

ФГБОУ ВО «ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
КАФЕДРА ИНФОРМАТИКИ И МАТЕМАТИЧЕСКОГО ОБЕСПЕЧЕНИЯ

ОТЧЕТ ПО ДИСЦИПЛИНЕ
«ВЕРИФИКАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ»

Выполнила студентка группы 22608 Ю.Л. Ипатова

Лектор:

к.ф.-м.н., доцент К.А. Кулаков

Итоговая оценка:

СОДЕРЖАНИЕ

| | | |
|------|---|----|
| 1 | Выбор и согласование объекта тестирования | 3 |
| 2 | План тестирования..... | 3 |
| 2.1 | Объект тестирования | 3 |
| 2.2 | Стратегия тестирования..... | 4 |
| 2.3 | Стратегия блочного тестирования..... | 7 |
| 2.4 | Стратегия интеграционного тестирования..... | 8 |
| 2.5 | Стратегия аттестационного тестирования | 10 |
| 2.6 | Стратегия нагрузочного тестирования | 11 |
| 2.7 | Критерии прохождения тестов | 12 |
| 2.8 | Критерий приостановления тестирования | 12 |
| 2.9 | Критерий возобновления работы | 12 |
| 3 | Детальный план тестирования | 12 |
| 3.1 | Блочные тесты | 12 |
| 3.2 | Интеграционные тесты..... | 20 |
| 3.3 | Аттестационные тесты | 37 |
| 3.4. | Нагрузочные тесты | 43 |
| 4 | Журнал тестирования | 45 |
| 4.1 | Блочные тесты | 45 |
| 4.2 | Интеграционные тесты..... | 45 |
| 4.3 | Аттестационные тесты | 47 |
| 4.4 | Нагрузочные тесты | 47 |
| 5 | Журнал найденных ошибок | 48 |
| 6 | Пример теста | 49 |
| 7 | Покрытие тестов..... | 51 |

1 Выбор и согласование объекта тестирования

Объектом тестирования является консольное приложение Vocabulary для вывода различных характеристик о слове.

Приложение использует следующие сторонние модули:

- Wordnik – модуль, который предоставляет толковый словарь и тезаурус английских слов;
- Glosbe – многоязычный словарь;
- Urbandict - толковый словарь с примерами использования слов;
- Bihugelabs– словарь синонимов, антонимов и однокоренные слов;

Приложение может выводить результат в следующих форматах:

- json;
- списки Python;
- словари Python.

2 План тестирования

2.1 Объект тестирования

Vocabulary – консольное приложение для вывода различных характеристик о слове с использованием сторонних модулей.

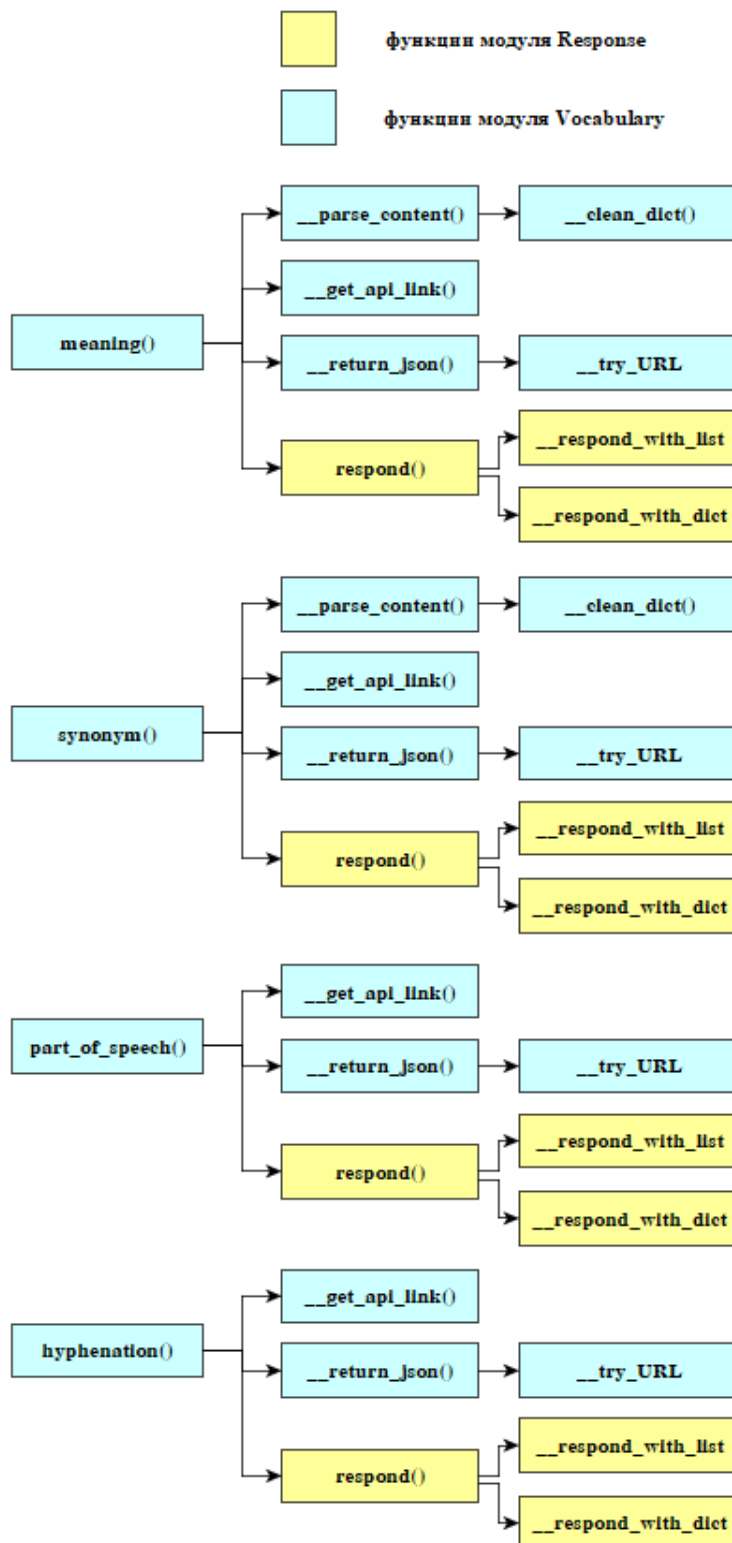
Перечень функциональностей объекта

- Ф1.** Вывод синонимов слова на разных языках
- Ф2.** Вывод антонимов слова на английском языке
- Ф3.** Вывод определений слова на разных языках
- Ф4.** Перевод слова на разные языки
- Ф5.** Вывод части речи слова на английском языке
- Ф6.** Вывод примеров использования слова на английском языке
- Ф7.** Вывод произношения слова на английском языке
- Ф8.** Вывод переносов слова на английском языке по строкам

2.2 Стратегия тестирования

Приложение состоит из двух модулей: модуль vocabulary и модуль responselib. Модуль vocabulary состоит из одного класса Vocabulary и 13 методов. Модуль responselib состоит из одного класса Response и 3 методов.

На рисунке 1 представлена граф вызова функций.



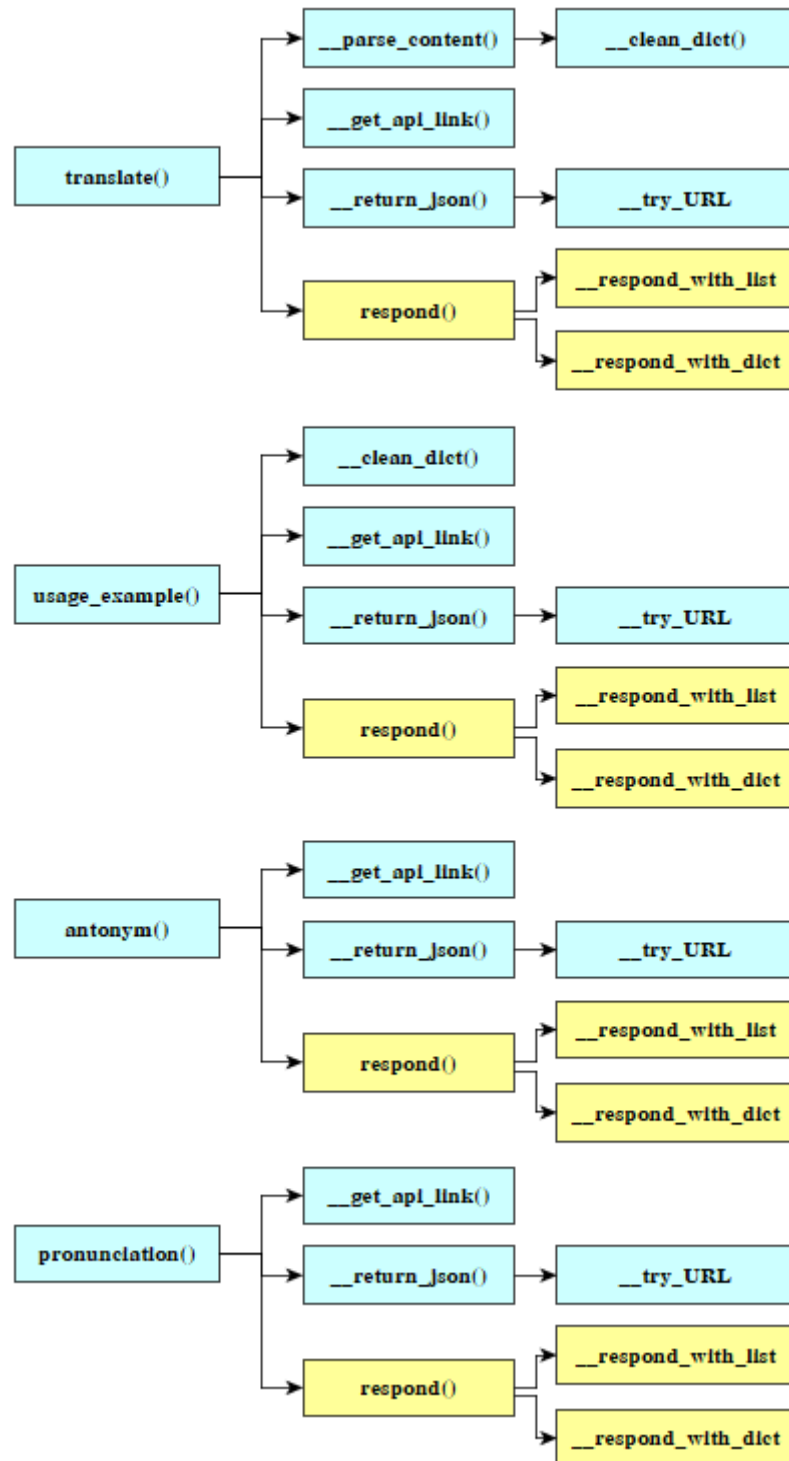


Рисунок 1. Граф вызова функций

Подробное описание методов:

- *__try_URL (message)* – выводит сообщение *message* об отсутствии подключения к Интернету. Функция тестируется интеграционными тестами.

- *__clean_dict (dictionary)* – преобразует поступающий на вход словарь *dictionary* к виду `[{'seq': 0, 'text': 'значение1'}, {'seq': 1, 'text': 'значение2'},...]`. Функция тестируется блочными и интеграционными тестами.
- *__parse_content (tuc_content, content_to_be_parsed)* – анализирует содержание, полученное от сторонних модулей. Возвращает список, который содержит отформатированные данные из содержания в виде json - объекта. *Tuc_content* – данные, полученный от стороннего модуля, *content_to_be_parsed* – данные, которые нужно получить из *tuc_content*. Функция тестируется блочными и интеграционными тестами.
- *__return_json (url)* – возвращает данные в формате json, полученные от следующего стороннего модуля *url*. Функция тестируется блочными и интеграционными тестами.
- *__respond_with_dict (self,data)* – преобразует данные *data* из формата json в словарь Python. Функция тестируется блочными и интеграционными тестами.
- *__respond_with_list (self,data)* – преобразует данные *data* из формата json в список Python. Функция тестируется блочными и интеграционными тестами.
- *respond (self, data, format)* – возвращает данные *data* в заданном формате *format*, по умолчанию формат имеет значение json. Функция тестируется интеграционными тестами
- *meaning (phrase, source_lang="en", dest_lang="en", format="json")* – возвращает определения слова *phrase* в заданном формате *format* в виде строки. По умолчанию *source_lang* и *dest_lang* имеют значение «en», *format* – «json». В случае неверного слова, возвращает False. Glosbe – используемый сторонний модуль. Функция тестируется блочными, интеграционными и аттестационными тестами.
- *synonym (phrase, source_lang="en", dest_lang="en", format="json")* – возвращает синонимы слова *phrase* в заданном формате *format* в виде строки. По умолчанию *source_lang* и *dest_lang* имеют значение «en», *format* – «json». В случае неверного слова, возвращает False. Glosbe – используемый сторонний модуль. Функция тестируется блочными, интеграционными и аттестационными тестами.

