



13.01.2021

# CSV File Importer

Dokumentation



Ulrich Stark  
OTH AMBERG-WEIDEN

## Inhalt

1	Grundlegendes.....	1
2	Funktionen .....	1
2.1	Importieren von CSV-Dateien.....	1
2.2	Importieren von XML-Dateien.....	2
2.3	Zusammenfügen .....	2
2.4	Exportieren .....	2
3	Grafische Anwendung .....	3
3.1	Ansicht der importierten Dateien .....	5
3.2	Vorschau-Tabelle .....	6
3.3	Fenster zum Exportieren.....	7
4	Programmierschnittstellen .....	8

# 1 Grundlegendes

Das Projekt teilt sich in das „src“ und das „example“ Verzeichnis auf. Im „src“-Verzeichnis befindet sich der Python-Quellcode für den CSV File Importer und einer grafischen Beispielanwendung, die mit dem Framework Tkinter erstellt wurde. Das „example“-Verzeichnis erhält Beispieldateien zum Importieren. Diese teilen sich in CSV, XML und XSL Dateien auf. Alle der Aufgabe beigelegten Dateien wurden beigelegt und wurden um eigene Testdateien ergänzt.

Entwickelt wurde mit dem Versionsverwaltungstool Git. Es steht als Git-Repository unter <https://github.com/ulrichstark/csv-importer> öffentlich bereit. Die Commit-Historie kann ebenfalls dort nachvollzogen werden.

Dem Projekt liegt außerdem noch diese Dokumentation als Microsoft-Word und PDF-Datei bei. Über das PowerShell Skript „generateDoc.ps1“ kann zu jedem Modul im „src“-Verzeichnis mithilfe von pydoc eine einfache Dokumentation generiert werden.

Um die grafische Beispielanwendung oder das Konsolen-Testprogramm zu starten, müssen zuerst alle benötigten Python Module installiert sein. Dazu zählen die Module pandas, lxml, tkinter, chardet, csv, re und pandastable. Dann sollte mit dem Kommandozeilenbefehl „cd src“ das aktuelle Verzeichnis auf das Quellcode-Verzeichnis geändert werden. Um dann die grafische Beispielanwendung zu starten, reicht der Befehl „python gui.py“. Zum Starten des Konsolen-Testprogramms benutzen Sie bitte den Befehl „python main.py“

## 2 Funktionen

### 2.1 Importieren von CSV-Dateien

Es können CSV-Dateien mit beliebigem Format importiert werden. Dabei erkennt das Programm automatisch das Trenn- und Quotierungszeichen und die Kodierung einer Datei. Außerdem wird versucht, aus der ersten Datenzeile im Vergleich zu den anderen Zeilen zu schließen, ob es sich bei ihr um eine Überschriftenzeile handelt.

Dem Benutzer steht es dabei frei, diese Parameter im Nachhinein zu verändern. Zum Beispiel, wenn das Trennzeichen einer Datei falsch erkannt wurde und somit kein fehlerfreies Importieren gewährleistet werden kann.

## 2.2 Importieren von XML-Dateien

Zusätzlich dazu steht auch der Import von XML-Dateien bereit. Dazu muss eine XSL-Datei (= Extensible Stylesheet Language) angegeben werden, die diese zu importierende XML-Datei in das CSV-Format „transformieren“ kann.

Auch bei dieser Möglichkeit des Importierens wird die Dateikodierung und die speziellen Parameter des CSV-Formats automatisch erkannt. Die Ausgabe der XSL-Datei kann also ein beliebiges, aber geläufiges Trenn- und Quotierungszeichen wählen und sollte trotzdem ohne Fehler importiert werden können.

## 2.3 Zusammenfügen

Jede importierte Datei (CSV oder XML) wird in eine interne Tabelle im Speicher des Computers zusammengefügt. Dabei spielt die Reihenfolge der importierten Dateien und die Spaltenanzahl eine wichtige Rolle. Der erste Import setzt mit seiner Anzahl der Spalten die benötigte Spaltenanzahl aller folgenden Imports fest. Sollte eine Datei importiert werden, die sich von dieser Anzahl unterscheidet, wird sie übersprungen und ein Fehler wird ausgegeben.

