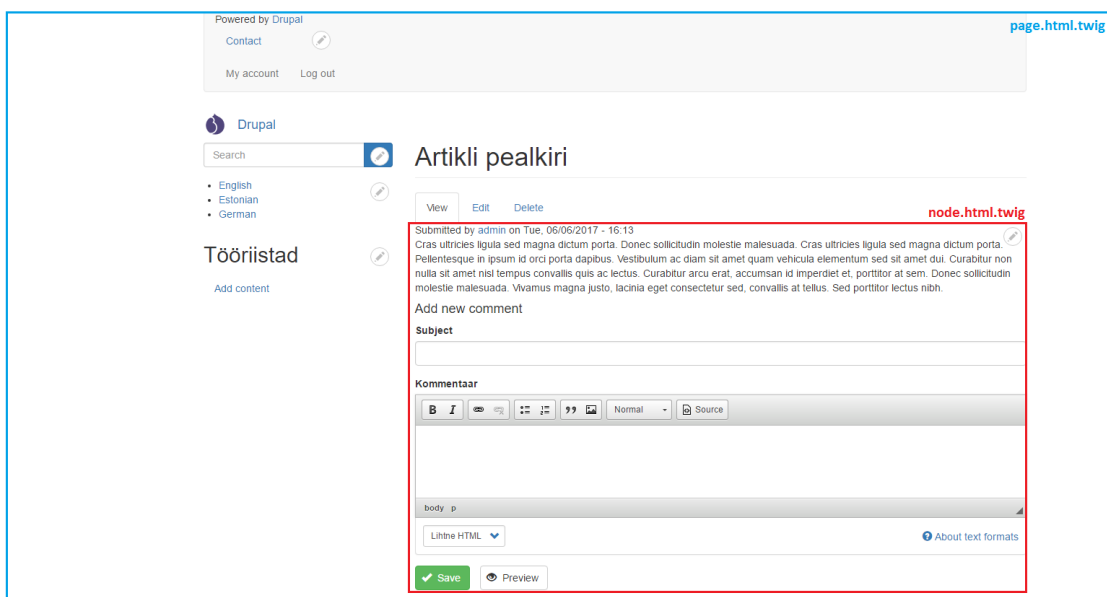
Elements Console Sources Network Timeline Profiles Application Security Audits ADBlock
<!-- BEGIN OUTPUT from 'themes/bootstrap/templates/block/block--system.html.twig' -->
...
<!-- THEME DEBUG -->
<!-- THEME HOOK: 'node' -->
<!-- FILE NAME SUGGESTIONS:
  * node--7--full.html.twig
  * node--7.html.twig
  * node--article--full.html.twig
  * node--article.html.twig
  * node--full.html.twig
  x node.html.twig
-->
<!-- BEGIN OUTPUT from 'themes/bootstrap/templates/node/node.html.twig' -->
▶ <article data-history-node-id="7" data-quickedit-entity-id="node/7" role="article" class="contextual-
region article is-promoted full clearfix" about="/en/node/7" typeof="schema:Article" data-quickedit-
entity-instance-id="0">...</article> == $0
<!-- END OUTPUT from 'themes/bootstrap/templates/node/node.html.twig' -->
<!-- END OUTPUT from 'themes/bootstrap/templates/block/block--system.html.twig' -->
</div>
<!-- END OUTPUT from 'themes/bootstrap/templates/system/region.html.twig' -->
</section>
::after
</div>
::after
</div>
<!-- END OUTPUT from 'themes/bootstrap/templates/system/page.html.twig' -->

```

Joonis 19. Drupal 8 sisuhaldussüsteemiga loodud artikli lehekülje osaline lähtekood koos Twig Debug kommentaaridega

Nagu järgmisel joonisel näha, siis kasutab loodud artikkel terve lehekülje kuvamiseks mallina faili „page.html.twig“, mis on joonistel tähistatud sinise värviga (vt joonis 19 ja 20).

Lisaks paistab lähtekoodi lähemal uurimisel, et lehekülje sisuline osa tuleb mallist nimega „node.html.twig“, mis on joonistel tähistatud punase värviga (vt joonis 19 ja 20).



Joonis 20. Drupal 8 sisuhaldussüsteemiga loodud artikli lehekülj ja artikli sisuline osa

Kopeerin mõlemad mallid enda teema kaustas asuvasse kausta nimega „*templates*“, mis on esimene koht, kust mu teema sisu välja kuvamiseks sellele vastavat malli otsib.

Esimesena avan faili „*node.html.twig*“ ja lisan sinna klassi „*artikkel*“. Teen seda nii, et klass „*artikkel*“ ümbritseks kogu mallis olevat sisu. Peale vahemälu (ingl k *cache*) puhastamist on Twig kommentaarides näha, et nüüd võetakse sisu kuvamiseks kasutatav mall minu teema „*templates*“ kaustast, kus sellele mallile rakendub ka uus klass „*artikkel*“ (vt joonis 21).

```
<!-- BEGIN OUTPUT from 'themes/uusteema/templates/node.html.twig' -->
▼<article data-history-node-id="7" data-quickedit-entity-id="node/7" role=
"article" class="contextual-region article is-promoted full clearfix" about="/
en/node/7" typeof="schema:Article" data-quickedit-entity-instance-id="0">
  ::before
  ▶<div class="artikkel">...</div> == $0
  ::after
</article>
<!-- END OUTPUT from 'themes/uusteema/templates/node.html.twig' -->
```

Joonis 21. Drupal 8 sisuhaldussüsteemiga loodud artikli lehekülje osaline lähtekood peale malli muutmist

Klasside abil saab laadilehti kasutades mallide sisu kujundada. Laadileht „*style.css*“ asub „*uusteema*“ alamkaustas nimega „*css*“.

Kasutan seda klassi, et läbi CSS koodi artikli sisu punaste piirjoontega ümbritseda (vt koodinäide 2).

```
.artikkel {border: 10px solid red;}
```

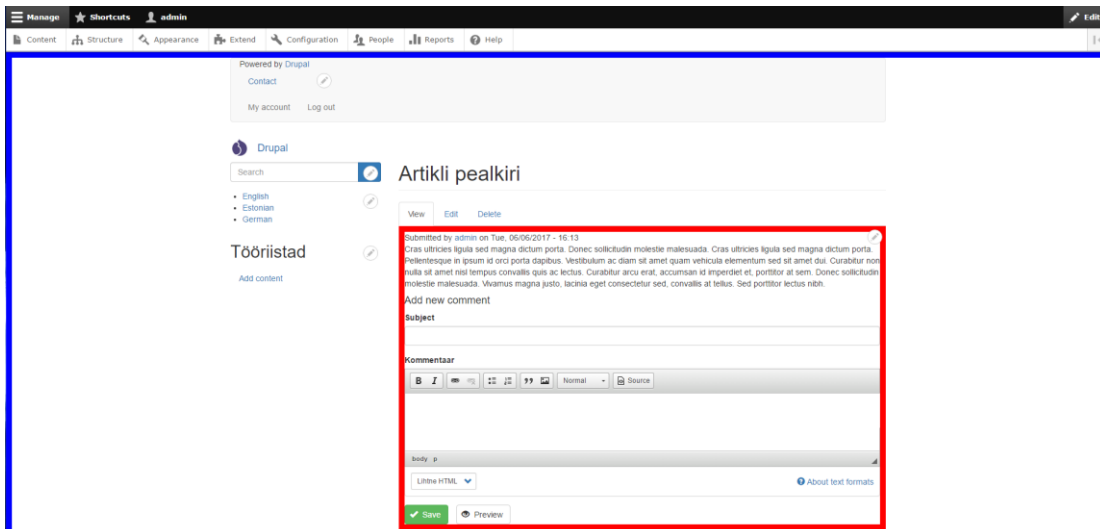
Koodinäide 2. Artikli lehekülje sisule rakenduv klass omab väärtust punast värvi piirjoonte lisamiseks

Nüüd lisan samal viisil klassi ka „*page.html.twig*“ faili ja määran selle klassi nimeks „*lehekylg*“. Selle klassiga lisan leheküljele sinist värvi piirjooned (vt koodinäide 3).

```
.lehekylg {border: 10px solid blue;}
```

Koodinäide 3. Artikli leheküljele rakenduv klass omab väärtust sinist värvi piirjoonte lisamiseks

Nüüd on näha, et ka testimise eesmärgil loodud artikli sisus on kujundus muutunud vastavalt teema mallides olevale koodile (vt joonis 22).



Joonis 22. Artikli lehekülg on ümbritsetud sinist värvi piirjoontega ja artikli sisuline osa on ümbritsetud punast värvi piirjoontega

Selleks, et muudatused rakenduksid ainult soovitud elementidele, tuleb mallide nimetusi veelgi täpsustada. Antud näites rakenduvad punased piirjooned hetkel kõigile sisutüüpidele, sest nad kõik kasutavad vaikimisi malli „*node.html.twig*“.

Selleks, et punased piirjooned ja muud mallis tehtud spetsiifilised kujundusmuudatused rakenduksid ainult artiklile, tuleb malli nime muuta. Malli täpseks nimetamiseks on abiks jällegi *Twig Debug*, mis soovib nimetust uute failide jaoks, mida juba kasutuses oleva malli asemel kasutada (vt joonis 23).