- *translate (phrase, source_lang, dest_lang, format="json")* – возвращает перевод слова *phrase* с языка *source_lang* на язык *dest_lang* в заданном формате *format* в виде строки. По умолчанию *format* имеет значение «json». В случае неверного слова, возвращает False. Glosbe – используемый сторонний модуль. Функция тестируется блочными, интеграционными и аттестационными тестами.
- *antonym (phrase, format="json")* – возвращает антонимы слова *phrase* в заданном формате *format* в виде строки. По умолчанию *format* имеет значение «json». В случае неверного слова, возвращает False. В случае, если антонимы не найдены, возвращает False. Bihugelabs – используемый сторонний модуль. Функция тестируется блочными, интеграционными и аттестационными тестами.
- *part_of_speech (phrase, format="json")* – возвращает часть речи слова *phrase* в заданном формате *format* в виде строки. По умолчанию *format* имеет значение «json». В случае неверного слова, возвращает False. Wordnik – используемый сторонний модуль. Функция тестируется блочными, интеграционными и аттестационными тестами.
- *usage_example (phrase, format="json")* – возвращает примеры использования слова *phrase* в предложениях (фразах) в заданном формате *format* в виде строки. По умолчанию *format* имеет значение «json». В случае неверного слова, возвращает False. Urbandict – используемый сторонний модуль. Функция тестируется блочными, интеграционными и аттестационными тестами.
- *pronunciation (phrase, format="json")* – возвращает произношение слова *phrase* в заданном формате *format* в виде строки. По умолчанию *format* имеет значение «json». В случае неверного слова, возвращает False. Wordnik – используемый сторонний модуль. Функция тестируется блочными, интеграционными и аттестационными тестами.
- *hyphenation (phrase, format="json")* – возвращает переносы слова *phrase* в заданном формате *format* в виде строки. По умолчанию *format* имеет значение «json». В случае неверного слова, возвращает False. Wordnik – используемый сторонний модуль. Функция тестируется блочными, интеграционными и аттестационными тестами.

2.3 Стратегия блочного тестирования

Объекты тестирования:

- модуль Vocabulary: `__parse_content()`, `__return_json()`, `__clean_dict()`, `meaning()`, `synonym()`, `translate()`, `antonym()`, `part_of_speech()`, `usage_example()`, `pronunciation()`, `hyphenation()`
- модуль ResponseLib: `__respond_with_list()`, `__respond_with_dict()`

Для проведения тестирования необходимы:

- консольное приложение IDLE Python,
- модуль unittest.

Для блочного тестирования необходимо определить все возможные варианты входных данных и ожидаемые результаты.

2.4 Стратегия интеграционного тестирования

Объектами интеграционного тестирования являются связки следующих функций:

- `respond ()` – `__respond_with_dict ()`
- `respond ()` – `__respond_with_list ()`
- `__return_json ()` – `__try_URL()`
- `__parse_content ()` – `__clean_dict ()`
- `synonym ()` – `respond()` – `__respond_with_dict()`
- `synonym ()` – `respond()`– `__respond_with_list()`
- `synonym ()` – `__return_json()` – `__try_URL()`
- `synonym ()` – `__get_api_link()`
- `synonym ()` – `__parse_content()` – `__clean_dict()`
- `translate ()` – `respond()` – `__respond_with_dict()`
- `translate ()` – `respond()`– `__respond_with_list()`
- `translate ()` – `__return_json()` – `__try_URL()`
- `translate ()` – `__get_api_link()`
- `translate ()` – `__parse_content()` – `__clean_dict()`
- `meaning ()` – `respond()` – `__respond_with_dict()`

- meaning () – respond() – __respond_with_list()
- meaning () – __return_json() – __try_URL()
- meaning () – __get_api_link()
- meaning () – __parse_content() – __clean_dict()
- antonym () – respond() – __respond_with_dict()
- antonym () – respond() – __respond_with_list()
- antonym () – __return_json() – __try_URL()
- antonym () – __get_api_link()
- usage_example () – respond() – __respond_with_dict()
- usage_example () – respond() – __respond_with_list()
- usage_example () – __return_json() – __try_URL()
- usage_example () – __get_api_link()
- usage_example() – __clean_dict()
- part_of_speech () – respond() – __respond_with_dict()
- part_of_speech () – respond() – __respond_with_list()
- part_of_speech () – __return_json() – __try_URL()
- part_of_speech () – __get_api_link()
- hyphenation () – respond() – __respond_with_dict()
- hyphenation () – respond() – __respond_with_list()
- hyphenation () – __return_json() – __try_URL()
- hyphenation () – __get_api_link()
- pronunciation () – respond() – __respond_with_dict()
- pronunciation () – respond() – __respond_with_list()
- pronunciation () – __return_json() – __try_URL()
- pronunciation () – __get_api_link()

На рисунке 2 представлены связки функций с этапами их выполнения:

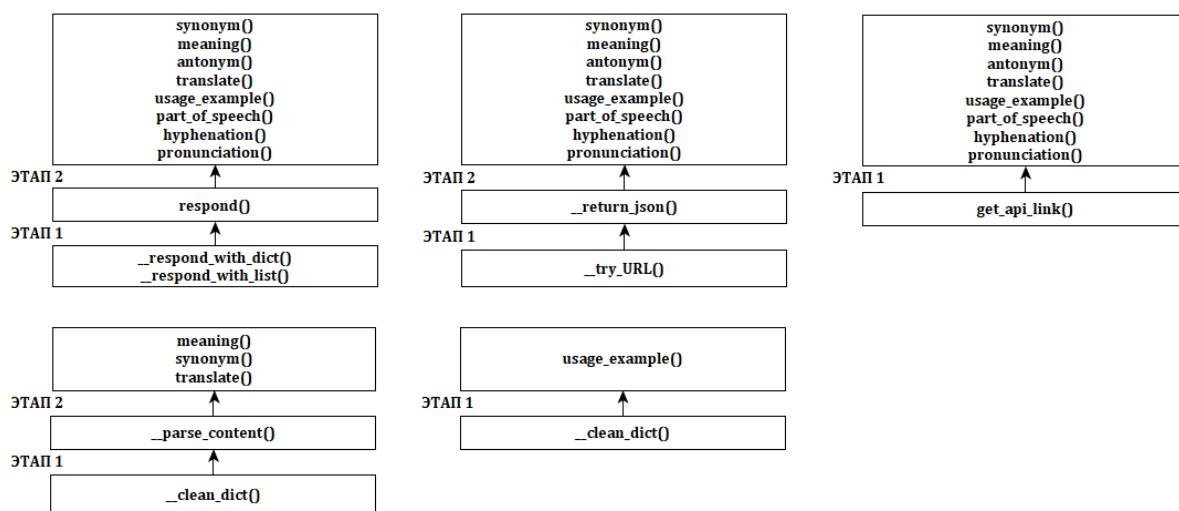


Рисунок 2. Интеграционное тестирование

Для проведения тестирования необходимы:

- консольное приложение IDLE Python,
- модуль unittest.

Для интеграционного тестирования необходимо определить все возможные варианты входных данных и ожидаемые результаты.

Для проведения интеграционного тестирования применяется стратегия восходящего тестирования.

2.5 Стратегия аттестационного тестирования

Объектами интеграционного тестирования являются следующие функциональности:

- **Ф1.** Вывод синонимов слова на разных языках
- **Ф2.** Вывод антонимов слова на английском языке
- **Ф3.** Вывод определений слова на разных языках
- **Ф4.** Перевод слова на разные языки
- **Ф5.** Вывод части речи слова на английском языке
- **Ф6.** Вывод примеров использования слова на английском языке
- **Ф7.** Вывод произношения слова на английском языке
- **Ф8.** Вывод переносов слова на английском языке по строкам

Для проведения тестирования необходимы:

- консольное приложение IDLE Python,
- модуль unittest.

2.6 Стратегия нагрузочного тестирования

В качестве специальных тестов будет проводиться нагрузочное тестирование сторонних модулей. Для этого будет осуществляться поступление множества запросов в тестируемом приложении. При множественном вызове функций приложения будет осуществляться нагрузка на сервера сторонних модулей. Поскольку некоторые сторонние модули используются несколькими сторонними модулями, то для тестирования будет браться лишь одна функция, вызывающая определенный сторонний модуль. Имитация множества запросов в приложении будет осуществляться с помощью разделения потоков.

Объекты тестирования:

- Функция `part_of_speech()`. Сторонний модуль Wordnik.

Заявленное количество одновременно поступающих запросов для стороннего модуля – 100 000

- Функция `antonym()`. Сторонний модуль Bighugelabs

Заявленное количество одновременно поступающих запросов для стороннего модуля – 10 000

- Функция `meaning()`. Сторонний модуль Glosbe

Заявленное количество одновременно поступающих запросов для стороннего модуля – 100 000

- Функция `usage_example()`. Сторонний модуль Urbandict

Заявленное количество одновременно поступающих запросов для стороннего модуля – 10 000

Для проведения тестирования необходимы:

- консольное приложение IDLE Python,
- модуль unittest.

Для нагрузочного тестирования необходимо определить в качестве входных данных количество запросов.

2.7 Критерии прохождения тестов

Любой тест считается успешно пройденным, если ожидаемый результат и фактический результат совпадают. Если тест завершается неудачей, то перед принятием решения целесообразно проверить правильность самого теста.

Если тест завершился неудачей и тест реализован правильно, то представляется заключение о найденной ошибке.

2.8 Критерий приостановления тестирования

Тестирование должно быть приостановлено, если количество не пройденных тестов превысит 40% от общего количества.

2.9 Критерий возобновления работы

Необходимо заново начать тестирование при получении уведомления, что найденные при тестировании ошибки исправлены.

3 Детальный план тестирования

3.1 Блочные тесты

Тест Б1

Функция. `__respond_with_dict(self,data)`

Описание. Проверка выполнения условия. Условие: если данные `data` являются списком, преобразовать их в словарь

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|--|--|
| Б1 | П | Данные <code>data</code> должны быть представлены в виде списка ['фраза1', 'фраза2', ...]. | Результат должен быть представлен в виде {0: 'фраза1', 1: 'фраза2', ...} |

Тест Б2

Функция. `__respond_with_dict(self,data)`

Описание. Проверка выполнения условия. Если элемент не представлен ни в одном из оговоренных форматов (список, словарь), то вывести ошибку

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|--|---------------------|
| Б2 | Н | Данные <code>data</code> должны быть представлены в любом формате, кроме словаря и списка. | False |

Тест Б3

Функция. `_respond_with_list(self,data)`

Описание. Проверка выполнения условия. Условие: если данные `data` являются словарем, преобразовать их в список

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|---|--|
| Б3 | П | Данные <code>data</code> должны быть представлены в виде словаря <code>{0: 'фраза1', 1: 'фраза2', ...}</code> | Результат должен быть представлен в виде списка <code>['фраза1', 'фраза2', ...]</code> . |

Тест Б4

Функция. `_respond_with_list(self,data)`

Описание. Проверка выполнения условия. Если элемент не представлен ни в одном из оговоренных форматов (список, словарь), то вывести ошибку.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|--|---------------------|
| Б4 | Н | Данные <code>data</code> должны быть представлены в любом формате, кроме словаря, списка и <code>json</code> . | False |

Тест Б5

Функция: `_return_json(url)`

Описание. Проверка выполнения условия. Если удалось получить ответ от стороннего модуля по ссылке `url`, то вывести ответ стороннего модуля, иначе ошибку.

