

## [Fast Application Development with CakePHP 4](#)

### [CakePHP 4 Alpha Installation](#)

### [Configuring Nginx for CakePHP 4 with Subfolders](#)

### [CakePHP 4: Adding User Data](#)

### [Creating a Sitemap for SEO](#)

## Fast Application Development with CakePHP 4

The aim of this book is to give a kickstart for fast application development with CakePHP 4. The book is organized in approx 10 User Stories and describes the development of a real CakePHP application. This application is about personal knowledge management.

The user stories cover:

User Story	Technical Story
1. Data Entry	Define Database Tables, Bake Models, Controllers, Templates
2. Data Analysis and Visualization	D3js.org, Rest API
3. Multiple Layout Systems such as bootstrap and amhtml	
4. Social Authentication	OAuth Konfigurations
5. Integration to social media (Twitter)	API to Twitter with OAuth
6. Html Caching for Scalability	Cache HTML using CakePHP Middleware
7. SEO (Sitemaps, RSS)	Batchjobs/Shells, File Writes using Symfony
8. Paywall (planned)	AMP-Paywall
9. Collaborative Multilanguage Support	
10. Quality Assurance	
11. End User Engagement	

## CakePHP 4 Alpha Installation

First, we need to install the latest version of CakePHP.

```
cd /usr/local/www/lf
composer create-project --prefer-dist cakephp/app:4.0.0-alpha1 lfcakephp4

Installing cakephp/app (4.0.0-alpha1)
- Installing cakephp/app (4.0.0-alpha1): Loading from cache
Created project in lfcakephp4alpha
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 76 installs, 0 updates, 0 removals
- Installing cakephp/plugin-installer (1.1.0): Loading from cache
- Installing cakephp/chronos (2.x-dev 6c7b974): Loading from cache
- Installing psr/http-message (1.0.1): Loading from cache
- Installing psr/http-server-handler (1.0.1): Loading from cache
- Installing psr/http-server-middleware (1.0.1): Loading from cache
- Installing psr/http-client (1.0.0): Loading from cache
- Installing psr/http-factory (1.0.0): Loading from cache
- Installing zendframework/zend-diactoros (2.1.1): Loading from cache
- Installing composer/ca-bundle (1.1.4): Loading from cache
- Installing psr/simple-cache (1.0.1): Loading from cache
- Installing zendframework/zend-httphandlerrunner (1.1.0): Loading from cache
- Installing psr/log (1.1.0): Loading from cache
- Installing aura/intl (3.0.0): Loading from cache
- Installing cakephp/cakephp (4.x-dev 8b11f82): Downloading (100%)
- Installing symfony/polyfill-ctype (v1.11.0): Loading from cache
- Installing symfony/polyfill-mbstring (v1.11.0): Loading from cache
- Installing twig/twig (v2.8.1): Downloading (100%)
- Installing jasny/twig-extensions (v1.2.0): Downloading (100%)
- Installing aptoma/twig-markdown (3.1.0): Downloading (100%)
- Installing asm89/twig-cache-extension (1.3.2): Downloading (100%)
- Installing wvrthaximus/twig-view (dev-master b925c33): Downloading (100%)
- Installing cakephp/bake (4.x-dev f33eb5b): Downloading (100%)
- Installing squizlabs/php_codesniffer (3.4.2): Loading from cache
- Installing slevomat/coding-standard (4.8.7): Loading from cache
- Installing cakephp/cakephp-codesniffer (dev-next 3227280): Loading from cache
- Installing jdorn/sql-formatter (v1.2.17): Downloading (100%)
- Installing composer/xdebug-handler (1.3.2): Loading from cache
- Installing composer/spdx-licenses (1.5.1): Loading from cache
- Installing symfony/filesystem (v4.2.7): Loading from cache
- Installing symfony/process (v4.2.7): Downloading (100%)
- Installing symfony/finder (v4.2.7): Downloading (100%)
- Installing symfony/contracts (v1.0.2): Loading from cache
- Installing symfony/console (v4.2.7): Downloading (100%)
- Installing justinrainbow/json-schema (5.2.8): Loading from cache
- Installing seld/phar-utils (1.0.1): Downloading (100%)
- Installing seld/jsonlint (1.7.1): Downloading (100%)
- Installing composer/semver (1.5.0): Loading from cache
- Installing composer/composer (1.8.5): Loading from cache
- Installing cakephp/debug_kit (4.x-dev 50b70f9): Downloading (100%)
- Installing mbletecte/mbletecteclib (2.8.33): Loading from cache
- Installing mb/env (2.1.2): Loading from cache
- Installing josegonzalez/dotenv (3.2.0): Loading from cache
- Installing sebastian/version (2.0.1): Loading from cache
- Installing sebastian/resource-operations (2.0.1): Loading from cache
- Installing sebastian/object-reflector (1.1.1): Loading from cache
- Installing sebastian/recursion-context (3.0.0): Loading from cache
- Installing sebastian/object-enumerator (3.0.3): Loading from cache
- Installing sebastian/global-state (2.0.0): Loading from cache
- Installing sebastian/exporter (3.1.0): Loading from cache
- Installing sebastian/environment (4.1.0): Loading from cache
- Installing sebastian/diff (3.0.2): Loading from cache
- Installing sebastian/comparator (3.0.2): Loading from cache
- Installing phpunit/php-timer (2.1.1): Loading from cache
- Installing phpunit/php-text-template (1.2.1): Loading from cache
- Installing phpunit/php-file-iterator (2.0.2): Loading from cache
- Installing theseer/tokenizer (1.1.2): Loading from cache
```

