

Пример создания и работы с базой данных через интернет с помощью сервиса easydb.io

Радионов Сергей Владимирович

Приамурский государственный университет им. Шолом-Алейхема

Студент

Аннотация

Целью данного исследования является создание базы данных в интернете без создания своего сервера. База данных будет создана с помощью сервиса easydb.io. Управление базой данных будет производиться, используя язык программирования python. Результатом исследования является БД, созданная в сервисе easydb.io.

Ключевые слова: python, база данных, интернет, программа.

An example of creating and working with a database on the Internet using the easydb.io service

Radionov Sergey Vladimirovich

Sholem Aleichem Priamursky State University

Student

Abstract

The purpose of this study is to create a database on the Internet without creating your own server. The database will be created using the easydb.io service. Database management will be done using the python programming language. The result of the study is a database created in the easydb.io service.

Keywords: python, database, internet, program.

В нашем мире широко распространены облачные технологии. Удобно, когда к своим данным есть доступ с любой точки мира, при наличии интернета. Для хранения большого количества данных используются базы данных. Таким образом, чтобы хранить большое количество данных в облаке нужен выделенный сервер. Но сервер нужно арендовать и платить за аренду. Эту проблему решает сервис EasyDB[10].

О.Г.Федоров в своей книге рассматривает понятия банка данных, базы данных и системы управления базой данных [1]. Статья П.А.Головина, Д.В.Денисова, В.А.Нечаева посвящена описанию распределенной базы данных. Приводятся основные возможности и требования к таким базам данных, показывается реализация возможностей системы в конкретном случае [2]. И.Е.Бронштейн в своей статье описал вывод типов для программного кода на языке Python. Сначала производится обзор описанных в научной литературе алгоритмов вывода типов для языков с

параметрическим полиморфизмом. Затем даётся описание нового алгоритма, являющегося модификацией одного из предыдущих: алгоритма декартова произведения. Показывается, как модуль вывода типов, использующий новый алгоритм, анализирует различные конструкции языка Python. Представляются результаты работы над прототипом [3]. В работе В.В.Найденова протестирована производительность пары аналогичных приложений, реализующих CRUD логику с помощью прослойки ORM. Сравнивается SQLAlchemy – де-факто стандартный ORM для Python с динамическим ORM для C++ собственной разработки – YB.ORM. Сравнивается производительность при использовании CPython и PyPy. Проверяется влияние отключения логов на производительность [4]. В статье И.Е.Бронштейн рассматриваются виды дефектов, которые обычно встречаются в программном коде на языке Python. Показывается, что возможные дефекты для Python не похожи на те, что часто встречаются в коде на Си/Си++ и, следовательно, необходимо исследование дефектов в крупных проектах с открытым исходным кодом. Даётся классификация найденных дефектов на основе того, нужен ли для нахождения ошибки вывод типов. Показывается, что существует небольшая доля "простых" дефектов, но для обнаружения большинства дефектов вывод типов необходим. Рассматривается вопрос, какие конструкции языка Python должны поддерживаться при выводе типов для нахождения реальных дефектов [5]. В работе Д.А.Кузнецов рассматривается структура интерфейса программы «Фармацевтическая экономическая безопасность» для фармацевтических организаций. Описывается функциональное предназначение и возможности пунктов основного меню прикладной программы [6]. Ю.А.Котов, А.В.Шаповалов в своей статье рассмотрели интерфейс программной реализации экспертной системы для восстановления простой замены букв текста. Описаны базовые элементы интерфейса, включающие выбор функциональных методов и базовых операций (замена, сдвиг, перебор). Не менее значимы иностранные исследования в данной сфере [8-9].

Для создания базы данных необходимо перейти на сайт веб-сервиса EasyDB [10]. На главной странице одна единственная кнопка (Рис.1), которую нужно нажать, чтобы создать базу данных в интернете. После нажатия на кнопку веб-сервис сообщит данные для подключения к созданной базе данных (Рис.2).