Верный формат ссылки `url` в зависимости от модуля:

Wordnik:

http://api.wordnik.com/v4/word.json/{word}/{action}?api_key=1e940957819058fe3ec7c59d43c09504b400110db7faa0509

Glosbe:

<https://glosbe.com/gapi/translate?from={source lang}&dest={dest lang}&format=json&pretty=true&phrase={word}>

Urbandict: <http://api.urbandictionary.com/v0/{action}?term={word}>

Bighugelabs: <http://words.bighugelabs.com/api/2/eb4e57bb2c34032da68dfeb3a0578b68/{word}/json>

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|--|---|
| Б5.1 | П | Ссылка <code>url</code> должна быть представлена в верном формате в зависимости от используемого стороннего модуля. Вместо описанных в ссылке параметров должны быть прописаны текущие значения параметров: <code>word</code> – текущая фраза на языке, который сторонний модуль может обработать <code>source_lang</code> – текущий язык в сокращенном виде (<code>en</code> , <code>es</code> , <code>fr</code> , ...), на котором задана фраза <code>word</code> . <code>dest_lang</code> -текущий язык в сокращенном виде (<code>en</code> , <code>es</code> , <code>fr</code> , ...), на | Результат должен быть представлен в формате <code>json</code> <code>{ "tuc" : [{ "ключ0": [{ "ключ1": "значение1", "ключ2": "значение2", ... }, { "ключ1": "значение1", "ключ2": "значение2", ... }, ...] }, { "ключ1": [{ "ключ1": "значение1", "ключ2": "значение2", ... }, { "ключ1": "значение1", "ключ2": "значение2", ... }, ...] }, ...] }</code> |

| | | | |
|-------------|---|--|-------|
| | | <p>котором мы хотим получить ответ от модуля.</p> <p>Action – текущее действие, которое сторонний модуль может выполнить согласно своему функционалу. Для wordnik из диапазона ['definitions', 'pronunciations', 'hyphenation']. Для urbandict из диапазона ['define']</p> | |
| Б5.2 | Н | Ссылка url должна быть представлена в формате, писанном в тесте Б7.1 или должна быть пустой | False |

Тест Б6.

Функция: `_parse_content (tuc_content, content_to_be_parsed)`

Описание: проверка выполнения условия. Если `content_to_be_parsed` принимает значение из диапазона ['text', 'phrase'], то получить из словаря значения с ключом 'text' или 'phrase' и записать их в новый словарь.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|-------------|-----------|---|--|
| Б6.1 | П | <code>Content_to_be_parsed</code> принимает значения из диапазона ['text', 'phrase']. Словарь должен быть представлен в виде {'ключ1': "значение1", 'ключ2': "значение2", ... } | Новый словарь должен быть представлен в виде {0: "значение1", 1: "значение2", ... } со значениями, полученными согласно выбранному <code>content_to_be_parsed</code> |
| Б6.2 | Н | <code>Content_to_be_parsed</code> принимает любые значения, кроме ['text', 'phrase']. Словарь может быть представлен в виде {'ключ1': "значение1", 'ключ2': "значение2", ... } или может быть представлен не словарь, а например, список. | False |

Тест Б7.

Функция: `_parse_content (tuc_content, content_to_be_parsed)`

Описание: проверка выполнения условия. Если в словаре есть дубликаты, то удалить их.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|-----------|-----------|---|---|
| Б7 | П | Словарь должен быть представлен в виде {'ключ1': 'значение1', 'ключ2': 'значение1', 'ключ3': 'значение2', ... } | Новый словарь должен быть представлен в виде {0: 'значение1', 1: 'значение2', ... } и в нем не должно быть дубликатов |

Тест Б8

Функция: `_clean_dict (dictionary)`

Описание: проверка выполнения цикла. Для каждого ключа из dictionary прописать новое значение 'seq' и записать ключ и значение в новый словарь

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|--|--|
| Б8.1 | П | Цикл проходит по ключам из dictionary. Ключом является целое число. Должен быть передан в верном формате словарь dictionary {0: 'фраза1', 1: 'фраза2', ...} | Новый словарь должен быть представлен в виде {0: 'seq', 1: 'seq', ...} |
| Б8.2 | Н | Цикл проходит по ключам из dictionary. Ключом является целое число. Dictionary должен быть передан в неверном формате, который не должен соответствовать следующему: {0: 'фраза1', 1: 'фраза2', ...} | False |

Тест Б9

Функция: _clean_dict (dictionary)

Описание: проверка выполнения цикла. Каждое значение из dictionary объявить ключом и прописать им в качестве значений 'text'. Записать ключи и значения в новый словарь.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|--|--|
| Б9.1 | П | Цикл проходит по значениям из dictionary. Должен быть передан в верном формате словарь dictionary {0: 'фраза1', 1: 'фраза2', ...} | Новый словарь должен быть представлен в виде {'фраза1': 'text', 'фраза2': 'text', ...} |
| Б9.2 | Н | Цикл проходит по значениям из dictionary. Dictionary должен быть передан в неверном формате, который не должен соответствовать следующему: {0: 'фраза1', 1: 'фраза2', ...} | False |

Тест Б10

Функция: _clean_dict (dictionary)

Описание: проверка выполнения цикла. Склеить два словаря с ключами и значениями в один список.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|---|---|
| Б10.1 | П | Цикл проходит по значениям и ключам из key_dict (новый словарь с ключами из dictionary и значениями seq) и value_dict (новый словарь с ключами, представленными в виде значений из dictionary и значениями) | Новый список должен быть представлен в виде [{'seq': 0, 'text': 'фраза1'}, {'seq': 1, 'text': 'фраза2'}, ...] |

| | | | |
|--------------|---|--|-------|
| | | 'text'). key_dict должен быть представлен в виде {0: 'seq', 1: 'seq', ...}. value_dict должен быть представлен в виде {'фраза1': 'text', 'фраза2': 'text', ...} | |
| Б10.2 | Н | Цикл проходит по значениям и ключам из key_dict (новый словарь с ключами из dictionary и значениями seq) и value_dict (новый словарь с ключами, представленными в виде значений из dictionary и значениями 'text'). key_dict должен быть представлен в неверном формате, который не соответствует виду {0: 'seq', 1: 'seq', ...}. value_dict должен быть представлен в неверном формате, который не соответствует виду {'фраза1': 'text', 'фраза2': 'text', ...} | False |

Тест Б11

Функция: meaning (phrase, source_lang="en", dest_lang="en", format="json")

Описание: проверка выполнения условия. Если полученные данные от стороннего модуля являются json – объектом, то выделить необходимые данные из полученных данных, иначе вывести ошибку

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|--------------|-----------|---|--|
| Б11.1 | П | Полученные данные должны быть представлены в виде {'ключ1': 'значение1', 'tuc': [{'meanings': [{ 'language': 'значение3', 'text': 'значение4'}, { 'language': 'значение5', 'text': 'значение6', ...}], ...}. | Результат должен быть представлен в виде [{'seq': 0, 'text': 'значение4'}, {'seq': 1, 'text': 'значение6'}, ...]. В качестве значений берутся данные по ключу 'text' из списка с ключом 'meanings'. Нумерация ведется от 0 до количества значений с ключом text из списка с ключом 'meanings'. |
| Б11.2 | Н | Полученные данные не должны быть представлены в виде {'ключ1': 'значение1', 'tuc': [{'meanings': [{ 'language': 'значение3', 'text': 'значение4'}, { 'language': 'значение5', 'text': 'значение6', ...}], ...}. | False |

Тест Б12

Функция: synonym (phrase, source_lang="en", dest_lang="en", format="json")

Описание: проверка выполнения условия. Если полученные данные от стороннего модуля являются json – объектом, то выделить необходимые данные из полученных данных, иначе вывести ошибку

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|--|---|
| Б12.1 | П | Полученные данные должны быть представлены в виде {'ключ1': 'значение1', 'tuc': [{'phrase': [{'language': 'значение3', 'text': 'значение4'}, {'phrase': 'значение5', 'language': 'значение6', ...}], ...} | Результат должен быть представлен в виде [{'seq': 0, 'text': 'значение4'}, {'seq': 1, 'text': 'значение6'}, ...]. В качестве значений берутся данные по ключу 'text' из списка с ключом 'phrase'. Нумерация ведется от 0 до количества значений с ключом text из списка с ключом 'phrase'. |
| Б12.2 | Н | Полученные данные не должны быть представлены в виде {'ключ1': 'значение1', 'tuc': [{'phrase': [{'language': 'значение3', 'text': 'значение4'}, {'language': 'значение5', 'text': 'значение6', ...}], ...} | False |

Тест Б13

Функция: translate (phrase, source_lang="en", dest_lang="en", format="json")

Описание: проверка выполнения условия. Если полученные данные от стороннего модуля являются json – объектом, то выделить необходимые данные из полученных данных, иначе вывести ошибку

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|---|---|
| Б13.1 | П | Полученные данные должны быть представлены в виде {'ключ1': 'значение1', 'tuc': [{'meanings': [{'language': 'значение3', 'text': 'значение4'}, {'language': 'значение5', 'text': 'значение6', ...}], ...} | Результат должен быть представлен в виде [{'seq': 0, 'text': 'значение1'}, {'seq': 1, 'text': 'значение2'}, ...]. В качестве значений берутся данные по ключу 'text' из списка с ключом 'meanings', где значения с ключом 'language' принимают значение параметра dest_lang. Нумерация ведется от 0 до количества значений с ключом 'text' из списка с ключом 'meanings', где значения с ключом 'language' принимают значение параметра dest_lang. |
| Б13.2 | Н | Полученные данные не должны быть представлены в виде {'ключ1': 'значение1', 'tuc': [{'meanings': [{'language': 'значение3', 'text': | False |

| | | | |
|--|--|--|--|
| | | {'значение4'}, {'language':'значение5', 'text':'значение6',...}], ...} | |
|--|--|--|--|

Тест Б14

Функция: antonym (phrase, format="json")

Описание: проверка выполнения условия. Если полученные данные от стороннего модуля являются json – объектом, то выделить необходимые данные из полученных данных, иначе вывести ошибку

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|--|--|
| Б14.1 | П | Полученные данные должны быть представлены в виде <pre>{ "часть_речи1": { "syn": ["знач1", "знач2", ...], "ant": ["знач1", "знач2", ...], "rel": ["знач1", "знач2", ...], "sim": ["знач1", "знач2", ...] }, "часть_речи2": { "syn": ["знач1", "знач2", ...], "ant": ["знач1", "знач2", ...], "rel": ["знач1", "знач2", ...], "sim": ["знач1", "знач2", ...] }, ... }</pre> | Результат должен быть представлен в виде <pre>[{'seq': 0, 'text': значение1}, {'seq': 1, 'text': значение2}, ...].</pre> В качестве значений берутся данные по ключу 'ant'. Нумерация ведется от 0 до количества значений в списке с ключом 'syn' |
| Б14.2 | Н | Полученные данные не должны быть представлены в виде <pre>{ "часть_речи1": { "syn": ["знач1", "знач2", ...], "ant": ["знач1", "знач2", ...], "rel": ["знач1", "знач2", ...], "sim": ["знач1", "знач2", ...] }, "часть_речи2": { "syn": ["знач1", "знач2", ...], "ant": ["знач1", "знач2", ...], "rel": ["знач1", "знач2", ...], "sim": ["знач1", "знач2", ...] }, ... }</pre> | False |

Тест Б15

Функция: hyphenation (phrase, format="json")

Описание: проверка выполнения условия. Если полученные данные от стороннего модуля являются json – объектом, то выделить необходимые данные из полученных данных, иначе вывести ошибку

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|--|--|
| Б15.1 | П | Полученные данные должны быть представлены в виде <pre>[{"seq": 0, "text": "знач1"}, {"seq": 1, "text": "знач2"}, ...].</pre> Format должен принимать значение из следующего диапазона ['json', 'dict', 'list'] | Результат должен быть представлен в виде согласно format: List: ['знач1', 'знач2',...] Dict: {0:'знач1', 1: 'знач2', ...} Json: [{"seq": 0, "text": "знач1"}, {"seq": 1, "text": "знач2"}, ...] |
| Б15.2 | Н | Полученные данные не должны быть представлены в виде | False |

| | | | |
|--|--|--|--|
| | | <pre>[{"seq": 0, "text": "знач1"}, {"seq": 1, "text": "знач2"}, ...]</pre> | |
|--|--|--|--|

Тест Б16

Функция: pronunciation (phrase, format="json")

Описание: проверка выполнения условия. Если полученные данные от стороннего модуля являются json – объектом, то выделить необходимые данные из полученных данных, иначе вывести ошибку

