

Pakete zu alt

(Softwareentwicklung: RESTful APIs mit Python und Flask entwickeln; iX 11/2020, S. 142)

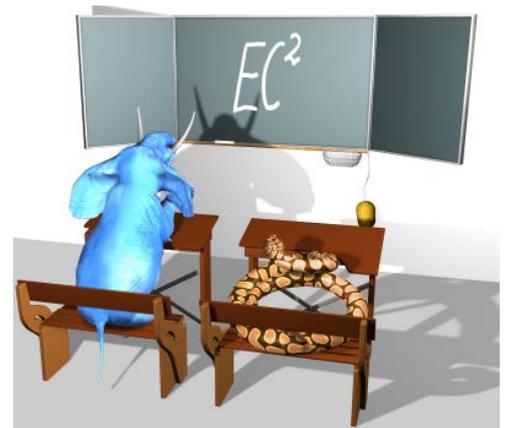
Irgendwie scheinen die Python-Listings im Artikel „Kommunikationshelfer“ aufgrund von Inkompatibilitäten (Abhängigkeit Flask / Werkzeug / flask_restplus) nicht (mehr) richtig zu funktionieren. Die

jeweils neuesten Versionen der Pakete sind vom April dieses Jahres oder älter. Das verwendete flask_restplus stammt sogar aus dem Jahr 2019 – ich gehe davon aus, dass der Artikel auch ein wenig älter ist. Abhilfe schafft der Community-Fork flask-restx in vielen Fällen, trotzdem ist es schade, dass der Artikel aktuell nicht ohne größere Probleme umgesetzt werden kann.

MARKUS DÜNSER, VIA E-MAIL

Vielen Dank für den Hinweis. Sie haben recht, das Flask-RESTPlus-Framework wird unter Flask RESTX seit Anfang des Jahres als Community-Fork weiterentwickelt. Die Installation erfolgt mit pip install flask-restx. Ersetzt man in der Import-Anweisung from flask_restplus import ... durch from flask_restx import ..., funktioniert die Beispieldanwendung wieder.

(Sebastian Bindick)



Der direkte Draht zu



Direktwahl zur Redaktion: 0511 5352-387

Redaktion iX | Postfach 61 04 07
30604 Hannover | Fax: 0511 5352-361
E-Mail: post@ix.de | Web: www.ix.de

www.facebook.com/ix.magazin
twitter.com/ixmagazin (News)
twitter.com/ix (Sonstiges)

Für E-Mail-Anfragen zu Artikeln, technischen Problemen, Produkten et cetera steht die Redaktion gern zur Verfügung.

post@ix.de	Redaktion allgemein
akl@ix.de	Alexandra Kleijn
aneg@ix.de	Alexander Neumann
avr@ix.de	André von Raison
cle@ix.de	Carmen Lehmann
csc@ix.de	Carina Schipper
fo@ix.de	Moritz Förster
jd@ix.de	Jürgen Diercks
map@ix.de	Matthias Parbel
mfe@ix.de	Markus Feilner
mm@ix.de	Michael Mentzel
nb@ix.de	Nicole Bechtel
odi@ix.de	Dr. Oliver Diedrich
rme@ix.de	Rainald Menge-Sonnentag
sun@ix.de	Susanne Nolte
un@ix.de	Bert Ungerer
ur@ix.de	Ute Roos

Listing-Service:

Sämtliche in iX seit 1990 veröffentlichten Listings sind über den iX-FTP-Server erhältlich: ftp.heise.de/pub/ix/

Version 13 statt 12 installieren

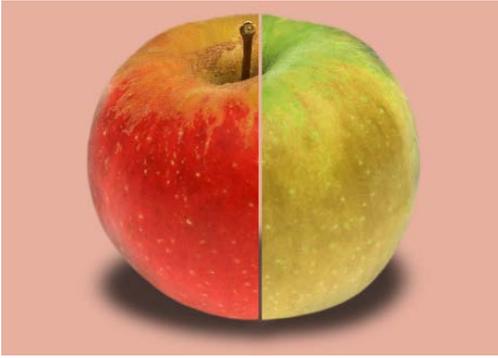
(Maschinelles Lernen: ML mit Python in PostgreSQL; iX 11/2020, S. 132)

Danke für den schönen Artikel „Elefant trifft Schlange“. Leider kommen die Extensions CREATE EXTENSION plpython3u nicht sofort zum Laufen, sondern verabschieden sich mit

```
SQL-Fehler [58P01]: ERROR: could not open extension control file
"/usr/share/postgresql/13/extension/plpython3u.control": No such file or directory
```