Für die Beschriftung der Spalten wird die erste gültige Überschriftenzeile eines Imports herangezogen. Wenn keine der importierten Dateien eine derartige Kopfzeile besitzt, kann das Programm aus den Zelleninhalten in der Spalte auf den Datentyp schließen und setzt diesen mit einer fortlaufenden Nummer als Beschriftung.

Folgende Datentypen können mithilfe von regulären Ausdrücken erkannt werden: Geo-Koordinaten, E-Mail-Adressen, URLs, Datum kombiniert mit Uhrzeit, Datum, Uhrzeit, Dezimalzahlen, Ganzzahlen oder boolesche Ausdrücke. Sollte keiner dieser Datentypen zutreffen, wird der allgemeine Typ „Text“ angenommen.

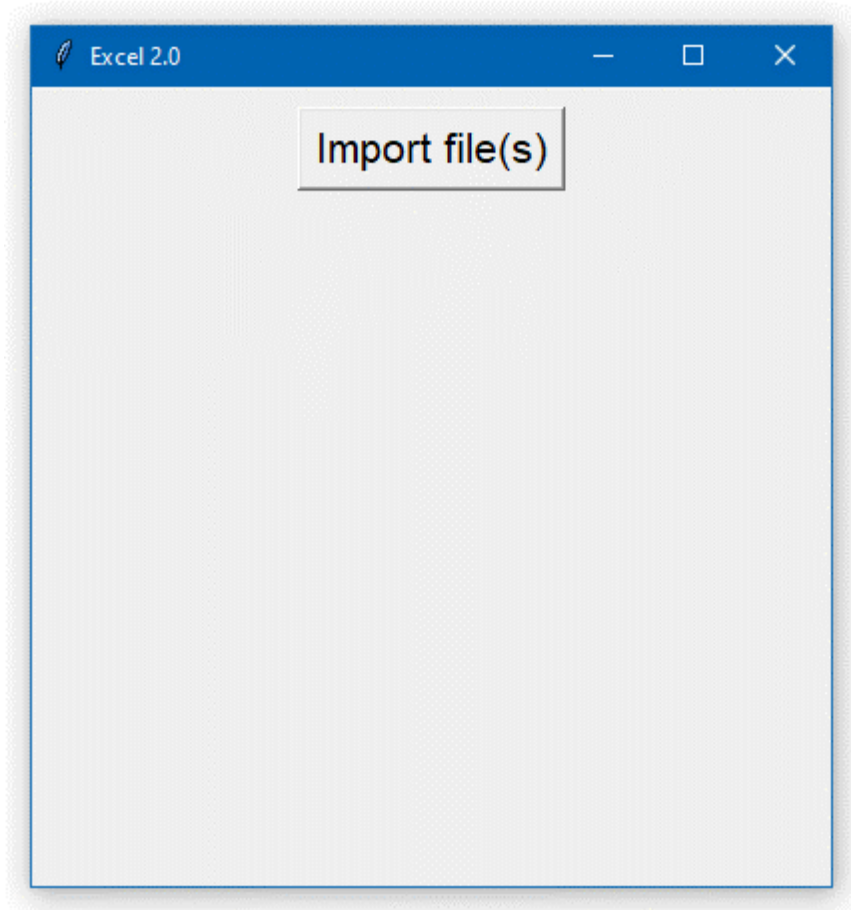
## 2.4 Exportieren

Die kombinierten importierten Dateien lassen sich daraufhin als CSV- und XML-Datei exportieren. Bei beiden Möglichkeiten müssen der Name, Pfad und Kodierung der zu exportierenden Datei ausgewählt werden.

Während des Exportierens einer CSV-Datei kann zusätzlich dazu noch ein beliebiges Trenn- und Quotierungszeichen gewählt werden. Diese werden zur Generierung der Datei genutzt.

### 3 Grafische Anwendung

Die Grafische Anwendung dient als Beispiel für ein Programm, das die Programmierschnittstellen dieses Projekts benutzt. Entwickelt wurde sie mit dem GUI-Framework Tkinter, das in Python standardmäßig integriert ist.