```
<!-- FILE NAME SUGGESTIONS:
 * node--7--full.html.twig
 * node--7.html.twig
 * node--article--full.html.twig
 * node--article.html.twig
 * node--full.html.twig
 x node.html.twig
-->
<!-- BEGIN OUTPUT from 'themes/uusteema/templates/node.html.twig' -->
```

Joonis 23. *Twig Debug* soovib hetkel sisu kuvamiseks kasutusel oleva malli asemel alternatiivseid nimetusi uute mallide loomiseks

Nimetan oma teema „*templates*“ kausta kopeeritud faili „*node.html.twig*“ ümber nii, et selle nimeks oleks „*node--article.html.twig*“. Nüüd kasutab konkreetset muudetud malli ainult artikkel ja teiste sisutüüpide kujundus kasutab sisu kuvamiseks mallina endiselt „*node.html.twig*“ faili.

5. Moodulid

Drupal koosneb moodulitest, mis lisavad sellele erinevat funktsionaalsust. Drupal sisuhaldussüsteemiga vaikimisi kaasas olevad moodulid on osa selle tuumast. Lisaks saab mooduleid ka juurde lisada võimaldades sellega kasutada lisafunktsioone, mida vaikimisi tuumas olevate moodulitega teha ei saanud.

Moodulid saab nende päritolu põhjal jagada kahte gruppi.

- Tuumamoodulid (ingl k *core modules*) – moodulid, mis on Drupali tuumas olemas ja mis paigaldatakse vaikimisi koos Drupal sisuhaldussüsteemiga.
- Kasutajate poolt loodud moodulid (ingl k *contributed modules*) – moodulid, mis on lisatud ja arendatud Drupali kommuuni kuuluvate kasutajate poolt ning avaldatud Drupali ametlikul veebilehel⁴. Siia alla kuuluvad ka kohandatud moodulid (ingl k *custom modules*), mis on arendatud iseseisvalt mingi kindla eesmärgi täitmiseks.

Drupal 8 puhul on moodulitega seotud mitmed muudatused.

Lisandunud on mitmed uued moodulid, mis pakuvad võimalusi, mida Drupal 7 puhul vaikimisi ei eksisteerinud. Sellised moodulid on näiteks:

- *Quick Edit* – Võimaldab hallata veebilehe sisu läbi graafilise kasutajaliidese.
- *Tour* – Võimaldab luua kasutajate jaoks ülevaateid, mis juhendavad kasutajat läbi veebilehekülje haldusliidese ja annavad selle kohta jooksvalt informatsiooni, kasutades selleks teksti sisaldavaid kaste.
- *Breakpoint* – Võimaldab hallata murdepunkte, mis määravad lehekülje teatud laiuse või kõrguse korral elementide kuvamise viisi. Seda moodulit kasutatakse mobiilsete seadmete jaoks kujunduse loomisel.
- *Responsive Image* – Võimaldab HTML5 `<picture>` elementi kasutades pilte paremini kuvada ja nende suurust sõltuvalt ekraanisuuruse muutumisega järk-järgult automaatselt väiksemaks või suuremaks muuta.

Mitmed Drupal 7 peal kasutuses olnud populaarsed moodulid on lisandunud Drupal 8 puhul juba tuuma. Sellised moodulid on näiteks:

⁴ https://www.drupal.org/project/project_module

- *Views* – Võimaldab kuvada veebilehel olevat sisu erinevates formaatides nii-öelda „vaadetena“.
- *CKEditor* – Tekstiredaktor, mis võimaldab HTML elemente kasutades teksti kujundada ja mis kuvab teostatud muudatusi koheselt visuaalselt ka teksti haldurile.
- *Locale* – Võimaldab Drupali kasutajaliidest kuvada erinevates keeltes ja seda tõlkida.
- *Fields** – Drupal 8 tuumas on vaikimisi toetatud moodulite *Link*, *Email*, *Telefon* ja *Date* poolt loodud väljad, mis Drupal 7 puhul eksisteerisid eraldi kasutajate poolt loodud moodulite funktsionaalsusena.

Mitmed Drupal 7 peal olnud moodulid on Drupal 8 puhul tuumast eemaldatud. Mõnel juhul on teatud mooduli funktsionaalsus liidetud mõne teise olemasoleva mooduliga ja eraldiseisev moodul eemaldatud. Mõne mooduli puhul on aga selle funktsionaalsus saavutatav selleks eraldi moodulit kasutamata. Sellised moodulid on näiteks:

- *Overlay* – Moodul võimaldas haldusliidest kuvada hallatava sisuga samal lehel eraldi modaalaknas, et peale haldamist oleks kerge samale leheküljele tagasi navigeerida. Sama funktsionaalsust pakub Drupal 8 puhul kontekstipõhine „tagasi eelmisele leheküljele“ (*back to site*) nupp halduri menüüs.
- *Blog* – Moodul võimaldas luua blogi. Blogi on nüüd võimalik luua kasutades tuumas olevat *Views* moodulit.
- *Fields** – Drupal 7 puhul eraldi moodulitena eksisteerinud *Number* ja *List* moodulid, mis võimaldasid vastavalt luua numbri - ja nimekirjavälju, on nüüd kombineeritud teiste väljade moodulitega. Numbrivälja funktsionaalsus on põhilises *Field* moodulis ja *List* mooduli funktsionaalsus on liigutatud *Options* moodulisse.

Tähtsamatest moodulitest ja nende poolt loodavatest uutest võimalustest Drupal 8 sisuhaldussüsteemis saab lähemalt lugeda järgmistes peatükkides.

5.1. Tuuma moodulid

Drupal 8 tuumas on olemas mitmed uued moodulid, mida sisuhaldussüsteemi eelmisel versioonil tuumas vaikimisi ei olnud, kuid mis olid tegelikult kasutajate poolt lisatud moodulite kujul peale paigaldamist kasutatavad ka seal. Selliste moodulite jõudmine Drupal sisuhaldussüsteemi tuuma on tingitud nende laia kasutatavuse ja konkreetsete moodulite

spetsiifiliste omaduste tõttu, mida nad Drupalile lisasid ja mille olemasolu aja jooksul enamuste sisuhaldusplatvormile loodavate veebilahenduste puhul arendajate jaoks iseenesest mõistetavaks muutus.

Muudatus, mille tulemusel sellised moodulid Drupali tuuma integreeriti, oli sisuhaldussüsteemi kasutajate jaoks kasulik, kuna nüüd toimub selliste moodulite uuendamine koos tuumaga ja seega on kogu protsess ühtlasem, sest uuendused nagunii valdavalt kasutuses olevatele moodulitele ja Drupali tuumale endale teostatakse samaaegselt. Lisaks muudab see kogu süsteemi ka stabiilsemaks, sest suure hulga kasutajate poolt lisatud moodulite asemel on nüüd sellised moodulid integreeritud Drupali tuuma ja eraldi lisanduvate moodulite hulk on selle võrra väiksem.

Lisaks on tuumas ka mõningad uued moodulid, mille poolt loodav funktsionaalsus on küll teatud meetoditega saavutatav ka Drupal 7 peal, kuid paraku on see oluliselt rohkem tööd nõudev kui Drupal 8 sisuhaldussüsteemis olev kasutuselevõtuks valmis (ingl k *out of the box*) lahendus.

Antud peatükk keskendub valikuliselt tähtsaimatele moodulitele, mis on Drupali erinevate versioonide tuumasid võrreldes uuema versiooni puhul sinna lisandunud ning loob nende omadustest ülevaate ja toob näiteid nende kasutamisest.

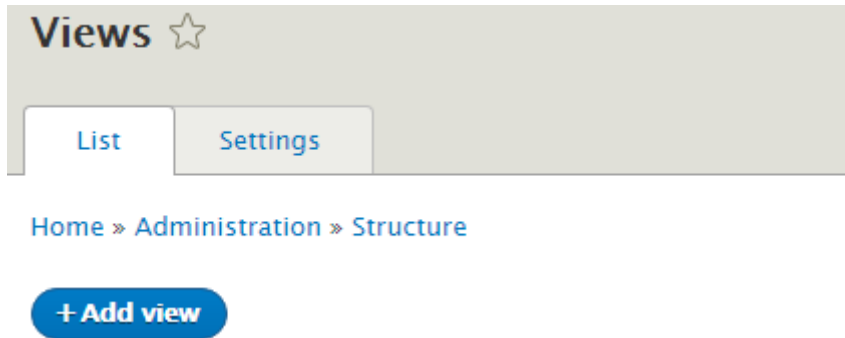
5.1.1. Views

Views on püsinud läbi aastate stabiilselt Drupal sisuhaldussüsteemil põhinevate veebilahenduste enimkasutatavate moodulite hulgas. Mooduli eesmärgiks on kuvada veebilehel olevat sisu erinevates formaatides ehk vaadetes. Seda rakendades saab luua erinevaid nimekirju, pildigaleriisid, slaiderrakendusi ja palju muud ilma selleks koodi kirjutamata.

Drupal 8 puhul on *Views* moodul juba vaikimisi tuumas olemas. Uuendusega on seotud ka näiteks *Blog* mooduli eemaldamine Drupal 8 tuumast, sest võrreldes Drupali eelmise versiooniga, saab nüüd blogi luua ka *Views* moodulit kasutades.

Views mooduli kasutamise demonstreerimiseks Drupal 8 puhul teen läbi kõige tüüpilisema kasutusjuhu, kus moodulit kasutades loon uue vaate lehekülje, mis kuvab nimekirja loodud postitustest. Lisaks loon ka vaate, kuvamaks kõige hiljutisemaid postitusi eraldiseisva plokina.

Uue vaate lisamiseks navigeerin läbi haldusmenüü vaadete haldusesse ja vajutan nupul „add view“ (vt joonis 24).



Joonis 24. Uue vaate lisamine

Algab uue vaate loomine (vt joonis 25).

The screenshot shows the "Add view" form. At the top, there is a header with "Add view" and a star icon. Below this, there is a breadcrumb trail: "Home » Administration » Structure » Views". The form is divided into four sections: "VIEW BASIC INFORMATION" with a "View name" field (marked with a red asterisk) and a "Description" checkbox; "VIEW SETTINGS" with "Show:" (Content), "of type:" (All), "tagged with:" (empty), and "sorted by:" (Newest first); "PAGE SETTINGS" with a "Create a page" checkbox; and "BLOCK SETTINGS" with a "Create a block" checkbox. At the bottom of the form, there are two buttons: "Save and edit" (blue) and "Cancel" (grey).