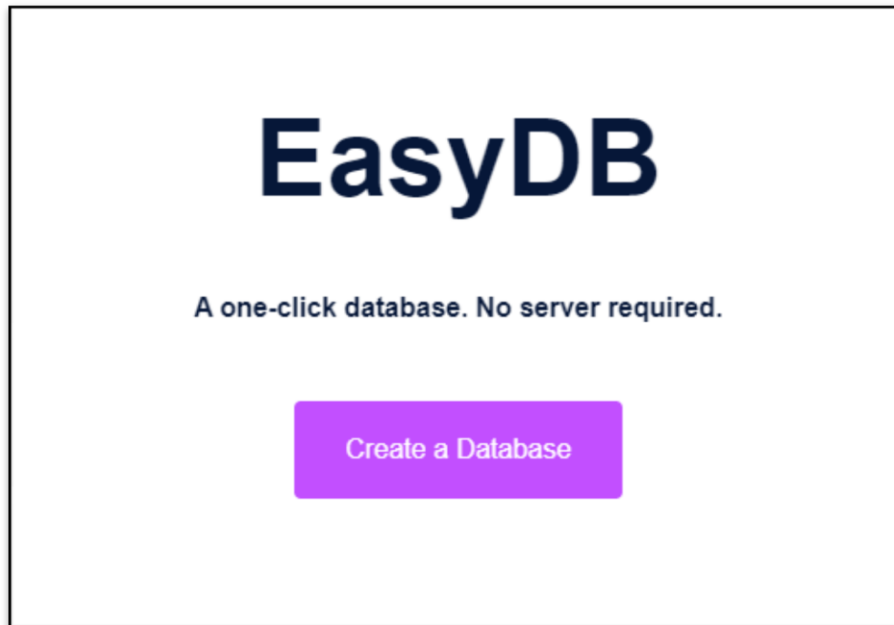


Рис.1. Главная страница

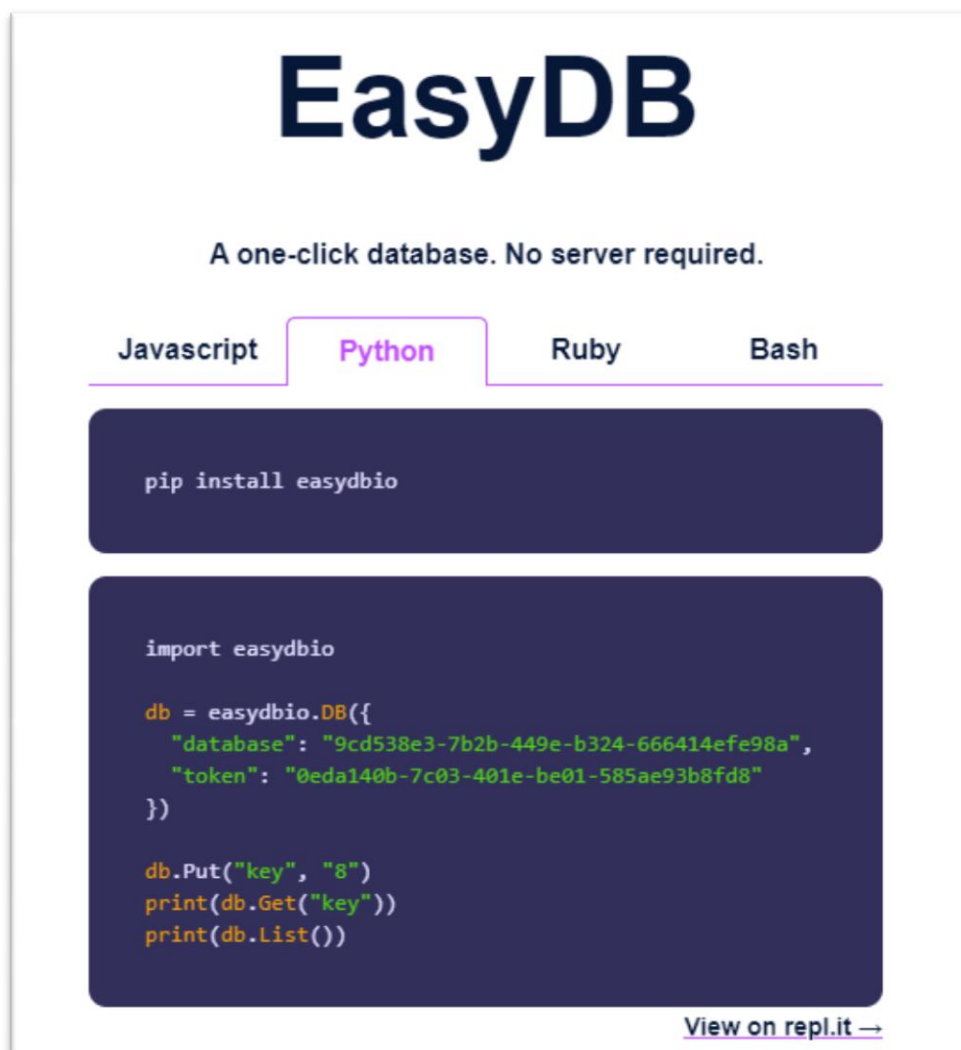


Рис.2. Данные для подключения к ДБ

Сервис создает базу данных временно, на 24 часа. Чтобы сохранить базу данных нужно зарегистрироваться и сохранить ее в личном кабинете (Рис.3). Чтобы сохранить