*Abbildung 1: Die Beispielanwendung direkt nach dem Starten*

Nach dem Starten wird der Benutzer von einem einfachen Fenster auf seinem Desktop empfangen, das einen Knopf mit der Aufschrift „Import file(s)“ enthält. Die restlichen Bedienelemente werden bewusst erst eingeblendet, nachdem der Nutzer Dateien zum Importieren ausgewählt hat. Dadurch wird er bei Programmstart nicht von unnötigen Eingabefeldern und Knöpfen abgelenkt.

Wenn er den Knopf zum Importieren der Dateien klickt, erscheint der vom Betriebssystem bereitgestellte Dateiauswahl-Dialog.

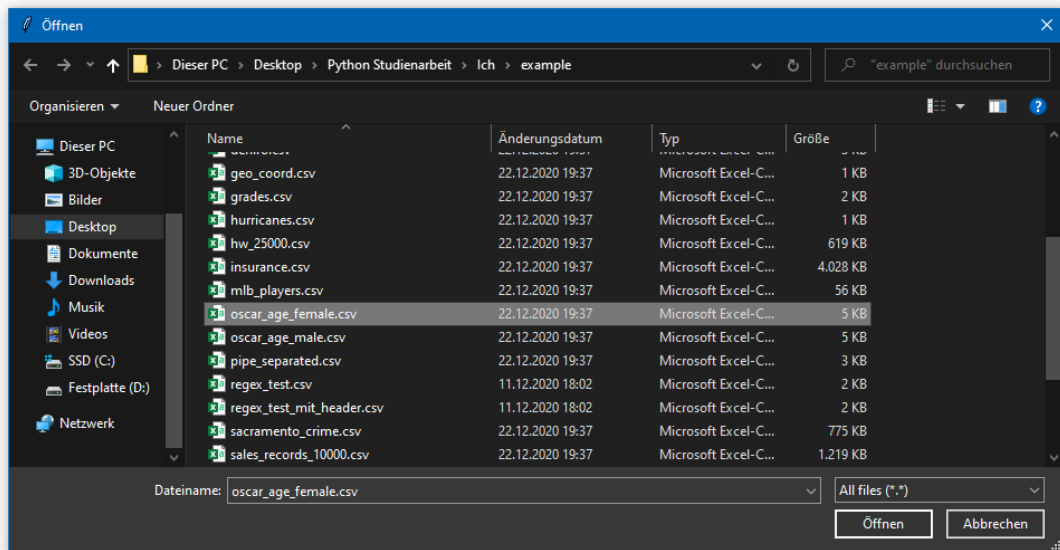


Abbildung 2: Der Dateiauswahl-Dialog zum Importieren der CSV- und XML-Dateien

Dieser Dialog wurde gezielt so konfiguriert, dass er eine Mehrfachauswahl an Dateien zulässt. Außerdem wird standardmäßig das Verzeichnis mit den Beispieldateien geöffnet. Diese beiden Faktoren steigern die Effizienz beim Benutzen und Testen der Beispielanwendung.

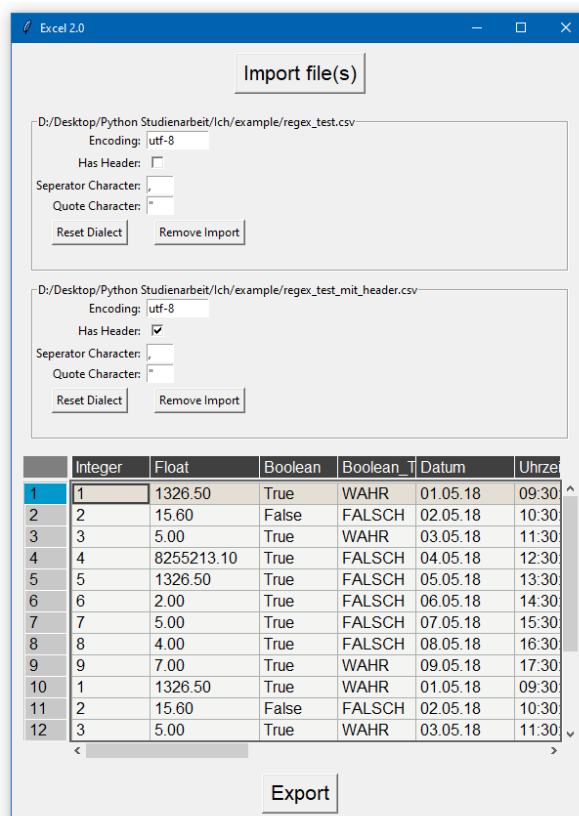


Abbildung 3: Ansicht der Beispielanwendung nach dem Importieren von zwei Dateien

Nachdem mindestens eine Datei importiert wurde, erscheinen die restlichen Bedienelemente. Das Fenster teilt sich nun in vier Bereiche.

