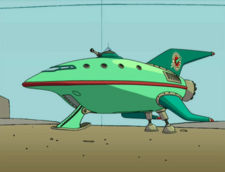


У Бендера отказал модуль отвечающий за правильное написания текста и вам необходимо помочь ему пофиксить эту проблему прикрутив к нему веб-сервис “YandexSpeller”.



Следуя **обязательным**указаниям ниже мы поможем ему:

1. Используй корабль “[ГуГлExpress](https://google.com/)” чтобы найти WSDL файл к веб-сервису “YandexSpeller” и инструкции к нему.

<https://yandex.ru/dev/speller/doc/dg/concepts/api-overview.html>

**2.** Найди операции, которые применяются в этом сервисе и используй их!

a. Чем отличаются операции, используемые в сервисе? (Ответ на этот вопрос напиши в письме с выполненным заданием)

b. Как отправить несколько фрагментов текста, чтобы получить их в разных массивах (как в примере ниже)? (Скопируй свою структуру **запроса**(да да, не ответ) и отправь в письме)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28 | <soap:Envelope xmlns:xsi="<http://www.w3.org/2001/XMLSchema-instance>" xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" xmlns:soap="<http://www.w3.org/2003/05/soap-envelope>">     <soap:Body>        <CheckTextsResponse xmlns="<http://speller.yandex.net/services/spellservice>">           <ArrayOfSpellResult>              <SpellResult>                 <error code="1" pos="9" row="0" col="9" len="7">                    <word>виззгом</word>                    <s>визгом</s>                 </error>                 <error code="1" pos="49" row="0" col="49" len="4">                    <word>дким</word>                    <s>диким</s>                 </error>              </SpellResult>              <SpellResult>                 <error code="1" pos="10" row="0" col="10" len="12">                    <word>рок-свёздами</word>                    <s>рок-звездами</s>                 </error>                 <error code="1" pos="45" row="0" col="45" len="6">                    <word>вломом</word>                    <s>взломом</s>                 </error>              </SpellResult>           </ArrayOfSpellResult>        </CheckTextsResponse>     </soap:Body>  </soap:Envelope> |

**3.** Создай новый проект в Soap UI на основе найденного WSDL файла.

a. Имя проекта должно содержать фамилию сотрудника, выполняющего задание (e.g.  “yandexSpeller [b.rodriguez]”)

**4.** Создай несколько отдельных тест сьют для будущих тест-кейсов. Бендер у нас парень современный, поэтому работает с запросами для версии SOAP 1.2. Используй именно эту версию.

a. Названия сьют должны иметь вид "Smoke", "MAT", “AT”

**5.** Проведи полное функциональное тестирование метода **CheckTexts**в рамках "Smoke", "MAT", "AT"

a. Помним про негласное правило 1 проверка = 1 кейс. Название каждого кейса обязательно должно отображать, что проверяется в данном кейсе (e.g. “*Lang” parameter - Check russian text with errors returns suggested corrections*). Названия проверок/кейсов типа *"testcase1" или “123asjdka”* не принимаются.

**6.** И ещё некоторые обязательные условия при выполнении миссии:

a. Использование “Properties”

b. Использование “Assertions”. Использовать assertions из списка ниже как минимум в каждой из первых 5 проверок/запросов.

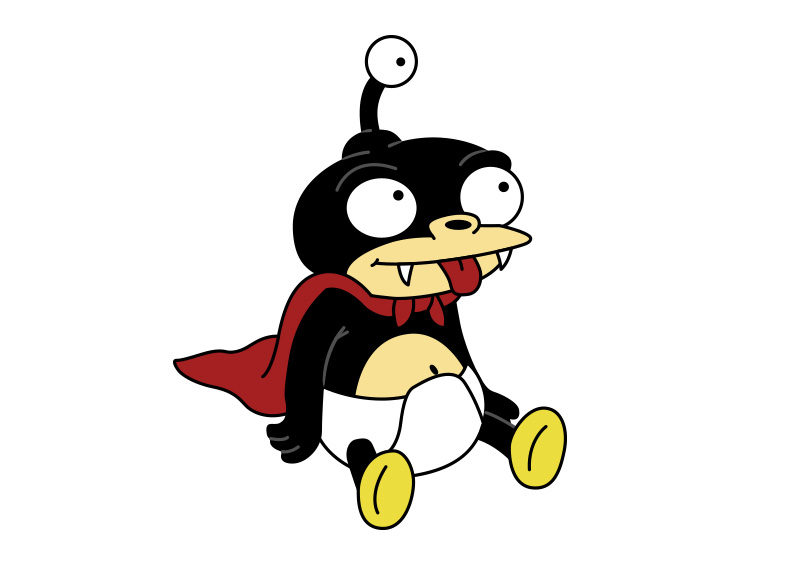
i. XPath Match

ii. Valid/Invalid HTTP Status Codes

iii. Response SLA

iv. Contains/Not Contains

c. В письме с выполненным заданием кратко описать, что проверяют следующие ассерты: Contains/Not Contains, Valid/Invalid HTTP Status Codes, XPath Match, SOAP Fault, Response SLA