```
- Installing sebastian/code-unit-reverse-lookup (1.0.1): Loading from cache
- Installing phpunit/php-token-stream (3.0.1): Loading from cache
- Installing phpunit/php-code-coverage (6.1.4): Loading from cache
- Installing doctrine/instantiator (1.2.0): Loading from cache
- Installing webmozart/assert (1.4.0): Loading from cache
- Installing phpdocumentor/reflection-common (1.0.1): Loading from cache
- Installing phpdocumentor/type-resolver (0.4.0): Loading from cache
- Installing phpdocumentor/reflection-docblock (4.3.0): Loading from cache
- Installing phpspec/prophecy (1.8.0): Loading from cache
- Installing phar-io/version (2.0.1): Loading from cache
- Installing phar-io/manifest (1.0.3): Loading from cache
- Installing myclabs/deep-copy (1.9.1): Loading from cache
- Installing phpunit/phpunit (7.5.9): Loading from cache
- Installing jakub-onderka/php-console-color (v0.2): Loading from cache
- Installing jakub-onderka/php-console-highlighter (v0.4): Downloading (100%)
- Installing dnoegel/php-xdg-base-dir (0.1): Loading from cache
- Installing nikic/php-parser (v4.2.1): Loading from cache
- Installing symfony/polyfill-php72 (v1.11.0): Loading from cache
- Installing symfony/var-dumper (v4.2.7): Downloading (100%)
- Installing psy/psysh (v0.9.9): Downloading (100%)
cakephp/app suggests installing markstory/asset_compress (An asset compression plugin which provides file concatenation and a flexible filter system for preprocessing and minification.)
cakephp/app suggests installing dereuromark/cakephp-ide-helper (After baking your code, this keeps your annotations in sync with the code evolving from there on for maximum IDE and PHPStan compatibility.)
cakephp/cakephp suggests installing lib-ICU (The intl PHP library, to use Text::transliterate() or Text::slug())
apptoma/twig-markdown suggests installing michelf/php-markdown (Original Markdown engine with MarkdownExtra.)
apptoma/twig-markdown suggests installing github-api (Needed for using GitHub's Markdown engine provided through their API.)
asm89/twig-cache-extension suggests installing psr/cache-implementation (To make use of PSR-6 cache implementation via PsrCacheAdapter.)
symfony/contracts suggests installing psr/cache (When using the Cache contracts)
symfony/contracts suggests installing psr/container (When using the Service contracts)
symfony/contracts suggests installing symfony/cache-contracts-implementation
symfony/contracts suggests installing symfony/service-contracts-implementation
symfony/contracts suggests installing symfony/translation-contracts-implementation
symfony/console suggests installing symfony/event-dispatcher
symfony/console suggests installing symfony/lock
composer/composer suggests installing ext-zip (Enabling the zip extension allows you to unzip archives)
composer/composer suggests installing ext-zlib (Allow gzip compression of HTTP requests)
cakephp/debug_kit suggests installing ext-pdo_sqlite (DebugKit needs to store panel data in a database. SQLite is simple and easy to use.)
ml/env suggests installing ml/vars (For loading of configs)
sebastian/global-state suggests installing ext-uopz (*)
sebastian/environment suggests installing ext-posix (*)
phpunit/php-code-coverage suggests installing ext-xdebug (^2.6.0)
phpunit/phpunit suggests installing phpunit/php-invoker (^2.0)
phpunit/phpunit suggests installing ext-soap (*)
phpunit/phpunit suggests installing ext-xdebug (*)
symfony/var-dumper suggests installing ext-iconv (To convert non-UTF-8 strings to UTF-8 (or symfony/polyfill-iconv in case ext-iconv cannot be used).)
psy/psysh suggests installing ext-openssl (Enabling the OpenSSL extension makes PsySH a lot happier :)
psy/psysh suggests installing ext-posix (If you have PCNTL, you'll want the POSIX extension as well.)
psy/psysh suggests installing ext-readline (Enables support for arrow-key history navigation, and showing and manipulating command history.)
psy/psysh suggests installing ext-pdo_sqlite (The doc command requires SQLite to work.)
psy/psysh suggests installing hoa/console (A pure PHP readline implementation. You'll want this if your PHP install doesn't already support readline or libedit.)
Writing lock file
Generating autoload files
> Cake\Composer\Installer\PluginInstaller::postAutoloadDump
> App\Console\Installer::postInstall
Created `config/app.php` file
Created `/usr/local/www/lf/lfcakephp4alpha/logs` directory
Created `/usr/local/www/lf/lfcakephp4alpha/tmp/cache/views` directory
Set Folder Permissions ? (Default to Y) [Y,n]? Y
Permissions set on /usr/local/www/lf/lfcakephp4alpha/tmp/cache
Permissions set on /usr/local/www/lf/lfcakephp4alpha/tmp/cache/models
Permissions set on /usr/local/www/lf/lfcakephp4alpha/tmp/cache/persistent
Permissions set on /usr/local/www/lf/lfcakephp4alpha/tmp/cache/views
Permissions set on /usr/local/www/lf/lfcakephp4alpha/tmp/sessions
Permissions set on /usr/local/www/lf/lfcakephp4alpha/tmp/tests
Permissions set on /usr/local/www/lf/lfcakephp4alpha/tmp
Permissions set on /usr/local/www/lf/lfcakephp4alpha/logs
Updated Security.salt value in config/app.php
```