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|--|---|
| Б16.1 | П | <p>Полученные данные должны быть представлены в виде <pre>[{"rawType": "знач1", "seq": знач2, "raw": "знач3"}, {"rawType": "знач1", "seq": знач2, "raw": "знач3"}, ...]</pre></p> <p>Format должен принимать значение из следующего диапазона ['json', 'dict', 'list']</p> | <p>Результат должен быть представлен в виде согласно format:</p> <p>List: [['знач1', 'знач3'], ['знач4', 'знач6'], ...]</p> <p>Dict: {знач2: {'rawType': 'знач1', 'raw': 'знач3'}, знач5: {'rawType': 'знач4', 'raw': 'знач5'}, ...}</p> <p>Json: [{"rawType": "знач1", "seq": знач2, "raw": "знач3"}, {"rawType": "знач1", "seq": знач2, "raw": "знач3"}, ...]</p> |
| Б16.2 | Н | <p>Полученные данные не должны быть представлены в виде <pre>[{"rawType": "знач1", "seq": знач2, "raw": "знач3"}, {"rawType": "знач1", "seq": знач2, "raw": "знач3"}, ...]</pre></p> | False |

Тест Б17

Функция: part_of_speech (phrase, format="json")

Описание: проверка выполнения условия. Если полученные данные от стороннего модуля являются json – объектом, то выделить необходимые данные из полученных данных, иначе вывести ошибку

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|--|---|
| Б17.1 | П | <p>Полученные данные должны быть представлены в виде <pre>[{"ключ1": "знач1", "ключ2": "знач2", "partOfSpeech": "знач3", ...}, {"ключ1": "знач1", "ключ2": "знач2", "partOfSpeech": "знач3", ...}, ...]</pre></p> <p>Format должен принимать значение из следующего диапазона ['json', 'dict', 'list']</p> | <p>Результат должен быть представлен в виде согласно format:</p> <p>List: ['знач1', 'знач2', ...]</p> <p>Dict: {0: 'знач 1', 1: 'знач2', ...}</p> <p>Json: [{"seq": 0, "text": "знач3"}, ...]</p> |
| Б17.2 | Н | <p>Полученные данные не должны быть представлены в виде <pre>[{"ключ1": "знач1", "ключ2":</pre></p> | False |

| | | | |
|--|--|--|--|
| | | "знач2", "partOfSpeech": "знач3", ...}, {"ключ1": "знач1", "ключ2": "знач2", "partOfSpeech": "знач3", ...}, ...] | |
|--|--|--|--|

Тест Б18

Функция: usage_example (phrase, format="json")

Описание: проверка выполнения условия. Если полученные данные от стороннего модуля являются json – объектом, то выделить необходимые данные из полученных данных, иначе вывести ошибку

| № теста | Тип Тест | Входные данные | Ожидаемый результат |
|---------|----------|--|--|
| Б18.1 | П | Полученные данные должны быть представлены в виде { 'ключ1': 'знач1', 'ключ2': 'знач2', 'example': 'знач3', ... } Format должен принимать значение из следующего диапазона ['json', 'dict', 'list'] | Результат должен быть представлен в виде согласно format: List: ['знач3', ...] Dict: {0: 'знач3', ...} Json: [{"seq": 0, "text": "знач3"}, ...] |
| Б18.2 | Н | Полученные данные не должны быть представлены в виде { 'ключ1': 'знач1', 'ключ2': 'знач2', 'example': 'знач3', ... } | False |

3.2 Интеграционные тесты

Тест И1. Этап 1. Взаимодействие _parse_content() – _clean_dict()

Описание теста: тест проверяет правильность преобразования данных полученных от стороннего модуля. Параметры content_to_be_parsed и tuc_content передаются в функцию _parse_content(). Функция _parse_content вызывает функцию _clean_dict().

| № теста | Тип Тест | Входные данные | Ожидаемый результат |
|---------|----------|--|---|
| И1 | 0 | content_to_be_parsed принимает значения из диапазона ['text', 'phrase']. Словарь tuc_content должен быть представлен в виде { 'ключ1': "значение1", 'ключ2': 'значение2', ... }. | Словарь должен быть представлен в виде {0: 'значение1', 1: 'значение2', ... } со значениями, полученными согласно выбранному content_to_be_parsed |

Тест И2. Этап 1. Взаимодействие _return_json() → _try_url()

Описание: тест проверяет вывод сообщения об отсутствии подключения к стороннему модулю. Параметр url передается в функцию _try_url(), остальные параметры передаются в функцию _return_json(). Функция _return_json() вызывает функцию _try_url().

| № теста | Тип Тест | Входные данные | Ожидаемый результат |
|---------|----------|---|-----------------------------------|
| И1 | 0 | ссылка url должна быть представлена в верном формате в зависимости от используемого стороннего модуля. Вместо описанных в ссылке параметров должны быть прописаны текущие значения параметров: word – текущая фраза на языке, который сторонний модуль может обработать source_lang – текущий язык в сокращенном виде (en, es, fr, ...), на котором задана фраза word. dest_lang - текущий язык в сокращенном виде (en, es, fr, ...), на котором мы хотим получить ответ от модуля. action – текущее действие, которое сторонний модуль может выполнить согласно своему функционалу. Для wordnik из диапазона ['definitions', 'pronunciations', 'hyphenation']. Для urbandict из диапазона ['define'] | вывод сообщения 'Connection Lost' |

Тест И3. Этап 1. Взаимодействие `respond()` → `_respond_with_dict()`

Описание: тест проверяет правильность получения преобразованных данных при задании соответствующих значений параметров `format` и `data`. Параметры `format` и `data` передаются в функцию `respond()`, параметр `data` передается в функцию `_respond_with_dict()`. Функция `respond()` вызывает функцию `_respond_with_dict()`.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|--|---------------------------------------|
| И3.1 | П | <code>format= 'dict'</code> Данные <code>data</code> представлены в виде списка следующего вида ['значение1', 'значение2', ...] | {0: 'значение1', 1: 'значение2', ...} |
| И3.2 | Н | <code>format = 'dict'</code> Данные <code>data</code> представлены в формате, который не соответствует виду ['значение1', 'значение2', ...] или пусты | False |

Тест И4. Этап 1. Взаимодействие `respond()` → `_respond_with_list()`

Описание: тест проверяет правильность получения преобразованных данных при задании соответствующих значений параметров `format` и `data`. Параметры `format` и `data` передаются в функцию `_respond()`, параметр `data` передается в функцию `_respond_with_list()`. Функция `respond()` вызывает функцию `_respond_with_list()`.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|---|--|
| И4.1 | П | <code>format = 'list'</code> Данные <code>data</code> представлены в виде словаря следующего вида <code>{0: 'значение1', 1: 'значение2', ...}</code> | <code>['значение1', 'значение2', ...]</code> |
| И4.2 | Н | <code>format = 'list'</code> Данные <code>data</code> представлены в формате, который не соответствует виду <code>{0: 'значение1', 1: 'значение2', ...}</code> или пусты | <code>False</code> |

Тест И5. Этап 1. Взаимодействие `meaning()` → `_get_api_link()`

Описание: тест проверяет правильность получения ссылки при задании соответствующего параметра `api`. Параметр `api` передается в функцию `get_api_link`. Функция `meaning()` вызывает функцию `_get_api_link()`.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|--|---|
| И5.1 | П | <code>api = 'glosbe'</code> | <code>https://glosbe.com/gapi/translate?from={edest_lang}&dest={source_lang}&format=json&pretty=true&phrase={word}</code> |
| И5.2 | Н | <code>api</code> может принять любое значение отличное <code>'glosbe'</code> | <code>False</code> |

Тест И6. Этап 2. Взаимодействие `meaning()` → `_parse_content()` → `_clean_dict()`

Описание: тест проверяет правильность преобразования данных к заданному виду. Параметры передаются в функцию `meaning()`. Функция `meaning()` вызывает функцию `_parse_content`, которая в свою очередь обращается к функции `_clean_dict()`. Взаимодействие `_parse_content()` и `_clean_dict()` проверяется тестом И1.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|---|--|
| И6.1 | П | <code>phrase</code> – слово, заданное на языке <code>source_lang</code> <code>source_lang</code> – язык, на котором задано слово <code>phrase</code> . Язык должен быть задан в сокращенном виде, например: <code>es</code> , <code>en</code> , <code>fr</code> <code>dest_lang</code> – язык, на котором будут | Получение определений слова <code>phrase</code> , заданного на языке <code>source_lang</code> , на языке <code>dest_lang</code> в виде <code>[[{seq: 0, 'text': 'значение1'}, {seq: 1, 'text': 'значение2', ...}]</code> |

| | | | |
|-------------|---|--|-------|
| | | выведены определения слова. Язык должен быть задан в сокращенном виде, например: es, en, fr format – формат, в котором выводить синонимы. Значение выбирается из диапазона ['json', 'list', 'dict'] | |
| И6.2 | Н | Одно или несколько из следующих параметров заданы в формате, который не соответствует формату описанному в тесте И6.1 | False |

Тест И7. Этап 2. Взаимодействие meaning() → _return_json() → _try_URL()

Описание: тест проверяет правильность получения данных функцией от стороннего модуля. Параметры передаются в функцию meaning(). Функция meaning() вызывает функцию _return_json(), которая вызывает функцию _try_URL(). Взаимодействие функций _try_URL() и _return_json() проверяется тестом И2

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|-------------|-----------|---|--|
| И7.1 | П | word – текущая фраза на языке, который сторонний модуль может обработать source_lang – текущий язык в сокращенном виде (en, es, fr, ...), на котором задана фраза word. dest_lang -текущий язык в сокращенном виде (en, es, fr, ...), на котором мы хотим получить ответ от модуля. | Результат должен быть представлен в формате json { "tuc": [{ "ключ0": [{ "ключ1": "значение1", "ключ2": "значение2", ... }, { "ключ1": "значение1", "ключ2": "значение2", ... }, ...] , "ключ1": [{ "ключ1": "значение1", "ключ2": "значение2", ... }, { "ключ1": "значение1", "ключ2": "значение2", ... }, ...] }] } |
| И7.2 | Н | Один или несколько параметров не должны соответствовать формату, описанному в тесте И7.1 | False |

Тест И8. Этап 2. Взаимодействие meaning() → respond() → _respond_with_dict()

Описание: тест проверяет правильность преобразования обработанных данных к заданному формату. Параметры передаются в функцию meaning(). Функция meaning() вызывает функцию respond(), которая осуществляет вызов функции _respond_with_dict(). Взаимодействие функций respond() и _respond_with_dict() проверяется тестом И3.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|----------------|---------------------|
|---------|-----------|----------------|---------------------|

| | | | |
|-------------|---|---|---------------------------------------|
| И8.1 | П | word – текущая фраза на языке, который сторонний модуль может обработать source_lang – текущий язык в сокращенном виде (en, es, fr, ...), на котором задана фраза word. dest_lang - текущий язык в сокращенном виде (en, es, fr, ...), на котором мы хотим получить ответ от модуля. Format = 'dict' | {0: 'значение1', 1: 'значение2', ...} |
| И8.2 | Н | Один или несколько параметров не должны соответствовать формату, описанному в тесте И8.1 | False |

Тест И9. Этап 2. Взаимодействие meaning() → respond() → _respond_with_list()

Описание: тест проверяет правильность преобразования обработанных данных к заданному формату. Параметры передаются в функцию meaning(). Функция meaning() вызывает функцию respond(), которая осуществляет вызов функции _respond_with_list(). Взаимодействие функций respond() и _respond_with_list() проверяется тестом И4.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|-------------|-----------|---|---------------------------------|
| И9.1 | П | word – текущая фраза на языке, который сторонний модуль может обработать source_lang – текущий язык в сокращенном виде (en, es, fr, ...), на котором задана фраза word. dest_lang - текущий язык в сокращенном виде (en, es, fr, ...), на котором мы хотим получить ответ от модуля. Format = 'list' | ['значение1', 'значение2', ...] |
| И9.2 | Н | Один или несколько параметров не должны соответствовать формату, описанному в тесте И9.1 | False |

Тест И10. Этап 1. Взаимодействие synonym() → _get_api_link()

Описание: тест проверяет правильность получения ссылки при задании соответствующего параметра api. Параметр api передается в функцию get_api_link. Функция synonym() вызывает функцию _get_api_link().