существующую БД или создать новую нужно нажать на «+».



Рис.3. Личный кабинет

После нажатия на кнопку появится окно с выбором действия (Рис.4). Первая кнопка отвечает за создание базы данных, а вторая за сохранение.

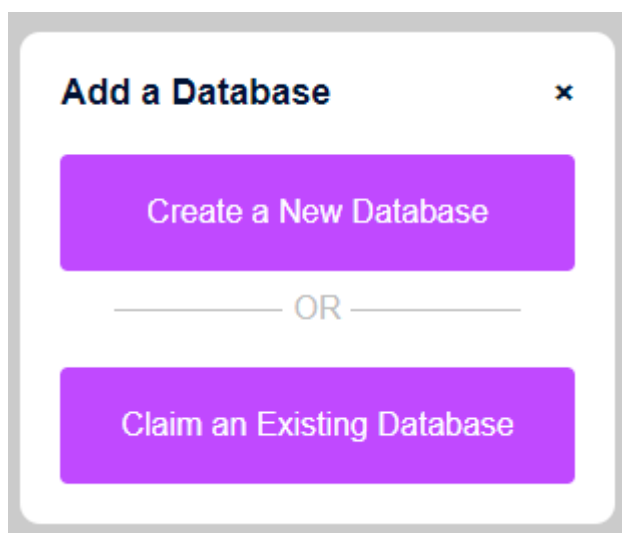
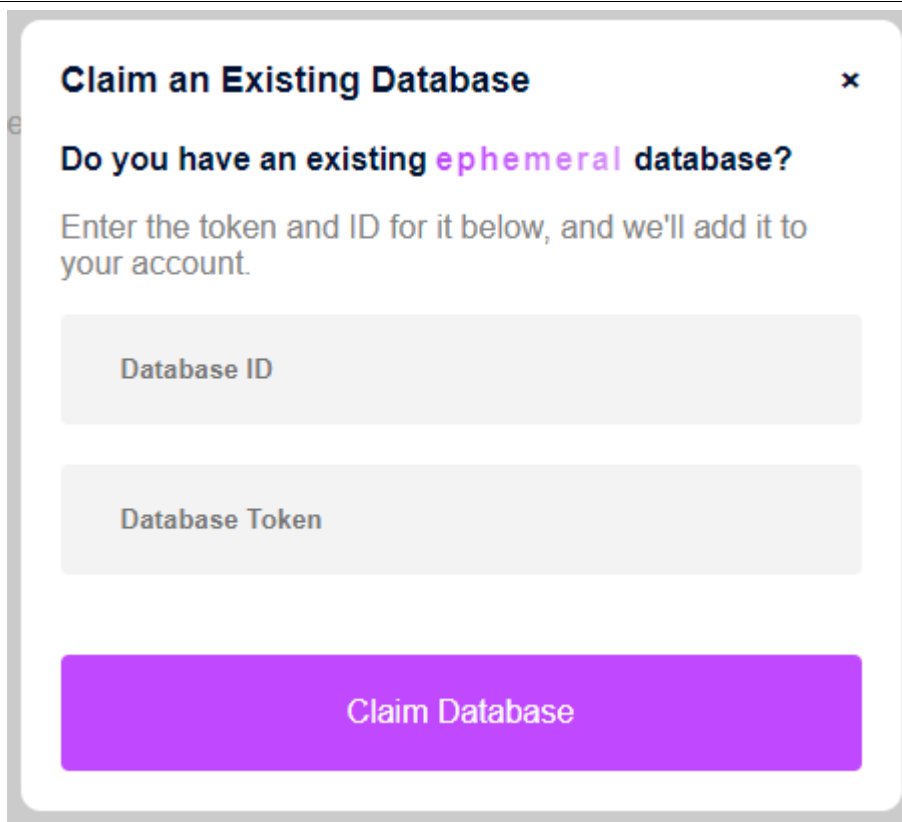