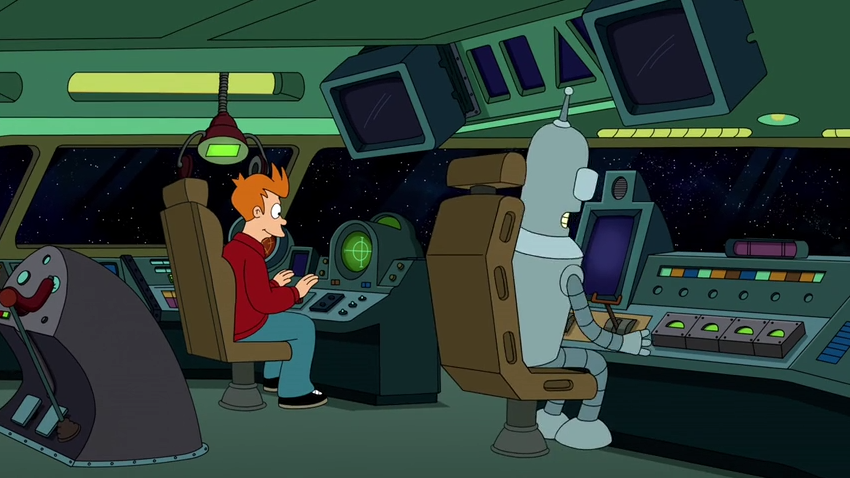
**7.** Эй, псс. Этот мелкий пухляш Ниблер хочет задать тебе несколько вопросов, которые должны помочь в генерации проверок. Ответы на эти вопросы напиши в письме с выполненным заданием :)

* 1. Какие входные параметры могут использоваться в запросе “YandexSpeller”?
  2. Перечислите коды ошибок, которые он может возвращать?
  3. Как заставить сервис выполнять проверки для трёх языков сразу?
  4. Можно ли использовать одновременно несколько опций в параметрах, в одном запросе? Если можно, то как это сделать? Список доступных опций есть в документации к сервису.

**8.** Не зли Бендера и покажи ему, где он ошибся. Создай отдельный тест сьют c названием “Angry Bender”. В нём создай отдельный тест кейс с проверками используя XPath Match и то, что он тебе наговорил в порыве злости:

*“Oh, no methods for Bender, huh? Fivne! I’ll go build my odwn API’s request, with blackack and hookers. In fauct, forgmat the API’s and the blackjack. Ah, screw the whole thing!”*

а. Бендер ожидает увидеть 5 правильных ассертов (слов) на панели приборов “Assertions” в одном тест кейсе, которые ему необходимо использовать. Хорошо, что у вас есть список слов: *Fine, own, blackjack, fact, forget*



[Подсмотреть вариант решения похожей задачи](https://www.google.com/url?q=https://community.smartbear.com/t5/SoapUI-Open-Source/SOAPUI-Property-Transfer-XPATH/m-p/190338&sa=D&ust=1580462934536000)

b. *(опционально)* Роботы-марсиане оставили нам табличку с секретным посланием. Это очень похоже на base64. Бендер рассказал, что это один XPath, который проверяет наличие сразу 5 исправлений в ответе! Проверь, что расшифрованное тобой послание является истиной и переделай его для проверки конкретных значений из списка слов задания 8а. Для данного задания создай отдельный тест кейс в тест сьюте “Angry Bender”

*ZXhpc3RzKC8vbnMxOmVycm9yWzFdL25zMTpzW3RleHQoKT0iVGV4dDEiXSkgYW5kIApleGlzdHMoLy9uczE6ZXJyb3JbMl0vbnMxOnNbdGV4dCgpPSJUZXh0MiJdKSBhbmQgCmV4aXN0cygvL25zMTplcnJvclszXS9uczE6c1t0ZXh0KCk9IlRleHQzIl0pIGFuZCAKZXhpc3RzKC8vbnMxOmVycm9yWzRdL25zMTpzW3RleHQoKT0iVGV4dDQiXSkgYW5kIApleGlzdHMoLy9uczE6ZXJyb3JbNV0vbnMxOnNbdGV4dCgpPSJUZXh0NSJdKQ==*





**9.** *(опционально)* Взять дополнительные миссии повышенной сложности у Скраффи:

* 1. Создать тест кейс в отдельной тест сьюте с использованием Script Assertion. Тест сьют должен называться “ScriptAssert”. В Script Assertion должна производиться проверка значения заголовка “Server”. Ожидаемый результат: *“nginx/1.6.2”. (Но если есть желание проверить другой заголовок - фил фри, ничего не имеем против)*
  2. Делаем цикличный тест кейс в отдельном тест сьюте (на 5 итераций) с использованием Groovy Script. Тест сьют должен называться “Loop”. Т.к. Бендер предпочитает что-нибудь, что горит, то передаваемые данные будут следующими:*["Пево", "Beeer", "Абсинт", "Самагон", "wisky"]*.  
     Также в этот тест кейс необходимо добавить XPath Assertion, который соответственно будет проверять наличие следующих предлагаемых исправлений в ответе:*["Пиво", "Beer", "Абсент", "Самогон", "Whisky"]*. XPath Assertion должен быть только один, а его ожидаемый результат должен динамически изменяться при каждом цикле тест кейса.  
     И помните, чрезмерное употребление вредит здоровью :)

