

# Anlegen/Aktualisieren von Abonnenten (Batch-Funktion)

## Beschreibung

Diese Funktion bietet Ihnen die Möglichkeit Abonnenten per Batch-Job in Backclick zu importieren oder zu aktualisieren.

Im Gegensatz zu der Funktion [Anlegen/Aktualisieren von Abonnenten](#) haben Sie hier die Möglichkeit mehrere tausend Datensätze in einem Aufruf an Backclick zu übergeben.

Bei dem Import haben sie auch zeitgleich die Möglichkeit einen Profilter zu erzeugen oder zu aktualisieren, um im Anschluss über die Funktion [Newsletter-Versand \(II\)](#) aufzurufen und einen Newsletter an die gerade importierten Abonnenten zu versenden.

Die Daten der Abonnenten müssen wie in einer Import-Datei von Backclick aufgebaut sein. Das bedeutet, dass alle Spalten durch das Tabulatorzeichen von einander getrennt sind und pro Zeile ein Datensatz enthalten ist.

In der ersten Zeile müssen die entsprechenden Felder aus Backclick angegeben werden, diese müssen in gewohnter Schreibweise (alles in Großbuchstaben) notiert sein.

Wenn die Schnittstelle mehrmals hintereinander aufgerufen wird, kann es sein, dass die Meldung "Another auto import is running. Please try again later." erscheint. Dies ist immer dann der Fall, wenn bereits einen Batchimport angestoßen wurde. In einem solchen Fall muss die Funktion immer wieder aufgerufen werden, so lange bis der erst Import abgeschlossen ist (die Fehlermeldung wird dann nicht mehr erscheinen).

Verfügbar ab Backclick-Version: 5.8.0

Um den Abonnenten-Import zu starten müssen Sie folgendes angeben:

- **ID des Mandanten**
- API-Key (optionale Authentifizierung, wenn nicht angegeben, dann wird die Authentifizierung über die IP-Adressen vorgenommen)
- **ID's der Verteilerlisten**, auf welche die Abonnenten importiert werden sollen
- **Abonentendaten**
- Information, ob ein Profilter angelegt werden soll (Standard: "createFilter" => false) *optionale Angabe*
- Information, welcher Filter angereichert werden soll (Standard: "filterToUpdate" => 0) *optionale Angabe*

Als Ergebnis erhalten Sie ein Objekt, welches die ID des angelegten oder aktualisierten Filters enthält. Haben Sie keinen Filter angelegt oder aktualisiert, dann steht an dieser Stelle immer eine 0.

Wenn Sie die Funktion aufrufen können bei Requests folgende Fehlermeldungen auftreten:

- Another auto import is running. Please try again later. => *Es läuft bereits ein Import. bitte später noch einmal probieren.*
- No data for import given. => *Sie haben im Import keine Daten angegeben.*
- No List-ID's given in import. => *Sie haben keine Verteilerlisten angegeben.*
- Should I create or update a filter? => *Sie haben angegeben, dass ein Filter angelegt werden soll, aber gleichzeitig die ID eines anzureichernden Filters angegeben.*
- Filter with ID #x is non existent. => *Der anzureichernde Filter existiert nicht.*
- Could not create import file. => *Problem beim Anlegen der temporären Datei aufgetreten.*

## Beispiel

### Beispiel für den Import und das Anlegen eines Profilters

```
$client = new SoapClient($wsdlUrl);
$importSubscribersRequest = new stdClass();
$importSubscribersRequest->mandatorId = 0;
$importSubscribersRequest->newsletterListIds = array(1);
$importSubscribersRequest->createFilter = true;
$importData = utf8_encode("EMAIL\tVORNAME\tNACHNAME\r\n");
for ($i = 0; $i < 10; $i++) {
    $importData = $importData . utf8_encode("user-" . $i . "@example.com\tVorname-" . $i . "\tNachname-" . $i . "\r\n");
}
$importSubscribersRequest->importData = $importData;
try {
    $result = $client->importSubscribers($importSubscribersRequest);
    var_dump($result);
} catch (Exception $e) {
    echo "Fehler aufgetreten: " . $e->getMessage() . "\n";
}
```

### Beispielausgabe

```
object(stdClass)#3 (1) {
    ["filterId"]=>
    int(1)
}
```

# Quellcode

Um ein funktionsfähiges Beispiel testen können, steht hier der Quellcode zum Download zur Verfügung.

[Beispiel für den Import von Abonnenten](#)