Erstens in den Knopf zum Importieren. Darauf folgend ein Rahmen („Frame“) für jede importierte Datei. Drittens eine Tabelle, die als Vorschau für die zusammengeführten Dateien dienen soll. Und zuletzt ein Knopf mit der Aufschrift „Export“. Ein Klick auf diesen öffnet das Fenster zum Exportieren.

### 3.1 Ansicht der importierten Dateien

In diesem Teil der Anwendung erhält jede importierte Datei einen Bereich abgegrenzt durch einen Rahmen („Frame“), in dem ihre zugehörigen Parameter angepasst oder zurückgesetzt werden können. Außerdem steht ein Knopf zum Entfernen dieses Imports bereit.

The image shows two separate frames for importing CSV files. Each frame has a title bar with the file path, followed by input fields for 'Encoding' (set to 'utf-8'), 'Has Header' (checkbox), 'Seperator Character' (comma), and 'Quote Character' (double quote). At the bottom of each frame are two buttons: 'Reset Dialect' and 'Remove Import'.

Frame 1: D:/Desktop/Python Studienarbeit/lch/example/regex\_test.csv  
Encoding: utf-8  
Has Header: ☐  
Seperator Character: ,  
Quote Character: "  
Buttons: Reset Dialect, Remove Import

Frame 2: D:/Desktop/Python Studienarbeit/lch/example/regex\_test\_mit\_header.csv  
Encoding: utf-8  
Has Header: ☒  
Seperator Character: ,  
Quote Character: "  
Buttons: Reset Dialect, Remove Import

Abbildung 4: Zwei importierte CSV-Dateien mit jeweils Parameter und Aktionen

Der Frame für eine importierte CSV-Datei ermöglicht die Anpassung der Dateikodierung, des Trenn- und Quotierungszeichens und ob die Datei eine Kopfzeile besitzt. Diese Werte werden benutzt, um die Datei zu importieren und werden in den meisten Fällen automatisch richtig detektiert bevor der Nutzer eine Eingabe machen muss. Mit dem „Reset Dialect“-Knopf werden die Änderungen des Benutzers verworfen und die Werte abermals aus der Datei detektiert.

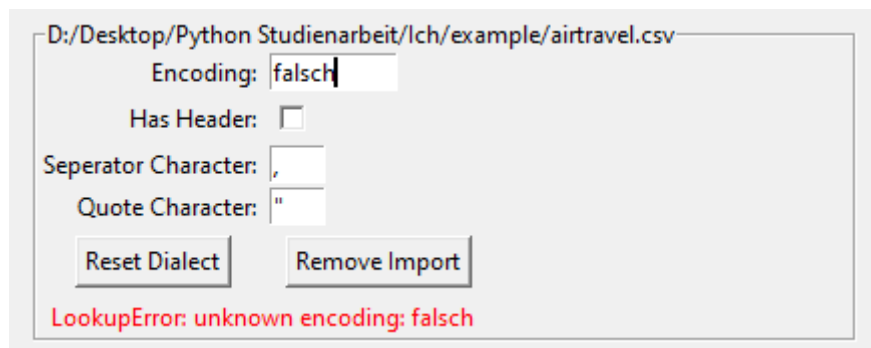
The image shows a single frame for importing an XML file. It has a title bar with the file path, followed by an 'XSL File' input field containing 'cdcatalog2csv.xsl'. At the bottom are two buttons: 'Select XSL File' and 'Remove Import'.

Frame: D:/Desktop/Python Studienarbeit/lch/example/cdcatalog.xml  
XSL File: cdcatalog2csv.xsl  
Buttons: Select XSL File, Remove Import

Abbildung 5: Importierte XML-Datei mit ausgewählter XSL-Datei

Im Frame einer importierten XML-Datei steht zusätzlich zu dem Entfernen-Knopf nur ein Knopf zum Auswählen einer XSI-Datei bereit.