**FAQ**

**Q1: Не совсем понятно как проводить тестирование нужного метода (какие проверки необходимо делать, что обычно проверяется)?**

A: При тестировании веб-сервисов проверяется тоже самое, что и при функциональном тестировании через графический интерфейс (можно использовать [web-testing checklist](https://wiki.itransition.com/display/A1QACEWT/Check+List)), только добавляются проверки на структуру тела запроса, headers и валидация полученного ответа. Примерно должно получиться такое количество кейсов: Smoke: 1, MAT: 20+, AT: 25+

**Q2: Как я могу отправить несколько текстов используя метод checkTexts?**

А: Для отправки нескольких текстов следует добавить новые spel:text теги с текстом

**Q3: При проверке "plain"="html" параметра, в ответе никак не обрабатываются теги, это баг или так должно быть?**

А: Чтобы отправить тэги в запросе их необходимо кодировать, так как в обратном случае (если отправлять не закодированные теги <b>bold text</b>) на сервер отправится не строка, а вложенный объект, который будет игнорироваться. Можно воспользоваться, например, следующим сервисом: <https://www.web2generators.com/html-based-tools/online-html-entities-encoder-and-decoder>

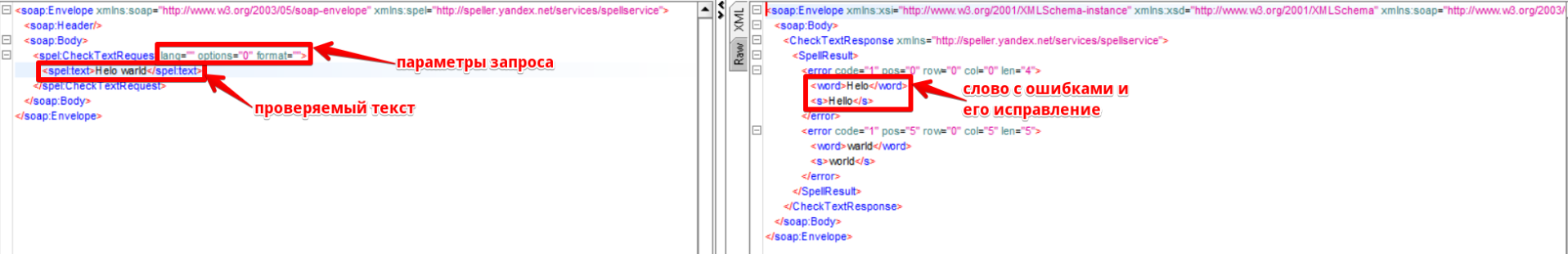
**Q4: Могу ли я использовать contains assertion для проверки атрибута тега в ответе?**

А: Для проверки конкретных значений в ответе лучше использовать XPath вместо Contains, так как contains ищет любое совпадение в ответе, а XPath ищет конкретное совпадение в указанном элементе.

Например, так можно проверить атрибут тэга при помощи XPath Match: //ns1:error/@code.

 Пример с кейсами для тестирования веб-сервиса

В данном примере рассматривается тестирование метода checkText YandexSpeller веб-сервиса:



1. Для Smoke теста в данном случае достаточно проверить что сервис работает и обрабатывает запрос, т.е. отправить текст с ошибкой и посмотреть что приходит ответ
2. В рамках МАТ тестирования необходимо покрыть все позитивные сценарии для данного метода:
   1. для отправляемого текста (можно использовать [web-testing checklist](https://wiki.itransition.com/display/A1QACEWT/Check+List))
   2. для всех используемых параметров: lang, options, format
   3. правильность обработки слова с ошибками в ответе (проверка атрибутов error тега (в котором находится слово с ошибкой и его исправление): code, pos, row, col, len)
3. В рамках АТ тестирования необходимо покрыть все негативные сценарии для данного метода:
   1. для отправляемого текста (можно использовать [web-testing checklist](https://wiki.itransition.com/display/A1QACEWT/Check+List))
   2. для всех используемых параметров: lang, options, format (отправка запроса с невалидными значениями параметров/разными типами данных параметров (null, integer))
   3. для структуры тела запроса: замена тэга на самозакрывающийся, отправка пустого тела, отправка тела без обязательных тегов, отправка тела с невалидными параметрами и т.д.

Перед выполнением домашнего задания рекомендуем **внимательно**изучить всю документацию к тестируемому сервису, составить простейший чек лист c проверками и уже по нему делать тест кейсы в SoapUI (одна проверка = один тест кейс).