## Configuring Nginx for CakePHP 4 with Subfolders

You find the standard setup for Nginx with CakePHP 4 in <https://book.cakephp.org/3.0/en/installation.html#nginx>.

In our case we will use subfolder in order to develop our solution in parallel to production environment.

The Application will run in <https://www.logikfabrik.com/k/> whereas /k/ is the subfolder. With the subfolder concept there are multiple versions of CakePHP possible.

```
server {
    listen 80;
    server_name logikfabrik.com www.logikfabrik.com;
    rewrite /(.*) https://www.logikfabrik.com/$1 permanent; # to prevent duplicate content
}

server {
    listen 443 ssl http2;
    server_name www.logikfabrik.com;
    access_log /var/log/nginx-access-lf.log;
    error_log /var/log/nginx-error-lf.log;
    root /usr/local/www/lf/webroot; # point to cakephp webroot, refer chapter setup CakePHP
    index index.php;
    server_tokens off;

    # serve static files
    location / {
        try_files $uri $uri/ /index.php?$args;
    }

    location ^~ /k/ { # prefix match not regular expressions
        alias /usr/local/www/lf/.../webroot;
        try_files $uri @kfallback;
    }

    location @kfallback {
        set $path $request_uri;
        if ($path ~ ^/k/(.*)$ ) {
            set $path /$1;
            rewrite ^ $path break;
        }
        include /usr/local/etc/nginx/fastcgi_params;
        fastcgi_pass 127.0.0.1:9000;
        fastcgi_index index.php;
        fastcgi_intercept_errors on;
        fastcgi_param REQUEST_URI $path;
        fastcgi_param QUERY_STRING $uri&$args;
        fastcgi_param SCRIPT_FILENAME /usr/local/www/lf/webroot/index.php;
    }
}
```

## CakePHP 4: Adding User Data

I as a end user want to have my knowledge into the application.