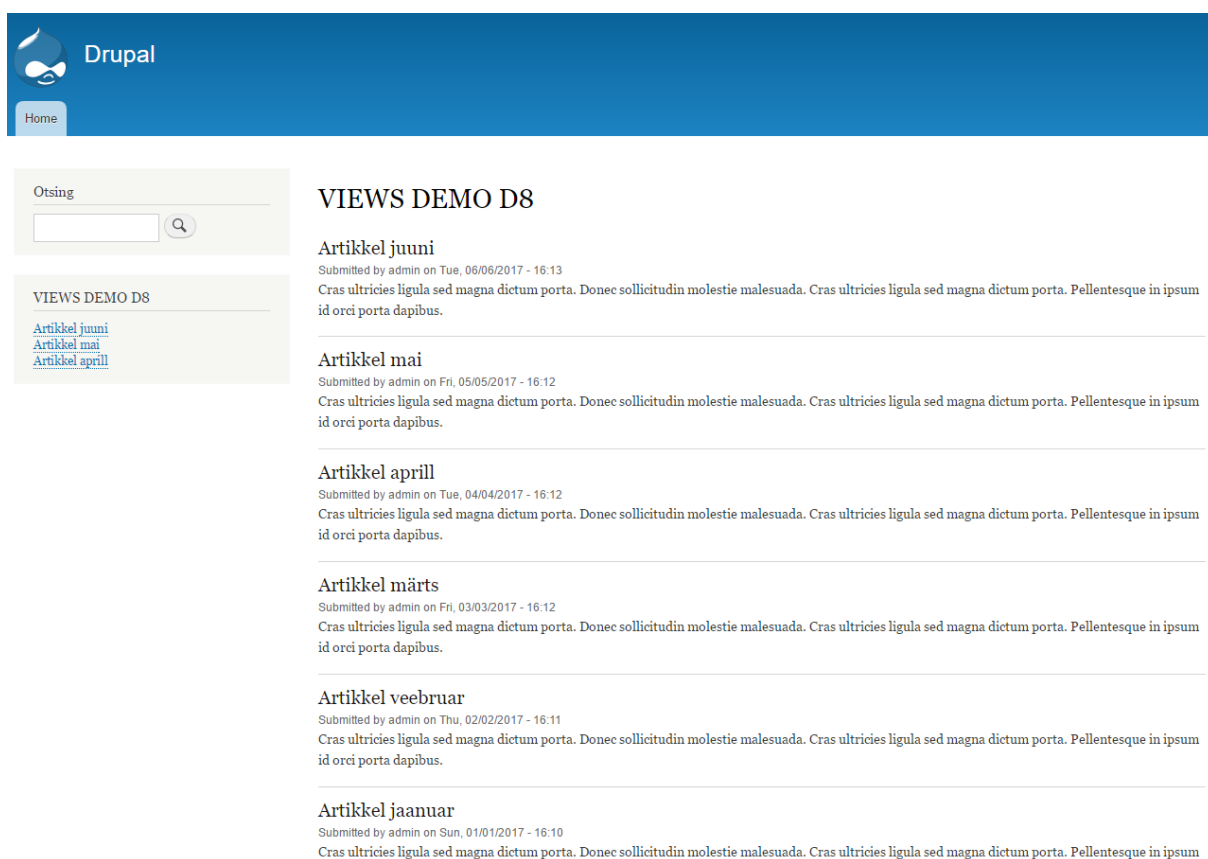
Joonis 25. Uue vaate loomine

Siin on eesmärgiks määrata vaate tüüp ja vaates kuvatav sisu. Esimese sammuna uue vaate loomisel määran vaatele nime, mille järgi antud vaadet haldusvaates ära tunda. Seejärel valin seadistustest kuvatava sisu tüübiks „*article*“ ja jätan sorteerimise vastavalt vaikimisi valikule „*newest first*“, mis tähendab, et kuvatakse välja ainult artiklid ja järjestatakse need nii, et uuemad on kuvatud eespool.

Aktiveerin mõlemad valikukastid „*create a page*“ ja „*create a block*“, mille tulemusel loon vaate nii lehekülje kui ka ploki kujul samaaegselt. Lehekülje vaate loon vaikimisi seadetega ja ploki vaate puhul täpsustan, et välja kuvataks ainult pealkirjad ning maksimaalselt kolm elementi.

Vajutan nuppu „*save and edit*“, mis salvestab vaate ning viib loodava vaate haldamise leheküljele. Seda tehes on loodud uus vaade, mis avaldub kahel erineval viisil (vt joonis 26):

- leheküljena – vaade kuvab kõigi artiklite nimekirja kronoloogilises järjekorras.
- plokina – vaade kuvab kolme kõige hiljutisemat artiklit.



The screenshot shows a Drupal website interface. At the top, there is a blue header with the Drupal logo and a 'Home' button. Below the header, there is a search bar labeled 'Otsing' and a sidebar with a list of articles under the heading 'VIEWS DEMO D8'. The main content area displays a list of articles, each with a title, submission date, and a short description. The articles are listed in chronological order from newest to oldest.

VIEWS DEMO D8

Artikkel juuni
Submitted by admin on Tue, 06/06/2017 - 16:13
Cras ultricies ligula sed magna dictum porta. Donec sollicitudin molestie malesuada. Cras ultricies ligula sed magna dictum porta. Pellentesque in ipsum id orci porta dapibus.

Artikkel mai
Submitted by admin on Fri, 05/05/2017 - 16:12
Cras ultricies ligula sed magna dictum porta. Donec sollicitudin molestie malesuada. Cras ultricies ligula sed magna dictum porta. Pellentesque in ipsum id orci porta dapibus.

Artikkel aprill
Submitted by admin on Tue, 04/04/2017 - 16:12
Cras ultricies ligula sed magna dictum porta. Donec sollicitudin molestie malesuada. Cras ultricies ligula sed magna dictum porta. Pellentesque in ipsum id orci porta dapibus.

Artikkel märts
Submitted by admin on Fri, 03/03/2017 - 16:12
Cras ultricies ligula sed magna dictum porta. Donec sollicitudin molestie malesuada. Cras ultricies ligula sed magna dictum porta. Pellentesque in ipsum id orci porta dapibus.

Artikkel veebruar
Submitted by admin on Thu, 02/02/2017 - 16:11
Cras ultricies ligula sed magna dictum porta. Donec sollicitudin molestie malesuada. Cras ultricies ligula sed magna dictum porta. Pellentesque in ipsum id orci porta dapibus.

Artikkel jaanuar
Submitted by admin on Sun, 01/01/2017 - 16:10
Cras ultricies ligula sed magna dictum porta. Donec sollicitudin molestie malesuada. Cras ultricies ligula sed magna dictum porta. Pellentesque in ipsum id orci porta dapibus.

Joonis 26. Lehekülje vaates on kuvatud kronoloogilises nimekirjas artiklid. Küljeribale on paigutatud vaade, mis sisaldab kolme kõige hiljutisema artikli pealkirju

Kuna *Views* moodul on osa Drupali tuumast, omab see nüüd ka rohkem ülesandeid administratiivsel poolel, kus moodul kontrollib enamuste nimekirjade ja plokkide kuvamist. Tänu sellele on kergem ka veebisaidi administratiivse poole sisu paigutust ja välja kuvamise viisi enda soovi järgi kohandada.

5.1.2. CKEditor

CKEditor moodul Drupal 8 tuumas on inspireeritud Drupal 7 puhul eksisteerivast *WYSIWYG* (*What You See Is What You Get*) moodulist ja see on otseseks järeltulijaks Drupal 7 sisuhaldusplatvormi *CKEditor* moodulile.

Drupal 7 sisuhaldussüsteemis eksisteeris moodul eraldi kasutajate poolt loodud mooduli kujul ja see tuli iga projekti puhul eraldi paigaldada ja seadistada. See oli väga aeganõudev tegevus ja arvestades mooduli populaarsust oli selle iga projekti puhul uuesti teostamine põhjendamatu. Vaikimisi Drupal 7 aga teksti kujundamiseks väljadel võimalusi ei pakkunud (vt joonis 27).



The image shows a screenshot of the Drupal 7 'Create Article' form. At the top, there is a dark blue header with the text 'Create Article' and a small icon. Below the header, there is a breadcrumb trail: 'Home » Add content'. The form has a 'Title' field with a red asterisk indicating it is required. Below the title field is a large, empty text area for the article body. At the bottom of the form, there is a 'Text format' dropdown menu set to 'Full HTML'. To the right of the dropdown is a link for 'More information about text formats'. Below the dropdown, there are two bullet points: 'Web page addresses and e-mail addresses turn into links automatically.' and 'Lines and paragraphs break automatically.'

Joonis 27. Drupal 7 ei omanud teksti haldamiseks vaikimisi WYSIWYG tekstihaldurit

Moodul võimaldab vastavat funktsionaalsust lubavatel tekstiväljadel kujundada teksti kasutades selleks mooduli poolt loodud laiendatud tekstiredaktorit, mille abil saab muuta teksti formaati ja selle vormingut otse läbi haldusliidese (vt joonis 28).



Joonis 28. CKEditor moodul loob teksti kujundamiseks välja kohale tööriba, mille abil saab teksti kerge vaevaga kujundada

5.1.3. Quick Edit

Quick Edit moodul lubab vastavate õigustega kasutajatel hallata lehekülje sisu läbi graafilise kasutajaliidese, ilma selleks eraldi lehekülge külastamata. Antud funktsionaalsus teeb sisu haldamise haldurite jaoks lihtsamaks, sest sisu muutmine samas kohas, kus see välja kuvatakse, on intuitiivsem ja on uute kasutajate poolt kergemini õpitav. Lisaks teeb see väikeste muudatuste tegemise oluliselt kiiremaks, kuna vähendab sisuhalduseks kuluvat aega lehekülgede vahel navigeerimiseks kuluva aja võrra.

Quick Edit mooduli kasutamiseks on vajalik määrata kasutajarollile õigus „*access in-place editing*“, mis kontrollib funktsionaalsuse kasutamist (vt joonis 29).

PERMISSION	ANONYMOUS USER	AUTHENTICATED USER	ADMINISTRATOR
Quick Edit			
Access in-place editing	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Joonis 29. Õiguste määramine kasutajarollidele Quick Edit mooduli funktsionaalsuse kasutamiseks

Lisaks on vajalik õigus „*use contextual links*“, mis kontrollib kontekstipõhiste linkide kasutajale kuvamist. Läbi nende linkide saab kasutajaliidest veebileheküljel avada (vt joonis 30).



Joonis 30. Sisu haldamiseks on sisu juures kuvatud kontekstipõhised lingid

Quick Edit mooduli funktsionaalsuse kasutamiseks tuleb vajutada vastaval lingil selle sisu juures, mida kasutaja soovib hallata.

Kasutajaliides avaneb koheselt ja sisuelementide vahel navigeerides kuvatakse iga konkreetse elemendi juures vastavad valikud selle halduseks ja muudatuste salvestamiseks (vt joonis 31).



Joonis 31. Sisu haldamine läbi graafilise sisuhaldusliidese

5.1.4. Mitmekeelsus

Ajal, mil veebilehed on paljude organisatsioonide peamised infokanalid, on tähtis, et infot saaks edasi anda üle maailma olenemata keelebarjäärist. Sellest tulenevalt on mitmekeelsuse tugi saanud tänapäeval sisuhaldussüsteemide jaoks üldlevinud standardiks.

Drupal 7 ei olnud vaikimisi mitmekeelne ja mitmekeelsuse saavutamine selles sisuhaldussüsteemis oli väga tülikas, sest mitmekeelsuse toe korrektne seadistamine oli aeganõudev ja paljude toimingutega protsess. Juba Drupal 7 sisuhaldussüsteemi paigaldades kuvati lihtsa keelevaliku asemel hulk instruksioone selle paigaldamiseks inglise keelele alternatiivses keeles (vt joonis 32).

Choose language



✓ Choose profile

▶ **Choose language**

Verify requirements

Set up database

Install profile

Configure site

Finished

Follow these steps to translate Drupal into your language:

1. Download a translation from the [translation server](#).
2. Place it into the following directory:

```
/profiles/standard/translations/
```

For more information on installing Drupal in different languages, visit the [drupal.org handbook page](#).

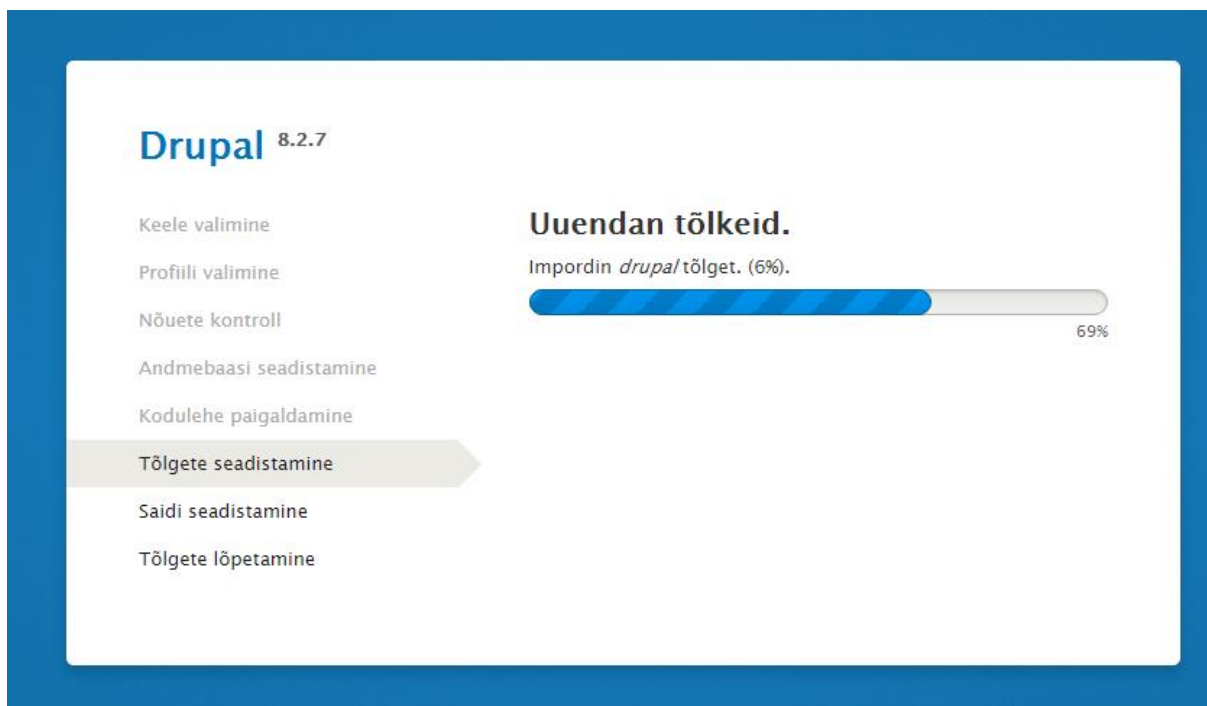
How should the installation continue?

- [Reload the language selection page after adding translations](#)
- [Continue installation in English](#)

Joonis 32. Drupal 7 paigaldamine toimub ainult vaikimisi lubatud inglise keeles

Lisaks ei piisanud kõige vajaliku tõlkimiseks vaid vaikimisi Drupal 7 sisuhaldussüsteemi tuumas olevatest komponentidest, vaid realselt mitmekeelse veebilahenduse loomiseks oli vajalik paigaldada *Internationalization* moodul. See moodul laiendas võimalusi Drupal sisuhaldussüsteemi spetsiifiliste olemite (ingl k *entity*) tõlkimiseks pakkudes tuge ka klassifikatsioonide (ingl k *taxonomy*), plokkide (ingl k *block*) ja muutujate tõlkimiseks. Lisaks lõi see ka võimaluse keelevahetaja abil määrata, millises keeles sisu veebileheküljel kuvatakse. Kogu *Internationalization* mooduli funktsionaalsus on Drupal 8 puhul integreeritud juba selle tuuma.

Drupal 8 on erinevalt oma eelkäijast juba vaikimisi mitmekeelne ja Drupal sisuhaldussüsteemi paigaldamisel inglise keelest erinevas keeles laetakse kasutajaliidese tõlked alla automaatselt (vt joonis 33).



Joonis 33. Drupal 8 paigaldamine toimub kasutaja poolt valitud keeles ja tõlked laetakse alla automaatselt

Lisaks eeldab Drupal 8 automaatselt, et kui paigaldamine toimub mingis muus keeles, kui inglise keel, siis ei ole soovitud inglise keelt loodavas veebilahenduses kasutada. See tähendab, et Drupal 8 sisuhaldussüsteemi puhul ei kuvata inglise keelt igal pool kasutajaliideses valikuna välja, kui paigaldades seda keelt ei valitud. See on loogilisem lähenemine, kui Drupali eelmises versioonis, kus inglise keelt täielikult eemaldada ei saanud, sest sisseehitatud kasutajaliides tugines sellele (Hojtsy, Drupal 8 multilingual tidbits 3: simple language setup, optional English, 2015).

Sisseehitatud inglise keele puuduseks oli Drupal 7 sisuhaldussüsteemis ka selle tõlgitavuse puudumine, mis tähendas, et läbi tõlkeliidese ei olnud võimalik asendada inglise keeles olevaid sõnu mingi muu sõnaga ja asendusi sai teha vaid teiste keelte jaoks (vt joonis 34).

Original text

Log in

Estonian

German

Save translations

Joonis 34. Drupal 7 sisuhaldussüsteemis polnud vaikimisi võimalik inglise keeles olevate tekstide jaoks (*original text*) teha sõnaasendusi. Inglise keeles olevaid tekste (*log in*) sai tõlkida vaid hiljem lisatud keeltesse.

Sõltuvalt projekti olemusest oli teinekord vaja vaikimisi olemasolevate ingliskeelsete tekstide asemel kasutada sünonüüme või alternatiivseid tekste, ent see oli saavutatav vaid teise inglise keele lisamisega eraldi keelena, sest vaikimisi paigaldatud inglise keele jaoks läbi tõkeliidese sõnaasendusi luua ei saanud. Teise inglise keele lisamine tekitas jällegi juurde põhjendamatu keerukust, sest vana inglise keel jäi sõltumata uue lisamisest ikka alles. Sellest tulenevalt tekkisid erinevatesse keelevalikutesse ja sisuhaldussüsteemi tõkeliidesesse duplikaadid, kuna välja kuvati mõlemat inglise keelt.

Alternatiivina sai probleemi lahendamiseks kasutada ka *String Overrides* moodulit, mille abil saab olemasolevad tekstid üle kirjutada, kuid veel ühe lisamooduli paigaldamine sellise ülesande jaoks oli tihti põhjendamatu (Hojtsy, Drupal 8 multilingual tidbits 12: English can now be translated too, 2015).

Ka Drupal 8 puhul ei ole vaikimisi paigaldatud inglise keel tõlgitav (vt joonis 35).

NAME	DEFAULT	INTERFACE TRANSLATION	OPERATIONS
English	<input checked="" type="radio"/>	not applicable	Edit
Estonian	<input type="radio"/>	6091 / 6128 (99.4%)	Edit

Save configuration

Joonis 35. Inglise keel ei ole ka Drupal 8 puhul vaikimisi tõlgitav

Drupal 8 sisuhaldussüsteemis on aga konkreetse keele haldusseadetes võimalik ka saidi vaikimisi inglise keel määrata tõlgitavaks (vt joonis 36).

Language code
en

Language name *

Direction *

Left to right

Right to left

Direction that text in this language is presented.

Enable interface translation to English

Save language

Joonis 36. Inglise keele tõlgitavaks muutmine Drupal 8 sisuhaldussüsteemis („Enable interface translation to English“)

Seadistust valides ja salvestades muutub veebilehekülje käitumine inglise keeles sarnaseks teiste lisatud keeltega ja seega saab muuta ka vaikimisi olemasolevaid ingliskeelseid tekste (vt joonis 37).

▼ FILTER TRANSLATABLE STRINGS

String contains

Leave blank to show all strings. The search is case sensitive.

Translation language **Search in**

English

*** Changes made in this table will not be saved until the form is submitted.**

SOURCE STRING	TRANSLATION FOR ENGLISH
Log in*	<input type="text" value="Sign in"/>
Log in to post new content in the forum.	<input type="text"/>

Joonis 37. Drupal 8 puhul saab tõlkida ka ingliskeelseid tekste, mis tähendab, et ka need tekstid saab vastavalt soovile asendada muude enda poolt defineeritud tekstidega

Erinevalt eelmisest versioonist, kus tõeliselt mitmekeelse veebilahenduse loomiseks tuli paigaldada ja seadistada suur hulk erinevaid mooduleid, on Drupal 8 koondanud kogu oma mitmekeelsust tagava funktsionaalsuse kõigest nelja erinevasse moodulisse, mis teostavad oma eesmärgi palju kasutajasõbralikumalt, kui Drupal 7 moodulid (vt tabel 3).

Tabel 3. Drupal 8 mitmekeelsust tagavad moodulid

Moodul	Selgitus
Language	Lubab kasutajatel seadistada keeli ning määrata neid sisule. See on mitmekeelsuse põhimoodul ja nõutud kõigi teiste mitmekeelsusega seotud moodulite poolt. Drupali paigaldamisel inglise keelele alternatiivses keeles aktiveeritakse see moodul koos <i>Interface Translation</i> mooduliga vaikimisi.
Configuration Translation	Loob kasutajaliidese seadistuste tõlkimiseks. See moodul teeb tõlgitavaks plokid, menüüd, vaated ja muud sarnased olemid. Mooduli funktsionaalsus sarnaneb <i>Internationalization</i> mooduli funktsionaalsusele Drupal 7 puhul.
Content translation	Lubab kasutajal sisu olemeid tõlkida. See moodul teeb tõlgitavaks asjad nagu sisutüübid, klassifikatsioonid ja kommentaarid. Mooduli funktsionaalsus on palju laiem, kui samanimelise mooduli oma Drupal 7 puhul, sest toetab kõiki erinevaid sisu olemeid ja võimaldab tõlkida ka välju.
Interface translation	Võimaldab tõlkida sisseehitatud kasutajaliidest. Mooduli funktsionaalsus on sama, mis Drupal 7 puhul <i>Locale</i> moodulil.

Keeled on lisaks olemasolevatele moodulitele ka koheselt kasutuselevõtuks valmis. Drupal 7 puhul oli vaikimisi aktiveeritud ainult vaikimisi keele tuvastamine, mis tähendas, et rohkemate keelte lisamisel tuli ikkagi eraldi seadistada URL-i põhine keeletuvastus.

Drupal 8 võimaldab seadistada iga administraatori õigustega kasutaja jaoks keele, milles haldusliidest kuvatakse. Tänu sellele uuendusele on halduril võimalik hallata muus keeles olevat sisu läbi haldusliidese, mis on endiselt talle tuttav keeles. Drupal 7 puhul pakkus sama funktsionaalsust moodul nimega *Administration Language*.

Lisaks on Drupal 8 tuumas toetatud ka *Transliteration* mooduli funktsionaalsus, mis Drupal 7 puhul vastutas Unicode tekstis sisalduvate erimärkide muutmiseks US-ASCII sümboliteks. See tähendab, et kohtades nagu andmebaas või kood, kus ei ole toetatud erimärgid nagu näiteks „õ“, „ä“, „ü“, asendatakse need märgid sümbolitega, mis on aksepteeritavad ka andmebaasi ja koodi poolt.

Näiteks sisutüübile eesti keeles nime määrates ei oska Drupal 7 ilma vastava mooduli paigaldamiseta täpitahti asendada nii, et ka masinloetav nimi tegelikku teksti meenutaks (vt joonis 38).

õuna sööv jänes külmetab Machine name: _una_s_v_j_nes_k_lmetab [Edit]

Joonis 38. Drupal 7 ei oska erimärke universaalselt loetavateks märkideks teisendada

Drupal 8 saab sellega juba vaikimisi ilma mingisuguste eelseadistusteta hakkama tänu tuumas olevale funktsionaalsusele (vt joonis 39).

õuna sööv jänes külmetab Süsteemne nimetus: ouna_soov_janes_kulmetab [Muuda]

Joonis 39. Drupal 8 teisendab erimärgid universaalselt loetavateks märkideks

Vastavalt Twig teema mootori kasutusele võtmisest tingitud muudatustele, toimub ka mallides sisu tõlgitavaks määramine teistmoodi (Translation API overview, 2017).

Drupal 7 puhul tuli teksti tõlgitavaks määramiseks ümbritseda see funktsiooniga *t()*, mis määras selles oleva sisu tõlgitavaks (vt koodinäide 4).