Der Fehler: Das Postgres-Docker-Image latest zieht inzwischen Version 13, nicht 12. Es funktioniert jedoch, indem man die entsprechende Extension installiert: `RUN apt-get install -y postgresql-plpython3-13`

ULRICH POTTHOFF, VIA E-MAIL



Sichtbare Spezialisierung

(C++-Tricks: Generischer Code mit Type Traits; iX 10/2020, S. 150)

Die Variante `template<class T> inline constexpr bool is_basic_v = is_fundamental_v<T> || is_enum_v<T> || sizeof(T)<=4;` geht von `sizeof(void*)==4` aus. Besser wäre statt der magischen 4 ein Vergleich mit `sizeof(void*)` oder `sizeof(int)`. Auf einem 32-Bit-System kann es auch effizienter sein, `double` per `const&` zu übergeben statt per Kopie. Bei kleinen Klassentypen kommt es stark auf das ABI an, ob kopieren besser als die `const&`-Übergabe ist. Besser wäre es gegebenenfalls, die size-Prüfung mit `is_trivial` zu ergänzen.

PETER SOMMERLAD, VIA E-MAIL

Sie haben natürlich recht: Nicht für alle Plattformen sind 4 Byte die Größe, ab der sich Referenzen gegenüber Kopieren lohnen. Im Prinzip müsste man das für jede Plattform einzeln bestimmen. Ich habe das mal auf einigen Plattformen mit der Google Benchmark Library probiert und war erstaunt, wie unterschiedlich unvorhersehbar die Grenzgröße war.

Hier zeigt sich, dass die Sichtbarkeit der eigenen Spezialisierung ein Problem sein kann. Jedoch ist das ein allgemeines Problem von Spezialisierungen und nicht von Traits im Speziellen. Leider gibt es beim Programmieren im Allgemeinen und insbesondere bei C++ so viele Implikationen und es sind so viele Details zu bedenken, dass ein Artikel einfach irgendwo einen Strich ziehen muss.

(Detlef Wilkening)

Freie statt Member-Funktionen

(C++-Tricks: C++-Idiome; iX 11/2020, S. 150)

Im Artikel „Mustergültig“ muss es `v.begin()` und `v.end()` heißen, da es sich um Member-Funktionen handelt.

NORBERT EICHENSEHER, VIA E-MAIL

Es gibt die Funktionen „begin“ und „end“ sowohl als Member-Funktionen als auch als freie Funktionen. Die Member-Funktionen gibt es seit C++98, die freien Funktionen sind in C++11 hinzugekommen. Beide machen genau das Gleiche.

Die Tendenz geht in C++ aber zu den freien Funktionen, da sie viele Vorteile gegenüber Member-Funktionen bieten. Hier bei `begin` und `end` wären dies die Erweiterbarkeit und Orthogonalität. Zum Beispiel gibt es für die alten Arrays aus C keine Member-Funktionen `begin` und `end` – unter anderem darum bietet der Standard für diese C-Arrays die freien Funktionen an. Und es wäre einfach nicht schön und bei generischem Code auch nicht gut zu schreiben, wenn man mal `begin(v)` und mal `v.begin()` schreiben müsste. (Detlef Wilkening)



Keine Alternative für Microsoft

(Microsoft: 20 Jahre Exchange 2000: Microsoft auf dem Weg ins RZ; iX 11/2020, S. 75)

Eine Anmerkung zur Verknüpfung von Exchange 2000 mit dem Windows 2000 Server und dem Active Directory: Exchange zerlegte Microsoft 1997 mit Version 5.5 in mehrere Teile. Die bereits mit dieser Ausgabe LDAP-fähigen Verzeichnisse von Exchange wurden von Exchange getrennt und in das Betriebssystem verschoben. Daher kommt dir große AD-Abhängigkeit von Exchange. Microsoft hat da nichts verknüpft, eher konnten sie die Trennung nicht anders lösen. Auch die Zertifikats-serverdienste von Windows basieren auf Teilen von Exchange 5.5.

HEINRICH POMMER, VIA E-MAIL

Die iX-Redaktion behält sich Kürzungen und auszugsweise Wiedergabe der Leserbriefe vor. Die abgedruckten Zuschriften geben ausschließlich die Meinung des Einsenders wieder, nicht die der Redaktion.