#### Acceptance Criteria

- For each entry I must enter a title
- for each entry I can enter a url and or a text
- for each entry I can enter multiple tags

#### Implementation

```
CREATE TABLE logikfabrik.entry (
  id character(36) NOT NULL,
  user_id character(36) NOT NULL,
  created timestamp with time zone DEFAULT ('now'::text)::timestamp with time zone,
  modified timestamp with time zone DEFAULT ('now'::text)::timestamp with time zone,
  etyp character(1) NOT NULL,
  etitle character varying(80) NOT NULL,
  etags character varying(255),
  eurl character varying(255),
  etext character varying(12000),
  etag jsonb,
  CONSTRAINT entry_etyp_check CHECK ((etyp = ANY (ARRAY['l'::bpchar, 'i'::bpchar, 'v'::bpchar, 't'::bpchar, 'd'::bpchar])))
);

CREATE TABLE user (
  id character(36) NOT NULL,
  nick character varying(24) NOT NULL,
  email character varying(255) NOT NULL,
  lang character(2) DEFAULT 'en'::bpchar,
  last_login timestamp with time zone,
  created timestamp with time zone DEFAULT ('now'::text)::timestamp with time zone,
  modified timestamp with time zone DEFAULT ('now'::text)::timestamp with time zone
);

ALTER TABLE ONLY user
  ADD CONSTRAINT user_pkey PRIMARY KEY (id);

CREATE UNIQUE INDEX entry_eurl ON entry USING btree (user_id, eurl, etags) WHERE (eurl IS NOT NULL);
CREATE INDEX entry_tags_index ON entry USING gin (etag);
CREATE INDEX entry_user ON entry USING btree (user_id);
CREATE UNIQUE INDEX user_email ON user USING btree (email);
CREATE UNIQUE INDEX user_nick ON user USING btree (nick);

ALTER TABLE ONLY entry
  ADD CONSTRAINT entry_user_fk FOREIGN KEY (user_id) REFERENCES user(id);
```

#### Baking models, Controller and Templates

As we have precisely defined our database structure we can generate models, controllers and templates.

```
bin/cake bake model user
```

```
One moment while associations are detected.
```

```
Baking table class for User...
```

```
Wrote `usr/local/.../src/Model/Table/UserTable.php`
```

```
Baking entity class for User...
```

```
Wrote `usr/local/.../src/Model/Entity/User.php`
```

```
Baking test fixture for User...
```

```
Wrote `usr/local/.../tests/Fixture/UserFixture.php`
```

```
Bake is detecting possible fixtures...
```

```
Baking test case for App\Model\Table\UserTable ...
```

```
Wrote `usr/local/.../tests/TestCase/Model/Table/UserTableTest.php`
```

```
Done
```

#### Creating a Sitemap for SEO

I as a product owner want to have a sitemap which allows search engines to index our content in order to make it findable.

#### Acceptance Criteria

- Sitemap is automatically created daily
- Sitemap contains all links to public text entries
- Sitemap conforms with <https://www.sitemaps.org/schemas/>
- Sitemap is validated by an external public sitemap validation tool
- Sitemap is found via a valid robots.txt

#### Implementation

Create a File named 'SitemapCommand.php' in src/Command with this content.

```
execute($q)->fetchAll('assoc');

    $fsys = new Filesystem();
    $fname = sys_get_temp_dir(). 'sitemap.xml';

    try {
```

```
$fsys->appendToFile($fname,'');
} catch (IOExceptionInterface $exception) {
    echo "An error occurred while creating your directory at ".$exception->getPath();
}
$fsys->appendToFile($fname,'');
}
$z=0; // Counter
foreach($rset as $irec) {
    $orec = "\n" . '' . "\n";
    $orec .= '' . 'https://www.logikfabrik.com/k/entries/view/' . $irec['id'] . '' . "\n";
    $orec .= '' . $irec['mod'] . '' . "\n";
    $orec .= '' ;

    try {
        $fsys->appendToFile($fname,$orec);
    } catch (IOExceptionInterface $exception) {
        echo "An error occurred while creating your directory at ".$exception->getPath();
    }
    $z++;
}
$o->out('Sitemap creation ended. Records written:' . $z);

$fsys->appendToFile($fname,'');

$fsys->rename($fname, WWW_ROOT . '/sitemap.xml');
}
}
```