```
<?php print t('Hello World'); ?>
```

Koodinäide 4. Teksti tõlgitavaks muutmise Drupal 7 mallis

Drupal 8 sisuhaldussüsteemis ei toeta Twig naturaalselt PHP formaati, mida Drupal 7 puhul toetas PHPTemplate. Sellest tulenevalt toimub tõlkimine mallides vastavalt Twig teemamootori poolt toetatud lahendusele (vt koodinäide 5).

```
{{ 'Hello World'|t }}
```

Koodinäide 5. Teksti tõlgitavaks muutmise Drupal 8 mallis

Lisaks on võimalik kasutada ka silte (ingl k *tags*), mis muudavad kogu nende vahele jääva teksti tõlgitavaks (vt koodinäide 6).

```
{% trans %}  
Hello World  
{% endtrans %}
```

Koodinäide 6. Teksti tõlgitavaks muutmise Drupal 8 mallis kasutades silte

Antud tekst avaldub jällegi tõlkeliideses ja on tõlgitav samamoodi nagu kõik muud tekstid (vt joonis 40).

▼ TÕLGITAVATE SÕNEDE FILTER

Sõne sisaldab

Kõigi sõnede näitamiseks jäta tühjaks. Otsing eristab suur- ja väiketähti.

Tõlke keel: Estonian

Otsitakse valikust: Nii tõlgitud kui tõlkimat

Filtreeri Tühista

⚠ * Sellesse tabelisse tehtud muudatusi ei salvestata enne vormi salvestamist.

LÄHTESÕNE	TÕLGE ESTONIAN JAOKS
Hello World*	Tere Maailm

Salvesta tõlge

Joonis 40. Drupal 8 mallis olev tekst on tõlgitav läbi tõlkeliidese

5.2. Kohandatud moodulid

Mooduli loomine Drupal 8 jaoks hõlmab mitmete eelpool mainitud uute võimaluste rakendamist. Kuigi üldises mooduli loomise protsessis ühtivad mitmed põhimõtted antud kahe versiooni vahel, on Drupal 8 uuendustest tulenevalt siiski mooduli loomise protsess ise ja teostatavad sammud vastavalt muudatustele arhitektuuris ja struktuuris veidi teistsugused.

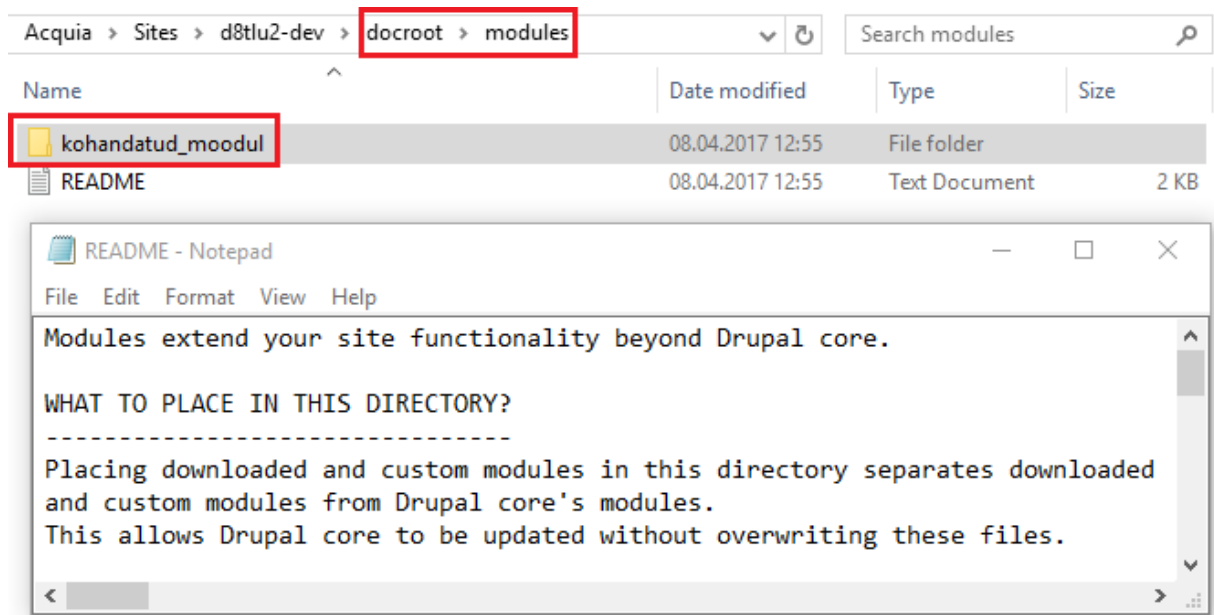
Antud näites loon mooduli, mis tekitab uue lehekülje. Lisaks loon mooduli abil uue ploki, mida saab seejärel läbi haldusliidese lehele paigaldada. Järgneva näite loomisel on abi saadud allikast (McCormick, kuupäev puudub).

Iga mooduli aluseks on **.info.yml* fail, mis on paigutatud loodava mooduli kausta. Enne faili loomist tuleb aga määrata moodulile masinloetav nimi. Seda nime kasutatakse moodulis

mitmetes failides ja funktsioonides ning selle nime abil seotakse loodav moodul Drupaliga. Moodulile masinloetava nime määramisel tuleb järgida mitmeid kindlaid reegleid, mis välistavad nime lugemisest tulenevate vigaste olukordade tekkimist süsteemis:

- nimi peab algama tähega.
- nimes tohib kasutada ainult väikeseid ladina tähti ja alakriipsu.
- nimi ei tohi sisaldada tühikuid.
- nimi peab olema unikaalne. See ei tohi kattuda teise mooduliga, teema nimega või paigaldusprofiiliga, mida on kasutatud antud veebilahenduses.
- nimi ei tohi kasutada spetsiifilisi reserveeritud termineid: *src*, *lib*, *vendor*, *assets*, *css*, *files*, *images*, *js*, *misc*, *templates*, *includes*, *fixtures*, *drupal*.

Vastavalt antud reeglitele loon uue kausta nimega „kohandatud_moodul“ ja paigaldan selle kausta „modules“ (vt joonis 41).



Joonis 41. Mooduli paigutamine Drupal 8 puhul toimub vastavalt dokumentatsioonile kausta *modules*

Kohandatud mooduleid ei ole enam vaja paigutada „sites/all/modules“ relatiivse teega kausta, sest kogu tuum on Drupal 8 puhul eraldi kaustas nimega „core“ ja koos tuumaga paiknevad seal ka kõik tuuma kuuluvad moodulid.

Nüüd saab luua ka eelnevalt mainitud *info.yml* faili ja paigutada selle loodud kausta „kohandatud_moodul“. See fail sisaldab mooduli metaandmeid ja annab Drupal 8

sisuhaldussüsteemile mooduli kohta infot. Kuna ka selle faili nimi peab sisaldama mooduli masinloetavat nime, määrän selle nimeks „kohandatud_moodul.info.yml“ (vt koodinäide 7).

