

Министерство науки и высшего образования Российской Федерации
Муромский институт
федерального государственного бюджетного образовательного учреждения
высшего образования
**«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»
(МИВлГУ)**

Факультет _____ ИТ _____

Кафедра _____ ИС _____

ЛАБОРАТОРНАЯ РАБОТА №1

По _____ Инструментальные средства ИС _____

Тема _____ Инструментальные средства документирования
_____ программного кода _____

Руководитель

Варламов А.Д.
(фамилия, инициалы)

(подпись) (дата)

Студент ИС-117
(группа)

Минеев Р. Р.
(фамилия, инициалы)

(подпись) (дата)

Муром 2021

Лабораторная работа №1.

Тема: Инструментальные средства документирования программного кода.

Цель работы: Научиться получать документацию разработчика по исходному коду программы с помощью современных генераторов документации.

Задание на работу: Реализовать API функции и задокументировать их, используя Apidoc.

Main.py

```
"""
@api {get} /ai-quotes/:id Request quote information by id
@apiName get
@apiGroup Quote

@apiDescription
    Get quote by it's id

@apiParam {Int} id quote unique ID.

@apiSuccess {json} quote The json contains id, author, quote text.
@apiSuccess {Int} 200 Status code of request.

@apiError {int} 404 The <code>id</code> of the quote was not found.

@apiParamExample {String} Request-Example:
{
    "id": 1
}

@apiSuccessExample {json} Success-Response:
HTTP/1.1 200 OK
{
    "id": 1,
    "author": "Elon Musk",
    "quote_text": "Mars is mine"
}
"""
def get(self, id=0):
    if id == 0:
        paragraphs = xml_file.findall('//quote')
        return get_quote_dict(random.randint(1, len(paragraphs))), 200
    quote = get_quote_dict(id)
    if quote is not None:
        return quote, 200
    return f'Quote with id {id} not found', 404

"""
@api {post} /ai-quotes Add new quote
@apiName post
@apiGroup Quote

@apiDescription
    Add new quote (quote id, quote author, quote text) to the xml files of quotes

@apiSuccess {int} 201 Status code of request.

@apiError {int} 400 The <code>id</code> of the quote already exists.

@apiParamExample {json} Request-Example:
{
    "id": 6,
    "auotor": "Elon Mask",
    "quote_text": "Mars is mine"
}
```

					МИВУ 09.03.02-01.005			
Изм	Лист	№ докум.	Подп.	Дата	Лабораторная работа №1 Инструментальные средства документирования программного кода	Литера	Лист	Листов
Студент	Минеев Р. Р.			18.03.		У	2	4
Руков.	Варламов А.Д.					МИ ВлГУ ИС-117		
Конс								
Н.контр.								
Утв.								

```

@apiSuccessExample {json} Success-Response:
  HTTP/1.1 201 OK
  {
    "id": 6,
    "author": "Elon Mask",
    "quote_text": "Mars is mine"
  }
"""
def post(self):
    parser = reqparse.RequestParser()
    parser.add_argument('id', type=int)
    parser.add_argument('author')
    parser.add_argument('quote_text')
    params = parser.parse_args()
    d = get_quote_dict(params['id'])
    if d is not None:
        return f'Quote with id {params["id"]} already exists', 400
    quote = {
        'id': params['id'],
        'author': params['author'],
        'quote': params['quote_text'],
    }
    write_in_xml(quote)
    return quote, 201

"""
@api {delete} /ai-quotes Delete quote
@apiName delete
@apiGroup Quote

@apiDescription
  Delete quote by id from the xml files of quotes

@apiSuccess {int} 200 Status code of request.

@apiError {int} 400 The <code>id</code> of the quote not exists.

@apiParamExample {json} Request-Example:
  {
    "id": 3
  }

@apiSuccessExample {json} Success-Response:
  HTTP/1.1 200 OK
  {
    Quote with id 3 is deleted
  }
"""
def delete(self):
    parser = reqparse.RequestParser()
    parser.add_argument('id', type=int)
    params = parser.parse_args()
    d = get_quote_dict(params['id'])
    if d is None:
        return f'Quote with id {params["id"]} not exists', 400
    root = xml_file.getroot()
    quote = xml_file.find(f'quote[@id="{params["id"]}"]')
    root.remove(quote)
    xml_file.write('quotes.xml')
    return f'Quote with id {params["id"]} is deleted.', 200

```

Quote - Request quote information by id

0.0.0

Get quote by it's id

GET

```
https://localhost:8888/ai-quotes/ai-quotes/:id
```

Параметр

Название	Тип	Описание
id	Int	quote unique ID.

Request-Example:

```
{
  "id": 1
}
```

Success 200

Название	Тип	Описание
quote	json	The json contains id, author, quote text.
200	Int	Status code of request.

Success-Response:

```
HTTP/1.1 200 OK
{
  "id": 1,
  "author": "Elon Musk",
  "quote_text": "Mars is mine"
}
```

Error 4xx

Название	Тип	Описание
404	int	The <code>id</code> of the quote was not found.

Вывод: В этой лабораторной работе были получены навыки в документировании кода, используя apidoc