Diese wird benutzt, um den Inhalt der XML-Datei in das CSV-Format zu überführen.



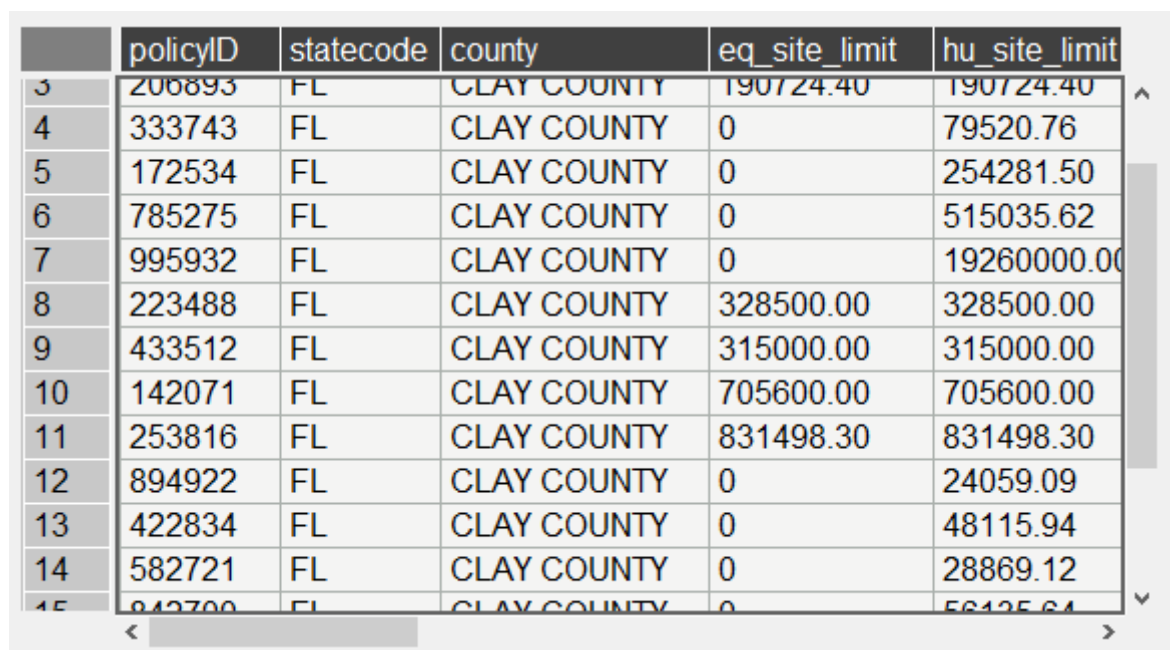
The screenshot shows a window titled "D:/Desktop/Python Studienarbeit/lch/example/airtravel.csv". It contains several input fields: "Encoding:" with the value "falsch", "Has Header:" with an unchecked checkbox, "Seperator Character:" with a comma, and "Quote Character:" with a double quote. Below these are two buttons: "Reset Dialect" and "Remove Import". At the bottom, a red error message reads "LookupError: unknown encoding: falsch".

Abbildung 6: Fehlermeldung bei falscher Angabe der Dateikodierung

Wenn beim Importieren einer Datei ein Fehler auftritt oder die Parameter ungültig sind, wird ein Fehler in dem Frame ausgegeben. Dadurch wird der Benutzer nicht durch einen Fehlerdialog blockiert, sondern erhält eine Fehlerbeschreibung direkt in dem Kontext des Imports, in dem der Fehler aufgetreten ist.

## 3.2 Vorschau-Tabelle

Die Tabelle unter den Rahmen der importierten Dateien dient dem Benutzer als Vorschau. Sie stellt dar, wie die zusammengefügte CSV- und XML-Dateien aussehen, wenn sie exportiert werden würden.