```
name: Kohandatud Moodul
description: Moodul loob uue lehekülje, mis sisaldab teksti „Tere Maailm!“
package: Kohandatud moodulid

type: module
core: 8.x
```

Koodinäide 7. kohandatud_moodul.info.yml

Esimesed failis olevad read määravad moodulile nime ja kirjelduse, mida kuvatakse välja sisuhaldussüsteemi haldusliideses olevas moodulite nimekirjas.

Package väärtus võimaldab mooduleid grupeerida. Näiteks kasutavad tuuma moodulid package väärtust „core“, mis grupeerib haldusliideses kõik tuuma moodulid.

Drupal 8 puhul on nõutud ka uus väärtus „type“, mis määrab laienduse tüübi. Selle rea väärtus võib olla *module*, *theme* või *profile*.

Viimane rida kirjeldab Drupali versiooni, millele antud moodul on mõeldud.

Kõik eelnevad read antud failis on kohustuslikud, kuid lisaks saab antud failis kirjeldada veel mitmeid valikulisi omadusi, millele ma antud näites ei keskendu. Näiteks saab määrata sõltuvused teistest moodulitest, kirjeldada mooduli seadistuste lehekülje asukohta ja peita moodulit haldusliideses olevast moodulite nimekirjast.

Erinevalt Drupal sisuhaldussüsteemi eelmisest versioonist, ei ole mooduli aktiveerimiseks enam kohustuslik *.*module* laiendusega faili olemasolu isegi siis, kui see ei sisalda koodi. Soovi korral võib selle siiski luua. Antud näite puhul oleks selle faili nimeks „kohandatud_moodul.module“.

Peale vahemälu puhastamist peaks uus moodul nüüd olema moodulite nimekirjas nähtav ja paigaldatav (vt joonis 42).

▼ KOHANDATUD MOODULID

<input checked="" type="checkbox"/> Kohandatud moodul	► Moodul loob uue lehekülje, mis sisaldab teksti „Tere Maailm!“
---	---

Joonis 42. Kohandatud moodul on nähtav moodulite nimekirjas

Mooduli abil uue lehekülje loomiseks tuleb luua mooduli jaoks fail, mis sisaldab funktsiooni loodaval leheküljel teksti kuvamiseks. Uue faili nimeks määratakse „*KohandatudmooduliController.php*“ (vt koodinäide 8).

```
<?php

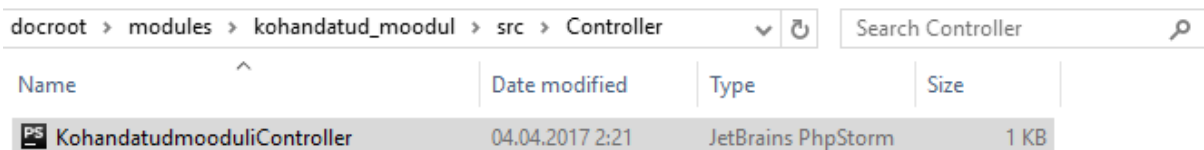
namespace Drupal\kohandatud_moodul\Controller;
use Drupal\Core\Controller\ControllerBase;


class KohandatudmooduliController extends ControllerBase {

    /**
     * Display the markup.
     *
     * @return array
     */
    public function content() {
        return array(
            '#type' => 'markup',
            '#markup' => $this->t('Tere Maailm!'),
        );
    }
}
```

Koodinäide 8. *KohandatudmooduliController.php*

Vastavalt kindla standardi järgi loodud struktuurile, peab loodud fail paiknema kaustas nimega „*Controller*“. Kaust nimega „*Controller*“ peab aga omakorda paiknema kaustas nimega „*src*“. Seega peaks faili „*KohandatudmooduliController.php*“ lõplik relatiivne teekond olema „*modules/kohandatud_moodul/src/Controller/KohandatudmooduliController.php*“ (vt joonis 43).



Name	Date modified	Type	Size
 KohandatudmooduliController	04.04.2017 2:21	JetBrains PhpStorm	1 KB

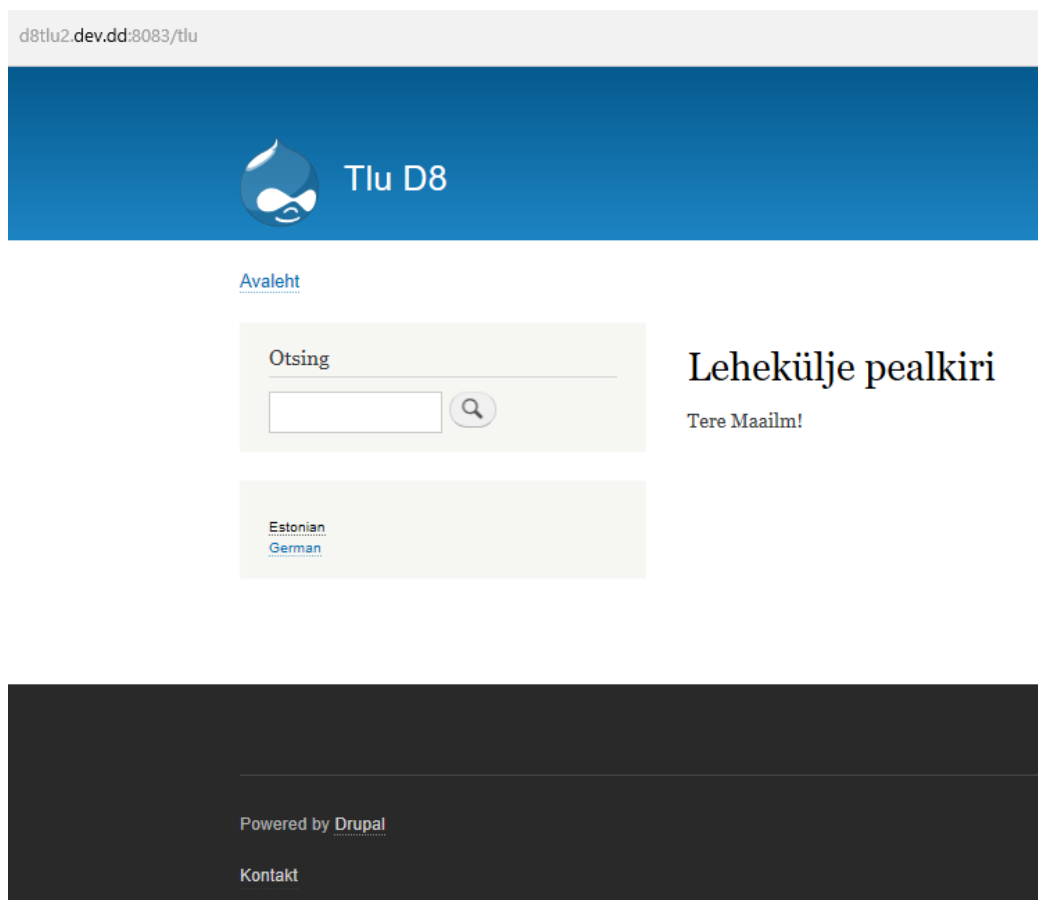
Joonis 43. Faili „*KohandatudmooduliController.php*“ asukoht

Failis oleva koodi toimimiseks on siiski vaja veel ühte faili, mis viitaks sellele failile. Selleks tuleb kausta „kohandatud_moodul“ luua uus fail nimega „kohandatud_moodul.routing.yml“ (vt koodinäide 9).

