

## Webentwicklung (WE) - FWPM

# Übung 1: "Praxiseinstieg"

## Lokales Projekt einrichten

Ziel dieser Übung ist es eine Basis für den Entwicklungsstart der Übungen und damit auch der Studienarbeit zu schaffen.

Dazu erforderlich ist zwingend, ein Entwicklersystem nach den kommunizierten Anforderungen zu haben. Sollte das nicht der Fall sein, ist das die erste Aufgabe.

1. System zur Entwicklung im LAMP Stack einrichten falls nicht schon geschehen
2. Einrichten des Repositories
  - a. Übertragen des Projektstubs von <https://inf-git.fh-rosenheim.de/we-ss21/musterloesung>
    - i. Z.B. über *Fork* oder einfachen Download der Dateien
3. Richte dir das Dependency Management über *Composer* ein
  - a. Editiere die *composer.json* und gib deinem Projekt deinen Namen
  - b. Der Name wird später für die Namespaces deiner Klassen gebraucht

## Projekt erreichbar machen

Mach dein Projekt lokal über deinen Webserver erreichbar damit du überhaupt lokal deine Entwicklung überprüfen kannst.

1. Mache dein Projekt lokal über eine Domain erreichbar. Nutze z.B. *webentwicklung.test* oder einen anderen frei gewählten Namen.
  - a. Du brauchst einen lokalen DNS Eintrag damit dein Browser nicht auf die DNS Struktur des Internets zugreift. Klassisch geht das über die sog. *hosts* Datei die auf allen gängigen Betriebssystemen existiert. Informiere dich wie du sie auf deinem Betriebssystem editieren kannst
  - b. Nutze deine lokale IP Adresse 127.0.0.1 (::1 für IPv6)
2. Richte deinen Webserver ein. Er muss sowohl auf den Aufruf deiner Domain reagieren als auch über Rewrites dynamische Anfragen in deine *index.php* weiterleiten

- a. Recherchiere nach einer aktuellen (!) Anleitung für den Webserver deiner Wahl. Achte auf die Serverversion, manches ändert sich über die Jahre
  - b. Vergiss nicht, das *public* Verzeichnis als Document Root zu verwenden
  - c. Du brauchst zusätzlich noch eine FastCGI Konfiguration
3. Der Aufruf deiner Domain sollte dir jetzt ein "Hello world!" im Browser ausgeben

## Grundlegendes Routing mit FastCGI Daten

Der Webserver übermittelt deiner Anwendung Daten über das FastCGI Protokoll. Anhand dieser Daten kannst du in deiner Anwendung auf Nutzereingaben, wie z.B. die einer URL, reagieren.

Teil dieser Übung ist es, deine *index.php* über die FastCGI Daten dynamisch zu machen.

1. Stelle alternative Inhalte dar, je nach aufgerufener URL
  - a. Nutze die magische globale `$_SERVER` Konstante um auf den angefragten Pfad zu reagieren
  - b. Stelle je nach Pfad eine andere Ausgabe dar
  - c. Du hast ein sehr rudimentäres Routing gebaut
2. Integriere dynamische Inhalte in deine Ausgabe.
  - a. Nutze die magische globale `$_REQUEST` Konstante um POST Parameter und Query Strings auswerten zu können
  - b. Baue die Parameter in deine Ausgabe ein, z.B. als "Hello `$_REQUEST['name']`"

**Bonus:** Deine Anwendung soll so wenig wie möglich auf Skriptlogik zurückgreifen. Lagere daher dein Routing in eine eigene PHP Klasse aus. Dazu muss dein Composer Autoloading richtig konfiguriert sein und deine Klasse nach dem [PSR-4](#) Standard benannt und mit Namespaces versehen sein.