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|-------------|-----------|----------------|---|
| И10. | П | api = 'glosbe' | https://glosbe.com/gapi/translate |

| | | | |
|-------|---|--|---|
| 1 | | | ?from={edest_lang}&dest={source_lang}&format=json&pretty=true&phrase={word} |
| И10.2 | Н | api может принять любое значение неравное 'glosbe' | False |

Тест И11. Этап 2. Взаимодействие synonym() → _parse_content() → _clean_dict()

Описание: тест проверяет правильность преобразования полученных данных к заданному виду. Параметры передаются в функцию synonym(). Функция synonym() вызывает функцию _parse_content, которая в свою очередь обращается к функции _clean_dict(). Взаимодействие _parse_content() и _clean_dict() проверяется тестом И1.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|--|---|
| И11.1 | П | phrase – слово, заданное на языке source_lang source_lang – язык, на котором задано слово phrase. Язык должен быть задан в сокращенном виде, например: es, en, fr dest_lang – язык, на котором будут выведены синонимы слова. Язык должен быть задан в сокращенном виде, например: es, en, fr format – формат, в котором выводить синонимы. Значение выбирается из диапазона ['json', 'list', 'dict'] | получение синонимов слова phrase, заданного на языке source_lang, на языке dest_lang в виде [{seq: 0, 'text': 'значение1'}, {'seq': 1, 'text': 'значение2', ...}] |
| И11.2 | Н | Один или несколько параметров не должны соответствовать формату, описанному в тесте И11.1 | False |

Тест И12. Этап 2. Взаимодействие synonym() → _return_json() → _try_URL()

Описание: тест проверяет правильность получения данных от стороннего модуля. Параметры передаются в функцию synonym(). Функция synonym() вызывает функцию _return_json(), которая вызывает функцию _try_URL(). Взаимодействие функций _try_URL() и _return_json() проверяется тестом И2

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|--|--|
| И12.1 | П | word – текущая фраза на языке, который сторонний модуль может обработать source_lang – текущий язык | Результат должен быть представлен в формате json { "tuc" : [{ "ключ0": [{ "ключ1": "значение1", |

| | | | |
|--------------|---|--|---|
| | | сокращенном виде (en, es, fr, ...), на котором задана фраза word. dest_lang -текущий язык в сокращенном виде (en, es, fr, ...), на котором мы хотим получить ответ от модуля. | “ключ2”: “значение2”, ... }, { “ключ1”: “значение1”, “ключ2”: “значение2”, ... }, ... }], “ключ1”: [{ “ключ1”: “значение1”, “ключ2”: “значение2”, ... }, { “ключ1”: “значение1”, “ключ2”: “значение2”, ... }, ... }], ...] |
| И12.2 | Н | Один или несколько параметров не должны соответствовать формату, описанному в тесте И12.1 | False |

Тест И13. Этап 2. Взаимодействие synonym() → respond() → _respond_with_dict()

Описание: тест проверяет правильность преобразования обработанных данных к заданному формату. Параметры передаются в функцию synonym(). Функция synonym() вызывает функцию respond(), которая осуществляет вызов функции _respond_with_dict(). Взаимодействие функций respond() и _respond_with_dict() проверяется тестом И3.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|--------------|-----------|---|---------------------------------------|
| И13.1 | П | word – текущая фраза на языке, который сторонний модуль может обработать source_lang – текущий язык в сокращенном виде (en, es, fr, ...), на котором задана фраза word. dest_lang - текущий язык в сокращенном виде (en, es, fr, ...), на котором мы хотим получить ответ от модуля. Format = 'dict' | {0: 'значение1', 1: 'значение2', ...} |
| И13.2 | Н | Один или несколько параметров не должны соответствовать формату, описанному в тесте И13.1 | False |

Тест И14. Этап 2. Взаимодействие synonym() → respond() → _respond_with_list()

Описание: тест проверяет правильность преобразования обработанных данных к заданному формату. Параметры передаются в функцию synonym(). Функция synonym() вызывает функцию respond(), которая осуществляет вызов функции _respond_with_list(). Взаимодействие функций respond() и _respond_with_list() проверяется тестом И4.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|----------------|---------------------|
|---------|-----------|----------------|---------------------|

| | | | |
|--------------|---|---|---------------------------------|
| И14.1 | П | word – текущая фраза на языке, который сторонний модуль может обработать source_lang – текущий язык в сокращенном виде (en, es, fr, ...), на котором задана фраза word. dest_lang - текущий язык в сокращенном виде (en, es, fr, ...), на котором мы хотим получить ответ от модуля. Format = 'list' | ['значение1', 'значение2', ...] |
| И14.2 | Н | Один или несколько параметров не должны соответствовать формату, описанному в тесте И14.1 | False |

Тест И15. Этап 1. Взаимодействие translate() → _get_api_link()

Описание: тест проверяет правильность получения ссылки при задании соответствующего параметра api. Параметр api передается в функцию get_api_link. Функция translate() вызывает функцию _get_api_link().

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|--------------|-----------|--|---|
| И15.1 | П | api = 'glosbe' | https://glosbe.com/gapi/translate?from={dest_lang}&dest={source_lang}&format=json&pretty=true&phrase={word} |
| И15.2 | Н | api может принять любое значение отличное 'glosbe' | False |

Тест И16. Этап 2. Взаимодействие translate() → _parse_content() → _clean_dict()

Описание: тест проверяет правильность преобразования полученных данных к заданному виду. Параметры передаются в функцию translate(). Функция translate() вызывает функцию _parse_content, которая в свою очередь обращается к функции _clean_dict(). Взаимодействие _parse_content() и _clean_dict() проверяется тестом И1.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|--------------|-----------|--|---|
| И16.1 | П | phrase – слово, заданное на языке source_lang source_lang – язык, на котором задано слово phrase. Язык должен быть задан в сокращенном виде, например: es, en, fr dest_lang – язык, на котором будут выведены переводы слова. Язык должен быть задан в сокращенном виде, например: es, | Получение переводов слова phrase, заданного на языке source_lang, на языке dest_lang в виде [{seq: 0, 'text': 'значение1'}, {seq: 1, 'text': 'значение2', ...}] |

| | | | |
|--------------|---|---|-------|
| | | en, fr format – формат, в котором выводить синонимы. Значение выбирается из диапазона ['json', 'list', 'dict'] | |
| И16.2 | Н | Один или несколько параметров не должны соответствовать формату, описанному в тесте И16.1 | False |

Тест И17. Этап 2. Взаимодействие translate() → __return_json() → __try_URL()

Описание: тест проверяет правильность получения данных от стороннего модуля. Параметры передаются в функцию translate(). Функция translate() вызывает функцию __return_json(), которая вызывает функцию __try_URL(). Взаимодействие функций __try_URL() и __return_json() проверяется тестом И2

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|--------------|-----------|--|--|
| И17.1 | П | word – текущая фраза на языке, который сторонний модуль может обработать source_lang – текущий язык в сокращенном виде (en, es, fr, ...), на котором задана фраза word. dest_lang - текущий язык в сокращенном виде (en, es, fr, ...), на котором мы хотим получить ответ от модуля. | Результат должен быть представлен в формате json { "tuc": [{ "ключ0": [{ "ключ1": "значение1", "ключ2": "значение2", ... }, { "ключ1": "значение1", "ключ2": "значение2", ... }, ...] , "ключ1": [{ "ключ1": "значение1", "ключ2": "значение2", ... }, { "ключ1": "значение1", "ключ2": "значение2", ... }, ...] }] } |
| И17.2 | Н | Один или несколько параметров не должны соответствовать формату, описанному в тесте И17.1 | False |

Тест И18. Этап 2. Взаимодействие translate() → respond() → __respond_with_dict()

Описание: тест проверяет правильность преобразования обработанных данных к заданному формату. Параметры передаются в функцию translate(). Функция translate() вызывает функцию respond(), которая осуществляет вызов функции __respond_with_dict(). Взаимодействие функций respond() и __respond_with_dict() проверяется тестом И3.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|--------------|-----------|--|---------------------------------------|
| И18.1 | П | word – текущая фраза на языке, который сторонний модуль может обработать | {0: 'значение1', 1: 'значение2', ...} |

| | | | |
|--------------|---|---|-------|
| | | source_lang – текущий язык в сокращенном виде (en, es, fr, ...), на котором задана фраза word. dest_lang - текущий язык в сокращенном виде (en, es, fr, ...), на котором мы хотим получить ответ от модуля. Format = 'dict' | |
| И18.2 | Н | Один или несколько параметров не должны соответствовать формату, описанному в тесте И18.1 | False |

Тест И19. Этап 1. Взаимодействие translate() → respond() → _respond_with_list()

Описание: тест проверяет правильность преобразования обработанных данных к заданному формату. Параметры передаются в функцию translate(). Функция translate() вызывает функцию respond(), которая осуществляет вызов функции _respond_with_list(). Взаимодействие функций respond() и _respond_with_list() проверяется тестом И4.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|--------------|-----------|---|---------------------------------|
| И19.1 | П | word – текущая фраза на языке, который сторонний модуль может обработать source_lang – текущий язык в сокращенном виде (en, es, fr, ...), на котором задана фраза word. dest_lang - текущий язык в сокращенном виде (en, es, fr, ...), на котором мы хотим получить ответ от модуля. Format = 'list' | ['значение1', 'значение2', ...] |
| И19.2 | Н | Один или несколько параметров не должны соответствовать формату, описанному в тесте И19.1 | False |

Тест И20. Этап 1. Взаимодействие usage_example() → _get_api_link()

Описание: тест проверяет правильность получения ссылки при задании соответствующего параметра api. Параметр api передается в функцию get_api_link. Функция usage_example() вызывает функцию _get_api_link().

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|--------------|-----------|-------------------------|--|
| И20.1 | П | api = 'urbandict' | http://api.urbandictionary.com/v0/{action}?term={word} |
| И20.2 | Н | api может принять любое | False |

| | | | |
|--|--|-------------------------------|--|
| | | значение неравное 'urbandict' | |
|--|--|-------------------------------|--|

Тест И21. Этап 2. Взаимодействие usage_example() → _clean_dict()

Описание: тест проверяет правильность преобразования полученных от стороннего модуля данных. Параметры передаются в функцию usage_example(). Функция usage_example() вызывает функцию _clean_dict().

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|---|--------------------------------------|
| И21.1 | П | phrase – слово, заданное на английском языке format – формат, в котором выводить синонимы. Значение выбирается из диапазона ['json', 'list', 'dict'] | {0:'значение1', 1:'значение2', ... } |
| И21.2 | Н | Один или несколько параметров не должны соответствовать формату, описанному в тесте И21.1 | False |

Тест И22. Этап 2. Взаимодействие usage_example() → _return_json() → _try_URL()

Описание: тест проверяет правильность получения данных от стороннего модуля. Параметры передаются в функцию usage_example(). Функция usage_example() вызывает функцию _return_json(), которая вызывает функцию _try_URL(). Взаимодействие функций _try_URL() и _return_json() проверяется тестом И2

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|---|--|
| И22.1 | П | word – текущая фраза на английском языке | Результат должен быть представлен в формате json { "tuc": [{ "ключ0": [{ "ключ1": "значение1", "ключ2": "значение2", ... }, { "ключ1": "значение1", "ключ2": "значение2", ... }, ...] , "ключ1": [{ "ключ1": "значение1", "ключ2": "значение2", ... }, { "ключ1": "значение1", "ключ2": "значение2", ... }, ...] }] } |
| И22.2 | Н | Один или несколько параметров не должны соответствовать формату, описанному в тесте И22.1 | False |

Тест И23. Этап 2. Взаимодействие usage_example() → respond() → _respond_with_dict()

Описание: тест проверяет правильность преобразования обработанных данных к заданному формату. Параметры передаются в функцию `usage_example()`. Функция `usage_example()` вызывает функцию `respond()`, которая осуществляет вызов функции `__respond_with_dict()`. Взаимодействие функций `respond()` и `__respond_with_dict()` проверяется тестом И3.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|---|---------------------------------------|
| И23.1 | П | word – текущая фраза на английском языке Format = 'dict' | {0: 'значение1', 1: 'значение2', ...} |
| И23.2 | Н | Один или несколько параметров не должны соответствовать формату, описанному в тесте И23.1 | False |

Тест И24. Взаимодействие `usage_example()` → `respond()` → `__respond_with_list()`

