

ITI0205: Veebirakendus

12. PHP. Intermediate Topics.

Martin Verrev

martin.verrev@taltech.ee

Täna sed teemad

- Rakenduse paigaldamine serverise
- Paketihaldus - Composer
- Ruutimine



<https://getcomposer.org/>

Composer is a tool for dependency management in PHP. It allows you to declare the libraries your project depends on and it will manage (install & update) them for you.

Installeerimine

<https://getcomposer.org/download/>

Vaikimisi:

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php composer-setup.php
...
php -r "unlink('composer-setup.php');"
```

Vajadusel seadistamine:

```
php composer-setup.php --install-dir=bin --filename=composer
```

Lisamine globaalsesse skooopi:

Näiteks `mv composer.phar /usr/local/bin/composer` Linuxil või lisamine `PATH` muutujasse Windowsil

Composer - peamised käsud

- Veendume et Composer on installeeritud `composer -v`
- Initsialiseerimine projektis `composer init`
- Projekti sõltuvuste installeerimine `composer install`
- Paketi installeerimine `composer require __package__ <version>`
- Pakettide uuendamine: `composer update`

Vaata ka: <https://getcomposer.org/doc/03-cli.md>

Packagist is the main Composer repository. It aggregates public PHP packages installable with Composer.

<https://packagist.org/>

Composer: failid ja kataloogid

- `composer.json` - A rough guide to the dependency versions that Composer should install
- `composer.lock` - An exact record of the dependency versions that have been installed.
- `vendor/` - kataloog kuhu sõltuvused installeeritakse

Installeeritud paketitde kasutamine

For libraries that specify autoload information, Composer generates a `vendor/autoload.php` file. You can include this file and start using the classes that those libraries provide without any extra work.

```
require __DIR__ . '/vendor/autoload.php';
```

Autoloading allows us to use PHP classes without the need to `require()` or `include()`

```
<?php
require "vendor/autoload.php";

use Acme\Page;

$page1 = new Page();
```

.htaccess is a configuration file for use on web servers running the Apache Web Server software. When a .htaccess file is placed in a directory which is in turn served via Apache Web Server, then it is detected and executed by the Apache Web Server software.

These .htaccess files can be used to alter the configuration of the Apache Web Server software to enable/disable additional functionality and features that the Apache Web Server software has to offer.

Ruutimine .htaccess failiga

Sõltub Apache `mod_rewrite` moodulist

```
RewriteEngine On

RewriteBase /

RewriteCond %{REQUEST_FILENAME} !-d
RewriteCond %{REQUEST_FILENAME} !-f

RewriteRule ^(.+)$ index.php [QSA,L]
```

https://httpd.apache.org/docs/2.4/mod/mod_rewrite.html

Ruutimine: rakenduse lihtne loogika

- Loeme `index.php` failis `$_SERVER['REQUEST_URI']` muutuja
- Töötleme seda ja lingime sobiva vaate
- Kui sobivat vaadet ei leidnud, siis kuvame veateadet

Teeme läbi rakenduse paigaldamise serverisse

Strateegiaid rakenduse paigaldamiseks

- Kasuta relatiivseid viiteid, `login.php` mitte `/login.php`
- Hoia konfiguratsioon lahus ülejäänud rakendusest.
- Loe rakenduse spetsiifiline konfiguratsioon kasutades `$_SERVER` muutujat
- Koodi saad importida kas Giti või SCP kaudu
- Andmebaas - tekita fail `dump.sql` mille saad serveris sisse lugeda `mysql -uparool`
`-p andmebaas < dump.sql`

Tänan!

Viiteid

- Composer: <https://getcomposer.org>
- How to use Packagist and Composer? <https://phpenthusiast.com/blog/how-to-use-packagist-and-composer>
- PHP: The Composer Lock File <https://daylerees.com/the-composer-lock-file/>
- How to Autoload with Composer: <https://phpenthusiast.com/blog/how-to-autoload-with-composer>
- The Simplest PHP Router: <https://www.taniarascia.com/the-simplest-php-router/>
- .htaccess Guide: <http://www.htaccess-guide.com/>
- RewriteCond and RewriteRule tricks.
<https://www.leaseweb.com/labs/2015/10/rewritecond-and-rewriterule-tricks-for-htaccess/>