Тестовое задание (python)

Разработать сервис терминологии и REST API к нему.

1. Описание

1.1. Описание проблемы

Существуют сервисы, которые обмениваются между собой электронными документами. Электронный документ представляет собой структуру в формате JSON.

Чтобы данные в полях таких документов были понятны принимающей системе и трактовались однозначно всеми участниками обмена, помимо структуры документа, необходимо прийти к общему соглашению кодирования контекста данных.

Для этого потребуется независимый сервис терминологии, который хранит коды данных и их контекст. Проще говоря, это база данных справочников, с кодами и значениями.

Пример кодируемых данных:

Справочник: Специальности медицинских работников.

Код справочника: 1

Версия справочника: 1.0

Дата публикации: 01.01.2022.

Элементы справочника (код - значение):

1 - Врач-терапевт

2 - Травматолог-ортопед

3 - Хирург

Пример кодирования информации в документах:

Направление пациента от врача-терапевта к врачу-травматологу с диагнозом "Другие и неуточненные травмы голеностопного сустава и стопы" (код S99 по справочнику Международного классификатора заболеваний МКБ-10).

```
"direction": {
    "patient": {
       "full_name": "  "
    },
    "doctor_from": {
       "full_name": "  ",
       "spec_code": 1
    },
    "doctor_to": {
       "full_name": "  ",
       "spec_code": 2
    },
    "diagnosis": {
       "dia_code": "S99"
    }
}
```

2. Требования к разрабатываемой системе

2.1. Технологии

Сервис должен быть написан на языке Python 3 с использованием фреймворка Django. В качестве хранилища данных использовать SQLite.

Версия Python >= 3.10

Версия Django >= 4.1

2.2. Формат данных и ограничения

Разрабатываемая система работает с нижеперечисленными объектами и их атрибутами.

"Справочник":

- Идентификатор
- Код (строка, 100 символов, обязательно для заполнения)
- Наименование (строка, 300 символов, обязательно для заполнения)
- Описание (текст произвольной длины)

"Версия справочника":

- Идентификатор
- Идентификатор справочника (обязательно для заполнения)
- Версия (строка, 50 символов, обязательно для заполнения)
- Дата начала действия версии (дата)

"Элемент справочника":

- Идентификатор

- Идентификатор Версии справочника (обязательно для заполнения)
- Код элемента (строка, 100 символов, обязательно для заполнения)
- Значение элемента (строка, 300 символов, обязательно для заполнения)

2.2.1. Ограничения

- Не может существовать более одного Справочника с одинаковым значением в поле Код.
- Не может существовать более одной Версии с одинаковым набором значений "Идентификатор справочника" и "Версия".
- У одного Справочника не может быть более одной версии с одинаковой "Датой начала".
- В одной Версии справочника не может существовать более одного Элемента справочника с одинаковым значением в поле Код.

2.3. Административная панель

Административная панель должна иметь следующие возможности:

- Добавление, изменение, удаление справочника. При редактировании справочника должны заполняются поля: Код (строка, обязательно для заполнения), наименование (строка, обязательно для заполнения), описание (текст произвольной длинны). При редактировании справочника на этой же странице должен быть отображен список имеющихся версий данного справочника с возможностью редактирования.
- В списке справочников должны отображаться поля Идентификатор, Код, Наименование, Текущая версия и Дата начала действия версии.
- Добавление, изменение, удалении версии справочника. При редактировании версии справочника заполняются поля: Справочник (выбор из списка), версия (строка), дата начала действия (выбор из календаря). При редактировании версии необходима возможность на этой же странице заполнить элементы справочника в этой версии.
- В списке версий справочника должны отображаться поля: Код справочника, Наименование справочника, Версия, Дата начала версии.
- Добавление, изменение, удаление элементов в версии справочника. При редактировании элементов заполняются поля: Версия (выбор из списка "справочник-версия"), код элемента (строка), значение элемента (строка).

Интерфейс должен быть на русском языке.

2.4. Требования к API

API должно предоставлять следующие методы.

2.4.1. Получение списка справочников (+ актуальных на указанную дату)

Метод: refbooks/ [?date=<date>]

Тип запроса HTTP: GET

Параметры запроса

Параметр	Описание
date	Дата начала действия в формате ГГГГ-ММ-ДД. Если указана, то должны возвратиться только те справочники, в которых имеются Версии с Датой начала действия раннее или равной указанной.

Формат ответа

Поле	Тип	Описание
refbooks	list	Список объектов справочников
refbooks[].id	string	Код справочника
refbooks[].name	string	Наименование справочника

Пример запроса

GET /refbooks/?date=2022-10-01

Пример ответа

```
{
    "refbooks": [
        {
            "id": "1",
            "code": "MS1",
            "name": "  "
        },
        {
            "id": "2",
            "code": "ICD-10",
            "name": "  -10"
        },
    ]
}
```

2.4.2. Получение элементов заданного справочника

Метод: refbooks/<id>/elements [?version=<version>]

Тип запроса HTTP: GET

Параметры запроса

Параметр	Описание

id	Идентификатор справочника
version	Версия справочника. Если не указана, то должны возвращаться элементы текущей версии. Текущей является та версия, дата начала действия которой позже всех остальных версий данного справочника, но не позже текущей даты.

Формат ответа

Поле	Тип	Описание
elements	list	Список элементов в версии справочника
elements[].code	string	Код элемента
elements[].value	string	Значение элемента

Пример запроса

GET /refbooks/1/elements?version=1.0

Пример ответа

```
{
    "elements": [
        {
             "code": "J00",
             "value": "  ()"
        },
        {
             "code": "J01",
             "value": " "
        }
    ]
}
```

2.4.3. Валидация элементов

Валидация элемента справочника - это проверка на то, что элемент с данным кодом и значением присутствует в указанной версии справочника.

Метод: refbooks/<id>/check_element?code=<code>&value=<value> [&version=<version>]

Тип запроса HTTP: GET

Параметры запроса

Параметр	Описание
id	Идентификатор справочника
code	Код элемента справочника
value	Значение элемента справочника
version	Версия справочника. Если не указана, то должны проверяться элементы в текущей версии. Текущей является та версия, дата начала действия которой позже всех остальных версий данного справочника, но не позже текущей даты.

3. Дополнительные требования

- Для API должны быть написаны тесты.
- Для описания и проверки API должен быть подключен swagger. Все методы и примеры использования должны быть описаны.
- Проект должен быть размещён на GitHub или аналогичном сервисе.