

Разработка системы проверки упражнений для образовательной платформы

Алексей Кладов

группа 545

руководитель Луцив Д. В.

рецензент Вяххи Н. И.

Санкт-Петербургский государственный университет

10 июня 2014 г.

Stepic

- Udacity
- Coursera
- edX

Много студентов \Rightarrow

- Масштабирование лекций.
- Масштабирование упражнений.

Stepic

Статус

- 2013 год
- 23000 студентов
- 2 курса на Coursera
- курсы CS Center

Технологии

- Linux
- Python 3, Django, Celery, Codejail
- Django **REST** framework
- CoffeeScript, Ember

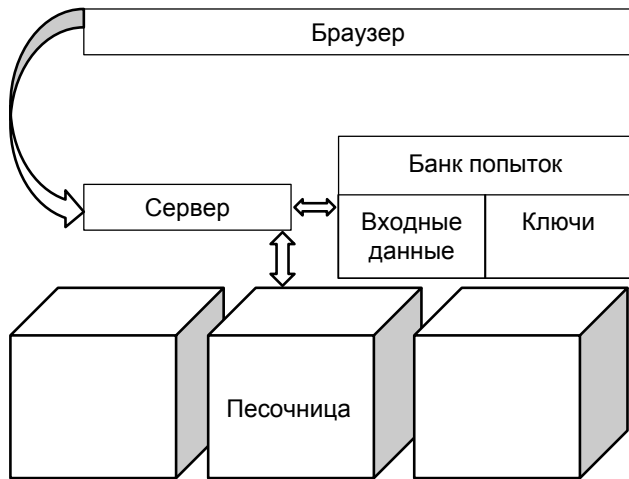
Постановка задачи

Реализовать расширяемую систему для создания и проверки упражнений для образовательной платформы Stepic.

Подзадачи

- Разработать фреймворк для расширения набора типов упражнений сторонними разработчиками.
- Реализовать с помощью фреймворка часто встречающиеся типы упражнений.
- Обеспечить масштабируемое и изолированное исполнение потенциально небезопасного кода упражнений.

Фреймворк

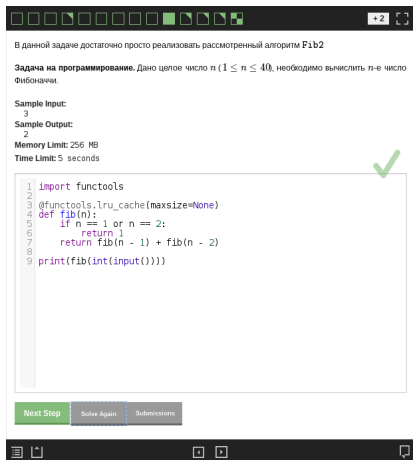


Архитектура решения

Примеры типов упражнений

- Choice
- Code
- Dataset
- Free Answer
- Math
- Number
- Sorting
- String
- **Admin**

Используются в текущих курсах!



The screenshot shows a coding problem interface. At the top, there are navigation icons and a '+2' badge. Below that, the problem description is in Russian: 'В данной задаче достаточно просто реализовать рассмотренный алгоритм Fib2' and 'Задача на программирование. Дано целое число n ($1 \leq n \leq 40$), необходимо вычислить n -е число Фибоначчи.' The sample input is '3' and the sample output is '2'. The constraints are 'Memory Limit: 256 MB' and 'Time Limit: 5 seconds'. A green checkmark is visible in the top right corner of the code editor area. The code editor contains the following Python code:

```
1 import functools
2
3 @functools.lru_cache(maxsize=None)
4 def fib(n):
5     if n == 1 or n == 2:
6         return 1
7     return fib(n - 1) + fib(n - 2)
8
9 print(fib(int(input())))
```

At the bottom of the code editor, there are three buttons: 'Next Step' (green), 'Solve Again', and 'Submissions'.

Пример упражнения (Code)

Изоляция и масштабирование

- Расширение Codejail (мультиязычность, сообщения об ошибках, коммуникация...)
- Создание профилей Apparmor.
- Масштабируемость при помощи Celery.
- Планы: управление конфигурациями.

Результаты

- ✓ Разработан фреймворк для создания новых типов упражнений.
- ✓ С помощью фреймворка реализовано 9 типов упражнений, которые успешно использованы в крупных курсах. Один тип упражнения был разработан сторонним разработчиком.
- ✓ На основе Celery и Codejail создана система масштабируемого и безопасного исполнения кода.

Документация фреймворка

<http://stepic-plugins.readthedocs.org/>