



Технически Университет - София

Факултет компютърни системи и технологии

Курсова работа

Дисциплина: Програмиране за Разпределителни Среда

Изготвил: Сиян Питов

Група : 42

Фак.№: 121221136

Съдържание:

- I. Задание**
- II. Кратко описание и технологията**
- III. III. Описание на проекта**
- IV. IV. Заключение**

I. Задание

Задания Курсови Работи ПРС КСИ 2023

Обобщено задание:

1. Реализирайте функционалност, специфична за XML формата, която няма директен аналог в JSON
(на среда по избор)
 - a. Функционалността е по ваш избор (с изключение на XML Schema)
 - b. Функционалността не може да е XML Schema
2. Реализирайте WEB услуга, която изпраща обработен с функционалността от т.1 файл WEB услугата да включва идентификация и/или автентификация.
(на среда по избор)

(изпълняват се и двете точки)

Детайлно задание:

1. Намерете функционалност свързана с XML, която няма директен аналог в JSON
 - a. не че не може да се направи с JSON, но не може под същата форма
 - b. Attributes не се зачита, защото се разглежда на упражнения
 - c. XML Schema не се зачина
2. Намерете готов XML файл или създайте подходящ XML файл с достатъчно данни, върху който може да се демонстрира функционалността (от т.1).
3. Създайте програма (на среда по избор), която да изпълнява функционалността (от т.1) върху демонстрационния файл (от т.2)
4. Реализирайте WEB услуга, която изпраща обработения файл (от т.3) на отсрещната страна
5. Реализирайте WEB услугата (от т.4) така че: да няма достъп до нея произволен потребител (идентификация/автентификация)
6. Документирате предните точки.

II. Кратко описание и технологията

Проектът се базира на **Python** и **Django**. Django се използва за реализация на уеб услугата с вградена автентификация.

Библиотеката **lxml** се използва за обработка на XML файла, включително поддръжка на namespaces.

III. Описание на проекта

1. XML файл: Използва се XML с namespaces, за да се отделят различните категории книги.

Примерно съдържание:

```
<?xml version="1.0" encoding="UTF-8"?>
<library xmlns:fiction="http://example.com/fiction" xmlns:nonfiction="http://example.com/nonfiction">
  <fiction:book>
    <fiction:title>The Great Gatsby</fiction:title>
    <fiction:author>F. Scott Fitzgerald</fiction:author>
    <fiction:year>1925</fiction:year>
  </fiction:book>
  <nonfiction:book>
    <nonfiction:title>Sapiens: A Brief History of Humankind</nonfiction:title>
    <nonfiction:author>Yuval Noah Harari</nonfiction:author>
    <nonfiction:year>2011</nonfiction:year>
  </nonfiction:book>
</library>
```

2. Python скрипт:

- Парсира XML файла.
- ? Добавя нова книга към файла.
- ? Запазва промените обратно в XML файла.

```
from lxml import etree as ET

xml_file = "library.xml"
namespaces = {
    "fiction": "http://example.com/fiction",
    "nonfiction": "http://example.com/nonfiction"
}

def parse_xml(file_path):
    tree = ET.parse(file_path)
    root = tree.getroot()
    return tree, root

def add_book(root, namespace, title, author, year):
    book = ET.Element(f"{{{namespace}}}book")
    ET.SubElement(book, f"{{{namespace}}}title").text = title
    ET.SubElement(book, f"{{{namespace}}}author").text = author
    ET.SubElement(book, f"{{{namespace}}}year").text = year
    root.append(book)

tree, root = parse_xml(xml_file)
add_book(root, namespaces["fiction"], "1984", "George Orwell", "1949")
tree.write(xml_file, pretty_print=True, xml_declaration=True, encoding="UTF-8")
```

3. Django уеб услуга:

- Използва се декоратор @login_required за защита на услугата.
- Изпраща обработения XML файл на потребителите след вход в системата.

```
from django.http import HttpResponse
from django.contrib.auth.decorators import login_required
import os

@login_required
def download_xml(request):
    file_path = os.path.join(os.path.dirname(__file__), 'library.xml')
    with open(file_path, 'r', encoding='utf-8') as file:
        response = HttpResponse(file.read(), content_type='application/xml')
        response['Content-Disposition'] = 'attachment; filename="library.xml"'
    return response
```

IV. Заключение

Курсовата работа показва успешно реализация на XML обработка и уеб услуга с автентикация. Django позволява бърза имплементация и сигурност при управление на достъпа до услугата, а lxml осигурява гъвкавост при обработка на XML файлове.