```
Kohandatud_moodul.content:
  path: '/tlu'
  defaults:
    _controller:
      '\Drupal\kohandatud_moodul\Controller\KohandatudmooduliController::content'
    _title: 'Lehekülje pealkiri'
  requirements:
    _permission: 'access content'
```

Koodinäide 9. kohandatud_moodul.routing.yml

Selle faili abil loob moodul uue relativse teekonna uuele leheküljele, määrab sellele pealkirja ja kuvab seal „KohandatudmooduliController.php“ failiga sinna määratud sisu. Peale vahemälu tühjendamist peaks lehekülg ka Drupalis nähtav olema (vt joonis 44).



Joonis 44. Kohandatud mooduliga loodud lehekülg

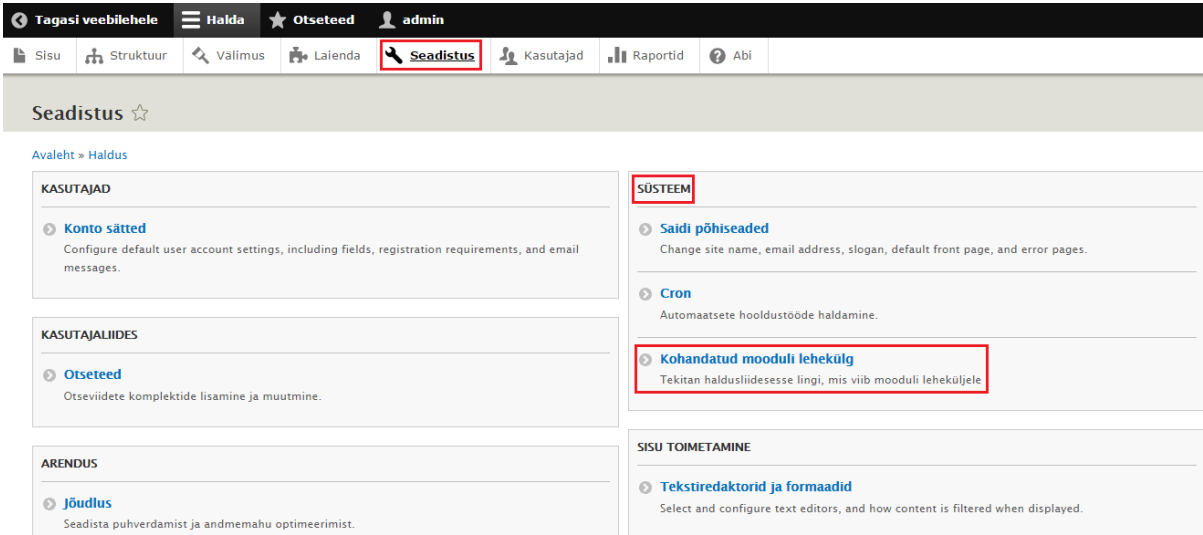
Leheküljele viitamiseks tuleks luua menüülink, mis võimaldaks mooduli poolt loodud leheküljele navigeerida ka läbi Drupal sisuhaldussüsteemi haldusliidese.

Selleks loon uue faili nimega „kohandatud_moodul.links.menu.yml“ (vt koodinäide 10).

```
kohandatud_moodul.admin:
  title: 'Kohandatud mooduli lehekülg'
  description: 'Tekitan haldusliidessesse lingi, mis viib mooduli leheküljele'
  parent: system.admin_config_system
  route_name: kohandatud_moodul.content
  weight: 100
```

Koodinäide 10. kohandatud_moodul.links.menu.yml

Failis määran menüülingile pealkirja ja kirjelduse ning kirjeldan tema asukohta menüüstruktuuris. Antud juhul määran menüülingi nii, et see oleks kuvatud seadistuste leheküljel süsteemi sektsioonis (vt joonis 45).



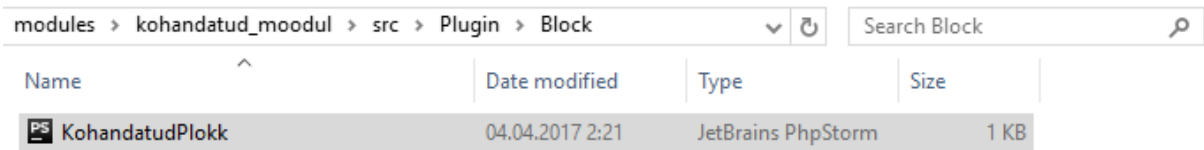
The screenshot shows the Drupal administration interface. The top navigation bar includes 'Tagasi veebilehele', 'Haldja', 'Otseteed', and 'admin'. The main navigation menu has 'Sisu', 'Struktuur', 'Väljumis', 'Lalenda', 'Seadistus', 'Kasutajad', 'Raportid', and 'Abi'. The 'Seadistus' menu item is highlighted with a red box. Below the navigation, the 'Seadistus' page is displayed. The left sidebar shows 'Avaleht > Haldus' and three sections: 'KASUTAJAD' with 'Konto sätted', 'KASUTAJALIIDES' with 'Otseteed', and 'ARENDUS' with 'Jõudlus'. The main content area is divided into two columns. The left column has 'SISTEEM' with 'Saadi põhiseaded' and 'Cron'. The right column has 'Kohandatud mooduli lehekülg' (highlighted with a red box) and 'SISU TOIMETAMINE' with 'Tekstiredaktorid ja formaadid'. The 'Kohandatud mooduli lehekülg' item has a description: 'Tekitan haldusliidessesse lingi, mis viib mooduli leheküljele'.


Joonis 45. Moodulile viitav menüülink seadistuste leheküljel

Mooduli abil ploki loomiseks tuleb kõigepealt luua uus pistikprogramm (ingl k *plugin*). Pistikprogrammid on Drupal 8 sisuhaldussüsteemi puhul uueks võimaluseks kompaksete tükkidena funktsionaalsuse lisamiseks.

Kausta „src“ tuleb luua uus kaust nimega „Plugin“, mis sisaldab omakorda kausta „Block“. Kausta nimi „Block“ väljendab pistikprogrammi tüüpi. Kausta „Block“ loon faili nimega

„KohandatudPlokk.php“, mille lõplik relatiivne tee on „modules/kohandatud_moodul/src/Plugin/Block/KohandatudPlokk.php“ (vt joonis 46).



Name	Date modified	Type	Size
 KohandatudPlokk	04.04.2017 2:21	JetBrains PhpStorm	1 KB

Joonis 46. Faili „KohandatudPlokk.php“ asukoht

Failis „KohandatudPlokk.php“ määratakse ploki informatsiooni. Väärtus *id* kirjeldab ploki kategooriat, väärtus *admin_label* loob ploki tõlkeliidese kaudu tõlgitava nime ja lisaks on failis ka funktsioon ploki teksti kuvamiseks (vt koodinäide 11).

```
<?php

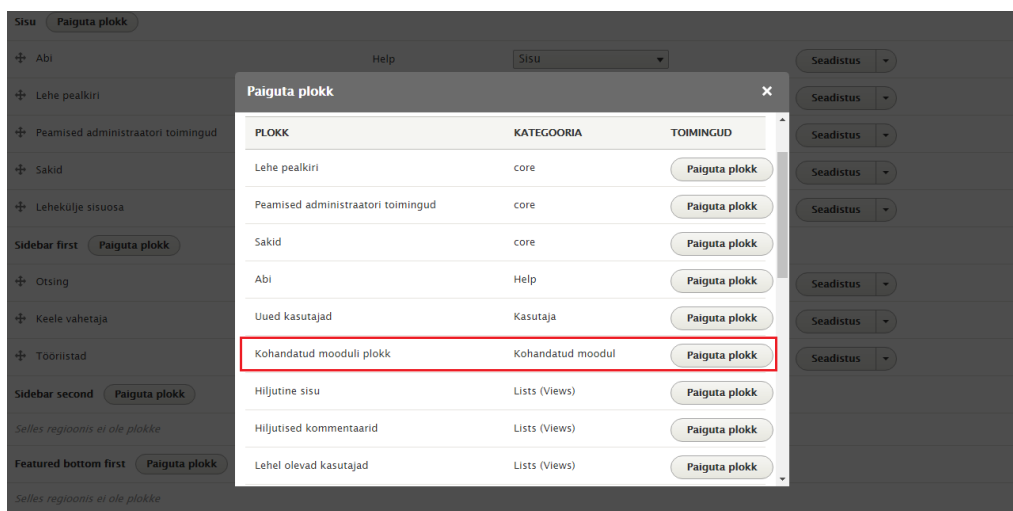
namespace Drupal\kohandatud_moodul\Plugin\Block;
use Drupal\Core\Block\BlockBase;

/**
 * Loon kohandatud ploki.
 *
 * @Block(
 *   id = "kohandatud_moodul",
 *   admin_label = @Translation("Kohandatud mooduli plokk"),
 * )
 */
class KohandatudPlokk extends BlockBase {

