

# Сервис конвертации

## Общее описание

Реализуйте простой Flask-сервер, осуществляющий конвертацию данных, переданных в запросе, в необходимый xml-формат и наоборот (см. раздел Описание форматов).

## Общие требования

- Данные для конвертации поступают в POST-запросе (Content-Type - application/json)
- Идентификация направления конвертации (json->xml / xml->json) - на усмотрение программиста
- Инструкция по запуску/зависимостям
- Плюсом будет
  - опциональная валидация xml по существующим xsd-схемам (см. раздел Описание форматов) (опциональность определяется параметром Flask-настроек)

## Примечание

- Часть JSON-ключей не сопоставляется с XML-тегами напрямую (по названию)
  - потребуется “ручное” сопоставление
- Для случаев, если какой-то из примеров данных не валиден (например, uid примера \*.json не удовлетворяет требованиям XSD-схемы, или некоторых необходимых данных нет)
  - исправьте входные данные
  - отметьте подобные “несостыковки” в readme

## Технологии

- Python version 3.11

## Описание форматов

- Конвертация JSON->XML
  - Формат поступающих в запросе данных - файл `App_info.json`
  - XSD-схема для конвертации в XML - `Add_Entrant_List.xml`
    - Схема содержит элемент *Fields* - содержимое определяется по *IdDocumentType* из схемы путём сопоставления с *Id* в файле `dict_document_type_cls.json`. Правила формирования XML-элемента приведены на рис. 1
- Конвертация XML->JSON
  - XSD-схема поступающих данных - файл `Get_Entrant_List.xsd`
    - Пример данных можно сгенерировать самостоятельно
  - Формат JSON-результата - аналогичен формату в `App_info.json`

```

fields": [
  {
    "type": "integer",
    "not_null": true,
    "xml_name": "IdOksm",
    "xml_cls": "OksmCls",
    "description": "Страна выдачи"
  },
  {
    "type": "character",
    "not_null": true,
    "xml_name": "Surname",
    "description": "Фамилия"
  },
  {
    "type": "character",
    "not_null": true,
    "xml_name": "Name",
    "description": "Имя"
  }
]
}

```

В данном JSON-объекте используются следующие ключи:

*type* – тип данных реквизита;

*not\_null* – обязательность наличия значения в данном реквизите;

*xml\_name* – наименование тэга, в котором необходимо передавать в Сервис приема значение данного реквизита;

*xml\_cls* – наименование классификатора или сущности, из которых необходимо передавать значение в данный реквизит (при необходимости);

*description* – наименование реквизита.

Тогда, согласно приведенному примеру, блок реквизитов из расширенного списка в формируемом вузом XML-документ может выглядеть следующим образом:

```

...
<Fields>
  <IdOksm>12</IdOksm>
  <Surname>Иванов</Surname>
  <Name>Иван</Name>
</Fields>

```

Рисунок 1 - Правила формирования XML-элемента Fields