The screenshot shows a table with 6 columns: an index column, "policyID", "statecode", "county", "eq\_site\_limit", and "hu\_site\_limit". The table contains 15 rows of data, all from "CLAY COUNTY" in "FL". The values for "eq\_site\_limit" and "hu\_site\_limit" are numerical values, some of which are truncated on the right side of the image.

	policyID	statecode	county	eq_site_limit	hu_site_limit
3	206893	FL	CLAY COUNTY	190724.40	190724.40
4	333743	FL	CLAY COUNTY	0	79520.76
5	172534	FL	CLAY COUNTY	0	254281.50
6	785275	FL	CLAY COUNTY	0	515035.62
7	995932	FL	CLAY COUNTY	0	19260000.00
8	223488	FL	CLAY COUNTY	328500.00	328500.00
9	433512	FL	CLAY COUNTY	315000.00	315000.00
10	142071	FL	CLAY COUNTY	705600.00	705600.00
11	253816	FL	CLAY COUNTY	831498.30	831498.30
12	894922	FL	CLAY COUNTY	0	24059.09
13	422834	FL	CLAY COUNTY	0	48115.94
14	582721	FL	CLAY COUNTY	0	28869.12
15	842700	FL	CLAY COUNTY	0	56125.64

Abbildung 7: Vorschautabelle unter den importierten Dateien



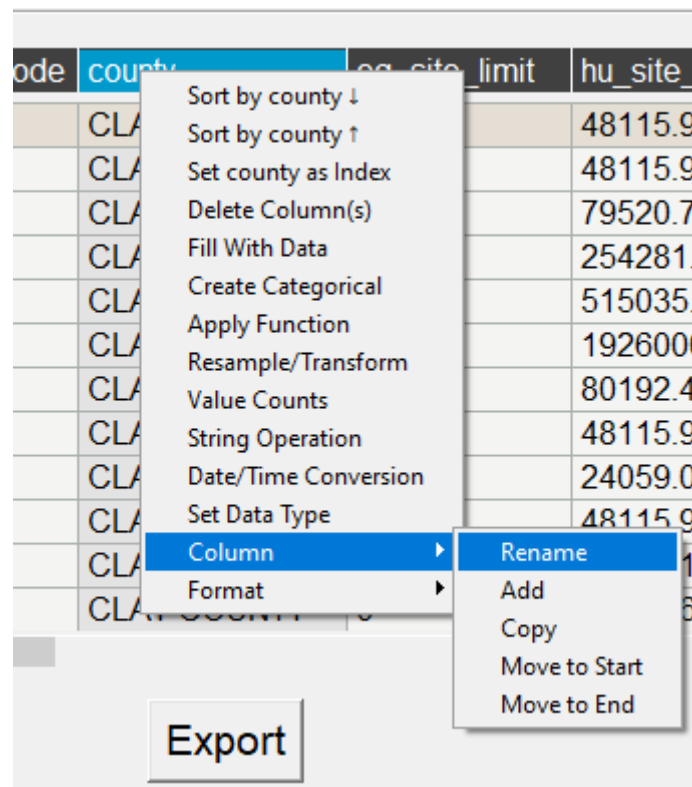


Abbildung 8: Kontextmenü einer Spalte für zusätzliche Aktionen

Außerdem kann der Benutzer damit die Vorschau nach einer bestimmten Spalte auf- oder absteigend sortieren, die Reihenfolge der Spalten ändern oder beispielsweise über das „Column“ Untermenü einer Spalte eine neue Überschrift geben.

### 3.3 Fenster zum Exportieren

Nach einem Klick auf den „Export“-Knopf im Hauptfenster öffnet sich ein Unterfenster zum Exportieren der zusammengeführten Dateien. Dieses enthält eine Tableiste, mit der der Benutzer auswählen kann, ob er eine CSV- oder XML-Datei exportieren will.