    /**
     * {@inheritdoc}
     */
    public function build() {
        return array(
            '#markup' => $this->t('See tekst tuleb kohandatud mooduliga tekitatud plokist!'),
        );
    }
}
```

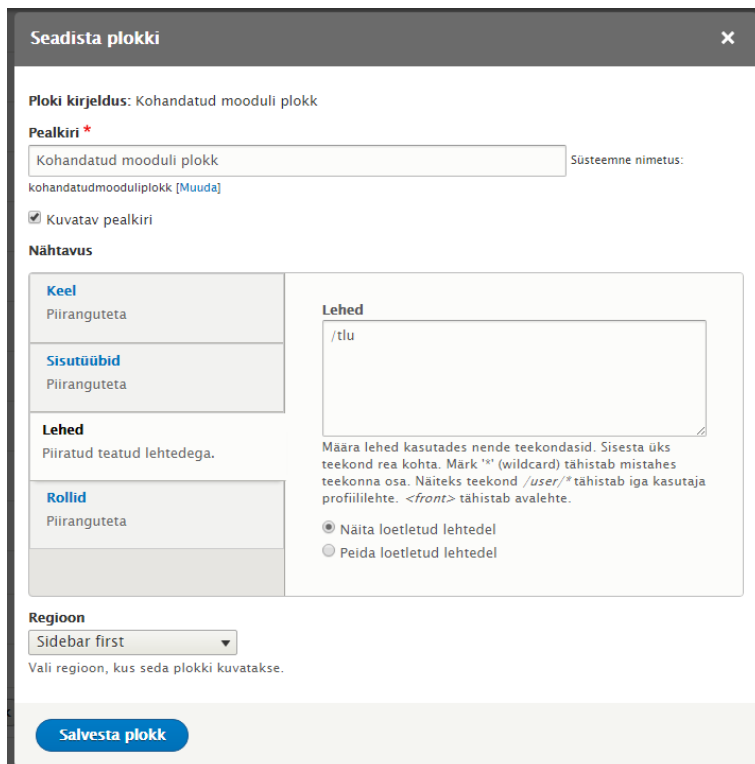
Koodinäide 11. KohandatudPlokk.php

Peale vahemälu tühjendamist peaks uus plokk olema nähtav sisuhaldussüsteemis olevas plokkide nimekirjas ja valmis paigaldamiseks (vt joonis 47).



Joonis 47. Kohandatud mooduliga tekitatud ploki paigutamine regiooni läbi haldusliidese

Lisan kohandatud mooduliga tekitatud ploki lehekülje vasakule küljeribale otsingukasti alla ja määran selle avalduma ainult mooduli poolt loodud leheküljel (vt joonis 48).



Joonis 48. Kohandatud ploki seadistamine läbi haldusliidese

Nüüd peaks kohandatud mooduliga loodud plokk olema nähtav vastaval leheküljel (vt joonis 49).



Joonis 49. Kohandatud mooduliga loodud plokk nähtav kohandatud mooduliga loodud lehekülje vasakul küljeribal

Kokkuvõte

Käesolevas bakalaureusetöös oli eesmärgiks võrrelda Drupal 8 sisuhaldussüsteemi Drupal 7-ga ja luua ülevaade Drupal 8 uutest võimalustest.

Töö eesmärkide saavutamiseks paigaldas Drupal sisuhaldussüsteemid ja kaardistasin nende kattuvad funktsionaalsused. Lisaks tõin töös eraldi välja detailid, mis olid Drupal sisuhaldussüsteemi vaikumisi paigaldusprofiilil uuenenud.

Tuginedes isiklikule kogemusele ja olemasolevatele materjalidele, sai teostatud valik töös lähemalt kajastavatest uutest võimalustest, millest tulenevalt keskendus in vaid tähtsamatele uuendustele Drupal 8 sisuhaldussüsteemis.

Töö käigus lõin põhjaliku ülevaate Drupal 8 sisuhaldussüsteemist ja keskendus in funktsionaalsuse kirjeldamisel selle uutele võimalustele. Lisaks põhjalikule ülevaatele rakendas in uusi funktsionaalsuseid vastavalt nende enimlevinud kasutusjuhtudele.

Töö järelalusena saab väita, et Drupal 8 on oma eelkäijast mitmel viisil parem, sest parandatud on mitmed Drupal 7 puhul eksisteerinud puudused ja uus versioon pakub mitmeid kasulikke võimalusi juba vaikumisi, nendeks lisaarendusi tegemata.

Autori hinnangul on bakalaureusetöö oma eesmärgi täitnud, kuna töö loob eesti keeles põhjaliku ülevaate Drupal 8 sisuhaldussüsteemist ja võrdleb selle omadusi eelmise versiooni omadega. Töös käsitletud näiteid saab kasutada põhjana sisuhaldussüsteemi veelgi põhjalikumana rakendamise korral veebiarenduses. Lisaks andis töö loomine juurde mitmeid teadmisi sisuhaldussüsteemi mõlema versiooni ja nendega seotud moodulite kohta.

Töö võimalike edasiarendustena näen veelgi täpsemat ülevaadet võimaluste kohta, mida antud töös vastavalt teostatud valikule ei käsitletud. Antud tööd saab edasi arendada nii teemade, kui ka moodulite loomisega Drupal 8 sisuhaldusplatvormile.

Summary

The New Possibilities of Drupal 8 CMS

Bachelor's Thesis

The main goal of this bachelor's thesis was to compare Drupal 8 CMS (*content management system*) to Drupal 7 and to create an overview of the new possibilities of Drupal 8 CMS.

In order to focus only on the new possibilities I had to install both versions of the CMS and map the overlapping functionalities. In addition to the changes in overlapping functionalities between the two versions, the thesis focuses on the completely new functionalities in Drupal 8, which did not exist on the previous version of the CMS.

This thesis does not cover all changes which took place between upgrading Drupal 7 to Drupal 8. It only covers the most important changes between the default installation profiles of the two versions of the CMS. The selection of the most important changes to cover in this thesis is based on existing documentation and on my own experience with Drupal.

The thesis creates a thorough overview of Drupal 8 CMS and of the new possibilities in its functionality. In addition to the overview, the thesis demonstrates how to use the main functionalities according to their most common use cases.

As a result of this thesis I can claim that the new version of Drupal is much more superior to its predecessor. It works well in many cases where the previous version of the CMS lacked and it has many expected functionalities ready to use out of the box instead of having to configure them like they had to be configured in Drupal 7.

As the author of this thesis I believe that it has filled its purpose since it creates a thorough overview of Drupal 8 and compares it to its predecessor in many different ways. In addition to that, the various examples found in this thesis can be used in practice by Drupal 8 developers. Also, i feel that i learned a lot about both versions of the CMS and their modules while writing this thesis.

As possible further developments of this thesis there can be done an even further overview of the possibilities already covered or an overview of the possibilities that did not get covered in this thesis. This thesis can also be used for useful information when developing Drupal 8 themes or modules.

Kasutatud kirjandus

- Buytaert, D. (09. 09 2013. a.). *Why the big architectural changes in Drupal 8*. Allikas: <http://buytaert.net/why-the-big-architectural-changes-in-drupal-8>
- Byron, A. (26. 05 2015. a.). *Everything You Need to Know About the Top Changes in Drupal 8*. Allikas: Slideshare: <https://www.slideshare.net/AcquiaInc/acquia-d8-webinar>
- Byron, A. (kuupäev puudub). *The Ultimate Guide to Drupal 8*. Allikas: Acquia: <https://www.acquia.com/sites/default/files/library/attachment/ultimate-guide-drupal-8v3.pdf>
- Cipix. (kuupäev puudub). *Understanding Drupal 8, Part 1: The general structure of the framework*. Allikas: Cipix.nl: <https://cipix.nl/understanding-drupal-8-part-1-general-structure-framework>
- Debugging Twig templates*. (09. 02 2017. a.). Allikas: Drupal.org: <https://www.drupal.org/docs/8/theming/twig/debugging-twig-templates>
- Drupal 8 documentation*. (kuupäev puudub). Allikas: Drupa.org: <https://www.drupal.org/docs/8>
- Drupal.org*. (18. 08 2016. a.). Allikas: Drupal 8 Accessibility Features: <https://www.drupal.org/docs/8/accessibility/drupal-8-accessibility-features>
- Hojtsy, G. (16. 10 2015. a.). *Drupal 8 multilingual tidbits 12: English can now be translated too*. Allikas: Gábor Hojtsy on Drupal: <http://hojtsy.hu/blog/2013-aug-20/drupal-8-multilingual-tidbits-12-english-can-now-be-translated>
- Hojtsy, G. (16. 10 2015. a.). *Drupal 8 multilingual tidbits 3: simple language setup, optional English*. Allikas: Gábor Hojtsy on Drupal: <http://hojtsy.hu/blog/2013-jun-17/drupal-8-multilingual-tidbits-3-simple-language-setup-optional-english>
- McCormick, K. (kuupäev puudub). *Creating custom modules*. Allikas: Drupal.org: <https://www.drupal.org/docs/8/creating-custom-modules>
- Overview*. (04. 08 2016. a.). Allikas: Drupal.org: <https://www.drupal.org/docs/7/system-requirements/overview>

Theming differences between Drupal 6, 7 & 8. (27. 02 2017. a.). Allikas: Drupal.org:
<https://www.drupal.org/docs/8/theming-drupal-8/theming-differences-between-drupal-6-7-8>

Tirez, S. (2014). *The Drupal 8 Theming guide*. GitBook. Allikas: <https://sqndr.github.io/d8-theming-guide/index.html>

Translation API overview. (22. 02 2017. a.). Allikas: Drupal.org:
<https://www.drupal.org/docs/8/api/translation-api/overview>

W3C. (kuupäev puudub). *G18: Ensuring that a contrast ratio of at least 4.5:1 exists between text (and images of text) and background behind the text.* Allikas: W3C:
<https://www.w3.org/TR/WCAG20-TECHS/G18.html>

LISAD

Lisa 1

	Drupal 8	Drupal 7
Paigaldusprofiil	<ul style="list-style-type: none"> • Mahult suur • Sisaldab rohkem faile • Paigaldamine aeglasem 	<ul style="list-style-type: none"> • Mahult väike • Sisaldab vähem faile • Paigaldamine kiirem
Juurdepääsetavus	Vastab WCAG 2.0 standarditele	Puudub
Adapteeruvus	Arendatud eelkõige mobiilset veebi silmas pidades	Puudub
Arhitektuur	Symfony 2	Drupal
Struktuur	<ul style="list-style-type: none"> • Intuiitivne – struktuur on kompaktsem ja sellest arusaamine loogilisem • Tuum asub eraldi kaustas 	<ul style="list-style-type: none"> • Keeruline – struktuurist arusaamine eeldab eelnevaid teadmiseid. • Tuum ei asu eraldi kaustas
Vaikimisi teema	Bartik	Bartik
Teema mootor	Twig	PHPTemplate
Moodulid	<ul style="list-style-type: none"> • Mitmed enimkasutatud moodulid tuumas • Eraldiseivate moodulite funktsionaalsused grupeeritud ühte moodulisse 	<ul style="list-style-type: none"> • Mitmed enimkasutatud moodulid ei kuulu tuuma • Palju eraldiseisvaid mooduleid
Views	Tuum	Eraldi moodul
CKEditor	Tuum	Eraldi moodul
Quick Edit	Tuum	Sarnane tulemus saavutatav eraldi moodulitega
Mitmekeelsus	Tuum	Sarnane tulemus saavutatav eraldi moodulitega