

WeclappClient

Ein schlanker, flexibler PHP-Client zur Anbindung der Weclapp-REST-API (v1).
Er unterstützt strukturierte Abfragen, Pagination, Sortierung und grundlegende CRUD-Operationen.

Die Antwortdaten werden als native PHP-Arrays zurückgegeben – ideal für einfache Integrationen oder eigene Modellierung.

📦 Installation

```
```bash
composer require voipcompetencecenter/weclappclient
```
```

🔗 Einrichtung

```
```php
use WeclappClient\Core\WeclappClient;

$client = new WeclappClient('mein-subdomain', 'mein-api-key');
```
```

🔍 Abfragen mit dem QueryBuilder

```
```php
$results = $client->query('/customer')
 ->whereEq('customerType', 'CUSTOMER')
 ->whereLike('company', '%GmbH%')
 ->orderBy('lastModifiedDate', 'desc')
 ->limit(10)
 ->all();
```
```

Unterstützte Filtermethoden

| Methode | Beschreibung |
|--|--|
| <code>`where(\$field, \$op, \$val)`</code> | Freie Filterbedingung |
| <code>`whereEq('feld', 'val')`</code> | Gleichheit |
| <code>`whereNe('feld', 'val')`</code> | Ungleichheit |
| <code>`whereGt`, `whereGe`</code> | Größer als, größer/gleich |
| <code>`whereLt`, `whereLe`</code> | Kleiner als, kleiner/gleich |
| <code>`whereLike`, `whereILike`</code> | (i)LIKE-Suchmuster |
| <code>`whereNotLike`, `whereNotILike`</code> | Negierte LIKE-Suchmuster |
| <code>`whereIn('feld', [...])`</code> | Feldwert in Liste enthalten |
| <code>`whereNull('feld')`</code> | Feld ist leer/ungesetzt (<code>`IS NULL`</code>) |

Sortierung

```
```php
->orderAsc('fieldName')
->orderDesc('fieldName')
->orderBy('fieldName', 'desc') // 'asc' ist Standard
```
```

Limitierung und Paging

```
```php
->limit(50) // holt max. 50 Ergebnisse über mehrere Seiten hinweg
->page(2, 25) // Seite 2 mit 25 Einträgen (klassische Paginierung)
```
```

📊 Zähler

```
```php
$count = $client->query('/customer')->whereEq('customerType',
'CUSTOMER')->count();
```
```

📄 Einzelne Datensätze abrufen

```
```php
// Ersten passenden Datensatz holen
$customer = $client->query('/customer')
->whereEq('customerType', 'CUSTOMER')
->first();

// Spezifischen Datensatz per ID laden
$customer = $client->query('/customer')->get($id);
```
```

🔄 CRUD-Operationen

```
```php
// Einzelobjekt laden
$customer = $client->query('/customer')->get($id);

// Neuer Eintrag
$created = $client->query('/customer')->create([
 'company' => 'Test GmbH',
 'customerType' => 'CUSTOMER',
 'partyType' => 'ORGANIZATION'
]);

// Update
$updated = $client->query('/customer')->update([
 'id' => $created['id'],
 'company' => 'Geändert GmbH'
]);

// Delete
$success = $client->query('/customer')->delete($created['id']);
```
```

📄 Rückgabeformat

Alle Abfragen liefern Arrays im Weclapp-Format zurück (entspricht `response['body']['result']`).

Beispiel:

```
```php
[
 'id' => 123456,
 'company' => 'Musterfirma GmbH',
 'customerNumber' => 'C-1000',
 // ...
]
```

## ## 🛠 Tests

Das Paket enthält Test-Skripte im `tests/` Verzeichnis:

```
```bash
# Query-Parameter testen (ohne API-Aufrufe)
php tests/test_query_params.php

# first() Methode testen (mit API-Aufrufen)
php tests/test_first_method.php
```
```

Erstellen Sie eine `.env` Datei im `tests/` Verzeichnis mit Ihren Weclapp-Zugangsdaten:

```
```env
WCLP_TEST_SUBDOMAIN=ihr-subdomain
WCLP_TEST_API_KEY=ihr-api-key
```
```

---

## ## 🧩 Für Entwickler: Wiederverwendung des QueryBuilders

Die Klasse `AbstractQueryBuilder` bildet die technische Grundlage für den QueryBuilder.

Sie kapselt API-spezifische Logik für Filter, Pagination, Sortierung und Zählerfunktionen.

Du kannst diese Klasse als Basis verwenden, um eigene API-Clients auf vergleichbarer Architektur aufzubauen – unabhängig von Weclapp.

```
```php
class MyApiQueryBuilder extends AbstractQueryBuilder
{
    public function all(): array
    {
        // eigene Logik zur vollständigen Abfrage
    }
}
```

📖 Hinweis zu Ressourcenklassen

Für umfangreichere Projekte mit Modellklassen (z. B. ``Customer``, ``Article``) und objektorientiertem Zugriff kann dieses Projekt durch ein erweitertes Paket mit `**ResourceClient**` und `**ResourceQueryBuilder**` ergänzt werden. Der Basis-Client funktioniert aber bewusst ohne diese Erweiterung – einfach, schnell, direkt.

📄 Changelog

Siehe `CHANGELOG.md` für detaillierte Versionshistorie.

📄 Lizenz

MIT Lizenz - siehe `LICENSE` Datei für Details.