Описание: тест проверяет правильность преобразования обработанных данных к заданному формату. Параметры передаются в функцию `usage_example()`. Функция `usage_example()` вызывает функцию `respond()`, которая осуществляет вызов функции `__respond_with_list()`. Взаимодействие функций `respond()` и `__respond_with_list()` проверяется тестом И4.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|---|---------------------------------|
| И24.1 | П | word – текущая фраза на языке, который сторонний модуль может обработать Format = 'list' | ['значение1', 'значение2', ...] |
| И24.2 | Н | Один или несколько параметров не должны соответствовать формату, описанному в тесте И24.1 | False |

Тест И25. Этап 1. Взаимодействие `antonym()` → `__get_api_link()`

Описание: тест проверяет правильность получения ссылки при задании соответствующего параметра `api`. Параметр `api` передается в функцию `get_api_link`. Функция `antonym()` вызывает функцию `__get_api_link()`.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|-------------------------|---|
| И25.1 | П | api = 'bighugelabs' | http://words.bighugelabs.com/api/2/eb4e57bb2c34032da68dfeb3a0578b68/{word}/json |
| И25.2 | Н | Api может принять любое | False |

| | | | |
|--|--|---------------------------------|--|
| | | значение неравное 'bighugelabs' | |
|--|--|---------------------------------|--|

Тест И26. Этап 2. Взаимодействие `antonym()` → `_return_json()` → `_try_URL()`

Описание: тест проверяет правильность получения данных от стороннего модуля. Параметры передаются в функцию `antonym()`. Функция `antonym()` вызывает функцию `_return_json()`, которая вызывает функцию `_try_URL()`. Взаимодействие функций `_try_URL()` и `_return_json()` проверяется тестом И2

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|--------------|-----------|---|---|
| И26.1 | П | word – текущая фраза на английском языке | Результат должен быть представлен в формате json <pre>{ "tuc" : [{ "ключ0": [{ "ключ1": "значение1", "ключ2": "значение2", ... }, { "ключ1": "значение1", "ключ2": "значение2", ... }, ... }], "ключ1": [{ "ключ1": "значение1", "ключ2": "значение2", ... }, { "ключ1": "значение1", "ключ2": "значение2", ... }, ... }] }</pre> |
| И26.2 | Н | Один или несколько параметров не должны соответствовать формату, описанному в тесте И26.1 | False |

Тест И27. Этап 2. Взаимодействие `antonym()` → `respond()` → `_respond_with_dict()`

Описание: тест проверяет правильность преобразования обработанных данных к заданному формату. Параметры передаются в функцию `antonym()`. Функция `antonym()` вызывает функцию `respond()`, которая осуществляет вызов функции `_respond_with_dict()`. Взаимодействие функций `respond()` и `_respond_with_dict()` проверяется тестом И3.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|--------------|-----------|---|---------------------------------------|
| И27.1 | П | word – текущая фраза на английском языке Format = 'dict' | {0: 'значение1', 1: 'значение2', ...} |
| И27.2 | Н | Один или несколько параметров не должны соответствовать формату, описанному в тесте И27.1 | False |

Тест И28. Этап 2. Взаимодействие `antonym()` → `respond()` → `_respond_with_list()`

Описание: тест проверяет правильность преобразования обработанных данных к заданному формату. Параметры передаются в функцию `antonym()`. Функция `antonym()` вызывает функцию `respond()`, которая осуществляет вызов функции

`_respond_with_list()`. Взаимодействие функций `respond()` и `_respond_with_list()` проверяется тестом И4.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|---|---------------------------------|
| И28.1 | П | word – текущая фраза на английском языке Format = 'list' | ['значение1', 'значение2', ...] |
| И28.2 | Н | Один или несколько параметров не должны соответствовать формату, описанному в тесте И28.1 | False |

Тест И29. Этап 1. Взаимодействие `part_of_speech()` → `_get_api_link()`

Описание: тест проверяет правильность получения ссылки при задании соответствующего параметра `api`. Параметр `api` передается в функцию `get_api_link`. Функция `part_of_speech()` вызывает функцию `_get_api_link()`.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|--|--|
| И29.1 | П | <code>api = 'wordnik'</code> | <code>http://api.wordnik.com/v4/word.json/{word}/{action}?api_key=1e940957819058fe3ec7c59d43c09504b400110db7faa0509</code> |
| И29.2 | Н | <code>api</code> может принять любое значение отличное от <code>'wordnik'</code> | False |

Тест И30. Этап 2. Взаимодействие `part_of_speech()` → `_return_json()` → `_try_URL()`

Описание: тест проверяет правильность получения данных от стороннего модуля. Параметры передаются в функцию `part_of_speech()`. Функция `part_of_speech()` вызывает функцию `_return_json()`, которая вызывает функцию `_try_URL()`. Взаимодействие функций `_try_URL()` и `_return_json()` проверяется тестом И2

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|--|--|
| И30.1 | П | word – текущая фраза на английском языке | Результат должен быть представлен в формате json { "tuc": [{ "ключ0": [{ "ключ1": "значение1", "ключ2": "значение2", ... }, { "ключ1": "значение1", "ключ2": "значение2", ... }, ...] , "ключ1": [{ "ключ1": "значение1", "ключ2": "значение2", ... }, { "ключ1": "значение1", "ключ2": "значение2", ... }, ...] }] , ... } |
| И30.2 | Н | Один или несколько параметров не | False |

| | | | |
|--|--|--|--|
| | | должны соответствовать формату, описанному в тесте И30.1 | |
|--|--|--|--|

Тест И31. Этап 2. Взаимодействие part_of_speech() → respond() → _respond_with_dict()

Описание: тест проверяет правильность преобразования обработанных данных к заданному формату. Параметры передаются в функцию part_of_speech(). Функция part_of_speech() вызывает функцию respond(), которая осуществляет вызов функции _respond_with_dict(). Взаимодействие функций respond() и _respond_with_dict() проверяется тестом И3.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|---|---------------------------------------|
| И31.1 | П | word – текущая фраза на английском языке Format = 'dict' | {0: 'значение1', 1: 'значение2', ...} |
| И31.2 | Н | Один или несколько параметров не должны соответствовать формату, описанному в тесте И31.1 | False |

Тест И32. Этап 2. Взаимодействие part_of_speech() → respond() → _respond_with_list()

Описание: тест проверяет правильность преобразования обработанных данных к заданному формату. Параметры передаются в функцию part_of_speech(). Функция part_of_speech() вызывает функцию respond(), которая осуществляет вызов функции _respond_with_list(). Взаимодействие функций respond() и _respond_with_list() проверяется тестом И4.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|---|---------------------------------|
| И32.1 | П | word – текущая фраза на английском языке Format = 'list' | ['значение1', 'значение2', ...] |
| И32.2 | Н | Один или несколько параметров не должны соответствовать формату, описанному в тесте И32.1 | False |

Тест И33. Этап 1. Взаимодействие hyphenation() → _get_api_link()

Описание: тест проверяет правильность получения ссылки при задании соответствующего параметра api. Параметр api передается в функцию get_api_link. Функция hyphenation() вызывает функцию _get_api_link().

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|-----------------|---|
| И33.1 | П | api = 'wordnik' | http://api.wordnik.com/v4/word.json/{word}/{action}?api_key=1e940957819058fe3ec7c59d43c0950 |

| | | | |
|--------------|---|---|--------------------|
| | | | 4b400110db7faa0509 |
| ИЗ3.2 | Н | api может принять любое значение неравное 'wordnik' | False |

Тест ИЗ4. Этап 2. Взаимодействие hyphenation() → _return_json() → _try_URL()

Описание: тест проверяет правильность получения данных от стороннего модуля. Параметры передаются в функцию hyphenation(). Функция hyphenation() вызывает функцию _return_json(), которая вызывает функцию _try_URL(). Взаимодействие функций _try_URL() и _return_json() проверяется тестом ИЗ2

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|--------------|-----------|---|---|
| ИЗ4.1 | П | word – текущая фраза на английском языке | Результат должен быть представлен в формате json { "tuc": [{ "ключ0": [{ "ключ1": "значение1", "ключ2": "значение2", ... }, { "ключ1": "значение1", "ключ2": "значение2", ... }, ...] , "ключ1": [{ "ключ1": "значение1", "ключ2": "значение2", ... }, { "ключ1": "значение1", "ключ2": "значение2", ... }, ...]] , ... }] |
| ИЗ4.2 | Н | Один или несколько параметров не должны соответствовать формату, описанному в тесте ИЗ4.1 | False |

Тест ИЗ5. Этап 2. Взаимодействие hyphenation() → respond() → _respond_with_dict()

Описание: тест проверяет правильность преобразования обработанных данных к заданному формату. Параметры передаются в функцию hyphenation(). Функция hyphenation() вызывает функцию respond(), которая осуществляет вызов функции _respond_with_dict(). Взаимодействие функций respond() и _respond_with_dict() проверяется тестом ИЗ3.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|--------------|-----------|---|---------------------------------------|
| ИЗ5.1 | П | word – текущая фраза на английском языке Format = 'dict' | {0: 'значение1', 1: 'значение2', ...} |
| ИЗ5.2 | Н | Один или несколько параметров не должны соответствовать формату, описанному в тесте ИЗ5.1 | False |

Тест ИЗ6. Этап 2. Взаимодействие hyphenation() → respond() → _respond_with_list()

Описание: тест проверяет правильность преобразования обработанных данных к заданному формату. Параметры передаются в функцию `hyphenation()`. Функция `hyphenation()` вызывает функцию `respond()`, которая осуществляет вызов функции `_respond_with_list()`. Взаимодействие функций `respond()` и `_respond_with_list()` проверяется тестом И4.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|---|---------------------------------|
| ИЗ6.1 | П | word – текущая фраза на английском языке Format = 'list' | ['значение1', 'значение2', ...] |
| ИЗ6.2 | Н | Один или несколько параметров не должны соответствовать формату, описанному в тесте ИЗ6.1 | False |

Тест ИЗ7. Этап 1. Взаимодействие `pronunciation()` → `_get_api_link()`

Описание: тест проверяет правильность получения ссылки при задании соответствующего параметра `api`. Параметр `api` передается в функцию `get_api_link`. Функция `pronunciation()` вызывает функцию `_get_api_link()`.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|--|--|
| ИЗ7.1 | П | <code>api = 'wordnik'</code> | <code>http://api.wordnik.com/v4/word.json/{word}/{action}?api_key=1e940957819058fe3ec7c59d43c09504b400110db7faa0509</code> |
| ИЗ7.2 | Н | <code>api</code> может принять любое значение отличное от <code>'wordnik'</code> | False |

Тест ИЗ8. Этап 2. Взаимодействие `pronunciation()` → `_return_json()` → `_try_URL()`

Описание: тест проверяет правильность получения данных от стороннего модуля. Параметры передаются в функцию `pronunciation()`. Функция `pronunciation()` вызывает функцию `_return_json()`, которая вызывает функцию `_try_URL()`. Взаимодействие функций `_try_URL()` и `_return_json()` проверяется тестом И2

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|--|---|
| ИЗ8.1 | П | word – текущая фраза на английском языке | Результат должен быть представлен в формате json { "tuc" : [{ "ключ0": [{ "ключ1": "значение1", "ключ2": "значение2", ... }, { "ключ1": "значение1", "ключ2": "значение2", ... }, ... }], "ключ1": [{ "ключ1": |

| | | | |
|--------------|---|---|--|
| | | | “значение1”, “ключ2”: “значение2”, ... }, { “ключ1”: “значение1”, “ключ2”: “значение2”, ... }, ... }], ... }] |
| И38.2 | Н | Один или несколько параметров не должны соответствовать формату, описанному в тесте И38.1 | False |

Тест И39. Этап 2. Взаимодействие `pronunciation()` → `respond()` → `__respond_with_dict()`

Описание: тест проверяет правильность преобразования обработанных данных к заданному формату. Параметры передаются в функцию `pronunciation()`. Функция `pronunciation()` вызывает функцию `respond()`, которая осуществляет вызов функции `__respond_with_dict()`. Взаимодействие функций `respond()` и `__respond_with_dict()` проверяется тестом И3.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|--------------|-----------|---|---------------------------------------|
| И39.1 | П | word – текущая фраза на английском языке Format = 'dict' | {0: 'значение1', 1: 'значение2', ...} |
| И39.2 | Н | Один или несколько параметров не должны соответствовать формату, описанному в тесте И39.1 | False |

Тест И40. Этап 2. Взаимодействие `pronunciation()` → `respond()` → `__respond_with_list()`

Описание: тест проверяет правильность преобразования обработанных данных к заданному формату. Параметры передаются в функцию `pronunciation()`. Функция `pronunciation()` вызывает функцию `respond()`, которая осуществляет вызов функции `__respond_with_list()`. Взаимодействие функций `respond()` и `__respond_with_list()` проверяется тестом И4.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|--------------|-----------|---|---------------------------------|
| И40.1 | П | word – текущая фраза на английском языке Format = 'list' | ['значение1', 'значение2', ...] |
| И40.2 | Н | Один или несколько параметров не должны соответствовать формату, описанному в тесте И40.1 | False |

3.3 Аттестационные тесты

Тест А1.