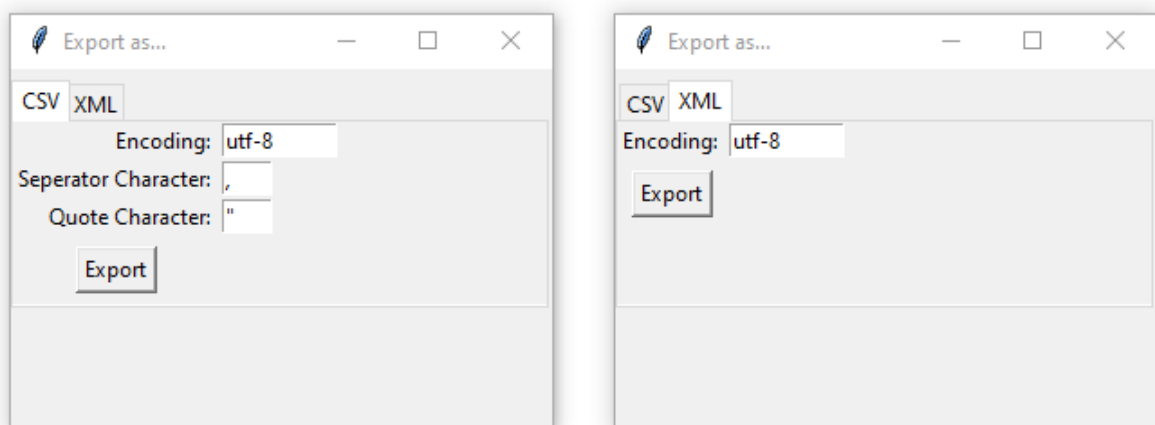


Abbildung 9: Die zwei Zustände des Exportierfensters

Vor dem Export in eine CSV-Datei lässt sich die Kodierung der Datei und das Trenn- und Quotierungszeichen ändern. Für den Export in eine XML-Datei steht nur das Ändern der Dateikodierung bereit.

Mit dem „Export“-Knopf im Unterfenster kann die Auswahl bestätigt werden und das Programm fragt den Benutzer mit dem Dateiauswahl-Dialog des Systems, wohin die Datei und mit welchem Namen sie gespeichert werden soll.

Wählt er einen gültigen Pfad und Namen aus, beginnt das Programm, die Datei zu exportieren und meldet dem Benutzer mit einem Dialog zurück, wenn der Export-Vorgang erfolgreich abgeschlossen wurde. Im Fehlerfall würde ein Fehlerdialog mit der relevanten Fehlermeldung erscheinen. Das könnte zum Beispiel eintreten, wenn der Benutzer sich für eine unbekannte Dateikodierung zum Exportieren entschieden hat.

## 4 Programmierschnittstellen

Die Programmierschnittstellen für das Importieren, Zusammenfügen und Exportieren befinden sich in vier Python Module.

Um Dateien zu importieren, muss ein Objekt der Klasse `Importer` instanziiert werden. Dann können beliebig viele Aufrufe an `importCSVFile` oder `importXMLFile` folgen. Mit einem Aufruf an die `reset`-Methode könnte man das `Importer`-Objekt auf den Stand nach dem Instanziiieren zurücksetzen, um mit dem Importieren neu zu beginnen.

Mit den Methoden, die mit „get“ beginnen, kann jederzeit während des Importier Vorgangs die aktuell zusammengefügte Dateien entweder als `Pandas DataFrame`, `NumPy Array`, Liste an Listen oder Dictionary zurückgegeben werden.

Zum Importieren einer CSV-Datei wird ihr Dialekt benötigt. Dazu muss ein Objekt der Klasse `Dialect` an die Methode `importCSVFile` des `Importer` übergeben werden. Bei Instanziierung des `Dialect` Objekts werden seine Felder mit Standardwerten belegt. Wenn diese Felder mit den Werten für eine bestimmte Datei oder aus einem Beispielstring ausgelesen werden sollen, eignen sich die Methoden `guessFromFile` und `guessFromSample` der `Dialect` Klasse.

Um die zusammengefügte Daten zu exportieren, wird die `Exporter` Klasse benutzt. Ihr Konstruktor erwartet ein Objekt vom Typ `Importer`. Der `Exporter` stellt mit der Methode `exportCSVFile` eine Methode bereit, um eine CSV-Datei mit einem bestimmten Dialekt und Dateikodierung zu exportieren. Mit der Methode `exportXMLFile` kann eine XML-Datei mit der übergebenen Dateikodierung exportiert werden.

Die Methoden aus dem „guess“ Modul werden intern von der Importer und Dialect Klasse genutzt, können aber auch aus anderen Quellen aufgerufen werden. Mit diesem Modul kann erkannt werden, ob eine CSV-Datei eine Kopfzeile hat und mit welchem Dialekt oder Dateikodierung sie abgespeichert wurde. Außerdem stellt dieses Modul eine Methode bereit, um den Datentyp eines Zelleneintrags zu erraten und darauf aufbauend Spaltenüberschriften für einen DataFrame zu generieren.