Рис.4. Окно выбора действия

Так как базу данных уже создали ранее, необходимо сохранить существующую. Для этого необходимо нажать на вторую кнопку, после чего появится диалоговое окно (Рис.5). В этом окне нужно ввести полученные ранее данные базы данных, а именно: «id» и «token».



Claim an Existing Database ×

Do you have an existing **ephemeral** database?

Enter the token and ID for it below, and we'll add it to your account.

Database ID

Database Token

Claim Database

Рис.5. Окно сохранения БД

После сохранения БД можно приступать к работе с ней из программы python. Для примера создадим скрипт main.py, в котором произведем основные операции с БД. Нужно скачать необходимые библиотеки, для этого в консоли введем команду «pip install easydbio requests». В скрипте импортируем библиотеку easydbio строкой кода «import easydbio». Далее выполняем подключение к созданной ранее базе данных (Рис.6).

```
db = easydbio.DB({
    "database": "ec297ef4-d772-437d-bb93-e10f41639819",
    "token": "2fbe6ca3-7b0b-4aa5-96ff-86d4f1436224"
})
```

Рис.6. Подключение к БД

Чтобы записать что-то в БД нужно выполнить функцию Put() и передать в нее ключ и значение «db.Put(“key”, “value”)». Чтобы получить значение нужно выполнить функцию Get(), в которую передаем ключ. Для получения списка записанных данных нужно выполнить команду List().

Таким образом, в данной работе рассмотрен быстрый способ создания удаленной базы данных и работы с ней.

Библиографический список

1. Федоров О.Г. Понятия банка данных, базы данных и системы управления базой данных // В книге: Информационные технологии в науке и образовании Федоров О.Г. Учебник. Москва, 2009. С. 454-463.
2. Головин П.А., Денисов Д.В., Нечаев В.А., Нечаев Д.А. Разработка распределенной базы данных на примере базы данных спортсменов олимпийского резерва // Научно-технический вестник Санкт-Петербургского государственного университета информационных технологий, механики и оптики. 2006. № 29. С. 85-86.
3. Бронштейн И.Е. Вывод типов для языка python // Труды Института системного программирования РАН. 2013. Т. 24. С. 161-190.
4. Найденов В.В. Тестирование производительности orm в языках python и c++ // RSDN Magazine. 2014. № 1. С. 05-08.
5. Бронштейн И.Е. Исследование дефектов в коде программ на языке python // Программирование. 2013. Т. 39. № 6. С. 25-32.
6. Кузнецов Д.А. Интерфейс программы "фармацевтическая экономическая безопасность" // Российский медико-биологический вестник им. академика И.П. Павлова. 2009. № 2. С. 162-165.
7. Котов Ю.А., Шаповалов А.В. Интерфейс программы восстановления простой замены букв текста // Современные тенденции развития науки и технологий. 2016. № 4-4. С. 57-59.
8. Smith A. W. et al. Application program interface that enables communication for a network software platform : пат. 7117504 США. 2006.
9. Parikh V., Moore R., Cheng H. Application program interface for a graphics system : пат. 6456290 США. 2002.
10. EasyDB URL: <https://easydb.io> (дата обращения: 17.01.2020).