Функциональность: Ф1

Функция: `synonym (phrase, source_lang = 'en', dest_lang = 'en', format = 'json')`

Описание: вывод синонимов слова при верно заданных параметрах `phrase`, `source_lang`, `dest_lang`, `format`.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|--|---|
| A1 | П | <code>phrase</code> – слово, заданное на языке <code>source_lang</code> <code>source_lang</code> – язык, на котором задано слово <code>phrase</code> . Язык должен быть задан в сокращенном виде, например: <code>es</code> , <code>en</code> , <code>fr</code> <code>dest_lang</code> – язык, на котором будут выведены синонимы слова. Язык должен быть задан в сокращенном виде, например: <code>es</code> , <code>en</code> , <code>fr</code> <code>format</code> – формат, в котором выводить синонимы. Значение выбирается из диапазона [<code>'json'</code> , <code>'list'</code> , <code>'dict'</code>] | вывод синонимов слова <code>phrase</code> , заданного на языке <code>source_lang</code> , на языке <code>dest_lang</code> в формате <code>format</code> . |

Тест A2.

Функциональность: Ф1

Функция: `synonym (phrase, source_lang = 'en', dest_lang = 'en', format = 'json')`

Описание: вывод значения `False` при неверно заданных параметрах `phrase`, `source_lang`, `dest_lang`, `format`.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|--|---------------------|
| A2 | Н | Один или несколько параметров не должны соответствовать формату, описанному в тесте A1 | <code>False</code> |

Тест A3.

Функциональность: Ф2

Функция: `antonym (phrase, source, format = 'json')`

Описание: вывод антонимов слова при верно заданных параметрах `phrase` и `format`.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|--|---|
| A3 | П | <code>phrase</code> – слово, заданное на английском языке <code>format</code> – формат, в котором выводить синонимы. Значение выбирается из диапазона [<code>'json'</code> , <code>'list'</code> , <code>'dict'</code>] | вывод антонимов слова <code>phrase</code> , заданного на английском языке в формате <code>format</code> . |

Тест А4.**Функциональность:** Ф2**Функция:** `antonym(phrase, format = 'json')`**Описание:** вывод значения False при неверно заданных параметрах `phrase` и `format`.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|--|---------------------|
| A4 | Н | Один или несколько параметров не должны соответствовать формату, описанному в тесте А3 | False |

Тест А5.**Функциональность:** Ф3**Функция:** `meaning(phrase, source_lang = 'en', dest_lang = 'en', format = 'json')`**Описание:** вывод определений слова при верно заданных параметрах `phrase`, `source_lang`, `dest_lang`, `format`.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|--|---|
| A5 | П | <code>phrase</code> – слово, заданное на языке <code>source_lang</code> <code>source_lang</code> – язык, на котором задано слово <code>phrase</code> . Язык должен быть задан в сокращенном виде, например: <code>es</code> , <code>en</code> , <code>fr</code> <code>dest_lang</code> – язык, на котором будут выведены синонимы слова. Язык должен быть задан в сокращенном виде, например: <code>es</code> , <code>en</code> , <code>fr</code> <code>format</code> – формат, в котором выводить синонимы. Значение выбирается из диапазона [<code>'json'</code> , <code>'list'</code> , <code>'dict'</code>] | вывод определений слова <code>phrase</code> , заданного на языке <code>source_lang</code> , на языке <code>dest_lang</code> в формате <code>format</code> . |

Тест А6.**Функциональность:** Ф3**Функция:** `meaning(phrase, source_lang = 'en', dest_lang = 'en', format = 'json')`**Описание:** вывод False при неверно заданных параметрах `phrase`, `source_lang`, `dest_lang`, `format`.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|----------------|---------------------|
|---------|-----------|----------------|---------------------|

| | | | |
|-----------|---|--|-------|
| A6 | Н | Один или несколько параметров не должны соответствовать формату, описанному в тесте A5 | False |
|-----------|---|--|-------|

Тест A7.

Функциональность: Ф4

Функция: translate (phrase, source_lang = 'en', dest_lang, format = 'json')

Описание: вывод перевода слова при верно заданных параметрах phrase, source_lang, dest_lang, format.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|----------------|------------------|--|---|
| A7 | П | phrase – слово, заданное на языке source_lang source_lang – язык, на котором задано слово phrase. Язык должен быть задан в сокращенном виде, например: es, en, fr dest_lang – язык, на котором будут выведены синонимы слова. Язык должен быть задан в сокращенном виде, например: es, en, fr format – формат, в котором выводить синонимы. Значение выбирается из диапазона ['json', 'list', 'dict'] | вывод перевода слова phrase, заданного на языке source_lang, на языке dest_lang в формате format. |

Тест A8.

Функциональность: Ф4

Функция: translate (phrase, source_lang = 'en', dest_lang, format = 'json')

Описание: вывод значения False при неверно заданных параметрах phrase, source_lang, dest_lang, format.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|----------------|------------------|--|----------------------------|
| A8 | Н | Один или несколько параметров не должны соответствовать формату, описанному в тесте A7 | False |

Тест A9.

Функциональность: Ф5

Функция: part_of_speech (phrase, format = 'json')

Описание: вывод части речи слова при верно заданных параметрах phrase, format.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|---|---|
| A9 | П | phrase – слово, заданное на английском языке format – формат, в котором выводить синонимы. Значение выбирается из диапазона ['json', 'list', 'dict'] | вывод частей речи слова phrase, заданного на английском языке в формате format. |

Тест A10.

Функциональность: Ф5

Функция: part_of_speech (phrase, format = 'json')

Описание: вывод значения False при неверно заданных параметрах phrase и format.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|--|---------------------|
| A10 | Н | Один или несколько параметров не должны соответствовать формату, описанному в тесте A9 | False |

Тест A11.

Функциональность: Ф6

Функция: usage_example (phrase, format = 'json')

Описание: вывод примеров использования слова при верно заданных параметрах phrase, format.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|---|--|
| A11 | П | phrase – слово, заданное на английском языке format – формат, в котором выводить синонимы. Значение выбирается из диапазона ['json', 'list', 'dict'] | вывод примеров слова phrase, заданного на английском языке в формате format. |

Тест A12.

Функциональность: Ф6

Функция: usage_example (phrase, format = 'json')

Описание: вывод значения False при неверно заданных параметрах phrase и format.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|---|---------------------|
| A12 | Н | Один или несколько параметров не должны соответствовать формату, описанному в тесте A11 | False |

Тест A13.

Функциональность: Ф7

Функция: pronunciation (phrase, format = 'json')

Описание: вывод части речи слова при верно заданных параметрах phrase, format.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|---|--|
| A13 | П | phrase – слово, заданное на английском языке format – формат, в котором выводить синонимы. Значение выбирается из диапазона ['json', 'list', 'dict'] | вывод произношения слова phrase, заданного на английском языке в формате format. |

Тест A14.

Функциональность: Ф7

Функция: pronunciation (phrase, format = 'json')

Описание: вывод значения False при неверно заданных параметрах phrase, source_lang, dest_lang, format.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|---|---------------------|
| A14 | Н | Один или несколько параметров не должны соответствовать формату, описанному в тесте A13 | False |

Тест A15.

Функциональность: Ф8

Функция: hyphenation (phrase, format = 'json')

Описание: вывод части речи слова при верно заданных параметрах phrase, format.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|---|--|
| A15 | П | phrase – слово, заданное на английском языке format – формат, в котором выводить синонимы. Значение выбирается из диапазона ['json', | вывод пререносов слова phrase, заданного на английском языке в формате format. |

| | | | |
|--|--|-----------------|--|
| | | 'list', 'dict'] | |
|--|--|-----------------|--|

Тест A16.

Функциональность: Ф8

Функция: hyphenation (phrase, format = 'json')

Описание: вывод значения False при неверно заданных параметрах phrase и format.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|---|---------------------|
| A16 | Н | Один или несколько параметров не должны соответствовать формату, описанному в тесте A15 | False |

3.4. Нагрузочные тесты

Тест Н1.

Тестируемый модуль: Wordnik

Тестируемая функция: part_of_speech()

Описание: нагрузочное тестирование стороннего модуля Wordnik согласно заявленным характеристикам. Заявленные характеристики: количество одновременно поступающих запросов до 100 000. Тестирование осуществляется за счет вызова функции part_of_speech().

| № теста | Тип теста | Входные данные | Косвенный данные | Ожидаемый результат |
|---------|-----------|---|--|---|
| Н1 | С | Phrase – слово на английском языке Format – значение из диапазона ['json', 'dict', 'list'] | Количество запросов в диапазоне [1, 100 000] | Функция вернет ответ от стороннего модуля |

Тест Н2.

Тестируемый модуль: Urbandict

Тестируемая функция: usage_example()

Описание: нагрузочное тестирование стороннего модуля Urbandict согласно заявленным характеристикам. Заявленные характеристики: количество одновременно поступающих запросов до 100 000. Тестирование осуществляется за счет вызова функции usage_example().

| № теста | Тип теста | Входные данные | Косвенный данные | Ожидаемый результат |
|---------|-----------|----------------|------------------|---------------------|
|---------|-----------|----------------|------------------|---------------------|

| | | | | |
|-----------|---|---|--|---|
| Н2 | С | Phrase – слово на английском языке Format – значение из диапазона ['json', 'dict', 'list'] | Количество запросов в диапазоне [1, 100 000] | Функция вернет ответ от стороннего модуля |
|-----------|---|---|--|---|

Тест Н3.

Тестируемый модуль: Glosbe

Тестируемая функция: meaning()

Описание: нагрузочное тестирование стороннего модуля Glosbe согласно заявленным характеристикам. Заявленные характеристики: количество одновременно поступающих запросов до 10 000. Тестирование осуществляется за счет вызова функции meaning().

| № теста | Тип теста | Входные данные | Косвенный данные | Ожидаемый результат |
|-----------|-----------|--|---|---|
| Н3 | С | Phrase – слово на языке source_lang Source_lang – язык, на котором задатся phrase в сокращенном виде, например en, es, fr, ... Dest_lang – язык, на котором выводить значения слова в сокращенном виде, например en, es, fr, ... Format – значение из диапазона ['json', 'dict', 'list'] | Количество запросов в диапазоне [1, 10 000] | Функция вернет ответ от стороннего модуля |

Тест Н4.