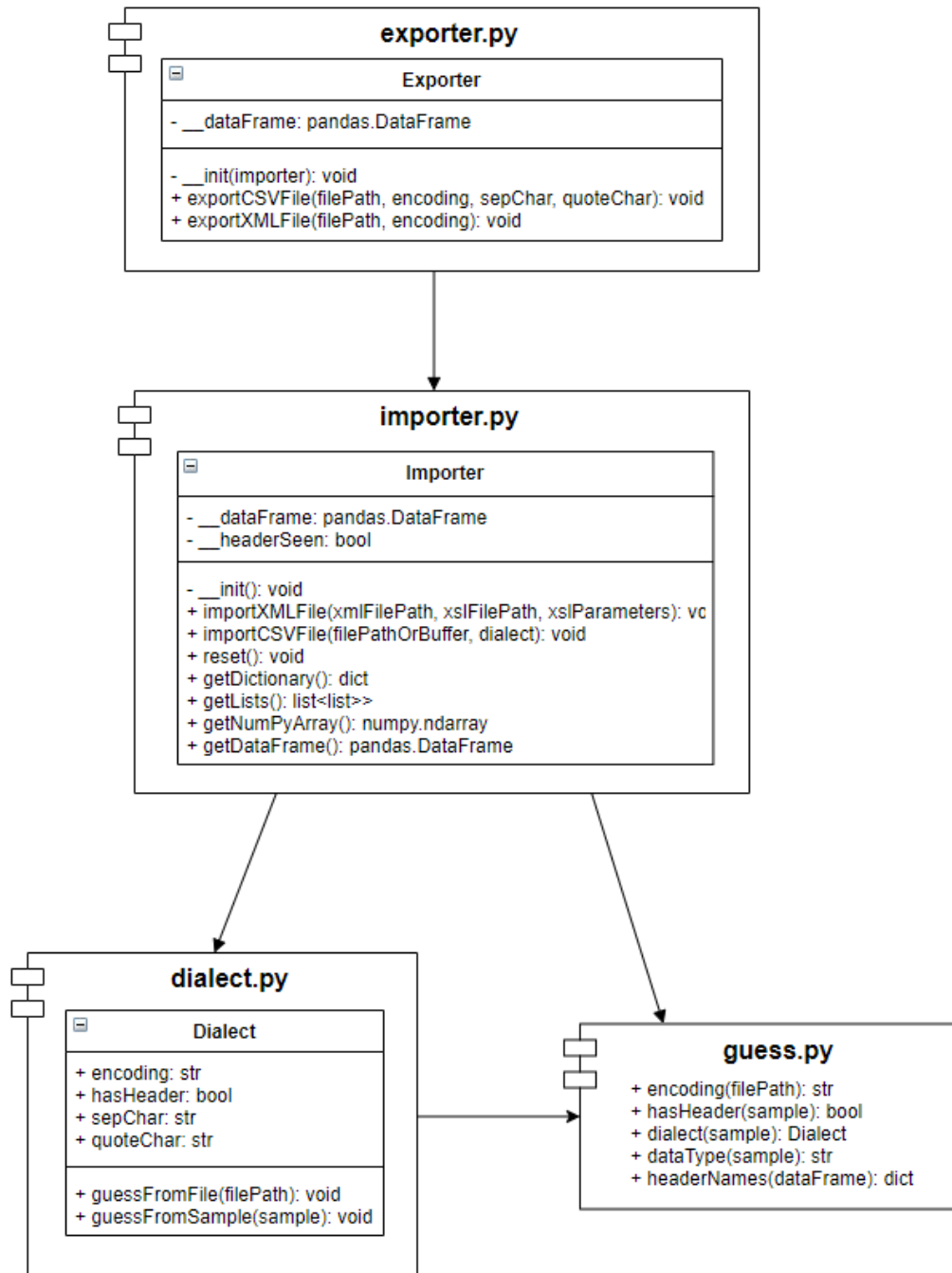


Abbildung 10: UML-Diagramm für Module der API