Тестируемый модуль: Bighugelabs

Тестируемая функция: antonym()

Описание: нагрузочное тестирование стороннего модуля Bighugelabs согласно заявленным характеристикам. Заявленные характеристики: количество одновременно поступающих запросов до 10 000. Тестирование осуществляется за счет вызова функции antonym()

| № теста | Тип теста | Входные данные | Косвенный данные | Ожидаемый результат |
|-----------|-----------|---|---|---|
| Н4 | С | Phrase – слово на английском языке Format – значение из диапазона ['json', 'dict', | Количество запросов в диапазоне [1, 10 000] | Функция вернет ответ от стороннего модуля |

| | | | | |
|--|--|---------|--|--|
| | | 'list'] | | |
|--|--|---------|--|--|

4 Журнал тестирования

4.1 Блочные тесты

| Номер теста | Результат | Дата проведения | Номер ошибки |
|-------------|------------|-----------------|--------------|
| Б1 | Пройден | 03.12.2018 | |
| Б2 | Пройден | 03.12.2018 | |
| Б3 | Пройден | 03.12.2018 | |
| Б4 | Пройден | 03.12.2018 | |
| Б5.1 | Пройден | 03.12.2018 | |
| Б5.2 | Пройден | 03.12.2018 | |
| Б6.1 | Не пройден | 03.12.2018 | 1 |
| Б6.2 | Пройден | 03.12.2018 | |
| Б7 | Пройден | 03.12.2018 | |
| Б8.1 | Не пройден | 03.12.2018 | 2 |
| Б8.2 | Пройден | 03.12.2018 | |
| Б9.1 | Пройден | 03.12.2018 | |
| Б9.2 | Пройден | 03.12.2018 | |
| Б10.1 | Пройден | 03.12.2018 | |
| Б10.2 | Не пройден | 03.12.2018 | 3 |
| Б11.1 | Пройден | 03.12.2018 | |
| Б11.2 | Пройден | 03.12.2018 | |
| Б12.1 | Пройден | 03.12.2018 | |
| Б12.2 | Пройден | 03.12.2018 | |
| Б13.1 | Пройден | 03.12.2018 | |
| Б13.2 | Пройден | 03.12.2018 | |
| Б14.1 | Пройден | 03.12.2018 | |
| Б14.2 | Пройден | 03.12.2018 | |
| Б15.1 | Пройден | 03.12.2018 | |
| Б15.2 | Пройден | 03.12.2018 | |
| Б16.1 | Пройден | 03.12.2018 | |
| Б16.2 | Не пройден | 03.12.2018 | 4 |
| Б17.1 | Пройден | 03.12.2018 | |
| Б17.2 | Пройден | 03.12.2018 | |
| Б18.1 | Пройден | 03.12.2018 | |
| Б18.2 | Пройден | 03.12.2018 | |

4.2 Интеграционные тесты

| Номер теста | Результат | Дата проведения |
|-------------|-----------|-----------------|
| И1 | Пройден | 04.12.2018 |
| И2 | Пройден | 04.12.2018 |
| И3.1 | Пройден | 04.12.2018 |
| И3.2 | Пройден | 04.12.2018 |
| И4.1 | Пройден | 04.12.2018 |
| И4.2 | Пройден | 04.12.2018 |
| И5.1 | Пройден | 04.12.2018 |
| И5.2 | Пройден | 04.12.2018 |
| И6.1 | Пройден | 04.12.2018 |
| И6.2 | Пройден | 04.12.2018 |
| И7.1 | Пройден | 04.12.2018 |

| | | |
|-------|---------|------------|
| И7.2 | Пройден | 04.12.2018 |
| И8.1 | Пройден | 04.12.2018 |
| И8.2 | Пройден | 04.12.2018 |
| И9.1 | Пройден | 04.12.2018 |
| И9.2 | Пройден | 04.12.2018 |
| И10.1 | Пройден | 04.12.2018 |
| И10.2 | Пройден | 04.12.2018 |
| И11.1 | Пройден | 04.12.2018 |
| И11.2 | Пройден | 04.12.2018 |
| И12.1 | Пройден | 04.12.2018 |
| И12.2 | Пройден | 04.12.2018 |
| И13.1 | Пройден | 04.12.2018 |
| И13.2 | Пройден | 04.12.2018 |
| И14.1 | Пройден | 04.12.2018 |
| И14.2 | Пройден | 04.12.2018 |
| И15.1 | Пройден | 04.12.2018 |
| И15.2 | Пройден | 04.12.2018 |
| И16.1 | Пройден | 04.12.2018 |
| И16.2 | Пройден | 04.12.2018 |
| И17.1 | Пройден | 04.12.2018 |
| И17.2 | Пройден | 04.12.2018 |
| И18.1 | Пройден | 04.12.2018 |
| И18.2 | Пройден | 04.12.2018 |
| И19.1 | Пройден | 04.12.2018 |
| И19.2 | Пройден | 04.12.2018 |
| И20.1 | Пройден | 04.12.2018 |
| И20.2 | Пройден | 04.12.2018 |
| И21.1 | Пройден | 04.12.2018 |
| И21.2 | Пройден | 04.12.2018 |
| И22.1 | Пройден | 04.12.2018 |
| И22.2 | Пройден | 04.12.2018 |
| И23.1 | Пройден | 04.12.2018 |
| И23.2 | Пройден | 04.12.2018 |
| И24.1 | Пройден | 04.12.2018 |
| И24.2 | Пройден | 04.12.2018 |
| И25.1 | Пройден | 04.12.2018 |
| И25.2 | Пройден | 04.12.2018 |
| И26.1 | Пройден | 04.12.2018 |
| И26.2 | Пройден | 04.12.2018 |
| И27.1 | Пройден | 04.12.2018 |
| И27.2 | Пройден | 04.12.2018 |
| И28.1 | Пройден | 04.12.2018 |
| И28.2 | Пройден | 04.12.2018 |
| И29.1 | Пройден | 04.12.2018 |
| И29.2 | Пройден | 04.12.2018 |
| И30.1 | Пройден | 04.12.2018 |
| И30.2 | Пройден | 04.12.2018 |
| И31.1 | Пройден | 04.12.2018 |
| И31.2 | Пройден | 04.12.2018 |

| | | |
|-------|---------|------------|
| И32.1 | Пройден | 04.12.2018 |
| И32.2 | Пройден | 04.12.2018 |
| И33.1 | Пройден | 04.12.2018 |
| И33.2 | Пройден | 04.12.2018 |
| И34.1 | Пройден | 04.12.2018 |
| И34.2 | Пройден | 04.12.2018 |
| И35.1 | Пройден | 04.12.2018 |
| И35.2 | Пройден | 04.12.2018 |
| И36.1 | Пройден | 04.12.2018 |
| И36.2 | Пройден | 04.12.2018 |
| И37.1 | Пройден | 04.12.2018 |
| И37.2 | Пройден | 04.12.2018 |
| И38.1 | Пройден | 04.12.2018 |
| И38.2 | Пройден | 04.12.2018 |
| И39.1 | Пройден | 04.12.2018 |
| И39.2 | Пройден | 04.12.2018 |
| И40.1 | Пройден | 04.12.2018 |
| И40.2 | Пройден | 04.12.2018 |

4.3 Аттестационные тесты

| Номер теста | Результат | Дата проведения |
|-------------|-----------|-----------------|
| A1 | Пройден | 05.12.2018 |
| A2 | Пройден | 05.12.2018 |
| A3 | Пройден | 05.12.2018 |
| A4 | Пройден | 05.12.2018 |
| A5 | Пройден | 05.12.2018 |
| A6 | Пройден | 05.12.2018 |
| A7 | Пройден | 05.12.2018 |
| A8 | Пройден | 05.12.2018 |
| A9 | Пройден | 05.12.2018 |
| A10 | Пройден | 05.12.2018 |
| A11 | Пройден | 05.12.2018 |
| A12 | Пройден | 05.12.2018 |
| A13 | Пройден | 05.12.2018 |
| A14 | Пройден | 05.12.2018 |
| A15 | Пройден | 05.12.2018 |
| A16 | Пройден | 05.12.2018 |

4.4 Нагрузочные тесты

| Номер теста | Результат | Дата проведения | Количество запросов |
|-------------|-----------|-----------------|---------------------|
| Н1 | Пройден | 05.12.2018 | 10 |
| | Пройден | | 100 |
| | Пройден | | 1 000 |
| | Пройден | | 10 000 |
| | Пройден | | 100 000 |
| Н2 | Пройден | 05.12.2018 | 10 |
| | Пройден | | 100 |
| | Пройден | | 1 000 |
| | Пройден | | 10 000 |

| | | | |
|----|---------|------------|---------|
| | Пройден | | 100 000 |
| Н3 | Пройден | 05.12.2018 | 10 |
| | Пройден | | 100 |
| | Пройден | | 1 000 |
| | Пройден | | 10 000 |
| | Пройден | | |
| Н4 | Пройден | 05.12.2018 | 10 |
| | Пройден | | 100 |
| | Пройден | | 1 000 |
| | Пройден | | 10 000 |

5 Журнал найденных ошибок

Ошибка 1.

Дата составления: 03.12.2018

Номер теста: Б6.1

Описание теста: проверка выполнения условия. Если `content_to_be_parsed` принимает значение из диапазона ['text', 'phrase'], то получить из словаря значения с ключом 'text' или 'phrase' и записать их в новый словарь.

Входные данные: `content_to_be_parsed = 'text'`

Ожидаемый результат: {0: 'The process of breaking a word between syllables at the end of a line or to separate the parts of a compound word', 1: 'division of a word especially at the end of a line on a page', 2: 'connecting syllables and words by hyphens'}

Фактический результат: False

Ошибка 2.

Дата составления: 03.12.2018

Номер теста: Б8.1

Описание теста: проверка выполнения цикла. Для каждого ключа из `dictionary` прописать новое значение 'seq' и записать ключ и значение в новый словарь

Входные данные: `dictionary = {0: 'The process of breaking a word between syllables at the end of a line or to separate the parts of a compound word', 1: 'division of a word especially at the end of a line on a page', 2: 'connecting syllables and words by hyphens'}`

Ожидаемый результат: {0: 'seq', 1: 'seq', 2: 'seq'}

Фактический результат: {0: 'seq', 1: "", 2: 'seq'}

Ошибка 3.

Дата составления: 03.12.2018

Номер теста: Б10.2

Описание теста: проверка выполнения цикла. Склеить два словаря с ключами и значениями в один список.

Входные данные: value_dict = {'The process of breaking a word between syllables at the end of a line or to separate the parts of a compound word': 'text', 'division of a word especially at the end of a line on a page': 'text', 'connecting syllables and words by hyphens': 'text'}, key_dict = {}

Ожидаемый результат: False

Фактический результат: [{ 'seq': 0, 'text': 'The process of breaking a word between syllables at the end of a line or to separate the parts of a compound word'}, {'seq': 1, 'text': 'division of a word especially at the end of a line on a page'}, {'seq': 2, 'text': 'connecting syllables and words by hyphens'}]

Ошибка 4.

Дата составления: 03.12.2018

Номер теста: Б16.2

Описание теста: проверка выполнения условия. Если полученные данные от стороннего модуля являются json – объектом, то выделить необходимые данные из полученных данных, иначе вывести ошибку

Входные данные: {}

Ожидаемый результат: False

Фактический результат: исключение KeyError

6 Пример теста

Тест А5.

Функциональность: ФЗ

Функция: meaning (phrase, source_lang = 'en', dest_lang = 'en', format = 'json')

Описание: вывод определений слова при верно заданных параметрах phrase, source_lang, dest_lang, format.

| № теста | Тип теста | Входные данные | Ожидаемый результат |
|---------|-----------|---|--|
| А5 | П | phrase – слово, заданное на языке source_lang source_lang – язык, на котором задано слово phrase. Язык должен быть задан в сокращенном виде, | вывод определений слова phrase, заданного на языке source_lang, на языке dest_lang в формате format. |

| | | |
|--|---|--|
| | <p>например: es, en, fr</p> <p>dest_lang – язык, на котором будут выведены синонимы слова. Язык должен быть задан в сокращенном виде, например: es, en, fr</p> <p>format – формат, в котором выводить синонимы. Значение выбирается из диапазона ['json', 'list', 'dict']</p> | |
|--|---|--|

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

from vocabulary.vocabulary import Vocabulary as vb
from vocabulary.response import Response as rp
import unittest
import sys

try:
    import simplejson as json
except ImportError:
    import json
try:
    from unittest import mock
except Exception as e:
    import mock

class TestModule(unittest.TestCase):
    """Checks for the sanity of all module methods"""

    @mock.patch('vocabulary.vocabulary.requests.get')
    def test_meaning_found(self, mock_api_call):
        res = {
            "tuc": [
                {
                    "meanings": [
                        {
                            "language": "en",
                            "text": "the act of singing with closed lips"
                        }
                    ]
                }
            ]
        }

        mock_api_call.return_value = mock.Mock()
        mock_api_call.return_value.status_code = 200
        mock_api_call.return_value.json.return_value = res

        expected_result = '[{"seq": 0, "text": "the act of singing with closed lips"}]'
        expected_result = json.dumps(json.loads(expected_result))
        result = vb.meaning("humming", "en", "en", "json")

        if sys.version_info[:2] <= (2, 7):
            self.assertEqual(expected_result, result)
        else:
            self.assertCountEqual(expected_result, result)
```

7 Покрытие тестов

Расчёт покрытия тестами относительно исполняемого кода производится по формуле:

$\text{Covering} = \text{Testing_length} / \text{Code_length}$.

Testing_length - количество строк кода, покрытых тестами = 410

Code_length - общее количество строк кода в программе = 508

Коэффициент покрытия: 0,807

Процент покрытия: 80,7%